NASA Technical Memorandum 86382

Selecting Step Sizes in Sensitivity Analysis by Finite Differences

Jocelyn Iott, Raphael T. Haftka,
and Howard M. Adelman

AUGUST 1985

NASA Technical Memorandum 86382

Selecting Step Sizes in Sensitivity Analysis by Finite Differences

Jocelyn Iott and Raphael T. Haftka
Virginia Polytechnic Institute and State University
Blacksburg, Virginia

Howard M. Adelman
Langley Research Center
Hampton, Virginia

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1985

Summary

This paper deals with methods for obtaining near-optimum step sizes for finite difference approximations to first derivatives with particular application to sensitivity analysis. A technique denoted the finite difference (FD) algorithm, previously described in the literature, is reviewed and applied in a tutorial manner to the derivative of a sine function. The original FD algorithm is applicable to one derivative at a time. This paper describes an extension of the FD method to the calculation of several derivatives of matrix equations.

Both the original and extended FD algorithms were applied to sensitivity analysis for a data-fitting problem in which a polynomial is passed through data points, and the derivatives of the coefficients of the polynomial with respect to uncertainities in the data were calculated. The methods also were applied to sensitivity analysis of the structural response of a finite-element-modeled swept wing. In a previous paper, this sensitivity analysis of the swept wing required a time-consuming trial-and-error effort to obtain a suitable step size, but it proved to be a routine application for the FD algorithm herein.

Introduction

One of the key areas in the development of methods for optimization of engineering systems is sensitivity analysis—the calculation of derivatives of system response with respect to design parameters. In gradient-based optimization methods, derivatives are used to determine design changes to move toward an optimum design. Sensitivity derivatives also are used for rapid assessment of the effect of a change in a design parameter on the response of an engineering system. In these typical applications, especially when a large number of repetitive calculations are required, the finite difference method is often a useful technique for computing the derivatives.

One problem which arises in connection with the finite difference method is the choice of step size. The step size can contribute to two types of errors in the finite difference approximation—truncation error and condition error. Truncation error is the difference between the exact value and the computed value of a perturbed function if it is assumed that there is no loss of numerical precision in the calculation. Truncation error is generally represented by neglected terms in the Taylor series representation of a perturbed function. Truncation errors are increased by overly large step sizes. Condition errors are associated with numerical noise and are caused by loss of numerical precision. These errors may result from computer round-off error or the operation of subtracting large numbers which are very nearly equal. Condition errors in finite difference derivatives generally increase with decreasing step sizes.

In the absence of methods for selecting a suitable step size, a trial-and-error approach, which requires many function evaluations for each derivative, must be used (ref. 1). Automatically calculating an optimum step size can decrease the time and cost of the calculations. One such method is described in reference 1. This method, referred to as the finite difference (FD) algorithm, is formulated on the basis that the finite difference error is minimized when the truncation error is equal to the condition error. The FD algorithm has previously been demonstrated for single derivatives and proved to be effective in producing step sizes that gave good approximations to the exact derivatives (refs. 1 and 2). The need to apply the method to functions governed by matrix equations has led to an extension of the FD algorithm and is the main subject of this paper.

In this paper, the FD algorithm is applied to three examples: (1) the sine function, (2) a matrix equation (ref. 3, pp. 12–13) involving data fitting by a polynomial in which derivatives of coefficients with respect to uncertainties in the data are calculated, and (3) a finite element structural analysis of a swept wing. The polynomial and wing problems are used to test both the original and the extended versions of the FD algorithm. The finite element model of the swept wing is included in this study because in a previous sensitivity study, described in reference 4, the uncertainty of step size became a major concern, and a succession of trials was required to determine an acceptable step size. This paper contains a summary of the original FD algorithm, the development of the extended FD algorithm, and the results of applying the algorithms to the above problems.

Symbols

a_i	coefficient of ith term of Nth-order polynomial
$C(h)$	condition error in approximating first derivative by finite difference method
$\hat{c}(\Phi)$	condition error in approximating second derivative by finite difference method
e	bound on total error in approximating derivatives by finite difference method
$\mathbf{F}, \bar{\mathbf{F}}$	force vectors
f	exact value of a function

f'	exact first derivative of a function
f''	exact second derivative of a function
f''_b	bound on second derivative
h	step size
h_{opt}	optimum step size
h_s	step size used for estimating second derivative
$\mathbf{K}$	stiffness matrix
N	degree of polynomial
$T(h)$	truncation error in approximating first derivative by finite difference method
$\mathbf{U}, \bar{\mathbf{U}}$	exact and computed displacement vectors, respectively
x	independent variable
x_i, y_i	ith data point in Vandermonde problem
ϵ_A	bound on absolute error in computed function values
Φ	finite difference approximation to second derivative

Subscripts:

max	maximum
min	minimum

Summary of FD Algorithm

Truncation and Condition Errors

The FD algorithm (ref. 2) is used to calculate an optimum step size for a first derivative approximated by forward differences

$$f'(x) = \frac{f(x+h) - f(x)}{h} \tag{1}$$

Whenever finite difference formulas are used to approximate derivatives, there are two sources of error: truncation error and condition error. The truncation error $T(h)$ is a result of the neglected terms in the Taylor series expansion of the perturbed function and is represented in references 1 and 2 as being proportional to the step size as follows:

$$T(h) = \frac{h}{2} f''(x + \varsigma h) \qquad 0 \leq \varsigma \leq 1 \tag{2}$$

