THE UNIVERSITY of TENNESSEE at CHATTANOOGA
COLLEGE of ENGINEERING and COMPUTER SCIENCE

SIMCENTER
NATIONAL CENTER for COMPUTATIONAL ENGINEERING

Numerical Derivatives, Matrix-Vector Product, and Richardson Extrapolation Using Complex Variables

A Technical Report by
David L. Whitfield
Ramesh Pankajakshan
Kidambi Sreenivas
and
Lafayette K. Taylor

UTC-CECS-SimCenter-2008-01
March 2008

GRADUATE SCHOOL OF COMPUTATIONAL ENGINEERING
701 East M.L. King Boulevard • Chattanooga, TN 37403

Table of Contents

Acknowledgements

In 1984 the State of Tennessee created a Center of Excellence program for public institutions of higher education within the state. The purpose of these Centers, known as THEC (Tennessee Higher Education Commission) Centers of Excellence, is to build on the strength of the State's universities, expand the State's research base, and increase its national and international stature and economic competitiveness. The research reported herein was carried out under the auspices of the THEC Center of Excellence for Applied Computational Science and Engineering under the direction of Dr. Harry McDonald at the University of Tennessee at Chattanooga under the project entitled 'Advanced Computational Methods for Field Simulation Problems' directed by Dr. Roger Briley. This support is gratefully acknowledged. In addition, appreciation is extended to Holley Beeland of the SimCenter: National Center for Computational Engineering for compiling several pages of notes and equations to produce this report.

Abstract

The purpose of this report is to describe some interesting results using complex variables to obtain high-order relations for: (1) numerical derivatives of a function, (2) matrix-vector products, (3) Richardson extrapolation, and (4) the calculation of natural logarithms using differentiation. Derivations are presented for each of these topics along with supporting numerical results. It is intended that this material will serve as a useful reference and teaching aid.

1.0 Introduction

Ever since running across the article by Squire and Trapp [1] in SIAM Review the authors have been intrigued by the possibilities brought about by the use of complex variables to obtain numerical derivatives to, first of all, real functions, but it seems it might also be extended to obtain derivatives of complex functions. Squire and Trapp [1] visited the work of Lyness and Moler [2] and Lyness [3] of some thirty years before (forty years ago now). Squire and Trapp presented in rather simple form the preceding work of Lyness and Moler. This work was put to use immediately in Ref. [4] in obtaining the Jacobian matrix for the implicit numerical solution of nonlinear systems of equations. The history of this process was as follows. This writer took the basic concept to Dr. Kidambi Sreenivas (a PhD student of the writer at the time) and asked him to use this approach to compute the Jacobian matrix in an implicit three-dimensional unsteady unstructured Navier-Stokes code. Within two hours Dr. Sreenivas had this working [5]. Then, this writer took the basic concept to Dr. Ramesh Pankajakshan (an MS student of the writer and a PhD student of Dr. Roger Briley at the time) and Dr. Pankajakshan had this running in a three-dimensional unsteady structured multi-block code in about the same amount of time [6].

Following this experience the author took the idea to Dr. Jim Newman III, closed the door, and told him that this might be of interest for use in the design optimization area (Dr. Newman's specialty) but if not this writer would personally scold him if he ever mentioned this crazy idea to anyone. After patiently listening to the concept Dr. Newman said, "ADIFOR has just been made obsolete." That next summer Dr. Newman went to NASA Langley and worked with Dr. Kyle Anderson (another PhD student of the writer). Results of the summer's work were very successful [7].

The flow of this report is as follows. The basic derivation for obtaining the second-order accurate first derivative of a real function using complex variables is presented in Section 2.0 along with numerical results. Developments for obtaining a matrix-vector product and fourth-order first and second derivatives are also presented in Section 2.0 along with numerical results. The development of Richardson extrapolation using complex variables is presented in Section 3.0 along with numerical results. Section 4.0 is devoted to the calculation of natural logarithms using differentiation. A summary of this effort is given in Section 5.0.

2.0 Numerical Derivatives Using Complex Variables

Section 2.1 contains the development of the second-order accurate first derivative and Section 2.2 is an extension of this to the matrix-vector product. Section 2.3 involves the development of fourth-order accurate first and second derivatives.

2.1 Second-Order Accurate First Derivative

The second-order accurate first derivative using complex variables is developed in Section 2.1.1. However, in computational engineering one is usually concerned with systems of equations, both linear and nonlinear systems. In this case it is frequently necessary to obtain the

Jacobian matrix, for example, which involves the derivative of a vector function which in turn is a function of a vector. This development is presented in Section 2.1.2.

2.1.1 Function of a Single Variable

Squire and Trapp [1] presented an interesting way of determining the derivative of real functions using complex variables. They attribute the idea of using complex variables to develop differentiation formulas to Lyness and Moler [2,3]. The approach presented by Squire and Trapp is the following. Let $f(z)$ be an analytic function of the complex variable *z*, and assume that *f* is real on the real axis. Expand *f* in a Taylor series about the real point *x* to obtain

$$f(x+ih) = f(x) + ihf'(x) - \frac{h^2}{2!} f''(x) - \frac{ih^3}{3!} f'''(x) \cdots \tag{2.1}$$

where $i = \sqrt{-1}$. Taking the imaginary part of both sides of Eq. (2.1) and dividing by *h* gives

$$f'(x) = \frac{\mathrm{Im}\left[f(x+ih) \right]}{h} \tag{2.2}$$

Note that the truncation error of $f'(x)$ in Eq. (2.2) is $O(h^2)$. This is in contrast to the $O(h)$ error in the expression

$$f'(x) = \frac{f(x+h) - f(x)}{h} \tag{2.3}$$

which is currently used to obtain the numerical Jacobian in some versions of the TENASI and UNCLE codes. In fact, Eq. (2.2) gives the same order of accuracy as

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} \tag{2.4}$$

which would take two extra flux evaluations per matrix element as opposed to the one extra required by Eq. (2.3).

It is also interesting to note that whereas the derivative is obtained from Eq. (2.2), the function itself can be determined by taking the real part of Eq. (2.1) to obtain, to second order

$$f(x) = \mathrm{Real}\left[f(x+ih) \right] \tag{2.5}$$

Whether or not the calculation of a flux, for example, from Eq. (2.5) is beneficial in conservation law equation solvers remains to be seen because the flux is required more often, in general, than the derivative of the flux, and consequently the determination of the flux function from Eq. (2.5) may not be economical.

There are certain advantages of determining the elements of the Jacobian matrix using Eq. (2.2) as opposed to using either Eq. (2.3) or (2.4). These advantages are:

1. The truncation error is $O\left(h^2\right)$ as opposed to $O(h)$, which is the accuracy of the current method used given by Eq. (2.3).

2. Only one flux function evaluation per Jacobian element is required to gain second-order truncation error, as opposed to the two that would be required by using Eq. (2.4) to get second order.

3. It is not subject to subtractive cancellation error.

4. The results are insensitive to the value of *h*.

5. Since Eq. (2.2) permits accurate Jacobians without having to use extremely small values of *h*, it is quite likely that many computations that have been carried out in 64 bit arithmetic can now be carried out in 32 bit arithmetic. This will reduce the memory requirements tremendously (like a factor of two in most cases), and on many computers, the CPU time will be reduced by a factor of two.

Two disadvantages of determining the Jacobian elements using Eq. (2.2) are that it may take more CPU time and memory due to the complex arithmetic required. However, the amount of complex arithmetic is small, and consequently, the increase in CPU time and additional memory requirements experienced thus far have been negligible. Therefore, the two disadvantages, CPU time and memory, are negated by the speed and memory gained by the possibility of carrying out the computations in 32 bit arithmetic rather than 64 bit arithmetic.

