This dissertation has been
microfilmed exactly as received 67-17,827

PIERSON, Bion Lee, 1939-
DIRECT METHODS FOR DISCRETE-VARIABLE
APPROXIMATION TO OPTIMAL FLIGHT PATHS.

The University of Michigan, Ph.D., 1967
Engineering, aeronautical

University Microfilms, Inc., Ann Arbor, Michigan

# DIRECT METHODS FOR DISCRETE-VARIABLE APPROXIMATION TO OPTIMAL FLIGHT PATHS

by

Bion Lee Pierson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in the
University of Michigan
1967

Doctoral Committee:

Assistant Professor Alfred C. Robinson, Chairman
Professor Harm Buning
Professor Elmer G. Gilbert
Professor Donald T. Greenwood
Professor Robert M. Thrall
Associate Professor Franklin H. Westervelt

## Abstract

DIRECT METHODS FOR DISCRETE-VARIABLE APPROXIMATION TO OPTIMAL FLIGHT PATHS

by Bion Lee Pierson

This thesis develops computational procedures that are applicable to a wide variety of flight path optimization problems. The approach taken here is quite different from methods previously described in the literature. We seek a set of values that approximate the optimal solution only at discrete time points and not an approximate solution over a continuous time interval. The term direct methods is used because the necessary conditions of the variational theory are not employed.

The problem to be considered may be stated: find an admissible control $u(t)$ for $0 \leq t \leq T$ which will minimize a given scalar function of the end values $J(T,x(T))$ subject to a vector differential equation $\dot{x} = f(t,x,u)$ with $x(0) = x_o$, a vector inequality constraint $R(t,x,u) \geq 0$, and possibly some specified terminal boundary conditions $Q(T,x(T)) = 0$. A timewise discretization process in which each differential equation is replaced by an appropriate set of difference equations is used to approximate the continuous optimal control problem by an ordinary constrained minimization problem. Several types of difference equations are investigated. Instead of attempting to solve this constrained problem, however, we solve a sequence of unconstrained problems in

which the constraints are adjoined to the cost function J as penalty functions. Conjugate direction methods are particularly attractive for the unconstrained minimizations since they offer quadratic terminal convergence without requiring the calculation of second partial derivatives. Two such methods are investigated: the variable metric method of Davidon (1959) and the method of conjugate gradients originated by Hestenes and Stiefel (1952). The latter is decisively superior for the large number of variables involved. Each of these methods requires a linear minimization at each step of the iteration; three linear minimization schemes are studied.

A large portion of the thesis is devoted to the presentation of several example problems. For some of the problems analytical solutions are known and are used to check the accuracy of the approximate method. The example problems are selected to demonstrate the various capabilities of the method. In particular, two problems involving state space constraints are solved.

In spite of the disadvantages of dealing with a large number of variables, the presence of discretization errors, and the detrimental effect of round-off errors on terminal convergence, this discrete-variable approach contains several appealing features. In the first place it appears that the amount of computer programming effort is relatively low. In addition, crude initial guesses usually suffice to start the iteration process. Also, we are able to avoid

the stability problems experienced by indirect methods that treat the two-point boundary value problem as a series of initial-value problems. Variable-time problems are handled by simply considering the final time as an additional problem variable. Other system parameters may be optimized as well. If the actual optimal control is only piecewise continuous and the number of discontinuities is known in advance, each subarc can be discretized individually. If the switching times are unknown, they can be included as problem variables. Finally, state variable inequality constraints are in principle no more difficult to handle than are constraints placed on the control.

## Acknowledgments

The author wishes to express his gratitude to the members of his committee for their interest and encouragement during the course of this research. In particular, the thoughtful suggestions and criticisms of the committee chairman, A.C. Robinson, were most helpful.

The long and tedious task of typing the several rough drafts and this final manuscript was cheerfully and ably performed by the author's wife, Marcia. Her aid in editing the manuscript was also invaluable.

The author is also indebted to the University of Michigan for its support of the computing activity so necessary to this study.

## Table of Contents


| | Page |
|---|---|
| Acknowledgments | ii |
| List of Tables | v |
| List of Figures | vii |
| Chapter 1 Introduction | 1 |
| Chapter 2 Reduction of an Optimal Control Problem to a Problem in Ordinary Minimization | 8 |
| 2.1 Statement of a Continuous Problem | 8 |
| 2.2 The Discretization Process | 10 |
| 2.2.1 The Choice of Difference Equations | 11 |
| 2.2.2 Treatment of Inequality Constraints and Boundary Conditions | 22 |
| 2.2.3 Statement of the Discrete Problem | 23 |
| 2.3 Reduction to a Sequence of Unconstrained Ordinary Minimization Problems | 25 |
| Chapter 3 The Choice of Minimization Procedures | 28 |
| 3.1 The Variable Metric Method | 32 |
| 3.2 The Method of Conjugate Gradients | 33 |
| 3.3 Comparisons | 35 |
| Chapter 4 The Choice of Linear Minimization Procedures | 38 |
| 4.1 Modified Fibonacci Search | 39 |
| 4.2 Modified Newton Method | 41 |
| 4.3 Cubic Interpolation | 42 |
| 4.4 Comparisons | 43 |
| Chapter 5 A Discussion of the Computational Algorithm | 45 |
| 5.1 The Algorithm | 45 |
| 5.2 The Effect of LINMIN Accuracy | 50 |
| 5.3 The Effect of Round-Off Errors | 53 |
| 5.4 The Effect of the Differencing Scheme Used | 57 |

Page

5.5 The Use of Unequally Spaced Time Points 60

5.6 Inequality Constraints: $\psi^2$ versus $\psi^4$ 62

Chapter 6 Example Problems 65

6.1 The Brachistochrone Problem 65

6.1.1 An Unconstrained Brachistochrone Problem 66

6.1.2 A Constrained Brachistochrone Problem 71

6.2 An Example: Avoidance of Certain Stability Difficulties 77

6.3 A Simple Rocket Problem 81

6.4 A Minimum-Time Orbital Transfer Problem 89

6.5 An Example Problem with Discontinuous Optimal Control 99

6.6 A Two-Stage Booster Problem 101

6.6.1 The Unconstrained Solution 106

6.6.2 A Constraint on the Dynamic Pressure 109

6.6.3 An Altitude Constraint 116

Chapter 7 Summary 119

Bibliography 124

## List of Tables


| Table | | Page |
|---|---|---|
| 2-1 | The Coefficients $\alpha_{jpk}$ for Adams-Class Difference Equations | 16 |
| 2-2 | Local Truncation Error Coefficients $C_p$ | 18 |
| 3-1 | Gradient Methods: A Comparison Based on Function Values of a Quadratic Form | 36 |
| 3-2 | Comparisons of the Conjugate Gradients and Variable Metric Methods Based on Cumbersome Non-Quadratic Functions | 37 |
| 4-1 | Comparison of Linear Minimization Methods | 44 |
| 5-1 | The Effect of LINMIN Accuracy on Convergence | 51 |
| 5-2 | Comparison of Termination Criteria for the Cubic Interpolation | 52 |
| 5-3 | A Comparison of Single and Double Precision Computations in Terms of the Number of Iterations Required | 56 |
| 5-4 | Accuracy Comparison of First, Second, and Third Order Adams Interpolation Differencing Schemes | 59 |
| 5-5 | Comparison of Three Differencing Schemes on an Orbital Transfer Problem | 60 |
| 5-6 | Comparison of Inequality Constraint Handling Based on the Number of Iterations Required | 63 |
| 6-1 | Unconstrained Brachistochrone Problem: An Accuracy Comparison for ETA = .002 | 70 |
| 6-2 | Constrained Brachistochrone Problem: $x_N$ Obtained for Each K-min | 72 |
| 6-3 | Constrained Brachistochrone Problem: An Accuracy Comparison | 76 |
| 6-4 | A Simple Rocket Problem: An Accuracy Comparison | 86 |
| 6-5 | State Variables for the Orbital Transfer Problem: A Comparison of Double Precision Results | 98 |
| 6-6 | Two-Stage Booster Problem: Altitude and Velocity Points with (C) and without (U) the Dynamic Pressure Constraint | 115 |

Table | Page

6-7 Comparison: Two-Stage Booster Problem with (C) and without (U) the Altitude Constraint 118

## List of Figures


| Figure | | Page |
|---|---|---|
| 2-1 | An Example of Diverging Solution Points: Orbital Transfer Problem with Third-Order Milne-Simpson Differencing Scheme | 21 |
| 4-1 | The Basic Fibonacci Iteration | 40 |
| 5-1 | Basic Flow Chart of the Minimization Algorithm | 46 |
| 6-1 | Unconstrained Brachistochrone Problem: Optimal Trajectory | 68 |
| 6-2 | Unconstrained Brachistochrone Problem: Optimal Control Time History | 69 |
| 6-3 | Constrained Brachistochrone Problem: Optimal Trajectory | 75 |
| 6-4 | Constrained Brachistochrone Problem: Optimal Control Time History | 75 |
| 6-5 | Example Problem: Minimize $\int_0^1 (x^2+u^2)\,dt$ Subject to $\dot{x} = 20x + u$, $x(0) = 1$, $x(1) = 0$ | 80 |
| 6-6 | A Simple Rocket Problem: Trajectory Geometry | 82 |
| 6-7 | A Simple Rocket Problem: Optimal Trajectory | 84 |
| 6-8 | A Simple Rocket Problem: Optimal Trajectory Hodograph | 84 |
| 6-9 | A Simple Rocket Problem: Optimal Control Time History | 85 |
| 6-10 | Minimum-Time Earth-to-Mars Trajectory | 92 |
| 6-11 | Minimum-Time Earth-to-Mars Transfer: Optimal Control Time History | 95 |
| 6-12 | Minimum-Time Earth-to-Mars Orbital Transfer: Trajectory Time History | 96 |
| 6-13 | Control Time History for the Orbital Transfer Problem: Double Precision Results | 97 |
| 6-14 | A Minimum-Time Regulator Problem: Optimal Control Time History | 100 |
| 6-15 | A Minimum-Time Regulator Problem: Optimal Trajectory | 100 |

| Figure | | Page |
|---|---|---|
| 6-16 | Two-Stage Booster Problem: Trajectory Geometry | 103 |
| 6-17 | Two-Stage Booster Problem: Altitude and Velocity Time Histories | 107 |
| 6-18 | Boost Problem: $\theta$ and u Time Histories | 108 |
| 6-19 | Dynamic Pressure Constraint: h-v Diagram | 110 |
| 6-20 | Boost Problem with Dynamic Pressure Constraint: Dynamic Pressure versus Altitude | 113 |
| 6-21 | Boost Problem with Dynamic Pressure Constraint: Comparison of Control Points during First Burn | 114 |

# CHAPTER 1

## INTRODUCTION

During the last decade engineering research has been marked by a rapid increase of interest in optimization. Much of the motivation for this interest has resulted from the stringent demands placed on modern aerospace systems. Historically, much of the early work in optimization was directed toward problems of flight path optimization. With the accompanying rapid development of high-speed digital computers, more realistic, and thus more difficult, flight path optimization problems have come under attack.

Much of the recent research in this area has been concerned with the development of simple and efficient numerical methods for handling such problems. It is clear that considerable innovation, experimentation, and comparison will be required before the practicing engineer can select with any degree of confidence the method best suited to a given problem. What qualities does one look for when appraising a particular method? A good computational method would hopefully be characterized by (a) the ease with which it may be programmed, (b) modest demands on computer storage and time, (c) a fast and predictable convergence rate, (d) a useful error estimate or bound at the end of each iteration,

and (e) a flexibility that allows application to a wide range of problems.

This last attribute is of particular concern to those interested in trajectory optimization. Characteristically, the more elegant methods for solving optimal control problems exploit special properties inherent in some restricted class of problems. The restrictions of system linearity and certain convexity requirements, for example, are quite common. Unfortunately, most realistic flight path optimization problems possess few, if any, special properties to aid the development of efficient computational methods. Typically, such problems involve systems of highly nonlinear differential equations, inequality constraints which may not explicitly include the control functions, optimal controls that may be discontinuous, and possibly, additional control parameters.

It is common practice to loosely categorize optimization methods as either indirect or direct methods. The indirect methods seek to satisfy the necessary conditions arising from an application of the calculus of variations (B1, B2) or, alternately, the Pontryagin maximum principle (P1). This approach inevitably leads to a nonlinear two-point boundary value problem. One obvious possibility of solving this problem is to treat it as a sequence of initial-value problems. Using this scheme, one would guess values for the missing boundary conditions at the initial point, integrate numerically the system of equations until some stopping condition is met, and systematically alter the

unspecified initial conditions using knowledge of the terminal boundary condition errors. However, this process has met with only limited success because of the extreme sensitivity to small changes in the initial conditions (B3, B4, K1).

More recently, McGill and Kenneth (M1, K2) have developed a generalized Newton-Raphson method. With this approach the nonlinear two-point boundary value problem is replaced by a sequence of linear two-point boundary value problems. The boundary conditions are always satisfied exactly, though the differential equations are approximated. When applied to several small and medium-sized problems, this method has exhibited rather remarkable convergence. However, when it is applied to larger problems, its convergence properties are still subject to some doubt (A1). McGill (M2) has suggested extending the generalized Newton-Raphson method to permit the handling of state space constraints by using certain penalty functions.

Still another possibility, more in the spirit of this dissertation, is that of a finite-difference solution to the two-point boundary value problem (V1). Such finite-difference methods have been known for some time, although their use on the two-point boundary value problem resulting from an application of variational theory has received little attention.

The method of dynamic programming introduced by Bellman (B5) may also be classed as an indirect method. The principal advantage of the dynamic programming approach is

that it leads to a systematic means of numerical solution requiring a stepwise decision process. Although successfully applied to many other problems, this method has had little impact on flight path optimization because of the large computer storage requirements and running times.

Of the direct methods the method of steepest descent, developed independently by Bryson (B6) and Kelley (K3), enjoys widespread popularity among practitioners of trajectory optimization. The primary reason is that it is applicable to a wide class of problems. Indeed, many problems (for example, some reentry problems) have not been solved by any other method. The method of steepest descent is considered a direct method since it adjusts the control function to successively reduce the value of the cost functional. Briefly, the steps are: (a) choose a nominal control which generates a trajectory that approximately satisfies the boundary conditions; (b) integrate numerically the related adjoint equations backward in time to obtain the impulse response functions; (c) use these impulse response functions to change the control so as to maximize the decrease in the cost functional subject to a constraint on the size of the changes themselves. In practice, satisfaction of the terminal boundary conditions is often difficult. Also, the choice of step sizes involves a good deal of engineering judgment. Furthermore, this method exhibits slow convergence near the minimum. A large number of variants and improvements have been suggested. At the expense

of more cumbersome computations, methods based on the second variation yield improved terminal convergence properties (K4, B7, J1, M3). The basic procedure has also been extended to handle state space constraints (D1, K5).

This dissertation presents another means of seeking directly an approximate solution for a very general class of flight path optimization problems. The term discrete-variable approximation is used since we seek approximate values of the optimal solution only on a set of discrete time points rather than an approximate solution over a continuous time interval. In contrast to the other methods mentioned thus far, a timewise discretization process is employed at the outset to reduce the continuous optimal control problem to an ordinary minimization problem. Both equality and inequality constraints are incorporated into the cost function as penalty functions. A conjugate directions method, which provides rapid terminal convergence, is then used to solve numerically the reduced problem. This basic approach promises several desirable features. For example, terminal boundary values can always be satisfied exactly; and problems involving a variable end time, state space constraints, or additional parameter optimization can be treated in a natural manner.

Very little work which is similar to this approach can be found in the literature. Saltzer and Fetheroff (S1) consider a discrete-variable model of an orbital transfer problem, but their formulation cannot be readily extended to other problems. Rosen (R1) also considers the approximation

of the continuous problem by a discrete problem. However, he converts the resulting equality constraints into inequality constraints and proposes an iterative procedure which requires the solution of a nonlinear programming problem with linear constraints at each step. Although Ghonaimy and Bernholtz (G1) also suggest reducing the continuous problem to a discrete one, their approach is quite different. They reduce the continuous problem to a problem in nonlinear programming by representing the control functions with a linear combination of arbitrary polynomials. This approach bears a marked resemblance to the Ritz-Galerkin constructions in the calculus of variations.

To conclude this introduction, we now give a brief outline of what is to follow. In Chapter 2 we state a rather general optimal control problem and proceed to develop a discrete-variable approximation to this problem using a time-wise discretization process. The resulting sequence of ordinary minimization problems may be solved numerically using conjugate direction minimization methods; two such methods are presented and evaluated in Chapter 3. At each step of the iteration process a one-dimensional minimization must be performed. Three candidates for this task are compared in Chapter 4. In Chapter 5 we view the proposed algorithm in its entirety and discuss the important computational details. Several example problems have been treated using this algorithm and are presented in Chapter 6. Some of these problems admit analytical solutions which may be used to check the accuracy of the approximate method.

Finally, Chapter 7 contains a brief review of the advantages and disadvantages of this approach.

# CHAPTER 2

# REDUCTION OF AN OPTIMAL CONTROL PROBLEM TO A PROBLEM IN ORDINARY MINIMIZATION

## 2.1 Statement of a Continuous Problem

We begin by formulating a very broad class of optimal control problems. Let the system to be controlled be described on the time interval $[0,T]$ by the nonlinear vector differential equation

$$\dot{x} = f(t,x,u), \qquad x(0) = x_o , \tag{2-1}$$

where $x(t) = (x^1(t),\ldots,x^n(t))$ is the n-dimensional state vector or trajectory, $u(t) = (u^1(t),\ldots,u^m(t))$ is the m-dimensional control vector, and t is the time. The superscripts denote elements of a vector, while the use of subscripts will be reserved to distinguish one vector from another. Although the initial state vector $x(0)$ is usually specified as in (2-1) and is specified in all of the problems presented in Chapter 6, this is not essential. The computational methods developed in this thesis can easily handle problems in which one or more elements of the state vector are not specified initially. There may, in addition, be r inequality constraints of the form

$$R(t,x,u) \geq 0, \qquad R = (R^1,\ldots,R^r) . \tag{2-2}$$

In particular, we do not exclude the possibility that the control u may be absent from some or all components of R.

Such constraints are commonly called state space constraints and are considerably more difficult to handle than constraints which do involve the control. Terminal boundary conditions

$$Q(T,x(T)) = 0, \quad Q = (Q^1,\ldots,Q^q), \quad q \leq n \tag{2-3}$$

may also be present. The final time T need not be fixed. Finally, the cost functional or index of performance

$$J = J(T,x(T)) \tag{2-4}$$

is to be minimized. Maximization problems can, of course, be treated by reversing the sign of J.

The particular cost functional adopted here identifies this as a Mayer formulation. Many flight path optimization problems occur naturally in the Mayer form. For example, it may be desirable to achieve minimum flight time, maximum payload, or maximum range for a given mission. Furthermore, the Mayer formulation readily encompasses the important case of minimizing a functional of integral form by adjoining an additional state variable to the problem. For example, the problem

$$\min \int_0^T f^o(t,x,u)\, dt, \ \dot{x} = f(t,x,u), \ x(0) = x_o ,$$

where x and f are n-vectors, is equivalent to the Mayer problem

$$\min x^o(T), \ \dot{\underline{x}} = \underline{f}(t,\underline{x},u), \ \underline{x}(0) = \underline{x}_o ,$$

where $\underline{x} = (x^o,x^1,\ldots,x^n)$

$\underline{f} = (f^o,f^1,\ldots,f^n)$

and $\underline{x}_o = (0,x_o^1,\ldots,x_o^n)$.

The optimal control problem to be considered here can now be stated.

Problem A: Find an admissible control $u_*(t)$ which minimizes the cost functional (2-4) subject to the differential side conditions (2-1), the inequality constraints (2-2), and the terminal boundary conditions (2-3).

One usually assumes the real-valued functions f, R, Q, and J to be suitably smooth and u(t) to be piecewise continuous with piecewise continuous first and second derivatives (B2). For our purposes we require that f, R, Q, and J be of class $C^3$ in all arguments since the minimization procedures to be used are based on quadratic forms. The word admissible is commonly used to specify that class of functions from which the optimal control may be chosen. For problems in which the inequality constraints explicitly involve the control, a control must satisfy all of the inequalities in order to be termed admissible.

The questions of existence and uniqueness of optimal solutions are not trivial and cannot be taken for granted. There are many example problems which do not possess one of these properties. However, since our emphasis here is on the computational procedure used, we will assume the existence of a unique optimal control and corresponding trajectory.

## 2.2 The Discretization Process

It is our purpose in this section to devise a discrete approximation to Problem A of the previous section.

We begin by dividing the time interval $[0,T]$ into N equal intervals $[t_i,t_{i+1}]$, $i = 0,1,2,\ldots,N-1$, where each time point $t_i$ is given by $t_i = iT/N$. It is convenient to adopt the notation: $x_i = x(t_i)$, $u_i = u(t_i)$, and $f_i = f(t_i,x_i,u_i)$. The quantities $x_i$ (trajectory points), $u_i$ (control points), and possibly $t_N = T$ will become the discrete variables in the approximations that follow.

## 2.2.1 The Choice of Difference Equations

We now replace the differential equation (2-1) with an appropriate set of difference equations. For example, the set

$$\begin{aligned}
\phi_1 &= x_1 - x_o - 2b(5f_o+8f_1-f_2) = 0 \\
\phi_2 &= x_2 - x_1 - b(-f_o+13f_1+13f_2-f_3) = 0 \\
&\vdots \\
\phi_{N-1} &= x_{N-1} - x_{N-2} - b(-f_{N-3}+13f_{N-2}+13f_{N-1}-f_N) = 0 \\
\phi_N &= x_N - x_{N-1} - 2b(-f_{N-2}+8f_{N-1}+5f_N) = 0 ,
\end{aligned} \qquad (2\text{-}5)$$

where $b = T/24N$, may be used. (Note that each residual is a vector quantity of dimension n.) This replacement amounts to approximating the description of a continuous system with a large number of nonlinear algebraic equations. Presumably, the approximation becomes better as N becomes larger.

The choice of a differencing scheme is nearly unlimited. One would like to think that although one set of difference equations may be better than another, all reasonable differencing schemes would "work" in some sense. That such a view is overly optimistic can be quickly demonstrated

with a couple of simple examples.

Consider the problem: min $y(1)$ subject to $\dot{x} = u$, $\dot{y} = x^2+u^2$, $x(0) = 1$, $x(1) = 0$, and $y(0) = 0$. First, let $N = 2$ and try using first-order Adams interpolation formulas for the difference equations. The discrete approximation in this case is: min $y_2$ subject to the constraints

$$\begin{aligned} x_1 - 1 - (u_1+u_o)/4 &= 0 \\ - x_1 - (u_2+u_1)/4 &= 0 \\ y_1 - (x_1^2+u_1^2+1+u_o^2)/4 &= 0 \\ y_2 - y_1 - (u_2^2+x_1^2+u_1^2)/4 &= 0 \ . \end{aligned}$$

The first three constraint equations can be used to eliminate the variables $y_1$, $u_o$, and $u_2$ so that an equivalent unconstrained problem is:

$$\min y_2 = \min (1/4)(17-32x_1+8u_1+34x_1^2+4u_1^2) \ .$$

Since this quadratic form is clearly positive definite, the minimum exists and, in fact, is located at the point $x_1 = 1/2$, $u_1 = -1$. The minimum value of $y_2$ is 1.3750; the exact value of min $y(1)$ is 1.3130. Thus, this admittedly crude approximation yields a minimum value that is less than 5% higher than the actual minimum.

Now suppose we set $N = 4$ and use the second-order Milne-Simpson formula for our difference equations. In this case the original problem has the following discrete model: min $y_4$ subject to the constraints

$$
\begin{aligned}
x_2 - 1 &- (u_o+4u_1+u_2)/12 = 0 \\
x_3 - x_1 &- (u_1+4u_2+u_3)/12 = 0 \\
- x_2 &- (u_2+4u_3+u_4)/12 = 0 \\
y_2 &- (1+u_o^2+4x_1^2+4u_1^2+x_2^2+u_2^2)/12 = 0 \\
y_3 - y_1 &- (x_1^2+u_1^2+4x_2^2+4u_2^2+x_3^2+u_3^2)/12 = 0 \\
y_4 - y_2 &- (x_2^2+u_2^2+4x_3^2+4u_3^2+u_4^2)/12 = 0 \ .
\end{aligned}
$$

It is disturbing to note that the fifth constraint equation can be satisfied by any number of values for $y_1$ and $y_3$ (regardless of the values given to the remaining variables) since these variables appear in no other equations. However, suppose we ignore this non-uniqueness difficulty and simply discard the fifth equation. Then, it is again possible to use the remaining constraint equations to eliminate variables and derive the following equivalent unconstrained problem:

$$
\begin{aligned}
\min\ (1/432)\ (5220 &- 10368x_2+864u_2+3456u_1+288x_1^2+10440x_2^2 \\
&+721u_1^2+160u_2^2+721u_3^2+24x_1u_1+96x_1u_2+24x_1u_3-3456x_2u_1 \\
&+3456x_2u_3+296u_1u_2+2u_1u_3+296u_2u_3) \ .
\end{aligned}
$$

To test whether this quadratic form is positive definite, we may examine the matrix of coefficients

$$
\begin{bmatrix}
10440 & -3456 & 0 & 3456 & 0 \\
-3456 & 721 & 296 & 2 & 24 \\
0 & 296 & 160 & 296 & 96 \\
3456 & 2 & 296 & 721 & 24 \\
0 & 24 & 96 & 24 & 288
\end{bmatrix} .
$$

Unfortunately, this matrix is not positive definite since the determinate of the principal minor

$$
\begin{vmatrix}
10440 & -3456 \\
-3456 & 721
\end{vmatrix}
$$

is not positive; therefore, no minimum exists. Any attempt to solve this discrete problem numerically would invariably fail as the solution would diverge.

One possibility for constructing a set of difference equations is to approximate each element of x(t) by a polynomial of order p that exactly matches the given function at p+1 equally spaced time points. These interpolating polynomials can then be differentiated and evaluated at any one of the original time points to provide an approximation to $\dot{x}$ at that time point. If this approach is used, the following set of difference equations can be derived where h = T/N.

$$\begin{aligned}
\phi_0 &= -18x_0 + 24x_1 - 6x_2 - 12hf_0 = 0 \\
\phi_1 &= -6x_0 + 6x_2 - 12hf_1 = 0 \\
\phi_2 &= x_0 - 8x_1 + 8x_3 - x_4 - 12hf_2 = 0 \\
&\vdots \\
\phi_{N-2} &= x_{N-4} - 8x_{N-3} - 8x_{N-1} - x_N - 12hf_{N-2} = 0 \\
\phi_{N-1} &= -6x_{N-2} + 6x_N - 12hf_{N-1} = 0 \\
\phi_N &= 6x_{N-2} - 24x_{N-1} + 18x_N - 12hf_N = 0
\end{aligned}$$

This differencing scheme was one of many used in solving the orbital transfer problem presented in Section 6.4, and its use resulted in extremely slow convergence when compared to other schemes. Also, the accuracy obtained was poorer. These results were anticipated, since for these difference equations a particular control point $u_i$ can directly influence (through the $f_i$) only the single residual $\phi_i$.

By far the most attractive difference equations for

our purpose are those derived by replacing the elements of f with interpolating polynomials over some time interval and integrating this approximation to $\dot{x} = f$ over a single interval $[t_i, t_{i+1}]$. Difference equations of this class take the form

$$x_{i+1} = x_i + h\sum_{k=0}^{p} \alpha_{jpk} f_{i+j-k} , \qquad (2\text{-}6)$$

where $h = T/N$. They have been used almost exclusively in the work that follows. The index p is called the order of the equation, while j distinguishes the type. For example, $j = 0$ corresponds to the Adams extrapolation (Adams-Bashforth) formulas, and $j = 1$ corresponds to the Adams interpolation (Adams-Moulton) formulas. The coefficients $\alpha_{jpk}$ are given for the first few orders of types 0, 1, 2, and 3 in Table 2-1. All difference equations of this class are derived using equally spaced time points.

These difference equations cannot be used in the numerical integration of initial value problems if $j > 1$ since they would contain one or more $f_i$ that could not yet be evaluated. However, since in our case all of the discrete variables $x_i$ and $u_i$ will be available simultaneously, it is to our advantage to choose symmetrical formulas that exhibit minimum local truncation error. In this regard it will be helpful to sketch the derivation of the difference equations for $j = 2$.