The condition error is the difference between the numerical evaluation of a function and the exact value. One contribution to the condition error is round-off error in evaluating equation (1), which is comparatively small for most mainframe computers and is, therefore, considered negligible. However, if $f(x)$ is computed by a lengthy or ill-conditioned numerical process, the round-off contribution to the condition error can be substantial. Additionally, condition errors may result if $f(x)$ is calculated by an iterative process which is terminated early. The condition error $C(h)$ can be conservatively estimated as being inversely proportional to the step size. The following expression, used herein, is given in reference 2:

$$C(h) = \frac{2}{h}\epsilon_A \tag{3}$$

Determination of ϵ_A

In equation (3), ϵ_A is a bound on the absolute error in the computed function f. Obtaining an appropriate value of ϵ_A for use in equation (3) generally requires some knowledge of the problem being solved. A straightforward (but not always convenient) method of obtaining ϵ_A is to perform the calculation of f in single and double precision and subtract the results. (See ref. 2 for examples of this technique.) An alternative method for matrix problems is described in appendix A of this paper. A method for estimating ϵ_A also is given in the appendix of reference 5.

Development of Optimum Step Size Formula

A bound e on the total error is the sum of the truncation and condition errors

$$e = \frac{h}{2}\left|f''_b\right| + \frac{2}{h}\epsilon_A \tag{4}$$

where $|f''_b|$ is a bound on the second derivative in the interval $[x, x+h]$. In many cases, such a bound on the second derivative is not available, and a central finite difference approximation

$$f''_b \approx \Phi = \frac{f(x+h_s) - 2\,f(x) + f(x-h_s)}{h_s^2} \tag{5a}$$

is used, so that

$$e = \frac{h}{2}\left|\Phi\right| + \frac{2}{h}\epsilon_A \tag{5b}$$

It should be noted that equation (5a) is an approximation to $f''(x)$ and is a reasonable bound on f'' in the interval $[x, x+h]$ only if f'' varies slowly near x. The step size h_s used for estimating the second derivative is not necessarily the same as that used for estimating the first derivative. The optimum step size is found by minimizing the error defined in equation (4). Taking the derivative of e with respect to h, substituting Φ for f_b'', and setting the resulting expression equal to zero gives

$$\frac{\partial e}{\partial h} = \frac{|\Phi|}{2} - \frac{2}{h^2}\epsilon_A = 0 \tag{6}$$

Solving for h yields

$$h_{\text{opt}} = 2\sqrt{\frac{\epsilon_A}{|\Phi|}} \tag{7}$$

Clearly equation (7) is not applicable to derivatives of functions at points where the second derivative is zero or close to zero—for example, when the function is nearly constant, linear, or an odd function. For these special cases, the FD algorithm fails to provide a satisfactory finite difference step size. As discussed in reference 2, a computerized implementation of the algorithm terminates with an appropriate error message when $|\Phi|$ is less than a certain threshold value. Equation (6) leads to the observation that

$$\frac{h_{\text{opt}}}{2}\,|\Phi| = \frac{2}{h_{\text{opt}}}\epsilon_A \tag{8}$$

Therefore, the optimum step size is the one which balances the truncation and condition errors.

Estimation of Second Derivative

Before evaluating equation (7) for h_{opt}, the second derivative is evaluated from equation (5a). This requires a value for h_s, which is obtained by the following trial-and-error procedure. The relative condition error in f'' resulting from use of equation (5a) is shown in references 1 and 2 to be bounded by

$$\hat{c}(\Phi) = \frac{4\epsilon_A}{h_s^2}\,|\Phi| \tag{9}$$

The choice of an initial guess for h_s is based on the conditions that the function is well scaled and that the function and its second derivative are comparable in magnitude, such that

$$f'' = \frac{1+|f|}{(1+|x|)^2} \tag{10}$$

where $|x|$ is of the order unity. The trial value of h_s to be used in equation (5a) is calculated by substituting equation (10) for Φ in equation (7). This results in

$$h_s = 2(1+|x|)\sqrt{\frac{\epsilon_A}{1+|f|}} \tag{11}$$

The interval h_s is used to compute Φ from equation (5a); Φ, in turn, is used to calculate the error $\hat{c}(\Phi)$ from equation (9). If the error is too large (greater than 0.1), h_s is increased. If it is too small (less than 0.001), h_s is decreased. Once a value of h_s is found which produces an acceptable value for Φ, the value of Φ is used in equation (7) to produce an optimum step size, h_{opt}. The limit of 0.1 is chosen to prevent the condition error from being excessive. The limit of 0.001 is chosen because when the condition error is very small, there is a risk of high truncation errors.

Applications of FD Algorithm

Sine Function

A simple example to demonstrate the FD algorithm is sin x. This application is described in a tutorial fashion in appendix B to illustrate the steps in the procedure.

