# Generalized Perturbation Method for Solving Two-Point Boundary-Value Problems

R. G. GOTTLIEB*
*Lewis Industries, Kansas City, Mo.*

AND

D. G. HULL† AND W. T. FOWLER†
*The University of Texas at Austin, Austin, Texas*

**A method for solving two-point boundary-value problems is presented, and since it contains the standard perturbation method as a subcase, it is referred to as the generalized perturbation method. The philosophy of the method is to iterate on the histories of the variables whose initial values are unknown. For problems of a physical origin, it is not difficult to make reasonable guesses for these functions. Because variable histories are being used and not just the initial values, the generalized perturbation method is much less sensitive to the guesses than the standard perturbation method. More storage and computation per iteration is required, but the number of iterations may be considerably less.**

## Introduction

HISTORICALLY, the generalized perturbation method is a result of research on the computation of nonthrusting spacecraft trajectories. In Ref. 1, the problem of computing the free-fall trajectory of a spacecraft from a given initial time and position to a given final time and position is approached from the point of view of Hamilton's principle. This principle states that the motion is such that the integral of the Langrangian—difference of kinetic and potential energies—is an extremal. The extremal problem is recast in the optimal control format in which the velocity components are the control variables. Then, the first-order numerical optimization method discussed in Ref. 2 is applied to solve a trajectory problem in Earth-moon space. In Ref. 3, a second-order optimization method is developed and applied to the trajectory problem. Also, it is shown that the methods of Refs. 1 and 3 can be obtained directly from the equations of motion, that is, without using Hamilton's principle. As a result of the previous investigations, it is apparent that the approach discussed in Refs. 1 and 3 can be applied to the solution of any two-point boundary-value problem regardless of its origin. Since the method contains the perturbation method[4] as a subcase, it is called the generalized perturbation method.

## Generalized Perturbation Method

Consider a two-point boundary-value problem with system equations

$$\begin{aligned} \dot{x} &= f(t, x, v, a) \\ \dot{v} &= g(t, x, v, a) \end{aligned} \tag{1}$$

where $x$ is a $p$ vector, $v$ is an $n$-$p$ vector, and $a$ as a $q$ vector of parameters. With regard to the boundary conditions, it is assumed that

$$\begin{aligned} t_o &= 0, \quad x_o \equiv \text{given} \\ t_f &= 1, \quad h(x_f, v_f, a) = 0 \end{aligned} \tag{2}$$

where $h$ is an $n-p+q$ vector of final conditions. A normalized time is used for the independent variable, so that if the actual final time is unspecified, it appears as one of the parameters. It is possible to use dimensional time, but extrapolation is required if the change in final time from one iteration to another is positive.

The philosophy of the generalized perturbation method is to guess the functions $v(t)$ and the parameters $a$ and to set up an iterative procedure which drives the guesses to the values that satisfy the differential equations (1) and boundary conditions (2). In order to make the derivation of the algorithm clearer, Eqs. (1) are rewritten as

$$\dot{x} = f(t, x, u, a) \tag{3}$$

$$\dot{v} = g(t, x, u, a) \tag{4}$$

$$\dot{v} - u = 0 \tag{5}$$

where the guess for $v(t)$ has been denoted by $u(t)$. In the terminology of optimal control theory, it is apparent that the functions $u$ can be interpreted as control variables and Eqs. (5) as optimality conditions.

Once a guess has been made for $u(t)$ and $a$, $x(t)$ and $v(t)$ follow from Eqs. (3), (4) and the boundary conditions $t_o = 0$, $x_o \equiv$ given, $v_o = u_o$ and $t_f = 1$. However, because of the arbitrary choice for $u(t)$ and $a$, it is to be expected that $v - u \neq 0$ and $h \neq 0$. The effect of a variation in the chosen $u(t)$ and $a$ can be obtained by variational differentiation of the relevant equations, that is,

$$\delta\dot{x} = f_x \delta x + f_u \delta u + f_a \delta a \tag{6}$$

$$\delta\dot{v} = g_x \delta x + g_u \delta u + g_a \delta a \tag{7}$$

$$\delta(v - u) = \delta v - \delta u \tag{8}$$

$$\delta h = h_{x_f} \delta x_f + h_{v_f} \delta v_f + h_a \delta a \tag{9}$$

where, since $t_f = 1$, partial variations ($t = \text{const}$) and total variations can be identified by the same symbol $\delta$. Because it is desired to have $v - u = 0$ and $h = 0$ after the iteration, the following relations hold:

$$\delta(v - u) = -P(v - u), \quad \delta h = -Qh \tag{10}$$

The percentages $P$ and $Q$ are introduced to scale the amount of correction requested since the full correction predicted by linearized equations may lead to divergence. The quantities $P$ and $Q$ may have to be small for the first few iterations, but near convergence, they should be made unity.