Consider the scalar differential equation $\dot{x} = f(t)$. Following the presentation of Collatz (C1,p6), we replace f(t) by the Newtonian polynomial which coincides with f(t)

| j | p | k = 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 3/2 | -1/2 | | | | |
| | 2 | 23/12 | -16/12 | 5/12 | | | |
| | 3 | 55/24 | -59/24 | 37/24 | -9/24 | | |
| 1 | 1 | 1/2 | 1/2 | | | | |
| | 2 | 5/12 | 8/12 | -1/12 | | | |
| | 3 | 9/24 | 19/24 | -5/24 | 1/24 | | |
| | 4 | 251/720 | 646/720 | -246/720 | 106/720 | -19/720 | |
| 2 | 1 | -1/2 | 3/2 | | | | |
| | 2 | -1/12 | 8/12 | 5/12 | | | |
| | 3 | -1/24 | 13/24 | 13/24 | -1/24 | | |
| 3 | 1 | -3/2 | 5/2 | | | | |
| | 2 | 5/12 | -16/12 | 23/12 | | | |
| | 3 | 1/24 | -5/24 | 19/24 | 9/24 | | |
| | 4 | 11/720 | -74/720 | 456/720 | 346/720 | -19/720 | |
| | 5 | 11/1440 | -93/1440 | 802/1440 | 802/1440 | -93/1440 | 11/1440 |

TABLE 2-1 The Coefficients $\alpha_{jpk}$ for Adams-Class Difference Equations

at the time points $t_{i+2}$, $t_{i+1}$, $t_i$, ..., $t_{i+2-p}$

$$\mathcal{P}_p(t) = f_{i+2} + \frac{u-2}{1!}\nabla f_{i+2} + \frac{(u-2)(u-1)}{2!}\nabla^2 f_{i+2}$$
$$+ \ldots + \frac{(u-2)(u-1)\ldots(u-3+p)}{p!}\nabla^p f_{i+2} ,$$

where $u = (t-t_i)/h$ and $\nabla f_i = f_i - f_{i-1}$, $\nabla^2 f_i = \nabla(\nabla f_i)$, etc. For some $\tau$, $t_{i+2-p} < \tau < t_{i+2}$, the remainder at non-tabular points is given by

$$\mathcal{R}_{p+1} = f(t) - \mathcal{P}_p(t)$$
$$= \frac{(u-2)(u-1)\ldots(u-2+p)}{(p+1)!} h^{p+1} f^{(p+1)}(\tau) .$$

Integrating $\dot{x} = f(t)$ over the interval $[t_i, t_i+h]$ gives

$$x_{i+1} = x_i + \int_{t_i}^{t_i+h} f(t)\, dt$$
$$= x_i + h\sum_{k=0}^{p} \beta_k \nabla^k f_{i+2} + S_{p+1} , \qquad (2\text{-}7)$$

where $$\beta_k = \frac{1}{k!}\int_0^1 (u-2)(u-1)\ldots(u-3+k)\, du .$$

The first few $\beta_k$ are: $\beta_0 = 1$

$\beta_1 = -3/2$

$\beta_2 = 5/12$

$\beta_3 = 1/24$

$\beta_4 = 11/720$

$\beta_5 = 11/1440$

$\beta_6 = 271/60480$ .

An estimate for the local truncation error $S_{p+1}$ is given by

$$\left|S_{p+1}\right| = \left| \int_{t_i}^{t_i+h} \frac{(u-2)(u-1)\ldots(u-2+p)}{(p+1)!} h^{p+1} f^{(p+1)}(\tau)\, dt \right|$$

$$\leq h^{p+2} \left|\beta_{p+1}\right| \left|f^{(p+1)}\right|_{max} ,$$

where $\left|f^{(p+1)}\right|_{max}$ is the largest magnitude of the (p+1)th derivative of f in the time interval $\left[t_{i+j-p}, t_{i+j}\right]$. Finally, since $\nabla^k f_{i+2} = \sum_{\ell=0}^{k} (-1)^{\ell} \binom{k}{\ell} f_{i+2-\ell}$, we may write the finite-difference equations in Lagrangian form

$$x_{i+1} = x_i + h \sum_{k=0}^{p} \alpha_{2pk} f_{i+2-k}$$

after dropping the local truncation error term from (2-7).

We have noted that an estimate for the local truncation may be written as

$$LTE = C_p h^{p+2} \left|f^{(p+1)}\right|_{max} .$$

Some of the coefficients $C_p$ are given in the following table.

| type, j | p = 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 0 | 5/12 | 3/8 | 251/720 | 95/288 | 19087/60480 |
| 1 | 1/12 | 1/24 | 19/720 | 3/160 | 863/60480 |
| 2 | 5/12 | 1/24 | 11/720 | 11/1440 | 271/60480 |
| 3 | 23/12 | 3/8 | 19/720 | 11/1440 | 191/60480 |

TABLE 2-2 Local Truncation Error Coefficients $C_p$

This information suggests, for example, that the third-order type 2 difference equation is to be preferred over all other third-order equations in this class. This equation is used in the differencing scheme (2-5) mentioned earlier with the exception of the first and last equations. Although it is no guarantee of better accuracy for a given problem, the choice of difference equations with minimum $C_p$ is probably the best selection guide available.

From the above development it seems advantageous to employ a set of difference equations that utilize an equal number of f-points on either side of $f_i$. This symmetry property is present in a class of difference equations based on central differences (C1,p86) and known as the Milne-Simpson equations. These equations have the form

$$x_{i+1} = x_{i-1} + h \sum_{k=-p+1}^{p-1} \nu_{p,k} f_{i+k}, \qquad \nu_{p,k} = \nu_{p,-k},$$

where the first few $\nu_{p,k}$ are given by

| p | k = 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 1 | 2 | | | |
| 2 | 4/3 | 1/3 | | |
| 3 | 126/90 | 26/90 | 1/90 | |
| 4 | 5192/3780 | 1167/3780 | 12/3780 | 5/3780 . |

A third-order Milne-Simpson differencing scheme was tried without success on the orbital transfer problem presented in Section 6.4. Although the difference equations were nearly satisfied throughout the computations, the

quantity to be minimized took on values approaching negative infinity. Evidently, a minimum did not exist when this differencing scheme was used. Since only the trajectory points $x_{i-1}$ and $x_{i+1}$ are explicitly present in each difference equation, there arises in the course of the computations effectively two "solution" sequences corresponding to alternate time points. Furthermore, the largest integrand weighting falls on $f_i$; this reinforces a separation of these two sequences (C1,p90). This alternating effect can be observed in Figure 2-1, which shows a plot of one of the state variables, the radial velocity $v_r$, versus time. Both quantities are dimensionless; the initial and final values of the radial velocity are specified to be zero. Note that the initial trajectory points were chosen close to the optimal values. The dotted lines represent smooth curves faired through the two separate and diverging sets of "solution" points.

One further point needs to be made. In all of the problems solved thus far the resulting control points appear to represent a very smooth function of time. This is apparently due to the fact that the difference equations are all based on polynomial approximation. When problems are encountered for which the actual optimal control is discontinuous, this computational method may not provide a sufficiently accurate approximation to the exact solution. An example of this is presented in Section 6.5.

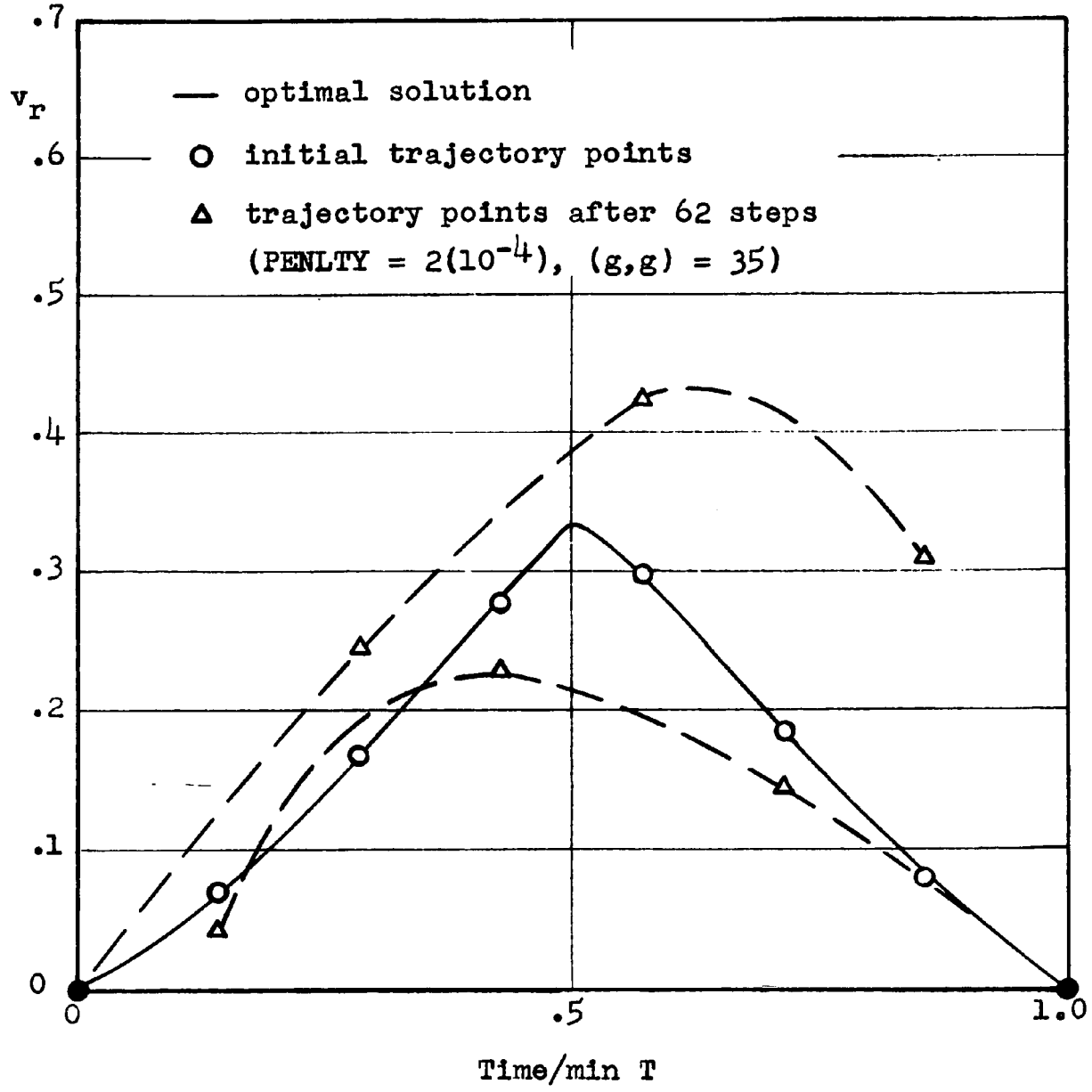



FIGURE 2-1 An Example of Diverging Solution Points: Orbital Transfer Problem with Third-Order Milne-Simpson Differencing Scheme

## 2.2.2 Treatment of Inequality Constraints and Boundary Conditions

The most straightforward way of handling the inequality constraints (2-2) in our discrete approximation is to impose the constraints only at each time point $t_i$. Thus, we use the notation

$$R(t_i,x_i,u_i) = R_i \geq 0, \quad i = 0,1,2,\ldots,N ,$$

in replacing the r continuous inequalities with r(N+1) algebraic inequalities. For most problems the inequality constraints are binding on the optimal solution during some non-zero time interval, and this discrete representation is satisfactory. Of course, the times at which the solution enters or leaves the constraint can at best be determined within an interval h = T/N.

There are some problems for which an inequality constraint is binding on the optimal solution for only one instant of time. Suppose $R^1(t,x,u) \geq 0$ is such a constraint. Then the following device may be used. Suppose the smallest value of $R_i^1$ occurs at the time point $t_m$, and approximate $R^1(t,x,u)$ with a polynomial in t that coincides with $R^1(t,x,u)$ at several of the time points near $t_m$. Next, obtain analytically the minimum of this approximating polynomial. (If several local minima exist, select the minimum nearest $t_m$.) We then insist that this minimum, which depends on several of the $t_i$, $x_i$, and $u_i$, remain non-negative. This device has been used on a problem involving a constraint on the maximum dynamic pressure during atmospheric boost. It

is discussed in Section 6.6.2.

One inherent advantage of using a discrete-variable approximation lies in the ability to treat state space inequality constraints. These constraints, which do not explicitly involve the control, are especially difficult, both from the theoretical and computational viewpoints. In the discrete formulation the $x_i$ and $u_i$ are placed on an equal footing so that in principle it does not matter whether or not the components of $R_i$ contain any control points $u_i$.

The terminal boundary conditions (2-3) remain unchanged except for notation: $Q(T,x_N) = 0$. In practice, however, if a particular terminal boundary condition, say $Q^j = x_N^k - c = 0$, merely serves to specify the final value c of a component of the vector $(T,x_N)$, then that component is simply not considered to be a variable, and the number c is used in all computations. This obvious, but important, observation is an inherent advantage enjoyed by all finite-difference methods in the handling of specified boundary values. Therefore, in what follows, we will use the terminal boundary conditions $Q(T,x_N) = 0$ only to specify relationships involving two or more of the components of $(T,x_N)$ that remain as variables.

### 2.2.3 Statement of the Discrete Problem

Since the usually sharp distinction between state and control variables is no longer present when discrete variables $x_i$ and $u_i$ are used, it is convenient to define a single vector $z = (x_1,\ldots,x_N,u_0,u_1,\ldots,u_N,T)$ to be the vector

of all problem variables. Here, we think of $x_1, x_2, \ldots$ as each being a row vector. As was pointed out in the previous section, any elements of the vector $(T, x_N)$ which are specified quantities are not included in z. If the original problem contains, in addition to the control functions, a finite number of control parameters, it is clear that for the discrete approximation we can simply include these parameters in z as additional problem variables.

We still desire to minimize $J(T, x(T)) = J(T, x_N)$, although J is now considered a function rather than a functional since the final values $(T, x_N)$ now depend on only a finite number of discrete variables. Also, define the vectors

$$\Phi = (\phi_1, \ldots, \phi_N, Q)$$
$$\Psi = (R_o, R_1, \ldots, R_N) .$$

The components of $\Phi$ will be collectively referred to as residuals. Finally, then, the discrete approximation to Problem A may be stated as follows.

Problem B: Find a vector $z_* \in \{z \mid \Phi(z) = 0, \ \Psi(z) \geq 0\}$ that minimizes J(z).

Perhaps the most distressing aspect of this discrete problem is that the dimension of z may be very large. For example, if the original time interval is divided into 20 equal parts, and the dynamic system is of order 4 with a scalar control, the discrete problem will contain roughly 20 x (4+1) = 100 variables. At any rate we now face the problem of minimizing a function of many variables subject to both equality and

inequality algebraic constraints.

## 2.3 Reduction to a Sequence of Unconstrained Ordinary Minimization Problems

The constrained ordinary minimization Problem B is called a nonlinear programming problem. It differs from the typical nonlinear programming problem, however, in that a large number of equality constraints are present because of the difference equations. Although some nonlinear programming algorithms (most notably, Rosen's gradient projection method (R2)) are capable of handling equality constraints, they become unfeasible when the number of equality constraints becomes large. Moreover, many available algorithms are limited to problems possessing objective functions J that are convex, separable, or quadratic and having either linear constraints or convex constraint sets.

Rather than solve the constrained problem, we choose to solve a sequence of unconstrained problems of the form

$$\min F_\nu = \min \left\{ J + K_\nu \left[ \sum_i (\Phi^i)^2 + \sum_j H(\psi^j)(\psi^j)^2 \right] \right\} ,$$

where H is the Heaviside unit step function

$$H(\alpha) = \begin{cases} 1, & \alpha \geq 0 \\ 0, & \alpha < 0 , \end{cases}$$

and $K_\nu = c^{\nu-1} K_1$, $\nu = 1,2,\ldots$, for $c > 1$, $K_1 > 0$. We have made a particularly simple choice for the sequence $\{K_\nu\}$; any monotone increasing sequence would suffice. We will refer to each of these unconstrained minimization problems

as a K-min. Let us denote by PENLTY the expression that multiplies K . Then as the index $\nu$ becomes large, PENLTY approaches zero, and we may expect min $F_\nu$ to approach $J(z_*)$. Intuitively, the constraints are more strictly enforced as $K_\nu$ becomes larger; certainly a z which produces a significant constraint violation will be a poor candidate for $z_*$ because of the large positive quantity added to J.

The use of such penalty functions was originally proposed by Courant (C2). The concept of penalty functions has been given a rigorous basis by various approximation theorems which in essence tell us that if the sequence of solutions to the unconstrained problem converges to something, that something is the solution to the constrained problem (B8, R3). These theorems only require that $F_\nu$ be lower semicontinuous; we will insist that $F_\nu$ be of class $C^3$. The question of whether the sequence of solutions to the unconstrained problem converges at all has been answered affirmatively only for the case in which $F_\nu$ is convex and the set $\{z \mid \Psi(z) \geq 0\}$ is convex (P2, F1). For the typical flight path optimization problem these properties are either absent or difficult to verify. However, in the results to be reported here, no difficulty has been encountered in this regard.

In recent years the concept of a penalty function has been successfully applied to a wide variety of problems (K5, K6, M2). In particular the nonlinear programming method SUMT of Fiacco and McCormick (F1, F2, F3) solves a sequence of problems of the form

$$\min \left\{ J(z) + r_k \sum_j (\psi^j(z))^{-1} + r_k^{-\frac{1}{2}} \sum_i (\Phi^i(z))^2 \right\}$$

for $\{r_k\}$ a monotone decreasing sequence. Since violations of the inequality constraints are never tolerated, it may be difficult to find an initial vector $z_o$ which satisfies the inequality constraints.

# CHAPTER 3

## THE CHOICE OF MINIMIZATION PROCEDURES

Having arrived at a discrete-variable approximation which requires the solution of a sequence of unconstrained minimization problems, let us consider one of these problems for some fixed K (dropping the $\nu$ subscripts). Let the minimum occur at $z_*$: $F(z_*) = \min F(z)$, where z is an M-dimensional vector of problem variables. We will assume that F(z) is of class $C^3$ and that both F(z) and g(z) = grad F(z) are readily available in analytic form. Without further restrictions on F(z) the methods discussed in this chapter can at best locate only a local minimum.

There have been a large number of minimization methods devised which require only the repeated evaluation of F(z) (S2, F4, L1). These methods are designed largely for problems in which F(z) is either very difficult to compute or can be obtained only as the result of a complicated experiment. However, it is reasonable to assume that methods which use the M additional pieces of information contained in the gradient vector will prove superior to these direct search procedures. Therefore, we will limit our choice to one of the gradient methods.

We will further restrict our attention to those iterative minimization procedures which locate $z_*$ as the limit of a sequence $z_0, z_1, z_2, \ldots$, where

$$F(z_{i+1}) = \min_{\theta_i} F(z_i+\theta_i p_i), \quad i \geq 0 . \qquad (3\text{-}1)$$

Here the scalar parameter $\theta_i$ will be referred to as the step size. The iterations (3-1) will be called linear minimizations since at each step F is to be minimized along the line that passes through $z_i$ in the direction $p_i$. Methods of successive linear minimizations are often referred to as optimum gradient methods for obvious reasons. They differ from one another only in the choice of the $p_i$, the successive directions of search. The two most well known of these methods should be mentioned. In the steepest descent or first-order gradient method, each direction of search is that which locally promises the greatest rate of decrease in F: $p_i = -g(z_i) = -g_i$. Although this method has the ability to converge from a poor initial guess $z_o$, it usually exhibits a very slow rate of convergence. In the Newton or second-order gradient method the inverse of the Hessian matrix of second partial derivatives is used as a weighting matrix:

$$p_i = -B_i^{-1} g_i, \qquad B_i = \left( \frac{\partial^2 F}{\partial z^j \partial z^k} \right) \Bigg|_{z_i} . \qquad (3\text{-}2)$$

If F(z) is a positive definite quadratic function, the minimum can be reached in a single iteration with step size one. This can be easily verified by examining the quadratic function obtained from a Taylor expansion of F about $z_*$. At any rate, for the assumed continuity of F, the convergence of the Newton method is quite rapid near the minimum. However, the method may fail to converge at all starting from a poor initial guess $z_o$ since there is no guarantee that $B_o$ will be positive definite. In addition the calculation of

the $\frac{1}{2}M(M+1)$ derivatives at each step is much too laborious for our purposes.

What is needed is a method which combines the desirable features of both the first-order and second-order gradient methods and yet requires only the calculation of first-order derivatives. Methods based on quadratic forms are particularly attractive in this regard. A method is said to be quadratically convergent if, apart from round-off errors, the minimum of a quadratic function can be located in a finite number of iterations. We can expect any quadratically convergent method to possess good terminal convergence since, for the assumed regularity, $F(z)$ can be written as a generalized Taylor expansion about $z_*$

$$F(z) = F(z_*) + \tfrac{1}{2}((z-z_*),B_*(z-z_*)) + 0(z-z_*)^3 , \qquad (3\text{-}3)$$

where we have used the notation $(x,y)$ to denote the scalar product $\sum_i x^i y^i$. For $z$ sufficiently close to $z_*$, $F(z)$ can be represented by the quadratic function $F(z_*) + \frac{1}{2}((z-z_*), B_*(z-z_*))$ regardless of the global nature of $F(z)$.

Consider, then, the special case in which $F(z)$ is the quadratic function

$$F(z) = F(z_*) + \tfrac{1}{2}((z-z_*),A(z-z_*)) , \qquad (3\text{-}4)$$

where $A$ is a positive-definite symmetric matrix. Our iterations take the form

$$z_{i+1} = z_i + \theta_i^* p_i , \qquad (3\text{-}5)$$

where $\theta_i^*$ is defined by $F(z_i+\theta_i^* p_i) = \min_{\theta_i} F(z_i+\theta_i p_i)$. At this

minimum the slope $dF/d\theta_i = (g(z_i+\theta_i p_i), p_i)$ is necessarily zero. Thus, we have

$$(g_{i+1}, p_i) = 0 \ . \tag{3-6}$$

Repeated application of (3-5) yields

$$z_M = z_{j+1} + \sum_{i=j+1}^{M-1} \theta_i^* p_i, \quad 0 \leq j \leq M-1 \tag{3-7}$$

Since by differentiating (3-4) the gradient vector evaluated at $z_i$ is found to be $g_i = A(z_i - z_*)$, it follows that $g_M - g_{j+1} = A(z_M - z_{j+1})$. This result may be combined with (3-7) to yield

$$g_M = g_{j+1} + \sum_{i=j+1}^{M-1} \theta_i^* A p_i, \quad 0 \leq j \leq M-1 \ . \tag{3-8}$$

But, after forming the scalar product of $p_j$ and both sides of (3-8), in view of (3-6) we have

$$(g_M, p_j) = \cancelto{0}{(g_{j+1}, p_j)} + \sum_{i=j+1}^{M-1} \theta_i^* (p_j, A p_i) \ . \tag{3-9}$$

Now, if the vectors $p_0, p_1, \ldots, p_{M-1}$ are conjugate directions, that is, if

$$(p_j, A p_i) = 0, \quad i \neq j \ , \tag{3-10}$$

then $(g_M, p_j) = 0$ for all $0 \leq j \leq M-1$. It can be shown that any set of conjugate directions $p_0, p_1, \ldots, p_{M-1}$, not containing the null vector, forms a basis for $E_M$. Therefore, $g_M = 0$, which implies $z_M = z_*$. This demonstrates that any method using iterations (3-1) is quadratically convergent if the $p_i$ are conjugate directions. Furthermore, the minimum is reached in at most M steps.

## 3.1 The Variable Metric Method

We recall that in both the steepest descent and the Newton methods the directions of search are chosen according to the relation $p_i = -G_i g_i$. In the case of the former, $G_i$ is always the identity matrix; for the latter, $G_i$ is the matrix $B_i^{-1}$ which is the very best choice if F is a quadratic function. By selecting the $G_i$ somewhere between these two extremes, one might expect that an attractive compromise method could be developed.

Davidon (D2) recently devised a gradient method which he called the variable metric method because the weighting matrix $G_i$ is altered during each iteration. This method proceeds by generating a sequence of symmetric positive-definite matrices $G_0, G_1, G_2, \cdots$. The initial choice $G_0$ may be any symmetric positive-definite matrix; unless one has some particular insight into the problem at hand, the identity matrix is the most convenient choice. Thus, the first step is usually taken in the direction of steepest descent. Subsequent members of the sequence are computed using the recursion formula

$$G_{i+1} = G_i + \frac{\theta_i^*}{(p_i, y_i)} P_i - \frac{G_i Y_i G_i}{(y_i, G_i y_i)} \qquad (3\text{-}11)$$

where $y_i = g_{i+1} - g_i$ and the symmetric matrices $P_i$ and $Y_i$ have elements $p_i^j p_i^k$ and $y_i^j y_i^k$ ($j^{th}$ row, $k^{th}$ column), respectively. Each direction of search is chosen as $p_i = -G_i g_i$, and F is minimized along the line $z_i + \theta_i p_i$ to determine each succeeding point $z_{i+1}$.

An excellent discussion of this method has been given by Fletcher and Powell (F5). They have shown that the $p_i$ chosen in this manner are conjugate directions; thus, the variable metric method is quadratically convergent. Furthermore, if F(z) is a quadratic function, then as $z_i$ approaches $z_*$, $G_i$ becomes $-B^{-1}$, and the step size $\theta_i^*$ approaches unity. Thus, this method is able to accumulate information about F(z) in the weighting matrix $G_i$ and finally to provide a measure of the curvature of the function near its minimum. Also, it can be shown that the process is stable in the sense that each $p_i$ points downhill: $(g_i, p_i) < 0$. This amounts to showing that each $G_i$ is positive-definite.

Experience has shown that a large portion of the total running time may be spent in obtaining a $G_i$ that approximates $B^{-1}$; the convergence is quite rapid after that point (F5). For our purposes it would seem advantageous to save the $G_i$ present at the end of each K-min and use it as the $G_o$ for the next K-min. This tactic has been successful; preliminary computations indicate that the number of steps per K-min can be reduced by a factor of 2 or 3.

## 3.2 The Method of Conjugate Gradients

The equivalence between solving a system of linear algebraic equations and minimizing a related quadratic function has been recognized for some time. Hestenes and Stiefel (H1) developed their method of conjugate gradients for the purpose of solving linear systems. Several other discussions of this method, all in the context of linear

systems, are available (H2, B9, A2). Fletcher and Reeves (F6) have explored the use of this method for more general function minimization.

In the method of conjugate gradients the first step is taken in the direction of steepest descent: $p_0 = -g_0$. Thereafter, the direction of search is a linear combination of the current gradient vector and the previous direction of search:

$$p_i = - g_i + \frac{(g_i, g_i)}{(g_{i-1}, g_{i-1})} p_{i-1}, \quad i > 0 . \qquad (3\text{-}12)$$

This recursion formula can be derived by a straightforward but tedious application of the Gram-Schmidt orthogonalization procedure. The set of conjugate directions $p_0, p_1, \ldots, p_i$ is formed by the A-orthogonalization of the vectors $-g_0, -g_1, \ldots, -g_i$. A particularly cogent explanation has been given by Beckman (B9). Each succeeding $g_{i+1}$ is produced by minimizing F along the line $z_i + \theta_i p_i$ so that $(g_{i+1}, p_i) = 0$.

This quadratically convergent method has the twin virtues of simplicity and modest computer storage requirements. Only scalar products need be computed; only three vectors need be stored. These advantages become significant for large M.

A further consequence of (3-12) is that if F(z) is a quadratic function the $g_i$ are mutually orthogonal; that is, $(g_i, g_j) = 0$ for $0 \leq i, j \leq M-1$, $i \neq j$. This led Hestenes and Stiefel (H1) to suggest

$$p_i = - g_i + \frac{(g_i, g_i) - (g_i, g_{i-1})}{(g_{i-1}, g_{i-1})} p_{i-1} \qquad (3\text{-}13)$$

as an alternative to (3-12) in an effort to combat round-off errors. It is also possible that (3-13) could provide improved convergence when applied to non-quadratic functions. However, in a comparison using the constrained brachistochrone problem of Section 6.1.2, this alternate form resulted in an 11% increase in computer running time.

It has proved advantageous to restart the method of conjugate gradients every M+1 steps, that is after M+1 iterations to take a step in the direction of steepest descent. If allowed to continue much past M iterations, the method may produce subsequent $p_i$ which are nearly parallel; this results in very slow convergence. Unlike the variable metric method, the method of conjugate gradients exhausts its potential theoretical advantage after M iterations. Also, there is no guarantee that all $p_i$ will point downhill: $(g_i,p_i) < 0$. Thus, it has been found necessary to restart the method whenever the slope $(g_i,p_i)$ is positive. This usually occurs only in the very early stages of computation when a particularly crude initial guess $z_o$ has been chosen.

## 3.3 Comparisons

The method of conjugate gradients and the variable metric method were first tested on several simple quadratic forms. The results displayed in Table 3-1 illustrate the decrease in F obtained for the function

$$F(z) = 10(z^1)^2 + 20(z^1)(z^2) + 12(z^2)^2 + 4(z^2)(z^3) + 3(z^3)^2$$

starting from the point $z_o = (10,10,10)$. The methods of

steepest descent, conjugate gradients, conjugate gradients with restart, and variable metric required 123, 37, 20, and 4 iterations, respectively, to reduce (g,g) below $10^{-10}$. The first step for each of the methods was in the direction of steepest descent; during this step F was reduced from 4900 to nearly 143. The minimum F of zero occurs at the origin: $z_* = (0,0,0)$.

| Step No. | Steepest Descent | Conjugate Gradients | Conjugate Gradients With Restart | Variable Metric |
|---|---|---|---|---|
| 2 | 8.76 | 4.920 | 4.920 | 4.919 |
| 3 | 4.98 | 2.44 | 2.44 | .0000425 |
| 4 | 3.81 | .0357 | .0357 | .00000000000203 |
| 8 | 2.74 | .00477 | .00465 | |
| 12 | 2.11 | .0000456 | .000000469 | |
| 16 | 1.53 | .00000122 | .0000000122 | |
| 20 | 1.05 | .000000646 | .000000000164 | |

TABLE 3-1 Gradient Methods: A Comparison Based on Function Values of a Quadratic Form

We first note that the method of steepest descent is hopelessly inferior to either of the conjugate direction methods. Secondly, the variable metric method is clearly superior in terms of the number of iterations needed. Finally, the rate of convergence of the method of conjugate gradients is considerably improved when every fourth step is taken in the negative gradient direction. The exercise of the restart

feature described in the previous section will be implicit in all future references to the method of conjugate gradients.