Polynomial-Fit (Vandermonde) Problem

The polynomial-fit example is based on the problem of passing an Nth-order polynomial through $N+1$ data points, $x_i = i$, $y_i = f(x_i)$, $i = 0, 1, 2, \ldots, N$. The form of the polynomial is

$$f(x) = \sum_{i=0}^{N} a_i x^i$$

To solve for the coefficients a_i, a system of $N+1$ equations in $N+1$ unknowns is generated. It is required to calculate the derivatives of the coefficients a_i with respect to the data point locations x_i. The equation for computing the coefficients is

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^N \\ 1 & x_2 & x_2^2 & x_2^3 & \cdots & x_2^N \\ 1 & x_3 & x_3^2 & x_3^3 & \cdots & x_3^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & x_N^3 & \cdots & x_N^N \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_N \end{Bmatrix} = \begin{Bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{Bmatrix} \tag{12}$$

The matrix in equation (12) is referred to as the Vandermonde matrix and is useful in the present work because the range of values (from unity to N^N)

causes it to become ill-conditioned even for relatively small values of N. For example, $N = 10$ generates values from 1 to 10^{10}.

In this example, the value for N was selected to be 12 because it produced condition errors which were significant but not catastrophic. The values of x_0 and y_0 were set to zero, so that $a_0 = 0$. The FD algorithm was used to find an optimum step size to use in approximating the derivatives of each coefficient of the polynomial with respect to the location of the third data point, i.e., $\partial a_i/\partial x_3$, $i = 1, 2, ..., 12$. The data points y_i were generated from the equation $y_i = i + i^5 + i^9$ at values of i ranging from 0 through 12. This method of generating data gives a means for estimating ϵ_A because the polynomial which exactly fits these data has coefficients $a_1 = a_5 = a_9 = 1.0$, and the remaining coefficients should be 0.0. Any deviation in the values of the coefficients is the result of numerical imprecision. A standard matrix subroutine denoted MATOPS was used to solve the system of equations. MATOPS offers three options (full pivoting, partial pivoting, and no pivoting) corresponding to three levels of accuracy. The least accurate option was used for evaluating derivatives, and a first estimate for ϵ_A was obtained by subtracting the least accurate solutions from the most accurate. The second method used for estimating ϵ_A was the iterative improvement technique described in appendix A.

The two estimates of ϵ_A were very close, but both were sensitive to the values of x_i. For the nominal point where $x_i = i$, ϵ_A was evaluated as 8.3×10^{-4}. However, at the nearby point of $x_i = 0.01 + i$, ϵ_A was estimated as 0.153. The low value of the error for $x_i = i$ may be explained by a reduction in round-off errors when all the terms in the Vandermonde matrix are exact integers.

Based on an ϵ_A value of 0.153, equation (11) gave $h_s = 2.2$, and $\hat{c}(\Phi) = 6.8 \times 10^{11}$ from equation (9). Because $\hat{c}(\Phi)$ was outside the interval [0.001,0.1], h_s was reduced, successively, by factors of 10 till a value of 2.2×10^{-3} resulted in $\hat{c}(\Phi) = 0.06$. The optimum step size, h_{opt}, was found from equation (7) to be 1.71×10^{-4}. The derivative $\partial a_1/\partial x_3$ was computed by repeating the solution of equation (12) with x_3 replaced by $x_3 + h_{\text{opt}}$, while the remaining x_i values as well as the y_i values were left unchanged from their nominal values. The value of the derivative was obtained as -4.36028×10^6. The exact value of the derivative was calculated analytically by Gaylen A. Thurston of NASA Langley Research Center to be -4.36003×10^6, so that the error in the derivative at h_{opt} is 250, or 0.0057 percent. The derivatives as a function of the step size are listed in table I for a wide range of h values bounding h_{opt}. It is evident that the choice of h_{opt} is indeed a good one.

TABLE I. FINITE DIFFERENCE EVALUATION OF $\partial a_1/\partial x_3$ FOR POLYNOMIAL-FIT PROBLEM

Step size, h	Finite difference derivative	Derivative error[a]	Derivative error, percent
10^{-1}	-5.59003×10^6	1.23×10^6	28.2108
10^{-2}	-4.46303	1.03×10^5	2.3624
10^{-3}	-4.37003	1.00×10^4	0.2294
1.71×10^{-4} (h_{opt})	-4.36028	0.25×10^3	0.0057
10^{-4}	-4.35313	0.69×10^4	0.1583
10^{-5}	-4.36095	0.92×10^3	0.0211
10^{-6}	-4.34963	1.04×10^4	0.2385
10^{-7}	-4.32533	3.47×10^4	0.7959
10^{-8}	-4.05803	3.02×10^5	6.9266
10^{-9}	-5.31903	9.59×10^5	21.9953

[a]Relative to exact value (-4.36003×10^6).

It was of interest to evaluate the error estimate of equation (5b) for this problem because the exact error was available. Using equations (5b) and (8) gives the bound for the minimum error $e_{\min}$ as

$$e_{\min} = \frac{4\epsilon_A}{h_{\text{opt}}} \tag{13}$$

So for $\epsilon_A = 0.153$ and $h_{\text{opt}} = 1.71 \times 10^{-4}, e_{\min} = 3580$. This value is fourteen times larger than the actual error. To resolve this large difference between the error predicted by the FD algorithm and the actual error, the derivative was calculated in a narrow range of step sizes in the neighborhood of h_{opt}. A sample of the results is given in table II. It was clear that near h_{opt}, the value of the finite difference derivative exhibits a random oscillation, and the small error obtained at h_{opt} is somewhat fortuitous.

