RE-176J

SOLUTION OF VARIATIONAL PROBLEMS BY MEANS OF A GENERALIZED NEWTON-RAPHSON OPERATOR

May 1964

Grumman

RESEARCH DEPARTMENT

GRUMMAN AIRCRAFT ENGINEERING CORPORATION
BETHPAGE NEW YORK

Grumman Research Department Report RE-176J

SOLUTION OF VARIATIONAL PROBLEMS BY MEANS OF A GENERALIZED NEWTON-RAPHSON OPERATOR

by

Robert McGill

and

Paul Kenneth
Systems Research Section

May 1964

SAME AS N64-20951

NASA TMX 53021

Approved by: C. E. Mack, Jr.
C. E. Mack, Jr.
Director of Research

SOLUTION OF VARIATIONAL PROBLEMS BY MEANS OF A GENERALIZED NEWTON-RAPHSON OPERATOR*

Robert McGill† and Paul Kenneth‡
Research Department
Grumman Aircraft Engineering Corporation
Bethpage, N.Y.

ABSTRACT

23751

This paper presents the development of an indirect method for solving variational problems by means of an algorithm for obtaining the solution to the associated nonlinear two-point boundary value problem. The method departs from the usual indirect procedure of successively integrating the nonlinear equations and adjusting arbitrary initial conditions until the remaining boundary conditions are satisfied. Instead, an operator is introduced which produces a sequence of sets of functions which satisfy the boundary conditions but in general do not satisfy the nonlinear system formed by the state equations and the Euler-Lagrange equations. Under appropriate conditions this sequence converges uniformly and rapidly (quadratically) to the solution of the nonlinear boundary value problem. The computational effectiveness of the algorithm is demonstrated by three numerical examples.

---

*This work was partially supported by the Applied Mathematics Division, AFOSR under Contract No. AF49(638)-1207, and the Aero-Astrodynamics Laboratory, NASA Marshall Space Flight Center under Contract No. NAS8-1549.

†Research Scientist. Member AIAA.

‡Research Mathematician.

INTRODUCTION

The mathematical theory used for the study of optimization problems is the Calculus of Variations. Application of this theory to meaningful models of physical situations generally results in a mathematical representation of the solution which requires some numerical technique to effect solutions of use to the engineer. Since the major computational device available today is the high speed digital computer, e.g., the IBM 7094, an a priori requirement for a numerical algorithm is that it be systematically adaptable to high speed digital computation. For the Calculus of Variations there are two general numerical approaches; the Direct Methods, and the Indirect Methods. The direct methods proceed by solving a sequence of nonoptimal problems with the property that each successive set of solution functions yields an improved value for the functional being optimized. An example of such a procedure is the Method of Gradients which has been applied to a variety of problems with considerable success. The indirect methods are concerned to find by numerical means a set of functions which satisfy the necessary conditions for an extremal, i.e., the Euler-Lagrange differential equations. These necessary conditions and boundary conditions form a <u>nonlinear</u> boundary value problem and it is here that the numerical difficulty arises. The usual approach to this problem is the systematic variation of arbitrarily chosen initial conditions until the remaining boundary conditions are met. This technique has proved largely unsuccessful owing to increased dimensionality of the interesting problems and to the sensitivity of boundary conditions to small changes in initial conditions. In lieu of this an algorithm has been developed which proceeds by solving a sequence of <u>linear</u> boundary value problems such that the sequence of solutions converges to the solution of the nonlinear problem. Since the linear boundary value problem is easily handled numerically the algorithm is readily adaptable to high speed digital computation.

In what follows we shall discuss this approach in some detail including a discussion of the numerical application. This is followed by three numerical examples to illustrate the computational effectiveness of the method.

For comparison with other methods for handling the associated boundary value problem see Breakwell, et al.,[1] Scharmack,[2] and Kelley, et al.[3]. For a direct comparison of gradient, second variation, and generalized Newton-Raphson techniques, as applied to a specific optimization problem, see Kopp, et al.[4].

THE GENERALIZED NEWTON-RAPHSON OPERATOR

We are concerned with nonlinear operator equations of the following form

$$BX = 0$$

where X is an element of an appropriate metric space S and B is a nonlinear operator.

