MATHEMATICS OF COMPUTATION, VOLUME 33, NUMBER 148
OCTOBER 1979, PAGES 1257–1264

Adaptive Numerical Differentiation

By R. S. Stepleman and N. D. Winarsky

Abstract. It is well known that the calculation of an accurate approximate derivative $f'(x)$ of a nontabular function $f(x)$ on a finite-precision computer by the formula $d(h) = (f(x + h) - f(x - h))/2h$ is a delicate task. If h is too large, truncation errors cause poor answers, while if h is too small, cancellation and other "rounding" errors cause poor answers. We will show that by using simple results on the nature of the asymptotic convergence of $d(h)$ to f', a reliable numerical method can be obtained which can yield efficiently the theoretical maximum number of accurate digits for the given machine precision.

1. Introduction. Let $f\colon R \longrightarrow R$ be differentiable at a point x. One of the simplest and most common ways discussed in the literature to find an approximate value of the derivative f' is to choose an h and define

$$d(h) = \frac{f(x + h) - f(x - h)}{2h}, \tag{1.1}$$

and then use (1.1) as our approximation. However, the crucial step in (1.1), the choice of h, is fraught with uncertainty when (1.1) is performed on a finite-precision digital computer. If h is chosen too small, the finite precision of the computer can cause cancellation and other "rounding" errors to be made that result in poor answers; while if h is chosen too large, $d(h)$ need not be a good approximation to f' even in "exact" arithmetic. Several authors have given algorithms that attempt to deal with this situation, see, e.g., Curtis and Reid [1], Dahlquist and Bjorck [2], Dumontet and Vignes [3], and Oliver and Ruffhead [4]. Often they are based on estimating the amount of rounding and truncation error made. Since these can be related to various derivatives of f, these must be explicitly estimated, e.g., Dumontet and Vignes [3] construct an algorithm that is based on a clever scheme to estimate f''' by difference quotients.

We propose a simple device that avoids any explicit estimation of rounding and truncation errors (and any unreliabilities inherent in that approach) and allows the computer to find, adaptively, the best value that it can for h, the "one" that gives the least error in the approximate derivative. In the next section we prove the following simple results: choose $\{h_i\}$ any sequence tending monotonically to zero (with h_0 sufficiently small), then for essentially all differentiable f (see Theorem 2.4 for technical

Received November 7, 1978.
AMS (MOS) subject classifications (1970). Primary 65D25.

details) $d(h_i)$ tends monotonically to f'. Under stronger hypotheses we also have $|d(h_i) - d(h_{i+1})| \leqslant |d(h_i) - d(h_{i-1})|$, for all i. The basic idea of our algorithm (see Section 3 for details) is: choose a sequence $\{h_i\}$ tending monotonically to zero. Let h_j be the first h_i in the sequence so that the finite precision of the computer causes either of the above criteria to be violated, then approximate f' by $d(h_{j-1})$. Essentially, we are using violation of monotonicity as a stopping criterion for a numerical process that would not necessarily terminate in exact arithmetic. (It is clear that it does terminate in finite precision since for all h sufficiently small, $h \neq 0$ we have $F(X \oplus H) = F(X \ominus H)$ and, thus, $D(H) = 0$. The capital letters here represent the finite-precision representations of the small lettered quantities. While the circled operators represent the machine approximations to the arithmetic operators.) That this algorithm is effective and competitive with known algorithms will be demonstrated in Section 4. The idea of using violation of a criterion such as monotonicity as a way of terminating a numerical process on a finite-precision computer at an "accurate" answer has applications more general than numerical differentiation and can be applied to iterative as well as noniterative processes (see Rutishauser [6], and Stepleman [7], [8]).

2. **Theoretical Foundation.** In this section we prove, for completeness, several simple and useful results that give a firm basis to the algorithm outlined in the next section.

Definition 2.1. Let g: $R \longrightarrow R$ and $x_0 \in R$. We call g well behaved at x_0, denoted by $g \in \mathrm{WB}(x_0)$, if there exist two intervals $I_1 = [x_0, x_0 + \epsilon)$ and $I_2 = (x_0 - \epsilon, x_0]$, $\epsilon > 0$, such that g is either convex or concave on I_1 and independently convex or concave on I_2.

