<table>
<thead>
<tr>
<th>TO:</th>
<th>Approved for public release; distribution is unlimited.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM:</td>
<td>Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 02 MAR 1965. Other requests shall be referred to Office of Naval Research, 875 North Randolph Street, Arlington, VA 22203-1995.</td>
</tr>
<tr>
<td>AUTHORITY</td>
<td>ONR ltr, 15 Jun 11977</td>
</tr>
</tbody>
</table>
NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.
Office of Naval Research
Contract Nonr-1866 (16)          NR-372-012

A SUCCESSIVE SWEEP METHOD
FOR SOLVING OPTIMAL PROGRAMMING PROBLEMS

By
Stephen R. McReynolds and Arthur E. Bryson, Jr.

March 2, 1965

Technical Report No. 463

Cruft Laboratory
Division of Engineering and Applied Physics
Harvard University * Cambridge, Massachusetts
A SUCCESSIVE SWEEP METHOD
FOR SOLVING OPTIMAL PROGRAMMING PROBLEMS

by
Stephen R. McReynolds and Arthur E. Bryson, Jr.

March 2, 1965

The research reported in this document was made possible through support extended to Cruft Laboratory, Harvard University, by the U. S. Army Research Office, the U. S. Air Force Office of Scientific Research, and the U. S. Office of Naval Research under the Joint Services Electronics Program by Contract Nonr-1866(16). Reproduction in whole or in part is permitted for any purpose of the United States Government.

Technical Report No. 463

Cruft Laboratory
Division of Engineering and Applied Physics
Harvard University
Cambridge, Massachusetts
A SUCCESSIVE SWEEP METHOD FOR SOLVING OPTIMAL PROGRAMMING PROBLEMS

Stephen R. McReynolds and Arthur E. Bryson, Jr.
Division of Engineering and Applied Physics
Harvard University

ABSTRACT

An automatic, finite-step numerical procedure is described for finding exact solutions to non-linear optimal programming problems. The procedure represents a unification and extension of the steepest-descent, and second variation techniques. The procedure requires the backward integration of the usual adjoint-vector differential equations plus certain matrix differential equations. These integrations correspond, in the ordinary calculus, to finding the first and second derivatives of the performance index respectively. The matrix equations arise from an Inhomogeneous Ricatti transformation, which generates a linear "feedback control law" that preserves the gradient histories, but permits changing them by controlled amounts, while also changing terminal conditions by controlled amounts. Thus, in a finite number of steps, the gradient histories can be made identically zero, as required for optimality, and the terminal conditions satisfied exactly. One forward plus one backward sweep, correspond to one step in the Newton-Raphson technique for finding maxima and minima in the ordinary calculus.

As by-products, the procedure produces (a) the functions needed to show that the program is, or is not, a local maximum (the generalized Jacobi test) and (b) the feedback gain programs for neighboring optimal paths to the same, or a slightly different, set of terminal conditions.

CLASS OF PROBLEMS TREATED

The method is applicable to a class of non-linear optimal programming problems where one wishes to determine control functions \( u(t) \) in \( t_0 \leq t \leq t_1 \) so as to minimize (or maximize) a performance index of the form

\[
J = \int_{t_0}^{t_1} L(x(t),u(t),t) dt + H(x(t),u(t),t)
\]

subject to the constraints

\[
\dot{x} = f(x(t),u(t),t)
\]

\( x(t_0) = x_0 \) is an n-component state vector \( u(t) \) is an m-component control vector

A further restriction on the class of problems treated in this paper is that we assume \( \frac{\partial^2 L}{\partial u^2} \) is a positive-definite (or negative-definite) matrix over the whole interval \( t_0 \leq t \leq t_1 \) where \( L \) is the variational Hamiltonian introduced in the next section.

The final time, \( t_1 \), may be given explicitly in Eqns. (4). For simplicity of presentation, we will first discuss the case where the final time \( t_1 \) is given explicitly.

