JOURNAL OF COMPUTATIONAL PHYSICS 99, 56–67 (1992)

# On the Errors Incurred Calculating Derivatives Using Chebyshev Polynomials

KENNETH S. BREUER* AND RICHARD M. EVERSON

*Center for Fluid Mechanics, Turbulence and Computation, Brown University, Providence, Rhode Island 02912*

Received October 10, 1989; revised December 21, 1990

The severe errors associated with the computation of derivatives of functions approximated by Chebyshev polynomials are investigated. When using standard Chebyshev transform methods, it is found that the maximum error in the computed first derivative grows as $N^2$, where $N+1$ is the number of Chebyshev polynomials used to approximate the function. The source of the error is found to be magnification of roundoff error by the recursion equation, which links coefficients of a function to those of its derivative. Tight coupling between coefficients enables propagation of errors from high-frequency to low-frequency modes. Matrix multiplication techniques exhibit errors of the same order of magnitude. However, standard methods for computing the matrix elements are shown to be ill-conditioned and to magnify the differentiation errors by an additional factor of $N^2$. For both the transform and matrix methods, the errors are found to be most severe near the boundaries of the domain, where they grow as $(1-x^2)^{-1/2}$ as $x$ approaches $\pm 1$. Comparisons are made with the errors associated with derivatives of functions approximated by Fourier series, in which case it is reported that the errors only grow linearly with $N$ and are evenly distributed throughout the domain. A method for reducing the error is discussed. © 1992 Academic Press, Inc.

## 1. INTRODUCTION

Recent years have seen widespread use of pseudo-spectral methods for the solution of partial differential equations. The pseudo-spectral technique represents a function by a generalized Fourier expansion. Derivatives are obtained by operations in transform space, while nonlinear terms are calculated in physical space. Basis functions are sines and cosines for problems with periodic boundary conditions, while Chebyshev polynomials are frequently used for non-periodic problems.

The principal advantage of spectral methods is their promise of "spectral accuracy" [1]; that is, if the function being represented is infinitely smooth, then the $k$th coefficient of the expansion decays faster than any inverse power of $k$. Consequently very good approximations to the function may be obtained with relatively few terms.

Derivatives, too, are obtained with spectral accuracy if infinite precision arithmetic is used. In this paper we examine some difficulties that arise in computing derivatives using Chebyshev polynomials on a finite precision computer.

A Chebyshev polynomial of degree $k$ on $[-1, 1]$ is defined by

$$T_k(x) = \cos k\xi, \qquad \xi = \arccos x. \tag{1}$$

The polynomials obey the recurrence relation

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \qquad \text{for} \quad k \geqslant 2. \tag{2}$$

Polynomials are orthogonal with weight $w(x) = (1-x^2)^{-1/2}$,

$$\int_{-1}^{1} T_j(x)\, T_k(x)\, w(x)\, dx = \frac{\pi}{2} c_k\, \delta_{jk}, \tag{3}$$

where $c_0 = 2$ and $c_k = 1$ for $k > 0$.

The Chebyshev expansion of a function $u(x)$, $x \in [-1, 1]$ is

$$u(x) = \sum_{k=0}^{\infty} a_k T_k(x), \tag{4}$$

where the coefficients are given by the inner product

$$a_k = \int_{-1}^{1} u(x)\, T_k(x)\, w(x)\, dx. \tag{5}$$

In practice the expansion is approximated by the first $N+1$ terms and the approximation is required to equal $u$ at $N+1$ collocation points. We shall concentrate on the popular Gauss–Lobatto points:

$$x_j = \cos \frac{\pi j}{N}, \qquad 0 \leqslant j \leqslant N. \tag{6}$$

* Current address: Department of Aeronautics and Astronautics, Room 33-214, Massachusetts Institute of Technology, Cambridge, MA 02139.

0021-9991/92 $3.00

These collocation points have the particular advantage of allowing the integrals yielding the Chebyshev coefficients $a_k$ (Eq. (5)) to be evaluated in $O(N \log N)$ operations by a fast Fourier transform (FFT).

Term by term differentiation of Eq. (4) gives a formal representation of the derivative of $u$,

$$u'(x) = \sum_{k=0}^{\infty} b_k T_k(x), \tag{7}$$

where

$$b_k = \frac{2}{c_k} \sum_{\substack{p=k+1 \\ p+k \ \text{odd}}}^{\infty} p a_p. \tag{8}$$

Manipulations with trigonometric identities yield the following central recurrence relation in Chebyshev space:

$$c_k b_k = b_{k+2} + 2(k+1)\, a_{k+1}. \tag{9}$$

Since, when differentiating a polynomial of degree $N$, $b_k = 0$ for $k \geqslant N$ the non-zero coefficients, $b_k$, are computed for decreasing $k$ in $2N$ multiplications or additions. Applying the recurrence twice yields the coefficients, $d_k$, for a Chebyshev expansion of the second derivative of $u$.

Alternative procedures for calculating derivatives, which avoid transformations to and from Chebyshev space, require instead a matrix multiplication. These "collocation matrix" methods are asymptotically less efficient than the Chebyshev transform method (needing $O(N^2)$ operations), but are common for small problems. We discuss these methods in Section 4.