The FD algorithm also was used for derivatives of the remaining polynomial coefficients with respect to x_3. Table III lists the second derivative step size h_s, the optimum step size h_{opt}, and the value of ϵ_A for all 12 derivatives. It is seen from table III that h_{opt} varies from 1.642×10^{-4} to 3.054×10^{-4}.

Swept-Wing Problem

A finite element structural model of a swept wing is shown in figure 1. The Engineering Analysis Language (EAL) finite element program (ref. 6) was used to model the wing. The model contains 194 elements consisting of rods, triangular membranes, and shear panels connected at 88 nodes and constrained at the root. Additional details of the wing model are available in references 4 and 7.

TABLE II. FINITE DIFFERENCE EVALUATION OF $\partial a_1/\partial x_3$ IN THE NEIGHBORHOOD OF $h_{\text{opt}} = 1.71 \times 10^{-4}$ FOR POLYNOMIAL-FIT PROBLEM

Step size, h	Finite difference derivative	Derivative error[a]	Derivative error, percent
0.4356×10^{-4}	-4.35986×10^{6}	170	0.0039
0.5445	−4.35980	230	0.0053
0.6806	−4.35911	920	0.0211
0.8507	−4.36278	2750	0.0631
1.063	−4.35696	3070	0.0704
1.329	−4.35715	2880	0.0661
1.662	−4.36103	1000	0.0229
1.714 (h_{opt})	−4.36028	250	0.0057
2.077	−4.36072	690	0.0158
2.596	−4.36110	1070	0.0245
3.245	−4.36372	3690	0.0846
4.056	−4.36583	5800	0.1330
5.071	−4.36493	4900	0.1124
6.338	−4.36316	3130	0.0718

[a]Relative to exact value (-4.36003×10^{6}).

TABLE III. VALUES OF ϵ_A, h_s, AND h_{opt} FOR DERIVATIVES $\partial a_i/\partial x_3$ FOR POLYNOMIAL-FIT PROBLEM

Coefficient	ϵ_A	h_s	h_{opt}
a_1	0.1526	2.209×10^{-3}	1.714×10^{-4}
a_2	0.4560	5.395	1.794
a_3	0.5643	0.600	1.642
a_4	0.3886	4.982	2.004
a_5	0.1674	2.314	2.093
a_6	4.777×10^{-2}	1.748	2.449
a_7	9.273×10^{-2}	0.770	2.112
a_8	1.231×10^{-2}	2.807	2.478
a_9	1.099×10^{-3}	5.931	2.884
a_{10}	6.311×10^{-5}	2.010	2.633
a_{11}	2.102×10^{-6}	3.668	2.913
a_{12}	3.086×10^{-8}	4.444	3.054

The problem, as described in reference 4, concerns derivatives of the wing displacements with respect to element cross-sectional dimensions. In order to achieve accurate results, in reference 4 a time-consuming trial-and-error process was used to determine an appropriate step size. The derivative of the normal displacement at node 44 with respect to one of the design variables is chosen to investigate the application of the FD algorithm. The design variable selected controls the thickness of the triangular membrane elements 1 through 6 located at the root of the wing model as shown in figure 1.

The first step is to calculate an initial value for h_s from equation (11). This calculation requires values of the function f and the independent variable x and an estimate of the error bound ϵ_A for this problem. In this example, x is the nominal value of the design variable (0.2 in.), f is the normal displacement of node 44 (40.8 in.), and ϵ_A (obtained by the method described in appendix A) is 1.0×10^{-8} in. These values result in an initial h_s value of 3.775×10^{-5} in. used in equation (5a) to approximate Φ, which was then used to calculate $\hat{c}(\Phi)$ from equation (9). The value for $\hat{c}(\Phi)$ was 0.06 and fell within the acceptable range [0.001,0.1]. The value of the optimum step size h_{opt}, 9.4×10^{-6} in., produced a 0.0059 percent error in the finite difference derivative relative to the exact value obtained in reference 4. This step is better than the one (2×10^{-5} in.) obtained by trial and error in reference 4 and was achieved with far fewer function evaluations. The variation of the error with step size is summarized in table IV and indicates that indeed the value of 9.4×10^{-6} in. is close to optimal. Unlike the case of the polynomial-fit problem, for this problem equation (13) predicts a minimum error which is very close to the actual error (0.004 in. compared with 0.003 in.).

TABLE IV. VARIATION OF DERIVATIVE OF TIP DISPLACEMENT WITH STEP SIZE FOR SWEPT-WING PROBLEM

Step size,[a] h, in.	Error[b] in finite difference derivative, percent
2×10^{-3}	0.9400
2×10^{-4}	0.0960
[c]2×10^{-5}	0.0098
[d]9.4×10^{-6}	0.0059
2×10^{-6}	0.0200

[a]Nominal thickness = 0.2 in.
[b]Relative to analytically calculated value of −50.907 in. (ref. 4).
[c]Best value obtained by trial and error in reference 4.
[d]Step size calculated by FD algorithm.