The use of Eq. (2.2) for determining Jacobians has been used in the numerical solution of two-dimensional inviscid incompressible Euler equations, three-dimensional incompressible Navier-Stokes equations on structured multiblock grids by Pankajakshan [6], and three-dimensional compressible Navier-Stokes equations on unstructured grids by Sreenivas [5]. The experience gained thus far has demonstrated this method is indeed insensitive to the value of *h* used.

2.1.2. Vector Function of Multiple Variables

In this subsection, and in the following Section 2.2, the representation $F(x)$ has the following meaning. Let *x* be a vector defined as $x = \left(x_1,\ x_2,\ \ldots,\ x_n\right)^T$.

$$F'(x) = a_{ij} = \frac{\partial F_i(x)}{\partial x_j} \tag{2.6}$$

where the use of a subscript *i* has no relation to the square root of minus one.

Fulks [8], for example, presents the Taylor series for a function of several variables. Utilizing this Taylor series for any component of $F(x)$, say $F_i(x)$, one has

$$F_i(x+v) = F_i(x) + [v\cdot\nabla]F_i(x) + \frac{1}{2!}[v\cdot\nabla]^2 F_i(x) + \frac{1}{3!}[v\cdot\nabla]^3 F_i(x) + R \tag{2.7}$$

where v is a vector $v = (v_1,\ v_2,\ \ldots,\ v_n)^T$ and ∇ is the gradient operator defined as (see, for example, [9])

$$\nabla = \left(\frac{\partial}{\partial x_1},\ \frac{\partial}{\partial x_2},\ \ldots,\ \frac{\partial}{\partial x_n}\right)^T \tag{2.8}$$

Consider the vector v to be a vector defined as $v=hw$ where h is a constant and $w = (w_1,\ w_2,\ \ldots,\ w_n)^T$. Then Eq. (2.7) becomes

$$F_i(x+hw) = F_i(x) + h[w\cdot\nabla]F_i(x) + \frac{h^2}{2!}[w\cdot\nabla]^2 F_i(x) + \frac{h^3}{3!}[w\cdot\nabla]^3 F_i(x) + R \tag{2.9}$$

Consider the second term on the right side of Eq. (2.9). Expanding this term one has

$$h[w\cdot\nabla]F_i(x) = h\left[w_1\frac{\partial F_i(x)}{\partial x_1} + w_2\frac{\partial F_i(x)}{\partial x_2} + \ldots + w_n\frac{\partial F_i(x)}{\partial x_n}\right] \tag{2.10}$$

If w is taken as the j^{th} unit vector e_j, then Eq. (2.10) becomes

$$h\left[e_j\cdot\nabla\right]F_i(x) = h\frac{\partial F_i(x)}{\partial x_j} \tag{2.11}$$

Therefore, using e_j for w in Eq. (2.9) the Jacobian matrix is

$$F'(x) = a_{ij} = \frac{\partial F_i(x)}{\partial x_j} = \frac{F_i(x+he_j) - F_i(x)}{h} + 0(h) \tag{2.12}$$

If the vector v is taken as $v = ihw$ (where $i=\sqrt{-1}$) in Eq. (2.7) instead of $v = hw$, then the equation analogous to Eq. (2.9) is

$$F_i\left(x+ihw\right)=F_i\left(x\right)+ih\left[w\cdot\nabla\right]F_i\left(x\right)-\frac{h^2}{2!}\left[w\cdot\nabla\right]^2F_i\left(x\right)$$
$$-\frac{ih^3}{3!}\left[w\cdot\nabla\right]^3F_i\left(x\right)+R \tag{2.13}$$

By again taking w to be the j^{th} unit vector e_j, and following the same argument as above, the Jacobian matrix can be obtained by taking the imaginary part of both sides of Eq. (2.13) (just as was done to obtain Eq. (2.2) from Eq. (2.1)) to yield

$$F'\left(x\right)=a_{ij}=\frac{\partial F_i\left(x\right)}{\partial x_j}=\frac{\operatorname{Im}\left[F_i\left(x+he_j\right)\right]}{h}+0\left(h^2\right) \tag{2.14}$$

Note that Eq. (2.14) is second-order accurate as opposed to first-order accurate like Eq. (2.12). The reason being, of course, that the $0\left(h^2\right)$ term in Eq. (2.13) is real and thus eliminated by taking the imaginary part of Eq. (2.13). Perhaps most important is that, like Eq. (2.2), Eq. (2.14) has no subtractive cancellation error.

2.2 Matrix-Vector Product

There is frequently the need in numerical computations to evaluate a matrix-vector product such as $F'\left(x\right)w$. This need occurs, for example, in GMRES and various Newton-iterative solvers [10, 11]. An expression for $F'\left(x\right)w$ can be determined by returning to Eq. (2.9) and (2.10). Note Eq. (2.10) is simply the scalar h times the product of the i^{th} row of $F'\left(x\right)$ and w. Using Eq. (2.10) in Eq. (2.9) one has

$$F_i\left(x+hw\right)=F_i\left(x\right)+h\left[i^{th} row\, of\; F'\left(x\right)\right]w+0\left(h^2\right) \tag{2.15}$$

Since the $F_i\left(x\right)$ are components of $F\left(x\right)$, then

$$F'\left(x\right)w=\frac{F\left(x+hw\right)-F\left(x\right)}{h}+0\left(h\right) \tag{2.16}$$

A more accurate expression for $F'\left(x\right)w$ can be obtained by returning to Eq. (2.13) where the vector v was taken as ihw (where $i=\sqrt{-1}$) instead of $v=hw$ as in Eq. (2.9). Using the same argument leading to Eq. (2.15) one has, analogous to Eq. (2.15)

$$F_i\left(x+ihw\right)=F_i\left(x\right)+ih\left[i^{th} row\, of\ F'\left(x\right)\right]w$$
$$-\frac{h^2}{2!}\left[w\cdot\nabla\right]^2 F_i\left(x\right)-\frac{ih^3}{3!}\left[w\cdot\nabla\right]^3 F_i\left(x\right)+R \tag{2.17}$$

By noting once again that $F_i\left(x\right)$ are simply components of $F\left(x\right)$ one has, by taking the imaginary part of Eq. (2.17)

$$F'\left(x\right)w=\frac{\text{Im}\left[F\left(x+ihw\right)\right]}{h}+0\left(h^2\right) \tag{2.18}$$

Equation (2.18) is obviously a second-order accurate expression for $F'\left(x\right)w$ as opposed to the first-order accurate expression given by Eq. (2.16). Moreover, there is no subtraction cancellation error as there is in Eq. (2.16).

2.3 Fourth-Order Accurate First and Second Derivatives

To obtain the fourth-order accurate first derivative consider the Taylor series expansions

$$f\left(x\pm h\right)=f\left(x\right)+\frac{\left(\pm h\right)}{1!}f'\left(x\right)+\frac{\left(\pm h\right)^2}{2!}f''\left(x\right)+\frac{\left(\pm h\right)^3}{3!}f'''\left(x\right)$$
$$+\frac{\left(\pm h\right)^4}{4!}f^{IV}\left(x\right)+\frac{\left(\pm h\right)^5}{5!}f^{V}\left(x\right)+\frac{\left(\pm h\right)^6}{6!}f^{VI}\left(x\right)+0\left(h^7\right) \tag{2.19}$$

and

$$f\left(x\pm ih\right)=f\left(x\right)+\frac{\left(\pm ih\right)}{1!}f'\left(x\right)+\frac{\left(\pm ih\right)^2}{2!}f''\left(x\right)+\frac{\left(\pm ih\right)^3}{3!}f'''\left(x\right)$$
$$+\frac{\left(\pm ih\right)^4}{4!}f^{IV}\left(x\right)+\frac{\left(\pm ih\right)^5}{5!}f^{V}\left(x\right)+\frac{\left(\pm ih\right)^6}{6!}f^{VI}\left(x\right)+0\left(h^7\right) \tag{2.20}$$

where $i=\sqrt{-1}$. Note that by taking the difference of $f\left(x+h\right)$ and $f\left(x-h\right)$ using Eq. (2.19) and dividing by $2h$ one obtains

$$\frac{f\left(x+h\right)-f\left(x-h\right)}{2h}=f'\left(x\right)+\frac{h^2}{3!}f'''\left(x\right)+\frac{h^4}{5!}f^{V}\left(x\right)+0\left(h^6\right) \tag{2.21}$$

