NASA TECHNICAL MEMORANDUM

NASA TM X-58088
March 1972

AN INDIRECT OPTIMIZATION METHOD WITH IMPROVED CONVERGENCE CHARACTERISTICS

A Dissertation Presented to the Faculty
of the Cullen College of Engineering,
University of Houston, in Partial
Fulfillment of the Requirements for
the Degree Doctor of Philosophy

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MANNED SPACECRAFT CENTER

HOUSTON, TEXAS 77058

NASA TM X-58088

AN INDIRECT OPTIMIZATION METHOD WITH IMPROVED CONVERGENCE CHARACTERISTICS

Harold Hughes Doiron
Manned Spacecraft Center
Houston, Texas 77058

AN INDIRECT OPTIMIZATION METHOD WITH IMPROVED CONVERGENCE CHARACTERISTICS

A Dissertation

Presented to

the Faculty of the Graduate School

The University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

Harold Hughes Doiron

May 1970

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The purpose of this thesis is the presentation of an improved method for obtaining numerical solutions of a certain class of two-point boundary value problems which often arise in optimal control theory. These problems are characterized by systems of nonlinear ordinary differential equations with nonlinear boundary conditions.

A general problem in optimal control theory is often stated in the following manner. Given a system which is described by a set of nonlinear ordinary differential equations

$$\dot{x} = f(x,u,t) \tag{1.1}$$

where x is an n vector describing the state (position and velocity) of the system as a function of time t, u is a q vector of time varying controls which can be applied to the system, and f is an n vector of nonlinear functions; it is required to determine the control histories $u(t)$, so that an extremum (maximum or minimum) of some scalar performance index $\phi(x(t_f),t_f)$ is obtained at some terminal time t_f. The system must satisfy certain initial conditions in the form of an m vector of functions

$$n\left(x\left(t_o\right),t_o\right) = 0 \tag{1.2}$$

and the controls must not only extremize the performance index, but also yield a final system state which satisfies a k vector of nonlinear functions

$$\psi\left(x\left(t_f\right),t_f\right) = 0 \qquad k \leq n \tag{1.3}$$

The problem as stated is known as a Mayer problem in the calculus of variations. Simple transformations are given by Bliss [1] which transform the Lagrange problem (where the performance index is a definite integral) and the more general Bolza problem into Mayer problems. With these transformations, a large number of optimal control problems can be stated in the convenient Mayer form.

Since the control problem stated is rarely amenable to analytic solution, methods for obtaining numerical solutions are necessary. The availability of large digital computers coupled with a demand for solutions in various space age applications has resulted in considerable research activity in the solution of the optimal control problem by numerical methods. The majority of this work has occurred in the past ten years and methods of solution are basically divided into either direct or indirect methods.

The direct methods are so-called because they seek to directly manipulate the control histories $u(t)$ in order that an augmented functional (which includes the given performance index and a measure of terminal constraint satisfaction) is extremized. The direct methods are not investigated in this thesis, but because of their importance, a brief reference to some of these methods is made. The most popular direct methods are the gradient or steepest ascent methods developed independently by Kelley [2], and Bryson, Denham, Carroll and Mikami [3,4]. Many extensions to the basic method have been made and are discussed in the recent text by Sage [5]. Significantly different direct approaches are Bellman's dynamic programing [6,7], the conjugate gradient method discussed by Lasdon, Mitter, and Waren [8]; and the optimal sweep method introduced by McReynolds and Bryson [9].

Indirect methods are those in which the control histories are not directly manipulated. Instead, the control problem is transformed into a two-point boundary value problem by deriving certain ordinary differential equations and boundary conditions which must be satisfied for mathematical optimality. The governing differential equations and boundary conditions are obtained from either the necessary conditions of the classical calculus of variations (see for example, Bliss [1] or Sage [5]), the Pontryagin maximum principle [10], or the theory of dynamic programing as discussed by Dreyfus [11]. The various successive approximation techniques for solving the resulting nonlinear two-point boundary value problem are called indirect optimization methods. There are several generally applicable methods which have been developed in recent years. A discussion of these methods is deferred until Chapter II, where the general boundary value problem to be considered in this thesis is presented.

A common problem with existing indirect optimization methods is that initial approximations to either the solution or initial conditions of the boundary value problem are required. The success in solving the problem is sometimes extremely sensitive to the accuracy of the initial approximations. Convergence, when it occurs, is generally more rapid for indirect methods than for direct methods. Ideally, one seeks a method which converges rapidly to a solution from an arbitrary initial approximation. One of the major reasons for this investigation is to improve upon the performance of existing indirect methods with regard to this ideal characteristic. To this end, several new innovations and improvements to known techniques are combined in a unified approach. This includes the introduction of a power series integration method which exhibits several characteristics uniquely suited for determining numerical solutions of nonlinear two-point boundary value problems. Moreover, a new approach is presented for solving problems where the terminal boundary conditions are general functions of the final state and unknown final time. The computational algorithm is derived such that the differential equations to be integrated have improved numerical stability. Consequently, numerical difficulties due to ill conditioned matrices of boundary values can be avoided.

The indirect optimization method presented here is applied to the solution of a minimum time, planar, Earth-Mars transfer problem for a constant low-thrust rocket. The problem was chosen because it has been used to test many other methods and thus, a direct comparison with previous results could be made. The optimization method is also applied to a similar problem where a minimum time rendezvous with a moving target (the planet Mars) is required.

CHAPTER II

THE INDIRECT APPROACH TO TRAJECTORY OPTIMIZATION

In order to apply an indirect trajectory optimization method, it is necessary to formulate the optimal control problem as a two-point boundary value problem. This is accomplished by deriving certain ordinary differential equations with accompanying boundary conditions which must be satisfied. In this chapter, necessary conditions from the calculus of variations are used to formulate the control problem as a two-point boundary value problem and previous numerical methods for solving such problems are discussed.

The control problem which will be considered in this thesis can be stated: a system is described by a set of nonlinear ordinary differential equations

$$\dot{x}(t) = f(x(t),u(t),t) \tag{2.1}$$

where x is an n vector and u is a q vector, $q \leq n$. The initial state is specified at some initial time t_o

$$x\left(t_o\right) = x_o$$

and the terminal state $x(t_f)$ and time t_f are given implicity by the following k vector of functions

$$\psi\left(x\left(t_f\right),t_f\right) = 0 \qquad (2.2)$$

It is necessary to determine the q vector of controls $u(t)$ from some admissible set of controls so that the performance index $\phi(x(t_f),t_f)$ is minimized. The admissible set of controls will be taken to be the set of all piecewise continuous functions on the interval $[t_o,t_f]$.

An augmented functional is formed by adjoining to the given performance index, using Lagrange multipliers, the contraints (2.1) and (2.2) to obtain

$$J = \phi\left(x\left(t_f\right),t_f\right) + \nu^T \psi\left(x\left(t_f\right),t_f\right) + \int_{t_o}^{t_f} \lambda^T(t)(f - \dot{x})\, dt \qquad (2.3)$$

where ν is a k vector of constant Lagrange multipliers, and $\lambda(t)$ is an n vector of time dependent Lagrange multipliers. The problem can now be viewed as one of seeking a minimum of the performance index J subject to no additional constraints. A necessary condition for J to have an extremum is that the first variation of J vanish.

It is helpful when considering variations of J to introduce the scalar function called the Hamiltonian

$$H(\lambda,x,u,t) = \lambda^T f(x,u,t)$$

such that by definition

$$\dot{x} = f(x,u,t) = \left(\frac{\partial H}{\partial \lambda}\right)^T$$

Using the Hamiltonian and an integration by parts, the performance index is written

$$J = \phi\left(x\left(t_f\right),t_f\right) + \nu^T \psi\left(x\left(t_f\right),t_f\right)$$

$$- \lambda^T(t)x(t)\Big|_{t_o}^{t_f} + \int_{t_o}^{t_f} \left\{H + \dot{\lambda}^T x\right\}dt$$

The requirement that the first variation of J vanish along an optimal trajectory with final time not specified results in the following necessary conditions.

$$\dot{x} = \left(\frac{\partial H}{\partial \lambda}\right)^T \tag{2.4}$$

$$\dot{\lambda} = -\left(\frac{\partial H}{\partial x}\right)^T \tag{2.5}$$

$$\frac{\partial H}{\partial u} = 0 \tag{2.6}$$

$$\left(\frac{\partial\phi}{\partial x} + \nu^T \frac{\partial\psi}{\partial x} - \lambda^T\right)\Bigg|_{t_f} = 0 \qquad (2.7)$$

$$\psi\left(x\left(t_f\right), t_f\right) = 0 \qquad (2.8)$$

$$\left(H + \frac{\partial\phi}{\partial t} + \nu^T \frac{\partial\psi}{\partial t}\right)\Bigg|_{t_f} = 0 \qquad (2.9)$$

The derivation of these equations is adequately treated in several texts and the reader is referred in particular to Bliss [1] for classical problem formulations or to the recent book by Sage [5] for a treatment in the more modern control notation. Equations (2.4) and (2.5) represent 2n simultaneous ordinary differential equations in the 2n + q variables of the vectors λ, x, and u. Equation (2.6) provides q conditions by which the q variables of the control vector u can be eliminated from equations (2.4) and (2.5) so that 2n differential equations in 2n dependent variables are obtained. It is assumed that this is possible, since, in general, it is not always possible to solve explicitly for each of the control variables via equation (2.6).

The conditions (2.4) to (2.9) are merely necessary conditions for an extremum of J. A further necessary condition for J to be minimized is given by the non-negativity of the Weierstrass E-Function

$$E = F(t,x,\dot{X},U,\lambda) - F(t,x,\dot{x},u,\lambda) - \frac{\partial F}{\partial \dot{x}}(t,x,\dot{x},u,\lambda)(\dot{X} - x)$$

$$- \frac{\partial F}{\partial u}(t,x,\dot{x},u,\lambda)(U - u) \geq 0$$

where $F = \lambda^T(f(x,u,t) - \dot{x})$ and $\dot{X}$ and U are nonoptimal but permissible values for $\dot{x}(t)$ and $u(t)$. This condition is normally applied in conjunction with equation (2.6). For the example problems which will be presented, this condition is used to resolve an ambiguity in sign resulting from the application of equation (2.6) (see Appendix A).

Equations (2.7), (2.8), and (2.9) yield n, k, and one algebraic equations, respectively, which must be satisfied at the final time. Since the Lagrange multipliers ν are constants, $\dot{\nu} = 0$. Adjoining these k trivial differential equations with those of equations (2.4) and (2.5) yields the following $2n + k$ system

$$\left.\begin{aligned} \dot{x} &= \left(\frac{\partial H}{\partial \lambda}\right)^T \\ \dot{\lambda} &= -\left(\frac{\partial H}{\partial x}\right)^T \\ \dot{\nu} &= 0 \end{aligned}\right\} \tag{2.10}$$

with the $2n + k + 1$ boundary conditions needed in the case of unknown final time, given by

$$\left.\begin{aligned} x\left(t_o\right) &= x_o && \text{(n conditions)} \\ \lambda^T\left(t_f\right) &= \left.\frac{\partial \phi}{\partial x}\right|_{t_f} + \nu^T \left.\frac{\partial \psi}{\partial x}\right|_{t_f} && \text{(n conditions)} \\ \psi\left(x\left(t_f\right), t_f\right) &= 0 && \text{(k conditions)} \\ H\Big|_{t_f} + \left.\frac{\partial \phi}{\partial t}\right|_{t_f} + \nu^T \left.\frac{\partial \psi}{\partial t}\right|_{t_f} &= 0 && \text{(1 condition)} \end{aligned}\right\} \tag{2.11}$$

To simplify reference to these equations, the system (2.10) is written as an N vector of nonlinear first order differential equations

$$\dot{z} = f(z,t) \tag{2.12}$$

and the boundary conditions (2.11) are generalized to

$$z_i\left(t_o\right) = z_{oi} \qquad i = 1,2,\ldots m \tag{2.13}$$

$$h_i\left(z\left(t_f\right),t_f\right) = 0 \qquad i = 1,2\ldots(N - m + 1) \tag{2.14}$$

Thus, the solution of the optimal control problem is reduced to finding the solution of a system of nonlinear ordinary differential equations with two-point boundary conditions, the terminal boundary conditions in general being nonlinear functions of the terminal state and unknown terminal time.

It should be noted that for many problems the terminal conditions (2.8) may not be very complex, and, as a consequence, the Lagrange multipliers ν may be eliminated from equations (2.7) and (2.9) analytically so that $N + 1$ terminal boundary conditions not involving ν are obtained. This allows the deletion of the k trivial differential equations $\dot{\nu} = 0$ with the associated k terminal boundary conditions, and thus reduces the dimension of the problem considerably. This approach is used in the example problems which will be presented, although the computational algorithm which is developed in succeeding chapters can be applied to the more difficult problem given by equations (2.10) and (2.11).

Since the system (2.12) is, in general, nonlinear, with two-point boundary conditions (2.13) and (2.14), solutions are not easily obtained. The essential problem involved is to determine the missing initial conditions for the Lagrange multipliers so that at some later time equations (2.14) are satisfied.

A numerical method for solving the more simplified version of the above boundary value problem with fixed final time was considered in 1949 by Hestenes [12], who also formulated the general optimal control problem given above [13]. Hestenes [14] explained that his early work was not actively pursued due to a lack of interest in the problem.

Breakwell [15], in 1959, published the general control problem formulation in the form given above and presented numerical results for a variety of problems. The problems were solved by repeated numerical integrations of the nonlinear differential equations with perturbed initial conditions and using an interpolation scheme for determining the initial conditions which would yield the desired terminal condition. A similar approach was used by Melbourne [16], and Melbourne, Sauer, and Richardson [17] for solving fixed time duration optimal payload trajectories for continuous low thrust orbit transfer maneuvers between the Earth and several other planets. These efforts are representative of some of the first attempts to obtain solutions by straightforward "brute force" tactics. These methods resulted in considerable frustration and generally poor convergence or no convergence at all. Although some success was realized through such approaches, the general problems with convergence motivated the development of the direct methods.

The systematic algorithms of contemporary indirect optimization methods can be traced to the papers of Goodman and Lance [18] in 1956 and the work of Kalaba [19] in 1959, although neither of the papers was directly concerned with trajectory optimization. Goodman and Lance [18] discussed numerical solutions of systems of linear differential equations with two-point boundary conditions by the adjoint equations of Bliss [20]. They also proposed a method called complementary functions which utilizes the principle of superposition of particular and homogeneous (or complementary) solutions. In addition, they outlined an approach for solving nonlinear problems by relating initial and final boundary value perturbations of a nominal solution with a system of linear adjoint equations. Kalaba [19] developed the early ideas of Hestenes [12] and produced a method conceptually different from those proposed by Goodman and Lance [18]. This method was called Quasilinearization and required iterative solutions of a system of linear differential equations which were derived from a Taylor series expansion of the nonlinear equations. An initial solution approximation which satisfied initial and final boundary conditions was iteratively improved by repeatedly solving the derived linear equations. Kalaba [19] gave a convergence proof and demonstrated the method for second order differential equations. Both of these approaches for solving nonlinear boundary value problems were restricted to fixed intervals of the independent variable and simple boundary conditions, and, therefore, were not directly applicable to the general boundary value problem considered here. Extensions of the methods soon followed, however, and in this thesis those stemming from the ideas of Goodman and Lance [18] are called Perturbation methods while those following Kalaba and Hestenes are called Quasilinearization methods. The Perturbation and Quasilinearization classifications will be more clearly distinguished in Chapter IV.

The adjoint equation Perturbation approach of Goodman and Lance [18] was extended to solve variable final time optimization problems by Jurovics and McIntyre [21]. Jazwinski [22] developed the method further to allow for boundary conditions which are general functions of the problem variables and time. A procedure for handling inequality constraints on state and control variables was also presented. Breakwell, Speyer, and Bryson [23] independently derived a method similar to Jazwinski's through considerations of the second variation of the calculus of variations.

The alternate Perturbation approach of Goodman and Lance [18], involving complementary functions, was also studied by Breakwell, Speyer, and Bryson [23] and compared to the adjoint Perturbation method from an operational standpoint of a computer storage requirement versus a matrix inversion requirement. Lewallen [24] made extensive comparisions of the two Perturbation techniques and found them to have equivalent convergence characteristics. Further study of the Perturbation methods have been made by Shipman and Roberts [25] and Lastman [26] to show their connection with the famous Kantorovich theorem [27] on Newton's method in functional analysis. Armstrong [28] has proposed a Perturbation method which seeks to iteratively reduce a norm of terminal constraint dissatisfaction and which displays some characteristics of direct methods.

Adaptation of the Quasilinearization approach for optimal control problems was studied by McGill and Kenneth [29], [30], who extended Kalaba's [19] convergence proof for systems of differential equations, and modified the method to solve variable final time problems. Their approach for solving variable final time problems involved the solution of a sequence of fixed final time problems and was inefficient.

A novel approach for solving variable final time problems with the Quasilinearization method was developed independently by Conrad [31] and Long [32] and involves a change of independent variable to one integrated between fixed limits. Further extensions and improvement of the change of variable approach have been proposed by Johnson [33] and Leondes and Paine [34]. Leondes and Paine [34] have also extended McGill and Kenneth's [29] convergence proof for problems with bounded control variables. A different technique for handling variable final time problems with the Quasilinearization method has been proposed by Lewallen [35]. This approach is similar to the one used by Jazwinski [22] for the adjoint Perturbation method, and Lewallen [35] has shown this method to have convergence properties superior to the other above-mentioned Quasilinerization methods. This method is also applicable to problems with general-type boundary conditions. Numerical techniques for handling inequality constraints on control and state variables with the Quasilinearization method have been studied by Kenneth and Taylor [36] and McGill [37].

Although the methods of indirect trajectory optimization are well developed, the methods sometimes are unable to converge to a solution from arbitrary initial solution guesses. Van Dine [38], [39] has sought to circumvent this problem by solving the linear boundary value problem of the Quasilinearization approach with a finite difference technique. Results have been obtained by this approach for fixed final time problems and control variable inequality constraints, but it is doubtful whether the accuracy of other indirect methods can be obtained. Although the method is claimed to avoid the convergence problems of other indirect methods, no direct comparisons on convergence have been published.

The comparison of various direct and indirect trajectory optimization methods by Kopp and McGill [40], Moyer and Pinkham [41], Tapley and Lewallen [42] and Tapley, Fowler, and Williamson [43] have pointed out the desirability for an indirect method with ability to converge from poor initial solution estimates. These studies have indicated that direct methods are more likely to converge from poor solution estimates, but that indirect methods have more rapid and accurate convergence when it occurs. Various strategies have been suggested for improving the range of convergence of indirect methods, but implementation of these strategies often requires considerable skill and effort on the part of the user in order to retain the rapid convergence characteristics of the methods. Several of these schemes have been investigated by Lewallen, Tapley, and Williams [44]. In spite of notable improvement with these strategies, convergence sensitivities remain a problem.

In the following chapters, a method for solving the nonlinear boundary value problem is presented, which displays convergence properties superior to previously published indirect optimization methods. Both Quasilinearization and Perturbation approaches are considered, and a Perturbation approach is selected because of its minimum storage requirement, ease of implementation, and fewer necessary integrations per iteration. The method, which is not developed according to standard perturbation formulations, reveals a new scheme for handling the variable final time problem resulting in a few number of iterations required for convergence. Numerical difficulties which sometimes occur with adjoint equations or perturbation equations are avoided through an alternate method for solving linear boundary value problems. A power series numerical integration scheme is used which allows for a variable integration step size and simultaneous integration of reference and perturbed solutions. This eliminates the approximations of functions evaluated on the reference trajectory necessary without simultaneous integration of reference and perturbed solutions. The characteristically high accuracy capability of power series integration, together with elimination of approximations used in the iterative solution process, give the method presented here a capability for obtaining extremely accurate numerical solutions of boundary value problems in ordinary differential equations.

CHAPTER III

SOLUTION OF LINEAR DIFFERENTIAL EQUATIONS WITH NONLINEAR TWO-POINT BOUNDARY CONDITIONS

An integral part of the method presented in Chapter IV for solving nonlinear boundary value problems requires numerical solutions of linear differential equations with nonlinear boundary conditions. Numerical methods for solving linear differential equations with linear boundary conditions are well known and include (1) the method of complementary solutions [18], (2) the method of adjoint equations [18], [20], and, (3) the method of Green's functions [45]. An alternate method which has received attention in several recent papers [46] to [50] is known as the method of particular solutions. The method is extended here in order to solve systems of linear differential equations subject to two-point nonlinear boundary conditions with an unspecified terminal value of the independent variable.