Extension of FD Algorithm to Vector Derivatives

When the derivative of a vector with respect to a design parameter is required, the FD algorithm is not always useful in its original form. As seen in the polynomial-fit example (table III), the FD algorithm gives a different optimum step size for the

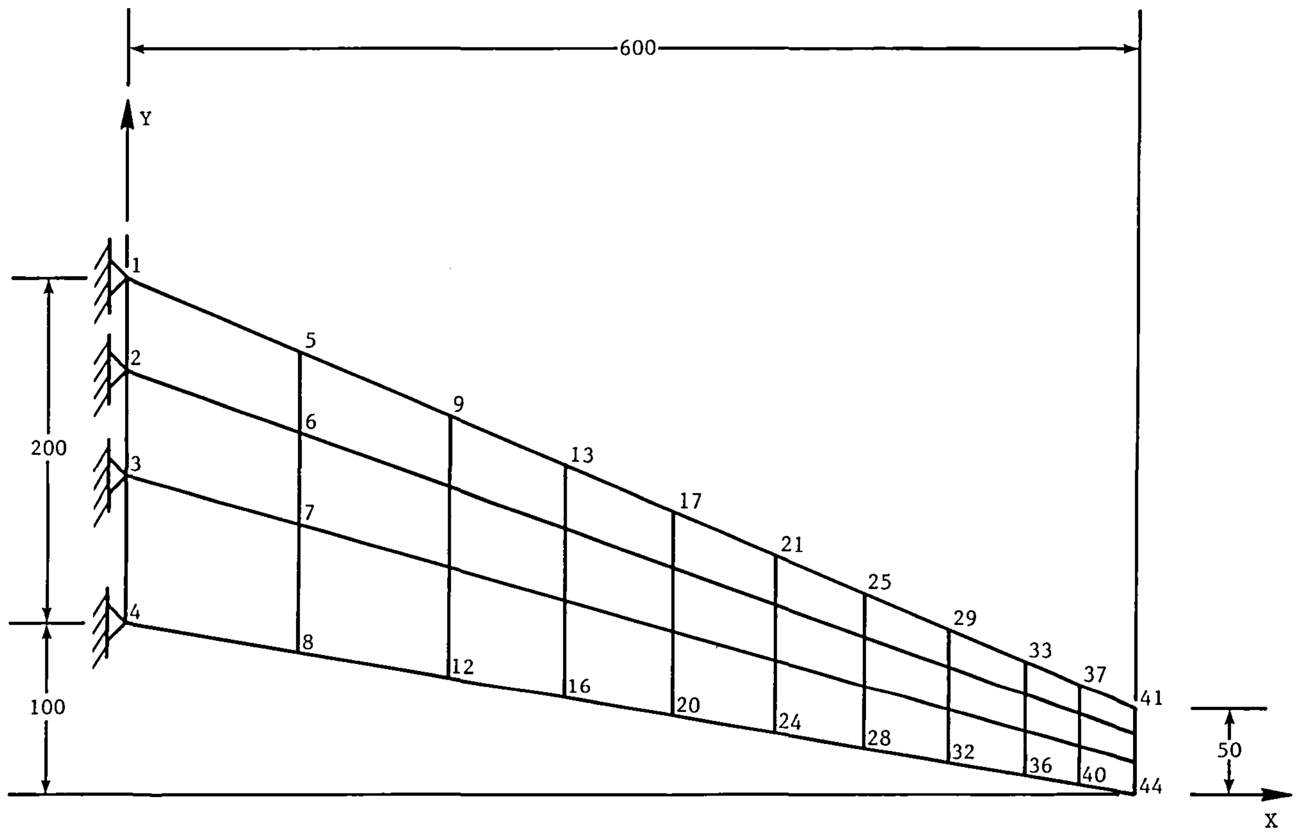

Figure 1. Geometry, element numbers, and node numbers for swept wing. Dimensions in inches.

derivative of each component of the vector. From the standpoint of efficiency, it is desirable in the case of a derivative of a vector to find a single step size for all components which is the optimum compromise among them. It is possible, of course, that no such compromise exists if the components have widely different condition and truncation errors. However, in most practical cases it is expected that such a compromise will be possible, and an extension of the FD algorithm described below is appropriate.

In seeking a compromise among the step size requirements of the different components, it is natural to seek to minimize some norm of the error vector. Given an error vector $\mathbf{E}$ with components e_i, $i = 1, 2, \ldots, N$, there are three norms which may be used:

$$\|\mathbf{E}\|_1 = \sum_{i=1}^{N} |e_i| \tag{14}$$

$$\|\mathbf{E}\|_2 = \sqrt{N} e_{\text{rms}} = \sqrt{\sum_{i=1}^{N} e_i^2} \tag{15}$$

$$\|\mathbf{E}\|_\infty = \max_i |e_i| \tag{16}$$

where e_{rms} denotes the root-mean-square of the error.

The extended FD algorithm proceeds through the following steps:

1. Select an initial h_s (e.g., by using eq. (11)), and calculate Φ from equation (5a) for each component of the vector.

2. Estimate the condition error in Φ ($\hat{c}(\Phi)$ from eq. (9)) for each component, and vary h_s to bring that error into the required interval of [0.001,0.1]. Failure to find a common h_s is an indication that it is futile to use a single step size for all components. In that case, the components may be divided into two or more groups and subsequent steps repeated for each group separately.