CASE WHERE FINAL TIME IS GIVEN EXPLICITLY

In the usual manner we introduce the auxiliary scalar functions

\[
H(\lambda,x,u,t) = L(x,u,t) + \lambda^T f(x,u,t)
\]

\[
H_\lambda = \frac{\partial H}{\partial \lambda} \]

where \( \lambda(t) \) is an n-component vector of influence functions and \( v \) is a q-component constant vector. We regard \( u(t) \) as control functions and \( v \) as control parameters, and introduce a modified performance index \( J \)

\[
J = \int_{t_0}^{t_1} [H(\lambda(t),x(t),u(t),t)] dt
\]

subject to the constraints

\[
\dot{\lambda} = -H_\lambda \lambda - f^T \lambda - H_u u
\]

\( \lambda(t_1) = \delta \) specified

Suppose we arbitrarily choose some control functions \( u(t) \) and some control parameters \( v \), integrate Eqns. (2) forward with initial conditions (3), and Eqns. (8) backward with boundary conditions (10). In general, Eqns. (4) and (9) will not be satisfied. Now, consider a perturbation around this path:

\[
6x = f_x 6x + f_u 6u
\]

\[
\lambda = -H_\lambda 6x - f^T_x 6\lambda - H_u 6u
\]

\[
H_u u + f_x u + f_u u + \delta H_u u \]

\( \delta x(t_1) \) specified

\[
\delta x(t_0) \]

\[6\lambda \] specified

\[
\delta x(t) \]

We may regard (11)-(16) as a linear,
inhomogeneous two-point boundary value problem that determines the functions $\delta x(t)$, $\delta u(t)$, and the parameters $\delta v$ in terms of specified functions $\delta u(t)$ and specified parameters $\delta x(t_0)$ and $\delta v$. This is very close to the viewpoint taken by Merriam (Ref. 2) and Kelley, Kopp, and Moyer (Ref. 3).

To solve this two-point boundary value problem we may solve (13) for $\delta u(t)$ in terms of $\delta x(t)$, $\delta x(t_0)$, and $\delta H(t)$, provided $H(t)$ is non-singular:

$$\delta u = -H^{-1}u [-\delta H + H \delta x + f^T \delta x]$$

and, upon substituting (17) into (11) and (12), we obtain

$$\delta x = A \delta x + B \delta \lambda + v$$

where

$$A = f \left[ -f \frac{h}{u} \right]$$

$$B = f \left[ \frac{h}{u} \right]$$

$$C = -H \delta x + H \delta H + f$$

$$v = f \left[ \frac{h}{u} \right]$$

$$w = -H \delta u \frac{H}{u}$$

**THE INHOMOGENEOUS RICATTI TRANSFORMATION**

In view of Eqs. (15) and (16), let us introduce the following inhomogeneous Ricatti transformation (suggested in Refs. 4, 5):

$$\delta x(t) = F(t)\delta x(t) + H(t)\delta v + h(t)$$

$$\delta u^T(t)\delta x(t) + Q(t)\delta u + g(t)$$

where $\delta v$ and $\delta u$ are constant infinitesimal vectors, $F(t)$, $Q(t)$, and $h(t)$ are matrices, and $x(t)$ and $g(t)$ are vector functions. Now, differentiate (17) and (18) with respect to time:

$$\delta \lambda(t) = P(t)\delta x(t) + R(t)\delta v + h(t)$$

$$\delta u^T(t)\delta x(t) + Q(t)\delta u + g(t)$$

Using (25) in (18) gives

$$\delta \lambda = (A+BP)\delta x + BR \delta v + B \delta v + B \delta v$$

Equating (19) and (27) and using (25) and (29) to eliminate $\delta x$ and $\delta \lambda$, we have:

$$(-A^T + PA + PA^T)\delta x + [(A+PB)R+P] \delta v = 0$$

In a similar fashion, substitute (29) into (28):

$$(-A^T + PA + PA^T)\delta x + [(A+PB)R+P] \delta v = 0$$

Viewing (30) and (31) as identities, valid for arbitrary values of $\delta x$ and $\delta u$, it follows that the coefficients of $\delta x$ and $\delta u$ must vanish; this yields differential equations for $P$, $R$, $Q$, $h$, and $g$. Also, if we require that (30) and (31) be equivalent to (15) and (16) at the terminal time, we obtain boundary conditions for $P$, $R$, $Q$, $h$, and $g$:

$$P = A^T \delta H + H \delta x$$

$$R(t) = [A+PB]R + P$$

$$Q = [A^T + PA + PA^T]Q$$

$$h = \left[ f \left[ \frac{h}{u} \right] \right]$$

$$g = \left[ f \left[ \frac{h}{u} \right] \right]$$

Note that (32) is a nonlinear matrix differential equation (a matrix Ricatti equation), while (33) is a linear matrix differential equation using the solution of (32); (34) is a linear vector differential equation using the solution of (32), and (35) is a vector quadrature using the solution of (35).

By integrating (32)-(36) backward along with (3) from $t_0$ to $t_1$ (a "backward sweep") we generate all possible solutions to (11)-(13) that satisfy the terminal conditions (15)-(16). We may think of (25)-(26) as "boundary conditions" at time $t < t_1$ that are equivalent to the boundary conditions (15)-(16) at time $t = t_1$. Thus the boundary conditions at the terminal time are "sweep" backward to the initial time; a "forward sweep" then generates the required particular solution that also satisfies the initial conditions (14). This is precisely the approach taken by Bryson and Frazier (Ref. 6) to solve the linear smoothing problem except that the sweeps occur in the opposite order: the "forward sweep" is the Kalman-Bucy filter which involves a matrix Ricatti equation, and the "backward sweep" gives the smoothing solution that satisfies the terminal conditions.

After completing the backward sweep, the required values of $\delta u$ in terms of the desired infinitesimal changes $\delta x(t_0)$, $\delta x(t_0)$, and $\delta v$ can be obtained using (26):

$$\delta u = \left[ Q^{-1}(\delta u-R^T \delta x) \right]_{t=t_0}$$

Having these values of $\delta u$, we could, in principle, substitute them into (29) and integrate these equations forward with (25) and (17) to find $\delta x(t)$, $\delta x(t_0)$, and $\delta u(t)$ (a "forward sweep"). Alternatively, using (25) we could regard (17) as a linear feedback law for determining $\delta u(t)$:

$$\delta u(t) = \delta x(t) \delta x(t) + f^T \delta x(t)$$

$$+ f^T \delta x(t)$$

Note that $\delta u$ in (37) may be evaluated at the initial time $t = t_0$ as was done in (37) or we may evaluate it at several intermediate times in the manner of a sampled-data feedback law or we may evaluate it continuously as the continuous feedback law. If we do evaluate $\delta u$ continuously, then (38) becomes

McReynolds

and Bryson

2
\[ \delta u(t) = -H^{-1}((H_u + f^T(P-RQ^{-1}R^T))\delta x + [f^T(P-RQ^{-1}]\delta x
+ [-SH + f^T(h-RQ^{-1}y)]) \] (38a)

Now, the first term in square brackets on the right hand side of (38a) is a linear feedback on deviations \( \delta x(t) \) from the nominal state variable histories and will keep \( \delta H(t) = 0, \delta x = 0 \), for \( \delta x(t_0) \neq 0 \).

The second term in square brackets is the forcing function necessary to produce the desired changes \( \delta x \) while holding \( \delta H(t) = 0 \). The third term in square brackets is the forcing function necessary to produce the desired changes \( \delta H(t) \) while holding \( \delta x = 0 \); it is a linear functional of \( \delta H(t) \), and vanishes if \( \delta H(t) = 0 \). We could, therefore, integrate (2) forward (a "forward sweep"), using (38) in

\[ u(t) = u_{old}(t) + \delta u(t) \] (39)

\[ \delta x(t) = x(t) - x_{old}(t) \] (40)

The advantage of this procedure over previous gradient procedures is that we have separate, precise control over the desired changes \( \delta H(t) \) and \( \delta x \).