The method of particular solutions is very similar to the method of complementary functions with the exception that the general solution is obtained by superposition of several particular solutions of the given set of differential equations rather than superposition of a single particular solution and several complementary or homogeneous solutions. When numerical solutions with digital computers are to be obtained, the method of particular solutions displays several important advantages over the above-mentioned methods. First of all, unlike the other methods, only one set of differential equations (the given set) need be programed for solution which reduces the programing complexity. More important, however, is the fact that each solution integrated is a physically possible solution. Therefore, each equation integrated possesses the stability inherent in the physical system model with the result that solution values at the boundaries are closer in magnitude than would be expected for values of homogeneous solutions. This generally results in more numerical accuracy in the determination of superposition constants from inversion of matrices of boundary values. The first stated advantage motivated Miele's work [46]. Holloway [47] encountered numerical instabilities with the method of complementary functions and was led to study superposition of particular solutions because of the second stated advantage.

Other discussions of the method and various applications to two-point and multipoint boundary value problems in ordinary and partial differential equations are given by Luckinbill and Childs [48], Baker and Childs [49], and Heideman [50]. These applications have been limited to problems with linear boundary conditions at specified boundary points. A more general approach for solving problems with terminal boundary conditions given as general nonlinear functions of the problem variables and an unspecified terminal time is developed below.

Consider the N dimensional linear vector differential equation

$$\dot{\underline{y}}(t) = A(t)y + b(t) \tag{3.1}$$

where A and b are a given N × N matrix and N vector, respectively,, of time varying functions. Boundary conditions are given at the initial specified time t_o in the form

$$y_i\left(t_o\right) = y_{oi} \qquad i = 1,2\ldots m < N \tag{3.2}$$

and terminal conditions are specified as general functions

$$h_i\left(y\left(t_f\right),t_f\right) = 0 \qquad i = 1,2\ldots r < N \tag{3.3}$$

If the terminal time t_f is specified, then $r = N - m$. If t_f is not specified, then $r = N - m + 1$.

A general solution of equation (3.1) satisfying equations (3.2) and (3.3) can be represented by

$$y(t) = \sum_{k=1}^{S+1} \alpha_k p^k(t), \qquad S = N - m$$

with the auxiliary condition

$$\sum_{k=1}^{S+1} \alpha_k = 1$$

where any S of the p^k are linearly independent particular solutions of equation (3.1), and the α_k are superposition constants. Initial conditions for the $p^k(t)$ are chosen such that

$$\left.\begin{array}{ll}
p_i^1\left(t_o\right) = y_{oi} & i = 1,2\ldots m \\
p_i^1\left(t_o\right) = \text{any value} & i = m+1,m+2\ldots N \\
p_i^k\left(t_o\right) = \beta_{ik}p_i^1\left(t_o\right) + \gamma_{ik} & k = 2,3\ldots(S+1) \\
 & i = 1,2\ldots N \\
\text{where} & \\
\beta_{ik} = \begin{cases} 1 \text{ if } i \neq k+m-1 \\ \beta_i \text{ if } i = k+m-1 \end{cases} & \\
\gamma_{ik} = \begin{cases} 0 \text{ if } i \neq k+m-1 \\ \gamma_i \text{ if } i = k+m-1 \\ \quad \text{and } \left|p_i^1\left(t_o\right)\right| < \gamma_i \end{cases} &
\end{array}\right\} \qquad (3.4)$$

The particular choice of β_{ik} and γ_{ik} given, insure the condition for linear independence of particular solutions. The choice of the constants β_i and γ_i is free except that β_i, $\gamma_i \neq 0$, and $\beta_i \neq 1$. These constants can be chosen, depending on the sensitivity of the system, to control the magnitude of terminal values of the particular solutions.

The superposition constants α_k are to be chosen so that the equations in (3.3) are satisfied and the condition for superposition of particular solutions

$$\sum_{k=1}^{S+1} \alpha_k - 1 = 0 \tag{3.5}$$

is satisfied. In order to determine the α_k, a formal substitution of $\sum_{k=1}^{S+1} \alpha_k p^k(t)$ is made for $y(t)$ in equation (3.3), and equation (3.5) is written as h_{r+1}, to obtain an r+1 vector of functions h with elements

$$\left.\begin{aligned} h_i\Big(y\big(\alpha_1,\alpha_2,\ldots\alpha_{S+1},t_f\big),t_f\Big) &= 0 \qquad i = 1,2\ldots r \\ h_{r+1}\big(\alpha_1,\alpha_2\ldots\alpha_{S+1}\big) &= 0 \end{aligned}\right\} \tag{3.6}$$

For the case where t_f is unknown, r = S + 1, and equation (3.6) represents S + 2 nonlinear equations in the S + 2 unknowns α_k and t_f. The equations in (3.6) can be solved for the α_k and t_f by a Newton-Raphson [51] iterative procedure.

The Newton-Raphson procedure is employed in the following manner. An initial guess for the α_k and t_f is written as a column vector

$$\alpha^{(0)} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{S+1} \\ t_f \end{bmatrix}$$

Successive approximations for the proper values of this vector are obtained by repeated solution of the following equation

$$\alpha^{(n+1)} = \alpha^{(n)} - \left[J^{(n)}\right]^{-1} h^{(n)} \qquad n=0,1,2... \tag{3.7}$$

where $J^{(n)}$ is the Jacobian matrix with elements

$$\begin{aligned} J_{ij} &= \frac{dh_i}{d\alpha_j} && i = 1,2...S+2 \\ &&& j = 1,2...S+1 \\ J_{ij} &= \frac{dh_i}{dt_f} && i = 1,2...S+2 \\ &&& j = S+2 \end{aligned} \tag{3.8}$$

and the n superscript denotes evaluation with the nth approximation for $\alpha^{(n)}$.

Expanding the derivatives of equation (3.8) and denoting elements of $y(t)$ by y_ℓ

$$\frac{dh_i}{d\alpha_j} = \sum_{\ell=1}^{N} \frac{\partial h_i}{\partial y_\ell}\frac{\partial y_\ell}{\partial \alpha_j} + \frac{\partial h_i}{\partial \alpha_j} = \sum_{\ell=1}^{N} \frac{\partial h_i}{\partial y_\ell} p_\ell^j\left(t_f\right) + \frac{\partial h_i}{\partial \alpha_j}$$

since

$$y_\ell(t) = \sum_{j=1}^{S+1} \alpha_j p_\ell^j(t)$$

Also,

$$\frac{dh_i}{dt_f} = \sum_{\ell=1}^{N} \frac{\partial h_i}{\partial y_\ell} \dot{y}_\ell\left(t_f\right) + \frac{\partial h_i}{\partial t_f} = \sum_{\ell=1}^{N} \frac{\partial h_i}{\partial y_\ell}\left(\sum_{k=1}^{S+1} \alpha_k \dot{p}_\ell^k\left(t_f\right)\right) + \frac{\partial h_i}{\partial t_f}$$

The method of solution requires the forward integration of the (S+1) particular solutions of equation (3.1), $p^k(t)$, with initial conditions given by equation (3.4) from t_o to some assumed final time t_a. At the assumed final time, the Jacobian matrix and the functions h_i are evaluated using initial guesses for the α_k. The equation (3.7) yields new approximations for α_k and t_f. To continue the iteration of equation (3.7), it is necessary to integrate the particular solutions from the assumed final time to the new estimate for final time. The forward and backward integrations which may be necessary from the sequence of final times generated by the iteration of equation (3.7) may be excessively cumbersome for some numerical integration methods. However, the power series integration scheme discussed in Appendix B is well suited for this problem.

Application of the power series integration method yields Mth order polynomial approximations for $p_\ell^k(t_f)$ and $\dot{p}_\ell^k(t_f)$ written as

$$
\begin{aligned}
p_\ell^k\left(t_f\right) &= \sum_{i=1}^{M} a_{k,\ell,i} \left(t_f - t_a\right)^{i-1} \\
\dot{p}_\ell^k\left(t_f\right) &= \sum_{i=1}^{M} b_{k,\ell,i} \left(t_f - t_a\right)^{i-1}
\end{aligned}
\tag{3.10}
$$

where the $a_{k,\ell,i}$ and $b_{k,\ell,i}$ are known power series coefficients determined by the method of Appendix B and t_a is used as the origin of the power series expansions. If t_a is sufficiently close to the true final time, the equations (3.10) represent sufficiently accurate formulae in the application of equation (3.7). If $|(t_f - t_a)|$ becomes too large, a new center of expansion can be used as explained in Appendix B so that a specified accuracy is retained.

With sufficiently close initial approximation for α_k and t_f, the sequence (3.7) is rapidly convergent and yields the desired values for the α_k and t_f. Upon convergence of equation (3.7), the general solution of equation (3.1) satisfying equation (3.2) and equation (3.3) can be obtained by integrating (3.1) over the interval $[t_o,t_f]$ with initial conditions

$$y_i\left(t_o\right) = y_{oi} \qquad i = 1,2...m$$

$$y_i\left(t_o\right) = \sum_{k=1}^{S+1} \alpha_k p_i^k\left(t_o\right) \qquad i = m+1,m+2,...N$$

In this manner, the solution $y(t)$ can be constructed without storing the solutions $p^k(t)$.

Although the method requires initial approximations, this has not been found to be a problem. If it happens that estimates are available for the missing initial conditions, then using these estimates for $p^1(t_o)$ implies α_1 should be chosen near unity with other values of α_k near zero. A very accurate method of estimating the α_k is obtained when the method is used in an iterative technique for solving nonlinear boundary value problems. This is discussed in the following chapter.

CHAPTER IV

METHODS FOR SOLVING NONLINEAR TWO-POINT BOUNDARY VALUE PROBLEMS WITH NONLINEAR BOUNDARY CONDITIONS

A fundamental idea in most contemporary methods for solving nonlinear boundary value problems in ordinary differential equations is to iteratively solve an associated set of linear differential equations. The solutions either converge to the nonlinear solution or provide a sequence of initial conditions which converge to the proper set of initial conditions for the nonlinear system. The linear differential equations are obtained by Taylor Series expansions of the functions defining the nonlinear system. This linearization process is common to many methods appearing under the various titles of Quasilinearization [19], [31], [35]; Generalized Newton-Raphson Method [30], [32]; Modified Newton's Method [26]; Second Variation Methods [23]; Adjoint Method [21]; and Method of Perturbation Functions [42]. The systems of linear differential equations used by these methods are similar and in many cases identical. However, the actual sequence of approximate solutions generated may differ considerably depending on the type of initial solution approximation used, the manner in which given boundary conditions are employed, and the reference solution used in the linearization expansion for each iteration.

The type of reference solution used for each iteration provides a basic classification of the methods into two groups which in this thesis will be called Perturbation methods and Quasilinearization methods.

Perturbation methods use a solution of the given nonlinear system with approximate values for unknown initial conditions as the reference solution. The reference solution for Quasilinearization methods is a solution of a system of linear differential equations derived from the nonlinear system. Both Quasilinearization and Perturbation methods are developed below using the ideas of Chapter III for obtaining solutions of linear systems with nonlinear boundary conditions. The Quasilinearization method obtained is recognized to be very similar to one proposed by Lewallen [35]. The Perturbation method obtained is significantly different from previously derived Perturbation methods and offers some decided advantages over presently known methods.

Consider the system of N nonlinear differential equations

$$\dot{z} = f(z,t) \tag{4.1}$$

with initial conditions

$$z_i\left(t_o\right) = z_{oi} \qquad i = 1,2,\ldots m \tag{4.2}$$

and final conditions and final time given implicitly as the first time the following q = N - m + 1 vector of functions is satisfied

$$h\left[z\left(t_f\right),t_f\right] = 0 \tag{4.3}$$

In general, a solution of equation (4.1) with initial conditions, equation (4.2), and arbitrary values for the unspecified initial conditions will not satisfy the terminal conditions, equation (4.3). Denote such a solution by ${}^1z(t)$, where the superscript is used to index the first approximation to a solution of equation (4.1) satisfying both the initial and final conditions.

Let w(t) be an N vector of functions satisfying

$$w_i\left(t_o\right) = z_{oi} \qquad i = 1,2...m$$

$$h\left[w\left(t_f\right),t_f\right] = 0$$

as well as the differential equation

$$\dot{w} = f(w,t) \qquad t_o \leq t \leq t_f \tag{4.4}$$

where the vector of functions f in equation (4.4) is the same vector of functions appearing in equation (4.1). Clearly, w(t) is a desired solution of equation (4.1) satisfying both equations (4.2) and (4.3). The functions appearing on the right-hand side of equation (4.4) can be expressed in a Taylor series expansion about the reference or approximate solution ${}^1z(t)$. That is

$$\dot{w} = f(w,t) = f\left({}^1z,t\right) + \frac{\partial f\left({}^1z,t\right)}{\partial z}\left(w - {}^1z\right) + \ldots \tag{4.5}$$

If the expansion is truncated after the second term, the following linear differential equation, which is an approximation to equation (4.5), is obtained

$$\dot{w} \approx {}^1\dot{y} = A(t){}^1y + b(t) \tag{4.6}$$

where

$$A(t) = \frac{\partial f\left({}^{1}z,t\right)}{\partial z}$$

and

$$b(t) = f\left({}^{1}z,t\right) - A(t)\,{}^{1}z(t)$$

A solution of equation (4.6) subject to the boundary conditions,

$$ {}^{1}y_i\left(t_o\right) = z_{oi} \qquad i = 1,2,\ldots m \tag{4.7}$$

$$ h\left[{}^{1}y\left({}^{1}t_f\right),{}^{1}t_f\right] = 0 \tag{4.8}$$

will yield an approximation for $w(t)$. The solution ${}^{1}y$ and corresponding value for final time ${}^{1}t_f$ can be obtained by the method of particular solutions described in Chapter III. The accuracy of the approximate solution obtained depends on the closeness of the solutions ${}^{1}z$ and w. Assume for the present that ${}^{1}z$ is sufficiently close to w so that ${}^{1}y$ is a better approximation than ${}^{1}z$. The manner in which additional approximate solutions are obtained determines whether the approach taken is categorized as a Perturbation method or a Quasilinearization method.

A Quasilinearization Method

Since ${}^{1}y$ is a time varying vector of functions which is a better approximation for w than is ${}^{1}z$, then it is reasonable to replace ${}^{1}z$

with ${}^{1}y$ in the Taylor series expansion of equation (4.5). The linear differential equation (4.6) is then written

$$\dot{w} \approx {}^{2}\dot{y} \approx f\left({}^{1}y,t\right) + \frac{\partial f\left({}^{1}y,t\right)}{\partial z}\left({}^{2}y - {}^{1}y\right) \tag{4.9}$$

and a solution of this equation subject to the given boundary conditions provides a new approximate solution ${}^{2}y$. According to the definition set forth previously, this is a Quasilinearization method, the reference solution in the Taylor series expansion being a solution of the linearized differential equation. Using ${}^{2}y$ as the next reference solution yields ${}^{3}y$ and so on for ${}^{4}y$, ${}^{5}y$, ${}^{6}y, \ldots {}^{n}y$. Under appropriate conditions provided by the convergence proofs of references [29] and [34], the sequence of solutions ${}^{n}y$ converges to the desired solution w. These convergence proofs are restricted to problems with more simple boundary conditions than those expressed in equation (4.3).

A closer observation of the Quasilinearization method with regard to the technique presented in Chapter III for solving the linear system with nonlinear boundary conditions reveals a fundamental difficulty. It may happen that ${}^{2}t_f$, the terminal time corresponding to the solution ${}^{2}y$, *may be larger than* ${}^{1}t_f$, *the value of* t_f *obtained in the solution* for ${}^{1}y$. In this case, ${}^{1}y$ is not known beyond ${}^{1}t_f$, and the differential equation (4.9) is not defined over the necessary range $t_o \leq t \leq {}^{2}t_f$. However, if only one Newton-Raphson iteration with equation (3.7) is made, and a linear extrapolation for ${}^{1}y$ is used on the interval $\left[{}^{1}t_f, {}^{2}t_f\right]$, it is not necessary to integrate ${}^{1}y$ past ${}^{1}t_f$ in order to construct ${}^{2}y$ and a workable iterative scheme is realized. Lewallen [35], using somewhat different arguments, has presented a Quasilinearization method which solves the variable final time problem essentially in the same manner proposed here. The primary difference between the method suggested here and the one proposed by Lewallen is that particular solutions rather than homogeneous solutions are used to construct the general solution of the linear system.

Lewallen [35] has compared this Quasilinearization approach with the Quasilinearization techniques described by Long [32], and McGill and Kenneth [30], and has found this particular approach to have better convergence characteristics. Another attractive feature of the method is the capability for solving problems with general terminal boundary conditions of the form given in equation (4.3).

A Perturbation Method

In order to proceed with the development of the Perturbation method, it is assumed that ${}^1y$ (the solution of the problem defined by equations (4.6) to (4.8)) has been obtained. It is further assumed that ${}^1y$ is a better approximate solution than ${}^1z$. (A method for satisfying this assumption is discussed later in the presentation.) In particular, ${}^1y(t_o)$ should be a better approximation for $w(t_o)$ than was ${}^1z(t_o)$. It is reasonable to expect that a solution of equation (4.1) using initial conditions, ${}^1y(t_o)$, will be a better approximation for w than was ${}^1z$. Such a solution is denoted by ${}^2z$. Replacing ${}^1z$ with ${}^2z$

in the Taylor Series expansion (4.5) and the associated linear differential equation (4.6) yields

$$\dot{w} \approx {}^{2}\dot{y} = f\left({}^{2}z,t\right) + \frac{\partial f\left({}^{2}z,t\right)}{\partial z}\left({}^{2}y - {}^{2}z\right) \qquad (4.10)$$

If the solution ${}^{2}y$ is to closely approximate w, it must also satisfy similar boundary conditions, hence

$$\left.\begin{aligned} {}^{2}y_i\left(t_o\right) &= z_{oi} \qquad i = 1,2,\ldots m \\ h\left[{}^{2}y\left({}^{2}t_f\right),{}^{2}t_f\right] &= 0 \end{aligned}\right\} \qquad (4.11)$$

The solution ${}^{2}y$ can be obtained by the method described in Chapter III, and the difficulty with final time encountered with the Quasilinearization method can be avoided. This is accomplished by generating ${}^{2}z$ by integration of equation (4.1) simultaneously with the integration of all particular solutions of equation (4.10). In this manner the reference solution ${}^{2}z$ is integrated to each estimate of final time required by the algorithm of Chapter III. The solution ${}^{2}z$ is thus defined over the entire range of time required for the solution of equation (4.10) subject to the boundary conditions (4.11). Consequently, as many Newton-Raphson iterations as desired with equation (3.7) can be made. This gives one the capability to obtain the initial conditions ${}^{2}y(t_o)$, so that the terminal conditions (4.11) are satisfied as accurately as desired. This capability has not been possible with existing methods for solving the problem defined by equations (4.1) to (4.3). Once ${}^{2}y(t_o)$ is determined, a third reference solution ${}^{3}z$ can be generated by integration of equation (4.1) using initial conditions ${}^2y(t_o)$. The new reference solution allows continuation of the iterative process. If ${}^1z$ is sufficiently close to w, the sequence of initial conditions ${}^ny(t_o)$ converges to $w(t_o)$, and hence, ${}^nz$ converges to w. By the definition set forth previously, this process is a Perturbation method, since the reference solution at each iteration is a solution of the given nonlinear system.

Modifications for Improving Convergence

In the foregoing discussions of Quasilinearization and Perturbation methods, it was assumed that the starting solution ${}^1z$ was sufficiently close to the true solution w, so that convergence of the methods resulted. In practice, it is sometimes difficult to find an initial approximate solution which will lead to convergence. Consequently, various modifications of the basic methods outlined above have been proposed to improve convergence characteristics [23], [26], [34], and [44]. These modifications are commonly referred to as "iteration schemes" or "convergence schemes," and the usefulness of a given method is often closely tied to the "convergence schemes" employed. Two modifications to the basic procedures described above are discussed here. They should be considered to be integral parts of the basic methods rather than schemes which are just added on.

Consider a simple version of the nonlinear problem described by equations (4.1) to (4.3) such that the terminal time is specified and terminal boundary conditions are given in the form

$$z_i\left(t_f\right) = z_{fi} \qquad i = k + 1, k + 2, \ldots (k + N - m)$$

for some k. (The restriction of fixed final time in this discussion can be removed by employing the transformation of independent variable as described by Conrad [31] and Long [32].) The following discussion assumes that the Perturbation method is used, but parallel arguments can be made for the Quasilinearization method.