3. Estimate the optimum step size h_{opt} for each component from equation (7). The largest and smallest values of h_{opt} are denoted $h_{\max}$ and $h_{\min}$, respectively.

4. Divide the interval $[h_{\min}, h_{\max}]$ into m equal subintervals resulting in $m + 1$ step size values $h_1 = h_{\min}, h_2, h_3, \ldots, h_{m+1} = h_{\max}$. Equation (4) is used to estimate the error in each component for each h_i value and the associated values of Φ and ϵ_A.

5. Select as h_{opt} that value of h_i which minimizes the desired error norm.

Applications of Extended FD Algorithm

Polynomial-Fit Problem

The application of the standard FD algorithm to the polynomial-fit problem discussed earlier produced 12 optimum step sizes, one for each coefficient. (See table III.) To calculate all 12 finite difference sensitivity derivatives with the same step size, the extended FD algorithm was used. To apply the extended FD algorithm, the second derivative step size h_s obtained for a_1 (2.209×10^{-3}, see table III) was used for all 12 coefficients. All 12 $\hat{c}(\Phi)$ values were in the acceptable range of [0.001,0.1]. The values for h_{opt} changed slightly from the values given in table III and ranged from 1.714×10^{-4} to 3.364×10^{-4}. Subdividing this range into 20 intervals resulted in 21 vectors, each containing 12 error components. The $\|\mathbf{E}\|_1$ and $\|\mathbf{E}\|_2$ norms (eqs. (14) and (15)) produced the same step size of 1.879×10^{-4}, whereas the $\|\mathbf{E}\|_\infty$ produced a step size of 1.797×10^{-4}. A check of the actual errors based on these step sizes showed that all 12 errors are acceptably small.

Swept-Wing Problem

To demonstrate the application of the extended FD algorithm to the swept wing, a set of 12 displacement derivatives were chosen. These are the derivatives of the normal displacements at nodes 5 through 10 and 39 through 44. (See fig. 1.) The FD algorithm was first used to find 12 optimum step sizes. For this purpose, $h_s = 3.775 \times 10^{-5}$ in. was used for all 12 components, and one step size was extracted for calculating all 12 derivatives. Unlike the previous case, the three norm criteria produced different step sizes: 9.05×10^{-6} in. from $\|\mathbf{E}\|_1$, 9.44×10^{-6} in. from $\|\mathbf{E}\|_2$, and 9.6×10^{-6} in. from $\|\mathbf{E}\|_\infty$. Derivatives using these step sizes were compared with analytical results from reference 4. All three gave sufficiently accurate results. For example, table V shows that a step size of 9.44×10^{-6} in. produced excellent results compared with the analytical solution.

Effect of ϵ_A

Because the error bound ϵ_A may be uncertain in some problems, the effect of ϵ_A on the step size calculations and derivative results is of interest. To test this effect, the value of the nominal ϵ_A was varied by a factor of 10 for each of the single-derivative problems. The results are shown in tables VI, VII, and VIII for the sine function, the polynomial-fit problem, and the swept-wing problem, respectively. In each case, the modified values of ϵ_A had a noticeable but small effect on the results, and the errors were still acceptable.

Concluding Remarks

This paper deals with methods for obtaining near-optimum step sizes for finite difference approxima-

TABLE V. FINITE DIFFERENCE DERIVATIVES OF SWEPT-WING TIP DEFLECTION FOR $h_{opt} = 9.44 \times 10^{-6}$ in.

Node number	Derivative: Finite difference method	Derivative: Analytical method of reference 4	Error, percent
5	−2.9372	−2.9374	0.00681
6	−3.7578	−3.7580	0.00532
7	−4.5608	−4.561	0.00438
8	−5.5679	−5.5682	0.00538
9	−10.645	−10.646	0.00939
10	−11.597	−11.598	0.00862
39	−47.737	−47.739	0.00419
40	−48.219	−48.222	0.00622
41	−49.773	−49.776	0.00603
42	−50.112	−50.115	0.00599
43	−50.496	−50.499	0.00594
44	−50.904	−50.907	0.00589

TABLE VI. EFFECT OF ϵ_A ON DERIVATIVE OF sin x

ϵ_A	h_{opt}	Finite difference derivative	Derivative error,[a] percent
Nominal (2.188×10^{-7})	1.108×10^{-3}	0.706679	0.06
0.1 × Nominal (2.188×10^{-8})	1.20×10^{-4}	0.708333	0.17
10 × Nominal (2.188×10^{-6})	3.464×10^{-3}	0.705831	0.18

[a]Relative to exact value (0.707107).

TABLE VII. EFFECT OF ϵ_A ON DERIVATIVE $\partial a_1/\partial x_3$ IN POLYNOMIAL-FIT PROBLEM

ϵ_A	h_{opt}	Finite difference derivative	Derivative error,[a] percent
Nominal (0.153)	1.714×10^{-4}	-4.3603×10^{6}	0.006
0.1 × Nominal (0.0153)	2.656×10^{-6}	-4.3596×10^{6}	0.009
10 × Nominal (1.53)	4.885×10^{-5}	4.3636×10^{6}	0.08

[a]Relative to exact value (-4.3600×10^{6}).

