$$R=\left[\cos aI+(1-\cos a)E_{ii}+\sin a\sum_{\alpha}\sum_{\beta}\epsilon_{i\alpha\beta}E_{\alpha\beta}\right]$$
$$\left[\cos bI+(1-\cos b)E_{jj}+\sin b\sum_{\lambda}\sum_{\sigma}\epsilon_{j\lambda\sigma}E_{\lambda\sigma}\right]$$
$$\left[\cos cI+(1-\cos c)E_{kk}+\sin c\sum_{\mu}\sum_{\nu}\epsilon_{k\mu\nu}E_{\mu\nu}\right]$$

Performing the indicated matrix multiplications, one obtains an expression for R containing 27 terms. This expression simplifies somewhat when we disallow the possibilities $i=j$ and $j=k$, as previously discussed (so that $\delta_{ij}=0$, and $\delta_{jk}=0$), and use repeatedly the well-known identity[3]

$$\sum_{\mu}\epsilon_{mn\mu}\epsilon_{pq\mu}=\delta_{mp}\delta_{nq}-\delta_{mq}\delta_{np} \tag{12}$$

The resulting expression for R *is*

$$R=\cos a\cos b\cos cI+(1-\cos a)\cos b\cos cE_{ii}$$
$$+\cos a(1-\cos b)\cos cE_{jj}+\sin a\sin b\cos cE_{ji}$$
$$-\sin a(1-\cos b)\cos c\epsilon_{ijl}E_{lj}-(1-\cos a)\sin b\cos c\epsilon_{ijl}E_{il}$$
$$+\sin a\cos b\cos c\epsilon_{ijl}(E_{jl}-E_{lj})+\cos a\sin b\cos c\epsilon_{ijl}(E_{li}-E_{il})$$
$$+\cos a\cos b(1-\cos c)E_{kk}+(1-\cos a)\cos b(1-\cos c)\delta_{ik}E_{ik}$$
$$+\sin a\sin b(1-\cos c)\delta_{ik}E_{jk}$$
$$-(1-\cos a)\sin b(1-\cos c)\epsilon_{ijk}E_{ik}$$
$$+\sin a\cos b(1-\cos c)\epsilon_{ilk}E_{lk}+\cos a\sin b(1-\cos c)\epsilon_{jlk}E_{lk}$$
$$+\cos a\cos b\sin c\epsilon_{jkl}(E_{lj}-E_{jl})+(1-\cos a)\cos b\sin c\epsilon_{kil}E_{il}$$
$$-\cos a(1-\cos b)\sin c\epsilon_{jkl}E_{jl}+\sin a\sin b\sin c\epsilon_{kil}E_{jl}$$
$$+\sin a(1-\cos b)\sin c\epsilon_{ijl}\sum_{\mu}\sum_{\nu}\epsilon_{jk\nu}E_{\mu\nu}+\cos a\sin b\sin cE_{kj}$$
$$+(1-\cos a)\sin b\sin c\delta_{ik}E_{ij}+\sin a\cos b\sin c(E_{ki}-\delta_{ik}I) \tag{13}$$

The general expression Eq. (13), is now specialized by considering the two possibilities $i=k$, and $i\neq k$, as separate cases:

1) If $i\neq k$, then $\delta_{ik}=0$, and $l=k$ in permutation symbols of the form ϵ_{ijl}, $l=j$ in symbols of the form ϵ_{ilk}, and $l=i$ in symbols of the form ϵ_{jkl}. Since i, j, and k are all different, we may write the identity matrix as $I=E_{ii}+E_{jj}+E_{kk}$. The matrix R in Eq. (13) then simplifies, in this case, to:

$$R=\cos b\cos cE_{ii}+\epsilon_{ijk}\cos b\sin cE_{ij}-\epsilon_{ijk}\sin bE_{ik}$$
$$+(-\epsilon_{ijk}\cos a\sin c+\sin a\sin b\cos c)E_{ji}$$
$$+(\cos a\cos c+\epsilon_{ijk}\sin a\sin b\sin c)E_{jj}+\epsilon_{ijk}\sin a\cos bE_{jk}$$
$$+(\sin a\sin c+\epsilon_{ijk}\cos a\sin b\cos c)E_{ki}$$
$$+(-\epsilon_{ijk}\sin a\cos c+\cos a\sin b\sin c)E_{kj}+\cos a\cos bE_{kk} \tag{14}$$