In Table 3-2 we present the number of steps and the computer running time required to solve several discrete-variable problems of the type discussed in Chapter 2. Case I involves the example problem presented in Section 6.2, while the constrained brachistochrone problem of Section 6.1.2 serves as the test problem for case II. In each instance the initial data and termination criteria were identical for both methods.

| Case | M | Number of Steps | | Running Time, Seconds | |
|---|---|---|---|---|---|
| | | Conjugate Gradients | Variable Metric | Conjugate Gradients | Variable Metric |
| I | 15 | 32 | 24 | 4.7 | 5.6 |
| II | 15 | 188 | 42 | 9.7 | 6.4 |
| | 30 | 104 | 68 | 9.6 | 28.7 |

TABLE 3-2 Comparisons of the Conjugate Gradients and Variable Metric Methods Based on Cumbersome Non-Quadratic Functions

Although the variable metric method requires far fewer steps, it is faster only when a small number of variables are present. The large effort expended in the matrix operations in (3-11) force a reliance on the method of conjugate gradients for large M. Since many problems of interest will involve 100 or more variables, the method of conjugate gradients will be used exclusively for the results presented in later chapters.

# CHAPTER 4

## THE CHOICE OF LINEAR MINIMIZATION PROCEDURES

Each of the minimization procedures considered in Chapter 3 requires a one-dimensional or linear minimization at every iteration. After selecting a direction of search $p_i$ we must compute the optimum step size $\theta_i^*$ that minimizes F along the line $z_i + \theta_i p_i$. Three techniques for accomplishing this task will be examined in this chapter.

Common to all three of these techniques is the need to bound the minimum along the line $z_i + \theta_i p_i$. First an estimate for $\theta_i^*$, say $\bar{\theta}_i$, is needed. If F is a quadratic function and $p_i$ is chosen so that $z_*$ can be reached in exactly one more step, the length of that step is given by $2(F(z_*)-F(z_i))/(g_i,p_i)$. If an estimate of $F(z_*)$, say EST, is available, this expression could be used to compute $\bar{\theta}_i$. However, this choice tends to overestimate $\theta_i^*$ and, unless EST is quite accurate, provides a poor estimate near the minimum. Another possibility is to express $\bar{\theta}_i$ as an arbitrary function of previous optimum step sizes: $\theta_{i-1}^*, \theta_{i-2}^*, \cdots$. The choice of

$$\bar{\theta}_i = \begin{cases} -2\left|EST-F(z_i)\right|/(g_i,p_i), & i = 0 \\ 2\,\theta_{i-1}^*, & i > 0 \end{cases} \qquad (4\text{-}1)$$

has proved satisfactory. There is no systematic means for selecting a value for EST. However, almost any reasonable value will suffice since in (4-1) EST is used only for the first step size estimate.

We can now suppress the iteration subscript i and simply denote $F(z+\theta p)$ by $F_\theta$ and $g(z+\theta p)$ by $g_\theta$ for fixed z and p. The minimum can be bounded by testing $F_\theta$ for $\theta = \bar{\theta}, c\bar{\theta}, c^2\bar{\theta}, c^3\bar{\theta}, \ldots, \alpha, \beta$, where $\beta$ is the first step size for which either $F_\beta - F_\alpha$ or the slope $(g_\beta, p)$ is positive. The overall efficiency of the linear minimization is relatively insensitive to the value used for the parameter c; values of 2 and 4 have been used and found satisfactory. The important thing is to be able to bound the minimum in as few steps as possible and yet to provide a small bounding interval $[\alpha, \beta]$. Usually, only one step is needed to bound the minimum. Thus, $F_\alpha > F_{\theta^*} < F_\beta$, and $\alpha < \theta^* < \beta$. It remains to locate $\theta^*$ in the interval between $\alpha$ and $\beta$.

## 4.1 Modified Fibonacci Search

The Fibonacci search is based on the assumption that $F_\theta$ is unimodal. Even if $F_\theta$ is not unimodal the procedure will locate a local minimum since the $F_\theta$ considered here is continuous. Suppose that a trial step size $\theta'$ has been chosen within the bounding interval $[\alpha, \beta]$ as shown in Figure 4-1. We proceed by choosing another step size $\theta''$ located symmetrically with respect to $\theta'$: $\theta'' = \alpha + \beta - \theta'$. Suppose $\theta'$ is closer to $\beta$ than to $\alpha$. Then the interval $[\alpha, \theta'']$ is discarded if $F_{\theta''} > F_{\theta'}$; otherwise, the interval $[\theta', \beta]$ is discarded. The remaining interior step size becomes the $\theta'$ for the next iteration. By continuing in this way, the bounding interval may be reduced indefinitely.

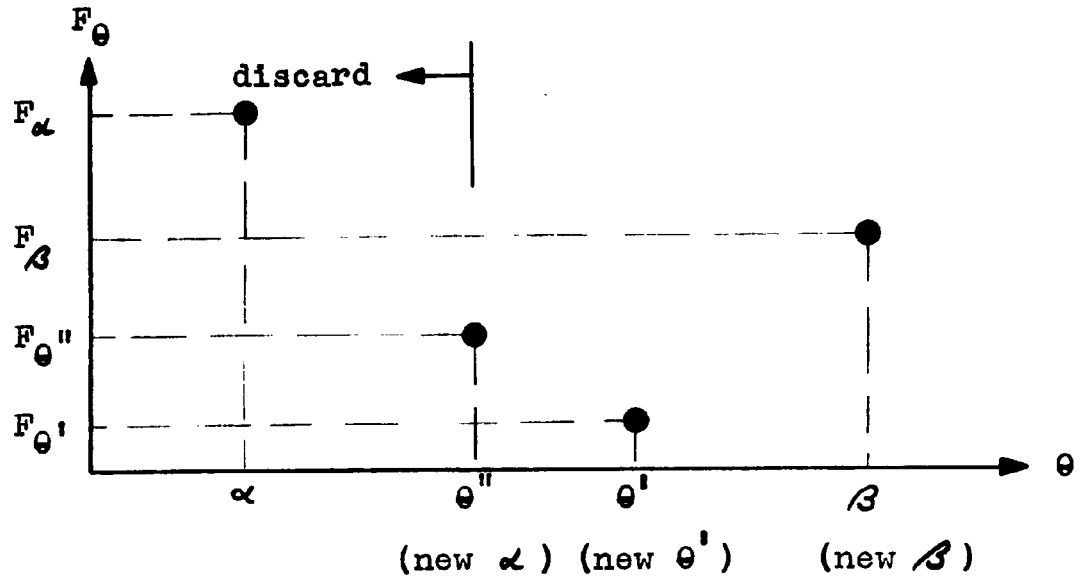



FIGURE 4-1 The Basic Fibonacci Iteration

Once started the search procedure described above is entirely automatic. The question that remains is the placement of the first trial step size $\theta'$. The answer involves the Fibonacci numbers: $\mathcal{F}_0 = \mathcal{F}_1 = 1$, $\mathcal{F}_{i+2} = \mathcal{F}_{i+1} + \mathcal{F}_i$ for $i = 0,1,2,\cdots$. If one is willing to make a total of k trials, the best choice for the first $\theta'$ is $\theta' = (\mathcal{F}_{k-1}/\mathcal{F}_k)(\beta-\alpha)$. This choice defines the Fibonacci search. However, since $\lim_{k\to\infty} (\mathcal{F}_{k-1}/\mathcal{F}_k) = \tau = .61803\cdots$, a good initial choice for $\theta'$, which is independent of k, is $\theta' = \alpha + \tau(\beta-\alpha)$. With this choice for the first $\theta'$ the bounding interval can be reduced to less than 1% of its original length with $k = 11$ trials. A full and lucid description of both the Fibonacci and the modified Fibonacci search has been given by Wilde (W1). He calls the modified Fibonacci search the method of the golden section since $(\theta'-\alpha)/(\beta-\alpha) = (\beta-\theta')/(\theta'-\alpha) = \tau$.

An uncritical programming of the search logic described above can produce an algorithm which will eventually diverge. As Overholt has pointed out (01), such an algorithm is inherently unstable because of the use of finite-precision arithmetic; in effect the roles of $\theta'$ and $\theta''$ become interchanged and the wrong interval is discarded. In order to prevent this occurrence, it is necessary to check whether $\theta'$ or $\theta''$ is the larger at every iteration.

## 4.2 Modified Newton Method

Another approach to the linear minimization problem is to seek a root of the equation

$$s(\theta) = (g_\theta, p) = 0 \qquad (4\text{-}2)$$

since the slope $dF_\theta/d\theta = (g_\theta, p)$ will necessarily be zero at the minimum. In the Newton method the recursion formula

$$\theta_{k+1} = \theta_k - s(\theta_k)/s'(\theta_k) \qquad (4\text{-}3)$$

is employed to produce a sequence of step sizes $\theta_1, \theta_2, \ldots$ which hopefully converge to $\theta^*$. Unfortunately, the computation of the derivative $s'$ requires knowledge of the matrix B of second-order partial derivatives which was ruled out earlier.

However, we may use the approximation

$$s'(\theta_k) \cong \frac{s(\theta_k+\delta) - s(\theta_k-\delta)}{2} \qquad (4\text{-}4)$$

and estimate $s'$ again only after several iterations have passed. The approximation (4-4) is obtained by fitting a

parabola to the three values $s(\theta_k-\delta)$, $s(\theta_k)$, and $s(\theta_k+\delta)$ and evaluating the derivative of that parabola at $\theta_k$. The parameter $\delta$ was arbitrarily chosen to be $(\beta-\alpha)/10$.

It is well known that the Newton method may not converge. If $s'$ becomes too small, a step size that is outside the interval $[\alpha,\beta]$ may be chosen. Also, the process may oscillate about a local extremum of s. Finally, the process does not distinguish between minima and maxima of $F_\theta$. Any algorithm of this method must be capable of handling these situations.

## 4.3 Cubic Interpolation

Once the minimum of $F_\theta$ has been bounded four pieces of information are available: $F_\alpha$, $s(\alpha)$, $F_\beta$, and $s(\beta)$. We can then fit a cubic polynomial to these data and take as the estimate of $\theta^*$ the step size $\theta_e$ for which this polynomial has its minimum. The equations for this choice are (D2):

$$\begin{cases} U = 3(F_\alpha - F_\beta)/(\beta-\alpha) + s(\alpha) + s(\beta) \\ W = (U^2 - s(\alpha)\, s(\beta))^{\frac{1}{2}} \\ \theta_e = \beta - (\beta-\alpha)\left[\dfrac{s(\beta) + W - U}{s(\beta) - s(\alpha) + 2W}\right] . \end{cases} \qquad (4\text{-}5)$$

If $\theta_e$ does not prove to be a sufficiently accurate estimate for $\theta^*$, either the interval $[\alpha,\theta_e]$ or the interval $[\theta_e,\beta]$ is discarded depending on whether or not $\theta_e$ will replace $\beta$ as an upper bound on $\theta^*$. The interpolation formulas (4-5) are then reapplied over the reduced interval. If F(z) is quadratic, only a single application of the cubic interpolation formulas is required to produce an exact

determination of $\theta^*$. Thus, the accuracy of the cubic interpolation prediction of $\theta^*$ increases as the minimum of F(z) is approached.

## 4.4 Comparisons

All of these linear minimization procedures may be terminated whenever

$$\frac{(g_\theta, p)^2}{(g_\theta, g_\theta)(p, p)} < \varepsilon^2 \qquad (4\text{-}6)$$

is satisfied. The left-hand side of this inequality represents the square of the cosine of the angle between $g_\theta$ and p; $\varepsilon$ is some arbitrarily small tolerance. Therefore, (4-6) requires that the angle between the gradient vector and the direction of search differ from the theoretical 90° by no more than $\sin^{-1}\varepsilon$.

With this termination criterion, algorithms for all three of the methods described here were programmed and tested on several representative problems. Some of the results are presented in Table 4-1. Cases I and III involve the unconstrained and constrained brachistochrone problems (Section 6.1), respectively; the problem of Section 6.2 was the test vehicle for case II.

In retrospect the results of these comparison runs could have been anticipated. Of the three methods examined, the modified Fibonacci search finished a poor third. The gradient information is largely wasted in this method since it is used only in checking whether the termination criterion

(4-6) is satisfied. The cubic interpolation process proved to be the best of the three. Although both it and the modified Newton method can locate $\theta^*$ in only one iteration (apart from round-off errors) if F is quadratic, the latter requires three evaluations of the function and its gradient elements for this iteration while the cubic interpolation requires only one.

Further tests revealed that when the cubic interpolation method was used the termination criterion could be reduced to simply

$$F_\alpha \geq F_{\theta_e} \leq F_\beta \ . \qquad (4\text{-}7)$$

The use of this criterion eliminates having to choose a value for the parameter $\varepsilon$. More will be said of this in the following chapter.

| Case | Linear Minimization | Number of Linear Minimizations Required | Running Time, Seconds |
|---|---|---|---|
| I | Fibonacci | 287 | 29.4 |
| | Newton | 181 | 17.7 |
| II | Fibonacci | 31 | 6.9 |
| | Newton | 24 | 5.6 |
| | Cubic | 24 | 5.1 |
| III | Newton | 718 | 115.2 |
| | Cubic | 506 | 56.9 |

TABLE 4-1 Comparison of Linear Minimization Methods

# CHAPTER 5

## A DISCUSSION OF THE COMPUTATIONAL ALGORITHM

In this chapter a computational procedure for handling the discrete-variable approximation described in Chapter 2 will be presented. All algorithms of new computational methods undergo evolutionary developments; the one presented here is no exception. It should be regarded as the most satisfactory prototype to date and not necessarily the final model.

The algorithm has been programmed using the MAD (Michigan Algorithm Decoder) descriptive language. All results have been obtained using the University's IBM 7090 computer. The discussion here will be of a general nature and will not require the use of specific computer instructions.

### 5.1 The Algorithm

In Chapter 2 a discrete-variable approximation to a continuous optimal control problem was formulated in which a sequence of unconstrained minimization problems were to be solved. The basic tools for solving this sequence of subproblems were discussed in Chapters 3 and 4. The computational algorithm can be best presented with the aid of the basic flow chart in Figure 5-1. The symbol $\longrightarrow$ denotes a replacement.

In /1/ the input data is read in, and several parameters are initialized. In particular, the index MARK is set equal to zero. Many of the variables are stored in

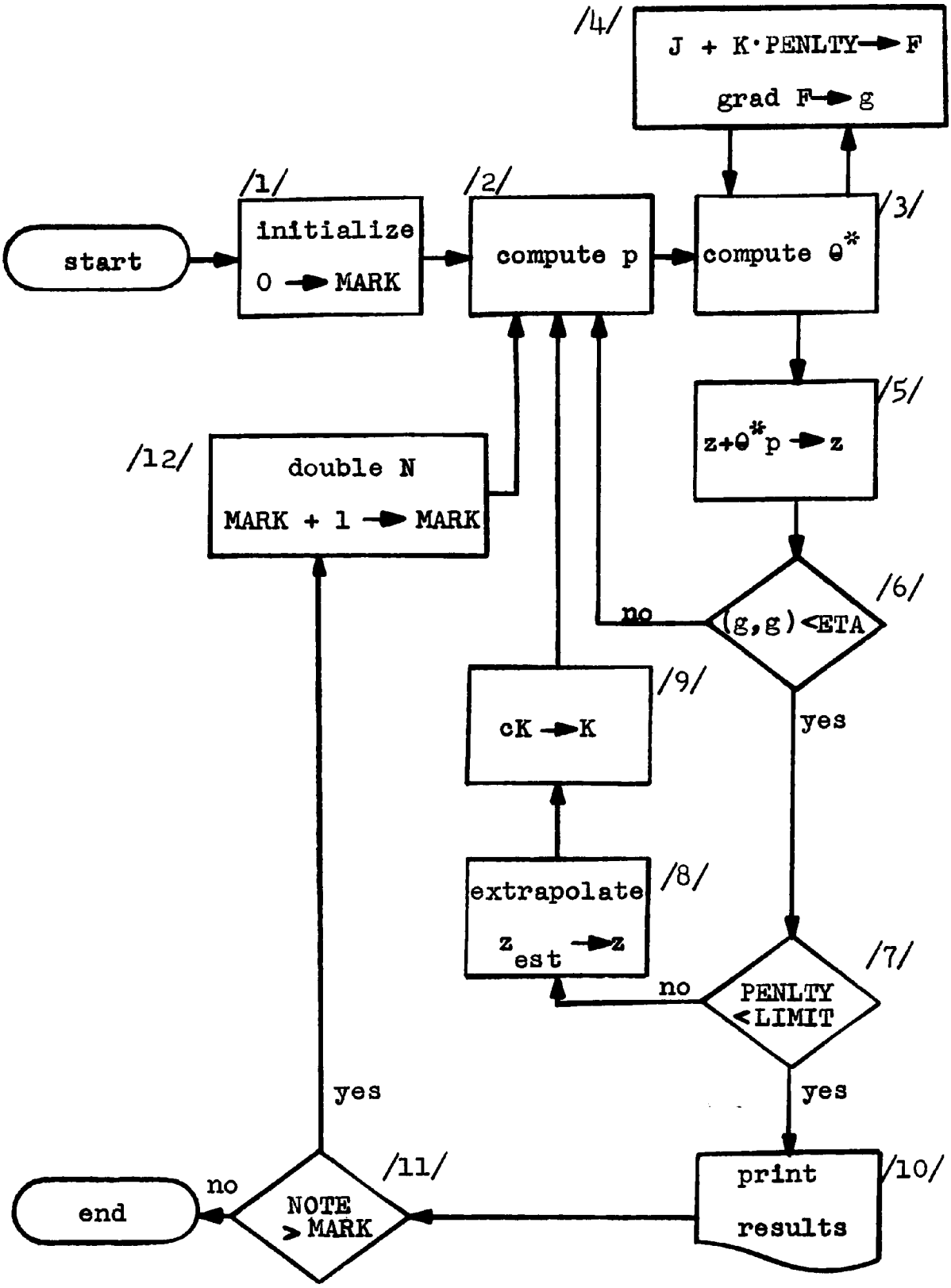



FIGURE 5-1 Basic Flow Chart of the Minimization Algorithm

a common block of the computer memory that is accessible to all of the subroutines.

Given values for $K_1$ and the vector $z_o$, we are ready to solve the first subproblem: min $F_1(z)$. The function of /2/ is to compute each direction of search p according to equation (3-12) of the method of conjugate gradients. Also, an estimate $\bar{\theta}$ of the optimum step size is provided using equation (4-1). If the slope (g,p) is positive, p is rechosen as -g; otherwise, we are ready to enter /3/.

Block /3/ will be referred to as the LINMIN; its purpose is to provide an acceptable value for $\theta^*$, the step size that minimizes $F(z+\theta p)$. The LINMIN may call on /4/ as needed to provide values for F and g along the line $z + \theta p$ given a step size $\theta$. The first task is to bound $\theta^*$ in an interval $[\alpha, \beta]$ as described in Chapter 4. Then the cubic interpolation formulas (4-5) are applied. As soon as this interpolation provides a step size for which F is not greater than either $F_\alpha$ or $F_\beta$, that step size is accepted as $\theta^*$.

The purpose of /4/ is to evaluate the function F and its gradient vector g given any vector $z + \theta p$. This subroutine is called FANDG, and it is here that the major programming and computing efforts fall. The residuals $\Phi^i$, the inequalities $\psi^i$, and the various derivatives associated with g must all be programmed here. Although each execution of FANDG results in the computation of both F and g, we will refer to the entire operation as simply a function evaluation. Anything that can be done to improve the efficiency of

FANDG or reduce the number of times it must be called will have a significant effect on the total computer running time.

In /5/ a step of length $\theta^*$ is taken along p to the next point z. If at this new z the square of the magnitude of the gradient vector is less than some arbitrary tolerance ETA (g = 0 at the minimum), the process is terminated, and we say we have completed a K-min. It has not been found necessary to guard against terminating near an inflection or saddle point. If the termination criterion /6/ is not satisfied, we return to /2/ to select the next direction of search.

In order to decide whether K is large enough for the subproblem min $F_y$ to provide a good approximation to problem B of Section 2.2, we ask that the measure of constraint satisfaction, PENLTY, be less than an arbitrary tolerance LIMIT. If PENLTY is less than LIMIT we continue to /10/ and print out the results. However, if the termination criterion /7/ has not been met, the penalty constant K is increased by a factor of c, and we begin solving a new subproblem.

Before increasing K, however, it should be possible to use the knowledge gained from previous K-mins to estimate the vector z for which the next K-min will occur. The use of such an estimate, say $z_{est}$, rather than simply the current z, as the $z_o$ for the subsequent subproblem should speed up the overall process. The $z_*$ of each subproblem can be thought of as a function of K. Since we are interested in K becoming large, let us assume that $z_*(K)$ can be expanded in a power series in $K^{-1}$

$$z_*(K) = \sum_{i=0}^{\infty} a_i (K^{-1})^i$$

so that if k subproblems have been solved $z_*(K)$ can be approximated by the polynomial

$$z_*(K) = \sum_{i=0}^{k} a_i (K^{-1})^i \ .$$

Suppose that the previous K-min occurred at $z_{*1}$, the one before that at $z_{*2}$, and so on. Then if k = 2, we may write

$$z_{*2} = a_o + a_1 K_1^{-1}$$
$$z_{*1} = a_o + a_1 (cK_1)^{-1}$$

and derive the first-order estimate

$$z_{est} = ((c+1)z_{*1} - z_{*2})/c. \qquad (5\text{-}1)$$

Similarly, if k = 3,

$$z_{*3} = a_o + a_1 K_1^{-1} + a_2 K_1^{-2}$$
$$z_{*2} = a_o + a_1 (cK_1)^{-1} + a_2 (cK_1)^{-2}$$
$$z_{*1} = a_o + a_1 (c^2 K_1)^{-1} + a_2 (c^2 K_1)^{-2} \ ,$$

and we can derive the second-order estimate

$$z_{est} = ((c^2+c+1)(cz_{*1} - z_{*2}) + z_{*3})/c^3 \ . \qquad (5\text{-}2)$$

In /8/ we have used (5-1) after two K-mins and (5-2) after three or more K-mins. Similar equations are used to provide estimates for min $F_{k+1}$.

Experience has shown that it is better first to solve a discrete problem with a small number of variables (small N) and then to increase the size of the problem rather

than to begin directly with the larger problem. Accordingly, the algorithm is equipped to automatically solve a sequence of problems with each succeeding problem involving twice as many time intervals as the previous one. The number of times this doubling process is to be used is denoted by NOTE and may be specified by the user. In /11/ whenever NOTE is greater than the running index MARK, the subroutine /12/ (DOUBLE) is called. Here the elements of the current z are relocated to correspond to the renumbering of the time points, and interpolated values are provided for variables corresponding to the new intermediate time points. This expanded vector is then used as the $z_0$ for the next K-min. The value of K remains unchanged.

There is no automatic selection of a value for the initial penalty constant $K_1$; a value must be provided by the user. A small $K_1$ means poor adherence to the constraints and allows a solution to the first subproblem which may differ considerably from that of the constrained problem. On the other hand a very large $K_1$ will result in an excessive running time for the first K-min because of an overly stringent enforcement of the constraints. For the problems solved thus far $K_1 \cong 10^3$ and $c \cong 5$ have worked satisfactorily.

## 5.2 The Effect of LINMIN Accuracy

The method of conjugate gradients requires that F be minimized along the line $z + \theta p$ at each iteration. Strictly speaking, the method is not quadratically convergent unless $(g_{i+1}, p_i) = 0$ for all i. In practice this linear

minimization is determined numerically and therefore inaccurately. It is natural to ask what effect this inaccuracy has upon the observed convergence of the method.

A series of comparison runs were obtained using the modified Fibonacci search for LINMIN and using the unconstrained brachistochrone problem (Section 6.1.1) as a test problem. The termination criterion (4-6) was used in LINMIN; the only difference between runs was the value used for $\varepsilon$. Each run consists of four K-mins for N=5. The results are presented in Table 5-1. It is apparent that for $\varepsilon$ too large the computational effort is considerably

| $\sin^{-1}\varepsilon$, Degrees | $\varepsilon^2$ | Running Time, Seconds | Iterations | Number of Times $(g,p) > 0$ Occurs |
|---|---|---|---|---|
| 1 | .000305 | 33.4 | 275 | 0 |
| 2 | .001218 | 29.4 | 287 | 0 |
| 4 | .004866 | 38.3 | 392 | 6 |
| 5 | .007597 | 45.7 | 443 | 9 |

TABLE 5-1 The Effect of LINMIN Accuracy on Convergence

increased. Even more alarming, for the two least accurate cases, the method will occasionally compute a p that points uphill. On the other hand if $\varepsilon$ is chosen too small, the undue amount of effort expended in LINMIN (requiring additional function evaluations) increases the total running

time even though fewer iterations are required. There may be no systematic way of selecting a value for $\varepsilon$, but the importance of the choice is clearly evident.

The trend of running time versus $\varepsilon$ seen in Table 5-1 is probably more striking than it would have been had the cubic interpolation been used for the LINMIN. The reason is that for large $\varepsilon$ the interpolated estimate of the minimum will almost always satisfy the termination criterion (4-6) after the first iteration. However, even with the cubic interpolation the running time can rise sharply if too small a value is chosen for $\varepsilon$. Fortunately, when the cubic interpolation LINMIN is used, it is possible to apply the seemingly crude termination criterion (4-7) in place of (4-6) and thus eliminate the hazardous choice of a value for $\varepsilon$. Table 5-2 provides a comparison based on nine K-mins with the constrained brachistochrone problem (Section 6.1.2).

| | A | B |
|---|---|---|
| Function Evaluations | 1494 | 1626 |
| Iterations | 708 | 757 |
| Running Time, Seconds | 94.5 | 94.3 |

TABLE 5-2 Comparison of Termination Criteria for the Cubic Interpolation

Column A corresponds to the $\varepsilon$-criterion with $\varepsilon^2 = .0005$; Column B corresponds to the criterion (4-7). Although the

former case affords more accurate linear minimizations which result in fewer iterations, this advantage is offset by the simplicity of the latter.

## 5.3 The Effect of Round-Off Errors

The discrete-variable approximation presented in this dissertation is inherently susceptible to round-off error difficulties since each of the residuals becomes small, not by each term becoming small but by the difference between terms becoming small. As K becomes large and PENLTY becomes small, the situation becomes more acute; operations involving the residuals may be accurate to only a few significant figures. In several instances when a relatively large K, low ETA, or both were used, the observed rate of convergence was much slower than the theoretical potential of the method of conjugate gradients would suggest.

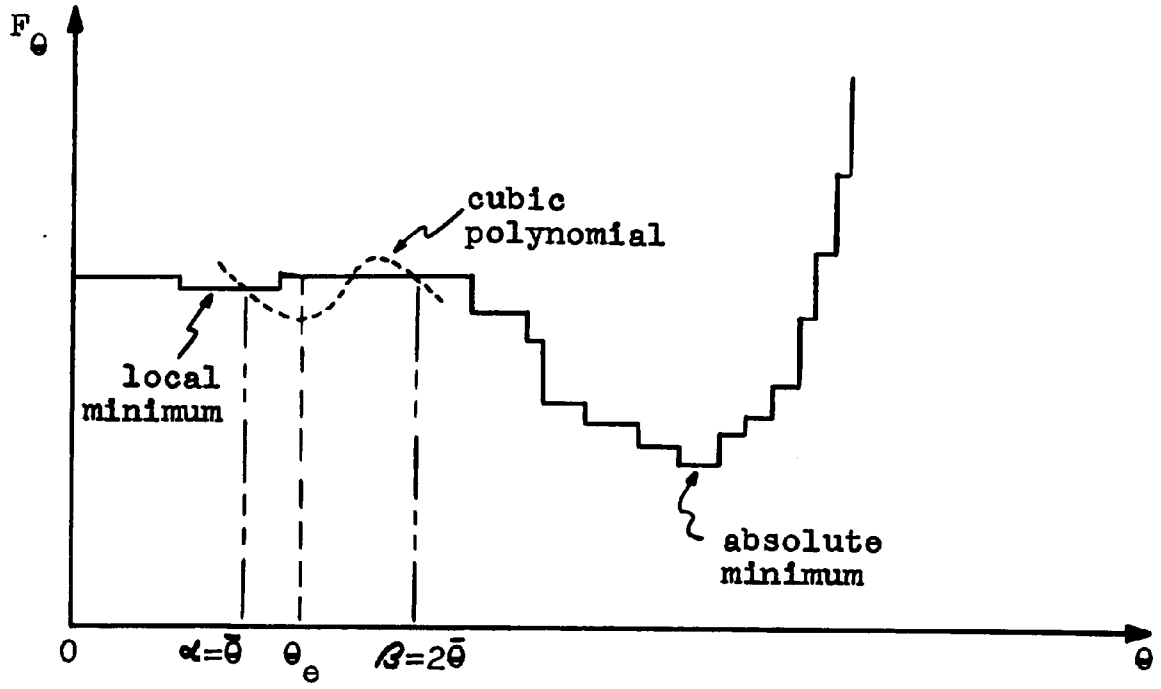
A more immediate concern is that the presence of round-off errors can lead to difficulty in terminating the LINMIN subroutine. The computer, of course, views $F_\theta$ as a sum of numerous step functions rather than as a continuous function. Consider $F = J + K\Sigma(\phi^i)^2$, and suppose that for step sizes $\bar{\theta}$ and $2\bar{\theta}$ the first two residuals, because of subtracting terms of magnitude near one, are accurate only to four significant figures:

| | $\bar{\theta}$ | $2\bar{\theta}$ |
|---|---|---|
| $\phi^1$ | .0001476 | .0001477 |
| $\phi^2$ | .0003214 | .0003214 . |

Thus, supposing that J and the other residuals remain unchanged, the computer will conclude that $F_{2\bar{\theta}} > F_{\bar{\theta}}$, and the bounding procedure will be stopped. However, if the same computations are performed on a computer with larger basic number size, there might result

| | $\bar{\theta}$ | $2\bar{\theta}$ |
|---|---|---|
| $\phi^1$ | .00014764 8 | .00014765 1 |
| $\phi^2$ | .00032144 3 | .00032140 8 |

so that, actually, $F_{2\bar{\theta}} < F_{\bar{\theta}}$. The original situation may be illustrated with the following sketch.