For the case of the nonlinear two-point boundary value problems of interest herein the operator equation $BX = 0$ is given by the following system of nonlinear differential equations and boundary conditions:

$$\dot{X} - F(X,t) = 0 \quad , \quad t \in [t_0, t_f]$$

$$\begin{array}{ll} x^{(1)}(t_0) = x_0^{(1)} & x^{(1)}(t_f) = x_f^{(1)} \\ \vdots & \vdots \\ x^{(\frac{N}{2})}(t_0) = x_0^{(\frac{N}{2})} & x^{(\frac{N}{2})}(t_f) = x_f^{(\frac{N}{2})} \quad , \end{array}$$

where

$$X = \left(x^{(1)}, \ldots, x^{(N)}\right)$$

$$F = \left(f^{(1)}, \ldots, f^{(N)}\right)$$

$$f^{(i)} = f^{(i)}\left(x^{(1)}, \ldots, x^{(N)}, t\right), \qquad i = 1, \ldots, N .$$

The metric space S is given by

$$S = \left\{X(t): \ x^{(i)}(t) \text{ is continuous on } [t_0, t_f], \quad i = 1, \ldots, N\right\},$$

with the metric

$$\rho(X_1, X_2) = \sum_{i=1}^{N} \max_t \left| x_2^{(i)}(t) - x_1^{(i)}(t) \right| , \qquad X_1, X_2 \in S .$$

We define an operator A on S by $X_{n+1} = AX_n$, $n = 0,1,\ldots$; X_0 arbitrary in S,

$$\dot{X}_{n+1} = J(X_n, t)\,[X_{n+1} - X_n] + F(X_n, t)$$

$$\begin{array}{ll} x_n^{(1)}(t_0) = x_0^{(1)} & x_n^{(1)}(t_f) = x_f^{(1)} \\ \vdots \quad \vdots & \vdots \quad \vdots \\ x_n^{(\frac{N}{2})}(t_0) = x_0^{(\frac{N}{2})} & x_n^{(\frac{N}{2})}(t_f) = x_f^{(\frac{N}{2})} \end{array}$$

$n = 1, 2, \ldots ,$

where $J(X,t)$ is the Jacobian matrix of partial derivatives of the $f^{(i)}$ with respect to the $x^{(j)}$, $i = 1,\ldots,N$, $j = 1,\ldots,N$. Under appropriate conditions the sequence $\{X_n\}$ converges strongly to the solution X^* of the operator equation $BX = 0$, i.e., $\lim_{n \to \infty} \rho(X_n,X^*) = 0$, where X^* is the solution of the nonlinear boundary value problem. The metric ρ implies uniform convergence for each of the component functions $x^{(i)}(t)$ of $X(t)$.

The operator A is called the Generalized Newton-Raphson operator since it may be obtained from a direct generalization of the Newton-Raphson sequence for finding roots of scalar equations. For the scalar case the operator equation $BX = 0$ becomes

$$f(x) = 0$$

and the sequence defining A becomes

$$0 = f'(x_n)[x_{n+1} - x_n] + f(x_n) \quad , \quad n = 0,1,2,\ldots .$$

The appropriate metric space S is the scalar field with the usual metric. As before, $x_{n+1} = Ax_n$, $n = 0,1,2,\ldots$, and x_0 is an approximate solution of $f(x) = 0$. As can be seen from the scalar application the basic concept involved is geometric; a curve is sequentially replaced by its tangent line, i.e., the nonlinear problem is replaced by a sequence of linear problems. Since there is a well developed structure for linear problems, e.g., superposition for systems of linear differential equations, the algorithm becomes computationally attractive. In addition, since the <u>linear</u> two-point boundary value problem can be reduced to repeated numerical integration of initial value problems, the method is readily adaptable to high speed automatic machine computation.

A basic generalization of Newton's Method, to operator equations in Banach spaces, was first obtained by Kantorovitch.[5] Warga[6] considered the solution of the initial value problem for first order differential equations by a special case of Kantorovitch's generalization. The algorithm was apparently first suggested for boundary value problems by Hestenes[7] who called it "Differential Variations," and later further developed by Bellman and Kalaba[8] who refer to the technique as "Quasilinearization." Kalaba gives a convergence proof[8], based on monotonicity and convexity arguments, for the case of a single second order differential equation with two-point boundary conditions. A convergence proof for N dimensional systems was given by McGill and Kenneth.[9] The latter proof proceeds by establishing sufficient conditions for the operator A to be a contraction of a complete metric space into itself. The desired results then follow from the Contraction Mapping Principle.[10] The method is also mentioned by Kelley[11] who remarks that computational experience with the technique is lacking.

NUMERICAL APPLICATION

In this section we present a brief description of a numerical procedure for solving the linear system. This procedure, with appropriate modifications, was used in obtaining the solutions to the numerical examples included in this report.

At the $n+1^{st}$ stage of the iteration we have the linear system

$$\dot{X}_{n+1} = J(X_n,t)[X_{n+1} - X_n] + F(X_n,t)$$

which is equivalent to

$$\dot{X} = C(t)X(t) + D(t) \quad , \quad t \in [t_0,t_f]$$

$$X = (x_1,\ldots,x_N)$$

$$\begin{array}{ll} x_1(t_0) = x_{10} & x_1(t_f) = x_{1f} \\ \vdots \quad\quad \vdots & \vdots \quad\quad \vdots \\ x_{\frac{N}{2}}(t_0) = x_{\frac{N}{2}0} & x_{\frac{N}{2}f}(t_f) = x_{\frac{N}{2}f} \end{array}$$