2) If $i=k$, then $\epsilon_{ijk}=0$, and $l=6-(i+j)$ is the integer from the set $\{1,2,3\}$ different from i and j. In this case i, j, and l are all different, so that $I=E_{ii}+E_{jj}+E_{ll}$. The rotation matrix, Eq. (13), then simplifies to:

$$R=\cos bE_{ii}+\sin b\sin cE_{ij}-\epsilon_{ijl}\sin b\cos cE_{il}$$
$$+\sin a\sin bE_{ji}+(\cos a\cos c-\sin a\cos b\sin c)E_{jj}$$
$$+\epsilon_{ijl}(\cos a\sin c+\sin a\cos b\cos c)E_{jl}+\epsilon_{ijl}\cos a\sin b\,E_{li}$$
$$-\epsilon_{ijl}(\sin a\cos c+\cos a\cos b\sin c)E_{lj}$$
$$+(-\sin a\sin c+\cos a\cos b\cos c)E_{ll} \tag{15}$$

The elements of the two classes of three-rotation sequence matrices are simply the coefficients of the basis matrices in Eqs. (14) and (15). The expressions for these elements can be programmed as a simple FORTRAN subroutine, the inputs to which are the three rotation axes, specified by i, j, and k, and the three angles of rotation a, b, and c. The computation of the rotation matrix elements then requires only a single call to this subroutine.

These algorithms can also be used to calculate the elements of the degenerate matrices defined by Eqs. (2-4). We illustrate this procedure by considering a matrix of type (2). In this case, one simply replaces the angle c by $(b+c)$ and the angle b by 0, and assigns to the index j in the subroutine call the value $j=6-(i+k)$. For example, if $k=1$ and $i=3$, then $j=2$ (the value from the set $\{1,2,3\}$ different from i and k).

As a final remark, we wish to emphasize the usefulness of the analytical expression, Eq. (11), for computing the value of the permutation symbol as a function of its indices. Our version of a FORTRAN subroutine implementing the preceding algorithms is available upon request.

References

[1] Cupit, C.R., "Rotation Matrix Generation," *Simulation*, Vol. 15, Oct. 1970, pp. 145-147.

[2] Ohkami, Y., "Computer Algorithms for Computation of Kinematical Relations for Three Attitude Angle Systems," *AIAA Journal*, Vol. 14, Aug. 1976, pp. 1136-1137.

[3] Jeffreys, H. and Jeffreys, B.S., *Methods of Mathematical Physics*, 3rd ed., Cambridge, 1962, p. 73.

Numerical Derivatives for Parameter Optimization

David G. Hull*
University of Texas at Austin, Austin, Texas
and
Walton E. Williamson†
Sandia Laboratories, Albuquerque, N.M.

Introduction

THE accuracy with which a minimal point can be computed by a parameter optimization method using numerical derivatives depends on the accuracy with which the derivatives can be computed. Numerican derivatives are computed by differencing the performance index at a known point and a perturbed point. Hence, the problem is to predict the size of the perturbation so that the derivative has the most accuracy. If the perturbation is too large, the derivative is corrupted by truncation error, and if it is too small, by roundoff error. Hence, derivative accuracy is achieved by balancing the effects of these two error sources. In this paper a procedure for determining the perturbation is proposed for both first-order and second-order methods taking advantage

Received April 10, 1978; revision received Oct. 6, 1978. Copyright © American Institute of Aeronautics and Astronautics, Inc., 1978. All rights reserved.

Index category: Analytical and Numerical Methods.