An approximate initial solution ${}^{1}z$ is obtained in the manner described previously to yield at the fixed final time the values

$${}^{1}z_i\left(t_f\right) \qquad i = 1,2,\ldots N$$

Let w be a solution of the problem. Instead of, as before, seeking an approximation for w on each iteration, a function ${}^{n}w$ (in this case $n = 1$) is sought which has initial conditions

$${}^{n}w_i\left(t_o\right) = z_{oi} \qquad i = 1,2,\ldots m$$

and terminal conditions

$${}^{n}w_i\left(t_f\right) = \varepsilon z_{fi} + (1 - \varepsilon)\, {}^{n}z_i\left(t_f\right) \qquad \begin{array}{c} i = k + 1, k + 2, \ldots (k + N - m) \\ 0 < \varepsilon \leq 1 \end{array}$$

That is, ${}^{n}w$ has terminal values between those of the reference solution and the desired solution. By choosing ε sufficiently small, the approximation

$${}^{n}\dot{w} \approx {}^{n}\dot{y} = f\left({}^{n}z,t\right) + \frac{\partial f\left({}^{n}z,t\right)}{\partial z}\left({}^{n}y - {}^{n}z\right) \tag{4.12}$$

with

$$^{n}y_i(t_o) = {}^{n}w_i(t_o) \qquad i = 1,2,\ldots m$$

$$^{n}y_i(t_f) = {}^{n}w_i(t_f) \qquad i = k + 1, k + 2, \ldots (k + N - m)$$

can be made as accurate as desired. To insure convergence, the approximation must be sufficiently accurate such that the initial condition vector

$$^{n+1}z(t_o) = {}^{n}y(t_o)$$

is sufficiently close to $^{n}w(t_o)$, and hence, $^{n+1}z(t_f)$ sufficiently close to $^{n}w(t_f)$, so that $^{n+1}z(t_f)$ is closer to $w(t_f)$ than is $^{n}z(t_f)$. By choosing ε sufficiently small, initial conditions for successive reference solutions are obtained which yield terminal conditions closer to the desired conditions than the previous reference solution. Repetition of the process for $n = 1,2,3\ldots$ results in the construction of a sequence of terminal conditions $^{n}w(t_f)$ converging to $w(t_f)$, a sequence of initial conditions $^{n}z(t_o)$ converging to $^{n}w(t_o)$, and hence, solutions ^{n}z converging to w.

A practical consideration is that some method for choosing ε is necessary. From the above discussion it is apparent that there exists a value at each iteration which will work, but no a priori means of determining ε at each iteration is known. A trial and error method could be employed but this could be very inefficient. A simple method for choosing ε is proposed here which is based on practical considerations and which improves chances for convergence in a specified maximum number of iterations. Since it is apparent that the numerical methods proposed here will be used only with the aid of digital computers, it is always necessary to limit the number of iterations which can be attempted in order to conserve computer time and costs.

First it will be noted that the scheme given above for choosing boundary values is equivalent to the following procedure.

1. Solve equation (4.12) subject to the given initial and terminal boundary values

$$ {}^{n}y_i\left(t_o\right) = z_{oi} \qquad i = 1,2,\ldots m $$

$$ {}^{n}y_i\left(t_f\right) = z_{fi} \qquad i = k + 1,k + 2,\ldots(k + N - m) $$

2. Using the method of Chapter III, determine trial values for the missing initial conditions

$$ {}^{n}y_i\left(t_o\right) \qquad i = m + 1,m + 2,\ldots N $$

3. Compute a final set of missing initial conditions for the next reference trajectory ${}^{n+1}z$ from

$$ {}^{n+1}z_i\left(t_o\right) = {}^{n}z_i\left(t_o\right) + \varepsilon\left[{}^{n}y_i\left(t_o\right) - {}^{n}z_i\left(t_o\right)\right] \qquad i = m + 1,m + 2,\ldots N $$

It is shown in Appendix C that the values obtained for ${}^{n+1}z_i(t_o)$ using a given ε are equivalent for both the case where terminal conditions are modified before solving for initial conditions and the case described in steps 1, 2, 3 above. Implementation of the second approach is easier than the first approach described because the change in initial conditions

$$ {}^{n}\Delta z(t_o) = {}^{n}y_i(t_o) - {}^{n}z_i(t_o) $$

can be computed before choosing ε. Furthermore, the latter method is simpler to apply for the more general problem with unspecified final time and nonlinear terminal boundary conditions, since it is only necessary to modify the missing initial conditions computed for each new reference trajectory.

The following method for choosing ε is proposed. Let M be the maximum number of solution iterations which can be allowed because of limitations of computer time and costs. When initial guess values are chosen for the missing initial values of the starting solution, also estimate the maximum deviation of a guessed value from its true value. Denote the maximum of the estimated deviations by d. Also choose a suitable norm for measuring the computed change ${}^{n}y(t_o) - {}^{n}z(t_o)$. For example

$$ \rho\left[{}^{n}y(t_o), {}^{n}z(t_o)\right] = \left\{ \sum_{i=m+1}^{N} \frac{\left[{}^{n}y_i(t_o) - {}^{n}z_i(t_o)\right]^2}{N - m} \right\}^{1/2} $$

Compute a maximum allowable "initial condition change step size" from $\delta \approx \frac{d}{M}$, for example, $\delta = \frac{3}{4}\frac{d}{M}$. On each reference trajectory iteration, compute the norm, $\rho[{}^n y(t_o), {}^n z(t_o)]$, and compare its value to δ. Choose ε according to

$$\varepsilon = \begin{cases} \dfrac{\delta}{\rho\left[{}^n y(t_o), {}^n z(t_o)\right]} & \text{if } \rho\left[{}^n y(t_o), {}^n z(t_o)\right] > \delta \\ 1 & \text{if } \rho\left[{}^n y(t_o), {}^n z(t_o)\right] \leq \delta \end{cases} \tag{4.13}$$

Initial conditions for successive reference trajectories are computed from

$${}^{n+1}z_i(t_o) = {}^n z_i(t_o) + \varepsilon\left[{}^n y_i(t_o) - {}^n z_i(t_o)\right] \qquad i = m + 1, \ldots N \tag{4.14}$$

For variable final time problems it may also be necessary to modify successive final time estimates according to

$${}^{n+1}t_f = {}^n t_f + \varepsilon\ {}^n\Delta t_f \tag{4.15}$$

where ${}^n\Delta t_f$ is a computed change for final time determined in the solution for ${}^n y(t_o)$.

The attractive features of this method for inducing convergence from poor initial estimates is that (1) very little effort on the part of the user is required (all he must do is choose δ), (2) as the solutions begin to converge, the scheme does not retard convergence and (3) chances for convergence with one computer run are maximized consistent with available computer time. Of course, if δ is chosen to be much smaller than necessary, the rate of convergence may be slowed considerably. On problems which do not display convergence sensitivities, δ should be chosen arbitrarily large so that rate of convergence is not retarded.

In situations where δ is chosen to be too large, a second modification can be employed which may induce convergence. When δ is too large, a typical behavior is for initial conditions to be chosen in the proper direction for several iterations but then as the true values are approached, one or more of the unknown values may oscillate about its true value on successive iterations. When this occurs, halving the computed change for the oscillating value will bring it closer to its true value. Thus, the following procedure is proposed.

$$\left.\begin{array}{ll}
\text{(a)} & \text{Compute } {}^{n+1}z_i\left(t_o\right) \text{ as described above,} \\
 & \qquad i = m + 1, m + 2 \ldots N \\
\text{(b)} & \text{Compute } \left[{}^{n+1}z_i\left(t_o\right) - {}^{n}z_i\left(t_o\right)\right] \Big/ \left[{}^{n}z_i\left(t_o\right) - {}^{n-1}z_i\left(t_o\right)\right] \\
\text{(c)} & \text{If the above quantity is less than } -1/2\text{, compute} \\
 & {}^{n+1}z_i\left(t_o\right) = {}^{n}z_i\left(t_o\right) - \frac{1}{2}\left[{}^{n}z_i\left(t_o\right) - {}^{n-1}z_i\left(t_o\right)\right]
\end{array}\right\} \quad (4.16)$$

If the quantity in (b) is positive, the particular element of the vector is not oscillating on successive iterations. If the quantity in (b) is negative but larger than -1/2, then the oscillation has a convergent nature. In either case there is no reason to modify the computed value for ${}^{n+1}z_i(t_o)$. This modification may also be applied to successive final time estimates for variable final time problems.

There are other possible variations of the two basic modifications presented above. For example, one might reduce the value of δ whenever the norm $\rho[{}^{n}y(t_o),{}^{n}z(t_o)]$ is computed to be less than δ, and/or δ might be reduced whenever oscillation of one or more initial condition values or final time value occurs. Details of such procedures are best worked out through numerical experiments. When upper and lower bounds are known for missing initial conditions and/or final time, these bounds should be imposed in the event that the values chosen violate these bounds.

Comparison of Quasilinearization and Perturbation Computational Requirements

A basic goal of this investigation is to formulate an improved computational method for solving nonlinear two-point boundary value problems. While convergence characteristics are a major concern, other factors such as ease of implementation, computer storage requirements, computer time per iteration, and control of solution accuracy are also important. Two basic methods, Quasilinearization and Perturbation, have been proposed from a theoretical standpoint. A comparison of the computational requirements and restrictions of each method is made here. This comparison reveals the Perturbation method to be a more efficient computational scheme, especially when used in a unified approach with the particular solution method of Chapter III and the power series numerical integration method discussed in Appendix B.

A distinctive computational feature of the Quasilinearization method, often considered to be an advantage of the method, is that it is not necessary to program the given nonlinear system of equations for solution. For convenience in the previous presentation of the method, it was assumed that an initial guess solution was obtained by integration of the given nonlinear system with assumed initial conditions. This is not necessary since any guess solution satisfying only the boundary conditions can be used. This "advantage" of the method is lost, however, if a starting solution is generated by integration of the nonlinear system. With the Quasilinearization method, one has an option of storing each particular solution of the linear system at each integration step and forming the reference solution by the properly weighted sum of these solutions, or one may avoid the storage problem by integrating the linear system with the proper initial conditions to form the reference solution. With reference to equation (4.9), the latter approach still requires that the values for $f({}^{n}y,t)$ and $\partial f({}^{n}y,t)/\partial z\ {}^{n}y$ be stored at each numerical integration step. To simplify access to these stored quantities, one is forced to use numerical integration schemes which use a fixed integration step size. The choice of this step size is influenced not only by truncation error of the numerical integration scheme, but also by the required spacing of the stored quantities in order to achieve the necessary accuracy for the approximation of these functions along the reference trajectory. Thus, selection of integration step size in order to achieve a specified final solution accuracy is not a routine matter. The restriction to fixed numerical integration step size could be a serious handicap for problems where considerable integration speed and accuracy are realized through frequent changes in integration step size. Many of the boundary value problems arising in optimal control theory (for example, those in interplanetary navigation) have this property. The Quasilinearization method, with the minimum storage option, requires $N - m + 2$ integrations of the linear system (eq. (4.9)) at each reference solution iteration since $N - m + 1$ integrations are required to determine the proper initial conditions, and then these initial conditions must be used in one additional integration of the system to generate the reference trajectory.

In comparison, the Perturbation method offers some unique computational advantages. Since it is never necessary to generate the entire solution of the linear system (4.12), but only the initial conditions ${}^{n}y(t_o)$, there is no need for storing perturbed particular solutions of the linear system. Furthermore, since the reference solution ${}^{n}z$ can be generated by simultaneously integrating equation (4.1) forward with all particular solutions of equation (4.12), the quantities $f({}^{n}z,t)$ and $\partial f({}^{n}z,t)/\partial z\ {}^{n}z$ appearing in equation (4.12) need not be saved. They are merely computed from ${}^{n}z$ at each integration step, used in all integrations of equation (4.12) for the integration step, and then discarded. With this procedure, variable step integration schemes may be used since there is no need to restrict end points of numerical integration steps to coincide with previously stored information.

The combination of simultaneous and variable step integration of the nonlinear and linearized differential equations which is possible with the Perturbation method provides an additional advantage for this approach. The variable step capability allows one to use integration schemes which automatically determine an integration step size to yield a specified solution accuracy. The simultaneous integration of the nonlinear and linearized equations not only eliminates storage of the functions $f(^{n}z,t)$ and $\frac{\partial f(^{n}z,t)}{\partial z}\,{}^{n}z$, but it also eliminates the necessity for interpolation schemes used to convert discreet values of these functions into more accurate approximations over the integration step. Simultaneous integration automatically provides the interpolation for these functions through the mechanics of the particular integration scheme used. With the variable step power series integration method discussed in Appendix B, Taylor series expansions of these functions are generated which yield an approximation accuracy equal to the desired integration accuracy. The automatic step size selection of this method also relieves the user of the burden of determining beforehand an acceptable integration step size.

In addition to the above-mentioned computational advantages of the Perturbation method over the Quasilinearization method, the Perturbation method requires one less numerical integration per iteration of a comparable set of differential equations. This may not be immediately obvious since it has been previously indicated that $N - m + 1$ integrations are required to solve the linear system (4.12) and one integration of equation (4.1) is necessary to construct the reference trajectory. This totals to $N - m + 2$ integrations per iteration, but only $N - m + 1$ are required if the following observation is made.

Theorem 1: A solution ${}^{n}z$ of the nonlinear system (4.1) is identical to a particular solution of the linear system (4.12) if initial conditions of the two solutions are identical.

Proof: Let p be a particular solution of equation (4.12) and let ${}^{n}z$ be a solution of equation (4.1). Let x be defined

$$x(t) = p(t) - {}^{n}z(t)$$

which implies

$$\dot{x}(t) = \dot{p}(t) - {}^{n}\dot{z}(t)$$

Since p satisfies equation (4.12) and ${}^{n}z$ satisfies equation (4.1),

$$\dot{x} = \dot{p} - {}^{n}\dot{z} = f\left({}^{n}z,t\right) + \frac{\partial f\left({}^{n}z,t\right)}{\partial z}\left(p - {}^{n}z\right) - f\left({}^{n}z,t\right)$$

or

$$\dot{x} = \frac{\partial f}{\partial z}\left(p - {}^{n}z\right) = \frac{\partial f\left({}^{n}z,t\right)}{\partial z}\, x$$

Now this is a homogenous linear differential equation, and by hypothesis

$$x\left(t_o\right) = p\left(t_o\right) - {}^{n}z\left(t_o\right) = 0$$

For these initial conditions, it is well known (see, for example, Petrovski [52]) that the solution for $x(t)$ is

$$x(t) \equiv 0$$

which implies

$$p(t) \equiv {}^{n}z(t)$$

and thus the proof is complete.

Using this theorem, one of the $N - m + 1$ integrations of equation (4.12) can be eliminated since the reference trajectory ${}^{n}z$ can be used in its place.

A further point of comparison of the computational requirements for Perturbation and Quasilinearization methods is concerned with the manner in which convergence is detected for the methods. For the Perturbation method, a direct indication of convergence is given when the reference trajectory satisfies the terminal boundary conditions to some specified accuracy, or when the change in the initial condition vector is less than a specified accuracy. However, with the Quasilinearization approach, the reference trajectory does not satisfy the nonlinear system until convergence has occurred. To determine when successive trajectory iterations are converging, it is necessary to compute some suitable norm $\rho[{}^{n}y(t),{}^{n-1}y(t)]$. The computation of this norm requires a comparison of the successive reference trajectories at each integration step and consequently requires additional programing and computer time.

This comparison of the computational requirements and restrictions of the Quasilinearization and Perturbation methods indicates that the Perturbation method is somewhat easier to implement, and is better suited for adaptation with the method of particular solutions described in Chapter III and the power series integration scheme presented in Appendix B. Outlined below is a computational algorithm which combines these various concepts together with the proposed modifications for extending the range of convergence in a unified method for solving the nonlinear two-point boundary value problem in ordinary differential equations.

The Particular Solution Perturbation Method

To obtain an efficient computational algorithm utilizing the concepts set forth in this chapter and the preceding chapter, a study of the manner in which these various ideas are incorporated into an integrated framework is in order. Because the Perturbation concept is employed with the method of particular solutions, the algorithm described below is referred to as the Particular Solution Perturbation Method (PSPM).

On each solution iteration, the PSPM requires a simultaneous forward integration of the given N dimensional nonlinear system

$$^{n}\dot{z} = f\left(^{n}z,t\right) \tag{4.17}$$

together with S forward integrations of the derived linear system

$$^{n}\dot{y} = \frac{\partial f\left(^{n}z,t\right)}{\partial z}\,^{n}y + f\left(^{n}z,t\right) - \frac{\partial f\left(^{n}z,t\right)}{\partial z}\,^{n}z \tag{4.18}$$

where $S = N - m$ and m is the number of specified initial conditions

$$^{n}z_i\left(t_o\right) = z_{oi} \qquad i = 1,2,\ldots m$$

The terminal conditions specified for the nonlinear system are assumed to be of the form

$$h_i\left[z\left(t_f\right),t_f\right] = 0 \qquad i = 1,2,\ldots(S + 1)$$

so that the linear system is to satisfy boundary conditions given by

$$\left.\begin{aligned} {}^{n}y_i\left(t_o\right) &= z_{oi} \qquad & i &= 1,2,\ldots m \\ h_i\left[{}^{n}y\left({}^{n}t_f\right),{}^{n}t_f\right] &= 0 \qquad & i &= 1,2,\ldots(S + 1) \end{aligned}\right\} \tag{4.19}$$

Using the method of particular solutions, ${}^{n}y$ is expressed

$${}^{n}y(t) = \sum_{k=1}^{S+1} \alpha_k \, {}_{n}p^{k}(t)$$

Subject to

$$\sum_{k=1}^{S+1} \alpha_k = 1$$

where any S of the ${}_{n}p^{k}$ are linearly independent particular solutions of equation (4.18), and the α_k are superposition constants. Theorem 1 is used to write

$${}_{n}p^{1}(t) = {}^{n}z(t)$$

and the systems (4.17) and (4.18) are written

$$\left.\begin{aligned} {}_n\dot{p}^1 &= f\left({}_np^1,t\right) \\ {}_n\dot{p}^k &= \frac{\partial f\left({}_np^1,t\right)}{\partial z}\,{}_np^k + f\left({}_np^1,t\right) \\ &\quad - \frac{\partial f\left({}_np^1,t\right)}{\partial z}\,{}_np^1 \qquad k = 2,3,\ldots(S+1) \end{aligned}\right\} \qquad (4.20)$$

with m initial conditions for each solution provided by

$${}_np_i^k\left(t_o\right) = z_{oi} \qquad \begin{aligned} i &= 1,2,\ldots m \\ k &= 1,2,\ldots(S+1) \end{aligned}$$

and other boundary conditions

$$\left.\begin{aligned} h_i\left[\sum_{k=1}^{S+1} \alpha_k\,{}_np^k\left({}^nt_f\right),{}^nt_f\right] &= 0 \qquad i = 1,2,\ldots(S+1) \\ h_{S+2}\left(\alpha_1,\ldots,\alpha_{S+1}\right) = \sum_{k=1}^{S+1}\alpha_k - 1 &= 0 \end{aligned}\right\} \qquad (4.21)$$

to be satisfied by selection of proper values for α_k and ${}^nt_f$.

Since only m initial conditions for the solution ${}_np^1$ are specified, the remaining S missing initial conditions are taken to be the best available estimates for these values. For n = 1, the missing initial conditions are actually estimates, but for n = 2,3,4,..., the initial conditions are provided by the algorithm in the manner described previously for ${}^nz(t_o)$ (eqs. (4.14) and (4.16)). Initial conditions for the ${}_{n}p^{k}$, $k = 2,3,\ldots(S+1)$ are determined from ${}_{n}p^{1}$ according to the scheme of equation (3.4),

$$\left.\begin{array}{l} {}_{n}p_{i}^{k}\left(t_{o}\right) = \beta_{ik}\ {}_{n}p^{1}\left(t_{o}\right) + \gamma_{ik} \qquad \begin{array}{l} i = 1,2,\ldots N \\ k = 2,3,\ldots(S+1) \end{array} \\ \text{where} \\ \beta_{ik} = \begin{cases} 1 & \text{if } i \neq k+m-1 \\ \beta_{i} & \text{if } i = k+m-1 \end{cases} \\ \gamma_{ik} = \begin{cases} 0 & \text{if } i \neq k+m-1 \\ \gamma_{i} & \text{if } i = k+m-1 \text{ and } \left|{}_{n}p_{i}^{1}\left(t_{o}\right)\right| < \gamma_{i} \end{cases} \end{array}\right\} \quad (4.22)$$

and β_i and γ_i are perturbation factors prescribed by the user in order to control the magnitude of deviations between the various particular solutions.