Since the bounding procedure ended prematurely in the sense that the function increase was the result of using finite-precision arithmetic, the slopes $s(\alpha)$ and $s(\beta)$ are both negative. Thus, the LINMIN may require several iterations

before predicting the local minimum of $F_\theta$. It is even possible for LINMIN to compute a $\theta_e$ that is not between $\alpha$ and $\beta$ because of the small denominator $\beta - \alpha$ in the equation (4-5) for U. The overall convergence will almost certainly be slower since the actual minimum was not attained for this direction of search.

This difficulty in the linear minimization may be avoided in one of two ways. First, we can stop the bounding procedure whenever either the slope or $F_\beta - (F_\alpha + \zeta)$ becomes positive and then use $F_\alpha + \zeta \geq F_{\theta_e} \leq F_\beta$ as the termination criterion rather than (4-7). Typically, the positive constant $\zeta$ is on the order of the last significant digit of F. This guarantees that if the bounding procedure stops because of a function increase, it will be a "genuine" function increase and not one due to round-off errors. The second possibility is to use double precision arithmetic in computing all function values. Although this partial use of double precision prevents the difficulty in LINMIN at the price of significantly higher running times, it does little to revive the fast terminal convergence expected of the conjugate gradients method.

Experience has shown that it is necessary to use double precision for all computations in order to improve terminal convergence. The loss of accuracy in computing the residuals results in a corresponding loss of information in the gradient vector. This in turn results in inaccurate directions of search as well as inaccurate linear minimizations. Thus, when the method of conjugate gradients is

used, good performance depends on an accurate determination of the gradient vector.

A pair of comparison runs using the unconstrained booster problem of Section 6.6.1 were obtained to test the rate of convergence with and without double precision. The problem involved 40 variables; the termination values ETA = .09 and LIMIT = $10^{-8}$ were used. The results in terms of the number of iterations required are given in Table 5-3.

| K | Number of Iterations | | % Reduction Using Double Precision |
|---|---|---|---|
| | Single Precision | Double Precision | |
| 1000 | 561 | 456 | 18.7 |
| 5000 | 316 | 219 | 30.7 |
| 25000 | 338 | 168 | 50.3 |
| 125000 | 66 | 32 | 51.5 |

TABLE 5-3 A Comparison of Single and Double Precision Computations in Terms of the Number of Iterations Required

The reduction in iterations per K-min increases as K increases since the residuals become smaller and therefore less accurate with increasing K. A somewhat larger than 50% reduction is achieved during the last two K-mins. However, the computer running time required per iteration is nearly five and a half times greater when using double precision so that the overall advantage lies with the single precision computations.

This will no longer be the case if ETA and LIMIT are chosen sufficiently small.

## 5.4 The Effect of the Differencing Scheme Used

In Section 2.2.1 we pointed out that differencing schemes based on polynomial approximation for f(t,x,u) were to be preferred over those based on polynomial approximation for x. In this section we present some comparison data for differencing schemes of the former variety.

The effect of the order of the difference equations was examined using Adams interpolation formulas. The first, second, and third order differencing schemes employ the following sets of difference equations.

first order:

$$\begin{aligned} \phi_1 &= x_1 - x_o - h(f_o+f_1)/2 = 0 \\ &\vdots \\ \phi_N &= x_N - x_{N-1} - h(f_{N-1}+f_N)/2 = 0 \end{aligned} \tag{5-3}$$

second order:

$$\begin{aligned} \phi_1 &= x_1 - x_o - h(f_o+f_1)/2 = 0 \\ \phi_2 &= x_2 - x_1 - h(-f_o+8f_1+5f_2)/12 = 0 \\ &\vdots \\ \phi_N &= x_N - x_{N-1} - h(-f_{N-2}+8f_{N-1}+5f_N)/12 = 0 \end{aligned} \tag{5-4}$$

third order:

$$\begin{aligned} \phi_1 &= x_1 - x_o - h(f_o+f_1)/2 = 0 \\ \phi_2 &= x_2 - x_1 - h(-f_o+8f_1+5f_2)/12 = 0 \\ \phi_3 &= x_3 - x_2 - h(f_o-5f_1+19f_2+9f_3)/24 = 0 \\ &\vdots \\ \phi_N &= x_N - x_{N-1} - h(f_{N-3}-5f_{N-2}+19f_{N-1}+9f_N)/24 = 0 \end{aligned} \tag{5-5}$$

The constrained brachistochrone problem of Section 6.1.2 was used to test these differencing schemes. Each test run used the same data and termination tolerances: ETA = .1, LIMIT = $10^{-8}$. Table 5-4 illustrates the accuracy obtained for the state variables x and y and the control u at several time points (N=20). The most accurate values are underlined. Clearly, the higher order differencing schemes are more accurate. This must be balanced with the higher running times and the more cumbersome programming task incurred as the order is increased. At N=20 the number of function evaluations obtained per second of running time was 13.3, 12.4, and 11.8 for the first, second, and third order schemes, respectively. Approximately the same total number of iterations were used with each scheme. In this case the third order difference equations are to be preferred.

Another series of comparison runs were obtained using the minimum time orbital transfer problem of Section 6.4. The results presented in Table 5-5 are for N=20, ETA = .1, and LIMIT = $10^{-8}$. The minimum transfer times have been non-dimensionalized. Cases I and II are for the third order Adams interpolation scheme (5-5) and the third order scheme (2-5), respectively. The latter differencing scheme exhibits the minimum local truncation error estimate (see Section 2.2.1) and results in a significantly smaller transfer time without requiring any additional time per function evaluation. In case III a set of fifth order difference equations with minimum local truncation error coefficient (see Table 2-2) is used. Although the transfer

| Variable | Exact | First | Second | Third |
|---|---|---|---|---|
| $x_4$ | .02314 | .02988 | .02871 | .02385 |
| $x_8$ | .15444 | .15448 | .15591 | .15562 |
| $x_{12}$ | .37665 | .37332 | .37493 | .37747 |
| $x_{16}$ | .65418 | .64991 | .65298 | .65633 |
| $x_{20}$ | 1.00000 | .98300 | .98999 | .99754 |
| $y_4$ | .06080 | .05500 | .05658 | .06056 |
| $y_8$ | .17719 | .16950 | .17058 | .17544 |
| $y_{12}$ | .28833 | .28101 | .28428 | .28654 |
| $y_{16}$ | .42285 | .40053 | .40633 | .41262 |
| $y_{20}$ | .48708 | .50772 | .50501 | .49518 |
| $u_4$ | 1.02278 | .92277 | .92991 | .99476 |
| $u_8$ | .47477 | .60530 | .58762 | .53160 |
| $u_{12}$ | .46365 | .40370 | .42084 | .44677 |
| $u_{16}$ | .37163 | .42759 | .41020 | .37881 |
| $u_{20}$ | .00000 | .03714 | .02601 | .02996 |

TABLE 5-4 Accuracy Comparison of First, Second, and Third Order Adams Interpolation Differencing Schemes

time is again smaller, this marginal improvement is offset by the greater complexity of the difference equations.

| Case | K | Function Evaluations per Second | Transfer Time |
|---|---|---|---|
| I | 256000 | 7.8 | 3.346730 |
| II | 256000 | 7.8 | 3.342329 |
| III | 250000 | 6.9 | 3.341173 |

TABLE 5-5 Comparison of Three Differencing Schemes on an Orbital Transfer Problem

Within broad limits the choice of an Adams-class differencing scheme appears to be secondary when compared to the choices for ETA, LIMIT, and N with regard to the accuracy obtained for a given problem. For example, when the difference equations (2-5) are used on this same orbital transfer problem (N=10, K = 256000), reductions in ETA resulted in substantially lower transfer times:

| ETA | Transfer Time |
|---|---|
| .10 | 3.345068 |
| .02 | 3.331638 |
| .01 | 3.330933 |

Unfortunately, the use of smaller values for ETA intensifies the round-off difficulties described in the previous section.

## 5.5 The Use of Unequally Spaced Time Points

The adoption of equally spaced time points $t_i$ in

Section 2.2 allows the use of simple differencing schemes which in turn provide for a straightforward determination of the gradient elements. However, for some problems it may be desirable to use time points that are not equally spaced. If, for example, certain portions of the trajectory are known to be more sensitive than others to control variations, a more accurate discrete approximation may be possible if the $t_i$ are concentrated in those sensitive portions.

The use of unequally spaced time points can be implemented quite simply with the aid of an arbitrary time transformation. Define a new time variable $\tau$ and let the real time t be a function of $\tau$. For example, suppose $t/T = \sigma(\tau)$, $\sigma(0) = 0$, and $\sigma(1) = 1$. Then the differential equation (2-1) can be replaced by

$$dx/d\tau = (T\,\sigma'(\tau))\, f(T\sigma(\tau),x,u),\ x(0) = x_0\ . \qquad (5\text{-}6)$$

The $\tau_i$ can now be equally spaced ($\tau_i = i/N$, $i = 0,1,\ldots,N$) over the interval $[0,1]$, and a discrete approximation devised as before based on (5-6). The $t_i = T\,\sigma(\tau_i)$ will be distributed over the interval $[0,T]$ depending on the nature of the transformation. We insist that $\sigma(\tau)$ be monotonically increasing and differentiable on the interval $[0,1]$.

This device has been tried on the orbital transfer problem of Section 6.4 because of the symmetrical nature of its optimal control time history. Very few changes were required in the original program. Thus far, the use of unequally spaced time points does not represent an advantage over the use of equally spaced time points in this problem.

The transformation $\sigma(\tau) = \tau(1+a(\tau-\frac{1}{2})(\tau-1))$, $-2 < a < 4$, was used. For negative values of a the $t_i$ are grouped closer to the end points 0 and T; for positive a the $t_i$ are grouped about the midpoint T/2. When compared with a previous run (case II, Table 5-5) the use of a = -1.5 resulted in poorer accuracy although the total running time was 12.5% less. The use of a = 3.5 led to even poorer accuracy and very slow convergence.

## 5.6 Inequality Constraints: $\psi^2$ versus $\psi^4$

When inequality constraints are incorporated as penalty functions of the form $K\, H(\psi^i)(\psi^i)^2$, the method of conjugate gradients can be applied since continuous first order derivatives are available for the assumed continuity of $\psi^i(z)$. However, because of the presence of the step function $H(\psi^i)$, the second order partial derivatives will be discontinuous at $\psi^i(z) = 0$. Therefore, the quadratic convergence property of the conjugate gradients method cannot be expected to hold (K6).

An obvious solution is to use instead penalty functions of the form $K\, H(\psi^i)(\psi^i)^4$. In this case F(z) will exhibit continuous derivatives through the third order as required for quadratic terminal convergence. It may be necessary to multiply $\psi^i$ by an appropriate constant in order to assure proper constraint satisfaction for a given value of LIMIT. For example, both $(\psi^i)^2 = 10^{-8}$ and $(100\,\psi^i)^4 = 10^{-8}$ imply $\psi^i = \pm 10^{-4}$.

The constrained brachistochrone problem of Section

6.1.2 provides a logical test vehicle since solutions to that problems have been obtained with relative ease using the $\psi^2$-form penalty functions. The number of iterations required for each K-min are shown in Table 5-6 for three comparison runs. The third order Adams interpolation differencing scheme (5-5) is used; the termination levels are ETA = .1 and LIMIT = $10^{-8}$. In this problem a linear state space inequality constraint is imposed at each time point.

| N | K | A | B | C |
|---|---|---|---|---|
| 5 | 1000 | 103 | 432 | 117 |
| 5 | 4000 | 43 | 155 | 34 |
| 5 | 16000 | 20 | 14 | 12 |
| 5 | 64000 | 22 | 24 | 10 |
| 10 | 64000 | 104 | 178 | 108 |
| 10 | 256000 | 119 | 56 | 56 |
| 20 | 256000 | 162 | 228 | 147 |

TABLE 5-6 Comparison of Inequality Constraint Handling Based on the Number of Iterations Required

Column A corresponds to the use of $\psi^2$-form penalty functions. The other two comparison runs involved penalty functions of the form K H($\psi^i$)(CON $\psi^i$)4. Column B corresponds to the choice: CON = 100. This choice resulted in an unduly strict adherence to the inequality constraints during early

computations at each N level, although the terminal convergence was improved. By choosing CON = 10 it is possible to reduce considerably the number of iterations required. This is shown in column C. In both cases the use of the $\psi^4$-form penalty functions resulted in slightly better accuracy.

We conclude that the use of $\psi^4$-form penalty functions does significantly improve terminal convergence, although the overall performance depends critically on choosing a value for CON that is not too large. The difference between these two forms of penalty functions is a practical matter affecting the actual convergence and is not merely a mathematical fine point.

# CHAPTER 6

## EXAMPLE PROBLEMS

The practical utility of any computational method can only be demonstrated through application to representative problems. In this chapter we treat several optimal control problems using the algorithm described in Chapter 5. Many of these problems possess analytical solutions; this allows direct accuracy comparisons between the exact solution and our approximate numerical results. Not all of the examples considered here deal with flight path optimization. However, those problems which are not strictly speaking flight path optimization problems still serve to illustrate certain features that are characteristic of many flight path optimization problems.

### 6.1 The Brachistochrone Problem

The first two example problems considered here are based on the brachistochrone problem which is traditionally one of the first presented in any elementary course in the calculus of variations. The following characteristics make it an appropriate demonstration problem for our purposes. The second-order system is nonlinear in both the state and the control variables. Yet an analytical solution is known. Furthermore, even when a linear state space inequality constraint is imposed, an analytical solution is still possible. For this reason several authors have used a constrained brachistochrone problem as a demonstration piece (B10, D3, M2).

The original brachistochrone problem was proposed by John Bernoulli in 1696.

> Given two points that do not lie on the same vertical line, select a plane curve joining these two points so that a particle starting from rest will traverse this curve in the shortest possible time. The only force acting on the particle is derived from a uniform gravity field.

It may be modified by allowing the lower end point to assume any position on a specified vertical line, say $x = 1$. Suppose that the solution to this variable end point problem results in a minimum time T. Then an equivalent problem is to ask that the optimum path with free end point exhibit maximum horizontal distance for a fixed transfer time T.

## 6.1.1 An Unconstrained Brachistochrone Problem

We wish to consider the fixed-time problem just described. The state variables are x, the horizontal distance, and y, the vertical distance measured positively downward. The units of length and time are feet and seconds, respectively. The acute angle u measured between the x-axis and the tangent to the curve plays the role of a control variable. The problem can be stated in Mayer form as: minimize $-x(T)$ subject to

$$\begin{aligned} \dot{x} &= (64.4\ y)^{\frac{1}{2}} \cos u, \qquad x(0) = 0 \\ \dot{y} &= (64.4\ y)^{\frac{1}{2}} \sin u, \qquad y(0) = 0 \end{aligned} \tag{6-1}$$

and specified final time T. We will refer to this problem as the unconstrained brachistochrone problem. No generality is lost by taking the origin for the initial position of

the particle. The equations (6-1) imply that the particle starts from rest.

The analytical solution to the unconstrained brachistochrone problem is well-known. The optimal trajectory is a cycloid with infinite slope at the origin and zero slope at the end point. The optimal control angle varies linearly with time from 90 to 0 degrees. In the original problem formulation, the particle can reach a horizontal distance of one foot in a minimum time of .312354 seconds. Therefore, a fixed final time T = .312354 seconds will be used so that the maximum horizontal distance will be one foot.

The discrete-variable approximation involves 3N problem variables: $z = (x_1,\ldots,x_N,y_1,\ldots,y_N,u_1,\ldots,u_N)$. Note that there is no initial control point $u_o$ since $\dot{x}$ and $\dot{y}$ are both zero at the origin regardless of the initial value of the control. We use the third-order Adams interpolation differencing scheme (5-5) to approximate the differential equations (6-1).

A single-precision computer run with termination tolerances ETA = .005 and LIMIT = $10^{-8}$ has required 50.9 seconds of IBM 7090 time to provide numerical solutions for N=5, 10, and 20. A value of 1000 is used for the initial K; it is increased by a factor of 4 after each K-min. The initial vector $z_o$ consists of the simple straight-line trajectory shown in Figure 6-1. The trajectory points and control points obtained for N=20 are presented in Figures 6-1 and 6-2, respectively. In each figure the solid line represents the analytical solution. The discrete-variable

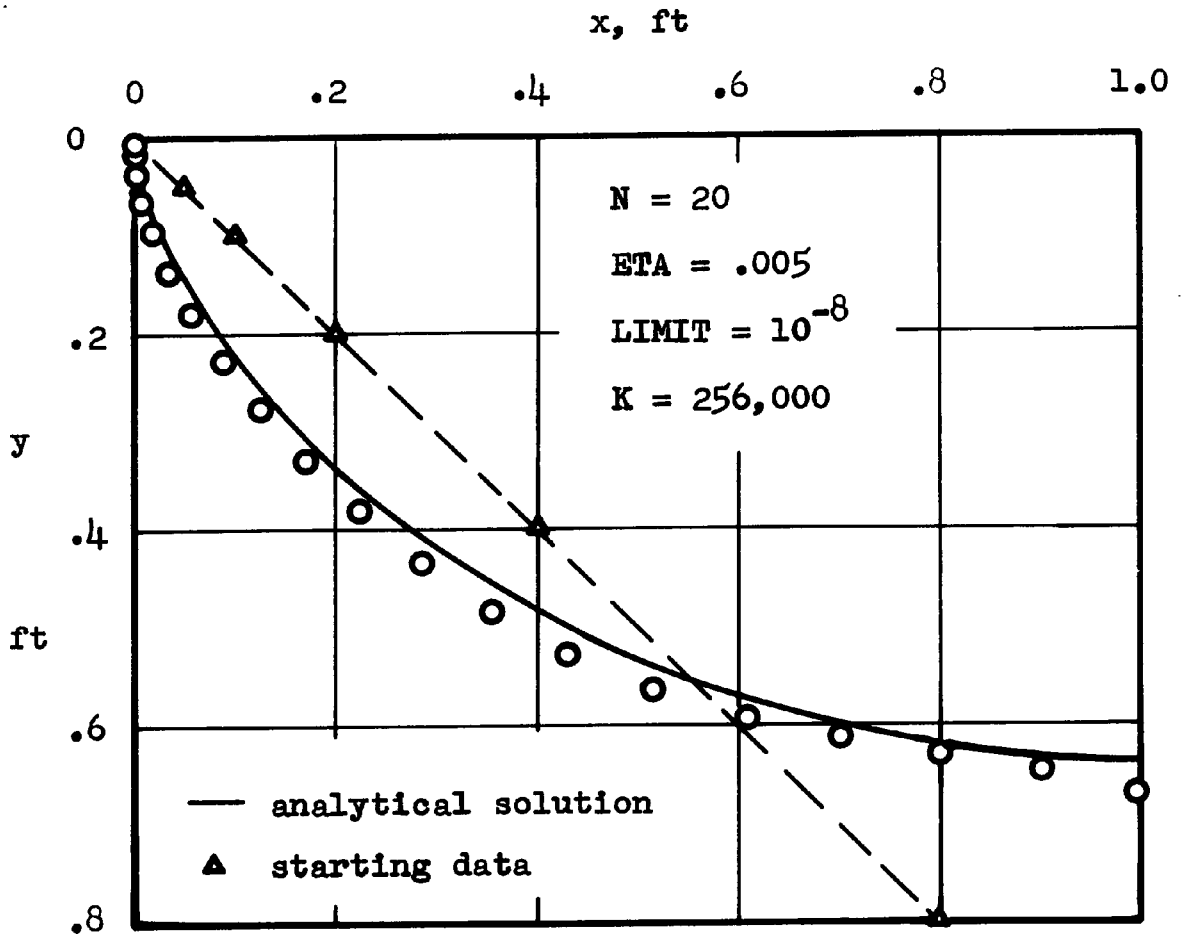



FIGURE 6-1 Unconstrained Brachistochrone Problem: Optimal Trajectory

approximation is reasonably accurate except for the y and u values near the end of the path.

The computations are repeated for ETA reduced to .002; all other factors remain the same. The computer time for this run is 143.4 seconds. This significant increase in running time, resulting from only a modest decrease in ETA, is indicative of the round-off error difficulties discussed in Section 5.3. Still, the use of a smaller value of ETA does provide a substantial increase in accuracy over that

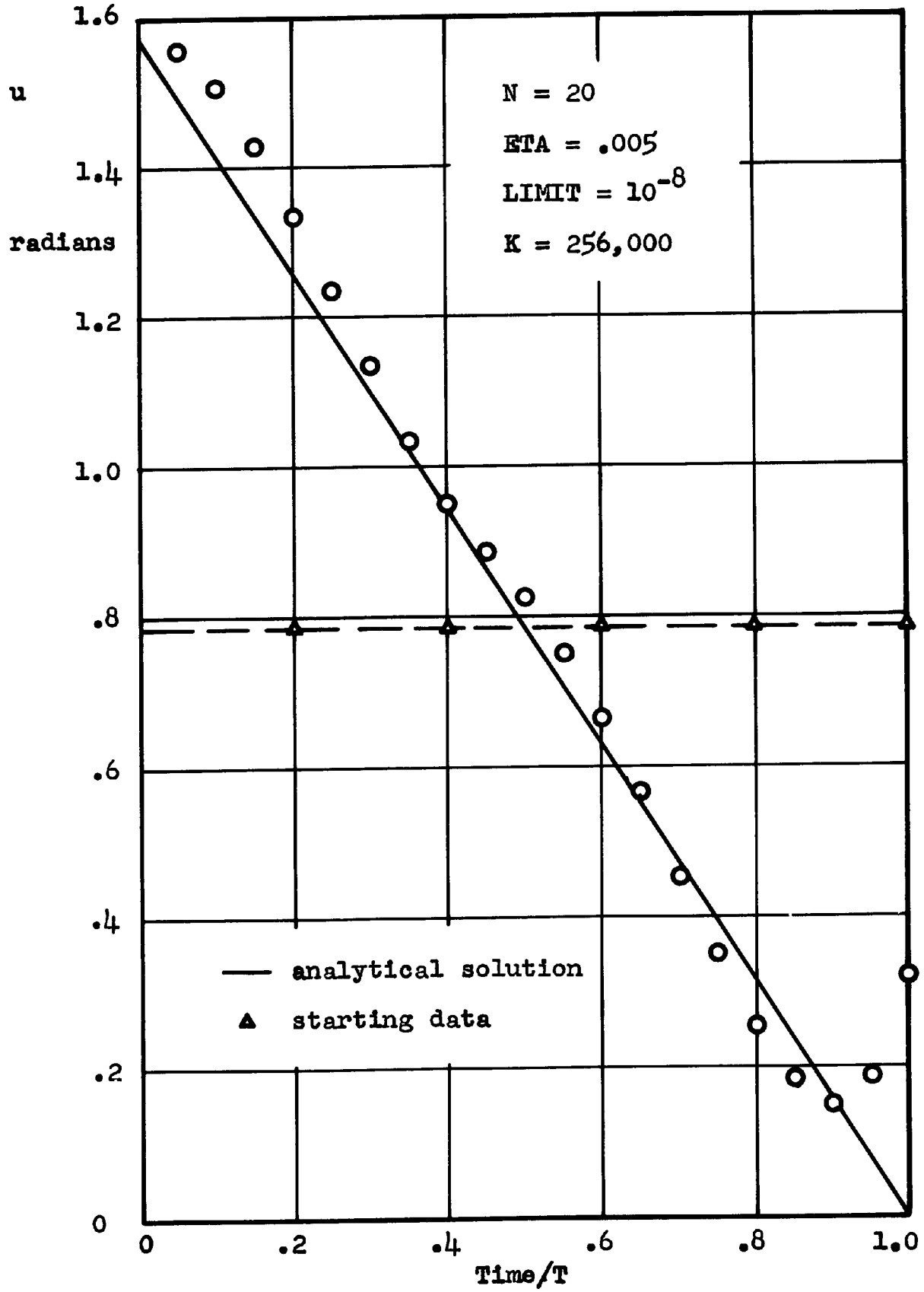



FIGURE 6-2 Unconstrained Brachistochrone Problem: Optimal Control Time History

| t/T | x, ft | | y, ft | | u, rad | |
|---|---|---|---|---|---|---|
| .05 | .00021 | .00019 | .00392 | .00402 | 1.4923 | 1.5240 |
| .10 | .00164 | .00112 | .01558 | .01587 | 1.4137 | 1.4664 |
| .15 | .00549 | .00387 | .03469 | .03536 | 1.3352 | 1.3981 |
| .20 | .01290 | .00967 | .06079 | .06215 | 1.2566 | 1.3186 |
| .25 | .02492 | .02006 | .09323 | .09557 | 1.1781 | 1.2201 |
| .30 | .04248 | .03657 | .13121 | .13462 | 1.0996 | 1.1261 |
| .35 | .06638 | .05992 | .17380 | .17822 | 1.0210 | 1.0343 |
| .40 | .09727 | .09070 | .21995 | .22526 | .9425 | .9521 |
| .45 | .13561 | .12877 | .26852 | .27499 | .8639 | .8873 |
| .50 | .18169 | .17397 | .31831 | .32671 | .7854 | .8164 |
| .55 | .23561 | .22687 | .36810 | .37907 | .7069 | .7454 |
| .60 | .29727 | .28770 | .41667 | .43057 | .6283 | .6564 |
| .65 | .36638 | .35753 | .46282 | .47803 | .5498 | .5308 |
| .70 | .44248 | .43633 | .50541 | .51840 | .4712 | .4240 |
| .75 | .52492 | .52153 | .54339 | .55237 | .3927 | .3391 |
| .80 | .61290 | .61159 | .57583 | .58044 | .3142 | .2670 |
| .85 | .70549 | .70541 | .60193 | .60267 | .2356 | .1985 |
| .90 | .80164 | .80192 | .62104 | .61984 | .1571 | .1610 |
| .95 | .90020 | .90000 | .63270 | .63516 | .0785 | .1540 |
| 1.00 | 1.00000 | .99916 | .63662 | .65153 | .0000 | .1786 |

**TABLE 6-1** Unconstrained Brachistochrone Problem: An Accuracy Comparison for ETA = .002

of the previous results. For N=20 the trajectory and control points corresponding to ETA = .002 are listed in Table 6-1. For the purpose of accuracy comparison, the corresponding analytical values are shown in the left columns. We note that $x_{20}$ is within 0.1 % of the maximum horizontal distance of one foot.

### 6.1.2 A Constrained Brachistochrone Problem

The brachistochrone problem may be further modified by insisting that the curve lie entirely above or on a given straight line. Specifically, we choose the inequality constraint

$$y \leq .1 + .5x \quad . \tag{6-2}$$

Since (6-2) does not explicitly contain the control u, it is a state space constraint. When (6-2) is enforced, we will refer to the resulting problem as the constrained brachistochrone problem.

It can be shown that the solution to the constrained brachistochrone problem consists of three subarcs: (1) a cycloid that has infinite slope at the origin and is tangent to the constraint line, (2) a portion of the constraint line, and (3) another cycloid which is tangent to the constraint line and has zero slope at the end point (see Figure 6-3). This analytical solution may be found in reference (B10); it will be used to provide a means of judging the accuracy of our numerical results. This analytical solution can also be used to determine that a maximum

horizontal distance of exactly one foot corresponds to a fixed final time of .323201 seconds. We will use this value for T.

Our discrete approximation will involve the same 3N problem variables as before. The inequality (6-2) is to hold at all of the N time points. The third-order Adams interpolation differencing scheme (5-5) will be used.