Generate by numerical integration a set $\left\{X^{(\frac{N}{2}+i)}(t)\right\}$, $i = 1, \ldots, \frac{N}{2}$, of solutions of the homogeneous system $\dot{X} = C(t)X(t)$ with initial conditions

$$X^{(\frac{N}{2}+1)}(t_0) = (0,0,\ldots,0,x_{\frac{N}{2}+1} = 1,0,\ldots,0)$$

$$X^{(\frac{N}{2}+2)}(t_0) = (0,0,\ldots,0,x_{\frac{N}{2}+2} = 1,0,\ldots,0)$$

$$\vdots$$

$$X^{(N)}(t_0) = (0,0,\ldots,0,\ldots,0,1) .$$

Generate a particular solution $X^{(P)}(t)$ of the nonhomogeneous system $\dot{X} = C(t)X(t) + D(t)$ with initial conditions

$$X^{(P)}(t_0) = (x_{10},x_{20},\ldots,x_{\frac{N}{2}0},K_1,K_2,\ldots,K_{\frac{N}{2}})$$

where K_i, $i = 1,\ldots,\frac{N}{2}$, are arbitrary, e.g., $K_1 = K_2 = \ldots = K_{\frac{N}{2}} = 0$. They should, however, following a suggestion by Richard Bellman, be chosen to preserve numerical precision in solving the $\frac{N}{2}$ simultaneous linear equations given below. The solution $X(t)$ of the nonhomogeneous system with the prescribed boundary conditions is then given by

$$X(t) = c_{\frac{N}{2}+1}X^{(\frac{N}{2}+1)}(t) + c_{\frac{N}{2}+2}X^{(\frac{N}{2}+2)}(t) + \ldots + c_N X^{(N)}(t) + X^{(P)}(t) ,$$

where the $\frac{N}{2}$ constants $c_{\frac{N}{2}+i}$, $i = 1, \ldots, \frac{N}{2}$, are determined from e boundary conditions at $t = t_f$ by the solution of $\frac{N}{2}$ simultaneous linear equations.

For the purpose of conserving rapid access storage and also as a check on the solution of the linear system the solution $X(t)$ was not obtained from the linear combination given above. Rather it was calculated by once more integrating the nonhomogeneous system $\dot{X} = C(t)X(t) + D(t)$ with initial conditions

$$X(t_0) = (x_{10}, x_{20}, \ldots, x_{\frac{N}{2}0}, c_{\frac{N}{2}+1} + K_1, c_{\frac{N}{2}+2} + K_2, \ldots, c_N + K_{\frac{N}{2}}) .$$

The latter procedure requires the storage of only the final values of the vectors $\left\{X^{(\frac{N}{2}+i)}\right\}$, $i = 1, \ldots, \frac{N}{2}$, and the final value of $X^{(P)}$ for the computation of $X(t)$. The solution $X(t)$ is of course stored since it is required for the determination of $C(t)$ and $D(t)$ for the next iteration.

ORBITAL INTERCEPT EXAMPLE

The first example although not an optimization problem serves to illustrate the application of the algorithm to a given nonlinear boundary value problem.

The problem solved is that of determining the free fall path which a space vehicle must follow in transferring from a specified position three hundred miles above the earth to another specified position six hundred miles above the earth, with a fixed transit time. The vehicle is assumed to be in coasting flight and the perturbing effect of the moon is included. A schematic diagram of the problem is shown in Fig. 1, where $X_0(t) = \left(x_0(t), y_0(t), z_0(t)\right)$, the starting vector, is of the simplest possible form, namely, the straight line joining the two points in space; $X^*(t)$ is the solution vector.

The unit of length is taken to be the radius of the earth and the principal gravitational constant is normalized to one. This results in a time unit of 805.46 seconds.

The sixth order nonlinear system and two-point boundary conditions which furnish the mathematical description of the problem are given by:

$$\ddot{x} = -K\frac{x}{r^3} + K_M\left(\frac{x_M - x}{\delta^3} - \frac{x_M}{r_M^3}\right)$$

$$\ddot{y} = -K\frac{y}{r^3} + K_M\left(\frac{y_M - y}{\delta^3} - \frac{y_M}{r_M^3}\right) \qquad ; \qquad t \in [0,2]$$

$$\ddot{z} = -K\frac{z}{r^3} + K_M\left(\frac{z_M - z}{\delta^3} - \frac{z_M}{r_M^3}\right)$$

$$x(0) = 1.076000 \qquad x(2) = 0.$$

$$y(0) = 0. \qquad y(2) = 0.576000$$

$$z(0) = 0. \qquad z(2) = 0.997661$$

$$r = [x^2 + y^2 + z^2]^{\frac{1}{2}}$$

$$r_M = [x_M^2 + y_M^2 + z_M^2]^{\frac{1}{2}}$$

$$\delta = \left[(x_M - x)^2 + (y_M - y)^2 + (z_M - z)^2\right]^{\frac{1}{2}} .$$

For simplicity the lunar coordinates, x_M, y_M, z_M, are assumed constant.