The largest current codes (see, for example, [2]) use 200 to 500 Chebyshev modes and this number is expected to rise as more complex problems are tackled. In this paper we examine the character and sources of errors incurred when calculating derivatives of functions represented by a Chebyshev polynomial expansion on a machine with finite precision. Briefly, we find that the error in the first derivative grows like $N^2$ for (even moderately) large $N$.

We draw attention to two related works. Trefethen and Trummer [3] discuss the behavior of the eigenvalues of collocation matrices for Fourier, Legendre, and Chebyshev polynomials. Their apparent growth like $N^2$ (when calculated with finite precision arithmetic) accounts for anomalous timestep restrictions in numerical solutions of boundary value problems. Greengard [4] also noticed the problem examined in this paper and discusses Chebyshev spectral methods applied to integral equations.

The remainder of the paper is organized as follows. In the following section we give details of the manifestation of errors in the computed derivatives. In Section 3, w⌒ ısolate the sources of the error to roundoff error and strong coupling between Chebyshev modes. Comparison with conventional Fourier series helps elucidate the problem, and a technique for alleviation of the error is discussed. Differentiation by matrix methods is examined in Section 4. Here, similar problems exist, but a careless computation of the matrix elements is shown to lead to $O(N^4)$ errors in the first derivative. Finally, in Section 5, we make some general concluding remarks.

## 2. APPEARANCE OF ERRORS IN COMPUTED DERIVATIVES

We estimate the errors incurred in calculating by the Chebyshev polynomial approximation by comparing numerically calculated derivatives with the known derivatives of an example function, $u(x)$. Two approximations are being made here; one in truncating the expansion (Eq. (4)) and another by making an imperfect numerical calculation, i.e., one affected by roundoff error. We shall only be concerned with the fidelity of a numerical calculation, always retaining sufficient terms in the expansion to adequately represent $u(x)$. Let $\hat{f}$ denote the numerical approximation to $f$. Two measures are used to characterize the error in a numerical approximation, $\hat{u}(x)$, to $u(x)$: the maximum or $L_\infty$-error,

$$E_\infty = \max_{-1 \leqslant x \leqslant 1} |\hat{u}(x) - u(x)|, \tag{10}$$

and the root mean square or $L_2$-error,

$$E_2 = \left[ \int_{-1}^{1} (\hat{u}(x) - u(x))^2 \, dx \right]^{1/2}. \tag{11}$$

We focus first on the recursion technique. Chebyshev coefficients, $a_k$, of the example function were found by FFT, from which the coefficients, $b_k$ and $d_k$, for Chebyshev expansions of first and second derivatives were computed by successive applications of the recursion equation (9). Inverse FFTs recover the approximations to $u'(x)$ and $u''(x)$. Figure 1 shows the growth of the maximum and root-mean-square errors, for both the first and second derivatives, as a function of the number of collocation points $N$. The pair of lines at the bottom of the graph indicate the $L_\infty$ and $L_2$ errors accrued by performing only the forward and backward transforms without any intervening derivative calculations. The computations here (and in all subsequent results unless explicitly noted) were performed using IEEE 64-bit floating point arithmetic on a Silicon Graphics Power Series workstation. The example function used here, and throughout this paper, is given by

$$u(x) = \frac{\sin 8(x+1)}{(x+1.1)^{3/2}}. \tag{12}$$

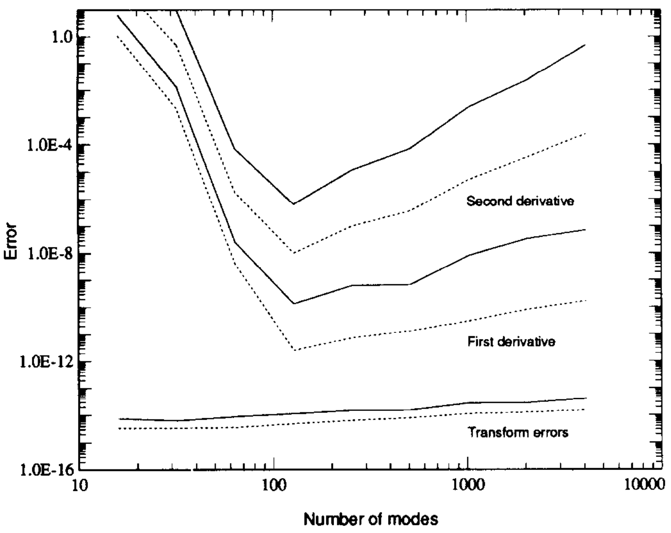

FIG. 1. Maximum (solid line) and rms (dotted lines) errors for the first and second derivatives of the example function $u(x) = \sin(8(x+1))/(x+1.1)^{3/2}$. The derivatives are calculated using standard transform techniques. The lower two lines show the maximum and rms errors of the Chebyshev transform alone.