At each iteration of the PSPM, $S+1$ vector differential equations (4.20) are integrated from t_o to the best estimate for t_f, and the Newton-Raphson algorithm, equation (3.7), is used to determine values of α_k and t_f which satisfy the boundary conditions (4.21). However, in order to efficiently incorporate this algorithm into the PSPM, the following observations are made. Each iteration of the Newton-Raphson algorithm yields estimates of the superposition constants from which an estimate of the initial conditions

$$^{n}y\left(t_{o}\right) = \sum_{k=1}^{S+1} \alpha_{k}\ {}_{n}p^{k}\left(t_{o}\right)$$

can be made. This estimate can be used to compute an estimate for the change in the initial conditions of the nonlinear system, ${}^{n}\Delta z(t_o)$, where

$$^{n}\Delta z\left(t_o\right) = {}^{n+1}z\left(t_o\right) - {}^{n}z\left(t_o\right) = {}^{n}y\left(t_o\right) - {}^{n}z\left(t_o\right) = \sum_{k=1}^{S+1} \alpha_k \, {}_{n}p^{k}\left(t_o\right) - {}_{n}p^{1}\left(t_o\right)$$

Using equation (4.22) and simplifying, the estimated change in initial conditions can be expressed as a function of the α_k and perturbation factors

$$^{n}\Delta z_i\left(t_o\right) = 0 \qquad i = 1,2,\ldots m$$

$$^{n}\Delta z_i\left(t_o\right) = \alpha_k\left[{}_{n}p_i^{1}\left(t_o\right)\left(\beta_i - 1\right) + \gamma_{ik}\right] \qquad \begin{array}{l} i = m + 1, m + 2, \ldots N \\ k = i - m + 1 \end{array}$$

A suitable norm for this estimated change $\rho[{}^{n}\Delta z(t_o)]$ can be computed and compared to the maximum allowable norm for this change (the value δ appearing in eq. (4.13)). If

$$\rho\left[{}^{n}\Delta z\left(t_o\right)\right] > \delta$$

then additional iterations of the Newton-Raphson algorithm are not useful since this would only serve to compute ${}^{n}\Delta z(t_o)$ to greater accuracy, with the subsequent application of the convergence modification (4.14) wasting this effort. Therefore, in this situation, only one Newton-Raphson iteration should be made. When the norm of ${}^{n}\Delta z(t_o)$ is less than δ, then an indication that the PSPM is in the terminal stages of convergence is obtained, and continued iterations of the Newton-Raphson algorithm can be expected to yield better estimates of the unknown initial conditions and final time. The effect of the number of Newton-Raphson iterations allowed for the case when $\rho[{}^{n}\Delta z(t_o)]$ is less than δ is a subject of investigation in the following chapter.

When the final Newton-Raphson iteration is made on each reference solution, and the subsequent estimate of ${}^{n}\Delta z(t_o)$ is obtained, the modification (4.16) is applied to yield a final value for ${}^{n}\Delta z(t_o)$. Initial conditions for the next reference solution are then computed from

$$_{n+1}p^{1}\left(t_o\right) = {}_{n}p^{1}\left(t_o\right) + {}^{n}\Delta z\left(t_o\right)$$

In this manner, if convergence occurs, the initial conditions ${}_{n}p^{1}(t_o)$ converge to the proper initial conditions of the desired nonlinear solution. Since ${}^{n}y(t)$ also converges to the desired nonlinear solution, the following result is obtained at convergence

$${}_{n}p^{1}\left(t_o\right) = {}^{n}z\left(t_o\right) = {}^{n}y\left(t_o\right) = \alpha_1 \; {}_{n}p^{1}\left(t_o\right) + \sum_{k=2}^{S+1} \alpha_k \; {}_{n}p^{k}\left(t_o\right)$$

This condition is satisfied if $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = \ldots = \alpha_{S+1} = 0$. Besides offering a simple and positive test for convergence of the PSPM, the above mentioned final converged values for the α_k provide reasonable estimates for these values which are required by the Newton-Raphson algorithm. These estimates become increasingly more accurate as the PSPM converges.

In the next chapter, the convergence characteristics of the PSPM are investigated and compared with published results for other Perturbation and Quasilinearization methods. The effects on convergence by the various modifications are investigated separately in order to evaluate the effectiveness of each.

CHAPTER V

DISCUSSION OF RESULTS

In this chapter, the convergence characteristics of the Particular Solution Perturbation Method (PSPM) are investigated on two typical nonlinear boundary value problems which result from the formulation of an optimal control problem for solution by an indirect method. The problems are formulated from the same basic optimal control problem and differ only in the boundary conditions which are imposed. The basic control problem considered is the determination of the thrust vector control for a minimum time, planar, Earth-Mars, orbit transfer for a spacecraft with a continuously firing, low-thrust rocket engine. This problem was selected because it has been used to test several other optimization methods, and consequently considerable data were available from which a direct comparison of results could be made.

The equations of motion for the thrusting rocket are formulated in heliocentric, polar coordinates where only the gravitational attraction of the Sun is considered (Fig. 1). In addition, it is assumed that the thrust vector of the rocket can be turned continuously and effortlessly so that the spacecraft is idealized as a point mass with negligible rotational dynamics. The nonlinear ordinary differential equations to

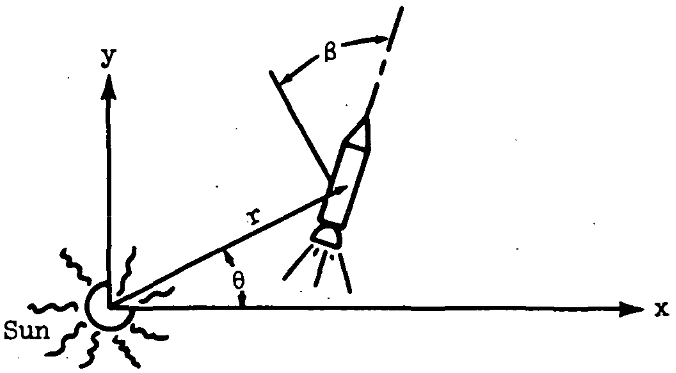


Figure 1.- Coordinate system.

be solved for the determination of minimum time transfer trajectories are derived in Appendix A and include the spacecraft equations of motion

$$\dot{Z}_1 = \dot{u} = \frac{v^2}{r} - \frac{GM}{r^2} + \frac{T}{m} \sin \beta = f_1$$

$$\dot{Z}_2 = \dot{v} = \frac{-uv}{r} + \frac{T}{m} \cos \beta = f_2$$

$$\dot{Z}_3 = \dot{r} = u = f_3$$

$$\dot{Z}_4 = \dot{\theta} = \frac{v}{r} = f_4$$

$$\dot{Z}_5 = \dot{m} = -c = f_5$$

and the associated Euler-Lagrange differential equations

$$\dot{Z}_6 = \dot{\lambda}_1 = \left(\frac{v}{r}\right)\lambda_2 - \lambda_3 = f_6$$

$$\dot{Z}_7 = \dot{\lambda}_2 = -\left(\frac{2v}{r}\right)\lambda_1 + \left(\frac{u}{r}\right)\lambda_2 - \left(\frac{1}{r}\right)\lambda_4 = f_7$$

$$\dot{Z}_8 = \dot{\lambda}_3 = \left(\frac{v^2}{r^2} - \frac{2GM}{r^3}\right)\lambda_1 - \left(\frac{uv}{r^2}\right)\lambda_2 + \left(\frac{v}{r^2}\right)\lambda_4 = f_8$$

$$\dot{Z}_9 = \dot{\lambda}_4 = 0 = f_9$$

where T is the constant thrust of the rocket, c is the constant mass flow rate of the exhaust, GM is the gravitational constant of the Sun, $\sin\beta = -\lambda_1/\sqrt{\lambda_1{}^2 + \lambda_2{}^2}$ and $\cos\beta = -\lambda_2/\sqrt{\lambda_1{}^2 + \lambda_2{}^2}$.

Example Problem 1

For the first example problem considered, it is required only that the spacecraft reach an assumed circular Mars orbit with zero radial velocity and tangential velocity equal to that of Mars. The final angle θ is not specified. The known initial conditions are the position, velocity, and mass of the spacecraft as it leaves an assumed circular Earth orbit; the normalized value of one Lagrange multiplier; and a known zero value for the constant λ_4, which results from not specifying a value for $\theta(t_f)$;

$$Z_1(t_o) = u(t_o) = 0$$

$$Z_2(t_o) = v(t_o) = 1$$

$$Z_3(t_o) = r(t_o) = 1$$

$$Z_4(t_o) = \theta(t_o) = 0$$

$$Z_5(t_o) = m(t_o) = 1$$

$$Z_8(t_o) = \lambda_3(t_o) = -1$$

$$Z_9(t_o) = \lambda_4 = 0$$

The normalization of the Lagrange multipliers and other system parameters is discussed in Appendix A. The terminal boundary conditions at the unknown final time are

$$h_1\left[Z(t_f),t_f\right] = Z_1(t_f) - 0 = 0$$

$$h_2\left[Z(t_f),t_f\right] = Z_2(t_f) - 0.81012728 = 0$$

$$h_3\left[Z(t_f),t_f\right] = Z_3(t_f) - 1.5236790 = 0$$

For this problem, the dimension of the vector Z is $N = 9$, with $m = 7$ specified initial conditions and $S + 1 = N - m + 1 = 3$ terminal conditions given since final time is unknown. The unspecified initial conditions are

$$Z_6(t_o) = \lambda_1(t_o)$$

$$Z_7(t_o) = \lambda_2(t_o)$$

Example Problem 2

For the second example problem, it is required that the final spacecraft central angle $\theta(t_f)$ be equal to the central angle of Mars at the time of rendezvous. The central angle of Mars at the end of the transfer trajectory is computed from a known central angle of the planet at the beginning of the transfer, the constant angular velocity of the Mars about the Sun, and the time of flight,

$$\theta\left(t_f\right) = \theta_M\left(t_o\right) + \frac{v_M}{r_M} t_f$$

Thus, for this problem an additional terminal boundary condition is added to the set given for example problem 1,

$$h_4\left[Z\left(t_f\right),t_f\right] = Z_3\left(t_f\right) - \theta_M\left(t_o\right) - \frac{v_M}{r_M} t_f = 0$$

Since, in this case, the terminal value of θ is constrained, λ_4 cannot be determined to be zero, and the initial and constant value for λ_4 is unknown. Therefore, for this example problem there are three unspecified initial conditions: $\lambda_1(t_o)$, $\lambda_2(t_o)$, and $\lambda_4(t_o)$.

Numerical Results for Example Problem 1

The solution of example problem 1 provided correct initial multiplier values and final time as follows:

$$\lambda_1\left(t_o\right) = -0.494865$$

$$\lambda_2\left(t_o\right) = -1.07855$$

$$t_f = 3.319437 \text{ (1 time unit = 53.132355 days)}$$

These values were obtained using a relative error bound of 10^{-5} with the power series integration scheme of Appendix B. Convergence of the PSPM was detected by requiring that the sum of the absolute values of the superposition constants α_2 and α_3 be less than 1×10^{-4}. For this problem, this convergence criterion was more demanding than requiring that the initial condition and final time changes be less than the specified convergence tolerance, since it was observed that changes in these values were about one order of magnitude smaller than values of α_2 and α_3. All computations were made in single precision arithmetic (eight significant figures) on the Univac 1108 digital computer. Each iteration of the PSPM required approximately 2 seconds of computer time.

In order to evaluate the convergence sensitivity of the PSPM to initial guess values for λ_1, λ_2, and t_f, the problem was solved many times using starting guesses which deviated from the true values by known percentages. The deviations from the true values were chosen in a systematic manner so that the data could be presented in the form of convergence envelopes. The convergence envelope shown in Figure 2 was constructed from all initial guess data having a final time error of -20 percent (a guessed final time less than the actual final time). The convergence envelope was formed by locating the percentage deviations used for the initial guess values of the two Lagrange multipliers on a Cartesian coordinate grid. Each problem attempted was located on

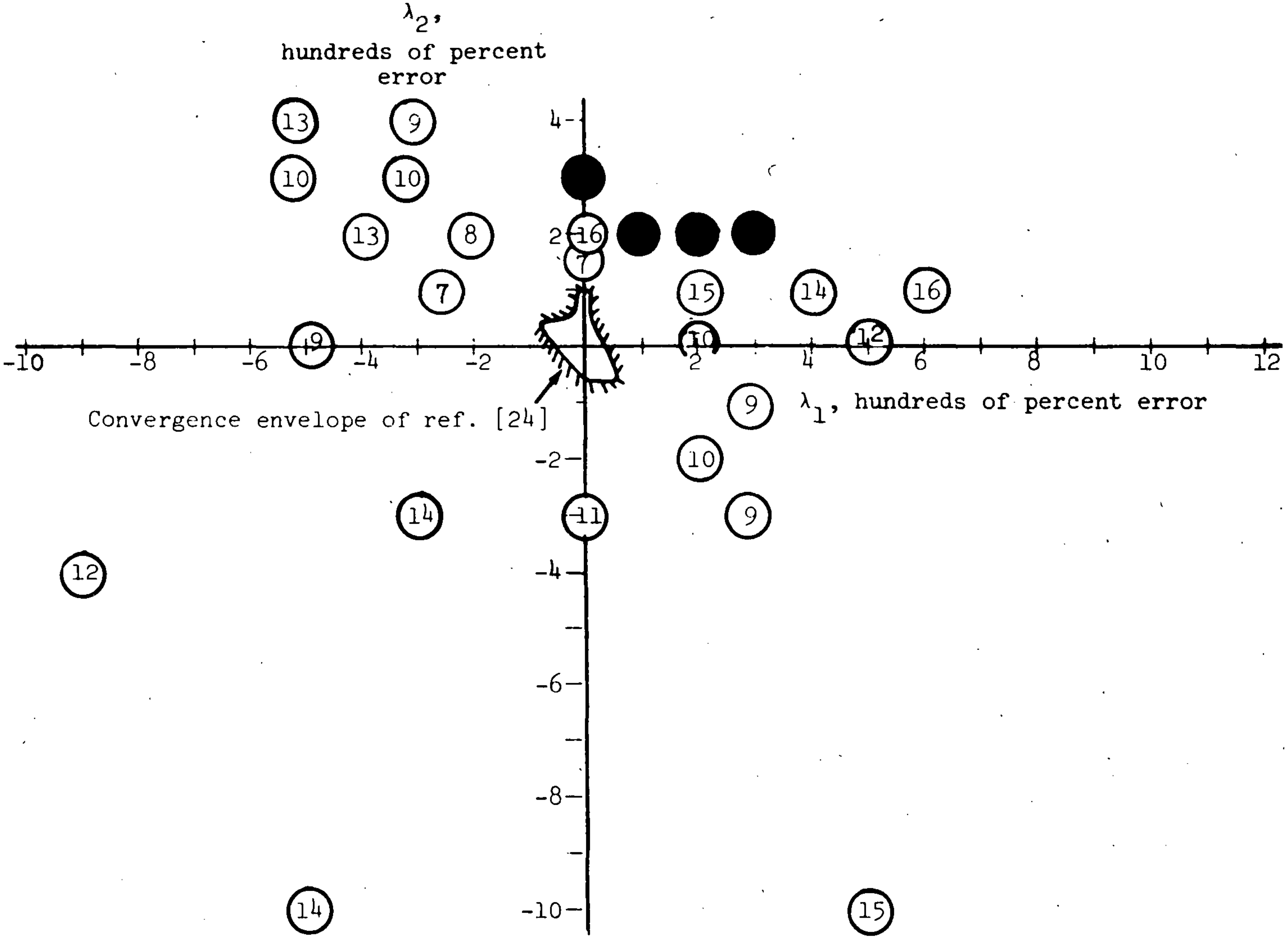


Figure 2.- Convergence envelope for -20 percent final time error.

the grid by a small circle. Darkened circles represent initial guess values which did not lead to convergence in 20 iterations of the PSPM. Open circles containing numbers represent initial guess values for the multipliers which did lead to convergence, and the numbers in the circles represent the number of iterations required. On divergent trials, typical behavior of the method was to successively select multiplier changes in the wrong direction on each iteration. Also shown in Figure 2 is the boundary of a convergence envelope obtained for this problem by Lewallen [24], who investigated and compared several trajectory optimization methods. In reference [24], similar sized convergence boundaries were presented for three methods; the Method of Adjoint Functions studied by Jazwinski [22]; the Method of Perturbation Functions discussed by Breakwell, Speyer, and Bryson [23]; and Lewallen's [35] Modified Quasilinearization Method. These methods typically required 11 to 20 iterations for initial multiplier errors along the outer edge of the convergence boundary shown. The superior convergence characteristics of the PSPM are evident.

Presented in Figures 3 and 4 are similar convergence envelopes for cases with 0 percent and +20 percent deviations in initial guesses for final time, respectively. Also shown in these figures are typical convergence envelopes presented in references [24] and [42] on the same problem with the three methods mentioned previously. The superior convergence characteristics of the PSPM are again indicated by these data.

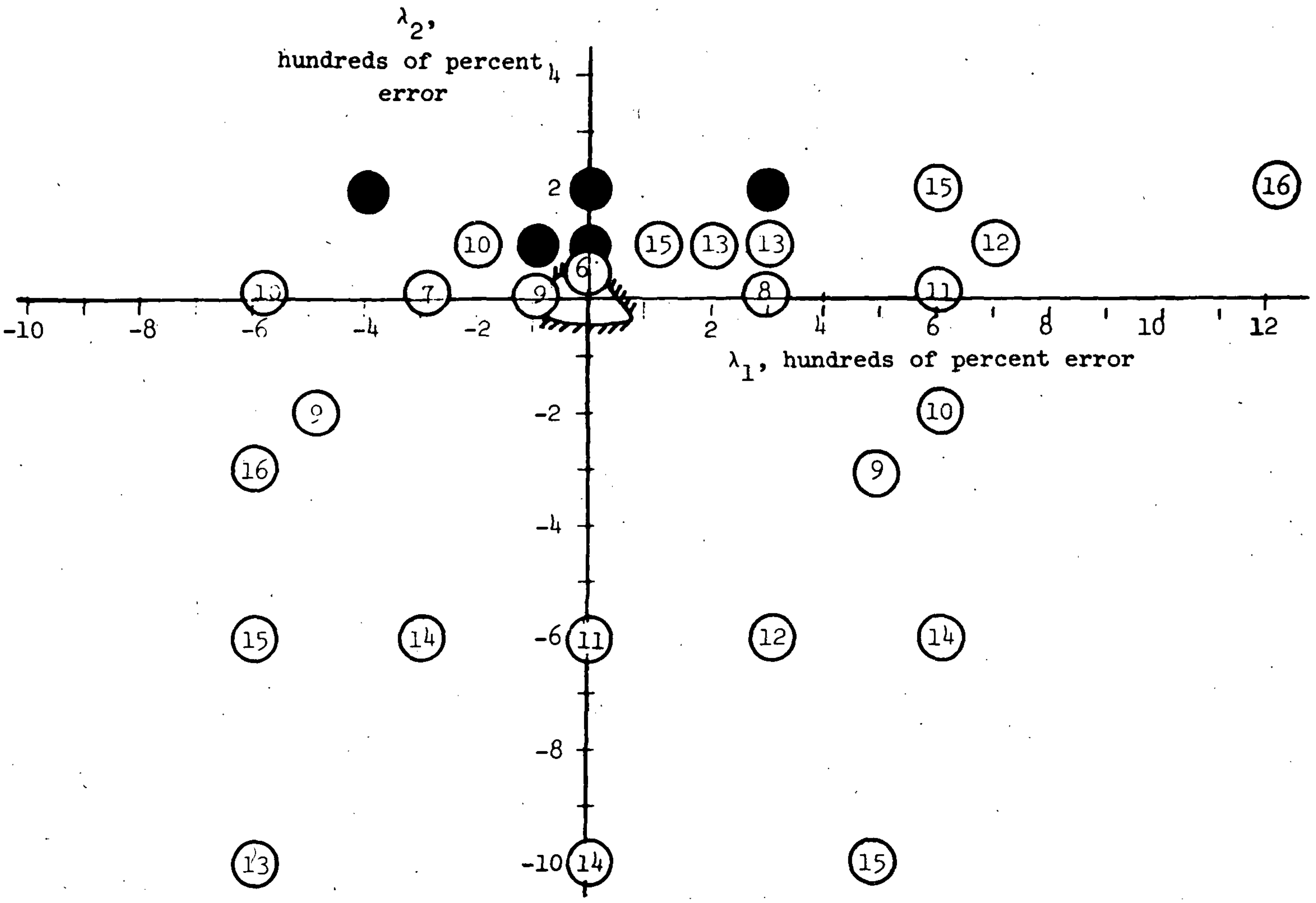


Figure 3.- Convergence envelope for 0 percent final time error.