From Eqs. (8) and (10), it is seen that

$$\delta u = \delta v + P(v - u) \tag{11}$$

so that the differential equations (6) and (7) can be rewritten as

$$\begin{aligned} \delta\dot{x} &= f_x \delta x + f_u \delta v + f_a \delta a + P f_u (v - u) \\ \delta\dot{v} &= g_x \delta x + g_u \delta v + g_a \delta a + P g_u (v - u) \end{aligned} \tag{12}$$

Received July 30, 1973; revision received December 3, 1973.
Index category: Navigation, Control and Guidance Theory.

* Vice-President. Member AIAA.
† Associate Professor, Department of Aerospace Engineering and Engineering Mechanics. Member AIAA.

**Table 1 Iteration sequence, fixed-step integration[a]**

| Iteration | $\lvert\delta u_1\rvert_{\max}$ | $\lvert\delta u_2\rvert_{\max}$ | $\lvert\delta x_{1f}\rvert$ | $\lvert\delta x_{2f}\rvert$ |
|---|---|---|---|---|
| 1 | 1.30 E+1 | 9.99 E+0 | 2.46 E−3 | 7.86 E−3 |
| 2 | 1.03 E+1 | 8.74 E+0 | 8.62 E−5 | 1.78 E−4 |
| 3 | 2.75 E+0 | 2.00 E+0 | 6.20 E−6 | 7.93 E−6 |
| 4 | 1.35 E+0 | 1.12 E+0 | 5.24 E−6 | 7.12 E−6 |
| 5 | 1.54 E−1 | 2.63 E−1 | 9.06 E−6 | 1.04 E−5 |
| 6 | 1.22 E−2 | 1.79 E−2 | 8.67 E−6 | 1.13 E−5 |
| 7 | 9.96 E−6 | 2.54 E−5 | 8.66 E−6 | 1.13 E−5 |
| 8 | 2.26 E−8 | 1.45 E−8 | 8.66 E−6 | 1.13 E−5 |

[a] 2400 steps.

At this point, it is assumed that

$$\begin{aligned}\delta x &= A(t)\,\delta v_o + C(t)\,\delta a + M(t)\\ \delta v &= B(t)\,\delta v_o + D(t)\,\delta a + N(t)\end{aligned} \tag{13}$$

with

$$A_o = 0, \quad B_o = I, \quad C_o = 0, \quad D_o = 0, \quad M_o = 0, \quad N_o = 0 \tag{14}$$

to guarantee satisfaction of the initial conditions (2). Substituting Eqs. (13) into Eqs. (12) and observing that the resulting equations must hold for arbitrary $\delta v_o$ and $\delta a$ leads to the following differential equations for the unknown matrices:

$$\begin{aligned}\dot{A} &= f_x A + f_u B\\ \dot{B} &= g_x A + g_u B\\ \dot{C} &= f_x C + f_u D + f_a\\ \dot{D} &= g_x C + g_u D + g_a\\ \dot{M} &= f_x M + f_u N + P f_u (v-u)\\ \dot{N} &= g_x M + g_u N + P g_u (v-u)\end{aligned} \tag{15}$$

whose initial conditions are given by Eqs. (14). Now, Eqs. (9) and (13) can be combined to produce values for $\delta v_o$ and $\delta a$, that is,

$$\begin{bmatrix}\delta v_o\\ \delta a\end{bmatrix} = -\left[h_{x_f} A_f + h_{v_f} B_f \,\vdots\, h_{x_f} C_f + h_{v_f} D_f + h_a\right]^{-1} \times (Qh + h_{x_f} M_f + h_{v_f} N_f) \tag{16}$$

whereas Eqs. (11) and (13) lead to

$$\delta u = B\,\delta v_o + D\,\delta a + N + P(v-u) \tag{17}$$

The following is a summary of the computational procedure for the generalized perturbation method: 1) Guess $u(t)$ and $a$; store $u(t)$. 2) Integrate the differential equations (3, 4, and 15), subject to the initial conditions $t_o = 0$, $x_o \equiv$ given, $v_o = u_o$ and Eqs. (14) and the stopping condition $t_f = 1$. Store $v(t)$, $B(t)$, $D(t)$, and $N(t)$. 3) Compute $\delta v_o$, $\delta a$ and $\delta u$ from Eqs. (16) and (17). 4) Using $u = u + \delta u$ and $a = a + \delta a$, go to step 2. The iteration process is stopped when the variations $\delta u$ and $\delta a$ and the end condition dissatisfactions become sufficiently small.