By repeating this forward-backward sweep several times we can bring \( H(t) \) and \( x(t) \) precisely to zero while increasing the performance index; the required number of steps depends on the successful range of linearization of (11)-(16). We suggest that if \( N \) steps are to be used, it would be reasonable to choose

\[ \delta u^{(r)}(t) = -c(t)_{u}^{(r-1)} \] (41)

\[ \delta x^{(r)}(t) = -c(t)_{x}^{(r-1)} \] (42)

where \( c(t) = r/N \) and \( r \) is the step number; in this way, larger and larger reductions in the "residuals" are taken each step and, on the last step, the whole remaining correction is made, bringing \( H(t) \) and \( x(t) \) precisely to zero.

LOCAL OPTIMALITY - GENERALIZED JACOBI TEST

When we have succeeded in bringing \( H(t) = 0 \) and \( x(t) = x_{old}(t) \), we have generated an admissible extremal path. For this case, the feedback law (38a) simplifies to:

\[ \delta u(t) = -H^{-1}([H_u + f^T(P-RQ^{-1}R^T)]\delta x(t) \] (43)

since \( \delta x = 0 \) and \( \delta H(t) = 0 \) implies that \( v = w = h = g = 0 \) (see Eqns. (23), (24), (35), (36)). If the symmetric \( n \times n \) matrix \( H_u \) is positive (or negative) definite and the symmetric \( n \times n \) matrix \( P-RQ^{-1}R^T \) is finite over the semi-open interval \( t_0 \leq t \leq t_1 \), then (43) indicates \( \delta u(t) = 0 \) if \( \delta x(t) = 0 \) and we are assured that we have generated a path that is at least a local optimal path. This is a generalized Jacobi test; if \( P-RQ^{-1}R^T \) becomes infinite at some point this is called a conjugate point to the terminal manifold \( \delta x(t_1,t_1) = 0 \). An extremal path is not an optimal path if it contains a conjugate point (see e.g. Ref. 4).

INTERPRETATION OF THE MATRICES \( P, Q, \) AND \( R \)

Let us define a return function \( V(u,v,x,t) \) which is the value of \( J \) in (7) when starting from state \( x \) at time \( t \leq t_1 \) using the control functions \( u(t) \) in (2) and the control parameters \( v \). Infinitesimal variations away from a given set of initial conditions, \( \delta x(t) \), and infinitesimal changes in the control parameters, \( \delta u \), while holding \( \delta H(t) = 0 \), will produce an infinitesimal change in the return function, \( \delta V \), given by:

\[ \delta V = \left[ \frac{\delta^2 V}{\delta x^2} \right] \delta x(t) + \left[ \frac{\delta^2 V}{\delta u \delta x} \right] \delta x(t) \delta u(t) + \left[ \frac{\delta^2 V}{\delta u^2} \right] \delta u(t)^2 \] (44)

From (44) it is clear that:

\[ \lambda^T(t) = \frac{\delta V}{\delta x(t)} \quad \psi^T = \frac{\delta V}{\delta u(t)} \] (45)

\[ P(t) = \frac{\frac{\delta^2 V}{\delta u \delta x}}{\frac{\delta^2 V}{\delta x^2}} \quad R = \frac{\frac{\delta^2 V}{\delta u^2}}{\frac{\delta^2 V}{\delta x^2}} \] (46)

From (26), or (45)-(46), we can also write

\[ \frac{k(t)}{\delta x(t)} \quad \psi(t) = \frac{\delta V}{\delta u(t)} \] (47)

and we note these quantities are similar to the steepest-ascent quantities \( \lambda^{(0)}(t) \) and \( \psi^{(0)}(t) \) of Bryson and Denham (Ref. 7).

If the path is extremal \( (H(t) = 0) \) and satisfies the terminal conditions \( \delta x(t) = 0 \), then \( V = V(x,t) \) is the optimal return function of Hamilton-Jacobi-Bellman theory (see e.g. Ref. 8).

Equation (44), using (26) with \( \delta x = 0 \), \( x = 0 \), to eliminate \( \delta u \) becomes

\[ \delta V = \lambda^T \delta x + \frac{1}{2} \delta x^T \left[ \frac{\delta^2 V}{\delta x^2} \right] \delta x \] (47)

which gives the infinitesimal change in the optimal return function for infinitesimal changes in the initial conditions \( \delta x(t) \) holding the final conditions constant \( \delta H(t) = 0 \).