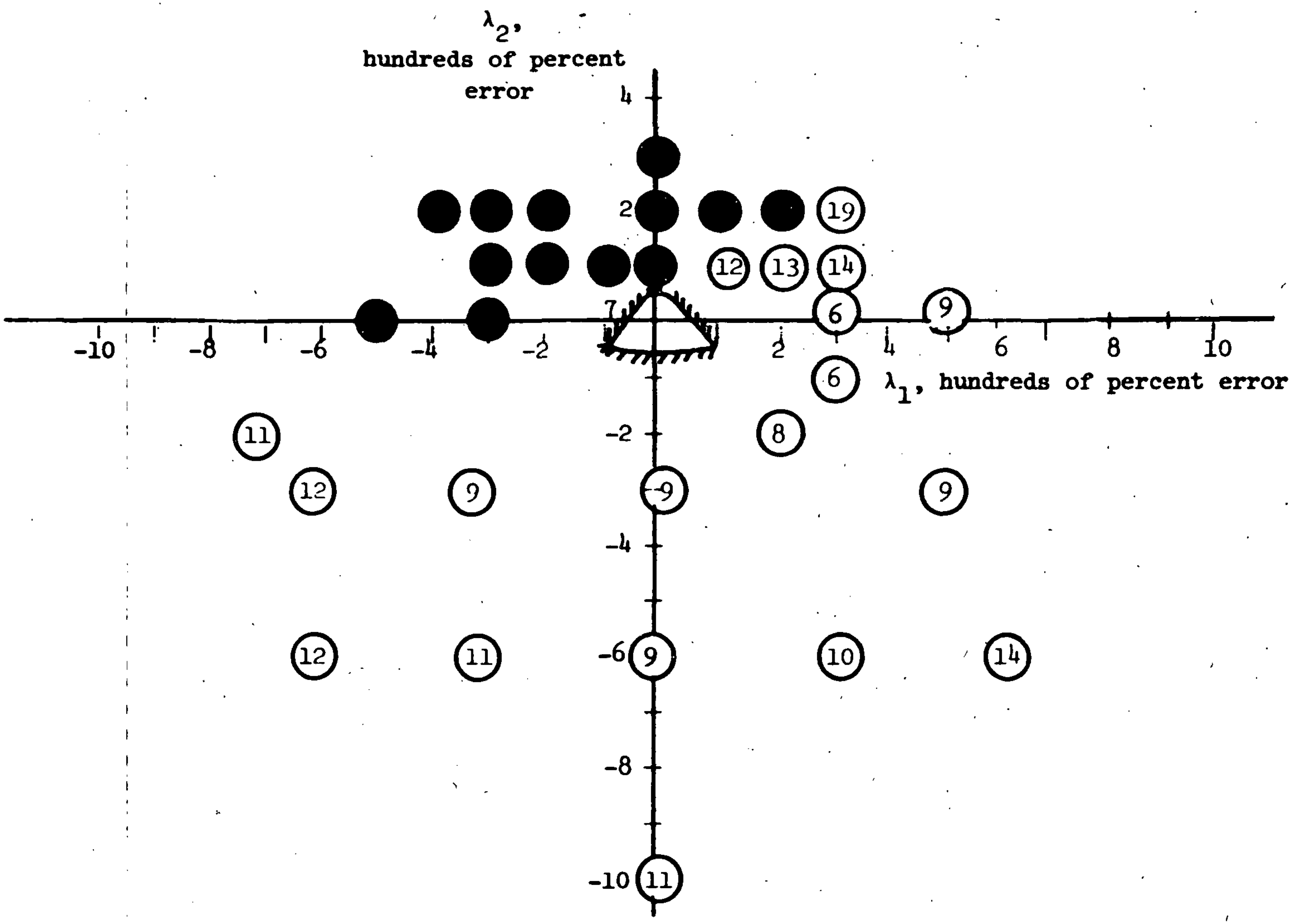


Figure 4.- Convergence envelope for +20 percent final time error.

The most probable reasons for this marked difference in convergence characteristics of the PSPM and the other methods (which are quite similar methods) are discussed below.

For the data presented in Figures 2, 3, and 4, the value used for the maximum allowable initial condition step size norm, δ of equation (4.13), was 0.5. Each time that the requested step size norm was less than δ, the value of δ was set equal to the norm of the requested change. For significant errors in the initial values of the Lagrange multipliers, typical values for $\rho[{}^{n}\Delta Z(t_o)]$, the norm of the requested change in $\lambda_1(t_o)$, $\lambda_2(t_o)$, and t_f varied between 8 and 5000. This means that for some cases, the value of ε used in the convergence modification of equation (4.14) was on the order of 1×10^{-4} and the requested changes in $\lambda_1(t_o)$, $\lambda_2(t_o)$, and t_f were reduced by this fraction. In comparison, the various methods studied in references [24] and [42] were implemented with a fractional correction scheme which had the essential effect of halving the computed initial condition changes and final time changes on the first few iterations. With this scheme, for multiplier errors below the indicated boundary in Figures 2, 3, and 4, the first iteration yielded multipliers in the upper part of the envelopes. The PSPM also diverges in these upper regions due to subsequent multiplier changes being selected in the "wrong direction." However, for multiplier guesses in the lower half of the envelopes, the fractional correction computed for the PSPM was sufficiently small to prevent "stepping over" the solution. Had the PSPM been implemented with the fractional correction scheme of references [24] and [42], the PSPM convergence characteristics would have been similar to the convergence characteristics of the methods studied in these references. However, not all of the desirable convergence characteristics of the PSPM can be attributed to this one item alone. It is expected that many cases which converged with the PSPM would not have if the convergence modification (4.16) had not been used.

A vivid illustration of the importance of the modification (4.16) is presented by the data of Table I. These data represent the values of the Lagrange multipliers and final time estimates obtained on successive iterations with and without the convergence modification (4.16). The initial guesses correspond to multiplier errors of 0 percent and -50 percent with a terminal time error of -20 percent. The PSPM will never converge from these initial guesses without the modification, and with it convergence is obtained in eight iterations. Of the eight iterations, only three required application of the modification as indicated by the (H) symbol in the table for values affected by the halving feature.

Another feature of the PSPM which contributed in part to the good convergence performance shown in Figures 2, 3, and 4 was the use of upper and lower bounds for t_f. Upper and lower bounds of 4.4 and 2.2 were specified, and although these bounds were rarely approached, they were imposed in several instances. Since bounds on the Lagrange multipliers $\lambda_1(t_o)$ and $\lambda_2(t_o)$ were not easily determined, no upper and lower bounds for their values were specified in this study.

For those starting guesses indicated in Figures 2, 3, and 4 which did not lead to convergence of the PSPM, the typical behavior of the

TABLE I

COMPARISON OF PSPM WITH AND WITHOUT CONVERGENCE MODIFICATION (4.16)

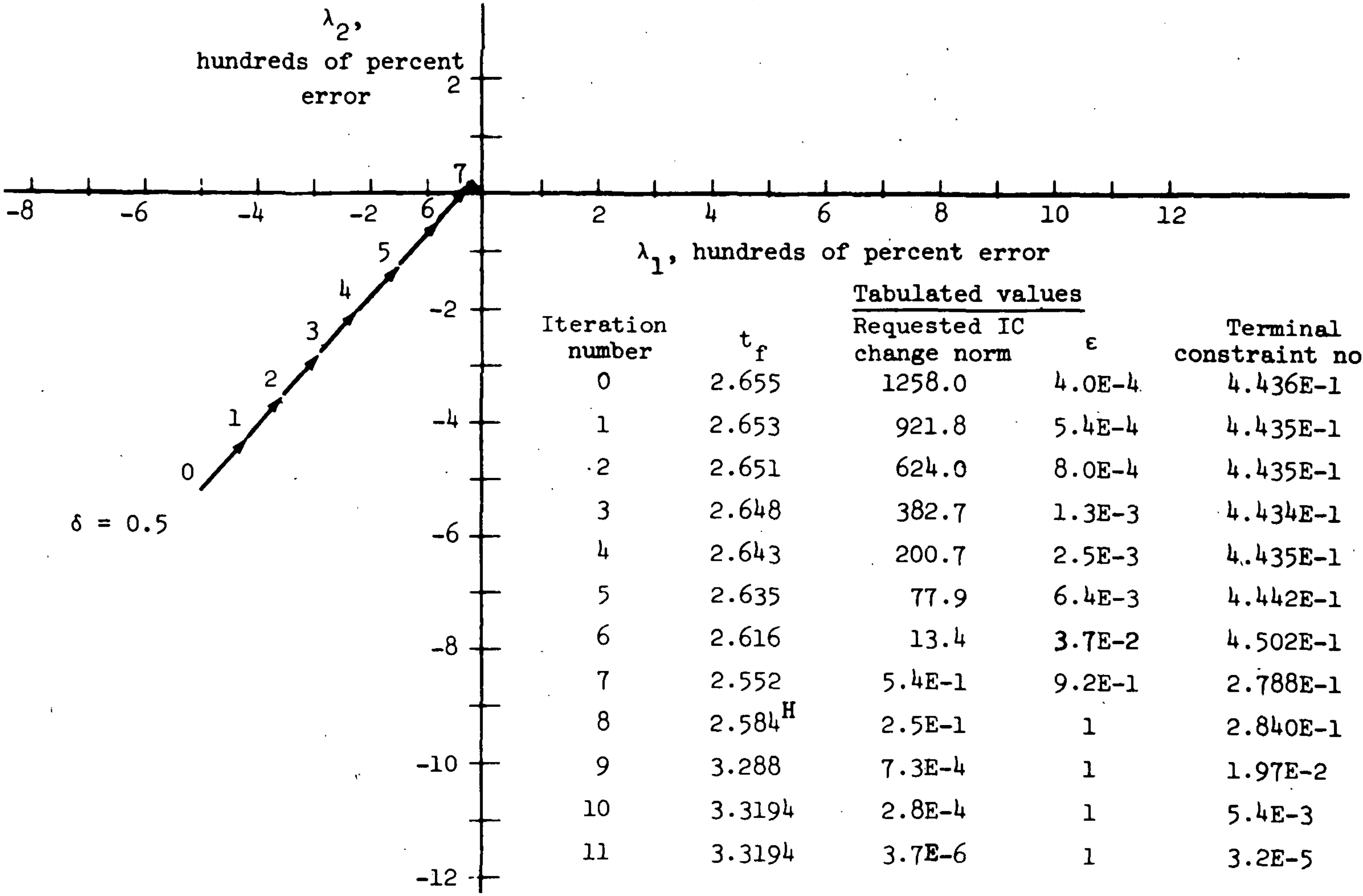
Iteration number	Results with modification (4.16)				Results without modification (4.16)			
	λ_1	λ_2	t_f	$\rho\left[{}^{n}\Delta Z(t_o)\right]$	λ_1	λ_2	t_f	$\rho\left[{}^{n}\Delta Z(t_o)\right]$
0	-0.4948	-1.62	2.655	8.3	-0.4948	-1.62	2.655	8.3
1	-0.4722	-0.7565	2.592	4.4E-1	-0.4722	-0.7565	2.592	4.4E-1
2	-0.4835^H	-1.188^H	2.624^H	1.8E-1	-0.7451	-1.44	2.74	3.1
3	-0.481	-1.086	3.165	1.1E-2	-0.455	-0.7597	2.594	4.6E-1
4	-0.4822^H	-1.078	3.320	4.0E-3	-0.841	-1.37	2.807	2.3
5	-0.4888	-1.079	3.318	3.3E-3	-0.461	-0.752	2.604	4.6E-1
6	-0.4944	-1.0786^H	3.3191^H	3.2E-4	-0.824	-1.38	2.82	2.3
7	-0.49486	-1.0785	3.31943	3.5E-6	-0.456	-0.752	2.61	4.7E-1
8	-0.49486	-1.0785	3.31943	3.1E-7	-0.825	-1.39	2.79	2.5
					etc.	etc.	etc.	etc.

(H) indicates value was obtained from the halving feature of the modification.

1.0E-1 indicates 1.0 × 10^{-1}.

PSPM was to select both multiplier initial value changes to be in the wrong direction on each iteration. Usually after 20 iterations the magnitude of the initial values for $\lambda_1(t_o)$ and $\lambda_2(t_o)$ were so large, that the effects of λ_3 on the solution of the Euler-Lagrange equations were negligible. Consequently, although increasingly larger values were obtained for $\lambda_1(t_o)$ and $\lambda_2(t_o)$, their ratio remained almost constant and each successive reference trajectory was an essential repeat of the previous reference trajectory. With this type of behavior, it was apparent that the PSPM would not converge in any number of iterations for the particular choice of initial Lagrange multipliers. Any initial guesses for $\lambda_1(t_o)$ and $\lambda_2(t_o)$ which were large in magnitude compared to $\lambda_3(t_o) = -1$, caused the PSPM to generate very similar reference trajectories on the first several iterations. However, in most cases, the multiplier changes on these first few iterations were made in the proper direction, and convergence resulted. This behavior suggests that the convergence space of the PSPM is boundless in the lower quadrants of the envelopes of Figures 2, 3, and 4 when a value of δ is used which will prevent the method from "stepping over" the solution and selecting values in the upper quadrants of the convergence envelopes.

The operation of the PSPM is illustrated graphically in Figure 5. Each arrow represents the change in the values of $\lambda_1(t_o)$ and $\lambda_2(t_o)$ taken on each iteration. Also presented in tabulated form is the value of t_f at each iteration, the requested step size norm, the fractional reduction factor ε, and the value of the terminal constraint norm obtained with the reference solution of each iteration. The initial



		Tabulated values		
Iteration number	t_f	Requested IC change norm	ε	Terminal constraint norm
0	2.655	1258.0	4.0E-4	4.436E-1
1	2.653	921.8	5.4E-4	4.435E-1
2	2.651	624.0	8.0E-4	4.435E-1
3	2.648	382.7	1.3E-3	4.434E-1
4	2.643	200.7	2.5E-3	4.435E-1
5	2.635	77.9	6.4E-3	4.442E-1
6	2.616	13.4	3.7E-2	4.502E-1
7	2.552	5.4E-1	9.2E-1	2.788E-1
8	2.584^H	2.5E-1	1	2.840E-1
9	3.288	7.3E-4	1	1.97E-2
10	3.3194	2.8E-4	1	5.4E-3
11	3.3194	3.7E-6	1	3.2E-5

Figure 5.- Successive iterates of the PSPM.

value of δ was taken to be 0.5, and δ was set equal to the norm of the requested change whenever this norm was less than δ. Figure 5 illustrates that the proper direction for the multiplier change is chosen at each iteration. Similar results can be expected for much larger errors in the lower left quadrant of the convergence envelope. It is interesting to note the behavior of the terminal constraint norm of the reference solution at each iteration for the problem presented in Figure 5. Although the PSPM takes each step in the proper direction, the successive values of the terminal constraint norm initially decrease very slightly and actually increase for the fourth, fifth, and sixth iterations. This behavior suggests that fractional correction procedures utilizing the value of a terminal constraint norm of the reference trajectory may not work well on this problem.

An Improved Method for Choosing δ

Perhaps the most appropriate criticism of the PSPM as presented is the necessity for choosing a value for δ, the maximum initial condition change norm. If δ is chosen too large, the method may "step over" the solution into the divergent region. On the other hand, if δ is chosen too small, the convergence of the method is unduly retarded. For example, if $\delta = 0.25$ had been selected for the problem presented in Figure 5, it would have taken 14 iterations to arrive at the same multiplier values obtained in seven iterations when δ was chosen to be 0.5. In order to illustrate the sensitivity (or insensitivity) of the PSPM to the value chosen for δ, the problem of Figure 5 was solved using δ values of 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, and 2.0. The results of this investigation are presented in Table II.

TABLE II

NUMBER OF ITERATIONS REQUIRED FOR VARIOUS VALUES OF δ

δ	No. iterations required	δ	No. iterations required
0.25	20	1.25	13
0.5	12	1.5	Diverge
0.75	14	1.75	10
1.0	19	2.0	Diverge

These results indicate that for $\delta = 1.5$, the allowable change in initial conditions was large enough to allow the method to "step over" the solution. The convergence with $\delta = 1.75$ was coincidental since the second iteration produced the same multipliers as the 7th iteration of the case when $\delta = 0.5$.

The behavior of the PSPM shown in Figure 5 suggests an approach for making the selection of δ a self-adapting feature of the method. When each successive initial condition change vector is taken in the same direction as the previous change vector, an indication that δ can be increased is obtained. This behavior can be detected by forming the dot product of successive initial condition (and final time) change vectors and computing the cosine of the angle between successive change vectors. When this angle is near zero, successive Lagrange multiplier and final time values lie very near a line connecting the initially assumed values for these parameters and values of these parameters near the true values (as indicated in Fig. 5). By referring to the table of Figure 5, one may plot the requested step size norm as a function of distance moved along this "line" which we will designate as the "convergence path." The data of the table in Figure 5 are plotted in this manner in Figure 6. At convergence, the requested change is zero. An estimate of the distance to move along the convergence path in order to obtain multipliers and final time which yield a zero change (converged values) is obtained by estimating the point of intersection of the curve and the horizontal axis in Figure 6. The slope of this curve can be computed numerically by evaluating the successive requested norms and keeping track of the distance moved along the convergence path on successive iterations. Graphically, the estimated distance to move along the convergence path using the self-adapting approach is shown by the intersection of the dashed lines and the horizontal axis in Figure 6. To implement this self-adapting approach, the initial iteration is made with any small value for δ, ($\delta_1 = 0.5$ in Fig. 6). If the change vector of the second iteration is in the approximate same direction as the first iteration, then the distance to move along the convergence path is found by

$$\delta_2 = \frac{(\text{Requested norm 2})(\delta_1)}{(\text{Requested norm 1}) - (\text{Requested norm 2})}$$

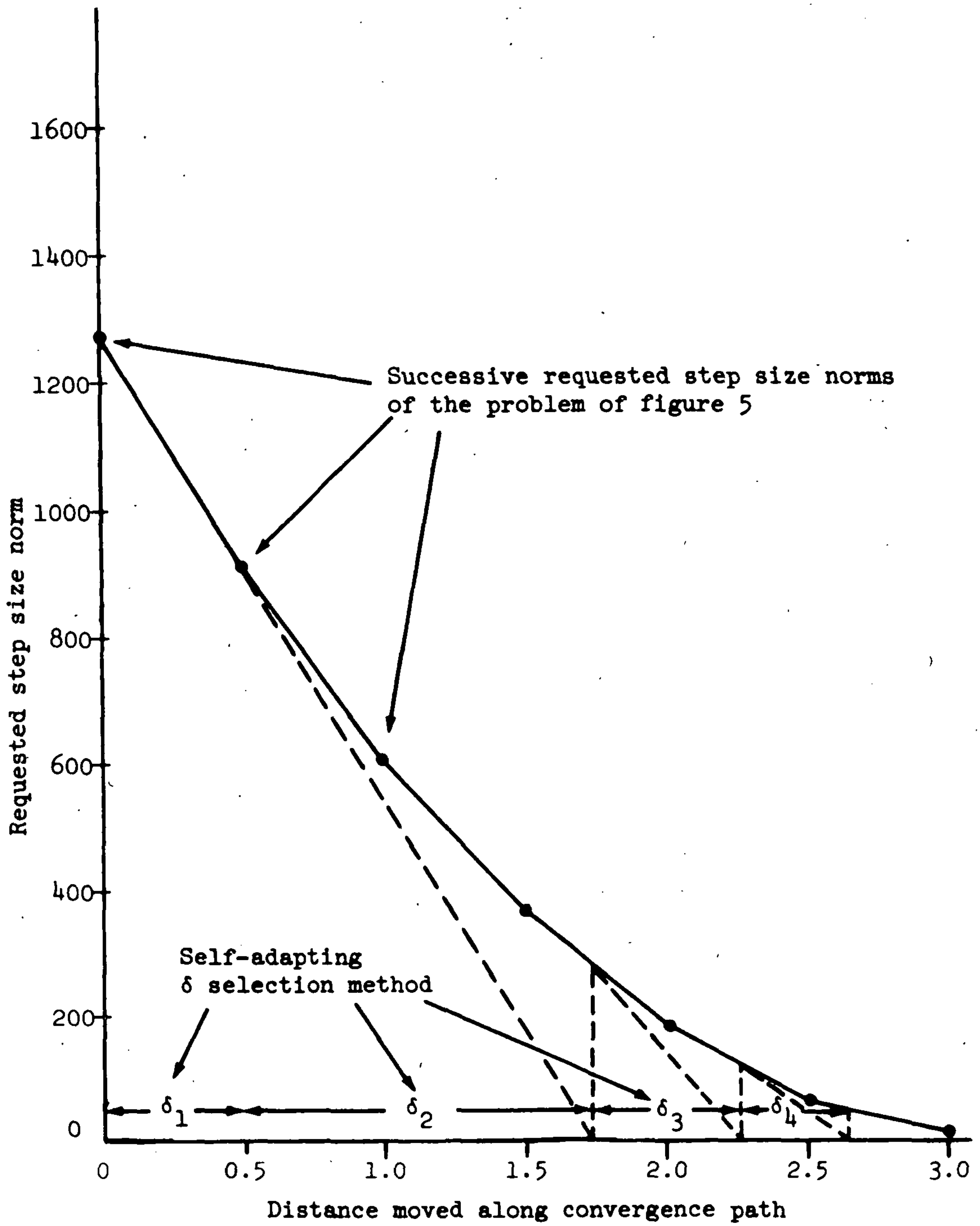


Figure 6.- Behavior of requested step size norm along convergence path.

Successive values for δ at each iteration are chosen in this manner until successive change vectors are not approximately along the same line or until the computed value for δ is less than the initial specified value δ_1.

This self-adapting feature was incorporated into the PSPM and studied on a variety of initial multiplier guesses. Typical results with this scheme are illustrated by solving the problem of Figure 5 with $\delta_1 = 0.25$ in 12 iterations instead of the 20 required when $\delta = 0.25$ at each iteration. Using $\delta_1 = 0.25$ with initial final time error of -20 percent and initial Lagrange multiplier errors of -1000 percent for both λ_1 and λ_2, convergence was obtained in 16 iterations. Similarly, with initial multiplier errors of +1000 percent, -1000 percent, and final time error of -20 percent, convergence was obtained in 14 iterations.