*Professor, Dept. of Aerospace Engineering and Engineering Mechanics. Member AIAA

†Member of Technical Staff, Aerodynamics Dept. Member AIAA.

of the iterative nature of the parameter optimization problem. While the discussion is for a function of a single variable, it applies to a function of n variables because the derivatives are computed one variable at a time.

First-Order Method

The formula for the numerical derivative is obtained from the Taylor series expansion of the performance index $G(x)$ about a known point x. If only first-order terms are used, the derivative is approximated by

$$G_x(x) = [G(x.\delta x) - G(x)]/\delta x \qquad (1)$$

where δx is the perturbation. This formula also applies to a function of n variables. To compute the jth derivative, only the jth variable is perturbed.

A standard choice for the perturbation δx is

$$\delta x = \eta |x| \qquad |x| \geq 1$$

$$\delta x = \eta \qquad |x| < 1 \qquad (2)$$

where η is assumed to be $O(10^{-6})$. For a function of n variables, this procedure is applied to each variable, and the same value of η is used for each variable. Hence, for a surface with widely varying curvatures, each of the computed derivatives will have a different accuracy, and convergence will be limited by the worst one. The purpose of the method proposed below is to eliminate this problem.

From Eq. (1), it is apparent that the number of accurate significant figures in G_x depends on how much $G(x+\delta x)$ differs from $G(x)$. Hence, the best perturbation is the one that changes G as much as possible without truncation error being a factor. Assume for the moment that a value of $G_x(x)$ is known, so that to the first order

$$\delta G = G_x(x)\delta x \qquad (3)$$

To perturb G in the kth significant figure requires that $\delta G/G = 10^{-k} \equiv \epsilon$. Combining these two relations yields

$$\delta x = \epsilon G(x)/G_x(x) \qquad (4)$$

This equation gives, to first order, the perturbation that changes G in a particular significant figure.

To compute the derivative at a point, G_x would be estimated from Eqs. (1) and (2) and improved by using Eqs. (1) and (4). The value of G_x in (4) would be the first computed value. On the other hand, for iterative parameter optimization, the value of δx obtained from Eq. (4) would be used to compute G_x during the next iteration. However, this procedure cannot be continued to the minimum because G_x goes to zero and the predicted δx becomes infinite.

The value of ϵ is determined in the same way the step size is determined for a fixed-step numerical integration method. Here, the derivative is computed for a range of values of ϵ (say 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, ...), and succesive values of G_x (say 0.3482, 0.3378, 0.3374, 0.3371,...) are compared. The desired value of ϵ is the largest value for which two adjacent values of G_x agree to the largest number of significant figures. In the above example, the desired value of ϵ is $\epsilon = 10^{-2}$. The value of ϵ would be computed during the first iteration, and it could be checked every so many iterations thereafter. This is probably not necessary for the first-order method.

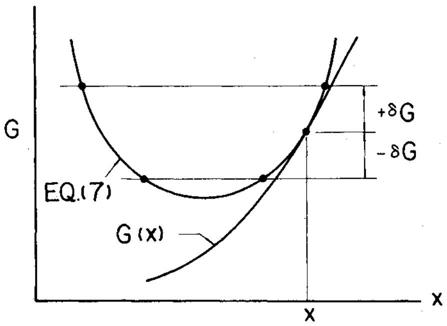

Fig. 1 Perturbation for a given δG.

The purpose of the proposed method is to provide uniform accuracy in the derivatives so that the greatest degree of convergence can be achieved. In a sense, this procedure could be referred to as local scaling. However, this nomenclature would be more obvious for second-order derivatives, where the curvature is used in the computation.

Second-Order Method

To second order in the perturbation, the expression for the derivative is given by

$$G_x = [G(x+\delta x) - G(x-\delta x)]/(2\delta x) \qquad (5)$$

A standard choice for the perturbation size is given by Eq. (2) where $\eta = O(10^{-4})$. Again, for a function of n variables, each of the derivatives will not have the same relative accuracy.