It should be pointed out that, if one sets $u = v$ in the preceding equations, the generalized perturbation method reduces to the standard perturbation method. Hence, a computer program of the generalized perturbation method also contains the standard perturbation method.

## Example

In this section, the generalized perturbation method is used to find the coasting trajectory of a spacecraft going from a fixed point near the Earth to a fixed point near the moon in a given time. If the Earth-moon-spacecraft system is approximated by the circular restricted three-body model, the nondimensional equations of motion in inertial coordinates are given by

$$\begin{aligned}\dot{x}_1 &= v_1\\ \dot{x}_2 &= v_2\\ \dot{v}_1 &= -(1-\mu)[x_1 + \mu\cos t]r_e^{-3} - \mu[x_1 - (1-\mu)\cos t]r_m^{-3}\\ \dot{v}_2 &= -(1-\mu)[x_2 + \mu\sin t]r_e^{-3} - \mu[x_2 - (1-\mu)\sin t]r_m^{-3}\end{aligned} \tag{18}$$

where

$$\begin{aligned}r_e^2 &= [x_1 + \mu\cos t]^2 + [x_2 + \mu\sin t]^2\\ r_m^2 &= [x_1 - (1-\mu)\cos t]^2 + [x_2 - (1-\mu)\sin t]^2\end{aligned} \tag{19}$$

and where $\mu = 0.012$. The chosen end conditions

$$\begin{aligned}t_o &= 0, \quad x_{1o} = -0.0272244, \quad x_{2o} = -0.00759068\\ t_f &= 0.0701248, \quad x_{1f} = 0.758519, \quad x_{2f} = 0.640512\end{aligned} \tag{20}$$

are characteristic of an Apollo mission, that is, transfer from a 100-mile Earth orbit to a 50-mile moon orbit in 75 hr. Since the final time is fixed, it is not necessary to normalize the time.

Since the quantities being guessed are velocity distributions and since the position equations are so simple, it is easy, but not necessary, to choose velocity histories which satisfy the final conditions. The velocity histories used in this study are the following:

$$\begin{aligned}u_1 &= (\dot{\xi} - \eta)\cos t - (\xi + \dot{\eta})\sin t\\ u_2 &= (\xi + \dot{\eta})\cos t + (\dot{\xi} - \eta)\sin t\end{aligned} \tag{21}$$

where

$$\begin{aligned}\xi &= -0.0272244 + 2.9t(1 - t/2t_f)\\ \eta &= -0.00759068 - (5t_f/2\pi)\sin(2\pi t/t_f)\end{aligned} \tag{22}$$

are the velocity histories in rotating coordinates.

In this study, the quantity $P$ is chosen to be $P = 0.7$ for three iterations and $P = 1.0$ for the remaining iterations. For the velocity guesses used here, it is possible to use $P = 1.0$ for all iterations, but the number of iterations increases. Since the velocity guesses satisfy the final conditions, $Q = 1$ is used all the time.

The trajectory is first obtained using a fixed stepsize integrator, that is, a fourth-order Adams-Moulton, Adams-Bashforth predictor-corrector with a fourth-order Runge-Kutta starter. Computations have been made for different numbers of steps from 300 to 2400. Integration accuracy for 300 steps is not good enough to achieve convergence ($\lvert\delta u_k\rvert_{\max} < 10^{-6}$), so these results are not shown in the tables. Table 1 shows the iteration sequence for 2400 steps, and Table 2 shows the effect of stepsize on the converged results. Table 4 shows that the cost in computer time and storage required to achieve the level of accuracy obtained with 2400 steps is high. Finally, the effect of stepsize on the converged values of the initial velocity components is presented in Table 5 along with the results of a very accurate computation.