Experience with this scheme is limited at the present time and undoubtedly its effectiveness is somewhat problem dependent. For example, if the curve of Figure 6 were concave instead of convex, the scheme may cause δ to be chosen too large. In such cases it may be necessary to restrict the maximum value that δ can attain. That is, the method would be allowed to be self-adapting within a range of values between δ_1 and some δ_{max}. Further investigations of this scheme on various problems are recommended in order to evaluate its effectiveness as a general approach.

Numerical Investigations With Distinctive Features of the PSPM

Besides the convergence modifications, there are several distinctive features of the PSPM that may cause it to operate differently from other Perturbation and Quasilinearization methods. One of these features is the capability for forcing the solution of the linearized equations to satisfy given terminal constraint functions to a specified accuracy at each iteration. This capability was used only in the terminal stages of convergence for the results presented and had no pronounced effect on whether convergence was actually obtained. It was found that the most optimum use of the capability was to restrict the PSPM to use only one Newton-Raphson iteration when the initial condition step size norm was greater than δ, and to use no more than two to four Newton-Raphson iterations when this norm was less than δ. By restricting the PSPM to use only one Newton-Raphson iteration at all PSPM iterations, the method was operated in a fashion very similar to the Perturbation methods discussed by Lastman [26] and Lewallen [24]. The primary difference in the normal PSPM operation and the restricted operation was that one to three total trajectory iterations were saved in the normal operation mode at the expense of one to five extra Newton-Raphson iterations. It is believed that the fewer trajectory iterations required resulted from better final time estimation obtained during the last several iterations. A definite savings in computer time was realized since the computer time required for a Newton-Raphson iteration is small compared to the time required for a total trajectory iteration. This savings in computer time averaged about 20 percent for the cases compared. This result was not consistent for all cases and there was some dependence on the initial value of δ, since this affected the point in the terminal convergence phase where extra Newton-Raphson iterations were started.

The generality of the PSPM operation makes it possible to use initial guess values for the α's other than the 1, 0, 0 values discussed previously. An examination of the dh_i/dt_f terms of the Jacobian matrix of equation (3.7) reveals that when the 1, 0, 0 values are chosen, only the reference solution influences these elements of the matrix on the first Newton-Raphson iteration. However, the influence of the perturbed particular solutions can be obtained on the first iteration by assigning "weighting factors" to the various solutions with the initial choice of the α's. A typical choice investigated for example problem 1 was $\alpha_1 = 0.4$, $\alpha_2 = 0.3$, $\alpha_3 = 0.3$, so that the sum of the values totaled to 1 and more "weight" was given to the reference solution ${}_np^1$. During terminal stages of convergence, the initial guess was switched back to $\alpha_1 = 1$, $\alpha_2 = 0$, $\alpha_3 = 0$. The results with this type of operation are inconclusive. In a comparison with the normal α selection procedure on a set of four different cases, this operation produced convergence in fewer iterations for two of the cases and required more iterations for the other two. This unique feature of the PSPM makes it more general than other Perturbation and Quasilinearization methods, and it may be found to be more useful for other problems.

Another distinctive feature of the PSPM investigated was the capability for using different perturbation factors, β_i and γ_i appearing in equation (4.22). Results were compared on various starting vectors using all $\beta_i = 1.2$ in one case and all $\beta_i = 0.5$ in the other. All γ_i were selected to be 0.1. If only one Newton-Raphson iteration at each solution iteration of the PSPM is made with the standard 1, 0, 0 ... guess on the α's, then theoretically the results with different perturbation factors should be identical. However, significant differences were noted due to purely numerical causes. These differences were significant enough to cause the method to require a different number of iterations for convergence when different perturbation factors were used. However, this difference was never more than one or two iterations. The results do point out the importance of minimizing numerical round-off errors in the matrix inversion computations. In this connection, an important advantage results from using the particular solution method for solving the linear system, since the user can exercise control over the numerical values which form the Jacobian matrix in equation (3.7) by selection of appropriate perturbation factors.

Results With Example Problem 2

The second example problem, having a higher dimensionality and more complex terminal boundary conditions, would appear to be a more difficult problem to solve than the first example problem considered. However, once the first example problem is solved, the difficulty of guessing Lagrange multipliers for the second example problem is greatly reduced. The family of problems obtained by considering optimal trajectories for different launch dates is most easily parameterized by the value θ_o, the central angle of Mars at the launch time t_o. For the "open problem" discussed previously, the Lagrange multipliers and final time for $\lambda_3(t_o) = -1$ were found to be

$$\lambda_1(t_o) = -0.494865$$

$$\lambda_2(t_o) = -1.07855$$

$$\lambda_4 = 0$$

$$t_f = 3.319437$$

and the corresponding value of θ_o is easily determined to be $\theta_o = 0.7264$ radian by using the time of flight, the angular velocity of Mars, and the final central angle of the spacecraft in the open problem. To solve the second example problem for any value of θ_o, a succession of problems having initial Mars central angles defined by

$$\theta_{o_i} = \theta_{o_{i-1}} + \Delta\theta_o \qquad i = 1, 2, 3 \ldots$$

where $\Delta\theta_o$ is some small increment, is solved in sequence using the converged values of the previous problem as starting guesses for the next. The process is continued until a solution with the desired value for θ_o is obtained. The convergence characteristics of the PSPM were investigated on this example problem by studying allowable magnitudes for $\Delta\theta_o$.

Using the converged values for the open problem, a solution was first obtained for $\theta_o = 0.8$. Repeated attempts to solve the problem with $\theta_o = 0.9$ using guess values from the $\theta_o = 0.8$ solutions ended in failure. It was decided that $\Delta\theta_o = 0.1$ was too large and $\Delta\theta_o$ was reduced to 0.02. After solving several problems with this increment for $\Delta\theta_o$, difficulties were again encountered. After reducing $\Delta\theta_o$ further to 0.002, the sequence of converged problems shown in Table III was obtained. The data in the table relate θ_o with the corresponding converged values of multipliers and final time. An examination of these data indicates that the PSPM was displaying good convergence characteristics on the boundary value problem but was getting nowhere

TABLE III

CONVERGED MULTIPLIERS AND FINAL TIME FOR

VARIOUS INITIAL MARS LEAD ANGLES

Lead angle θ_o, rad	λ_1	λ_2	λ_4	t_f
0.7264	-0.4948	-1.078	0.000	3.3194
0.800	-0.2321	-1.994	-0.5177	3.3586
0.820	-0.0638	-2.58	-0.8379	3.3776
0.840	0.2434	-3.64	-1.4106	3.3985
0.860	0.9824	-6.1718	-2.7660	3.4208
0.880	5.1917	-20.5144	-10.4111	3.4443
0.882	7.0329	-26.7821	-13.7496	3.4467
0.884	10.5011	-38.5871	-20.0369	3.4491
0.886	19.4653	-69.0958	-36.2848	3.4515
0.888	96.8223	-332.3584	-176.4839	3.4540

with finding a solution to the optimization problem defined by $\theta_o = 0.9$. After plotting the data in Table III versus θ_o, and noting the asymptotic character of the Lagrange multipliers as θ_o approached 0.9, it was realized that all multipliers were seeking large values with respect to the normalized value of -1 for $\lambda_3(t_o)$. This suggested that the unnormalized value of $\lambda_3(t_o)$ approaches zero as θ_o approaches 0.9. Since it was known that λ_4 would not be zero for this problem because of the constrained final central angle $\theta(t_f)$, the multipliers were normalized to $\lambda_4 = -1$. This eliminated the difficulty with convergence.

With the problem normalized to $\lambda_4 = -1$, it was found that a value of $\Delta\theta_o = 0.5$ radian could be used to generate optimal trajectories for $\theta_o = 1.0, 1.5, 2.0, \ldots 6.5, 7.0$ with an average of 11 iterations per problem. In this study, a maximum step size norm, δ, equal to 0.5 was used without the self-adapting feature previously discussed. Optimal trajectories for $\theta_o < 0.7264$ were also obtained. In this case it was necessary to normalize the multipliers to $\lambda_4 = +1$ in order to obtain the proper sign relationships between the multipliers. A plot of converged multipliers and final time as a function of θ_o is given in Figure 7.

The good convergence characteristics of the PSPM were also demonstrated for this example problem by solving the problem for $\theta_o = 3$ using initial guess values for λ_1, λ_2, λ_3, and t_f from the converged values of the problem with $\theta_o = 1$ in 10 iterations. The difference in the initial guess trajectory and the final converged

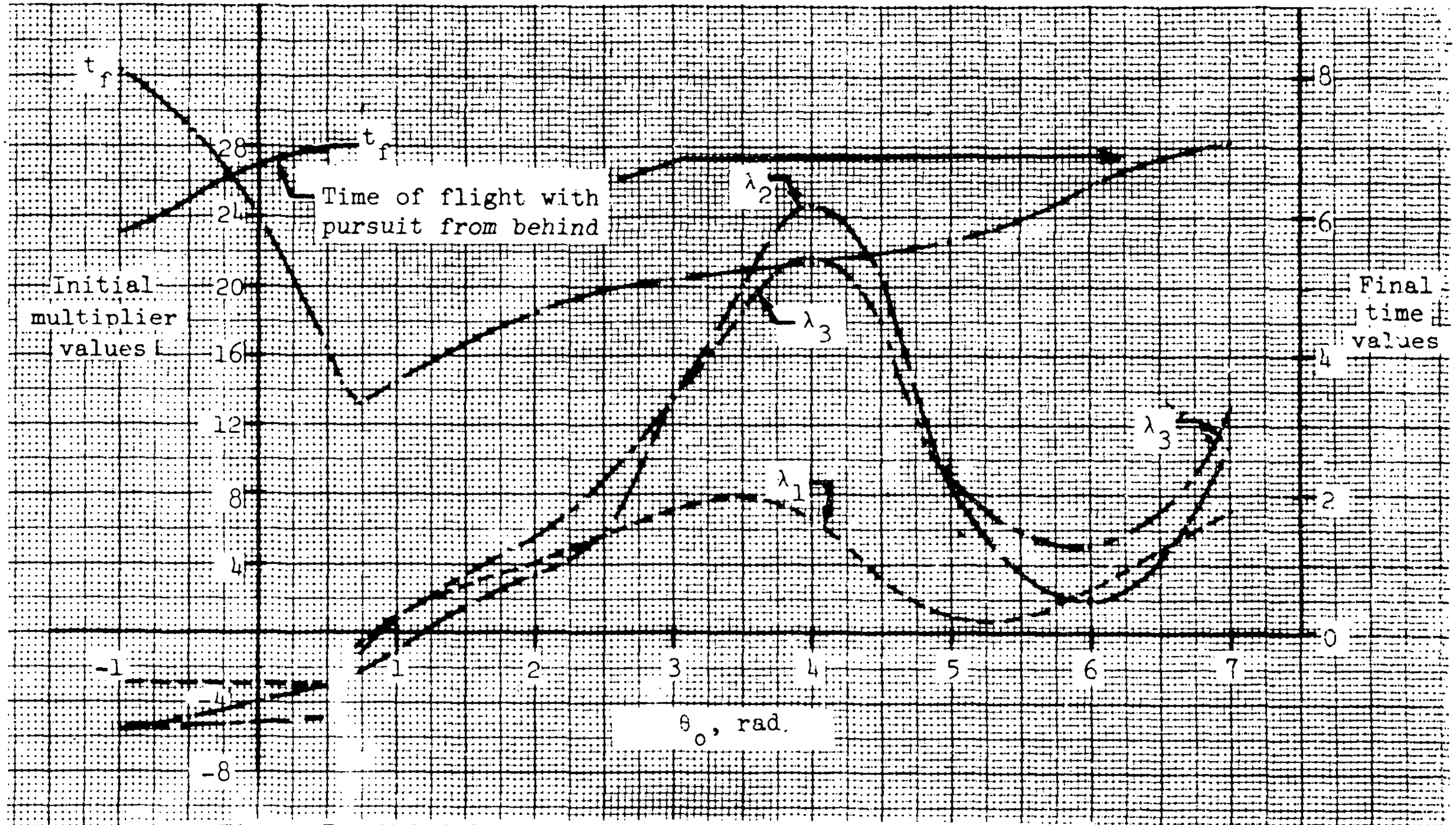


Figure 7.- Multipliers and transfer times as a function of the relative position of Earth and Mars at launch.

trajectory is illustrated in Figure 8, where several of the optimal trajectories for different initial values of θ_o are displayed.

Interesting features evident in Figure 8 are the two distinctive types of optimal trajectories which result from launching before and after the most favorable launch date which corresponds to the open problem ($\theta_o = 0.7264$ radian). This behavior has been previously discussed by Kelley [53], who solved this problem with different numerical values for thrust and initial mass using a direct optimization method. The trajectories corresponding to early ($\theta_o > 0.7264$) launch dates have a "pursuit from behind" character, while the spacecraft when departing from late ($\theta_o < 0.7264$) launch dates tends to "wait" for Mars to overtake it. The severe time-of-flight penalty associated with not launching on the most favorable date is shown in Figure 7. It is also evident from Figure 7 that the "pursuit from behind" type of trajectory has a shorter transfer time than the "waiting" type for most of the unfavorable launch dates.

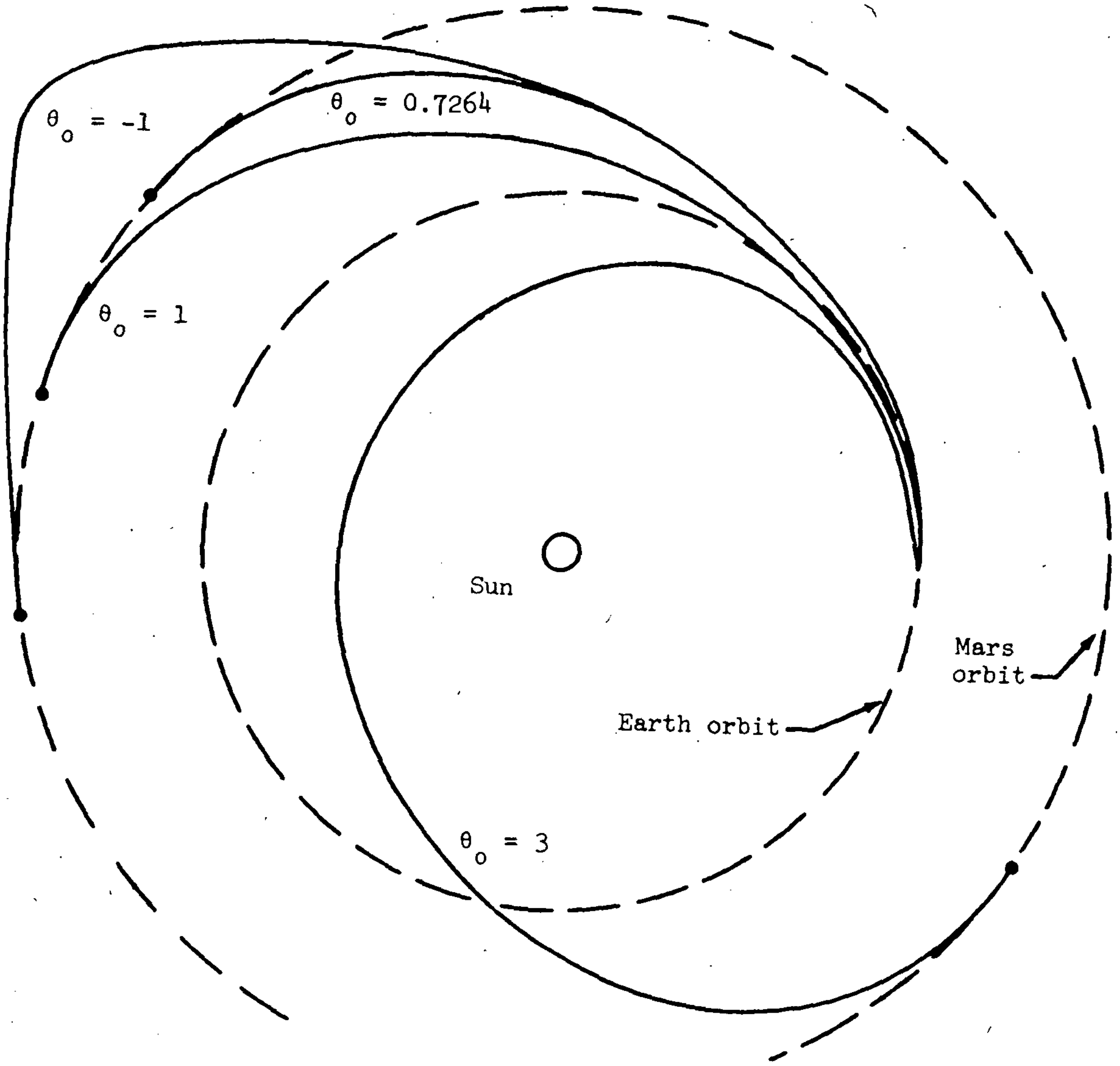


Figure 8.- Optimal transfer trajectories for various relative positions of Earth and Mars at launch.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Several important extensions and modifications to existing indirect optimization methods have been made. The method of particular solutions was extended in order to solve linear boundary value problems with boundary conditions specified in the form of nonlinear functions of the dependent and independent variables. This extension was incorporated into a Perturbation method for solving the nonlinear boundary value problem which results from formulating optimal control problems for solution by an indirect method. This new Perturbation method, called the Particular Solution Perturbation Method (PSPM), reveals a new approach for solving problems with unknown final time which can reduce the number of trajectory iterations required for convergence to the optimal trajectory. The application of this new method for treating unspecified final time problems was simplified by the use of a power series numerical integration method which was ideally suited for the forward and backward variable step integration required. The method is not restricted for use with power series integration, however, and may be implemented with any numerical integration scheme.

The PSPM was found to have excellent convergence characteristics. The range of convergence of the indirect optimization approach was extended far beyond that of previous methods without compromising the rapid convergence of this approach, and thus now places the indirect optimization approach in a more competitive position with direct methods. Although the PSPM utilizes several features not available with currently more well known indirect optimization methods, the excellent convergence characteristics are primarily due to the easily applied modifications of equations (4.13) to (4.16) together with upper and lower bounds placed on allowable values of the unknown final time. Thus, it is expected that with these modifications, other indirect optimization methods currently programed need not employ the particular solution method presented and the more unfamiliar power series integration in order to obtain the good convergence characteristics displayed by the PSPM.

As a result of this study, several areas are recommended for future investigations. Although the good convergence characteristics of the PSPM are not believed to be unique to the example problems presented, the convergence characteristics of this method should be studied on other problems of larger dimension and of a different nature (such as atmospheric reentry problems with inequality constraints on control and state variables) in order to support the claims made here.

It would appear that the use of regularizing transformations discussed by Tapley, Szebehely, and Lewallen [54] would be as beneficial with power series integration as with more conventional integration schemes in solving trajectory optimization problems. This should be investigated.

The methods presented here for solving two-point boundary value problems are not restricted to the typical problem which results from trajectory optimization. The modifications employed to extend the range of convergence should be equally as beneficial on any two-point or multi-point boundary value problem solved by a Perturbation or Quasilinearization method, and this should be investigated. The advantages of power series integration may be more fully realized for multipoint boundary value problems because of the ease with which the method can obtain solution values at any value of the independent variable. The details of adapting the method for solving nonlinear multipoint boundary value problems have been worked out in this study, and several such problems should be solved to test the usefulness of the power series method. The Newton-Raphson method utilized for solving unspecified final time problems could be applied to multipoint boundary value problems where several boundary conditions at unspecified values of the independent variable are known. This should be demonstrated.

Finally, this investigation has revealed the Perturbation approach to have several practical advantages over the Quasilinearization approach for solving nonlinear boundary value problems. Besides requiring fewer integrations per iteration and less computer storage than the Quasilinearization method, the Perturbation approach admits the capability for simultaneous integration of the reference solution and linearized equations, which in turn allows for variable step integration and capability for extreme solution accuracy. However, the convergence of the Quasilinearization approach has been rigorously established [19], [29], [34] for boundary value problems of a less general nature than considered in this investigation, while the Perturbation approach is lacking in this regard. When compared in numerical studies [24], [42], the methods have displayed similar convergence characteristics, and the Perturbation approach as modified in this study exhibits convergence characteristics superior to the Quasilinearization method reported in reference [35] for the same example problem. Further theoretical investigations of the Perturbation method are needed to establish the necessary and sufficiency theorems for convergence which must exist. In the past, theoretical investigations of the Quasilinearization method have been easier because the method involves iterative solutions of a system of linear differential equations only, while the Perturbation approach involves iterative solution of both linear and nonlinear systems. However, it was established in this investigation that the nonlinear solution at each iteration of the Perturbation method is a particular solution of the linear system. In addition, as shown in Appendix C, initial and final values of the nonlinear reference solution can be related through the same fundamental set of solutions used to construct the general solution of the linear system. Perhaps some advantage can be made of these properties in future theoretical investigations of the Perturbation method.