Defining the $f'(x)$ in Eq. (2.21) to be $f'_R(x)$ one has

$$f'_R(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{3!} f'''(x) - \frac{h^4}{5!} f^V(x) + 0(h^6) \tag{2.22}$$

Also note that by taking the difference of $f(x+ih)$ and $f(x-ih)$ using Eq. (2.20) and dividing by $i2h$ one obtains

$$\frac{f(x+ih) - f(x-ih)}{i2h} = f'(x) - \frac{h^2}{3!} f'''(x) + \frac{h^4}{5!} f^V(x) + 0(h^6) \tag{2.23}$$

Defining the $f'(x)$ in Eq. (2.23) to be $f'_C(x)$ one has

$$f'_C(x) = \frac{f(x+ih) - f(x-ih)}{i2h} + \frac{h^2}{3!} f'''(x) - \frac{h^4}{5!} f^V(x) + 0(h^6) \tag{2.24}$$

The summation of Eqs. (2.22) and (2.24), or the average of $f'_R(x)$ and $f'_C(x)$, eliminates the $0(h^2)$ term, hence

$$f'(x) = \frac{f'_R(x) + f'_C(x)}{2} + 0(h^4) \tag{2.25}$$

or

$$f'(x) = \frac{\left[f(x+h) - f(x-h)\right] - i\left[f(x+ih) - f(x-ih)\right]}{2h} + 0(h^4) \tag{2.26}$$

Equation (2.25) or (2.26) has potential subtraction cancellation errors, although $f'(x)$ as given by these equations is fourth-order accurate.

To obtain the fourth-order accurate second derivative consider again the Taylor series expansions given by Eqs. (2.19) and (2.20). Note that by averaging $f(x+h)$ and $f(x-h)$ from Eq. (2.19) one obtains

$$\begin{aligned} f_{R_{avg}}(x) = \frac{f(x+h) + f(x-h)}{2} = f(x) + \frac{h^2}{2!} f''(x) + \frac{h^4}{4!} f^{IV}(x) \\ + \frac{h^6}{6!} f^{VI}(x) + 0(h^8) \end{aligned} \tag{2.27}$$

Taking the average of $f(x+ih)$ and $f(x-ih)$ from Eq. (2.20) results in

$$f_{C_{avg}}(x) = \frac{f(x+ih)+f(x-ih)}{2} = f(x) - \frac{h^2}{2!}f''(x) + \frac{h^4}{4!}f^{IV}(x) - \frac{h^6}{6!}f^{VI}(x) + 0(h^8) \tag{2.28}$$

Subtracting Eq. (2.28) from Eq. (2.27) gives

$$f''(x) = \frac{f_{R_{avg}}(x) - f_{C_{avg}}(x)}{h^2} + 0(h^4) \tag{2.29}$$

or

$$f''(x) = \frac{\left[f(x+h)+f(x-h)\right]-\left[f(x+ih)+f(x-ih)\right]}{2h^2} + 0(h^4) \tag{2.30}$$

Similar to the first derivative, Eq. (2.29) or (2.30) has potential subtraction cancellation errors, although $f''(x)$ as given by these equations is fourth-order accurate.

2.4 Numerical Results

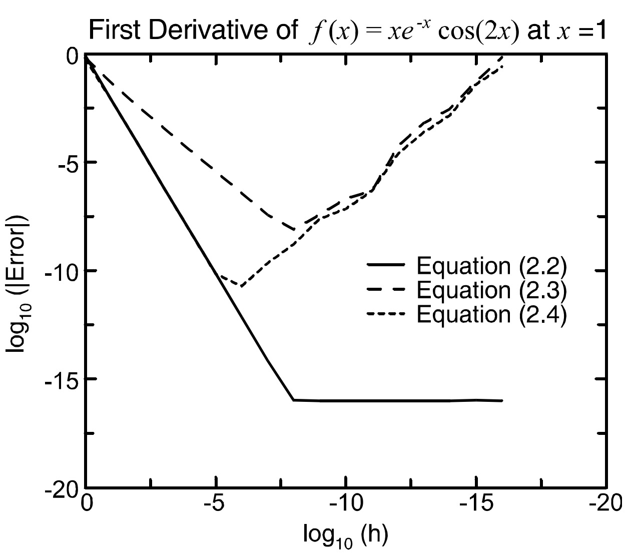

Figure 1

The accuracy afforded by using the complex variable approach for obtaining the first derivative of a scalar function (Eq. (2.2)) versus commonly used finite differences is shown in Fig. 1. The example function is one used in [12] given by

$$f(x) = xe^{-x}\cos(2x) \tag{2.31}$$

The numerical derivative of Eq. (2.31) at $x = 1$ is determined using Eqs. (2.2), (2.3), and (2.4). The most accurate and most dependable derivative is that given by Eq. (2.2) which is based on the use of complex variables and is second-order accurate as illustrated both theoretically and numerically in Fig. 1. The first-order accurate one-sided finite difference numerical derivative given by Eq. (2.3) is shown numerically in Fig. 1 to be just that; i.e. first-order accurate. The second-order central finite difference (Eq. (2.4)) follows the second-order accuracy of Eq. (2.2) to about $h = 10^{-5}$. However, Eq. (2.4) and (2.3) both eventually incur subtraction cancellation errors and cannot be depended upon for small step sizes. On the other hand Eq. (2.2) does not encounter this error and produces high quality results to extremely small step sizes. The same trend as illustrated in Fig. 1 has been observed in practice to be true for each element of the Jacobian matrix as derived in Section 2.1.2.

The accuracy afforded by using the complex variable approach of determining the product of a matrix times a vector versus the commonly used approach is illustrated in Fig. 2. The example vector used is one taken from page 84 of [9]. It was used in [9] as an example for computing the elements of a Jacobian matrix using classical finite differences but it is used here as an example for the computation of $F'(x)w$ where $x=(x_1,x_2)^T$ and $w=(w_1,w_2)^T$. The example vector is

$$F(x)=\begin{bmatrix} F_1(x) \\ F_2(x) \end{bmatrix}=\begin{bmatrix} 3x_1^2-2x_2 \\ x_2^3-\dfrac{1}{x_1} \end{bmatrix} \tag{2.32}$$

The Jacobian matrix for Eq. (2.32) is

$$F'(x)=\frac{\partial F_i(x)}{\partial x_j}=\begin{bmatrix} 6x_1 & -2 \\ \dfrac{1}{x_1^2} & 3x_2^2 \end{bmatrix} \tag{2.33}$$

The analytical value for $F'(x)w$ is therefore

$$F'(x)w=\begin{bmatrix} 6x_1w_1-2w_2 \\ \dfrac{1}{x_1^2}w_1+3x_2^2w_2 \end{bmatrix} \tag{2.34}$$