The following two lemmas whose proofs are well known, (see, e.g., Ortega and Rheinboldt [5]) will help us understand and use well behaved functions.

LEMMA 2.2. *Let g: $R \longrightarrow R$ and $g \in C^1(I)$, I some interval. Then g convex on I is equivalent to g' monotone increasing on I, while g concave on I is equivalent to g' monotone decreasing on I.*

LEMMA 2.3. *Let g: $R \longrightarrow R$ and $g \in C^2(I)$, I some interval. Then g convex on I is equivalent to $g'' \geqslant 0$ on I, while g concave on I is equivalent to $g'' \leqslant 0$ on I.*

Using the above lemmas, it is not difficult to see that most functions are in $\mathrm{WB}(x_0)$. Exceptions are functions that wiggle infinitely often such as $x^2 \sin(1/x)$ in the neighborhood of zero; for example, $g''(x_0) \neq 0$ implies $g \in \mathrm{WB}(x_0)$.

THEOREM 2.4. *Let f: $R \longrightarrow R$ be in $C^1(I)$, $I = (x_0 - \epsilon, x_0 + \epsilon)$, for some $\epsilon > 0$ arbitrary. Assume*

$$g(t) = f(x_0 + t) - f(x_0 - t) \tag{2.1}$$

satisfies $g \in \mathrm{WB}(0)$. Then for $h > 0$ sufficiently small, $d(h)$ converges monotonically to $f'(x_0)$.

Proof. Since $d(h) = g(h)/(2h)$, simple differentiation yields

$$d(h)' = \frac{g'(h)h - g(h)}{2h^2} = \frac{g'(h)h - [g(h) - g(0)]}{2h^2}$$

$$= \frac{[g'(h) - g'(t)]}{2h}, \qquad 0 < t < h. \tag{2.2}$$

Then since $g \in \mathrm{WB}(0)$, we have for h sufficiently small that $d(h)'$ has exactly one sign from Lemma 2.2. Since for $h_1 > h_2$

$$d(h_1) - d(h_2) = \int_{h_2}^{h_1} d(s)'ds,$$

the monotonicity follows. The rest of the result follows since it is well known that $\lim d(h) = f'(x_0)$ for $f \in C^1(I)$.

LEMMA 2.5. *Let g defined by (2.1) be in $C^2(I)$, $I = [0, \epsilon)$, $\epsilon > 0$ arbitrary and both g and $g' \in \mathrm{WB}(0)$. Then*

$$F(h) = \left| g'(h) - \frac{g(h)}{h} \right| \tag{2.3}$$

is monotone increasing for h sufficiently small.

Proof. Since $g \in \mathrm{WB}(0)$, we can without loss of generality assume g is convex for h sufficiently small and then from Lemma 2.2

$$F(h) = g'(h) - \frac{g(h)}{h}.$$

Differentiation yields

$$F'(h) = g''(h) - \frac{g'(h) - g(h)/h}{h}$$

$$= g''(h) - \frac{g'(h) - g'(t)}{h}, \qquad 0 < t < h, \tag{2.4}$$

$$= g''(h) - g''(s)(1 - t/h), \qquad t < s < h.$$

Since $g' \in \mathrm{WB}(0)$ it follows that g'' is monotone, while g convex implies $g'' \geqslant 0$ from Lemma 2.3, and (2.1) gives $g''(0) = 0$. If g'' is monotone increasing $F'(h) \geqslant 0$ follows from (2.4), while if g'' is monotone decreasing it follows that $F'(h) = 0$ since $g''(0) = 0$ and $g'' \geqslant 0$. In any case, the conclusion follows.

THEOREM 2.6. *Let the hypothesis of Lemma 2.5 hold. In addition suppose we have a positive monotone decreasing sequence $\{h_i\}$ that satisfies for any $\beta > 1$*

$$h_{i+1} = \frac{h_i}{\beta}, \qquad i = 0, 1, 2, \ldots. \tag{2.5}$$

Then for all $i \geqslant N$, N some positive integer,

$$|d(h_{i+1}) - d(h_i)| \leqslant |d(h_i) - d(h_{i-1})|. \tag{2.6}$$

Proof. As in Theorem 2.4,

$$|d(h_i) - d(h_{i-1})| = \int_{h_i}^{h_{i-1}} \frac{1}{2h} \left| g'(h) - \frac{g(h)}{h} \right| dh.$$