The convergence for increasing K and N is presented in Table 6-2 in terms of the $x_N$ present after each K-min.

| N | K | $x_N$, feet |
|---|---|---|
| 5 | 1000 | .99425 |
| 5 | 4000 | .98180 |
| 5 | 16000 | .97672 |
| 5 | 64000 | .97541 (SOE = .97508) |
| 10 | 64000 | .99711 |
| 10 | 256000 | .99614 (FOE = .99590) |
| 20 | 256000 | .99818 |

TABLE 6-2 Constrained Brachistochrone Problem: $x_N$ Obtained for Each K-min

The termination tolerances used are ETA = .1 and LIMIT = $10^{-8}$. First-order estimates (FOE) and second-order estimates (SOE) for $x_N$ are shown where appropriate. We note that as K is increased $x_N$ is actually driven further from its optimal

value of one. However, as N is doubled $x_N$ more closely approximates max x(T). For 20 time points $x_N$ is within .2% of the optimum value. The remaining discrepancy can be attributed to three causes: (1) the unconstrained minimum has not been accurately determined (ETA too large); (2) the constraints are not well enough satisfied (K too small); (3) the set of difference equations do not adequately represent the differential equations (N too small).

These results have been obtained using the rather crude initial vector: $z_o$ = (.05, .15, .4, .7, 1, .05, .15, .3, .45, .5, .7854, .7854, .4636, .4636, 0). Indeed, one of the principal advantages of this discrete-variable method is that it is not necessary to provide an accurate first guess; almost any reasonable set of values will suffice. The $z_o$ supplied by the user is quickly modified since the dimension of $z_o$ is usually small and the gradient vector will initially be quite large in magnitude. We recall that the algorithm automatically provides a $z_o$ for all subsequent K-mins.

Figure 6-3 shows the optimal trajectory obtained for this same computer run. The analytical solution is represented by the solid line; the arrows indicate the points where the exact solution enters and leaves the constraint line. The complete numerical solution for N=5, 10, and 20 requires a running time of 65.1 seconds and provides a close approximation to the exact solution. It is interesting to observe that a reasonably good approximation can be obtained

with only 15 variables (N=5). This portion of the computations takes only 9.1 seconds.

The corresponding optimal control time history is presented in Figure 6-4. The optimal control is a continuous function of time which is made up of three sections each of which is a linear time function. Therefore, at the two joining corners the first derivative of u is discontinuous. As was mentioned earlier, the control points appear to be samples of a very smooth time function since the difference equations are based on polynomial interpolation. This is evident in Figure 6-4 where the control points have difficulty in approximating the exact optimal control near the corners. Also, in general one must expect somewhat less accuracy for the control points since they do not have the benefit of an integration process as do the trajectory points.

The complete results for this computer run are listed in Table 6-3 along with the corresponding analytical results. In each case the exact values are shown in the left column. These numerical data have been computed using single precision arithmetic. For reasonable time expenditures only modest accuracy gains can be obtained without the use of double precision in conjunction with smaller ETA and larger N.

We should emphasize at this point that the discrete-variable approach demonstrates the ease of handling state space constraints of this type. The constraint is implemented simply by checking whether the inequality (6-2) is violated for each time point. Then each violation is

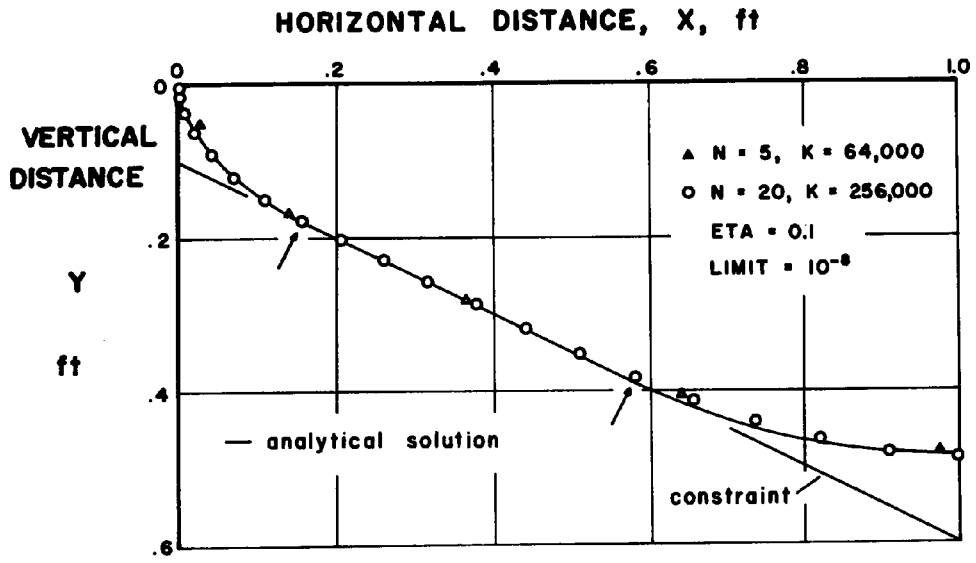



FIGURE 6-3 Constrained Brachistochrone Problem: Optimal Trajectory

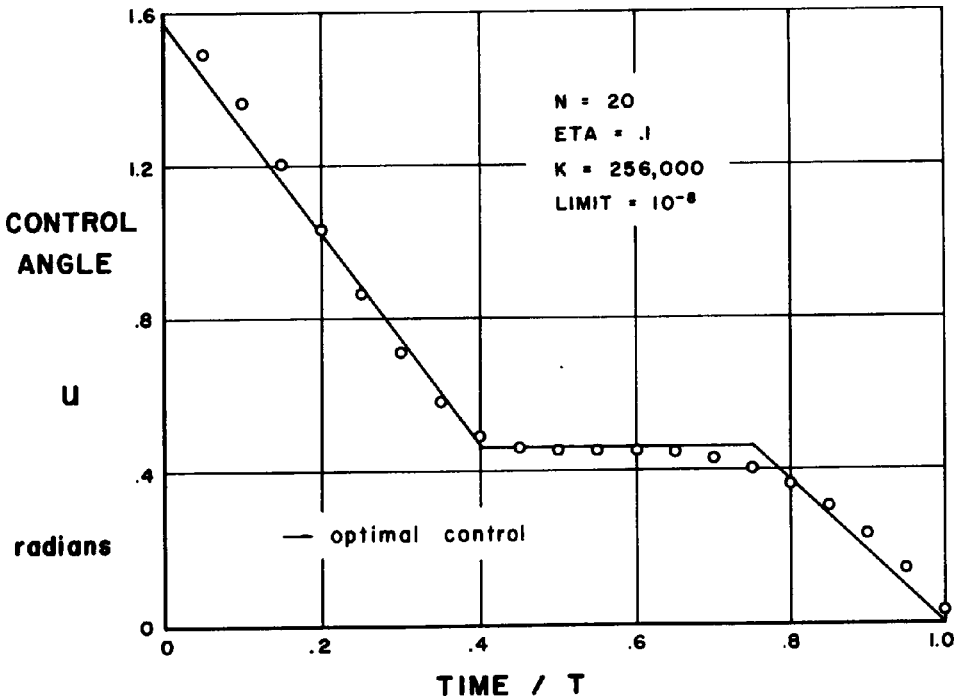



FIGURE 6-4 Constrained Brachistochrone Problem: Optimal Control Time History

| t/T | x, feet | | y, feet | | u, radians | |
|---|---|---|---|---|---|---|
| .05 | .00038 | .00033 | .00418 | .00422 | 1.43379 | 1.49232 |
| .10 | .00303 | .00222 | .01640 | .01664 | 1.29679 | 1.36281 |
| .15 | .01002 | .00815 | .03576 | .03653 | 1.15979 | 1.20348 |
| .20 | .02314 | .02073 | .06080 | .06220 | 1.02278 | 1.03246 |
| .25 | .04369 | .04167 | .08967 | .09124 | .88578 | .86593 |
| .30 | .07243 | .07146 | .12020 | .12113 | .74878 | .71458 |
| .35 | .10951 | .10955 | .15012 | .14989 | .61177 | .58533 |
| .40 | .15444 | .15463 | .17719 | .17664 | .47477 | .49527 |
| .45 | .20494 | .20485 | .20247 | .20243 | .46365 | .46548 |
| .50 | .25881 | .25877 | .22941 | .22920 | .46365 | .45740 |
| .55 | .31605 | .31615 | .25803 | .25741 | .46365 | .45656 |
| .60 | .37665 | .37687 | .28833 | .28725 | .46365 | .45672 |
| .65 | .44062 | .44099 | .32031 | .31855 | .46365 | .44986 |
| .70 | .50794 | .50876 | .35397 | .35068 | .46365 | .43393 |
| .75 | .57864 | .58043 | .38932 | .38278 | .46365 | .40612 |
| .80 | .65418 | .65624 | .42285 | .41361 | .37163 | .36421 |
| .85 | .73540 | .73630 | .45021 | .44163 | .27872 | .30663 |
| .90 | .82105 | .82045 | .47046 | .46500 | .18582 | .23258 |
| .95 | .90975 | .90812 | .48289 | .48171 | .09291 | .14159 |
| 1.00 | 1.00000 | .99818 | .48708 | .48979 | .00000 | .03464 |

TABLE 6-3 **Constrained Brachistochrone Problem: An Accuracy Comparison**

multiplied by an appropriate constant, raised to the fourth power, and added to PENLTY. The amount of additional effort is quite small both in terms of programming and computer time. Although the constraint considered here is linear, nonlinear constraints would in principle be no more difficult. Again, in this discrete-variable formulation state space constraints are no more difficult than are constraints involving the control. No assumptions are necessary regarding the number or position of points at which the solution enters or leaves the constraint boundary.

## 6.2 An Example: Avoidance of Certain Stability Difficulties

Many indirect methods (see Chapter 1) face computational difficulties because of the stability properties of the differential equations involved. For example, consider the illustrative problem

$$\text{minimize} \int_0^1 (x^2+u^2)\, dt \text{ subject to}$$

$$\dot{x} = -ax + u,\ x(0) = 1, \text{ and } x(1) = 0\ . \qquad (6\text{-}3)$$

An application of the necessary conditions of the calculus of variations leads to the linear two-point boundary value problem

$$\ddot{x} - (1+a^2)x = 0,\ x(0) = 1,\ x(1) = 0\ . \qquad (6\text{-}4)$$

There is no difficulty in obtaining an analytical solution. In fact, the optimal control $u_*$ and the corresponding optimal trajectory $x_*$ are given by

$$u_*(t) = (a-bw)\cosh wt + (w-ab)\sinh wt$$
$$x_*(t) = \cosh wt - b \sinh wt$$
$$\text{where } w = (1+a^2)^{\frac{1}{2}} \text{ and } b = \coth w. \qquad (6\text{-}5)$$

However, the differential equation in (6-4) is clearly unstable for all real values of the parameter a. Any numerical method which depends in part on the numerical integration of the equation, either forward or backward in time, is susceptible to a dangerous growth of computational errors. For $a^2$ sufficiently large these errors can completely obscure the true solution; indeed, the integration process may break down altogether.

In the discrete-variable algorithm used here there is no longer any notion of forward or backward integration in time. With the use of a set of difference equations all time points and their corresponding discrete variables are put on an equal basis and treated simultaneously. We can therefore expect to avoid the propagation and growth of computational errors mentioned above.

The construction of a discrete-variable model for the example problem (6-3) can proceed in one of two ways. We can devise a discrete approximation J(z) for the integral to be minimized, or we can convert the problem to Mayer form as discussed in Section 2.1. Both approaches have been successfully used. The former approach involves N fewer variables, but the results are sensitive to the discrete approximation J(z) used.

In Figure 6-5 we present some results for the case

$a = -20$. For this large a magnitude of a, there is even some difficulty in evaluating the exact solution (6-5) on the computer. For these results we have used the third-order differencing scheme (5-5) to approximate the differential equation in (6-3) and the Euler-Maclaurin formula (K7)

$$\int_o^T f(t)\,dt \cong h\sum_{i=0}^{N-1}\tfrac{1}{2}(f_i+f_{i+1}) - h^2(\dot{f}_N-\dot{f}_o)/12 \qquad (6\text{-}6)$$

as the discrete approximation for the integral to be minimized. The formula (6-6) represents an improvement on the familiar trapezoidal rule. In our case the derivative $\dot{f}$ introduces both an $\dot{x}$ and a $\dot{u}$; $\dot{x}$ is readily available, but $\dot{u}$ requires numerical approximation. No particular difficulty has been experienced in the numerical computations. The entire problem with N=20 and 40 takes only 22 seconds. The termination tolerances ETA = .1 and LIMIT = $5(10^{-8})$ are used. The minimum obtained for $\int_o^1 (x^2+u^2)\,dt$ is 40.3957; this compares favorably with the exact value 40.0250.

As this simple example demonstrates, the discrete-variable algorithm is able to avoid stability difficulties that result from the growth of computational errors during numerical integration. The entire "solution," as represented by the discrete variables, is governed by the set of difference equations and as such is available at all stages of the computations. As the computations progress, the difference equations are more nearly satisfied since they are added to the cost function in the form of penalty functions.

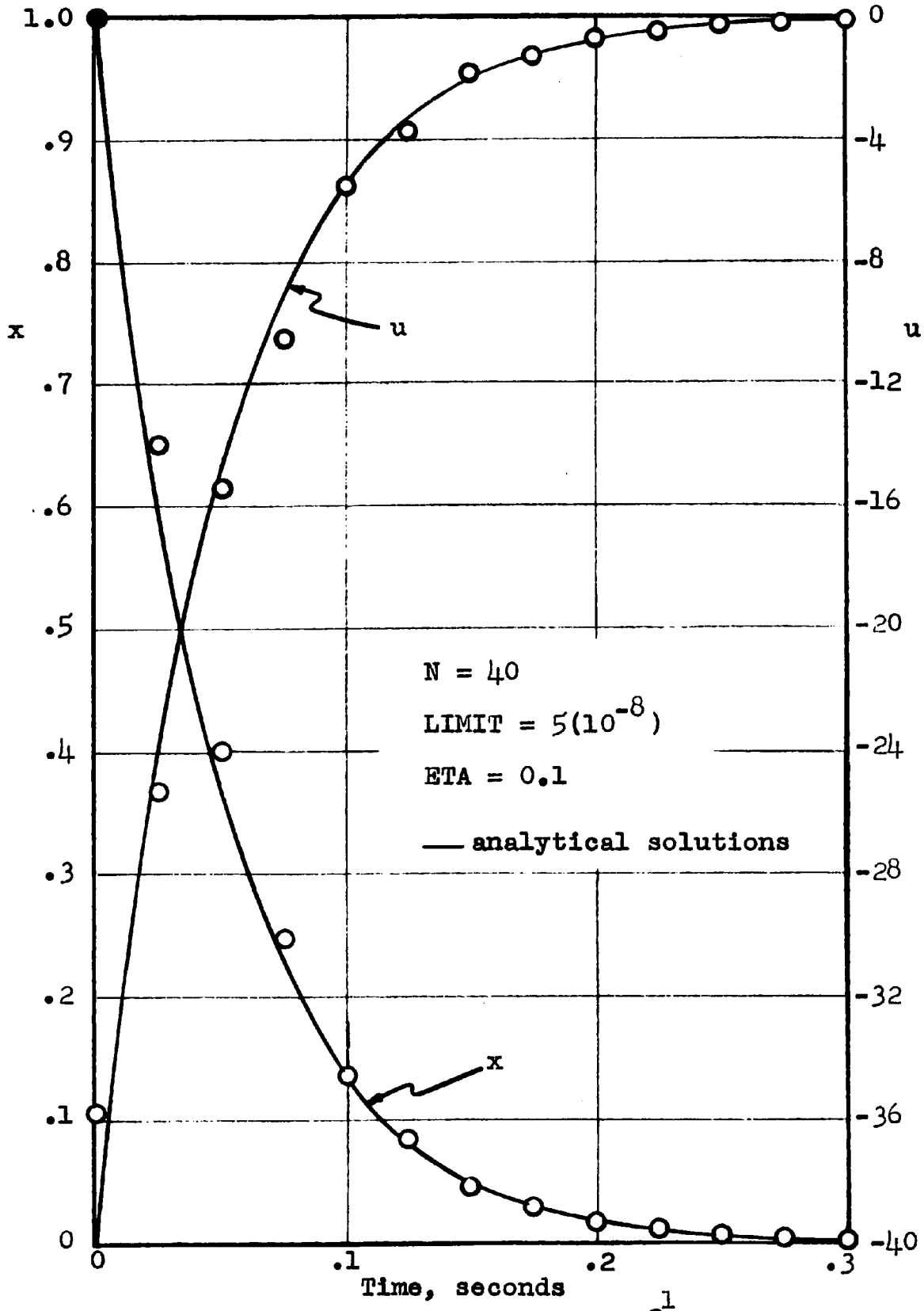



FIGURE 6-5 Example Problem: Minimize $\int_0^1 (x^2+u^2)\,dt$

Subject to $\dot{x} = 20x + u$, $x(0) = 1$, $x(1) = 0$

## 6.3 A Simple Rocket Problem

Next we consider a simple rocket problem for which the optimal control is known and exhibits continuous derivatives of all orders. The problem has been used by several authors (D3, B7).

> Assuming constant gravitational acceleration $g_o$, vacuum flight, and constant rocket thrust acceleration A, find the thrust direction time history required for the rocket to attain a given altitude at a given time with zero vertical speed and maximum horizontal speed. The terminal range is not specified.

Mathematically, the optimal control problem may be stated as follows. Minimize $-V_x(T)$ subject to

$$\begin{aligned} \dot{X} &= V_x, & X(0) &= 0 \\ \dot{Y} &= V_y, & Y(0) &= 0 \\ \dot{V}_x &= A \cos u, & V_x(0) &= 0 \\ \dot{V}_y &= A \sin u - g_o, & V_y(0) &= 0 \end{aligned} \qquad (6\text{-}7)$$

and the terminal conditions: $V_y(T) = 0$, and $Y(T)$ and $T$ are both specified. Here X is the horizontal range, Y is the altitude, $V_x$ is the horizontal velocity component, $V_y$ is the vertical velocity component, and u is the inclination of the thrust vector to the horizontal. The trajectory geometry is sketched in Figure 6-6.

It can be shown (B7) that the optimal control is given by $u_*(t) = \tan^{-1}(b-ct)$. However, one must first solve two simultaneous transcendental equations, say $e^1(b,c) = 0$ and $e^2(b,c) = 0$, for b and c. These equations were solved quickly and accurately by applying Davidon's

variable metric method to the unconstrained minimization problem: $\min \sum_{i=1}^{2} (e^i)^2$. With $u_*(t)$ determined, the state variables are obtained from the analytical integration of the system equations in (6-7). It is this semi-analytical solution which we take as the "exact" solution used later for accuracy comparisons.

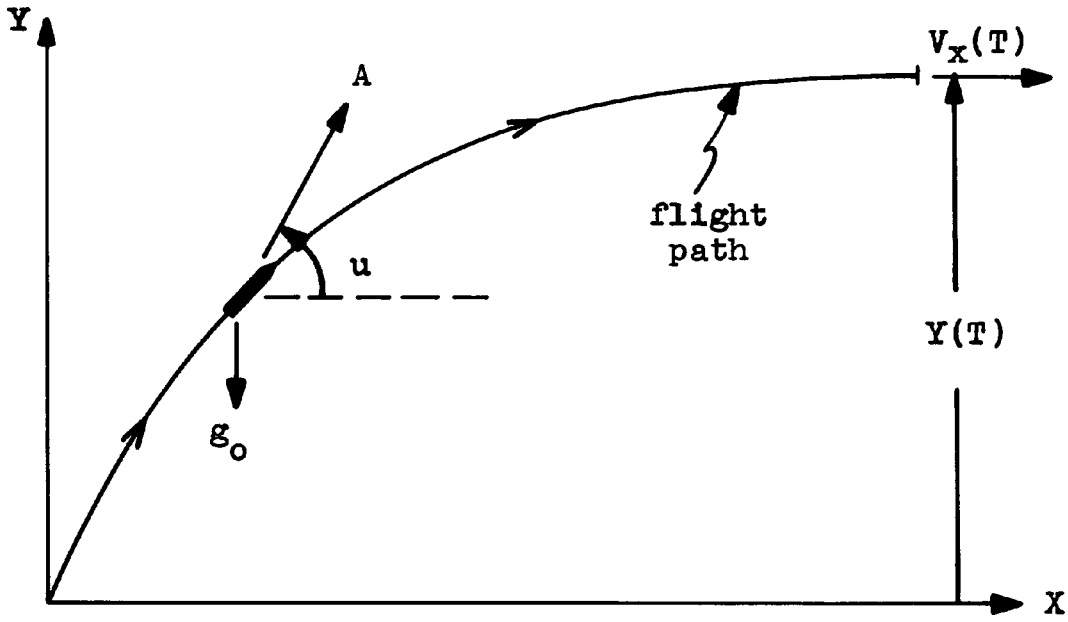



FIGURE 6-6 A Simple Rocket Problem: Trajectory Geometry

In turning to the discrete-variable calculations, a non-dimensional formulation in which all problem variables have a maximum magnitude near unity is strongly recommended. Otherwise, an overflow termination may result during the early computations. For this problem Y(T) and T were taken as the basic units of length and time, respectively.

The numerical results presented here correspond to the values: Y(T) = 100,000 feet, T = 100 seconds, $A/g_o$ = 2, and $g_o$ = 32 feet per second squared. The third-order differencing scheme (5-5) is used throughout. For 99 variables (N=20) an approximate solution has been obtained in less than a minute of computer running time with ETA = .1 and LIMIT = $10^{-8}$. In another computer run a 199-variable (N=40) solution has been reached in slightly more than 4 minutes with ETA = .08 and LIMIT = $10^{-8}$. This latter solution is displayed along with the exact results in Figures 6-7, 6-8, and 6-9. Figure 6-7 presents the optimal trajectory. The discrete-variable approach can be seen to provide a good approximation to the optimal trajectory with the largest discrepancies occurring in the mid-portion of the flight. The hodograph corresponding to this problem is shown in Figure 6-8. The value obtained for $V_x(T)$ is 3500.3 feet per second which is only .22% less than the exact value of 3508.1 feet per second. Figure 6-9 contains the optimal control time history. The control points approximate the optimal thrust direction very closely except near the end of the flight. We have achieved a better approximation to the optimal control for this problem than for the constrained brachistochrone problem since the optimal control for the latter problem exhibited a discontinuous first derivative (see Figure 6-4).

In the ETA = .08 computer run described above, the 99-variable solution terminates with (g,g) = .068 and PENLTY = $2.5(10^{-9})$. For the purpose of direct comparison

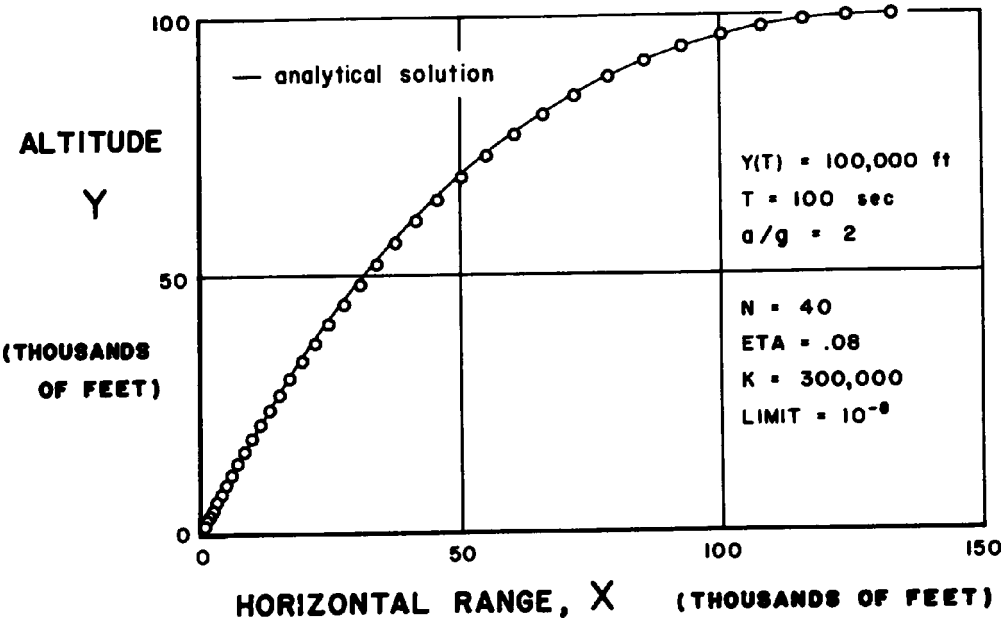



FIGURE 6-7 A Simple Rocket Problem: Optimal Trajectory

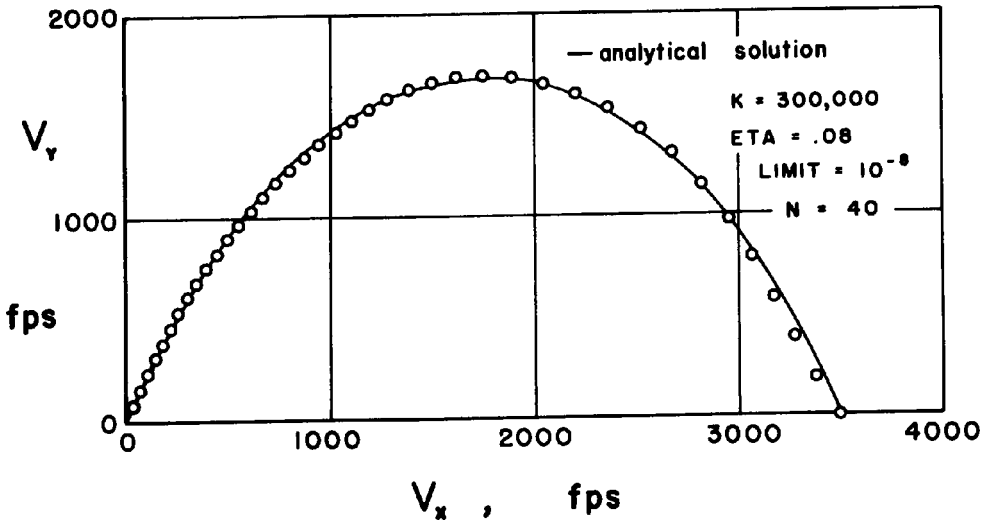



FIGURE 6-8 A Simple Rocket Problem: Optimal Trajectory Hodograph

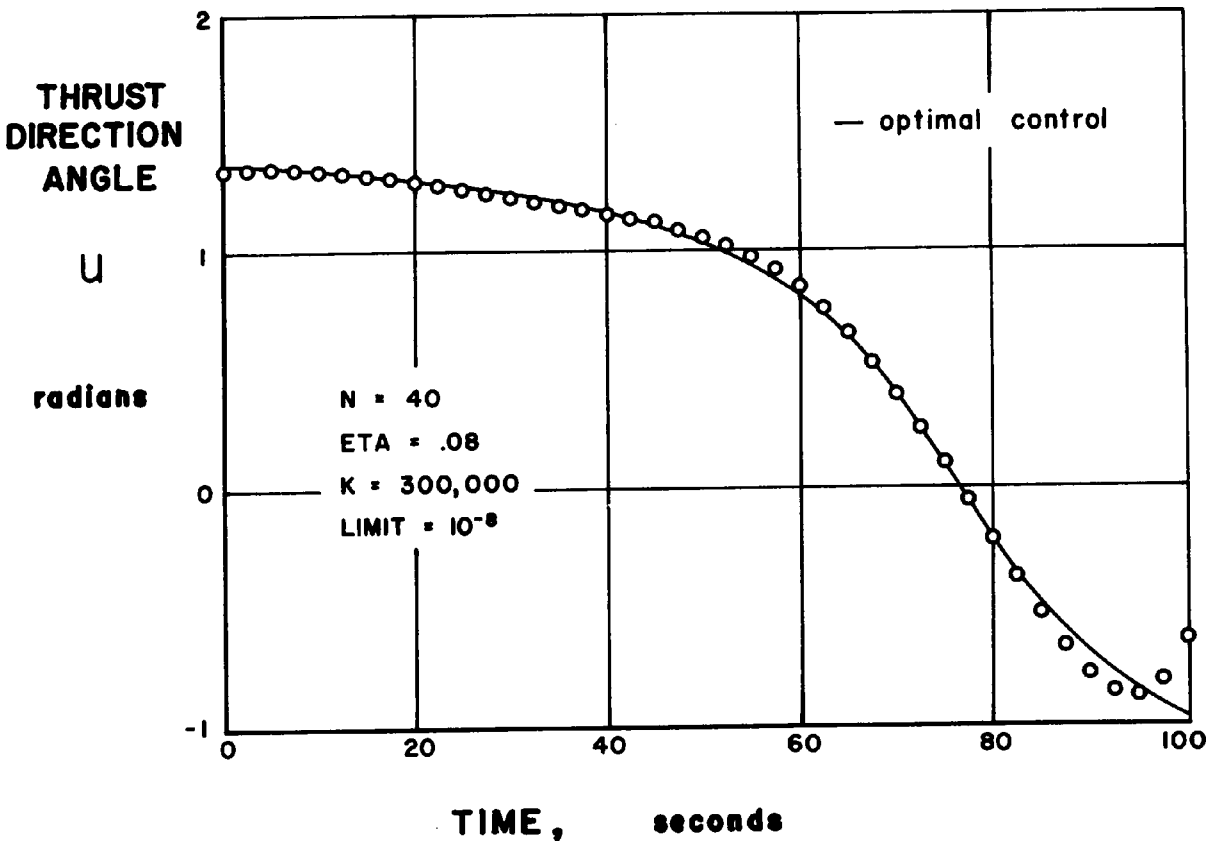



FIGURE 6-9 A Simple Rocket Problem: Optimal Control Time History

| t/T | x, ft | | | y, ft | | |
|---|---|---|---|---|---|---|
| | E | SP | DP | E | SP | DP |
| .00 | 0 | 0 | 0 | 0 | 0 | 0 |
| .05 | 165 | 181 | 168 | 383 | 380 | 383 |
| .10 | 676 | 723 | 679 | 1528 | 1519 | 1529 |
| .15 | 1558 | 1634 | 1558 | 3429 | 3415 | 3432 |
| .20 | 2837 | 2945 | 2840 | 6080 | 6060 | 6084 |
| .25 | 4548 | 4698 | 4551 | 9472 | 9443 | 9477 |
| .30 | 6729 | 6942 | 6732 | 13593 | 13548 | 13599 |
| .35 | 9425 | 9728 | 9429 | 18429 | 18359 | 18435 |
| .40 | 12691 | 13106 | 12696 | 23959 | 23856 | 23967 |
| .45 | 16593 | 17132 | 16599 | 30157 | 30016 | 30166 |
| .50 | 21213 | 21866 | 21221 | 36985 | 36810 | 36993 |
| .55 | 26653 | 27388 | 26664 | 44385 | 44196 | 44393 |
| .60 | 33041 | 33810 | 33056 | 52272 | 52101 | 52279 |
| .65 | 40535 | 41294 | 40554 | 60512 | 60397 | 60515 |
| .70 | 49319 | 50048 | 49341 | 68887 | 68853 | 68886 |
| .75 | 59573 | 60272 | 59598 | 77070 | 77114 | 77063 |
| .80 | 71407 | 72078 | 71435 | 84611 | 84733 | 84602 |
| .85 | 84795 | 85435 | 84831 | 91025 | 91214 | 91012 |
| .90 | 99594 | 100168 | 99643 | 95902 | 96092 | 95882 |
| .95 | 115616 | 116043 | 115688 | 98954 | 99047 | 98941 |
| 1.00 | 132684 | 132963 | 132781 | 100000 | 100000 | 100000 |