The norm used here to measure accuracy is the L_2 norm given by the square root of the inner product, or

$$\|u\|=\sqrt{u\bullet u} \tag{2.35}$$

where the vector $u=(u_1,u_2)^T$ is defined as

$$u=\left[F'(x)w\right]_{num}-\left[F'(x)w\right]_{exact} \tag{2.36}$$

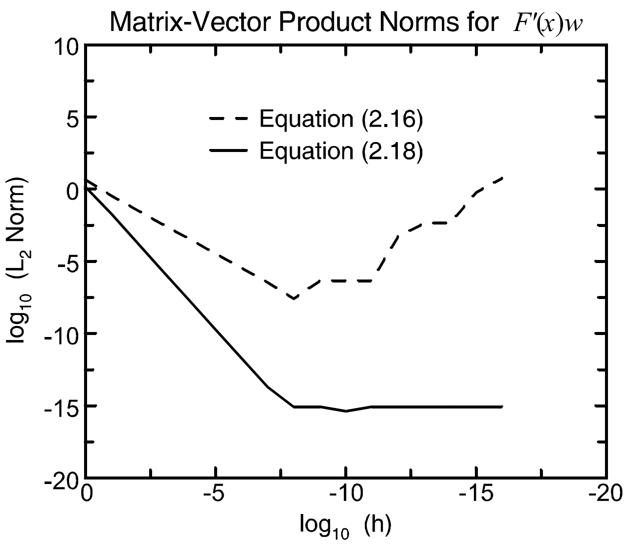

Figure 2

where $\left[F'(x)w\right]_{num}$ is either Eq. (2.16) or Eq. (2.18) and $\left[F'(x)w\right]_{exact}$ is Eq. (2.34). The L_2 norms for *u* defined by Eq. (2.36) are given in Fig. 2 with *x* and *w* taken as unit vectors. Equation (2.16) illustrates the expected first-order accuracy and Eq. (2.18) illustrates the expected second-order accuracy. Equation (2.16) also experiences the problem of subtraction cancellation error, whereas Eq. (2.18) does not.

It should be mentioned that when the vectors *x* and *w* vary greatly in scaling, the use of Eq. (2.18) in place of Eq. (2.16) could be imperative. In the simple example just presented *x* and *w* were taken as unit vectors. However, some problems in engineering and physics might have components of *x* and *w* that vary several orders of magnitude. In such cases Eq. (2.16) might be considerably more in error than Eq. (2.18).

The final numerical results for this section are results for fourth-order first and second derivatives of a function as derived in Section 2.3. The example function used is again that given by Eq. (2.31).

Numerical results for the second and fourth-order accurate first derivatives are given in Fig. 3. The result using Eq. (2.2) is the same as that in Fig. 1 and is included here for reference. The fourth-order accurate first derivative from Eq. (2.25) shows rapid (fourth-order) convergence as expected. Also expected is the eventual subtraction cancellation error that begins at about $h = 10^{-3}$, although convergence is already about 10^{-13}. It is interesting to note the results for the second-order first derivative. Plotted in Fig. 3 is the result from Eq. (2.22), which is the same as the result for Eq. (2.4) plotted in Fig. 1. A second-order result based on complex variables can be derived using Eq. (2.24) to obtain

$$f_c'(x) = \text{Real}\left[\frac{f(x+ih) - f(x-ih)}{i2h}\right] + 0\left(h^2\right) \qquad (2.37)$$

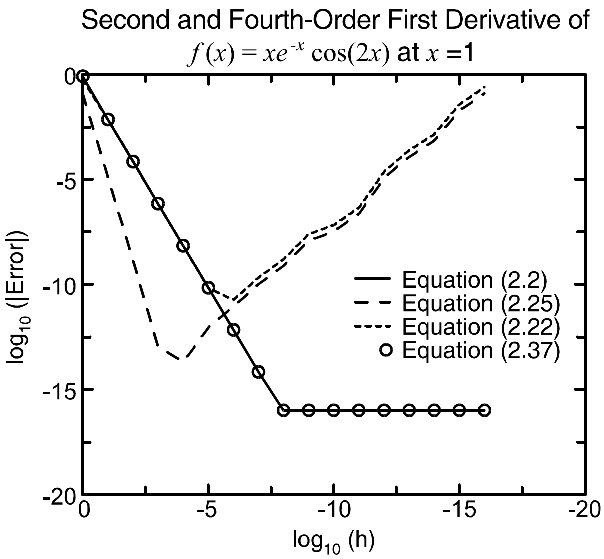

Figure 3

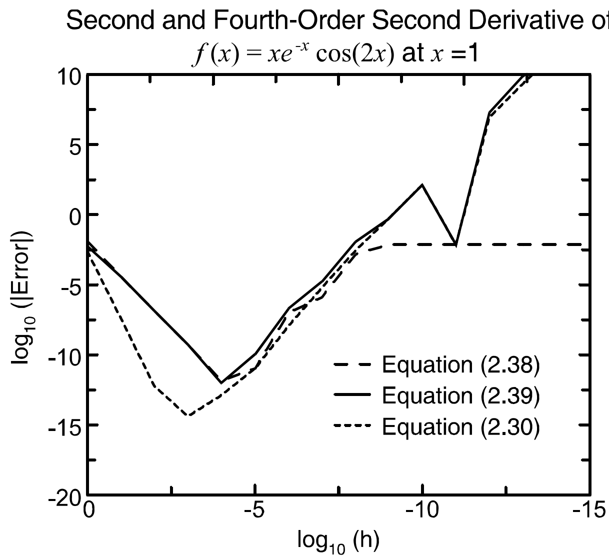

Figure 4

Although Eq. (2.37) is second-order accurate and should be susceptible to subtraction cancellation error, it shows no sign of such error in Fig. 3 and produces the same result as Eq. (2.2). Several evaluation points other than $x=1$ were investigated and in all cases the results produced by Eqs. (2.2) and (2.37) matched. This is indeed a curious result, which is under investigation.

Three equations are investigated numerically for second and fourth-order accurate second derivatives. The first is the second-order accurate classical finite difference expression which can be derived from Eq. (2.27)

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + 0(h^2) \tag{2.38}$$

Another interesting second-order accurate expression can be derived from Eq. (2.28) which is

$$f''(x) = -\text{Real}\left[\frac{f(x+ih) - 2f(x) + f(x-ih)}{h^2}\right] + 0(h^2) \tag{2.39}$$

The third equation is Eq. (2.30) which is fourth-order accurate. All three of these equations are evaluated numerically for the example function given by Eq. (2.31) at $x=1$ and the results are plotted in Fig. 4. All three of these expressions exhibit the expected order of accuracy and subtraction cancellation error. The subtraction cancellation error for the fourth-order accurate second derivative (Eq. (2.30)) occurs at a rather large value of $h=10^{-2}$. The convergence, however, is already greater than eight orders of magnitude.

3.0 Richardson Extrapolation Using Complex Variables

Richardson extrapolation, or deferred approach to the limit [13], is a powerful tool used in many areas of numerical analysis. This extrapolation method is mentioned and used in nearly all books on numerical analysis but few seem to give details on the method, such as the error term. Perhaps one of the best books on the subject is Henrici [14]. Richardson extrapolation as developed below is based on the use of complex variables and hence differs form any other previous work so far as known. The development here, therefore, cannot follow Henrici [14] exactly, but an attempt will be made to follow Henrici [14] as much as practical. The method is developed in Section 3.1 followed by numerical results in Section 3.2.

3.1 Development of the Method

Consider the following expansion

$$A\left(i\delta^m h\right) = a_0 + a_1\left(i\delta^m h\right) + a_2\left(i\delta^m h\right)^2 + \cdots + a_k\left(i\delta^m h\right)^k + \ldots \tag{3.1}$$

where $0 < \delta < 1$, $i = \sqrt{-1}$, and $m = 0,1,2,\ldots$. The quantity of interest is a_0 which results in the limit of $A\left(i\delta^m h\right)$, i.e.

$$\lim_{h\to 0} A\left(i\delta^m h\right) = a_0 \tag{3.2}$$

Classical Richardson extrapolation begins by eliminating the $O(h)$ term from an expression similar to Eq. (3.1) but without *i* and with *m=0*. However, in what follows the $O(h)$ term will be eliminated by taking the real part of a complex expression so the focus here will be on eliminating the $O\left(h^2\right)$ term, which can be accomplished as follows

$$\begin{aligned} A\left(i\delta^m h\right) - \delta^2 A\left(i\delta^{m-1}h\right) &= a_0 - \delta^2 a_0 + a_1 i\left[\left(\delta^m h\right) - \delta^2\left(\delta^{m-1}h\right)\right] \\ &+ a_2 i^2\left[\left(\delta^m h\right)^2 - \delta^2\left(\delta^{m-1}h\right)^2\right] + a_3 i^3\left[\left(\delta^m h\right)^3 - \delta^2\left(\delta^{m-1}h\right)^3\right] \\ &+ a_4 i^4\left[\left(\delta^m h\right)^4 - \delta^2\left(\delta^{m-1}h\right)^4\right] + \ldots + a_k i^k\left[\left(\delta^m h\right)^k - \delta^2\left(\delta^{m-1}h\right)^k\right] + \ldots \end{aligned} \tag{3.3}$$