Using Lemma 2.5, we have for $i \geqslant N$ some positive integer, that

$$|d(h_i) - d(h_{i-1})| \geqslant \ln(h_{i-1}/h_i) \frac{|g'(h_i) - g(h_i)/h_i|}{2} . \tag{2.7}$$

Here ln is the natural logarithm function. Similarly,

$$\begin{aligned} |d(h_i) - d(h_{i+1})| &= \int_{h_{i+1}}^{h_i} \frac{1}{2h} \left| g'(h) - \frac{g(h)}{h} \right| dh \\ &\leqslant \ln(h_i/h_{i+1}) \frac{|g'(h_i) - g(h_i)/h_i|}{2} . \end{aligned} \tag{2.8}$$

Then the conclusion follows from (2.5), (2.7) and (2.8).

3. The Algorithm. In this section we discuss the practical implementation of the algorithm suggested in the last section. In order to do this we must make particular choices of the two numbers h_0 and β. The quantity h_0 must be chosen small enough so that we are in the asymptotic range of $d(h)$ approximating f' (i.e., Theorem 2.4 and Theorem 2.6 holding), while not so small that the rounding error is the dominant part of the total error. The quantity $\beta > 1$ must be chosen to avoid the danger of having an abrupt transition from $D(H_i)$ dominated by truncation error to $D(H_{i+1})$, dominated by rounding errors because this could cause the algorithm to terminate before $D(H_i)$ is a good approximation to f'. Here, of course,

$$D(H) = (F(X \oplus H) \ominus F(X \ominus H)) \oslash (2 \otimes H), \tag{3.1}$$

where the capital letters represent the finite-precision representation of the small lettered quantities and the operators represent the machine approximation to the standard arithmetic operators.

Our algorithm contains an iteration to choose the delicate quantity h_0; because of this we need only a reasonable first guess at h_0, and we now give a heuristic discussion of our choice. A main source of error in the approximate derivative is the calculation of $D(H)$ instead of $d(h)$.

It is clear that given any x we will have a nonempty set of positive h such that $X \oplus H = X$ and thus $D(H) = 0$. For h in this set all significance is lost. For h not in this set we can estimate the dominant rounding error (following the analysis of Dumontet and Vignes [3]) assuming

$$F(x) - f(x) = \epsilon(x) f(x) \tag{3.2}$$

with

$$|\epsilon(x)| \leqslant P, \qquad P > 0, \tag{3.3}$$

and that for h we do not use H but

$$\tilde{H} = (H \oplus X) \ominus X. \tag{3.4}$$

The result is that the rounding error e_r satisfies

$$e_r \equiv D(\tilde{H}) - d(h) \doteq \frac{F(X)}{2h}(\epsilon_1 - \epsilon_2), \qquad \epsilon_1, \epsilon_2 \in [-P, P]. \tag{3.5}$$

The standard estimate of the truncation error e_t is

$$\begin{aligned} e_t &\equiv d(h) - f'(x) = \frac{1}{6} f'''(y)h^2, \qquad y \in (x - h, x + h), \\ &\doteq \frac{1}{6} f'''(x)h^2. \end{aligned} \tag{3.6}$$

Dumontet and Vignes [3] show that the expected value of the absolute value of the total error is

$$\overline{|e_r + e_t|} \doteq \frac{P|F(X)|}{3h} + \frac{h^5(f'''(x))^2}{36P|F(X)|} - \frac{h^8|f'''(x)|^3}{648P^2(F(X))^2}, \quad \text{for } P > \left|\frac{h^3 f'''(x)}{GF(x)}\right|. \tag{3.7}$$

To find a starting guess for h_0 we calculate an estimate for h_P the h that minimizes (3.7). This is given by

$$h_P = \left(\frac{1.67P|F(X)|}{f'''(x)}\right)^{1/3} = O(P^{1/3}). \tag{3.8}$$

To check for violation of monotonicity on convergence of $D(h)$ to f' we need to calculate at least h_0, h_1 and h_2 (and hopefully no more); and thus, we would like $h_0 \doteq \beta h_P$. Thus, using this as a heuristic guide, we choose as our starting guess for h_0

$$h_0 = \beta P^{1/3} x. \tag{3.9}$$

The x appears as a scaling factor to insure $X \oplus H_0 \neq X$ for $x \neq 0$; note that $X \oplus H_P = X$ is certainly possible.