TABLE 6-4 **A Simple Rocket Problem: An Accuracy Comparison**

| t/T | $V_x$, ft/sec | | | $V_y$, ft/sec | | |
|---|---|---|---|---|---|---|
| | E | SP | DP | E | SP | DP |
| .00 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| .05 | 66.9 | 72.4 | 67.2 | 157.9 | 151.7 | 152.9 |
| .10 | 138.3 | 144.5 | 137.8 | 304.8 | 303.5 | 305.0 |
| .15 | 215.1 | 221.0 | 215.4 | 455.5 | 454.2 | 455.4 |
| .20 | 297.9 | 304.9 | 298.0 | 604.6 | 603.0 | 604.6 |
| .25 | 387.8 | 398.1 | 387.9 | 751.7 | 749.1 | 751.7 |
| .30 | 486.0 | 501.3 | 486.1 | 896.2 | 892.0 | 896.3 |
| .35 | 594.2 | 614.7 | 594.3 | 1037.4 | 1031.3 | 1037.4 |
| .40 | 714.3 | 738.5 | 714.6 | 1173.9 | 1166.3 | 1173.9 |
| .45 | 849.2 | 873.6 | 849.5 | 1304.1 | 1296.4 | 1304.0 |
| .50 | 1002.2 | 1022.4 | 1002.7 | 1425.1 | 1419.6 | 1424.9 |
| .55 | 1178.0 | 1189.9 | 1178.7 | 1532.3 | 1532.1 | 1532.0 |
| .60 | 1382.4 | 1384.2 | 1383.2 | 1618.2 | 1626.0 | 1617.7 |
| .65 | 1621.4 | 1616.6 | 1622.3 | 1670.4 | 1684.9 | 1669.6 |
| .70 | 1898.5 | 1892.6 | 1899.4 | 1669.0 | 1685.0 | 1667.9 |
| .75 | 2207.2 | 2201.8 | 2208.0 | 1588.7 | 1603.8 | 1587.5 |
| .80 | 2525.4 | 2520.0 | 2526.7 | 1411.1 | 1426.2 | 1409.9 |
| .85 | 2824.7 | 2816.6 | 2826.1 | 1140.7 | 1148.8 | 1138.5 |
| .90 | 3088.4 | 3067.0 | 3091.7 | 800.5 | 789.9 | 799.9 |
| .95 | 3314.5 | 3276.7 | 3320.0 | 414.6 | 387.9 | 417.9 |
| 1.00 | 3508.1 | 3497.4 | 3510.7 | 0.0 | 0.0 | 0.0 |

**TABLE 6-4 (continued)** A Simple Rocket Problem: An Accuracy Comparison

| t/T | u, radians | | |
|---|---|---|---|
| | E | SP | DP |
| .00 | 1.3670 | 1.3390 | 1.3608 |
| .05 | 1.3532 | 1.3462 | 1.3578 |
| .10 | 1.3375 | 1.3384 | 1.3354 |
| .15 | 1.3193 | 1.3187 | 1.3197 |
| .20 | 1.2981 | 1.2911 | 1.2981 |
| .25 | 1.2732 | 1.2592 | 1.2733 |
| .30 | 1.2434 | 1.2257 | 1.2434 |
| .35 | 1.2073 | 1.1912 | 1.2070 |
| .40 | 1.1627 | 1.1551 | 1.1624 |
| .45 | 1.1066 | 1.1135 | 1.1062 |
| .50 | 1.0343 | 1.0575 | 1.0338 |
| .55 | .9387 | .9770 | .9381 |
| .60 | .8097 | .8480 | .8088 |
| .65 | .6331 | .6494 | .6320 |
| .70 | .3960 | .3923 | .3945 |
| .75 | .1019 | .1018 | .1001 |
| .80 | -.2104 | -.2172 | -.2115 |
| .85 | -.4869 | -.5353 | -.4911 |
| .90 | -.7017 | -.7916 | -.6696 |
| .95 | -.8598 | -.8853 | -.8915 |
| 1.00 | -.9756 | -.6474 | -.9263 |

TABLE 6-4 (continued) A Simple Rocket Problem: An Accuracy Comparison

this solution and the corresponding analytical results are listed in Table 6-4. For each variable the exact values are shown under the column heading E, and the approximate values are shown under the column heading SP. Several attempts to obtain higher accuracy by selecting smaller ETA values have been frustrated by very slow convergence. It is clear that for these single precision runs the round-off errors are seriously hampering progress toward the minimum.

This same 99-variable problem has been re-run in double precision starting from the same initial data used for the single precision run. The value of K is still 300,000. After 14 minutes of computing time and 867 steps the calculations are terminated. At this point (g,g) = .028, and PENLTY = $2.55(10^{-9})$. The results are considerably more accurate than before; they are listed in Table 6.4 under the DP column headings. In most cases the trajectory points agree with the exact values to 3 or 4 significant figures.

## 6.4 A Minimum-Time Orbital Transfer Problem

We now treat what has become almost a standard test problem in flight path optimization. The problem is to transfer a spacecraft between two coplanar circular orbits in minimum time. The spacecraft moves under the influence of two forces: a single inverse-square gravitational force and a thrust force of constant magnitude. The direction of this thrust force serves as the control function. The mass loss is assumed to be linear with time; thus the minimum-time transfer will be equivalent to a minimum-fuel transfer.

There is no known analytical solution to this problem.

Mathematically, the problem is to find a thrust direction program u(t) that will minimize the transfer time T subject to the system

$$\begin{aligned} \dot{r} &= v_r \\ \dot{v}_r &= (v_c^2/r) - (\mu/r^2) + (F \sin u)/(m_o+\dot{m}t) \\ \dot{v}_c &= -(v_r v_c/r) + (F \cos u)/(m_o+\dot{m}t) \end{aligned} \qquad (6\text{-}8)$$

and the specified boundary values $r(0)$, $v_r(0)$, $v_c(0)$, $r(T)$, $v_r(T)$, and $v_c(T)$. In (6-8) r is the distance between the spacecraft and the center of attraction; $v_r$ and $v_c$ are the radial and circumferential velocity components; $\mu$ is the appropriate gravitational constant; F is the thrust magnitude; $m_o$ is the initial vehicle mass; and $\dot{m}$ is the constant rate of fuel consumption. The control angle u is measured clockwise from the local horizontal.

If the gravitational fields of the Earth and Mars are neglected, the above formulation can be used to approximate an Earth-to-Mars transfer with low-thrust ion rocket propulsion. Since the terminal heliocentric angle is not specified, we are tacitly assuming that the launch date is properly chosen to allow a minimum-time rendezvous with Mars. Using the parameter values F = .127 lb, $m_o$ = 46.58 slugs, and $\dot{m}$ = -6.937 slugs/sec, several authors (K8, K4, M1, H5) have obtained minimum Earth-to-Mars transfer times of 193.0 to 193.2 days.

This problem serves as an example of a variable-time problem. All of the other methods used on this problem

require special treatment for variable-time problems; in effect, the transfer time T is treated as a specified parameter and is changed at the end of each iteration. In our discrete-variable formulation, we simply include T as an additional problem variable. In principle there is no more difficulty with variable-time problems than with fixed-time problems. However, in practice, the gradient vector element $\partial F/\partial T$ is more difficult to compute than the others since $h = T/N$ is contained in each difference equation. For this problem the time is also present explicitly in the differential equations (6-8).

For numerical computations on this Earth-to-Mars transfer the variables have been normalized so that the unit of length is the astronomical unit r(0); the unit of mass is $m_o$; and the unit of time is chosen so that $\mu = 1$. With these choices the unit of time is 58.18 days. The normalized boundary values become

$$r(0) = 1.000 \qquad r(T) = 1.525$$
$$v_r(0) = 0.000 \qquad v_r(T) = 0.000$$
$$v_c(0) = 1.000 \qquad v_c(T) = .8098 \ .$$

The third-order differencing scheme (2-5) has been used in obtaining the results to be presented here. Our computations involve 4N-1 discrete variables. Again, with N=5 rough choices suffice for the initial vector $z_o$: namely T = 3.06 and the values

| $t_i/T$ | $r_i$ | $v_{r_i}$ | $v_{c_i}$ | $u_i$ |
|---|---|---|---|---|
| 0.0 | | | | 1.047 |
| .2 | 1.1 | .1 | .96 | 1.047 |
| .4 | 1.2 | .2 | .92 | 1.047 |
| .6 | 1.3 | .2 | .88 | 4.712 |
| .8 | 1.4 | .1 | .84 | 4.712 |
| 1.0 | | | | 4.712 . |

This choice for the initial $u_i$ corresponds to a thrust direction 60 degrees above the horizontal for the first half of the flight and radially inward thereafter.

The minimum-time trajectory is shown in Figure 6-10.

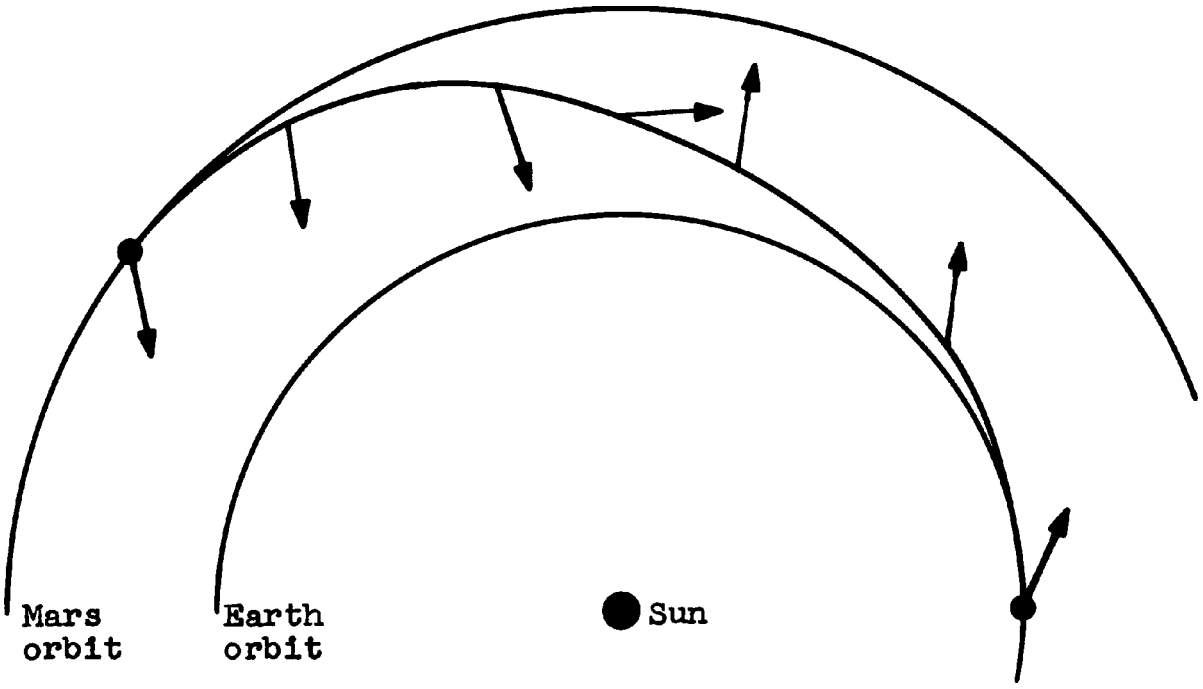



FIGURE 6-10 Minimum-Time Earth-to-Mars Trajectory

Although the heliocentric angle $\theta$ does not enter our problem directly, it may be obtained by integrating $\dot{\theta} = v_c/r$ after the optimal $r(t)$ and $v_c(t)$ have been determined. The arrows

in Figure 6-10 indicate the optimal thrust direction at representative points along the trajectory.

With ETA = .1, a 79-variable (N=20) solution has been obtained in 1 minute 35.5 seconds; the transfer time is 194.46 days. With ETA = .02, a 159-variable (N=40) solution has been obtained in 3 minutes 38.8 seconds; the transfer time is 193.79 days. The superior results of the latter run are primarily due to the use of a smaller ETA. In both cases, LIMIT = $10^{-8}$, and computation has begun with a 19-variable (N=5) problem. Figure 6-11 shows the control points obtained from these two runs. The solid line represents an optimal control time history obtained recently by Handelsman (H3). Although the control points do not match this solution as well as one would like, the minimum transfer times obtained are within .8% of the established value. The trajectory points obtained from the ETA = .02 run are presented in Figure 6-12; only alternate points are plotted. An optimal solution obtained by Herzberg (H4) using the generalized Newton-Raphson method is indicated by a solid line. The trajectory points are noticeably more accurate than the control points.

It is very difficult to make comparisons between numerical solutions obtained by several investigators. However, some idea of the computational efficiency of the discrete-variable algorithm used here can be gained by comparing the published computer running times for this Earth-to-Mars transfer problem:

| Reference | Method | Computer Time |
|---|---|---|
| K8 | steepest descent | 4 minutes |
| K4 | second variation | 2 minutes |
| M1 | Newton-Raphson | 36 seconds |
| H5 | initial-value problems | 15 minutes |

In each case the computing has been performed on an IBM 7094, which is roughly twice as fast as the IBM 7090. Although our discrete-variable algorithm appears to be competitive timewise, it has provided less accuracy.

A more accurate solution has been obtained for the N=20 problem by continuing the computations in double precision with K = 500,000. The starting data is taken from the previous single precision computer run with ETA = .1. After a total of 30 minutes of computing time, (g,g) = .15 and PENLTY = $3.1(10^{-10})$. The results are disappointing in that for this problem the use of double precision has not provided the rapid convergence expected of the conjugate gradients method. The control points from this run are shown in Figure 6-13. They represent only a modest improvement over those shown in Figure 6-11. The trajectory points are in close aggreement with the numerical values obtained by Herzberg. Both sets of values are listed in Table 6-5; for each variable Herzberg's results are shown in the left column. The minimum transfer time is 193.64 days.

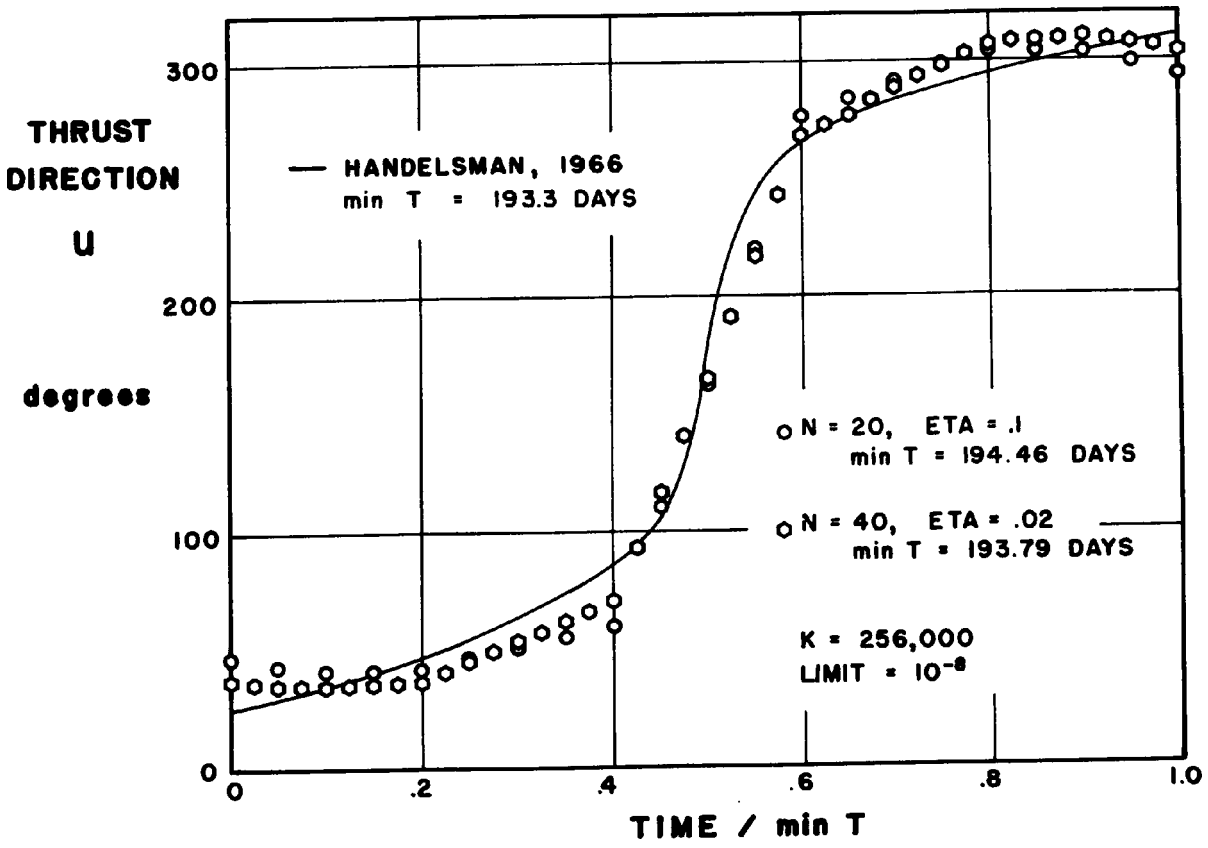



FIGURE 6-11 Minimum-Time Earth-to-Mars Transfer: Optimal Control Time History

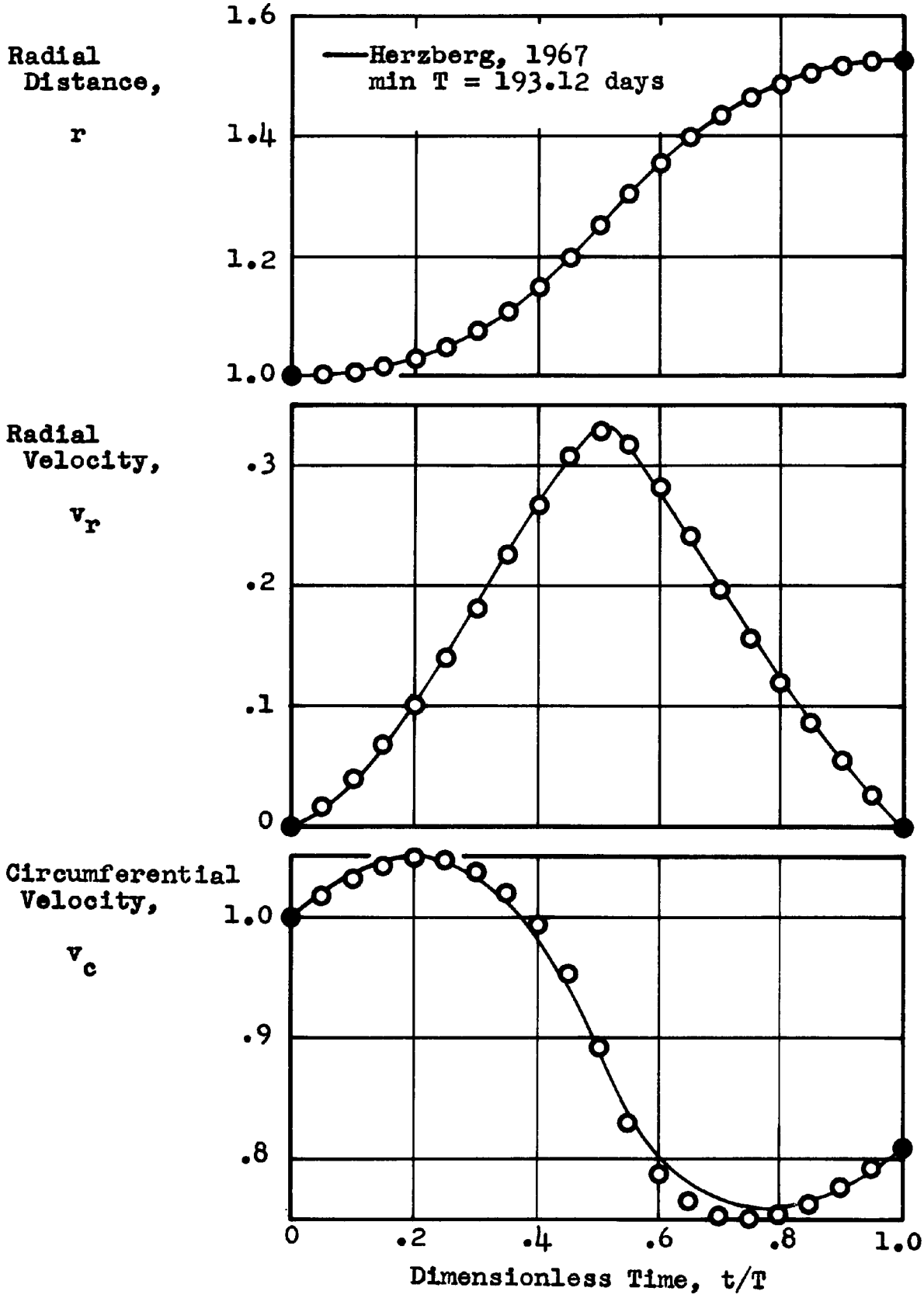



FIGURE 6-12 Minimum-Time Earth-to-Mars Orbital Transfer: Trajectory Time History

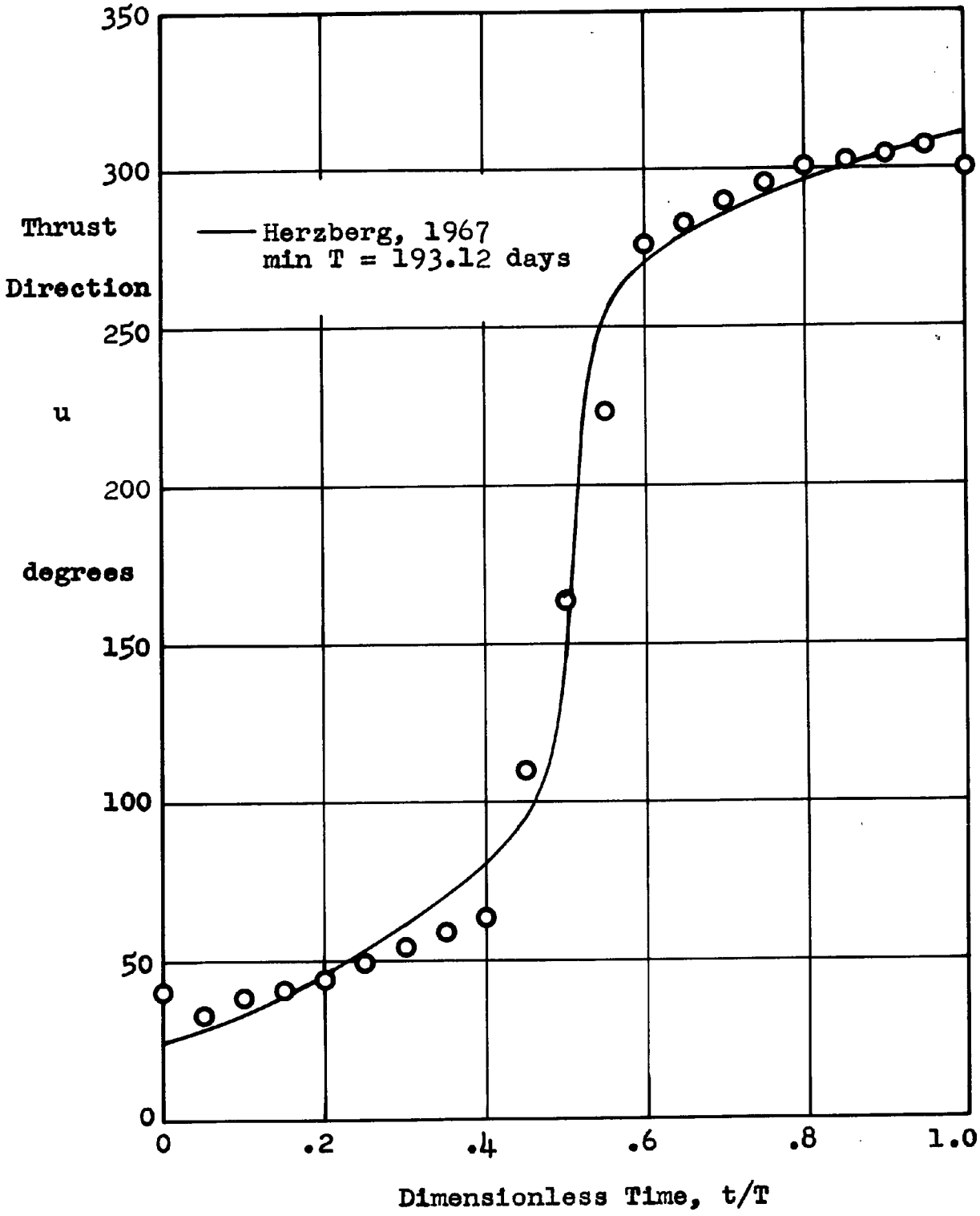



Figure 6-13 Control Time History for the Orbital Transfer Problem: Double Precision Results

| t/T | r | | $v_r$ | | $v_c$ | |
|---|---|---|---|---|---|---|
| .05 | 1.0010 | 1.0013 | .0140 | .0169 | 1.0199 | 1.0177 |
| .10 | 1.0051 | 1.0060 | .0363 | .0398 | 1.0360 | 1.0323 |
| .15 | 1.0135 | 1.0150 | .0661 | .0695 | 1.0467 | 1.0416 |
| .20 | 1.0274 | 1.0294 | .1022 | .1040 | 1.0503 | 1.0449 |
| .25 | 1.0477 | 1.0498 | .1427 | .1424 | 1.0458 | 1.0413 |
| .30 | 1.0749 | 1.0769 | .1855 | .1830 | 1.0327 | 1.0305 |
| .35 | 1.1092 | 1.1108 | .2283 | .2241 | 1.0108 | 1.0127 |
| .40 | 1.1505 | 1.1514 | .2689 | .2643 | .9808 | .9899 |
| .45 | 1.1983 | 1.1987 | .3055 | .3025 | .9427 | .9526 |
| .50 | 1.2515 | 1.2513 | .3330 | .3250 | .8928 | .8945 |
| .55 | 1.3059 | 1.3049 | .3145 | .3143 | .8378 | .8340 |
| .60 | 1.3551 | 1.3544 | .2777 | .2789 | .8043 | .7954 |
| .65 | 1.3979 | 1.3975 | .2384 | .2379 | .7821 | .7764 |
| .70 | 1.4342 | 1.4337 | .1987 | .1976 | .7685 | .7647 |
| .75 | 1.4639 | 1.4633 | .1597 | .1587 | .7619 | .7601 |
| .80 | 1.4873 | 1.4866 | .1223 | .1220 | .7615 | .7619 |
| .85 | 1.5047 | 1.5041 | .0871 | .0877 | .7666 | .7685 |
| .90 | 1.5164 | 1.5160 | .0546 | .0557 | .7766 | .7789 |
| .95 | 1.5230 | 1.5228 | .0253 | .0269 | .7911 | .7934 |
| 1.00 | 1.5250 | 1.5250 | .0000 | .0000 | .8098 | .8098 |

**TABLE 6-5** **State Variables for the Orbital Transfer Problem: A Comparison of Double Precision Results**

## 6.5 An Example Problem with Discontinuous Optimal Control

In Section 2.2.1 we have mentioned that the control points seem to represent a very smooth time function because the difference equations are based on polynomial interpolation. However, there are many flight path optimization problems for which one or more of the control functions are discontinuous. It is of interest then to see how well our discrete-variable algorithm works on such problems.