Another method for selecting δx has been proposed by Stewart.[1] It uses approximate second-derivative information to estimate the truncation error of the first-order derivative Eq. (1). Roundoff error is assumed to be due to storing G with a finite number of digits. Then, the perturbation is chosen to minimize the sum of truncation error and roundoff error. Finally, the derivative is computed using the second-order formula Eq. (5). There are two problems with this approach. First, the truncation and roundoff errors for the optimum perturbation are nearly equal for the method. This is not good because the roundoff error cannot be predicted accurately. Second, the best perturbation for a first-order method is smaller than that for a second-order method. (This is indicated by the choices $\eta = 10^{-6}$ and $\eta = 10^{-4}$ for the standard perturbations.) Hence, the predicted perturbation will create more round-off error when using the second-order formula.

A third approach for selecting the perturbation has been proposed by Curtis and Reid.[2] As in Ref. 1, the value of δx is found for the first-order method, and then the derivative is computed by the second-order formula. Specifically, the truncation error for the first-order derivative is obtained by differencing Eqs. (1) and (5). The roundoff error is assumed to be that due to storing x and G. Finally, δx is chosen such that the ratio of the truncation error and the roundoff error is a prescribed constant (for example 100). While this approach eliminates the first of the problems with Stewart's method, it does not eliminate the second.

The method proposed here eliminates both of these problems. It yields a perturbation for the second-order method, and the truncation error is made as large as possible. In general, the procedure uses first- and second-derivative information at one point to compute the perturbation for the next iteration. This is done by making the difference in Eq. (5) as large as possible.

In parameter optimization, three values of G are available: $G(x-\delta x)$, $G(x)$, and $G(x+\delta x)$. With these three function evaluations, the second derivative of G can be computed from

$$G_{xx} = [G(x+\delta x) - 2G(x) + G(x-\delta x)]/\delta x^2 \qquad (6)$$

to second order in the perturbation. Next assume that δx, G, G_x, and G_{xx} are available at a point.[‡] Then, the δx (say δx_p) which should have been used to perturb G in a particular

[‡] For the first iteration, Eq. (2) would be used for the δx needed to compute G_x and G_{xx}.

Table 1 Comparison of three perturbation schemes

Perturbation scheme	Convergence characteristics				
	Iterations	A	B	α	$\lvert G_x \rvert$
Eq. (2), $\eta = 10^{-4}$	31	$-2.2E-5$	$6.0E-5$	0.21	$2.0E+4$
Curtis-Reid	37	$2.6E-5$	$7.9E-6$	1.4	$4.1E+1$
Eq. (9), $\epsilon = 10^{-3}$	36	$-1.1E-7$	$1.6E-8$	0.88	$3.1E-1$

significant figure, that is, $\delta G = \epsilon G$, can be obtained from the Taylor series expansion

$$\delta G = G_x \delta x_p + G_{xx} \delta x_p^2 = \epsilon G \tag{7}$$

and is given by

$$\delta x_p = [-G_x \pm (G_x^2 + 2\epsilon G G_{xx})^{1/2}]/G_{xx} \tag{8}$$

As shown in Fig. 1, there exist four such values of δx_p: two solutions given by the ±, one solution for $\epsilon > 0$, and one solution for $\epsilon < 0$. If $G_{xx} > 0$, the desired δx_p is the smaller in magnitude of the two $\epsilon > 0$ solutions. Similarly, $\epsilon < 0$, if $G_{xx} < 0$. Since ϵ and G_{xx} are to have the same sign and they are multiplied together in Eq. (8), the term ϵG_{xx} is written as $\epsilon \lvert G_{xx} \rvert$, and ϵ is always taken as positive. Finally, the expression for δx_p can be rewritten as follows:

$$\delta x_p = \mathrm{Min} \lvert [-G_x \pm (G_x^2 + 2\epsilon G \lvert G_{xx} \rvert)^{1/2}]/G_{xx} \rvert \tag{9}$$

The perturbation given by Eq. (9) is used during the next iteration to compute G_x. Note that the quantity ϵ is computed at the starting point in the manner described for the first-order method, and it can be recomputed periodically during the iteration process if necessary.