We next consider how to decide if (3.9) is an adequate choice for h_0. The loss of figures due to the subtraction in (3.1) is the main contributor to e_r for useful h, and this can be measured computably by

$$\delta(h) = |(2 \otimes H \otimes D(H)) \oslash F(X)|. \tag{3.10}$$

Thus, the number of digits $N(h)$ lost in the subtraction is

$$N(h) = -\log(\delta(h)). \tag{3.11}$$

Observe that (3.5) yields

$$\delta(h) = \frac{2H}{F(X)}|e_r + d(h)| \doteq \left|\frac{2f'(x)}{F(X)} h\right|; \qquad h = O(h_P). \tag{3.12}$$

Thus, (3.8) and (3.12) give

$$N(h_P) \doteq \log(P^{-1/3}). \tag{3.13}$$

Note also that (3.12) implies that, if we have lost $N(h)$ digits at h, we can expect that

$$N\left(\frac{h}{\beta}\right) \doteq N(h) + \log \beta; \qquad N(h\beta) \doteq N(h) - \log \beta. \tag{3.14}$$

Since we need rounding error to be no worse than equal to truncation error at h_1, we want $N(h_1) \leqslant N(h_P)$. Thus, using (3.13) and (3.14), we would like

$$N(h_0) \leqslant N(h_P) - \log \beta = \log\left(\frac{P^{-1/3}}{\beta}\right). \tag{3.15}$$

We would also like to be sure that we have lost some digits, so that we know our starting guess for h_0 is not grossly large. Thus, we want

$$N(h_0) > 0. \tag{3.16}$$

If (3.15) and (3.16) are not both satisfied for (3.9), we use (3.14) and a bisection type zero finder to find such an h_0 using (3.9), as the initial guess. This is done in the following way: Suppose $N(h_0)$ does not satisfy (3.15). Then use (3.14) to predict a new larger h_0^1 that will. For this new h_0^1 calculate $N(h_0^1)$ by (3.10) and (3.11). If both (3.15) and (3.16) are satisfied, then we stop. If (3.15) is still not satisfied, then we repeat the loop. If (3.16) is not satisfied, we define $h_0^2 = (h_0 + h_0^1)/2$ and use the bisection method until both (3.15) and (3.16) are satisfied.

We next consider the choice of β. The only restriction we have now is $\beta > 1$. However, we do not want β overly large since this would mean, as we stated earlier, large changes in $N(h)$. This is quantified by (3.14). Our experience shows the algorithm is not very sensitive to the choice of β for any single digit β. We chose $\beta = 4$ for our results in the next section (i.e., approximately .6 digits change from $N(h_i)$ to $N(h_{i+1})$).

It is also possible to get an error estimate for the relative error, using (3.5) followed by (3.10) to obtain

$$\left|\frac{D(\tilde{H}) - d(h)}{d(h)}\right| = \left|\frac{e_r}{d(h)}\right| \doteq \frac{2P}{\delta(h)}. \tag{3.17}$$

At h near the optimal h we would expect the truncation error (3.6) to be about equal to the rounding error so that a computable error estimate is

$$E_T = \frac{4P}{\delta(h)}. \tag{3.18}$$

A simplified flowchart of the algorithm appears in Figure 1. Note that if $f(x + h)$ and $f(x - h)$ are different in sign there is no cancellation error so we cannot control on it. If it is not desired to obtain full accuracy of the derivative (which is, by (3.13), $\log P^{-2/3}$ digits) but only N "significant" figures or "accurate" digits, one can add the appropriate relative or absolute accuracy test and stop the process at this point.

The point $x = 0$ is a special case that does not fit into the discussion of this section using (3.9). For this x we choose our starting guess arbitrarily. For the results in the next section this point does not occur. In practice, we have found an $h_0 = .01\beta P^{1/3}$ to be satisfactory.