The time interval $[0,2]$ was divided into 100 parts and the necessary numerical integrations carried out by means of a high speed digital computer (IBM 7094) to an accuracy of seven significant figures. The results are exhibited in Table 1 where for brevity only six points in time are shown. $X_0(t)$ is the linear starting function; $X_1(t)$ is the first mapping; $X_2(t)$ is the second mapping, etc.; and $X^*(t)$ results from the integration of the actual non-linear equations with the initial velocities,

$$\dot{x}(0) = 0.101637$$

$$\dot{y}(0) = 0.472285$$

$$\dot{z}(0) = 0.818022 ,$$

obtained from the final iterate.

The sequence $\{X_n\}$ converged, within the accuracy of our computations, in three iterations with:

$$\rho(X_1,X_0) = 0.480116$$

$$\rho(X_2,X_1) = 0.133753$$

$$\rho(X_3,X_2) = 0.004375$$

$$\rho(X_4,X_3) = 0.000004 ,$$

where

$$\rho(X_{n+1},X_n) = \max_t |x_{n+1}(t) - x_n(t)| + \max_t |y_{n+1}(t) - y_n(t)| + \max_t |z_{n+1}(t) - z_n(t)| .$$

As a further check on the over-all accuracy the perturbing force was set to zero and the final value of the magnitude of the initial velocity was compared with that obtained by the closed form solution for the two-body problem. Within the accuracy of our computations these values were identical.

We note that we have simply and rapidly produced the numerical solution to a simple orbit determination problem, viz., given the position of a body at two distinct times, determine the time varying orbital elements of the body in the presence of perturbing forces. Solutions have also been produced even when the two points are exactly 180 degrees apart. In this case the straight line could not be used as a starting function since it is singular. However, a simple triangular path was sufficient to produce the characteristic rapid convergence.

LUNAR DESCENT EXAMPLE — MAXIMUM RANGE

A very simple variational problem was chosen for the second numerical example. This problem concerns the maximization of the translational range of a lunar vehicle during descent to rest from a hovering condition 1000 ft above the lunar surface. The time for the maneuver was fixed at 2.062 minutes.

For the purpose of generating this numerical example the following simplifying assumptions were made:

Constant thrust acceleration
Uniform gravitational field
Analysis restricted to two dimensions.

The problem then is reduced to finding the thrust steering angle time history which produces the maximum range in the given fixed time.

The associated boundary value problem may be obtained by the methods of Chapter 4 of Ref. 11, or by the Pontryagin maximum principle.[12] The resulting boundary value problem is given by the following nonlinear differential equations and boundary conditions :

$$\dot{u} = T \frac{\lambda_u}{(\lambda_u^2 + \lambda_v^2)^{\frac{1}{2}}} = f^{(1)} \quad ; \quad t \in [t_0, t_f]$$

$$\dot{v} = T \frac{\lambda_v}{(\lambda_u^2 + \lambda_v^2)^{\frac{1}{2}}} - g_M = f^{(2)}$$

$$\dot{y} = v = f^{(3)}$$

$$\dot{\lambda}_u = - 1 = f^{(4)}$$

$$\dot{\lambda}_v = - \lambda_y = f^{(5)}$$

$$\dot{\lambda}_y = 0 = f^{(6)}$$

$$u(t_0) = u_0 \qquad u(t_f) = u_f$$

$$v(t_0) = v_0 \qquad v(t_f) = v_f$$

$$y(t_0) = y_0 \qquad y(t_f) = y_f \; .$$

The state variable u is the indefinite integral of the range x, and y is the vertical height measured positive-up along the local vertical. The local gravitational constant g_M has the value appropriate to the moon. In addition, the adjoint variables have been scaled by putting λ_x at the initial time equal to one.

The unit of length was chosen equal to the initial altitude of 1000 ft and the local gravitational constant and vehicle mass were put equal to one. This resulted in the following normalized data for the problem:

$u_0 = 0.000$	$u_f = 0.000$	$x_0 = 0.000$
$v_0 = 0.000$	$v_f = 0.000$	
$y_0 = 1.000$	$y_f = 0.000$	
$T = 5.000$	$t_0 = 0.000$	
$g_M = 1.000$	$t_f = 9.000$	

This normalization resulted in a time unit of 13.70 seconds. A crude starting function $X_0(t)$ was chosen as follows:

$$u_0(t) \equiv 0$$

$$v_0(t) \equiv 0$$

$$y_0(t) = y_0 + \frac{y_f - y_0}{t_f - t_0} t$$

$$\lambda_{y_0}(t) \equiv c_3$$

$$\lambda_{u_0}(t) = c_1 - t$$

$$\lambda_{v_0}(t) = c_2 - c_3 t ,$$

where the three constants c_1, c_2, and c_3 correspond to an arbitrary estimate that the steering angle, measured from the local horizontal, should be initially zero, equal to $\pi/2$ at $t = \frac{t_f}{2}$, and slightly less than π at $t = t_f$.