It is observed that, as the minimum is approached, $G_x \to 0$ but G and G_{xx} approach constant values, meaning that δx approaches a constant value. Hence, while the proposed scheme may not yield accurate derivatives away from the minimum, the accuracy improves as the minimum is approached.

Finally, for a function of n variables, the above procedure is used to compute the derivative with respect to each variable. In a sense, the performance index is being scaled locally.

Example

The procedure presented here for computing numerical derivatives has been developed for use with the square-root variable-metric methods described in Refs. 2 and 3. The minimization algorithm with second-order numerical derivatives has been applied to a trajectory estimation problem of some substance, and as shown in Ref. 4, good results have been achieved.

The optimization problem solved in Ref. 4 consists of computing the initial conditions for a point mass set of differential equations along with the drag coefficient history which causes the solution to the state and observation equations to match radar data in a least-squares sense. The parameters associated with the drag coefficient consist of values of the drag coefficient at prescribed values of the time. An interpolation scheme is used to form the drag coefficient history. In all, the problem has 13 variables.

The initial attempt to solve the problem used Eq. (2) to determine the increments in the parameters to be used in computing the numerical partial derivatives. Convergence of the scheme was very slow, using this approach, and the optimization program would often prematurely indicate that a minimal solution had been found. Thus, after a few iterations the method would not be able to compute a decrease in the performance index, and the method would indicate that a minimal solution had been found. It was often obvious that these "solutions" were not minimal solutions but were the result of inaccurate derivatives that corrupted the one-dimensional search direction.

The sensitivity of the performance index to the various parameters in this problem varies greatly. This varying sensitivity does not allow accurate derivatives to be obtained by using the simple scheme described in Eq. (2). The scheme proposed in this paper, however, substantially improved the accuracy of the numerical partial derivatives and, hence, the convergence characteristics of the optimization method for this example problem.

To get some idea how the perturbation schemes compare, the trajectory problem of Ref. 4 has been solved using three schemes: Eq. (2) with $\eta = 10^{-4}$, Curtis-Reid (Ref. 2), and Eq. (9) with $\epsilon = 10^{-3}$. Convergence characteristics are presented in Table 1. The quantities A, B, and α are associated with the variable-metric optimization method. Both A and B should be small, $O(10^{-8})$, at convergence, and α should be near unity. The quantity $\lvert G_x \rvert$ is the square-root norm of the 13 derivatives at convergence. The perturbation obtained from the Curtis-Reid scheme has been limited to $\delta x > 10^{-7}$ because of roundoff-error considerations. The derivatives with respect to the drag coefficient parameters all used this value. Finally, the value $\epsilon = 10^{-3}$ for Eq. (9) has been computed in the first iteration and used for all iterations.

These results indicate that the perturbation scheme presented here allows the variable-metric optimization method to achieve a lower level of convergence than the other schemes. The better the convergence, the more confidence one has that the minimum has been achieved.

Acknowledgment

This research was supported in part by the Department of Energy.

References

[1] Stewart, G. W. III, "A Modification of Davidon's Method to Accept Difference Approximations for Derivatives," *Journal of the Association of Computing Machinery,* Vol. 14, No. 1, 1967, pp. 72-83.

[2] Curtis, A.R. and Reid, J.K., "The Choice of Step Lengths When Using Differences to Approximate Jacobian Matrices," *Journal of the Institute of Mathematics and its Applications,* Vol. 13, 1974, pp. 121-126.

[3] Williamson, W.E., "Square-Root Variable-Metric Method of Function Minimization," *AIAA Journal,* Vol. 13, Jan. 1975, pp. 107-109.

[4] Hull, D.G. and Tapley, B.D., "Square-Root Variable-Metric Methods for Minimization," *Journal of Optimization Theory and Applications,* Vol. 21, No. 3, 1977, pp. 251-259.

[5] Hull, D.G. and Williamson, W.E., "A Nonlinear Method for Parameter Identification Applied to a Trajectory Estimation Problem," *Journal of Guidance and Control,* Vol. 1, No. 4, 1978, pp. 286-288.