Consider the regulator problem: drive the vector $x = (x^1, x^2)$ from some specified initial state $(x_o^1, x_o^2)$ to the origin in minimum time subject to $\dot{x}^1 = x^2$, $\dot{x}^2 = u$, and $|u| \leq 1$. The optimal control for this problem is bang-bang and generates the well-known parabolic state space trajectories. In particular if we choose (1,0) as the initial state, the minimum time is 2 units.

The numerical results are much as expected with the control points representing a continuous approximation to the discontinuous optimal control. A 60-variable (N=20) solution has been obtained in 1 minute 6 seconds with ETA = .1 and LIMIT = $10^{-8}$. The minimum time is 2.027. The resulting control points are shown in Figure 6-14. Adherence to the inequality constraint on the control is particularly sharp; the control points for which the constraint is binding are accurate to five significant figures. The corresponding optimal trajectory is presented in Figure 6-15. Here again we obtain a high degree of accuracy except near the switching time. The computations were characterized

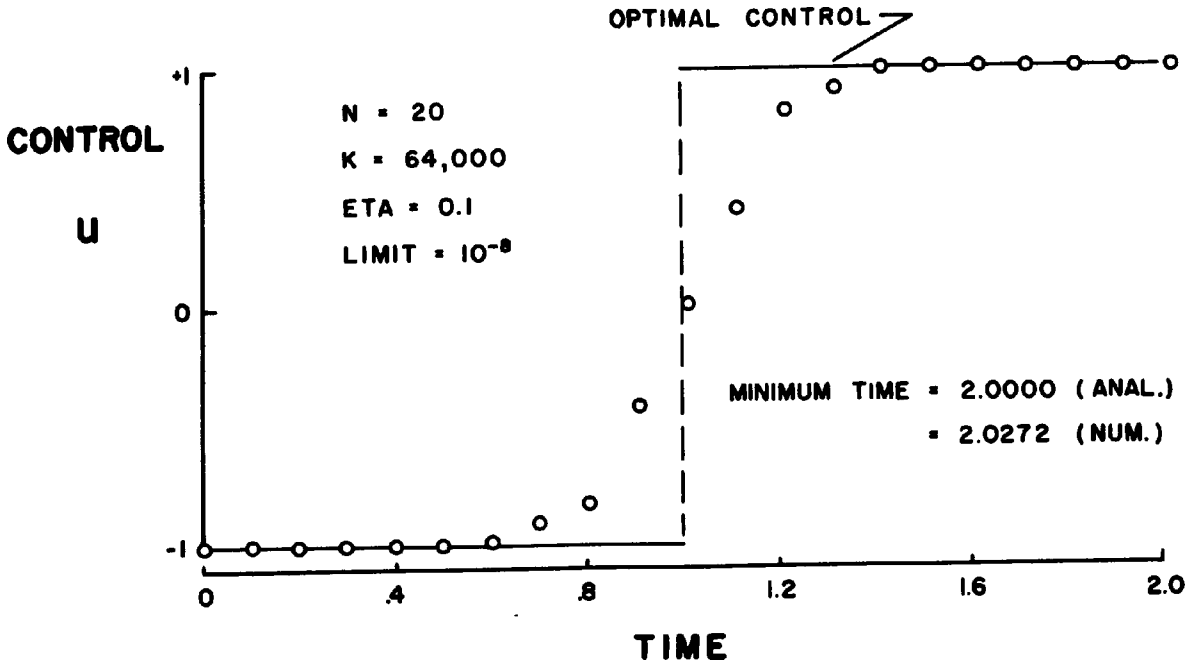



FIGURE 6-14 A Minimum-Time Regulator Problem: Optimal Control Time History

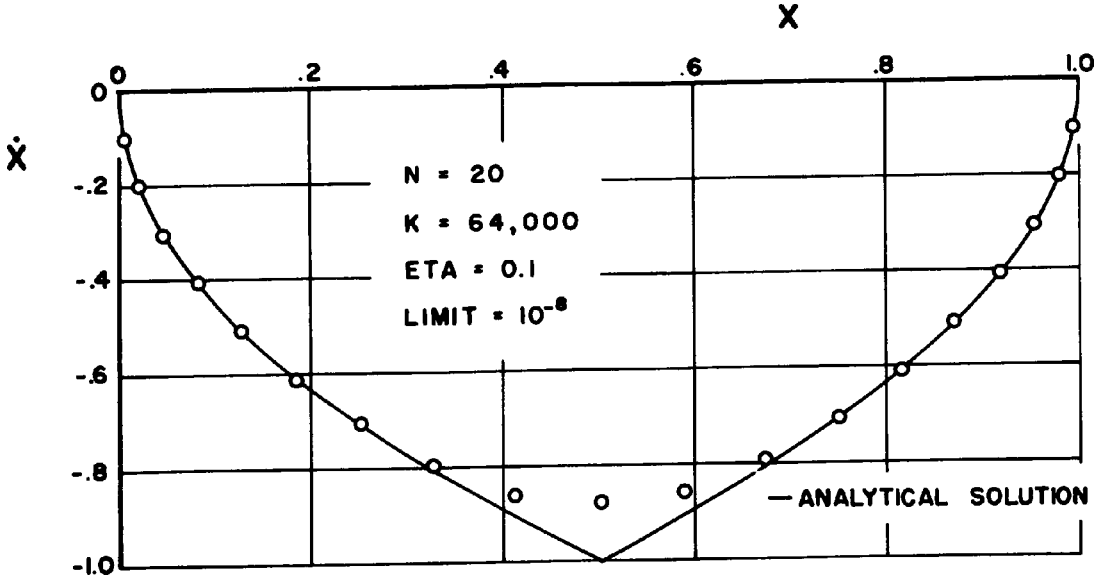



FIGURE 6-15 A Minimum-Time Regulator Problem: Optimal Trajectory

by a large effort after each doubling of the number of time intervals and a relatively small effort after each increase in K.

These numerical results have been obtained without assuming any prior knowledge of the nature of the optimal control. Often, when the necessary conditions of the calculus of variations are used, it is possible to determine the number of discontinuities in the piecewise continuous optimal control even though an analytical solution is not possible. Suppose there are s such discontinuities. Then the time interval $[0,T]$ can be divided into s+1 intervals $[0,T_1], [T_1,T_2], \ldots, [T_{s-1},T_s], [T_s,T]$, each of which is further divided into N equal subintervals. We could then approximate the differential equations by a set of difference equations over each of the s+1 time segments and include the switching times $T_1,T_2,\ldots,T_s$ as additional problem variables. This device has been used in treating the booster problem in the following section.

## 6.6 A Two-Stage Booster Problem

The two-dimensional boost problem treated here is meant to be representative of a moderately-difficult flight path optimization problem. It may be thought of as a more realistic model for the problem presented in Section 6.3. There is no analytical solution available for comparison purposes.

The problem is to find the optimal thrust direction angle $u_*(t)$ which maximizes the burnout speed $v(t_p)$ subject

to the system

$$\dot{h} = v \sin \gamma$$
$$\dot{v} = g_o\left[\frac{T(h)}{W(t)} \cos u - \frac{D(h,v)}{W(t)} - \left(\frac{r}{r+h}\right)^2 \sin \gamma\right]$$
$$\dot{\gamma} = (g_o/v)\left[\frac{T(h)}{W(t)} \sin u - \left(\frac{r}{r+h}\right)^2 \cos \gamma\right] + \frac{v \cos \gamma}{r+h} \quad (6\text{-}9)$$

with specified boundary values

$$h(t_I) = h_I, \quad v(t_I) = v_I, \quad \gamma(t_I) = \pi/2,$$
$$h(t_F) = h_F, \quad \gamma(t_F) = 0.$$

The initial time $t_I$ and final time $t_F$ are both specified. The state variables are the altitude h, the speed v, and the flight path angle $\gamma$ which is measured positively upward from the local horizontal to the velocity vector. The thrust direction angle u is measured from the velocity vector to the thrust vector. Figure 6-16 illustrates some of the geometry. Also, $g_o$ is the gravitational acceleration at the Earth's surface; r is the Earth's radius; T is the rocket thrust force; D is the aerodynamic drag; and W is the "weight," i.e., mass times $g_o$, of the rocket.

Several assumptions are inherent in equations (6-9). We make the usual assumptions of a point-mass rocket and a non-rotating Earth. An inverse-square gravitational force is used. Also, we assume the lift force to be negligibly small.

A two-stage booster configuration will be treated here. The staging time $t_s$ will be specified. The control u(t) may be discontinuous at the staging time as are T and W.

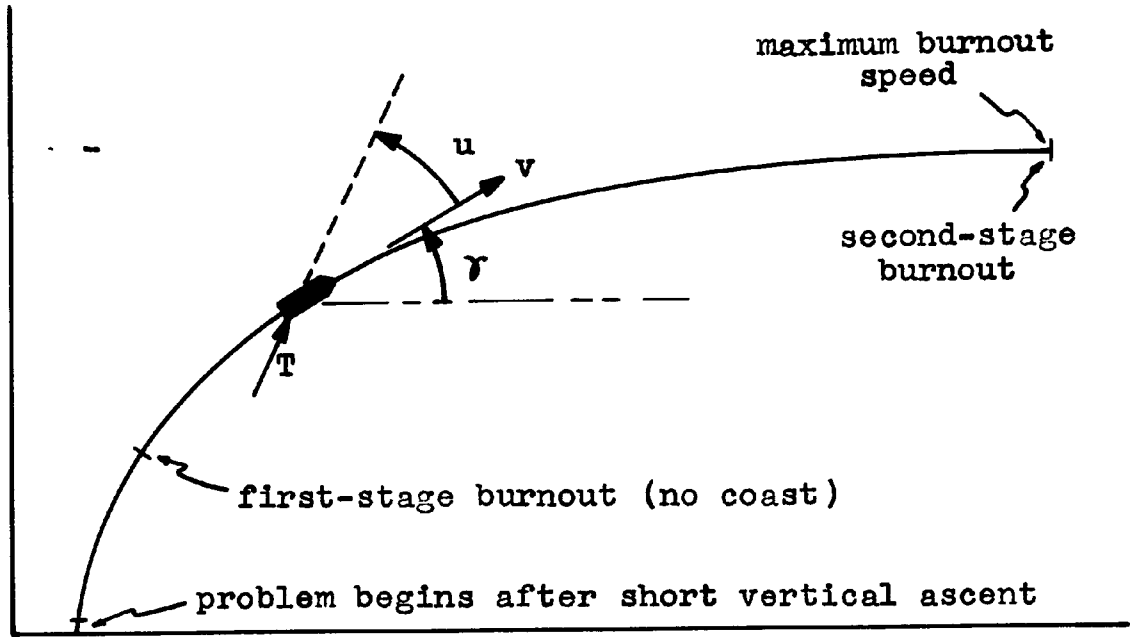



**FIGURE 6-16** Two-Stage Booster Problem: Trajectory Geometry

We arbitrarily exclude any coasting periods. An exponential atmosphere will be assumed for the first-stage operation. Atmospheric effects will, however, be neglected during second-stage operation. Therefore, we will use the expressions

$$T(h) = \begin{cases} T_1 + A_e P_1 (1 - e^{-h/\underline{h}}), & t < t_s \\ T_2 , & t > t_s \end{cases} \qquad (6\text{-}10)$$

for the thrust force and

$$D(h,v) = \begin{cases} \frac{1}{2}\rho_1 e^{-h/\underline{h}} v^2 C_D S, & t < t_s \\ 0 , & t > t_s \end{cases} \qquad (6\text{-}11)$$

for the drag force, $A_e$ being the nozzle exit area, $P_1$ the

standard sea level ambient pressure, $\underline{h}$ the scale height, $\rho_1$ the standard sea level density, $C_D$ the drag coefficient, and S the maximum cross-sectional area of the booster. We will neglect Mach number effects and consider the drag coefficient to be constant. Finally, we assume constant fuel flow rates so that

$$W(t) = \begin{cases} W_1 - \dot{W}_1 t, & t < t_s \\ W_2 - \dot{W}_2(t-t_s), & t > t_s \end{cases} . \qquad (6\text{-}12)$$

Normalized quantities have been used for the actual computations. The final altitude $h_F$ and $h_F/v_c$ where $v_c^2 = g_o r^2/(r+h_F)$ serve as the basic units of length and time, respectively. The forces T, D, and W have not been normalized since only their ratios occur in (6-9).

The booster is to place a given payload in Earth orbit at a burnout altitude (perigee) of 100 nautical miles; thus, $h_F$ = 608,000 ft. In (6-10) and (6-11) we use $P_1$ = 2116 psf, $\rho_1$ = .002378 slug/ft$^3$, and $\underline{h}$ = 23500 ft. The booster data has been taken from (B11) and is representative of a Saturn V class booster:

$A_e$ = 550 sqft
S = 850 sqft
$C_D$ = 0.5
payload weight = 250,000 lb
total launch weight = 5,855,000 lb

| | First Stage | Second Stage |
|---|---|---|
| thrust ($T_1$ and $T_2$) | 7,500,000 lb | 1,000,000 lb |
| jettison weight | 350,000 lb | 95,000 lb |
| fuel flow rate | 28,000 lb/sec | 2,400 lb/sec |
| time at burnout | 150 sec | 550 sec . |

The system (6-9) becomes singular at launch because of the term with v in the denominator. There are two ways to avoid this difficulty: another coordinate frame may be used for a short time subsequent to the launch, or the problem can begin after a short vertical ascent. We have chosen the latter course. The equations of motion for vertical ascent are numerically integrated using a fourth-order Runge-Kutta algorithm. After 5 seconds of flight time the following values for altitude and velocity have been obtained:

$$h_I/h_F = .00019293$$
$$v_I/v_c = .0018682 \quad .$$

These data have been used as initial conditions for the optimization problem.

The difference scheme will consist of two sets of the third-order difference equations (5-5) matched at the staging time. Thus, the time interval used in each set will be different. By matching we mean that only one trajectory point corresponding to $t = t_s$ is used for each state variable; it will serve both as final point for the first thrusting duration and as initial point for the second thrusting duration. Of course, in order to allow for a discontinuity in the control u at the staging time, we provide two control points for $t = t_s$, one for each set of difference equations.

## 6.6.1 The Unconstrained Solution

We began computing in single precision on a problem of 40 variables with ETA = .09 and LIMIT = $10^{-8}$. The following set of intentionally crude non-dimensional data was used to start the calculations.

| t, sec | h | v | $\gamma$ | u | |
|---|---|---|---|---|---|
| 5 | | | | -.02 | |
| 34 | .07 | .06 | 1.3 | -.02 | |
| 63 | .14 | .12 | 1.1 | -.02 | |
| 92 | .21 | .18 | .9 | -.02 | |
| 121 | .28 | .24 | .7 | -.02 | |
| 150 | .35 | .30 | .5 | -.02 | .2 |
| 230 | .50 | .44 | .4 | | .2 |
| 310 | .65 | .58 | .3 | | .2 |
| 390 | .80 | .72 | .2 | | .2 |
| 470 | .95 | .86 | .1 | | .2 |
| 550 | | 1.00 | | | .2 |

The computations were very time consuming. A 40-variable solution was obtained with an expenditure of nearly 4 minutes of computing time. However, after an additional 8 minutes and 1404 steps of computing, a solution for 80 variables had still not been attained. The poor rate of convergence proved to be the result of round-off errors as discussed in Section 5.3. A computer run using double precision was then performed starting from the best single-precision results. At the end of 5 minutes and 162 steps a solution with (g,g) = .034 and PENLTY = $5.4(10^{-10})$ was obtained.

This unconstrained solution is presented in Figures 6-17 and 6-18. The maximum final speed is 24509.9 fps relative to a non-rotating Earth. The altitude and velocity time-histories are shown in Figure 6-17. The most significant

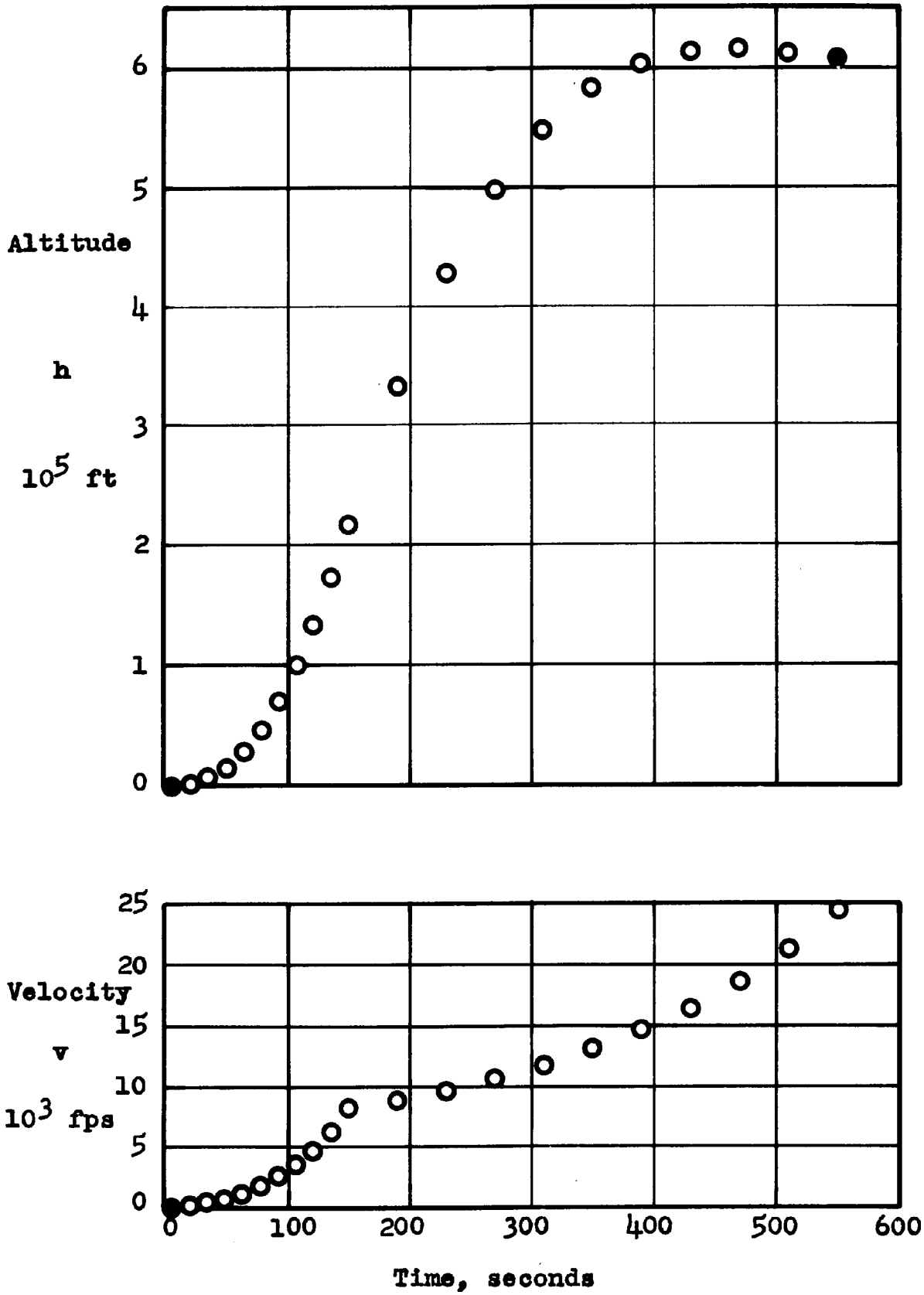



FIGURE 6-17 Two-Stage Booster Problem: Altitude and Velocity Time Histories

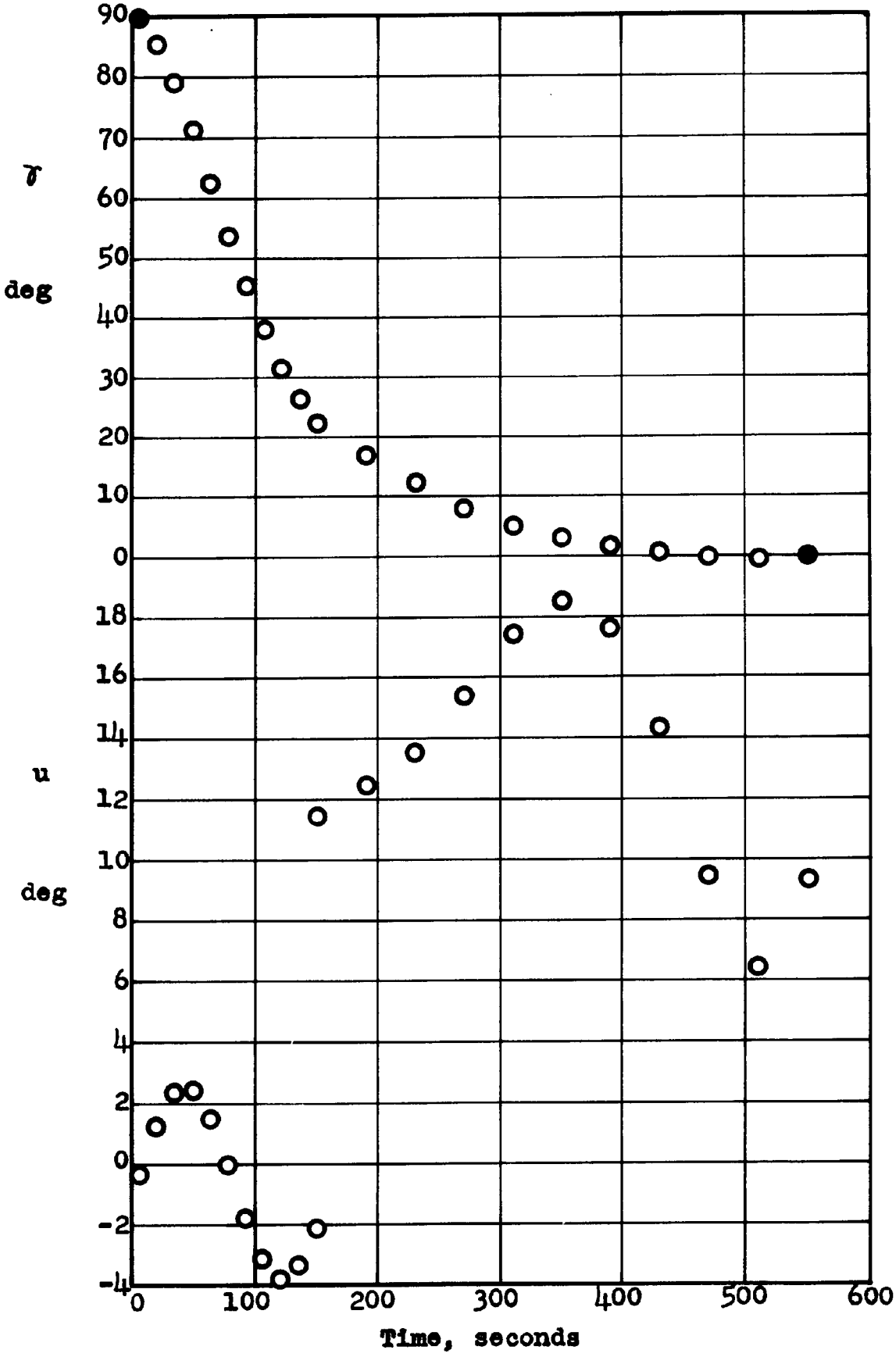



FIGURE 6-18 Boost Problem: $\gamma$ and u Time Histories

feature here is that the booster exceeds the specified final altitude near the end of the second-stage burn. This characteristic has been noted previously; see for example (B11). The flight path angle $\gamma$ and the thrust direction control angle u are plotted versus time in Figure 6-18. The first control point is slightly negative (-.267 degrees) in order to tip the booster off of its vertical flight path. The control then takes on small positive values to counter the side force due to gravity. During the final portion of the first stage burn, u is negative as the booster "turns the corner" and heads downrange. The control angle is positive throughout the second-stage burn; it drops off sharply to allow the booster to convert its excess potential energy into kinetic energy near burnout.

## 6.6.2 A Constraint on the Dynamic Pressure

We will now consider this same boost problem modified by imposing a state space constraint on the trajectory. Specifically, we require that the dynamic pressure

$$q(h,v) = \tfrac{1}{2}\rho_1 e^{-h/h} v^2 \qquad (6\text{-}13)$$

always be less than or equal to some specified maximum value, say QMAX. This inequality constraint can be illustrated graphically with the h-v diagram shown in Figure 6-19. Some of the unconstrained trajectory points of Figure 6-17 are also shown. For a rocket booster of the type considered here, the dynamic pressure exhibits a single sharp maximum. This maximum is usually attained in the altitude range of

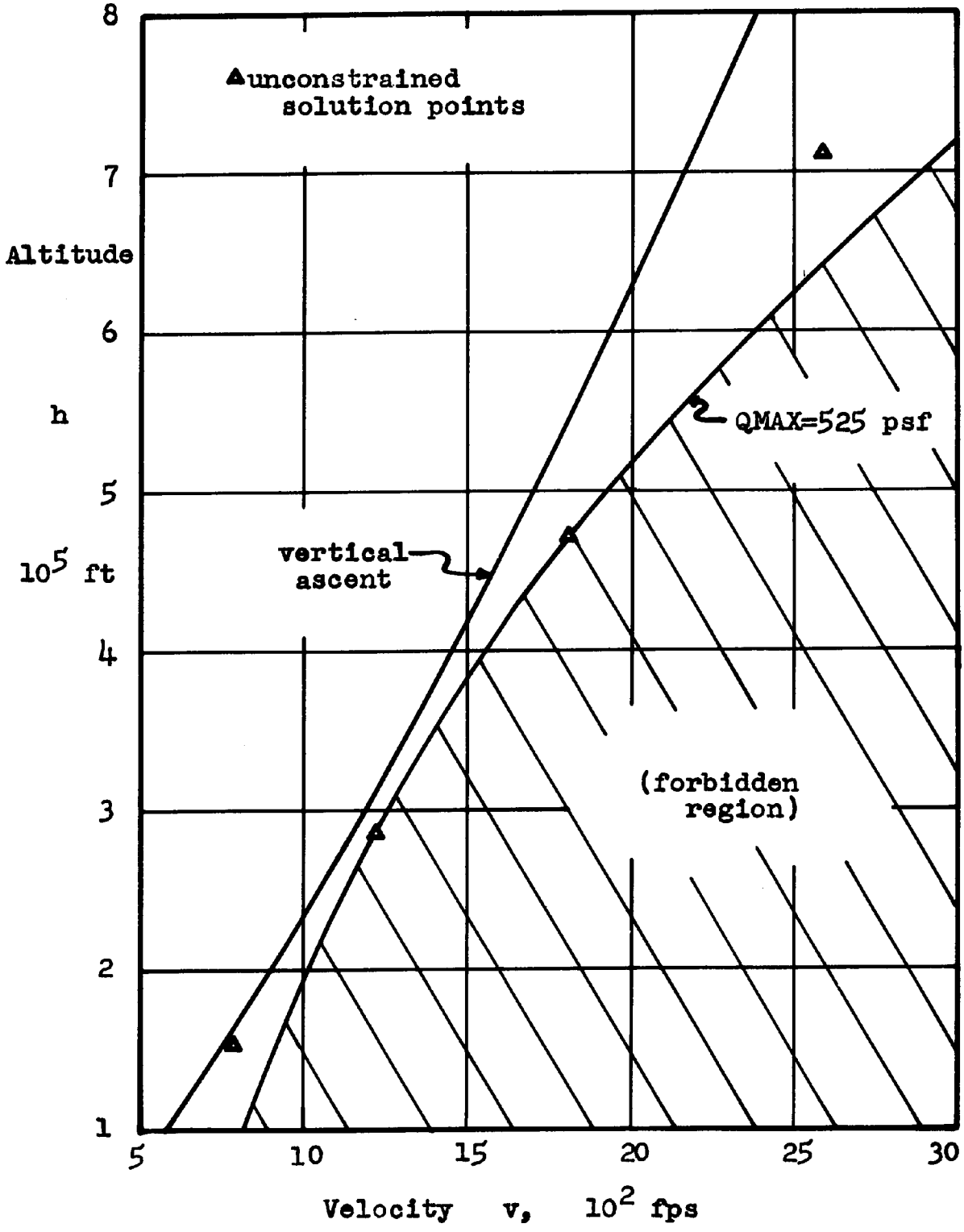



FIGURE 6-19 Dynamic Pressure Constraint: h-v Diagram

30,000 to 40,000 feet.