Note that the $O\left(h^2\right)$ term is eliminated from Eq. (3.3). Dividing Eq. (3.3) by $\left(1-\delta^2\right)$ yields

$$\begin{aligned}\frac{A\left(i\delta^m h\right)-\delta^2 A\left(i\delta^{m-1}h\right)}{1-\delta^2} &= a_0 + a_1 i\frac{\left(1-\delta\right)}{\left(1-\delta^2\right)}\left(\delta^m h\right) \\ &+a_3 i^3\frac{\left(1-\delta^{-1}\right)}{\left(1-\delta^2\right)}\left(\delta^m h\right)^3 + a_4 i^4\frac{\left(1-\delta^{-2}\right)}{\left(1-\delta^2\right)}\left(\delta^m h\right)^4 \\ &+\ldots+a_k i^k\frac{\left(1-\delta^{2-k}\right)}{\left(1-\delta^2\right)}\left(\delta^m h\right)^k+\ldots\end{aligned} \tag{3.4}$$

By taking the real part of Eq. (3.4) the quantity a_0 can be obtained to $O\left(h^4\right)$. However, before doing that it is convenient to compress the notation. Define

$$A_{m,0} = A\left(i\delta^m h\right) \tag{3.5}$$

where $m = 0, 1, 2, \ldots,$ and

$$A_{m,q+1} = \frac{A_{m,q}-\delta^{2(q+1)}A_{m-1,q}}{1-\delta^{2(q+1)}} \tag{3.6}$$

where $q = 0, 1, 2, \ldots, m-1$ and $m > q$. The exponent of δ differs from Henrici [14] because only even powers of δ will be eliminated in this development.

Equation (3.4) is therefore $A_{m,1}$, i.e. q = 0. To eliminate the $O\left(h^4\right)$ term in Eq. (3.4) consider the following which corresponds to $q = 1$

$$\left(1-\delta^4\right)A_{m,2} = A_{m,1}-\delta^4 A_{m-1,1} \tag{3.7}$$

Using Eq. (3.4)

$$\begin{aligned}\left(1-\delta^4\right)A_{m,2} &= a_0-\delta^4 a_0+a_1 i\frac{(1-\delta)}{\left(1-\delta^2\right)}\left[\left(\delta^m h\right)-\delta^4\left(\delta^{m-1}h\right)\right]\\ &+a_3 i^3\frac{\left(1-\delta^{-1}\right)}{\left(1-\delta^2\right)}\left[\left(\delta^m h\right)^3-\delta^4\left(\delta^{m-1}h\right)^3\right]+a_4 i^4\frac{\left(1-\delta^{-2}\right)}{\left(1-\delta^2\right)}\left[\left(\delta^m h\right)^4-\delta^4\left(\delta^{m-1}h\right)^4\right]\\ &+a_5 i^5\frac{\left(1-\delta^{-3}\right)}{\left(1-\delta^2\right)}\left[\left(\delta^m h\right)^5-\delta^4\left(\delta^{m-1}h\right)^5\right]+a_6 i^6\frac{\left(1-\delta^{-4}\right)}{\left(1-\delta^2\right)}\left[\left(\delta^m h\right)^6-\delta^4\left(\delta^{m-1}h\right)^6\right]\\ &+\ldots+a_k i^k\frac{\left(1-\delta^{2-k}\right)}{\left(1-\delta^2\right)}\left[\left(\delta^m h\right)^k-\delta^4\left(\delta^{m-1}h\right)^k\right]+\cdots\end{aligned}\tag{3.8}$$

Note that the $O\left(h^4\right)$ term in Eq. (3.8) is eliminated giving

$$\begin{aligned}A_{m,2} &= a_0+a_1 i\frac{(1-\delta)\left(1-\delta^3\right)}{\left(1-\delta^2\right)\left(1-\delta^4\right)}\left(\delta^m h\right)+a_3 i^3\frac{\left(1-\delta^{-1}\right)(1-\delta)}{\left(1-\delta^2\right)\left(1-\delta^4\right)}\left(\delta^m h\right)^3\\ &+a_5 i^5\frac{\left(1-\delta^{-3}\right)\left(1-\delta^{-1}\right)}{\left(1-\delta^2\right)\left(1-\delta^4\right)}\left(\delta^m h\right)^5+a_6 i^6\frac{\left(1-\delta^{-4}\right)\left(1-\delta^{-2}\right)}{\left(1-\delta^2\right)\left(1-\delta^4\right)}\left(\delta^m h\right)^6\\ &+\ldots+a_k i^k\frac{\left(1-\delta^{2-k}\right)\left(1-\delta^{4-k}\right)}{\left(1-\delta^2\right)\left(1-\delta^4\right)}\left(\delta^m h\right)^k+\cdots\end{aligned}\tag{3.9}$$

Equation (3.9) can be written

$$A_{m,2}=a_0+\sum_{p=1}^{k}a_p i^p\frac{\left(1-\delta^{2-p}\right)\left(1-\delta^{4-p}\right)}{\left(1-\delta^2\right)\left(1-\delta^4\right)}\left(\delta^m h\right)^p+O\left(h^{k+1}\right)\tag{3.10}$$

The general expression for $A_{m,q+1}$ is

$$A_{m,q+1}=a_0+\sum_{p=1}^{k}a_p i^p\frac{\left(1-\delta^{2-p}\right)\left(1-\delta^{4-p}\right)\ldots\left(1-\delta^{2q-p}\right)\left(1-\delta^{2(q+1)-p}\right)}{\left(1-\delta^2\right)\left(1-\delta^4\right)\ldots\left(1-\delta^{2q}\right)\left(1-\delta^{2(q+1)}\right)}\left(\delta^m h\right)^p+O\left(h^{k+1}\right)\tag{3.11}$$

Consider the term in Eq. (3.11) corresponding to $p=2(q+1)$ in the summation. This term is zero due to

$$1-\delta^{2(q+1)-p}=0\tag{3.12}$$

when $p = 2(q+1)$. This occurs when the $O\left(h^{2(q+1)}\right)$ term is eliminated. For example, when the $O\left(h^4\right)$ term was eliminated from Eq. (3.8) this occurred for $p = 2(q+1)$ when $q = 1$. The error term, therefore, is the term in the summation in Eq. (3.11) that corresponds to $p = 2(q+1)+1$. However, this means *p* is an odd integer and due to the coefficient i^p this means the term is imaginary. Consequently, this imaginary term, as all imaginary terms in Eq. (3.11) can be eliminated by taking the real part of Eq. (3.11). This means that the error term corresponds to the first even integer value of *p* greater than $2(q+1)$, which is $p = 2(q+2)$. The error term in Eq. (3.11) is, therefore, with $p = 2(q+2)$

$$E_{m,2(q+2)} = a_{2(q+2)} i^{2(q+2)} \frac{\left(1-\delta^{-2(q+1)}\right)\left(1-\delta^{-2q}\right)\cdots\left(1-\delta^{-4}\right)\left(1-\delta^{-2}\right)}{\left(1-\delta^{2}\right)\left(1-\delta^{4}\right)\cdots\left(1-\delta^{2q}\right)\left(1-\delta^{2(q+1)}\right)}\left(\delta^m h\right)^{2(q+2)} \tag{3.13}$$

or

$$E_{m,2(q+2)} = a_{2(q+2)} i^{2(q+2)} \frac{\left(\delta^{2(q+1)}-1\right)\left(\delta^{2q}-1\right)\cdots\left(\delta^{4}-1\right)\left(\delta^{2}-1\right)\delta^{-2-4-\cdots-2q-2(q+1)}}{\left(1-\delta^{2}\right)\left(1-\delta^{4}\right)\cdots\left(1-\delta^{2q}\right)\left(1-\delta^{2(q+1)}\right)}\left(\delta^m h\right)^{2(q+2)} \tag{3.14}$$

or

$$E_{m,2(q+2)} = a_{2(q+2)} i^{2(q+2)} (-1)^{q+1} \delta^{-2(1+2+\cdots+q+(q+1))} \left(\delta^m h\right)^{2(q+2)} \tag{3.15}$$