SUMMARY FOR CASE WHERE FINAL TIME IS GIVEN EXPLICITLY

(A) Estimate the control functions \( u(t) \) and integrate \( \dot{x} = f(x,u,t) \) forward with given values of \( x(t_0) \). Record the constants \( \delta x(t_0,t_1) \), and the functions \( u(t), x(t) \).

(B) Estimate the control parameters \( v \) and \( \lambda = -f^T x \) backward with \( \lambda(t_1) = [\delta x + f^T u]_{t=t_1} \), using \( u(t), x(t) \) to evaluate \( f^T x(u(t),u(t),t) \).

Calculate \( H = \lambda^T f \) and its derivatives \( H_u, H_u^2 \)

McReynolds and Bryson
where
\[ H_{uu} \text{ and } H_{xt} \text{ as you go.} \]
\[ H_{uu}^\dagger \text{ must also be calculated.} \]
\[ H_{uu} \text{ is positive (or negative) definite. If } H_{uu} \text{ does not satisfy the appropriate condition, better estimates of } u(t) \text{ and } v \text{ are required in (A).} \]

Simultaneously with (B), integrate Eqs. (32)-(36) for \( P, Q, R, h, \) and \( g \) backward, using the derivatives of \( H \) from (B) and \( \delta H(t) \) from (41). Record the forcing functions
\[ H_{uu}^{-1}(t)[-\delta u(t) + H_{ux}^T(t)u(t)] = U(t) \]
and the feedback gains
\[ H_{uu}^{-1}(t)[H_{ux}(t) + H_{xt}^T(t)P(t)] = K(t) \]
\[ \delta H(t) = H_{uu}^{-1}(t)R(t) = L(t) \]

(D) Determine and record the parameters \( dv \) from (37), i.e.
\[ dv = Q^T(t)[\delta v - g(t) - \delta H(t)] \]

(E) Repeat (A) using the improved estimates of \( u(t) \)
\[ u(t) = u_{old}(t) + U(t) - K(t)[x(t) - x_{old}(t)] - L(t)dv \]

(F) Repeat (B), (C), and (D) using the improved estimates \( u, v \),
\[ v = v_{old} + dv \]

(G) Repeat (E) and (F) until \( H_u(t) = 0 \),
\[ v(x(t_0), t_0) = 0 \]

CASE WHERE FINAL TIME IS GIVEN IMPLICITLY

If the final time, \( t_f \), is given implicitly in (4), then it is necessary to estimate \( t_f \) for the first forward sweep, in addition to \( u(t) \) and \( v \). A few additional equations must be integrated on the backward sweep in order to determine the required \( \delta t_f \) for the next forward sweep.

The development is the same as in the previous case through Eqn. (10); at that point an additional necessary condition is required to determine the final time, namely the transversality condition
\[ g(x(t_f), t_f) = [\xi x + \xi v + \xi u] = 0 \]

The development is again the same up to Eqs. (15) and (16) which are replaced by the following:
\[ \delta x(t_1) = x_{\xi x}(t_1), m_{\xi x}(t_1), n_{\xi x}(t_1) \]
\[ \delta v(t_1) = v_{\xi v}(t_1), m_{\xi v}(t_1), n_{\xi v}(t_1) \]
\[ \delta t_1 = m_{\xi t}(t_1), n_{\xi t}(t_1), s_{\xi t}(t_1) \]
where
\[ m_{\xi x}(t_1) = [H_{uu}^{-1}(t) + H_{ux}^T(t)P(t)] \]
\[ m_{\xi v}(t_1) = -H_{uu}^{-1}(t)H_{ux}^T(t) \]
\[ m_{\xi t}(t_1) = \delta t_1 \]
\[ n_{\xi x}(t_1) = -H_{uu}^{-1}(t)H_{ux}^T(t) \]
\[ n_{\xi v}(t_1) = \delta v(t_1) \]
\[ n_{\xi t}(t_1) = \delta t_1 \]
\[ s_{\xi t}(t_1) = \delta t_1 \]

\[ \alpha(t_1) = \delta t_1 \]

and
\[ \delta Q = \delta Q(t) \]

\[ \delta P = \delta P(t) \]

\[ \delta U = \delta U(t) \]

\[ \delta v = \delta v(t) \]

\[ \delta t = \delta t(t) \]

\[ \delta u = \delta u(t) \]

\[ \delta H = \delta H(t) \]

\[ \delta H_{uu} = \delta H_{uu}(t) \]

\[ \delta H_{ux} = \delta H_{ux}(t) \]

\[ \delta H_{xt} = \delta H_{xt}(t) \]