The sequence $\{X_n\}$ for this case converged uniformly to an accuracy of 5 significant figures in six iterations. The total computer time (IBM 7094) required for this problem was 18 seconds. The desired final value of the range $x_f = 100,200$ ft was obtained from

$$x_f = \int_{t_0}^{t_f} u^*(t)\,dt \;,$$

where $u^*(t)$ results from the integration of the nonlinear state and Euler-Lagrange equations with a complete set of initial values taken from the final iterate. This final integration of the nonlinear equations also served as an over-all check on the solution.

LOW THRUST ORBITAL TRANSFER EXAMPLE — MINIMUM TIME

The third and final example concerns the problem of minimizing the transfer time of a low thrust ion rocket between the orbits of Earth and Mars. This problem involves additional complications over the previous problems, the most significant of which is the fact that the final value of the independent variable is no longer fixed.

To simplify the problem as much as possible the rocket's thrust level was assumed constant, and thus the single control variable is the thrust direction. Further, the orbits of Earth and Mars were assumed to be circular and coplanar, and the gravitational attractions of the two planets on the vehicle were neglected. The following system parameters for the low-thrust vehicle were adopted from Ref. 5:

Initial Mass, m_0	46.58 slugs
Specific Impulse	4700 sec
Propellant Consumption Rate, $\dot{m}$,	-6.937×10^{-7} slugs/sec
Thrust, T,	0.127 lb
Thrust/Initial Weight	0.9×10^{-4}

The equations of motion are given by:

Radial Velocity

$$\dot{r} = f^{(1)} = u \quad ,$$

Radial Acceleration

$$\dot{u} = f^{(2)} = \frac{v^2}{r} - \frac{k}{r^2} + \frac{T \sin \theta}{m_0 + \dot{m}t} \quad ,$$

Circumferential Acceleration

$$\dot{v} = f^{(3)} = - \frac{uv}{r} + \frac{T \cos \theta}{m_0 + \dot{m}t} \quad ,$$

where u and v are the radial and circumferential velocities respectively; r is the radius; and θ is the thrust direction angle measured from the local horizontal. All the initial and final values of the state variables were specified, and the quantity to be minimized was t_f, the final time. Since the method as previously outlined required a fixed final time, the procedure was altered to suit the minimum time problem. What follows is a brief description of the modified procedure and a discussion of the numerical results.

The two-point boundary value problem resulting from the Euler-Lagrange equations is given by :

$$\dot{r} = u = f^{(1)}$$

$$\dot{u} = \frac{v^2}{r} - \frac{k}{r^2} + a(t) \frac{\lambda_u}{\left(\lambda_u^2 + \lambda_v^2\right)^{\frac{1}{2}}} = f^{(2)}$$

$$\dot{v} = - \frac{uv}{r} + a(t) \frac{\lambda_v}{\left(\lambda_u^2 + \lambda_v^2\right)^{\frac{1}{2}}} = f^{(3)}$$

$$\dot{\lambda}_r = \left(\frac{v^2}{r^2} - 2\frac{k}{r^3}\right)\lambda_u - \frac{uv}{r^2}\lambda_v \qquad = f^{(4)}$$

$$\dot{\lambda}_u = -\lambda_r + \frac{v}{r}\lambda_v \qquad = f^{(5)}$$

$$\dot{\lambda}_v = -2\frac{v}{r}\lambda_u + \frac{u}{r}\lambda_v \qquad = f^{(6)}$$

where

$$a(t) = \frac{T}{m_0 + \dot{m}t},$$

and the boundary conditions are :

$t = 0$	$t = t_f$ (unspecified)
$r(0) = r_0$	$r(t_f) = r_f$
$u(0) = u_0$	$u(t_f) = u_f$
$v(0) = v_0$	$v(t_f) = v_f$.

This may be written as

$$\dot{X} = F(X,t)$$

where

$$X = \left(x^{(1)},\ldots,x^{(6)}\right)$$

$$F = \left(f^{(1)},\ldots,f^{(6)}\right)$$

and

$$x^{(1)}(t) = r(t) \quad , \quad x^{(2)}(t) = u(t) \quad , \quad x^{(3)}(t) = v(t)$$

$$x^{(4)}(t) = \lambda_r(t) \quad , \quad x^{(5)}(t) = \lambda_u(t) \quad , \quad x^{(6)}(t) = \lambda_v(t)$$