TABLE VIII. EFFECT OF ϵ_A ON DERIVATIVE IN SWEPT-WING PROBLEM

ϵ_A, in.	h_{opt}, in.	Finite difference derivative	Derivative error,[a] percent
Nominal (0.10348×10^{-7})	9.4×10^{-6}	−50.904	0.0059
0.1 × Nominal (0.10348×10^{-8})	2.525×10^{-6}	−50.914	0.0138
10 × Nominal (0.10348×10^{-6})	2.929×10^{-5}	−50.900	0.0138

[a]Relative to exact value (−50.907).

tions to first derivatives with particular application to sensitivity analysis. A technique denoted the finite difference (FD) algorithm, previously described in the literature, is reviewed and applied in a tutorial manner to the derivative of a sine function. The original FD algorithm is applicable to one derivative at a time. This paper describes an extension of the FD algorithm to the calculation of several derivatives of matrix equations.

The FD algorithm requires an estimate of the error in the computed value of the functions being differentiated. This paper describes a simple method for calculating the error estimate for matrix problems. Denoted the iterative improvement technique, the method involves a second solution of the matrix equation. Since it uses the same decomposed matrix that was used in the original solution, it imposes a minimal computational burden on the analysis.

Both the original and extended FD algorithms were applied to sensitivity analysis for a data-fitting problem in which a polynomial is passed through data points, and the derivatives of the coefficients of the polynomial with respect to uncertainties in the data were calculated. The algorithms also were applied to sensitivity analysis of the structural response of a finite-element-modeled swept wing. In a previous paper, this sensitivity analysis of the swept wing required a time-consuming trial-and-error effort to obtain a suitable step size, but it proved to be a routine application for the FD algorithm herein.

One limitation of the FD algorithm is that it requires a finite lower bound on the second derivative of the function being differentiated. This bound is generally estimated by a central difference approximation. In certain special cases, such as when the function is nearly constant, linear, or an odd function, the second derivative is zero or close to zero, and in such a case, the FD algorithm is not a reliable provider of an optimum step size for the first derivative. In the absence of this special condition, however, the FD algorithm has been found to be useful, accurate, and reliable.

Appendix A

Iterative Improvement Technique for Estimating ϵ_A

A method for estimating ϵ_A is based on the so-called "iterative improvement" of a solution to a system of linear equations. This process is explained for the system

$$\mathbf{KU} = \mathbf{F} \tag{A1}$$

where $\mathbf{K}$ is an $n \times n$ matrix, and $\mathbf{F}$ and $\mathbf{U}$ are $n \times 1$ vectors. In finite element structural analysis, $\mathbf{K}$ is the stiffness matrix, $\mathbf{U}$ is a displacement vector, and $\mathbf{F}$ is a force vector. If $\bar{\mathbf{U}}$ is the solution obtained numerically, $\bar{\mathbf{F}}$ can be defined as

$$\bar{\mathbf{F}} = \mathbf{K}\bar{\mathbf{U}} \tag{A2}$$

Subtracting equation (A2) from equation (A1) gives

$$\mathbf{K}(\mathbf{U} - \bar{\mathbf{U}}) = \mathbf{F} - \bar{\mathbf{F}} \tag{A3}$$

so that an estimate of $\mathbf{U} - \bar{\mathbf{U}}$ may be obtained by solving equation (A3). The solution gives an order of magnitude estimate for the error ϵ_A, and since it uses the same decomposed matrix that was used in the original solution (eq. (A1)), it imposes a minimal computational burden on the analysis.

Appendix B

Application of FD Algorithm to Sine Function

A simple example to demonstrate the FD algorithm is $\sin x$. The derivative of this function was evaluated with a hand calculator which carries up to 10 significant digits. The function $\sin x$ is convenient because the exact derivatives are known and can be compared with the approximations. The bound on the absolute error in the computed function values ϵ_A is controlled by the number of digits used on the calculator. In this example, the number of digits used was six. The function $\sin x$ and its first derivative were evaluated at $x = 0.785398$ rad, or 45°. Therefore, $f(x) = \sin x = 0.7071067812$ to 10 significant figures, and $\sin x = 0.707107$ to 6 significant figures. The difference between these two numbers is $\epsilon_A = 2.188 \times 10^{-7}$.

The first step is the calculation of the initial step size h_s to be used in estimating the second derivative. From equation (11)

$$h_s = 2(1+|x|)\sqrt{\frac{\epsilon_A}{1+|f|}}$$

$$= 2(1+0.785398)\sqrt{\frac{2.188 \times 10^{-7}}{1+0.707107}}$$

$$= 0.001278$$

Using h_s in equation (5a) gives

$$\Phi = \frac{f(x+h_s) - 2\,f(x) + f(x-h_s)}{h_s^2}$$

$$= \frac{0.708010 - 2(0.707107) + 0.706202}{0.001278^2}$$

$$= -1.225$$

The relative condition error is calculated from equation (9)

$$\hat{c}(\Phi) = \frac{4\epsilon_A}{h_s^2\,|\Phi|}$$

$$= \frac{4(2.188 \times 10^{-7})}{(0.001278)^2(1.225)}$$

$$= 0.4374$$

Since $\hat{c}(\Phi)$ is outside the acceptable interval [0.001,0.1], h_s must be increased, and new values for Φ and $\hat{c}(\Phi)$ must be calculated. For convenience, h_s was increased by a factor of 10. The new value for $|\Phi|$ is 0.711656 for $\hat{c}(\Phi) = 0.008621$. Since this value falls within the acceptable interval, Φ may be used to calculate h_{opt} from equation (7).