Using what Henrici [14] calls a “well known formula”

$$1+2+\cdots+\ell = \frac{\ell(\ell+1)}{2} \tag{3.16}$$

Eq. (3.15) can be written

$$E_{m,2(q+2)} = a_{2(q+2)} i^{2(q+2)} (-1)^{q+1} \delta^{-(q+1)(q+2)} \left(\delta^m h\right)^{2(q+2)} \tag{3.17}$$

Because $i = \sqrt{-1}$ one has

$$i^{2(q+2)} (-1)^{q+1} = (-1)^{\frac{1}{2}2(q+2)+(q+1)} = (-1)^{2(q+1)+1} \tag{3.18}$$

The exponent of (-1), i.e. $2(q+1)+1$, is always an odd integer, therefore, the expression above is always (-1). Equation (3.17) then becomes

$$E_{m,2(q+2)} = -a_{2(q+2)} \delta^{-(q+1)(q+2)} \left(\delta^m h\right)^{2(q+2)} \tag{3.19}$$

Using Eqs. (3.19) and (3.13), Eq. (3.11) becomes

$$A_{m,q+1} = a_0 - a_{2(q+2)} \delta^{-(q+1)(q+2)} \left(\delta^m h\right)^{2(q+2)} \tag{3.20}$$

The quantity a_0 from Eq. (3.20) is, therefore

$$a_0 = \mathrm{Real}\left(A_{m,q+1}\right) + a_{2(q+2)}\delta^{-(q+1)(q+2)}\left(\delta^m h\right)^{2(q+2)} \tag{3.21}$$

where $A_{m,q+1}$ is determined recursively from Eqs. (3.5) and (3.6) with $m = 0,1,2,\ldots,$ and $q = 0,1,2,\ldots,m-1$. The index *m* can be zero in Eq. (3.5) and $m > q$ in Eq. (3.6). The error of a_0 is then of $O\left[\delta^{-(q+1)(q+2)}\left(\delta^m h\right)^{2(q+2)}\right]$.

3.2 Numerical Results

Numerical results for classical Richardson extrapolation will be presented first in order to demonstrate the improvement in accuracy that is obtained by using Richardson extrapolation based on complex variables. Quarteroni, Sacco, and Saleri [12] present the error in classical Richardson extrapolation as

$$A_{m,n} = a_0 + 0\left(\left(\delta^m h\right)^{n+1}\right) \tag{3.22}$$

with $m = 0,1,\ldots,n$. However, Henrici [14] presents $A_{m,n}$ as

$$A_{m,n} = a_0 + \left(-1\right)^n a_{n+1}\,\delta^{-\frac{n(n+1)}{2}}\left(\delta^m h\right)^{n+1}\left[1 + 0\left(\delta^m h\right)\right] \tag{3.23}$$

Note the similarity between Eq. (3.23) and Eq. (3.20).The example selected in [12] to demonstrate the accuracy of classical Richardson extrapolation is the approximation of the derivative of the function given by Eq. (2.31) at *x* = 0. In this case

$$A\left(\delta^m h\right) = \frac{f\left(x + \delta^m h\right) - f\left(x\right)}{\delta^m h} \tag{3.24}$$

and $\delta = 0.5$ and $h = 0.1$ as used in [12]. A table is presented in [12] (Table 9.9) of absolute errors $\left|A_{m,n} - a_0\right|$ for $m = 0,1,\ldots,5$ and $n = 0,1,\ldots,5$. Classical Richardson extrapolation is computed here also using Eqs. (2.31) and (3.24) with $\delta = 0.5$ and $h = 0.1$ and the results are given in Table 1. Table 1 is essentially the same as Table 9.9 in [12] except the error in Table 1 is defined as

$$\hat{E}_{m,n} = \log_{10}\left|A_{m,n} - a_0\right| \tag{3.25}$$

Numerical results using Richardson extrapolation based on complex variables as given by Eq. (3.21) are presented in Table 2 for the same example and conditions as used in Table 1 with

$$A\left(i\delta^m h\right)=\frac{f\left(x+i\delta^m h\right)-f\left(x\right)}{i\delta^m h} \tag{3.26}$$

The error in Table 2 is defined as

$$\hat{E}_{m,n}=\log_{10}\left|\mathrm{Real}\left(A_{m,n}\right)-a_0\right| \tag{3.27}$$

From Eq. (3.21) with $n = q + 1$

$$a_0=\mathrm{Real}\left(A_{m,n}\right)+a_{2(n+1)}\delta^{-n(n+1)}\left(\delta^m h\right)^{2(n+1)} \tag{3.28}$$

In this case the error improves two orders of magnitude with each unit increase in *n* in Eq. (3.28), whereas it only improves one order of magnitude with each unit increase in *n* in Eq. (3.23). Machine accuracy (64 bit arithmetic) is reached rather quickly in Table 2 whereas machine accuracy is never achieved in Table 1 for the same number of recursions.

Derivative of $f\left(x\right)=xe^{-x}\cos\left(2x\right)$ at x=0

$\hat{E}_{m,n}=\log_{10}\left|A_{m,n}-a_0\right|$

m	$\hat{E}_{m,0}$	$\hat{E}_{m,1}$	$\hat{E}_{m,2}$	$\hat{E}_{m,3}$	$\hat{E}_{m,4}$	$\hat{E}_{m,5}$
0	-9.46E-01					
1	-1.27E+00	-2.21E+00				
2	-1.59E+00	-2.77E+00	-3.65E+00			
3	-1.90E+00	-3.35E+00	-4.55E+00	-6.26E+00		
4	-2.20E+00	-3.94E+00	-5.45E+00	-7.50E+00	-8.52E+00	
5	-2.50E+00	-4.54E+00	-6.35E+00	-8.73E+00	-1.00E+01	-1.13E+01

Table 1. Classical Richardson Extrapolation

Derivative of $f\left(x\right)=xe^{-x}\cos\left(2x\right)$ at x=0

$\hat{E}_{m,n}=\log_{10}\left|\mathrm{Real}\left(A_{m,n}\right)-a_0\right|$

m	$\hat{E}_{m,0}$	$\hat{E}_{m,1}$	$\hat{E}_{m,2}$	$\hat{E}_{m,3}$	$\hat{E}_{m,4}$	$\hat{E}_{m,5}$
0	-1.82E+00					
1	-2.43E+00	-5.13E+00				
2	-3.03E+00	-6.34E+00	-8.59E+00			
3	-3.63E+00	-7.55E+00	-1.04E+01	-1.35E+01		
4	-4.23E+00	-8.75E+00	-1.22E+01	-1.57E+01	-1.57E+01	
5	-4.83E+00	-9.95E+00	-1.40E+01	-1.57E+01	-1.57E+01	-1.57E+01

Table 2. Richardson Extrapolation Based On Complex Variables

4.0 Calculation of Logarithms by Differentiation

Another interesting application of Richardson extrapolation is described in Section 12.5 of Henrici [14] having to do with the calculation of logarithms by differentiation. Consider the function

$$f(x) = a^x = e^{x\ln(a)} \tag{4.1}$$

The derivative of $f(x)$ leads to

$$f'(x) = f(x)\ln(a) \tag{4.2}$$

For x=0 one has

$$\ln(a) = f'(0) \tag{4.3}$$

Henrici [14] uses the one-sided approximation for the derivative of $f'(0)$ to obtain

$$S(h) = \frac{f(h) - f(0)}{h} = \frac{a^h - 1}{h} \tag{4.4}$$

then, by definition one has

$$\ln(a) = \lim_{h \to 0} S(h) \tag{4.5}$$

Henrici [14] uses $h = 2^{-n}$ where n=0,1,2,... and goes on to point out that the method is numerically unstable. To overcome this instability Henrici [14] ingeniously develops a sequence that is stable. He points out that although the sequence is stable, it converges intolerably slowly. To overcome this difficulty he shows that the sequence satisfies the conditions for Richardson extrapolation and then uses Richardson extrapolation to accelerate convergence.