The method proceeds as before by solving the following sequence of linear two point problems

$$\dot{X}_{n+1} = J(X_n, t)[X_{n+1} - X_n] + F(X_n, t) \qquad n = 0,1,\ldots ,$$

where $J(X,t)$ is the Jacobian matrix of partial derivatives of the $f^{(i)}$ with respect to the $x^{(j)}$, $i = 1,\ldots,6$, $j = 1,\ldots,6$. A starting vector, $X_0(t)$, and an estimated final time, t_{f_0}, are assumed and the sequence of linear boundary value problems is solved numerically by the procedure outlined previously, with the following boundary values:

$$\begin{array}{ll} t = 0 & t = t_{f_k} \\ x_n^{(1)}(0) = r_n(0) = r_0 & \\ x_n^{(2)}(0) = u_n(0) = u_0 & x_n^{(2)}(t_f) = u_n(t_f) = u_f \\ x_n^{(3)}(0) = v_n(0) = v_0 & x_n^{(3)}(t_f) = v_n(t_f) = v_f \\ x_n^{(4)}(0) = \lambda_{r_n}(0) = 1 & \end{array}$$

$$n = 1,2,\ldots .$$

Setting $\lambda_r(0) = 1$ accomplished the scaling of the multipliers. The iteration proceeds until $\bar{\rho}(X_{n+1},X_n) \leq \beta$ where

$$\bar{\rho}(X_{n+1},X_n) = \sum_{i=1}^{6} \max_{t \in [0,t_{f_k}]} |x_{n+1}^{(i)} - x_n^{(i)}| .$$

At this stage the final time, t_{f_k}, is adjusted automatically according to the difference $[r_f - r(t_{f_k})]$ by a scalar application of the Newton-Raphson procedure as follows

$$t_{f_{k+1}} = t_{f_k} + \frac{(t_{f_k} - t_{f_{k-1}})}{r(t_{f_k}) - r(t_{f_{k-1}})}[r_f - r(t_{f_k})] ,$$

where the derivative of the final time t_f with respect to the final radial distance r_f has been obtained by a finite difference approximation. The above iteration on X_n now continues for the new final time $t_{f_{k+1}}$ until $\bar{\rho}$ is again $\leq \beta$. The over-all process proceeds until $\rho \leq \epsilon$ where

$$\rho = \bar{\rho} + \frac{1}{b}|t_{f_{k+1}} - t_{f_k}|$$

and b is a scaling factor. The corresponding iterate X_{n+1} is accepted as the solution to the minimum time problem, and a final check is run by integrating the nonlinear Euler-Lagrange equations with a complete set of initial conditions taken from the final iterate.

For the purpose of numerical precision the data for the sample problem were normalized to obtain

$r_0 = 1.000$	$v_f = .8098$
$r_f = 1.525$	$u_f = 0.000$
$k = 1.000$	$m_0 = 1.000$
$v_0 = 1.000$	$\dot{m} = -.07487$
$u_0 = 0.000$	$T = .1405$.

This resulted in a time unit of 58.18 days. The starting vector $X_0(t)$ was chosen rather crudely as follows:

$$t_{f_0} = 178.0 \text{ days, or } 3.060 \text{ of our time units}$$

$$x_0^{(1)}(t) = r_0(t) = r_0 + \frac{r_f - r_0}{t_{f_0}} t$$

$$x_0^{(2)}(t) = u_0(t) \equiv 0$$

$$x_0^{(3)}(t) = v_0(t) = \left(\frac{k}{r_0(t)}\right)^{\frac{1}{2}}$$

$$x_0^{(4)}(t) = \lambda_{r_0}(t) \equiv 1.000$$

$$x_0^{(5)}(t) = \lambda_{u_0}(t) \equiv \begin{cases} .5200 & \text{for } t \in [0, \tfrac{1}{2} t_{f_0}] \\ -.5000 & \text{for } t \in (\tfrac{1}{2} t_{f_0}, t_{f_0}] \end{cases}$$

$$x_0^{(6)}(t) = \lambda_{v_0}(t) \equiv \begin{cases} .3000 & \text{for } t \in [0, \tfrac{1}{2} t_{f_0}] \\ 0.000 & \text{for } t \in (\tfrac{1}{2} t_{f_0}, t_{f_0}] \end{cases} .$$

The final two starting functions $\lambda_{u_0}(t)$ and $\lambda_{v_0}(t)$ correspond to a control angle $\theta_0(t)$ which is constant at $60°$ above the local horizontal for the first half of the transit time, and constant inward along the local vertical for the remaining half of the transit time (see Fig. 2).

The sequence $\{X_n\}$ converged uniformly to an accuracy of 5 significant figures with 4 shifts of the final time in 13 total iterations. The resultant minimum time was found to be 193.2 days; in agreement with results previously obtained by gradient methods.[11]

The total computer time (IBM 7094) required was 36 seconds. Figure 2 illustrates the behavior of the control angle program, where $\theta_0(t)$ is the starting function, $\theta_1(t)$ through $\theta_4(t)$ correspond to the 4 shifts of the final time t_f, and $\theta^*(t)$ results from the integration of the nonlinear state and Euler-Lagrange equations with the initial values taken from the final iterate. The curves for $\theta_2(t)$, $\theta_3(t)$, and $\theta_4(t)$ lie, within our plotting accuracy, on the solution curve $\theta^*(t)$; except for the final segments as indicated on the figure.