SELECTED BIBLIOGRAPHY

1. Bliss, G. A. *Lectures on the Calculus of Variations*, The University of Chicago Press, 1946.

2. Kelly, H. S. "Gradient Theory of Optimal Flight Paths," *ARS Journal* (now *AIAA Journal*), October 1960, p. 947.

3. Bryson, A. E., Denham, W. F., Carroll, F. J., and Mikami, K. "Determination of Lift or Drag Programs to Minimize Re-Entry Heating," *Journal of the Aerospace Sciences*, April 1962, p. 420.

4. Bryson, A. E., and Denham, W. F. "A Steepest-Ascent Method for Solving Optimal Programming Problems," *Journal of Applied Mechanics*, June 1962, p. 247.

5. Sage, A. P. *Optimum Systems Control*, Prentice-Hall, Englewood Cliffs, N. J., 1968.

6. Bellman, R. *Dynamic Programming*, Princeton University Press, Princeton, N. J., 1957.

7. Larson, R. E. "Dynamic Programming with Reduced Computational Requirements," *IEEE Transactions on Automatic Control*, April 1965.

8. Lasdon, L. S., Mitter, S. K., and Waren, A. D. "The Conjugate Gradient Method for Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. AC-12, No. 2., April 1967.

9. McReynolds, S. D., and Bryson, A. E. "A Successive Sweep Method for Solving Optimal Programming Problems," Joint Automatic Control Conference, June 1965, pp. 551-555.

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

11. Dreyfus, S. E. *Dynamic Programming and the Calculus of Variations*, Academic Press Inc., New York, 1965.

12. Hestenes, M. R. "Numerical Methods for Obtaining Solutions of Fixed End Point Problems in the Calculus of Variations," The RAND Corporation Memorandum No. RM-102, 1949.

13. Hestenes, M. R. "A General Problem In the Calculus of Variations with Applications to Paths of Least Time," The RAND Corporation Memorandum No. RM-100, 1950, ASTIA Document No. AD-112382.

14. Hestenes, M. R. "Variational Theory and Optimal Control Theory," Conference on Computing Methods in Optimization Problems, Los Angeles, 1964. Published in *Computing Methods in Optimization Problems*, ed. by Balakrisnan, A. V., and Neustadt, L. V. Academic Press New York, 1964, p. 4.

15. Breakwell, J. V. "The Optimization of Trajectories," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 7, 1959, pp. 215-247.

16. Melbourne, W. G. "Three Dimensional Optimum Thrust Trajectories for Power Limited Propulsion Systems," *ARS Journal*, Vol. 31, No. 12, December 1961, pp. 1723-1728.

17. Melbourne, W. G., Sauer, C. G., and Richardson, D. E. "Interplanetary Trajectory Optimization with Power-Limited Propulsion Systems," *Proceedings of the IAS Symposium on Vehicle System Optimization*, Garden City, New York, November 28-29, 1961, pp. 138-150.

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

19. Kalaba, R. E. "On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," *Journal of Mathematics and Mechanics*, Vol. 8, No. 4, 1959, pp. 519-574.

20. Bliss, G. A. *Mathematics for Exterior Ballistics*, Wiley, New York, 1944.

21. Jurovics, S. A., and McIntyre, J. E. "The Adjoint Method and Its Application to Trajectory Optimization," *ARS Journal*, September 1962, pp. 1354-1358.

22. Jazwinski, A. H. "Optimal Trajectories and Linear Control of Nonlinear Systems," *AIAA Journal*, Vol. 2, No. 8, 1964, pp. 1371-1379.

23. Breakwell, J. V., Speyer, J. C., and Bryson, A. E. "Optimization and Control of Nonlinear Systems Using the Second Variation," *SIAM Journal on Control*, Vol. 1, No. 2, 1963.

24. Lewallen, J. M. "An Analysis and Comparison of Several Trajectory Optimization Methods," The University of Texas, Ph. D. Thesis, 1966.

25. Shipman, J. S., and Roberts, S. M. "The Kantorovich Theorem and Two-Point Boundary Value Problems," IBM Journal of Research and Development, Vol. 10, 1966, pp. 402-406.

26. Lastman, G. J., "A Modified Newton's Method for Solving Trajectory Optimization Problems," AIAA Journal, Vol. 6, No. 5, May 1968, pp. 777-780.

27. Kantorovich, L. V., and Akilov, G. P. Functional Analysis in Normed Spaces, The Macmillan Co., New York, 1964, pp. 695-749.

28. Armstrong, E. S. "A Combined Newton-Raphson and Gradient Parameter Correction Technique for Solution of Optimal Control Problems," NASA TR R-293, Washington, D.C., October 1968.

29. McGill, R., and Kenneth, P. "A Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary Value Systems," Proceedings of the 14th International Astronautical Congress, Vol. 4, Gauthier-Villars, Paris, 1963.

30. McGill, R., and Kenneth, P. "Solution of Variational Problems by Means of A Generalized Newton-Raphson Operator," AIAA Journal, Vol. 2, No. 10, October 1964, pp. 1761-1766.

31 Conrad, D. A. "Quasilinearization Extended and Applied to the Computation of Lunar Landings," Proceedings of the 15th International Astronautical Congress, Warsaw, Poland, 1964, pp. 703-709.

32. Long, R. S. "Newton-Raphson Operator; Problems with Undetermined Endpoints," AIAA Journal, Vol. 3, No. 7, July 1965, p. 1352.

33. Johnson, R. W. "Optimum Design and Linear Guidance Law Formulation of Entry Vehicles for Guidance Parameters and Temperature Accumulation Along Optimum Entry Trajectories," University of California at Los Angeles, Ph. D. Thesis, 1966.

34. Leondes, C. T., and Paine, G. "Extensions in Quasilinearization Techniques for Optimal Control," Journal of Optimization Theory and Applications, Vol. 2, No. 5, 1968, pp. 316-330.

35. Lewallen, J. M. "A Modified Quasi-Linearization Method for Solving Trajectory Optimization Problems," AIAA Journal, Vol. 5, No. 5, May 1967, pp. 962-965.

36. Kenneth, P. and Taylor, G. E. "Solution of Variational Problems with Bounded Control Variables by Means of the Generalized Newton-Raphson Method," Recent Advances in Optimization Techniques, ed. by A. Lavi and T. P. Vogl, John Wiley and Sons, New York, 1966, pp. 471-487.

37. McGill, R. "Optimal Control, Inequality State Constraints, and the Generalized Newton-Raphson Algorithm," SIAM Journal on Control, Vol. 3, No. 2, 1965, pp. 291-298.

38. 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," AIAA Progress in Astronautics and Aeronautics: Methods in Astrodynamics and Celestial Mechanics, Vol. 17, ed. by R. L. Duncombe and V. G. Szebehely, Academic Press, New York, 1966, pp. 377-400.

39. Van Dine, C. P. "An Algorithm for the Optimization of Trajectories with Associated Parameters," AIAA Journal, Vol. 7, No. 3, March 1969, pp. 400-405.

40. Kopp, R. E., and McGill, R. "Several Trajectory Optimization Techniques Part I," Computing Methods in Optimization Problems, Academic Press, New York, 1964.

41. Moyer, G. H., and Pinkham, G. "Several Trajectory Optimization Techniques Part II," Computing Methods in Optimization Problems, Academic Press, New York, 1964.

42. Tapley, B. D., and Lewallen, J. M. "Comparison of Several Numerical Optimization Methods," Journal of Optimization Theory and Applications, Vol. 1, No. 1, July 1967, pp. 1-32.

43. Tapley, B. D., Fowler, W. T., and Williamson, W. E. "Computation of Apollo Type Re-Entry Trajectories," Joint Automatic Control Conference, Boulder, Colorado, August 1969.

44. Lewallen, J. M., Tapley, B. D., and Williams, S. D. "Iteration Procedures for Indirect Optimization Methods," Journal of Spacecraft and Rockets, Vol. 5, No. 3, March 1968, pp. 321-327.

45. Ince, E. L. Ordinary Differential Equations, Dover Publications, New York, 1956.

46. Miele, A. "Method of Particular Solutions for Linear Two-Point Boundary Value Problems," Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968, pp. 260-273.

47. Holloway, C. C. "Identification of the Earth's Geopotential," Ph. D. Dissertation, University of Houston, Houston, Texas, June 1968.

48. Luckinbill, D. L., and Childs, S. B. "Inverse Problems in Partial Differential Equations," Report RE 1-68, Project THEMIS, ONR Contract N00014-68-0151, University of Houston, Houston, Texas, August 1968.

49. Baker, B. E., and Childs, S. B. "Identification of Material Properties of a Nonlinearly Elastic Material by Quasilinearization," Report RE 5-69, Project THEMIS, ONR Contract N00014-68-A-0151, University of Houston, Houston, Texas, June 1969.

50. Heideman, J. C. "Use of the Method of Particular Solutions in Nonlinear Two-Point Boundary Value Problems," Journal of Optimization Theory and Applications, Vol. 2, No. 6, 1968, pp. 450-461.

51. Saaty, T. L., and Bram, J. Nonlinear Mathematics, McGraw-Hill, Inc., New York, 1964, pp. 56-70.

52. Petrovski, I. G. Ordinary Differential Equations, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1966, pp. 106-107.

53. Kelley, H. J. "Method of Gradients," in Optimization Techniques, ed. by G. Leitmann, Academic Press, New York, 1962, p. 245.

54. Tapley, B. D., Szebehely, V., and Lewallen, J. M. "Trajectory Optimization Using Regularized Variables," AIAA Journal, Vol. 7, No. 6, June 1969, pp. 1010-1016.

55. Fehlberg, E. "Numerical Integration of Differential Equations by Power Series Expansions, Illustrated by Physical Examples," NASA TN No. TN D-2356, October 1964.

56. Hartwell, J. G. "Simultaneous Integration of N-Bodies by Analytic Continuation with Recursively Formed Derivatives," Journal of the Aerospace Sciences, Vol. XIV, No. 4, July-August 1967, pp. 173-177.

57. Doiron, H. H. "Numerical Integration Via Power Series Expansion," M. S. Thesis, University of Houston, Houston, Texas, August 1967.

58. Apostol, T. M. Mathematical Analysis, Addison-Wesley, Reading, Massachusetts, 1964, pp. 414-415.

APPENDIX A

REDUCTION OF AN OPTIMIZATION PROBLEM TO A TWO-POINT BOUNDARY VALUE PROBLEM

For the example problem considered in Chapter V, it is necessary to determine the optimal thrust vector control for a constant low-thrust rocket in a planar Earth-Mars orbit transfer so that the transfer is completed in minimum time. The orbits of both Earth and Mars are assumed to be circular in this example. In this appendix, the necessary conditions for optimal control outlined in Chapter II are applied to the example problem considered in Chapter V in order to reduce the optimization problem to a two-point boundary value problem.

The equations of motion for the thrusting rocket, expressed in a polar coordinate system with origin at the sun, are given by:

$$\dot{u} = \frac{v^2}{r} - \frac{GM}{r^2} + \frac{T}{m} \sin \beta$$

$$\dot{v} = \frac{-uv}{r} + \frac{T}{m} \cos \beta$$

$$\dot{r} = u$$

$$\dot{\theta} = \frac{v}{r}$$

$$\dot{m} = -c$$

where T is the thrust magnitude, GM is the solar gravitational constant, c is the constant mass flow rate of the rocket exhaust, and β is the time varying thrust control angle.

In order to apply the necessary conditions outlined in Chapter II, the following substitutions are made:

$$x_1 = u$$

$$x_2 = v$$

$$x_3 = r$$

$$x_4 = \theta$$

$$x_5 = m$$

$$u_1 = \beta$$

So that the equations of motion are written in the form $\dot{x} = f(x,u,t)$ corresponding to equation (2.1),

$$\begin{aligned}
\dot{x}_1 &= \frac{x_2^2}{x_3} - \frac{GM}{x_3^2} + \frac{T}{x_5}\sin u_1 = f_1(x,u,t) \\
\dot{x}_2 &= \frac{-x_1 x_2}{x_3} + \frac{T}{x_5}\cos u_1 = f_2(x,u,t) \\
\dot{x}_3 &= x_1 = f_3(x,u,t) \\
\dot{x}_4 &= \frac{x_2}{x_3} = f_4(x,u,t) \\
\dot{x}_5 &= -c = f_5(x,u,t)
\end{aligned} \tag{A.1}$$

The Hamiltonian function is given by

$$H = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 + \lambda_5 f_5$$

and the Euler-Lagrange equations are obtained from the necessary conditions, equation (2.5),

$$\dot{\lambda} = -\left(\frac{\partial H}{\partial x}\right)^T$$

which, after some simplification, can be written

$$\left.\begin{aligned}
\dot{\lambda}_1 &= \left(\frac{x_2}{x_3}\right)\lambda_2 - \lambda_3 \\
\dot{\lambda}_2 &= -\left(\frac{2x_2}{x_3}\right)\lambda_1 + \left(\frac{x_1}{x_3}\right)\lambda_2 - \left(\frac{1}{x_3}\right)\lambda_4 \\
\dot{\lambda}_3 &= \left[\left(\frac{x_2}{x_3}\right)^2 - \frac{2GM}{x_3^3}\right]\lambda_1 - \left(\frac{x_1 x_2}{x_3^2}\right)\lambda_2 + \left(\frac{x_2}{x_3^3}\right)\lambda_4 \\
\dot{\lambda}_4 &= 0 \\
\dot{\lambda}_5 &= -\frac{T}{x_5^2}\left(\lambda_1 \sin u_1 + \lambda_2 \cos u_1\right)
\end{aligned}\right\} \qquad \text{(A.2)}$$

The control variable u_1 is eliminated from equations (A.1) and (A.2) by application of necessary conditions (2.6),

$$\frac{\partial H}{\partial u_1} = 0$$

with the result

$$\lambda_1\left(\frac{T}{x_5} \cos u_1\right) - \lambda_2\left(\frac{T}{x_5} \sin u_1\right) = 0$$

Simplifying,

$$\frac{\lambda_1}{\lambda_2} = \tan u_1$$

which implies,

$$\left.\begin{aligned} \sin u_1 &= \frac{\lambda_1}{\pm\sqrt{\lambda_1^2 + \lambda_2^2}} \\ \cos u_1 &= \frac{\lambda_2}{\pm\sqrt{\lambda_1^2 + \lambda_2^2}} \end{aligned}\right\} \qquad (A.3)$$

The necessary Weierstrass condition

$$E = F(t,x,\dot{X},U,\lambda) - F(t,x,\dot{x},u,\lambda) - \frac{\partial F}{\partial \dot{x}}(t,x,\dot{x},u,\lambda)(\dot{X} - \dot{x})$$

$$- \frac{\partial F}{\partial u}(t,x,\dot{x},u,\lambda)(U - u) \geq 0$$

where $F = \lambda^T(f(x,u,t) - \dot{x})$ and $\dot{X}$ and U are nonoptimal but permissible values for $\dot{x}$ and u, is imposed to resolve the ambiguity in sign appearing in equations (A.3). Since the equations of motion must be satisfied on a permissible trajectory,

$$F(t,x,\dot{X},U,\lambda) - F(t,x,\dot{x},u,\lambda) = 0$$

and since, from the optimality condition,

$$\frac{\partial F}{\partial u} = \frac{\partial H}{\partial u_1} = 0$$

the Weierstrass E-function reduces to

$$E = -\frac{\partial F}{\partial \dot{x}}(t,x,\dot{x},u,\lambda)(\dot{X} - \dot{x}) \geq 0$$

which for this problem simplifies to

$$E = \lambda^T(\dot{X} - \dot{x}) \geq 0$$

Substituting with $\dot{x} = f(x,u,t)$ and $\dot{X} = f(x,U,t)$

$$E = \lambda_1\left[\frac{T}{x_5}\left(\sin U_1 - \sin u_1\right)\right] + \lambda_2\left[\frac{T}{x_5}\left(\cos U_1 - \cos u_1\right)\right] \geq 0$$

Substituting for $\sin u_1$ and $\cos u_1$ with equations (A.3) yields, after some manipulation,

$$E = \frac{T}{x_5}\left(\pm\sqrt{\lambda_1^2 + \lambda_2^2}\right)\left[-1 + \cos\left(U_1 - u_1\right)\right] \geq 0$$

If the above expression is to be nonnegative for all admissible values of U_1, then the negative sign on the radical must be chosen, hence,

$$\sin u_1 = \frac{\lambda_1}{-\sqrt{\lambda_1^2 + \lambda_1^2}}$$

$$\cos u_1 = \frac{\lambda_2}{-\sqrt{\lambda_1^2 + \lambda_2^2}}$$

Substituting the above expressions into equations (A.1) and (A.2) eliminates the control parameter u_1 from the equations of motion.

The terminal boundary conditions, $\psi[x(t_f),t_f]$, which correspond to necessary conditions (2.8), are given by

$$\begin{aligned}
\psi_1 &= x_1\left(t_f\right) - 0 = 0 \\
\psi_2 &= x_2\left(t_f\right) - v_M = 0 \\
\psi_3 &= x_3\left(t_f\right) - r_M = 0 \\
\psi_4 &= x_4\left(t_f\right) - \theta_M\left(t_f\right) = x_4\left(t_f\right) - \theta_M\left(t_o\right) - \frac{v_M}{r_M} t_f = 0
\end{aligned} \tag{A.4}$$

where the subscript M refers to the value for Mars. The performance index $\phi[x(t_f),t_f]$ is simply t_f for a minimum time transfer. Applying necessary condition (2.7),

$$\left(\frac{\partial\phi}{\partial x} + \nu^T \frac{\partial\psi}{\partial x} - \lambda^T\right)\Bigg|_{t_f} = 0$$

yields

$$\begin{aligned}
\nu_1 - \lambda_1\left(t_f\right) &= 0 \\
\nu_2 - \lambda_2\left(t_f\right) &= 0 \\
\nu_3 - \lambda_3\left(t_f\right) &= 0 \\
\nu_4 - \lambda_4\left(t_f\right) &= 0 \\
- \lambda_5\left(t_f\right) &= 0
\end{aligned} \tag{A.5}$$

One version of the example problem in Chapter V assumes that the final angle $x_4(t_f)$ is not constrained. In this case it is seen from the above application that

$$\lambda_4(t_f) = \lambda_4 = 0$$

just as $\lambda_5(t_f)$ was determined to be zero since no constraint was placed on the final spacecraft mass.

The necessary condition corresponding to equation (2.9)

$$\left(\frac{\partial \phi}{\partial t} + \nu^T \frac{\partial \psi}{\partial t} + H\right)\Bigg|_{t_f} = 0$$

becomes

$$1 + \nu_4\left(-\frac{v_M}{r_M}\right) + \lambda_1(t_f)\dot{x}_1(t_f) + \lambda_2(t_f)\dot{x}_2(t_f) + \lambda_3(t_f)\dot{x}_3(t_f)$$
$$+ \lambda_4\dot{x}_4(t_f) + \lambda_5\dot{x}_5(t_f) = 0$$

Using equations (A.5), ν_4 can be eliminated from the above equation to yield

$$\left[1 + \lambda_1\dot{x}_1 + \lambda_2\dot{x}_2 + \lambda_3\dot{x}_3 + \lambda_4\left(\dot{x}_4 - \frac{v_M}{r_M}\right) + \lambda_5\dot{x}_5\right]_{t_f} = 0 \qquad \text{(A.6)}$$

Since the constant Lagrange multipliers ν have been eliminated from the terminal boundary conditions, there is no reason to compute them and the trivial differential equations $\dot{\nu} = 0$ appearing in equation (2.10) can be eliminated from the formulation of the two-point boundary value problem.

Equations (A.1) and (A.2) provide 10 ordinary differential equations to be solved with five initial conditions provided by the known initial position, velocity, and mass of the spacecraft as it leaves an assumed circular Earth orbit about the sun.

$$x_1\left(t_o\right) = 0$$

$$x_2\left(t_o\right) = v_{Earth}$$

$$x_3\left(t_o\right) = r_{Earth}$$

$$x_4\left(t_o\right) = 0$$

$$x_5\left(t_o\right) = m_o$$

Equations (A.5) yield a zero value at the final time for $\lambda_5(t_f)$,

$$\lambda_5\left(t_f\right) = 0$$

Since t_f is not specified, five additional boundary values are required for the 10 differential equations. Four of these conditions are provided by equations (A.4) and the fifth condition is provided by equation (A.6).