Henrici [14] considered as an example the calculation of ln(6). His method was coded and the results are given in Table 3. Table 3 is analogous to Table 12.5 in [14] except in [14] the objective was to compute ln(6) to seven significant digits whereas in Table 3 the objective was to compute ln(6) to machine accuracy in 64 bit arithmetic. Seven significant digits are obtained in Table 3 (and Table 12.5 [14]) at A(5,5). The $A_{m,n}$ used in Table 3 is Henrici's series as developed in [14].

Classical Richardson extrapolation was also investigated. In this case $A(\delta^m h) = S(\delta^m h)$ where *S(h)* is defined by Eq. (4.4). The results of this numerical experiment are given in Table 4. This series begins to converge faster than Henrici's (see Table 3) but as Henrici [14] noted, there is a stability problem with this approach as indicated in Table 4 by the fact that after a period of time the error becomes worse.

$$\hat{E}_{m,n} = \log_{10}\left|A_{m,n} - a_0\right| \text{ where } a_0 = \ln(6)$$

m	$\hat{E}_{m,0}$	$\hat{E}_{m,1}$	$\hat{E}_{m,2}$	$\hat{E}_{m,3}$	$\hat{E}_{m,4}$	$\hat{E}_{m,5}$
0	5.06E-01					
1	4.42E-02	-2.70E-03				
2	-3.29E-01	-7.69E-01	-9.81E-01			
3	-6.65E-01	-1.45E+00	-2.03E+00	-2.36E+00		
4	-9.82E-01	-2.09E+00	-3.01E+00	-3.70E+00	-4.12E+00	
5	-1.29E+00	-2.71E+00	-3.95E+00	-4.97E+00	-5.74E+00	-6.24E+00
6	-1.60E+00	-3.32E+00	-4.87E+00	-6.20E+00	-7.30E+00	-8.15E+00
7	-1.90E+00	-3.93E+00	-5.78E+00	-7.42E+00	-8.84E+00	-1.00E+01
8	2.20E+00	-4.53E+00	-6.68E+00	-8.63E+00	-1.04E+01	-1.18E+01
9	-2.50E+00	-5.13E+00	-7.59E+00	-9.84E+00	-1.19E+01	-1.36E+01
10	-2.80E+00	-5.74E+00	-8.49E+00	-1.10E+01	-1.34E+01	-1.54E+01
11	-3.11E+00	-6.34E+00	-9.40E+00	-1.23E+01	-1.47E+01	-1.51E+01
12	-3.41E+00	-6.94E+00	-1.03E+01	-1.35E+01	-1.52E+01	-1.52E+01
13	-3.71E+00	-7.54E+00	-1.12E+01	-1.45E+01	-1.52E+01	-1.52E+01
14	-4.01E+00	-8.15E+00	-1.21E+01	-1.52E+01	-1.51E+01	-1.51E+01
15	-4.31E+00	-8.75E+00	-1.30E+01	-1.57E+01	-1.57E+01	-1.57E+01
16	-4.61E+00	-9.35E+00	-1.39E+01	-1.50E+01	-1.50E+01	-1.50E+01
17	-4.91E+00	-9.95E+00	-1.47E+01	-1.52E+01	-1.52E+01	-1.52E+01
18	-5.21E+00	-1.06E+01	-1.49E+01	-1.50E+01	-1.50E+01	-1.50E+01
19	-5.51E+00	-1.12E+01	-1.49E+01	-1.49E+01	-1.48E+01	-1.48E+01
20	-5.82E+00	-1.18E+01	-1.52E+01	-1.52E+01	-1.54E+01	-1.54E+01

Table 3. Calculation of ln(6) Using Henrici's [14] Series

$$\hat{E}_{m,n} = \log_{10}\left|A_{m,n} - a_0\right| \text{ where } a_0 = \ln(6)$$

m	$\hat{E}_{m,0}$	$\hat{E}_{m,1}$	$\hat{E}_{m,2}$	$\hat{E}_{m,3}$	$\hat{E}_{m,4}$	$\hat{E}_{m,5}$
0	-7.68E-01					
1	-1.08E+00	-2.29E+00				
2	-1.39E+00	-2.91E+00	-4.24E+00			
3	-1.69E+00	-3.52E+00	-5.16E+00	-6.59E+00		
4	-2.00E+00	-4.12E+00	-6.07E+00	-7.81E+00	-9.33E+00	
5	-2.30E+00	-4.73E+00	-6.98E+00	-9.02E+00	-1.08E+01	-1.25E+01
6	-2.60E+00	-5.33E+00	-7.88E+00	-1.02E+01	-1.28E+01	-1.25E+01
7	-2.90E+00	-5.93E+00	-8.78E+00	-1.14E+01	-1.28E+01	-1.28E+01
8	-3.20E+00	-6.53E+00	-9.69E+00	-1.24E+01	-1.22E+01	-1.22E+01
9	-3.50E+00	-7.14E+00	-1.06E+01	-1.20E+01	-1.20E+01	-1.20E+01
10	-3.80E+00	-7.74E+00	-1.14E+01	-1.26E+01	-1.27E+01	-1.27E+01
11	-4.11E+00	-8.34E+00	-1.23E+01	-1.30E+01	-1.31E+01	-1.31E+01
12	-4.41E+00	-8.94E+00	-1.15E+01	-1.14E+01	-1.14E+01	-1.14E+01
13	-4.71E+00	-9.56E+00	-1.09E+01	-1.09E+01	-1.09E+01	-1.09E+01
14	-5.01E+00	-1.02E+01	-1.16E+01	-1.19E+01	1.25E+01	-1.32E+01
15	-5.31E+00	-1.04E+01	-1.01E+01	-1.01E+01	-1.00E+01	-1.00E+01
16	-5.61E+00	-1.05E+01	-1.03E+01	-1.01E+01	-1.01E+01	-1.00E+01
17	-5.91E+00	-1.05E+01	-1.05E+01	-1.06E+01	-1.06E+01	-1.07E+01
18	-6.21E+00	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
19	-6.51E+00	-8.95E+00	-8.82E+00	-8.76E+00	-8.73E+00	-8.72E+00
20	-6.82E+00	-9.21E+00	-8.92E+00	-8.80E+00	-8.74E+00	-8.72E+00

Table 4. Calculation of ln(6) Using Classical Richardson Extrapolation

Richardson extrapolation based on complex variables can also be applied to this problem and convergence is much faster. Analogous to Eq. (4.4) $A\left(i\delta^m h\right)$ is defined as

$$A\left(i\delta^m h\right) = \frac{a^{i\delta^m h} - 1}{i\delta^m h} \tag{4.6}$$

Results using this approach to calculate ln(6) are given in Table 5. Notice in Table 5 that convergence is much faster than in either Table 3 or 4 and no stability problem is indicated. Convergence to seven significant digits is achieved in Table 5 at A(2,1), and machine zero is reached at A(5,3).