Because the number of integration steps required to achieve an accurate solution with a fixed-step integrator can be large, the use of a variable-step integrator to minimize the number of steps has been investigated. The use of a variable-step integrator requires an interpolation method because the step pattern differs from iteration to iteration. In this connection, the trajectory has

**Table 2 Effect of stepsize on convergence[a]**

| Steps | $\lvert\delta u_1\rvert_{\max}$ | $\lvert\delta u_2\rvert_{\max}$ | $\lvert\delta x_{1f}\rvert$ | $\lvert\delta x_{2f}\rvert$ |
|---|---|---|---|---|
| 600 | 8.07 E−7 | 3.77 E−7 | 4.53 E−4 | 4.39 E−5 |
| 1200 | 3.61 E−7 | 1.91 E−7 | 8.74 E−5 | 3.83 E−5 |
| 2400 | 2.26 E−8 | 1.45 E−8 | 8.66 E−6 | 1.13 E−6 |

[a] Convergence: $\lvert\delta u_k\rvert_{\max} < 10^{-6}$.

**Table 3 Iteration sequence, variable-step integration**

| Iteration | $\lvert\delta u_1\rvert_{\max}$ | $\lvert\delta u_2\rvert_{\max}$ | $\lvert\delta x_{1f}\rvert$ | $\lvert\delta x_{2f}\rvert$ |
|---|---|---|---|---|
| 1 | 1.36 E+1 | 1.06 E+1 | 2.46 E−3 | 7.86 E−3 |
| 2 | 1.06 E+1 | 5.77 E+0 | 2.15 E−7 | 1.04 E−7 |
| 3 | 2.90 E+0 | 2.03 E+0 | 3.19 E−7 | 1.48 E−7 |
| 4 | 1.41 E+0 | 1.12 E+0 | 6.53 E−8 | 1.82 E−7 |
| 5 | 1.55 E−1 | 2.63 E−1 | 6.19 E−8 | 4.88 E−8 |
| 6 | 1.25 E−2 | 1.81 E−2 | 7.45 E−8 | 5.63 E−8 |
| 7 | 3.28 E−5 | 4.82 E−5 | 1.25 E−9 | 5.67 E−8 |
| 8 | 4.94 E−9 | 2.08 E−9 | 5.66 E−9 | 5.59 E−8 |

**Table 4 Comparison of computation characteristics**

| Steps | Iterations | Time per iteration | Storage |
|---|---|---|---|
| Fixed-step | | | |
| 600 | 9 | 1.7 sec | $<55{,}000_8$[a] |
| 1200 | 9 | 3.5 sec | $<55{,}000_8$[a] |
| 2400 | 8 | 7.2 sec | $70{,}000_8$[a] |
| Variable-step | | | |
| 175–225 | 8 | 2.1 sec | $<55{,}000_8$[a] |

[a] Minimum field length required to load UT CDC 6600.

also been obtained using Runge-Kutta-Fehlberg 3(4) integration in combination with cubic-spline interpolation. The orders of the integration and interpolation have been chosen to match the order of the fixed-step integrator so that comparisons between the variable-step and fixed-step integration would be fair. The RKF 3(4) integrator uses the difference of a third-order and a fourth-order series representation as an approximation of the local truncation error. The predicted stepsize is based on the greater of the maximum relative truncation error for variables whose values are greater than unity and the maximum absolute truncation error for variables whose values are less than unity. A step is accepted if the maximum truncation error is less than a specified tolerance ($10^{-6}$ in this study). The sequence of iterations is shown in Table 3, and a comparison with fixed-step integration is contained in Tables 4 and 5. The comparisons indicate that, from a results point of view, variable-step integration is better than fixed-step integration. On the other hand, the programing of the variable-step integration is more complex because of the need for interpolation.

## Discussion and Conclusions

There are two commonly used methods for solving two-point boundary-value problems: the perturbation method (or the shooting method) and quasi-linearization (or the generalized Newton-Raphson method).[4] For the formei, one guesses the missing initial conditions, integrates nonlinear state equations and linear perturbation equations, and computes changes in the initial guesses which drive the dissatisfactions in the final conditions to zero. For the latter, one guesses histories for all the state variables, integrates linear state equations and linear perturbation equations, and computes changes in the state histories which drive the dissatisfactions in the final conditions and state equations to zero. Compared with quasi-linearization, the perturbation method is easier to program, requires less storage, and uses less computer time per iteration, but it can be very sensitive, thereby requiring many more iterations to achieve convergence. On the other hand, when convergence is achieved with quasi-linearization, one must then integrate the nonlinear state equations to verify the solution of the problem.