From a computational point of view, it is desirable to normalize the values of the variables in the differential equations so that some degree of numerical magnitude compatibility is achieved. Since it was desired to compare the numerical results of this investigation with previously published results of reference [24], the normalization scheme of reference [24] was employed. In this scheme, the fifth equation in (A.2) is eliminated together with terminal condition (A.6). The initial values of the Lagrange multipliers are normalized to the unknown value of $\lambda_3(t_o)$ such that $\lambda_3(t_o)$ is specified to be -1. This is possible since the remaining Euler-Lagrange equations (A.2) are linear and homogeneous in the λ's, and since only the ratios of λ_1 and λ_2 appear in the equations of motion (A.1). In addition to providing better numerical accuracy, this normalization technique also reduces the complexity of the boundary value problem since an additional initial condition is obtained and the terminal condition (A.6) need not be used as a boundary condition. Since equation (A.6) must be satisfied on the optimal trajectory, it can be used to recover the unnormalized values of the Lagrange multipliers. However, there appears to be no practical reason to recover these values. Other normalized values of parameters of interest are

Gravitational constant of the Sun, GM = 1.0

Initial spacecraft mass, m_o = 1.0

Initial spacecraft velocity, v_{Earth} = 1.0

Initial spacecraft radius, r_{Earth} = 1.0

Terminal spacecraft velocity, v_{Mars} = 0.81012728

Terminal spacecraft radius, r_{Mars} = 1.5236790

Thrust = 0.14012969

Mass flow rate = 0.074800391

With these normalized values, units of various physical quantities are

Length unit = 1 astronomical unit = 1.495987×10^{11} meters

Mass unit = 6.7978852×10^{2} kilograms

Velocity unit = 2.9784901×10^{4} m/sec

Force unit = 4.0312370 newtons

Time unit = 5.0226355×10^{6} second = 58.132355 days

APPENDIX B

A POWER SERIES NUMERICAL INTEGRATION METHOD

One of the most important facets of any method for solving multipoint boundary value problems is the numerical integration scheme used. Since the numerical integration of differential equations consumes the bulk of computer time, it is desirable to have a fast and accurate integration method. Among the most popular integration methods are the well-known Runge-Kutta formulas and predictor-corrector methods. In this appendix, a power series integration method is presented which has several features which make it uniquely suited for use as an integration method in solving multipoint boundary value problems.

The capability for solving differential equations by power series expansions has been known since B. Taylor (1685-1731). However, this method has not enjoyed the popularity of other numerical integration schemes. This is probably due to the fact that it is an impractical method for hand calculation or even desk calculators, and thus did not receive the early attention and development of the currently more popular integration methods used on digital computers.

With modern digital computers, the cumbersome application of the power series method is easily overcome and its practicality is evidenced by its high accuracy, large step sizes, good speed, and variable step size capability. The use of power series as a method for digital computers has been studied by Fehlberg [55] and Hartwell [56]. Detailed programing steps for the method are given by Doiron [57]. The method is reported to have superior speed and accuracy characteristics on problems where integration must be made over large intervals of the independent variable with frequent changes in the integration step size.

Given a system of differential equations,

$$\dot{z}_i = f_i\left(z_1, z_2, \ldots z_N, t\right)$$

$$z_i\left(t_o\right) = z_{oi} \qquad i = 1, 2, \ldots N \tag{B.1}$$

the method assumes a power series expansion exists in a neighborhood of t_o for each of the variables z_i of the form

$$z_i(t) = \sum_{k=1}^{\infty} z_i^{(k)}\left(t - t_o\right)^{k-1} \tag{B.2}$$

where the $z_i^{(k)}$ are power series coefficients and t_o is the value of the independent variable where the power series expansion is made. In the following, it will be assumed that power series solutions of equation (B.1) exist.

Letting $(\dot{z}_i)^{(k)}$ denote the power series coefficients of $\dot{z}_i$, it is easily determined from term-by-term differentiation of (B.2) that

$$\left(z_i\right)^{(k+1)} = \frac{\left(\dot{z}_i\right)^{(k)}}{k} \tag{B.3}$$

which yields a recurrence relation for $(z_i)^{(k+1)}$ if $(\dot{z}_i)^{(k)}$ is known. Since $\dot{z}_i = f_i(z_1, z_2 \ldots z_N, t)$, it follows that power series expansions of the functions f_i exist with power series coefficients denoted by $f_i^{(k)}$. Thus, by equality of power series, we have equality of the power series coefficients, and (B.3) can be written

$$\left(z_i\right)^{(k+1)} = \frac{\left(f_i\right)^{(k)}}{k} \quad . \qquad \text{(B.4)}$$

The main difficulty in applying the method is involved with determining the coefficients $f_i^{(k)}$. The functional relationship between the z_i, as expressed by (B.1), must be carried out in "power series arithmetic" and when coefficients of like powers of $(t - t_o)$ are collected, these coefficients represent the $f_i^{(k)}$. It can be shown (see Theorem 13-27 of Apostol [58]) that the coefficients $f_i^{(k)}$ involve only the first k coefficients, $z_i^{(k)}$, of the power series for the z_i. This guarantees that equation (B.4) does actually represent a recurrence formula for $z_i^{(k+1)}$ in terms of the first k, $z_i^{(k)}$. The application of the power series arithmetic is greatly simplified through the use of auxiliary series and repetitive application of known algorithms for series addition, subtraction, multiplication, and division. Easily programed algorithms are also known for generating power series coefficients of transcendental functions of power series such as $\sin z$ and z^{β}, where β is some real number.

Let u, v, and w be power series of the form

$$u = \sum_{k=1}^{\infty} u_k \left(t - t_o\right)^{k-1}$$

$$v = \sum_{k=1}^{\infty} v_k \left(t - t_o\right)^{k-1}$$

$$w = \sum_{k=1}^{\infty} w_k \left(t - t_o\right)^{k-1}$$

The following power series operations are defined by:

Addition: $w = u + v \Rightarrow w_k = u_k + v_k$

Subtraction: $w = u - v \Rightarrow w_k = u_k - v_k$

Multiplication: $w = u \cdot v \Rightarrow w_k = \sum_{i=1}^{k} u_i v_{k-i+1}$

Division: $w = u/v$

$$w_k = \frac{u_k - \sum_{i=1}^{k-1} w_i v_{k-i+1}}{v_1} \quad , \quad w_1 = \frac{u_1}{v_1}$$

Square root:

$$w = \sqrt{u} \Rightarrow w_k = \frac{1}{2w_1}\left(u_k - \sum_{i=2}^{k-1} w_i w_{k-i+1}\right)$$

$$w_1 = \sqrt{u_1}$$

Integral or fractional powers:

$$w = u^{\beta} \Rightarrow w_1 = u_1^{\beta}$$

$$w_2 = \frac{\beta u_2 w_1}{u_1}$$

for $k > 1$

$$w_{k+1} = \frac{1}{ku_1}\left\{\beta k u_{k+1} w_1 + \sum_{i=1}^{k-1} [i(\beta + 1) - k] u_{i+1} w_{k-i+1}\right\}$$

Sine and cosine:

$$\begin{matrix} u = \cos w \\ v = \sin w \end{matrix} \Rightarrow \begin{matrix} u_1 = \cos w_1 \\ v_1 = \sin w_1 \end{matrix}$$

$$u_{k+1} = -\frac{1}{k}\sum_{i=1}^{k} i w_{i+1} v_{k-i+1}$$

$$v_{k+1} = \frac{1}{k}\sum_{i=1}^{k} i w_{i+1} u_{k-i+1}$$

These algorithms are sufficient for the differential equations which are solved in this thesis.

An Example Problem.

Given the differential equations,

$$\dot{x} = y + \frac{\cos w}{\left(x^2 + y^2\right)^{3/2}}$$

$$\dot{y} = x + \frac{\sin w}{\left(x^2 + y^2\right)^{3/2}}$$

$$\dot{w} = w \cdot y$$

with initial conditions, $x(t_o) = x_1$, $y(t_o) = y_1$, and $w(t_o) = w_1$, generate power series coefficients for x, y, and w for a power series expansion about t_o.

The coefficients are obtained by computing in sequence the k^{th} coefficient of each of the auxiliary series in equations (B.5) using the above algorithms, and then applying the recurrence relations (B.6) to obtain the $(k + 1)$st coefficients for x, y, and w. The process is repeated for $k = 1, 2, 3 \ldots N-1$, where N is the desired number of coefficients.

$$\left.\begin{aligned} a &= w \cdot y \\ u &= \cos w \\ v &= \sin w \\ b &= x \cdot x \\ c &= y \cdot y \\ r &= b + c \\ s &= r^{-3/2} \\ e &= u \cdot s \\ f &= v \cdot s \end{aligned}\right\} \qquad (B.5)$$

$$\left.\begin{aligned} x_{k+1} &= \frac{(\dot{x}_k)}{k} = \frac{(y_k + e_k)}{k} \\ y_{k+1} &= \frac{(\dot{y}_k)}{k} = \frac{(x_k + f_k)}{k} \\ w_{k+1} &= \frac{(\dot{w}_k)}{k} = \frac{(a_k)}{k} \end{aligned}\right\} \tag{B.6}$$

Once the desired number of coefficients N are computed, the next step in the integration process is to determine the integration step size $(t - t_o)$, which can be used with the available coefficients, while maintaining a specified numerical accuracy in the evaluation of the power series solutions. In general, a larger step size may be used with a larger number of coefficients. A practical limit for the number of coefficients which should be computed is determined by the magnitude of the N^{th} coefficient of the series to be evaluated. Digital computers have a largest and smallest value of the magnitude of a number which can be accurately represented. Depending on the radius of convergence of the series, the coefficients may approach one or the other of these limits. The number of coefficients computed should be limited to avoid these number magnitude limits.

It is assumed that N power series coefficients are available for evaluation. A method for determining the largest allowable integration step size is developed below. The equivalence of the power series expansion and Taylor series expansion of a function $x(t)$ about a point t_o is well known. The Taylor series expansion can be written

$$x(t) = x\left(t_o\right) + x'\left(t_o\right)\left(t - t_o\right) + \frac{x''}{2!}\left(t_o\right)\left(t - t_o\right)^2 + \ldots$$
$$+ \frac{x^N\left(t_o\right)}{N!}\left(t - t_o\right)^N + R_{N+1}$$

where the remainder, R_{N+1} is given by

$$R_{N+1} = \frac{x^{N+1}\left(t_1\right)}{(N+1)!}\left(t - t_o\right)^{N+1} \qquad t_o < t_1 < t$$

If it is assumed that $x^{N+1}(t_1) \approx x^{N+1}(t_o)$, then the truncation error R_{N+1} is of the order of the first neglected term in the summation of the series. Since in most applications it is desirable to limit the relative truncation error rather than the absolute truncation error, the step size $(t - t_o)$ should be chosen to meet a specified relative truncation error bound, e_{rel}. Let the relative truncation error be defined by

$$e_{rel} = \frac{\left|x(t)_{numerical} - x(t)_{exact}\right|}{\left|x(t)_{exact}\right|}$$

For all practical purposes, the denominator in the above expression need only represent the magnitude of $x(t)$, and, therefore, it may be approximated by the summation of the first p terms of the power series representation of $x(t)$. In light of the foregoing assumptions, if $x(t)$ is to be accurately represented by

$$x(t) = \sum_{k=1}^{N} x_k\left(t - t_o\right)^{k-1}$$

then a reasonable truncation error check would require that each of the last several terms (r terms, for example) of the summation be less in magnitude than

$$e_{rel} \sum_{k=1}^{N-r} x_k\left(t - t_o\right)^{k-1}$$

where e_{rel} is the specified relative error allowable. The value of r depends on the severity of the test desired. In practice, $r = 2$ has been sufficient to maintain the desired accuracy. With r specified, a requirement of the test can be stated.

$$\left|x_{N-r+1}\right|\left|\left(t - t_o\right)^{N-r}\right| < e_{rel}\left|\sum_{k=1}^{N-r} x_k\left(t - t_o\right)^{k-1}\right| \qquad (B.7)$$

It is desirable to solve for $(t - t_o)$ which will satisfy this test. Since the summation on the right of the inequality (B.7) is used only to approximate the magnitude of $x(t)$, let the magnitude be approximated by the constant term in the expansion x_1. Then, taking logarithms of both sides above yields

$$\log\left|t - t_o\right| < \frac{\log\left(e_{rel}\left|\frac{x_1}{x_{N-r+1}}\right|\right)}{N-r}$$

Exponentiating both sides yields an estimate for the allowable convergence interval

$$\left|t - t_o\right| < e\left[\frac{\log\left(e_{rel}\left|\frac{x_1}{x_{N-r+1}}\right|\right)}{(N-r)}\right] \qquad (B.8)$$

Since normally there are more than one series to be evaluated, it is necessary to determine the largest convergence interval common to all series to be evaluated. Noting in (B.8) that $|t - t_o|$ is a monotonic increasing function of $|x_1/x_{N-r+1}|$, it is only necessary to compute for each series the value analogous to $|x_1/x_{N-r+1}|$ to determine which of the series will yield the smallest convergence interval. A trial convergence interval can then be obtained by multiplying the quantity on the right of (B.8) by a positive number, P, less than unity to insure the inequality. A trial step size determined in this manner may still not satisfy the convergence test (B.7) because the approximation of x_1 for

$$\sum_{k=1}^{N-r} x_k\left(t - t_o\right)^{k-1}$$

may not be sufficiently accurate. Since a failure of the test (B.7) during evaluation for any one series would require reducing the convergence interval and reevaluating all series, it is desirable to choose the number P so that the likelihood of convergence test failure is very small. The choice of P is dependent on the number of coefficients used, since longer convergence intervals allow for a larger change in magnitude of the solutions. In practice, values of P of 0.9 or 0.8 have worked well with 10 to 20 term power series. It should be noted that special logic is required in the case where either of the coefficients x_1 or x_{N-r+1} is zero.

Returning to the example problem, once a trial convergence interval Δt for the series $x(t)$, $y(t)$, and $w(t)$ has been determined, the series can be evaluated for any value t_1, $|t_1 - t_o| < \Delta t$. If the solution is required for some t_1 outside the convergence interval, the series may be evaluated at $t_2 = t_o \pm \Delta t$ and new series expansions about t_2 can be obtained. The analytical continuation can be repeated until power series which will converge at the desired value of t are obtained.

APPENDIX C

EQUIVALENCE OF TWO METHODS FOR MODIFYING BOUNDARY CONDITIONS OF LINEAR DIFFERENTIAL EQUATIONS

The following derivation is made to support the claim made in Chapter IV regarding the equivalence of two methods for determining modified initial conditions for the reference solution of the Particular Solution Perturbation Method. To simplify notation in the derivation, the general solution of the N dimensional linear system (4.12) with m specified initial conditions is written as a linear combination of particular solutions

$$^{n}y(t) = \sum_{k=1}^{S+1} \alpha_k \, {}_{n}p^{k}(t) = {}_{n}P(t)\alpha \qquad S = N - m$$

where ${}_{n}P(t)$ is an N by $(S+1)$ matrix formed with columns of particular solutions

$$_{n}P(t) = \left[{}_{n}p^{1} \quad {}_{n}p^{2} \quad \cdots \quad {}_{n}p^{S+1} \right]$$

and α is the vector of superposition constants

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{S+1} \end{bmatrix}$$

Let terminal boundary conditions at the fixed final time be specified

$$^{n}y_i\left(t_f\right) = z_{fi} \qquad i = k + 1, k + 2, \ldots k + S$$

for some $0 \leq k \leq N - 1$. An $(S+1)$ vector $y(t)$ and $(S+1)$ by $(S+1)$ submatrix $\left[P(t)\right]_S$ are formed by partitioning out the $(k+1)$st through the $(k+S)$ rows of $^{n}y(t)$ and $_{n}P(t)$ and then augmenting each by a row of 1's to obtain

$$y(t) = \begin{bmatrix} 1 \\ ^{n}y_{k+1} \\ ^{n}y_{k+2} \\ \vdots \\ ^{n}y_{k+S} \end{bmatrix} \qquad [P(t)]_S = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ {}_{n}p^{1}_{k+1} & {}_{n}p^{2}_{k+1} & & {}_{n}p^{S+1}_{k+1} \\ {}_{n}p^{1}_{k+2} & {}_{n}p^{2}_{k+2} & \cdots & \vdots \\ \vdots & \vdots & & \\ {}_{n}p^{1}_{k+S} & {}_{n}p^{2}_{k+S} & & {}_{n}p^{S+1}_{k+S} \end{bmatrix}$$

Method 1

Given terminal boundary conditions z_{fi}'s, $i = k + 1, \ldots (k + N - m)$ written as an $S = N - m$ vector z_f, modified terminal boundary conditions for the system (4.12) are formed by

$$^{n}y_f = \varepsilon z_f + (1 - \varepsilon)\ ^{n}z_f \tag{C.1}$$

where ${}^{n}y_f$ and ${}^{n}z_f$ are S vectors with elements

$$\begin{bmatrix} {}^{n}y_{k+1}(t_f) \\ {}^{n}y_{k+2}(t_f) \\ \vdots \\ {}^{n}y_{k+S}(t_f) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} {}^{n}z_{k+1}(t_f) \\ {}^{n}z_{k+2}(t_f) \\ \vdots \\ {}^{n}z_{k+S}(t_f) \end{bmatrix}$$

respectively, and ${}^{n}z(t)$ is the nth reference solution of a Perturbation method.

It follows from applying the boundary condition (C.1) and the auxiliary conditions $\sum_{k=1}^{S+1} \alpha_k = 1$ that

$$y(t_f) = \left[P(t_f)\right]_s \alpha = \begin{bmatrix} 1 \\ {}^{n}y_f \end{bmatrix}$$

Solving for α,

$$\alpha = \left[P(t_f)\right]_s^{-1} \begin{bmatrix} 1 \\ {}^{n}y_f \end{bmatrix} .$$

Initial conditions for the (n+1)st reference solution are obtained from

$$^{n+1}z\left(t_o\right) = {^n}y\left(t_o\right) = {_n}P\left(t_o\right)\alpha = \left[{_n}P\left(t_o\right)\right]\left[P\left(t_f\right)\right]_s^{-1}\begin{bmatrix}1\\ {^n}y_f\end{bmatrix} \qquad (C.2)$$

Before proceeding with an analysis of the alternate method for computing $^{n+1}z(t_o)$, it is necessary to establish an identity which will be needed. Using Theorem 1 of Chapter IV, it is possible to write

$$^n z(t) = {_n}p^1(t)$$

subject to

$$^n z\left(t_o\right) = {_n}p^1\left(t_o\right)$$

Forming an (S+1) vector z(t) by partitioning $^n z(t)$ and then augmenting with a 1 in the same manner that y(t) was formed from $^n y(t)$ yields $z(t) = [P(t)]_s\gamma$, where $\gamma = [1,0,0,\ldots 0]^T$. At the final time

$$z\left(t_f\right) = \left[P\left(t_f\right)\right]_s\gamma = \begin{bmatrix}1\\ {^n}z_f\end{bmatrix}$$

so that

$$\gamma = \left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ {}^{n}z_f \end{bmatrix}$$

which allows initial and final conditions for ${}^{n}z(t_o)$ to be related by

$$ {}^{n}z\left(t_o\right) = {}_{n}P\left(t_o\right)\gamma = \left[{}_{n}P\left(t_o\right) \left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ {}^{n}z_f \end{bmatrix} \qquad (C.3)$$

This seemingly awkward result will allow considerable simplification in the derivation which follows.

Method 2

For this method, ${}^{n}y(t_o)$ is computed without modifying terminal boundary conditions, and then ${}^{n+1}z(t_o)$ is computed by

$$ {}^{n+1}z\left(t_o\right) = {}^{n}z\left(t_o\right) + \varepsilon\left({}^{n}y\left(t_o\right) - {}^{n}z\left(t_o\right)\right) \qquad (C.4)$$

It is necessary to show that ${}^{n+1}z(t_o)$ computed in this manner is equal to the result, (C.2).

First, ${}^{n}y(t_o)$ is determined by applying the unmodified terminal boundary conditions. Using similar notation as before

$$y\left(t_f\right) = \left(P\left(t_f\right)_s \alpha = \begin{bmatrix} 1 \\ z_f \end{bmatrix}$$

so that

$$\alpha = \left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ z_f \end{bmatrix}$$

and consequently

$${}^n y\left(t_o\right) = {}_n P\left(t_o\right)\alpha = \left[{}_n P\left(t_o\right)\right]\left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ z_f \end{bmatrix}$$

Implementing the modification (C.4)

$$\begin{aligned} {}^{n+1}z\left(t_o\right) &= {}^n z\left(t_o\right) + \varepsilon \left\{ \left[{}_n P\left(t_o\right)\right]\left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ z_f \end{bmatrix} - {}^n z\left(t_o\right) \right\} \\ &= (1 - \varepsilon)^n z\left(t_o\right) + \varepsilon\left[{}_n P\left(t_o\right)\right]\left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ z_f \end{bmatrix} \end{aligned}$$

Using the result (C.3) and factoring

$${}^{n+1}z\left(t_o\right) = \left[{}_n P\left(t_o\right)\right]\left[P\left(t_f\right)\right]_s^{-1} \left\{ \varepsilon \begin{bmatrix} 1 \\ z_f \end{bmatrix} + (1 - \varepsilon)\begin{bmatrix} 1 \\ {}^n z_f \end{bmatrix} \right\}$$

Substituting with equation (C.1), the final result is obtained

$${}^{n+1}z\left(t_o\right) = \left[{}_n P\left(t_o\right)\right]\left[P\left(t_f\right)\right]_s^{-1} \begin{bmatrix} 1 \\ {}^n y_f \end{bmatrix} \qquad (C.5)$$

A comparison of equations (C.2) and (C.5) reveals that they are identical and therefore Methods 1 and 2 are shown to be equivalent.