Equations (17)-(24) are still applicable but, in view of (49)-(51), the inhomogeneous Riccati transformation beginning at (25) must be generalized to the following:
\[ \delta x(t) = \delta x(t), \delta v(t), \delta u(t) \]
\[ \delta v(t) = \delta v(t), \delta u(t), \delta x(t) \]
\[ \delta t(t) = \delta t(t), \delta v(t), \delta u(t) \]

Differentiating (55)-(57) with respect to time, using the fact that \( dv, dl, dv, \) and \( dt \) are constants, we obtain
\[ \delta x = A(x) + P(t) + \delta x \]
\[ \delta v = E(t) + \delta v \]
\[ \delta t = f(t) + \delta t \]

Using (55) in (18) gives
\[ 6x = A(x) + P(t) + \delta x \]
\[ 6v = E(t) + \delta v \]
\[ 6t = f(t) + \delta t \]

Using (55) in (19), together with (61), we can eliminate \( dl \) and \( 6x \) from (58)-(60), and obtain three equations like (30) and (31) in 6x, 6v, and 6t. These three equations are satisfied identically if we choose \( P, Q, R, h, \) and \( g \) to satisfy (32)-(36) and \( m, n, o \) to satisfy
\[ \dot{\delta} x + (A + \delta P) = 0 \]
\[ \dot{\delta} v = -R \delta m \]
\[ \dot{\delta} t = -m \delta t + \delta v \]

where the boundary conditions for \( m, n, o \) are given by (52)-(54). Note (62) is the same linear vector differential equation as (33) whereas (63) and (64) are simply quadratures.

If (62)-(65) are included in the backward integration sweep, then it is possible to solve for both \( dv \) and \( dt \) at \( t = t_0 \) using (56) and (57) where desired values of \( dv \) and \( dl \) for the next step are introduced. The desired value of \( \delta H_u(t) \) must be used in solving for \( h, g, \) and \( 8 \) from (35), (36), and (64).

REFERENCES


McReynolds and Bryson


APPENDIX

THE NEWTON-RAPHSON METHOD AND ITS APPLICATIONS TO ORDINARY CALCULUS PROBLEMS

In this Appendix the Newton-Raphson method is briefly stated. It will be seen that the Newton-Raphson method applied to optimization problems becomes a second-order iterative scheme which can be applied in the neighborhood of a non-singular optimum in order to obtain rapid convergence.

The formulation of second order steepest-ascent methods may be based upon a simple extension of the Newton-Raphson method used to solve a set of simultaneous nonlinear equations. Suppose one wishes to find an n-vector \( x = (x_1, \ldots, x_n) \) such that

\[ P(x) = 0 \quad P = (P_1, \ldots, P_n) \tag{A1} \]

The Newton-Raphson method generates a sequence \( (x^{(0)}, x^{(1)}, \ldots) \) by means of an iterative relation (A2).

\[ x^{(k+1)} = x^k - \left( \frac{\partial P}{\partial x} \right)^{-1} P \tag{A2} \]

The rationale for this is obtained by expanding \( P(x + dx) \) in a power series around \( x^k \).

\[ P(x + dx) = P(x^k) + \frac{\partial P}{\partial x} \bigg|_{x=x^k} dx + O(dx^2) \tag{A3} \]

Setting \( P(x + dx) = 0 \), one sees that

\[ dx = -P^{-1}(x^k) P + O(dx^2) \tag{A4} \]

by ignoring second and higher order terms on the right hand side of (A4) one obtains an estimate of the error in \( x \) within first order accuracy. Thus (A2) approximates the solution within a second order error. The method naturally assumes \( \frac{\partial P}{\partial x} \) to be nonsingular in the region containing \( (x^k) \) and the solution.