We anticipate that for the optimal constrained trajectory the dynamic pressure will equal QMAX at only one point. That is, the constraint

$$R^1(h,v) = 1 - q(h,v)/QMAX \geq 0 \qquad (6\text{-}14)$$

will be binding on the optimal trajectory at only one point, although the presence of the constraint (6-14) will, of course, affect the entire trajectory. Such is not the case with the constrained brachistochrone problem of Section 6.1.2 Accordingly, the pointwise enforcement of (6-14) is no longer an attractive approach since a large number of time points would be required to adequately represent the constraint. Furthermore, the constraint would always be satisfied at all but a few of these time points anyway. Instead we choose to replace (6-14) by the following discrete-variable approximation. We consider q as a function of time and fit a parabola to the three q-points $q_{m-1}$, $q_m$, and $q_{m+1}$, where $q_m$ corresponds to the time point $t_m$ and is the largest of all the $q_i$. The resulting parabola has a maximum value of

$$q_{max} = q_m - \frac{(q_{m-1} - q_{m+1})^2}{8(q_{m-1} - 2q_m + q_{m+1})} \quad . \qquad (6\text{-}15)$$

The quantity $\left[\text{CON } (1-q_{max}/QMAX)\right]^4$ is then added to PENLTY whenever $q_{max}$ exceeds QMAX. Note that such violations directly affect six trajectory points: three altitude points and three velocity points. Higher-order polynomials could be used as well, but the parabola is the simplest.

The smallest maximum dynamic pressure will be attained for a vertical flight path. A numerical integration of the vertical ascent equations of motion for the booster considered here provides a maximum dynamic pressure of 469 psf. The unconstrained optimal solution discussed earlier suggests a maximum q of 550 psf; a parabolic fit predicts 545 psf. Accordingly, a value of 525 psf has been selected for the constraint limit QMAX.

The computer results are disappointing. A 14-minute single-precision run has been started from the same initial data as given earlier for the unconstrained problem; a constraint weighting constant CON = 10 is used. A solution (ETA = .09, LIMIT = $10^{-8}$) to the 40-variable problem is obtained in nearly 6 minutes. During the remaining 8 minutes, convergence for the subsequent 80-variable problem is very slow. Computations have been continued from this point in double precision. At first the convergence is much improved, but the improvement is transitory. After three double precision runs totaling 21 minutes and 702 steps (over 8 cycles of the conjugate gradients method), the termination criteria have not yet been met. At this point (g,g) = 23 and PENLTY = $7.8(10^{-8})$. There are no unusual features of the computations other than the slow progress. We can only conclude that the dynamic pressure constraint used here is a particularly difficult one.

It is still instructive to compare the latest constrained results with the unconstrained solution presented earlier. In Figure 6-20 we show the dynamic pressure versus

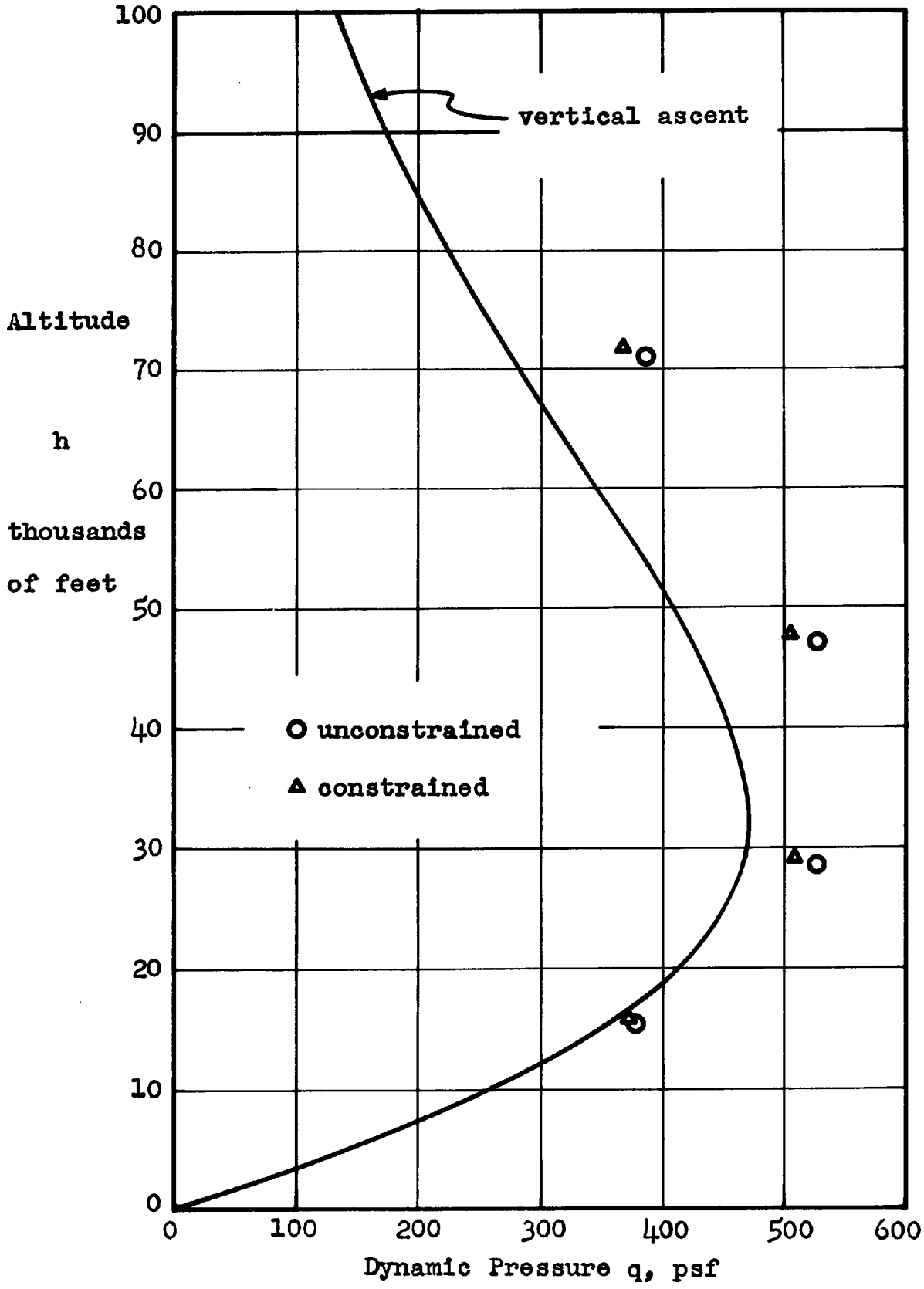



FIGURE 6-20 Boost Problem with Dynamic Pressure Constraint: Dynamic Pressure versus Altitude

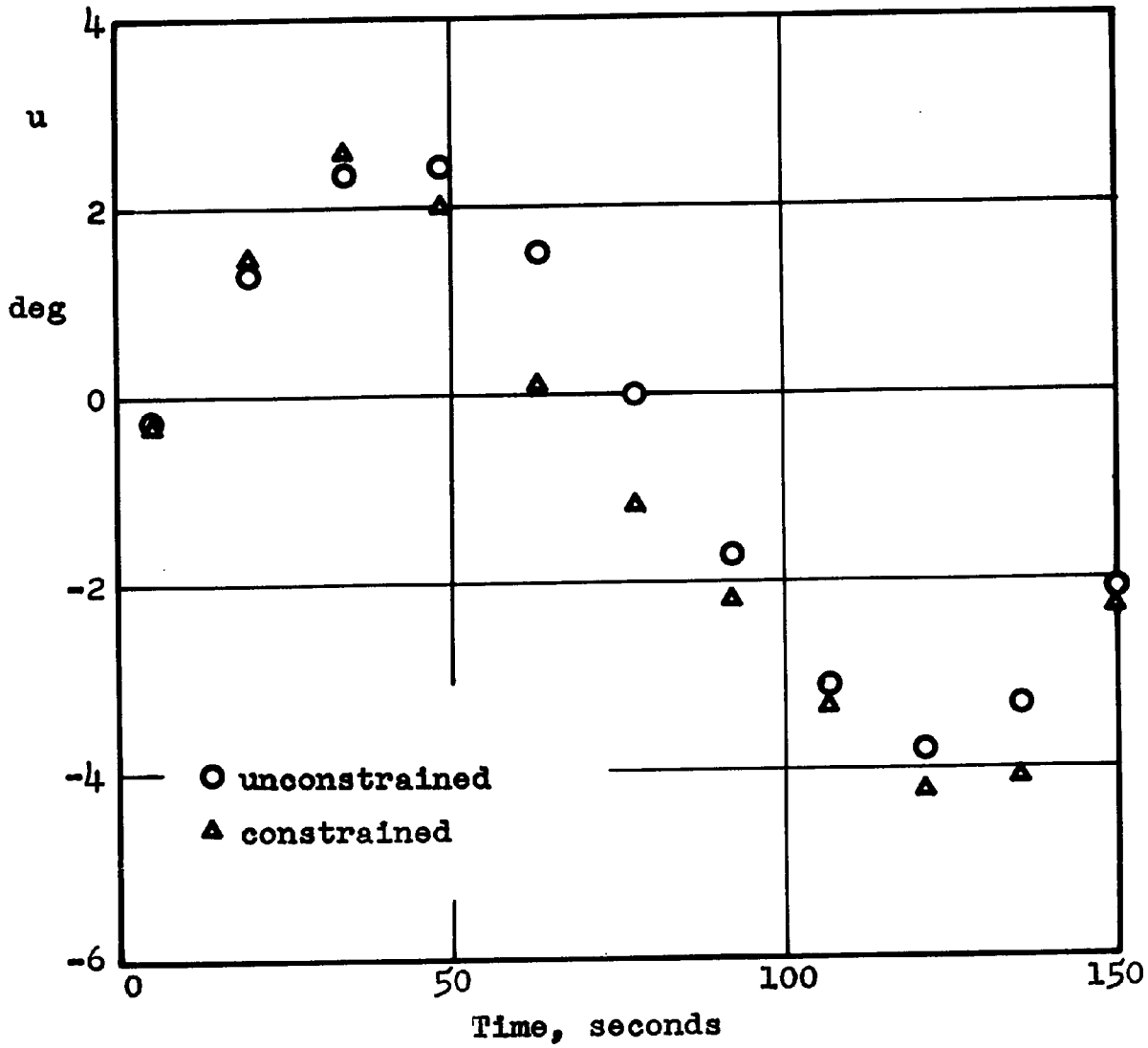



FIGURE 6-21 Boost Problem with Dynamic Pressure Constraint: Comparison of Control Points during First Burn

altitude. The control time histories during first-stage operation are shown in Figure 6-21. We note that after the initial tilt-over the control points become more positive than for the unconstrained case in order to maintain a more nearly vertical flight path. This is followed by more negative control points as the booster is steered downrange. The altitude and velocity points deviate relatively little

| Time seconds | h, ft | | v, fps | |
|---|---|---|---|---|
| | U | C | U | C |
| 19.5 | 2066 | 2147 | 221.4 | 223.8 |
| 34.0 | 6861 | 7048 | 458.1 | 461.1 |
| 48.5 | 15419 | 15745 | 782.7 | 783.1 |
| 63.0 | 28613 | 29142 | 1224.2 | 1217.3 |
| 77.5 | 47085 | 47780 | 1813.8 | 1802.0 |
| 92.0 | 71080 | 71895 | 2583.7 | 2569.2 |
| 106.5 | 100420 | 101401 | 3568.1 | 3551.1 |
| 121.0 | 134675 | 135917 | 4808.7 | 4788.7 |
| 135.5 | 173522 | 175016 | 6369.7 | 6346.5 |
| 150.0 | 217293 | 218839 | 8363.4 | 8338.2 |
| 190.0 | 334158 | 335466 | 8941.3 | 8915.9 |
| 230.0 | 427712 | 428756 | 9709.0 | 9684.6 |
| 270.0 | 498777 | 499543 | 10666.2 | 10640.9 |
| 310.0 | 549332 | 549844 | 11803.6 | 11778.0 |
| 350.0 | 582704 | 583008 | 13125.3 | 13099.1 |
| 390.0 | 602865 | 603024 | 14656.9 | 14629.8 |
| 430.0 | 613340 | 613416 | 16455.7 | 16427.8 |
| 470.0 | 616204 | 616239 | 18609.3 | 18580.9 |
| 510.0 | 612558 | 612569 | 21235.3 | 21206.5 |
| 550.0 | 608000 | 608000 | 24509.9 | 24480.8 |

**TABLE 6-6** **Two-Stage Booster Problem: Altitude and Velocity Points with (C) and without (U) the Dynamic Pressure Constraint**

from their unconstrained counterparts. These values are displayed in Table 6-6. The maximum burnout speed is 24480.8 fps, which is about 29 fps less than the unconstrained value.

### 6.6.3 An Altitude Constraint

Another state space constraint that can be imposed on this boost problem involves placing an upper limit on the altitude. We recall that several altitude points exceeded the specified final altitude near the end of the flight for the unconstrained problem (see Figure 6-17). Use of the inequality constraint

$$1 - h/HMAX \geq 0 , \qquad (6\text{-}16)$$

where HMAX = 608,000 ft, will allow us to determine the effect of this "overshoot" on the maximum final speed.

For our discrete-variable approximation the inequality (6-16) can be enforced on a pointwise basis as has been done for the constrained brachistochrone problem of Section 6.1.2. Only the second thrusting duration need be constrained.

The computations for this constrained problem have been performed in double precision. The starting data have been obtained from a previous unconstrained run, and the termination tolerances ETA = .03 and LIMIT = $10^{-8}$ are used. A constraint weighting constant CON = 50 is used in the altitude constraint penalty function (see Section 5.6). The dynamic pressure constraint (6-14) is no longer present.

After a total of nearly 25 minutes of computing time, a K-min (K = 125,000) has been obtained for the 80-variable problem with (g,g) = .021 and PENLTY = $7.7(10^{-10})$. The maximum final speed of 24499.0 fps is only about 10 fps less than that obtained earlier for the unconstrained problem. The altitude, flight path angle, and control points for the second thrust duration are listed in Table 6-7 under the column headings labeled C. For comparison the unconstrained values from the computer run presented in Section 6.6.1 are shown under the column headings labeled U.

The most significant feature of this problem is that the altitude constraint will apparently be binding on the optimal trajectory only at the instant of second-stage burnout, although the entire trajectory is, of course, affected by the presence of the constraint. In our discrete-variable approximation the altitude point corresponding to $t_{N-1}$ will necessarily violate the constraint since this is the only way the constraint can be felt.

| Time, sec | h, ft | | $\gamma$, degrees | | u, degrees | |
|---|---|---|---|---|---|---|
| | U | C | U | C | U | C |
| 150 | 217293 | 216373 | 22.61 | 22.36 | 11.47 | 11.43 |
| 190 | 334158 | 331900 | 17.09 | 16.83 | 12.50 | 12.44 |
| 230 | 427712 | 423942 | 12.20 | 11.93 | 13.58 | 13.57 |
| 270 | 498777 | 493320 | 8.13 | 7.88 | 15.41 | 15.49 |
| 310 | 549332 | 542078 | 5.02 | 4.79 | 17.45 | 17.66 |
| 350 | 582704 | 573676 | 2.85 | 2.66 | 18.55 | 19.01 |
| 390 | 602865 | 592363 | 1.45 | 1.33 | 17.64 | 18.36 |
| 430 | 613340 | 602166 | .57 | .56 | 14.34 | 15.64 |
| 470 | 616204 | 606488 | -.05 | .21 | 9.53 | 12.92 |
| 510 | 612558 | 608022 | -.42 | .02 | 6.45 | 4.05 |
| 550 | 608000 | 608000 | .00 | .00 | 9.38 | 7.63 |

**TABLE 6-7** Comparison: Two-Stage Booster Problem with (C) and without (U) the Altitude Constraint

# CHAPTER 7

## SUMMARY

All but the simplest of optimal control problems require the use of a large-scale digital computer for their solution. Thus all iterative solution procedures ultimately deal with a discrete rather than a continuous problem. The aim of this dissertation has been to proceed directly to a discrete model of the continuous optimal control problem and then to treat the resulting discrete-variable optimization problem. As we noted in Chapter 1, this approach has received very little attention in the literature.

The value of the method of conjugate gradients in efficiently minimizing a function of many variables must be emphasized. This method offers quadratic terminal convergence and yet only requires the calculation of first partial derivatives. It is doubtful that the approach presented here would be feasible if one had to rely on the slow convergence of direct search or steepest descent methods for the unconstrained minimizations.

Although the discrete-variable method used here has several inherent advantages, it is not without its shortcomings. Perhaps the most obvious of these is that the timewise discretization can result in a very large number of variables. We cannot hope to compete in terms of computer running time with other methods which typically divide the total time interval into 100 or more parts in the course of numerical integration. However, as the numerical results

of Chapter 6 illustrate, 20 or 40 time subintervals will often provide an adequate description of the state and control variable time histories.

A more serious concern is the dilemma that as we succeed in reducing the size of the residuals $\mathbf{F}^i$ the accuracy retained in the gradient vector diminishes. A poor determination of the gradient vector in turn frustrates the fast terminal convergence of the conjugate gradients method. The situation can be considerably improved by performing the computations in double precision. Since the basic difficulty lies in the nature of the residuals themselves, the use of double precision does not provide a cure; it only postpones the problem, usually long enough to allow a solution of the desired accuracy to be obtained. The improvement in the rate of convergence using double precision is purchased at the expense of significantly higher running times per iteration. However, for most descriptive languages the conversion from single to double precision is accomplished with very little additional programming effort.

With any numerical technique it is highly desirable to have available a bound for the error present in the approximate solution. To be more specific, suppose that the discrete-variable optimization problem has been solved without error; then what error, either pointwise or in the sense of an appropriate vector norm, exists between this solution and a hypothetical solution that matches at each time point $t_i$ the solution to the continuous optimal control problem? In other words, what have we lost in approximating the

continuous problem with a discrete model? Unfortunately, this author has been unable to find any such error analyses in the literature or any other means of approaching this question that promises success. In the absence of any useful error bounds we must rely on the results of two or more approximate solutions with different N to provide an indication of the order of magnitude of the error. Of course, one also gains a feel for the accuracy that can be expected by applying the numerical technique to problems with known analytical solutions as was done in Chapter 6.

In spite of the large number of variables present, the difficulty regarding round-off errors, and the lack of any useful error estimates, discrete-variable methods still offer several intrinsic advantages and deserve further attention. One advantage is the ease of starting the computations. The problems treated thus far indicate that the method is quite insensitive to the initial guesses for the problem variables. For that matter, our algorithm could even be used to quickly provide nominal or starting trajectories for other iterative methods. For problems with inequality constraints there is no question of first obtaining feasible points, i.e., points that satisfy the constraints, since the penalty functions used here allow constraint violations in the course of the computations.

Also, the discrete-variable methods do not face the two-point boundary value problem encountered by the indirect methods. In fact the use of difference equations allows the specified boundary values to be satisfied exactly

throughout the calculations. Since the network of difference equations provides a means of dealing with all trajectory and control points simultaneously, we are able to avoid the stability difficulties encountered by methods that treat the two-point boundary value problem as a series of initial-value problems. A simple example of this was given in Section 6.2.

Although objective measures of programming effort are lacking, the simplicity of the algorithm presented here makes it relatively easy to program for machine computations. This factor cannot be overlooked, for as larger and faster computers come into service a larger share of the total cost of computing will be taken up by the cost of programming.

Probably the most important asset of the discrete-variable methods is that they may be applied to such a wide variety of problems. For example, variable-time problems are handled by simply including the final time as an additional problem variable. Many flight path optimization problems involve control parameters such as stage separation times, coast periods, and propellant weights in addition to the control functions. These parameters are, of course, already discrete variables and as such may also be included as additional problem variables. As has been pointed out before, inequality constraints may be adjoined to the cost function as penalty functions; in the discrete-variable approach all variables can be treated as equals so that in principle state space constraints are no more difficult than constraints involving the control functions.

There is, of course, considerable room for additional experimentation with and improvements to the discrete-variable approach described in the previous chapters. For example, other termination criteria could be tried. Perhaps a more efficient linear minimization scheme could be devised. For more difficult problems it may even prove advantageous to abandon gradient methods altogether and use instead a minimization procedure based on function values alone. On the theoretical side, discretization error bounds on estimates have yet to be found, and a study of convergence properties that takes into account the use of finite-precision arithmetic would be helpful.

Nevertheless, the method presented here offers a viable approach to a wide class of problems. This is not to say that the method is uniformly better or worse than existing computational schemes. However, the timewise discretization at the outset provides a different perspective from which to view the computational process. The ease of starting computations, the avoidance of certain stability difficulties through the use of difference equations, the ability to optimize control parameters in addition to control functions, and the capability of handling state space constraints are all characteristics which make this method useful in treating realistic flight path optimization problems.

## Bibliography


A1 Alford, M.W. and Lear, C.W., "A Computational Method for the Optimization of Multistage Ballistic Systems," AIAA J. 4, 1668-1675 (1966).

A2 Antosiewicz, H.A. and Rheinboldt, W.C., "Numerical Analysis and Functional Analysis," Survey of Numerical Analysis, edited by J. Todd (McGraw-Hill, New York, 1962), pp 501-510.

B1 Bliss, G.A., Lectures on the Calculus of Variations (University of Chicago Press, Chicago, Illinois, 1946).

B2 Berkovitz, L.D., "Variational Methods in Problems of Control and Programming," J. Math. Anal. Appl. 3, 145-169 (1961).

B3 Bryson, A.E. and Ross, S.E., "Optimum Rocket Trajectories with Aerodynamic Drag," Jet Propulsion 28, 465-469 (1958).

B4 Breakwell, J.V., "The Optimization of Trajectories," J. Soc. Indust. Appl. Math. 7, 215-247 (1959).

B5 Bellman, R., Dynamic Programming (Princeton University Press, Princeton, New Jersey, 1957).

B6 Bryson, A.E. and Denham, W.F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," J. Appl. Mech. 29, 247-257 (1962).

B7 Breakwell, J.V., Speyer, J.L., and Bryson, A.E., "Optimization and Control of Nonlinear Systems Using the Second Variation," J. SIAM Control 1, 193-223 (1963).

B8 Butler, T. and Martin, A.V., "On a Method of Courant for Minimizing Functionals," J. Math. and Phys. 41, 291-299 (1962).

B9 Beckman, F.S., "The Solution of Linear Equations by the Conjugate Gradient Method," Mathematical Methods for Digital Computers, edited by A. Ralston and H.S. Wilf (John Wiley and Sons, Inc., New York, 1960), pp 62-72.

B10 Bryson, A.E., Denham, W.F., and Dreyfus, S.E., "Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for Extremal Solutions," AIAA J. 1, 2544-2550 (1963).

B11 Battle, C.T. and Gottlieb, R.G., "Optimization of Thrust Direction Histories and Vehicle Parameters for Three-Dimensional Ascent Trajectories," AIAA Paper 64-663, presented at the AIAA/ION Astrodynamics, Guidance, and Control Conference, Los Angeles, California (August 1964).

C1 Collatz, L., The Numerical Treatment of Differential Equations, third edition (Springer-Verlag, Berlin, 1960).

C2 Courant, R., "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," Bull. Amer. Math. Soc. 49, 1-23 (1943).

D1 Denham, W.F. and Bryson, A.E., "Optimal Programming Problems with Inequality Constraints II: Solution by Steepest-Ascent," AIAA J. 2, 25-34 (1964).

D2 Davidon, W.C., "Variable Metric Method for Minimization," Argonne National Laboratory, ANL-5990 (Rev. 2), Argonne, Illinois (February 1966).

D3 Dreyfus, S.E., "Variational Problems with State Variable Inequality Constraints," The RAND Corporation, P-2605-1, Santa Monica, California (August 1963).

F1 Fiacco, A.V. and McCormick, G.P., "The Sequential Unconstrained Minimization Technique for Convex Programming with Equality Constraints," Research Analysis Corporation, RAC-TP-155, McLean, Virginia (April 1965).

F2 _____ and _____, "The Sequential Unconstrained Minimization Technique for Nonlinear Programming, A Primal-Dual Method," Management Sci. 10, 360-366 (1964).

F3 _____ and _____, "Computational Algorithm for the Sequential Unconstrained Minimization Technique for Nonlinear Programming," Management Sci. 10, 601-617 (1964).

F4 Fletcher, R., "Function Minimization Without Evaluating Derivatives--A Review," Comput. J. 8, 33-41 (1965).

F5 Fletcher, R. and Powell, M.J.D., "A Rapidly Convergent Descent Method for Minimization," Comput. J. 6, 163-168 (1963).

F6 Fletcher, R. and Reeves, C.M., "Function Minimization by Conjugate Gradients," Comput. J. 7, 149-154 (1964).

G1 Ghonaimy, M.A.R. and Bernholtz, B., "On a Direct Method of Optimization: Linear Control Systems; On a Direct Method of Optimization: Nonlinear Control Systems," University of Toronto, Toronto, Canada (April 1965); also presented at the IFAC Congress on Automatic Control, London, England (June 1966).

H1 Hestenes, M.R. and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems," J. Res. Natl. Bur. of Standards 49, 409-436 (1952).

H2 Hestenes, M.R., "The Conjugate-Gradient Methods for Solving Linear Systems," Proceedings of the Symposia in Applied Mathematics, Vol. 6 (McGraw-Hill, New York, 1956), pp 83-102.

H3 Handelsman, M., private communication (October 1966).

H4 Herzberg, A.F., private communication (May 1967).

H5 Handelsman, M., "Optimal Free-Space Fixed-Thrust Trajectories Using Impulsive Trajectories as Starting Iteratives," AIAA J. 4, 1977-1082 (1966).

J1 Jazwinski, A.H., "Optimal Trajectories and Linear Control of Nonlinear Systems," AIAA J. 2, 1371-1379 (1964).

K1 Kulakowski, L.J. and Stancil, R.T., "Rocket Boost Trajectories for Maximum Burnout Velocity," ARS J. 30, 612-619 (1960).

K2 Kenneth, P. and McGill, R., "Two-Point Boundary Value Techniques," Advances in Control Systems: Theory and Applications, Vol. 3, edited by C.T. Leondes (Academic Press Inc., New York, 1966), Chapter 2.

K3 Kelley, H.J., "Gradient Theory of Optimal Flight Paths," ARS J. 30, 947-954 (1960).

K4 Kelley, H.J., Kopp, R.E., and Moyer, H.G., "A Trajectory Optimization Technique Based upon the Theory of the Second Variation," Progress in Astronautics and Aeronautics, Vol. 14, Celestial Mechanics and Astrodynamics, edited by V.G. Szebehely (Academic Press Inc., New York, 1964), pp 559-582.

K5 Kelley H.J., "Method of Gradients," Optimization Techniques, edited by G. Leitmann (Academic Press Inc., New York, 1962), Chapter 6.

K6 Kelley, H.J., Denham, W.F., Johnson, I.L., and Wheatley, P.O., "An Accelerated Gradient Method for Parameter Optimization with Nonlinear Constraints," J. Astronaut. Sci. 13, 166-169 (1966).

K7 Krylov, V.I., Approximate Calculation of Integrals, translated by A.H. Stroud (The Macmillan Company, New York, 1962), pp 206-216.

K8 Kelley, H.J., Kopp, R.E., and Moyer, H.G., "Successive Approximation Techniques for Trajectory Optimization," 1961 IAS Vehicle Systems Optimization Symposium (Institute of Aerospace Sciences, New York, 1961), pp 10-25.

L1 Leon, A., "An Annotated Bibliography on Optimization," Mental Health Research Institute, University of Michigan, Preprint 162, Ann Arbor, Michigan (July 1965).

M1 McGill, R. and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator," AIAA J. 2, 1761-1766 (1964).

M2 McGill, R., "Optimal Control, Inequality State Constraints, and the Generalized Newton-Raphson Algorithm," J. SIAM Control 3, 291-298 (1965).

M3 McReynolds, S.R. and Bryson, A.E., "A Successive Sweep Method for Solving Optimal Programming Problems," Proc. 1965 Joint Automatic Control Conference, 551-555 (1965).

O1 Overholt, K.J., "An Instability in the Fibonacci and Golden Section Search Methods," Nordisk Tidskrift for Informations-Behandling (BIT) 5, 284-286 (1965).

P1 Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., and Mishchenko, E.F., The Mathematical Theory of Optimal Processes (John Wiley and Sons, Inc., New York, 1962).

P2 Pomentale, T., "A New Method for Solving Conditioned Maxima Problems," J. Math. Anal. Appl. 10, 216-220 (1965).

R1 Rosen, J.B., "Iterative Solution of Nonlinear Optimal Control Problems," J. SIAM Control 4, 223-244 (1966).

R2 _____, "The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear Constraints," J. Soc. Indust. Appl. Math. 9, 514-532 (1961).

R3 Russell, D.L., "Penalty Functions and Bounded Phase Coordinate Control," J. SIAM Control 2, 409-422 (1965).

S1 Saltzer, C. and Fetheroff, C.W., "A Direct Variational Method for the Calculation of Optimum Thrust Programs for Power-Limited Interplanetary Flight," Astronaut. Acta 7, 8-20 (1961).

S2 Spang, H.A., "A Review of Minimization Techniques for Nonlinear Functions," SIAM Review 4, 343-365 (1962).

W1 Wilde, D.J., Optimum Seeking Methods (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1964), pp 24-35.

V1 Van Dine, C.P., Fimple, W.R., and Edelbaum, T.N., "Application of a Finite-Difference Newton-Raphson Algorithm to Problems of Low-Thrust Trajectory Optimization," Progress in Astronautics and Aeronautics, Vol. 17, Methods in Astrodynamics and Celestial Mechanics, edited by R.L. Duncombe and V.G. Szebehely (Academic Press Inc., New York, 1966), pp 377-400.