$$h_{\mathrm{opt}} = 2\sqrt{\frac{\epsilon_A}{|\Phi|}}$$

$$= 2\sqrt{\frac{2.188 \times 10^{-7}}{0.711656}}$$

$$= 1.108 \times 10^{-3}$$

Substituting h_{opt} into the forward difference formula (eq. (1)) gives an approximation to the first derivative

$$f'(x) \approx \frac{f(x+h_{\mathrm{opt}}) - f(x)}{h_{\mathrm{opt}}}$$

$$= \frac{0.707890 - 0.707107}{1.108 \times 10^{-3}}$$

$$= 0.706675$$

Comparing this result with the exact derivative indicates that the use of h_{opt} gives a relative error of 0.06 percent. Table BI shows derivatives calculated with step sizes larger and smaller than the optimum along with the error relative to the exact derivative.

TABLE BI. FINITE DIFFERENCE DERIVATIVES OF $\sin x$ AT 45° FOR VARIOUS STEP SIZES

Step size, h	Finite difference derivative	Derivative error,[a] percent
1.108×10^{-1}	0.666525	5.74
1.108×10^{-2}	0.703159	0.56
1.108×10^{-3} (h_{opt})	0.706675	0.06
1.108×10^{-4}	0.703971	0.40
1.108×10^{-5}	0.722022	2.11

[a]Relative to exact value (0.707107).

References

1. Gill, Philip E.; Murray, Walter; Saunders, Michael A.; and Wright, Margaret H.: *Computing the Finite-Difference Approximations to Derivatives for Numerical Optimization.* SOL 80-6 (Contract DAAG29-79-C-0110), Dep. Operations Research—SOL, Stanford Univ., May 1980.
2. Gill, Philip E.; Murray, Walter; Saunders, Michael A.; and Wright, Margaret H.: Computing Forward-Difference Intervals for Numerical Optimization. *SIAM J. Sci. & Stat. Comput.*, vol. 4, no. 2, June 1983, pp. 310–321.
3. Franklin, Joel N.: *Matrix Theory.* Prentice-Hall, Inc., c.1968.
4. Camarda, Charles J.; and Adelman, Howard M.: *Static and Dynamic Structural-Sensitivity Derivative Calculations in the Finite-Element-Based Engineering Analysis Language (EAL) System.* NASA TM-85743, 1984.
5. Lyness, J. N.: Has Numerical Differentiation a Future? *Proceedings of the Seventh Manitoba Conference on Numerical Mathematics and Computing,* Univ. of Manitoba, Winnipeg, 1977, pp. 107–129.
6. Whetstone, W. D.: *EISI-EAL Engineering Analysis Language Reference Manual—EISI-EAL System Level 2091.* Engineering Information Systems, Inc., July 1983.
7. Schmit, Lucien A., Jr.; and Miura, Hirokazu: *Approximation Concepts for Efficient Structural Synthesis.* NASA CR-2552, 1976.

Standard Bibliographic Page

1. Report No. NASA TM-86382	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Selecting Step Sizes in Sensitivity Analysis by Finite Differences		5. Report Date August 1985
		6. Performing Organization Code 506-53-53-07
7. Author(s) Jocelyn Iott, Raphael T. Haftka, and Howard M. Adelman		8. Performing Organization Report No. L-15938
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665		10. Work Unit No.
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Technical Memorandum
		14. Sponsoring Agency Code

15. Supplementary Notes

Jocelyn Iott and Raphael T. Haftka: Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
Howard M. Adelman: Langley Research Center, Hampton, Virginia.

16. Abstract

This paper deals with methods for obtaining near-optimum step sizes for finite difference approximations to first derivatives with particular application to sensitivity analysis. A technique denoted the finite difference (FD) algorithm, previously described in the literature and applicable to one derivative at a time, is extended to the calculation of several derivatives simultaneously. Both the original and extended FD algorithms are applied to sensitivity analysis for a data-fitting problem in which derivatives of the coefficients of an interpolation polynomial are calculated with respect to uncertainties in the data. The methods are also applied to sensitivity analysis of the structural response of a finite-element-modeled swept wing. In a previous study, this sensitivity analysis of the swept wing required a time-consuming trial-and-error effort to obtain a suitable step size, but it proved to be a routine application for the extended FD algorithm herein.

17. Key Words (Suggested by Authors(s)) Sensitivity analysis Finite element analysis Finite difference method Optimization methods Structural analysis	18. Distribution Statement Unclassified—Unlimited Subject Category 39		
19. Security Classif.(of this report) Unclassified	20. Security Classif.(of this page) Unclassified	21. No. of Pages 12	22. Price A02

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1985

End of Document