The Newton-Raphson method may be extended to finding a local maximum of a function of several variables \( f(x) \). If \( f \) is continuously differentiable, a local maximum \( x \) is characterized by being a solution to the following equations

\[ f_{x_i} = 0 \quad i = 1, \ldots, n \tag{A5} \]

Applying the Newton-Raphson method to these equations, one arrives at a second-order steepest-ascent method by merely identifying \( f_{x_i} \) with \( P_{x_i} \) in (A2).

The method may be readily extended to problems with constraints. Suppose the maximum of \( f \) is wanted subject to the added constraint

\[ g(x) = 0 \tag{A6} \]

Expanding (A6) around a nominal solution \( (x^k, x^l) \) one obtains the following set of linear, inhomogeneous equations to solve:

\[ 0 = (f_{x_i}^+ g_{x_i}) \bigg|_{x=x^k} (x^k, x^l) + g_{x_k} dx \tag{A7} \]

Solving (A8) yields corrections \( dx \) and \( dX \), and the second order steepest-ascent method becomes

\[ x^{k+1} = x^k + dx \tag{A9} \]

Several cautions must be exercised. One is that \( dx \) must be small in order to guarantee convergence, which implies that the original error should not be too big. Secondly, the nominal and the maximum must be non-singular and normal. This is necessary to guarantee the inversion of the basic equations. The non-singularity condition guarantees that one can solve for \( dx \). The normality condition guarantees that one can solve for \( dX \).

Thirdly, one should note that the second-order steepest-ascent method seeks out stationary solutions, regardless of whether they are local minima, local maxima, or saddle points. In order to be sure that the sequence converges to the desired extremum, the eigenvalues of the second derivative matrix must be checked. This can be seen for the problem without constraints by substituting (A2) with \( P = f_{x} \) into a power series for \( f \) around \( x^k \).

\[ f(x^{k+1}) = f(x^k) - \frac{1}{2} f_{xx}^{-1} f_{x}^2 + O(f_{xx}) \tag{A10} \]

In order to guarantee that \( f(x^{k+1}) > f(x^k) \), it is necessary to assume \( f_{xx} < 0 \).
**A SUCCESSIVE SWEEP METHOD FOR SOLVING OPTIMAL PROGRAMMING PROBLEMS**

An automatic, finite-step numerical procedure is described for finding exact solutions to nonlinear optimal programming problems. The procedure represents a unification and extension of the steepest-descent, and second variation techniques.

The procedure requires the backward integration of the usual adjoint-vector differential equations plus certain matrix differential equations. These integrations correspond, in the ordinary calculus, to finding the first and second derivatives of the performance index respectively. The matrix equations arise from an inhomogeneous Ricatti transformation, which generates a linear "feedback control law" that preserves the gradient histories, \( H(t) \), on the next step or permits changing them by controlled amounts, while also changing terminal conditions by controlled amounts. Thus, in a finite number of steps, the gradient histories can be made identically zero, as required for optimality, and the terminal conditions satisfied exactly. One forward plus one backward sweep correspond to one step in the Newton-Raphson technique for finding maxima and minima in the ordinary calculus.
Unclassified

<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization</td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>Calculus of variations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimal programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerical analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ricatti transformation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Newton-Raphson technique</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback control</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (corporate author) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.12 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all capital should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of the report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures. I.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b. c. & d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

   (1) "Qualified requesters may obtain copies of this report from DDC."

   (2) "Foreign announcement and dissemination of this report by DDC is not authorized."

   (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through"

   (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through"

   (5) "All distribution of this report is controlled. Qualified DDC users shall request through"

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (funding for) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

    It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (S), (U), (C) or (U).

    There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

Unclassified
As by-products, the procedure produces: (a) the functions needed to show that the program is, or is not, a local maximum, and (b) the feedback gain programs for neighboring optimal paths to the same, or a slightly different, set of terminal conditions.  (Authors)