We observe that for this particular example the approach just described is systematic, simple to apply, and yields rapid convergence from crude a priori starting functions.

By simple changes in the initial data, solutions were also generated for Earth to Venus and Earth to Jupiter transfers. The minimum times for these were 139.2 days and 478 [illegible] days respectively.

CONCLUSIONS

The numerical examples of this paper suggest that the Newton-Raphson operator technique may be a useful computational method for obtaining solutions to meaningful nonlinear boundary value problems; and in particular for obtaining extremals for variational problems. It may be of particular use in generating <u>families</u> of solutions for given variational problems with differing values for the relevant parameters; for in this case the solution for one set of parameters becomes the starting function for the succeeding problem. This implies that the desired family may be generated with reasonable computation time.

It should be noted that dividing the boundary conditions exactly in half was purely for convenience of discussion; the computational problem obviously simplifies if more conditions are known at one end than at the other. In addition, it is <u>not necessary</u> that the starting functions be continuous or meet the boundary conditions; all of the iterates, however, will have these properties. Also, although the examples shown have the terminal values of the state variables specified, this is not a necessary restriction. If a particular state coordinate is left unspecified, the transversality conditions require that the corresponding $\lambda(t_f)$ be zero. This simply changes the linear algebraic system to be solved for the coupling constants.

The solutions produced by this method satisfy the necessary conditions for optimality as given by the Pontryagin maximum principle,[12] and classically the Weierstrass and Clebsch necessary conditions. However, the questions of global optimality, and sufficiency, require further tests[2,13] and remain open.

We note certain reservations. Although it was possible, for the included examples, to obtain crude a priori starting functions sufficient to produce convergence, it is not clear that this will remain true for other more complex problems. If it should occur that starting functions sufficient for convergence are not easily obtainable then one might consider a hybrid approach, e.g., using a few steps of a gradient technique to produce the necessary starting functions.

Finally, we observe that application of this algorithm to problems with bounded control variables and/or state variable constraints requires further modification and extension of the technique. A sample problem with a state variable inequality constraint has been solved and will be reported at a later date; a problem with bounded control is presently under study.

REFERENCES

1. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., "Optimization and Control of Nonlinear Systems Using the Second Variation," J. SIAM Control, Ser. A, Vol. 1, No. 2, 193-223 (1963).

2. Scharmack, D. K., "A Modified Newton-Raphson Method for the Control Optimization Problem," AIAA Control and System Optimization Conference, Monterey, Calif. (Jan. 27-28, 1964).

3. Kelley, H. J., Kopp, R. E., Moyer, H. G., "A Trajectory Optimization Technique Based Upon the Theory of the Second Variation," AIAA Meeting on Astrodynamics, Yale Univ., New Haven, Conn. (Aug. 19-21, 1963).

4. Kopp, R., E., McGill, R., Moyer, H. G., Pinkham, G., "Several Trajectory Optimization Techniques," Conference on Computing Methods in Optimization Problems, UCLA, Los Angeles, Calif. (Jan. 1964).

5. Kantorovitch, L. V., Doklady Akad. Nauk SSSR (N.S.) 59, 1237-12[illegible]0 (1948).

6. Warga, J., "On a Class of Iterative Procedures for Solving Normal Systems of Ordinary Differential Equations," Jour. of Math. and Physics, Vol. XXXI, No. 4 (Jan. 1953).

7. Hestenes, M. R., "Numerical Methods of Obtaining Solutions of Fixed End Point Problems in the Calculus of Variations," Rand Corp. Rept. RM-102 (Aug. 1949).

8. Kalaba, R., "On Nonlinear Differential Equations, the Maximum Operation, and Monotone Convergence," Jour. of Math. and Mech., Vol. 8, No. 4, 519-574 (July 1959).

9. McGill, R., and Kenneth, P., "A Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary Value Systems," XIVth IAF Congress, Paris, France (Sept. 1963).

10. Kolmogorov, A. N., and Fomin, S. V., Elements of the Theory of Functions and Functional Analysis (Graylock Press, Rochester, N. Y., 1957), Vol. 1, Chap. II, pp. 43 ff.

11. Kelley, H. J., "Method of Gradients," Optimization Techniques, edited by G. Leitmann, (Academic Press, New York, 1962), Chap. 6, pp. 206-252.

12. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., The Mathematical Theory of Optimal Processes, translated from the Russian by K. N. Trirogoff, (Interscience Publishers, New York, 1962), Chap. II, p. 81.

13. Bliss, G. A., Lectures on the Calculus of Variations, (The Univ. of Chicago Press, Chicago, Ill., 1946), Chap. II, pp. 37 ff.