$$\hat{E}_{m,n} = \log_{10}\left|A_{m,n} - a_0\right| \text{ where } a_0 = \ln(6)$$

m	$\hat{E}_{m,0}$	$\hat{E}_{m,1}$	$\hat{E}_{m,2}$	$\hat{E}_{m,3}$	$\hat{E}_{m,4}$	$\hat{E}_{m,5}$
0	-2.02E+00					
1	-2.62E+00	-5.42E+00				
2	-3.22E+00	-6.62E+00	-9.74E+00			
3	-3.82E+00	-7.82E+00	-1.15E+01	-1.49E+01		
4	-4.43E+00	-9.03E+00	-1.34E+01	-1.54E+01	-1.54E+01	
5	-5.03E+00	-1.02E+01	-1.51E+01	-1.57E+01	-1.57E+01	-1.57E+01
6	-5.63E+00	-1.14E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
7	-6.23E+00	-1.26E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
8	-6.83E+00	-1.39E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
9	-7.44E+00	-1.52E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
10	-8.04E+00	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
11	-8.64E+00	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
12	-9.24E+00	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
13	-9.85E+00	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
14	-1.04E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
15	-1.10E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
16	-1.17E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
17	-1.23E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
18	-1.29E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
19	-1.35E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01
20	-1.41E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01	-1.57E+01

Table 5. Calculation of $\ln(6)$ Using Richardson Extrapolation Based on Complex Variables

The reason Henrici [14] was driven to the development of a stable sequence, and the source of the stability problems encountered by classical Richardson extrapolation as demonstrated in Table 4, is the use of the one-sided approximation for the derivative of $f(x)$ as defined by Eq. (4.4). It does not appear that the use of the classical second-order central difference formula for the derivative of $f(x)$ (see Eq. (2.21), for example) would have made any improvement. However, if one uses the second-order approximation based on complex variables as given by Eq. (2.2) for the first derivative of $f(x)$, then all of these problems go away. Based on Eqs. (2.2) and (4.1)-(4.3) one has

$$\ln(a) = f'(0) = \lim_{h \to 0} \frac{\operatorname{Im}\left(a^{ih}\right)}{h} \tag{4.7}$$

Equation (4.7) means that ln(a) can be determined directly with one computation and without recourse to series, recursion formulas, or interpolations. To illustrate the power of Eq. (4.7) consider Table 6 where the natural logarithm of a=6 is computed using several different approximations for the derivative of $f(x)$. Table 6 shows that the choice of h is very important for all methods except Eq. (4.7). Probably the best thing to do in practice when using Eq. (4.7) is to select h to be approximately equal to the square root of machine zero. The exact value used for h does not seem to be critical. Actually, experience in using Eq. (4.7) demonstrates that $h = 10^{-8}$ works well for either 32 bit or 64 bit arithmetic. Moreover, additional errors do not seem to be introduced if an $h = 10^{-16}$ is used for either 32 bit or 64 bit arithmetic. The calculation of ln(a) for numerous values of a (for example, $10^{-36} \le a \le 10^{99}$) indicates that $h = 10^{-12}$ is a reasonable value to use.

$$\hat{E}_{m,n} = \log_{10} \left| \text{Eq. (No.)} - a_0 \right| \text{ where } a_0 = \ln(6)$$

m	$\log_{10}(h)$	Eq.(2.2)	Eq.(2.3)	Eq.(2.21)	Eq.(2.25)
0	0.0E+00	-8.83E-02	5.06E-01	5.11E-02	-8.11E-01
1	-1.0E+00	-2.02E+00	-7.68E-01	-2.02E+00	-4.81E+00
2	-2.0E+00	-4.02E+00	-1.79E+00	-4.02E+00	-8.81E+00
3	-3.0E+00	-6.02E+00	-2.79E+00	-6.02E+00	-1.28E+01
4	-4.0E+00	-8.02E+00	-3.79E+00	-8.02E+00	-1.26E+01
5	-5.0E+00	-1.00E+01	-4.79E+00	-1.00E+01	-1.20E+01
6	-6.0E+00	-1.20E+01	-5.79E+00	-1.10E+01	-1.13E+01
7	-7.0E+00	-1.40E+01	-6.79E+00	-1.00E+01	-1.03E+01
8	-8.0E+00	-1.57E+01	-7.57E+00	-8.31E+00	-8.61E+00
9	-9.0E+00	-1.57E+01	-7.21E+00	-8.21E+00	-8.51E+00
10	-1.0E+01	-1.57E+01	-6.22E+00	-7.31E+00	-7.61E+00
11	-1.1E+01	-1.57E+01	-5.14E+00	-5.77E+00	-6.07E+00
12	-1.2E+01	-1.57E+01	-4.09E+00	-4.58E+00	-4.89E+00
13	-1.3E+01	-1.57E+01	-3.85E+00	-3.85E+00	-4.15E+00
14	-1.4E+01	-1.57E+01	-2.17E+00	-2.90E+00	-3.20E+00
15	-1.5E+01	-1.57E+01	-1.81E+00	-1.81E+00	-2.11E+00
16	-1.6E+01	-1.57E+01	-3.68E-01	-3.68E-01	-6.69E-01

Table 6. Calculation of ln(6) Using Different Differential Formulas

There has been some difficulty in determining exactly how modern calculators and computers calculate logarithms. It could be that Eq. (4.7) might have merit in this regard. Equation (4.7) does require taking a number to a power and the operation count for this may negate its usefulness in calculators and computers. Equation (4.7), however, does not require shifting or partitioning in the process of computing a logarithm, and it is exceedingly accurate.

5.0 Summary

Analytical and numerical results were presented on the use of complex variables to develop high-order relations for numerical derivatives, matrix-vector products, and Richardson extrapolation. Many of these results are new, although they have been and are currently being used by personnel within the SimCenter: National Center for Computational Engineering. There seem to be numerous useful applications of these results. For example, if calculators and computers do indeed use series, iterations, recursions, shifting, and/or partitioning to calculate logarithms, then the results presented in Section 4.0 dealing with the calculation of logarithms using differentiation could be of interest. In any case, it is intended that this material will serve as a useful reference and teaching aid to those that might find additional applications.

References

1. Squire, W. and Trapp, G., "Using Complex Variables to Estimate Derivatives of Real Functions," *SIAM Review*, Vol. 40, No. 1, March 1998, pp. 110-112.
2. Lyness, J.N. and Moler, C.B., "Numerical Differentiation of Analytic Functions," *SlAM Journal of Numerical Analysis,* Vol. 4, 1967, pp. 202-210.
3. Lyness, J.N., "Numerical Algorithms Based on the Theory of Complex Variables," *Proceedings ACM 22 & National Conference*, Thompson Book Company, Washington, D.C., 1967, pp. 124-134.
4. Whitfield, D.L., and Taylor, L.K., "Variants of a Two-Level Method for the Approximate Numerical Solution of Field Simulation Equations," *MSSU-EIRS-ERC-98-09*, July 1998.
5. Sreenivas, K., Private Communication, Mississippi State University, April 1998.
6. Pankajakshan, R., Private Communication, Mississippi State University, April 1998.
7. Newman, J.C., Anderson, W.K., and Whitfield, D.L., "Multidisciplinary Sensitivity Derivatives Using Complex Variables," *MSSU-COE-ERC-98-08*, July 1998.
8. Fulks, W., *Advanced Calculus, Second Edition*, John Wiley & Sons, Inc., New York, 1969.
9. Dennis, Jr., J.E., and Schnabel, R.B., *Numerical Methods for Unconstrained Optimization ad Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.
10. Kelly, C.T., *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1995.
11. Ortega, J.M. and Rheinboldt, W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, Inc., 1970.
12. Quarteroni, A., Sacco, R., and Saleri, F., *Numerical Mathematics*, Springer Verlag, 2000.
13. Richardson, L.F. and Gaunt, J.A., "The Deferred Approach to the Limit," *Philosophical Transactions of the Royal Society of London*, Vol. 226A, 1927, pp 299-361.
14. Henrici, P., *Elements of Numerical Analysis*, John Wiley & Sons, Inc., 1964, Third Printing, March 1967.