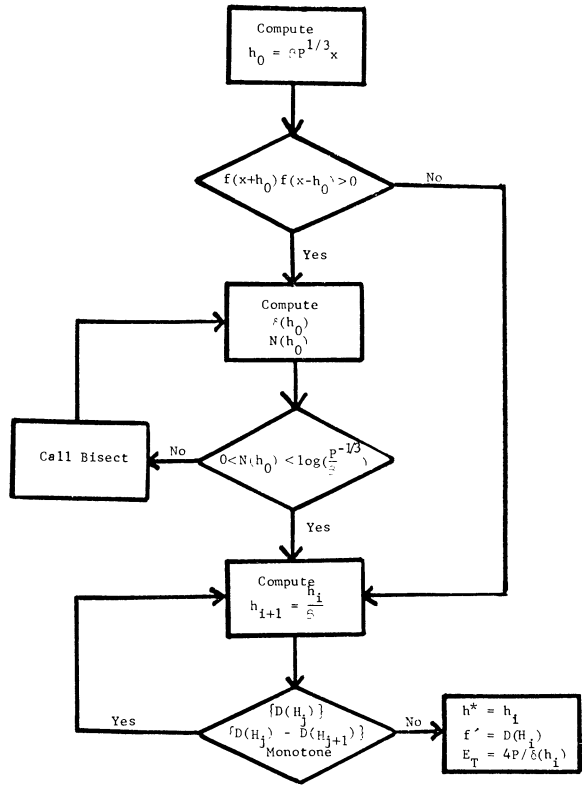

FIGURE 1

4. Numerical Experience. In this section we compare our numerical experienc with that of Dumontet and Vignes [3]. For each of the functions listed in the table v we found the derivative at 100 equally spaced values of $x \in [.1, 12.5]$. This was done on the IBM 370/168 in double precision, i.e., $P \doteq 1 \times 10^{-16}$. The following table presents our results on the averages of these 100 values for each function.

Function	Average Relative Error	Average Number of Function Evaluations	
e^x	$1.83 \cdot 10^{-11}$	8.94	(15)
Log x	$2.40 \cdot 10^{-11}$	7.86	(17)
$\sqrt{x}$	$8.74 \cdot 10^{-12}$	10.52	(15)
$\tan^{-1}x$	$9.43 \cdot 10^{-11}$	10.18	(20)
$\sin x$	$1.54 \cdot 10^{-11}$	9.12	(15)

The numbers in parentheses are those for the method of Dumontet and Vignes [3]. Thus, we see that at a cost of about 10 function evaluations we can get a derivative correct to about 11 significant figures for these elementary functions. This is in line

with what we would expect from the arguments of Section 3. In certain types of problems with repetitive derivative calculation it is possible to reduce the average number of function evaluations dramatically by the following technique. Suppose we have just found the optimal h^* for $f'(x_0)$, and we now want to differentiate at x_1 where x_1 is "close" to x_0. Then we can use h^* as the optimal h at x_0. We need only check to make sure that $N(h^*)$ has not changed "very much". Using this technique, the average number of function evaluations is between 2.1 and 2.8 for the examples in the table with the average relative error about 1×10^{-10}. This type of idea would be applicable to an iteration like Newton's method or evaluating the Jacobian repeatedly in the solution of stiff systems of differential equations.

RCA Laboratories
Princeton, New Jersey 08540

1. A. CURTIS & J. REID, "The choice of step lengths when using differences to approximate Jacobian matrices," *J. Inst. Math. Appl.*, v. 13, 1974, pp. 121–126.

2. G. DAHLQUIST & A. BJORCK, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

3. J. DUMONTET & J. VIGNES, "Determination du pas optimal dans le calcul des dérivées sur ordinateur," *R.A.I.R.O.*, v. 11, 1977, pp. 13–25.

4. J. OLIVER & A. RUFFHEAD, "The selection of interpolation points in numerical differentiation," *BIT*, v. 15, 1975, pp. 283–295.

5. J. ORTEGA & W. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

6. H. RUTISHAUSER, "Description of ALGOL 60," *Handbook for Automatic Computation*, Vol. 1a, Springer-Verlag, New York, 1967.

7. R. STEPLEMAN, "Analysis of convergence for fixed point iterations in R^1," *Proc. 1977 Conf. on Information Sciences and Systems*, John Hopkins University, 1977, pp. 389–394.

8. R. STEPLEMAN, "Monotone convergence and effective stopping criteria for numerical processes," *BIT*. (To appear.)