The generalized perturbation method combines the good qualities of each of the previous methods. Here, one guesses histories for those state variables whose initial conditions are not prescribed, integrates nonlinear state equations and linear perturbation equations, and computes changes in the state guesses which drive the dissatisfactions in final conditions and state equations to zero. In each category—programing, storage, computer time per iteration, sensitivity—the characteristics of the generalized perturbation method are between those of standard perturbation and quasi-linearization. Furthermore, the generalized perturbation method contains the standard perturbation method as a particular case, and in the final iterations near convergence where $u \cong v$, the generalized perturbation method is actually operating like the standard perturbation method. Hence, the generalized perturbation method behaves like quasi-linearization for the first few iterations and like the standard perturbation method for the final few iterations.

**Table 5 Comparison of converged results**

| Integration method | $v_{1o}$ | $v_{2o}$ |
|---|---|---|
| Fixed-step | | |
| 600 | 6.347153809 | −8.580619156 |
| 1200 | 6.514071582 | −8.501568599 |
| 2400 | 6.516054137 | −8.507016420 |
| Variable-step | 6.515404330 | −8.507174589 |
| Accurate values | 6.515405689 | −8.507171407 |

It should be pointed out that, if the two-point boundary-value problem being solved originates from an optimal control problem, the generalized perturbation method requires the taking and coding of many second derivatives. One of the complaints along this line is that it is very difficult to do this correctly for a practical problem because of the magnitude of the effort. There are, however, at least two ways in which the computer can be used to check derivatives. One is to compute all partials numerically and compare their values with those obtained analytically; and the other is to generate the partials by non-numeric computation, compute their values, and compare with those obtained analytically.

Finally, no attempt has been made to implement automated procedures for selecting $P$ and $Q$, although several exist. Basically, there are three types of procedures: a) searches to minimize the error after the step is taken, b) scaling to guarantee a decrease in the error after the step is taken, and c) percent correction which limits the magnitude of the relative change. Furthermore, these procedures can be employed in two different ways: a) $P \neq Q$ and b) $P = Q$. In general, $P$ controls the error in the differential equations, and $Q$ controls the error in the end conditions. Hence, if $P \neq Q$, separate error criteria must be established for the differential equations and the end conditions; whereas, if $P = Q$, the criteria are combined into one. A discussion of searches, scaling, and error criteria for the case where $P = Q$ is presented in Ref. 5 and the percent-correction approach is presented in Ref. 6. If one is able to guess $u(t)$ and $a$ such that $h \cong 0$, it seems logical to set $Q = 1$ to keep $h \cong 0$ while the error in the differential equations is being reduced ($0 < P \leqq 1$). For the problem considered here (Earth-moon transfer), it soon became evident that convergence could be achieved with $P = 1$ for all iterations. Since the number of iterations could be reduced by letting $P = 0.7$ for several iterations, it is possible that the number of iterations can be reduced further by employing one of the previous techniques for selecting $P$ and $Q$.

## References

[1] Hull, D. G., Fowler, W. T., and Gottlieb, R. G., "Calculation of Free-Fall Trajectories Using Numerical Optimization Methods," *Journal of the Astronautical Sciences*, Vol. 19, No. 5, 1972, pp. 390–398.

[2] Gottlieb, R. G., "Rapid Convergence to Optimum Solutions Using a Min-H Strategy," *AIAA Journal*, Vol. 5, No. 2, Feb. 1967, pp. 322–329.

[3] Hull, D. G., "On the Solution of Two-Point Boundary-Value Problems Associated with Spacecraft Trajectories," Rept. 1031, 1971, The Univ. of Texas at Austin, Applied Mechanics Research Lab., Austin, Texas.

[4] Tapley, B. D. and Lewallen, J. M., "Comparison of Several Numerical Optimization Methods," *Journal of Optimization Theory and Applications*, Vol. 1, No. 1, 1967, pp. 1–33.

[5] Miele, A. and Iyer, R. R., "Modified Quasilinearization Method for Solving Nonlinear Two-Point Boundary-Value Problems," *Journal of Mathematical Analysis and Applications*, Vol. 36, No. 3, 1971, pp. 674–692.

[6] Lastman, G. J. and Tapley, B. D., "Optimization of Nonlinear Systems with Inequality Constraints Explicitly Containing the Control," *International Journal of Control*, Vol. 12, No. 3, 1970, pp. 497–510.