X \ t	0.	0.4	0.8	1.2	1.6	2.0
x_0	1.076000	0.860800	0.645600	0.430400	0.215200	0.
x_1	1.076000	1.015153	0.845061	0.610986	0.323847	0.
x_2	1.076000	1.048799	0.900816	0.657001	0.346085	0.
x_3	1.076000	1.049839	0.902586	0.658550	0.346867	0.
x_4	1.076000	1.049840	0.902587	0.658551	0.346868	0.
x^*	1.076000	1.049840	0.902587	0.658551	0.346868	0.
y_0	0.	0.115200	0.230400	0.345600	0.460800	0.576000
y_1	0.	0.172927	0.324202	0.447591	0.537713	0.576000
y_2	0.	0.184664	0.348339	0.475158	0.553667	0.576000
y_3	0.	0.185100	0.349180	0.476056	0.554172	0.576000
y_4	0.	0.185100	0.349180	0.476057	0.554173	0.576000
y^*	0.	0.185100	0.349180	0.476057	0.554173	0.576000
z_0	0.	0.199532	0.399064	0.598597	0.798129	0.997661
z_1	0.	0.299519	0.561534	0.775250	0.931347	0.997661
z_2	0.	0.319848	0.603341	0.822998	0.958980	0.997661
z_3	0.	0.320602	0.604800	0.824553	0.959854	0.997661
z_4	0.	0.320603	0.604798	0.824555	0.959855	0.997661
z^*	0.	0.320603	0.604798	0.824555	0.959855	0.997661
X / t	0.	0.4	0.8	1.2	1.6	2.0

Table 1 Time Histories for Intercept Example

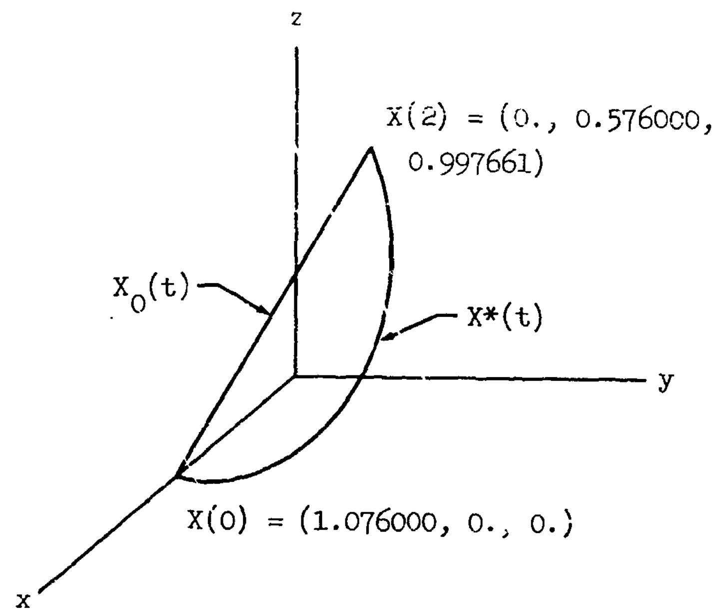

Figure 1. Schematic Diagram for Intercept Example

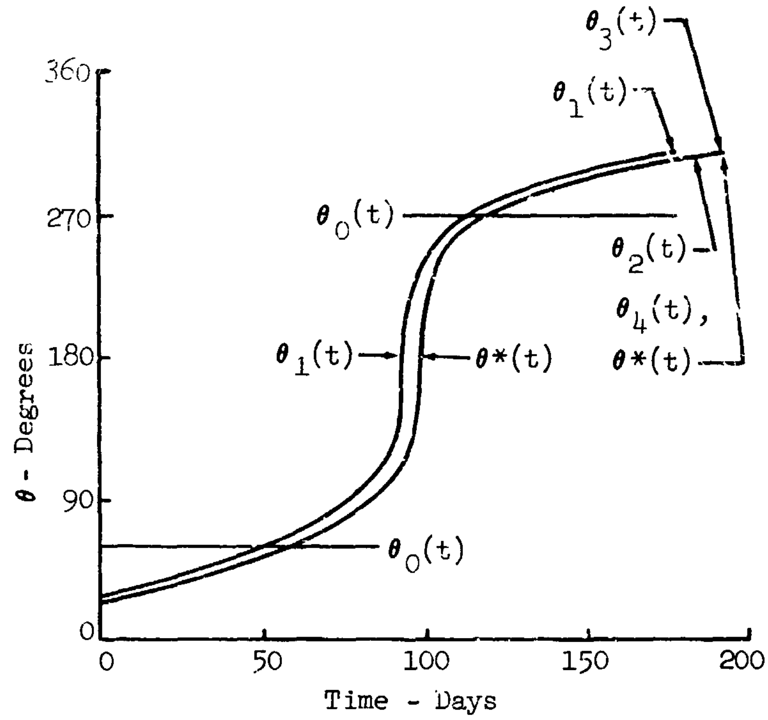

Figure 2. Control Angle Programs for Orbital Transfer Example