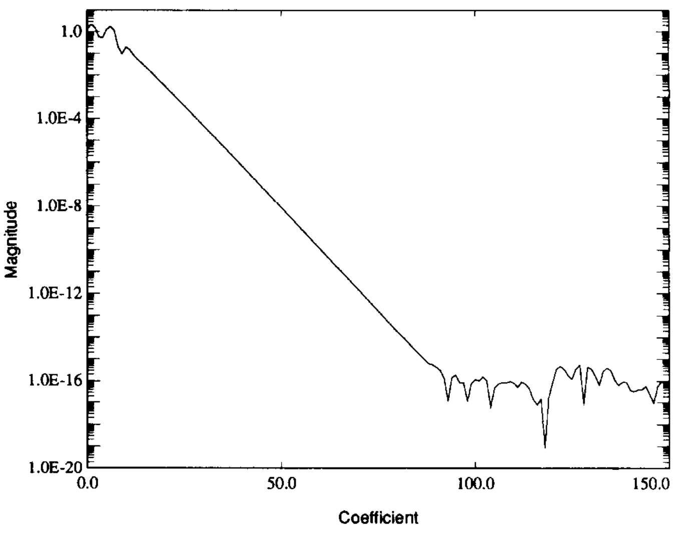

FIG. 2. Spectrum of the Chebyshev modes for the example function $u(x) = \sin(8(x+1))/(x+1.1)^{3/2}$. The approximately constant magnitude of the $a_k$ for high values of $k$ represents the machine precision (found to be independent of $N$).

When $N$ is small there are insufficient Chebyshev modes to adequately resolve $u$, but the more serious problem and the subject of this paper, is the quadratic growth of the $L_\infty$-error in the first derivative for large $N$. The $L_2$-error grows rather more slowly, but still faster than $N$. The severity of the problem is highlighted by recognizing that the maximum error in the second derivative is of the same order as the function itself when $N = 4096$. The minimum in each curve in Fig. 1 is thus determined by competition between increasing resolution of $u$ and growing errors resulting from the calculation of the Chebyshev coefficients of $u'$ and $u''$. Clearly, these errors render the recursion technique useless as $N$ becomes large.

The computed spectrum of Chebyshev coefficients for the example function is shown in Fig. 2. The decay of the coefficients, $a_k$, is geometric: $a_k \propto \exp(-\gamma N)$ (with $\gamma$ dictated by the singularity at $x = -1.1$). The function is fully resolved to machine precision by about 100 Chebyshev modes. As we shall see, it is the presence of this roundoff "plateau" in the $a_k$ due to the finite precision of the computer which provides the seed for the errors observed in Fig. 1. The magnitude of error due to roundoff effects introduced by the forward Chebyshev transform was estimated by examining the magnitude of the coefficients at the high-frequency end of the $a_k$ spectrum. As indicated by Fig. 2, this is approximately $10^{-16}$ and was found to be independent of $N$.

We must emphasize that these results are essentially independent of the complexity of the example function used. This is demonstrated in Fig. 3 which shows the $L_\infty$ and $L_2$ errors for the first derivative of both the original function, $u(x)$ (Eq. (12), and a more complicated function,

$$v(x) = \frac{\sin 8(x+1) + \sin 400(x+1)}{(x+1.1)^{3/2}}. \tag{13}$$

For comparison purposes, the errors in $u'$ and $v'$ have been normalized by their respective root-mean-square amplitudes. Because $v(x)$ requires more modes for complete

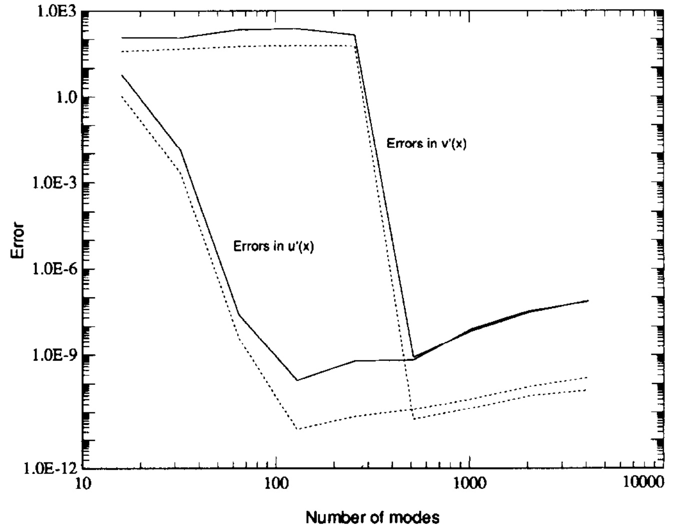

FIG. 3. Maximum (solid line) and rms (dotted line) errors for the first derivative for the original example function, $u(x)$, and a more complicated function, $v(x)$ (see text for full definitions of $u$ and $v$). The derivatives are calculated using transform methods and the errors are normalized by the rms value of the corresponding function.

resolution, the point at which the truncation error falls below the "derivative error" occurs at a higher value of $N$ than for the original example function, $u(x)$. However, once the number of modes is sufficient, the two error curves are very close. One should note that, since both example functions possess the same singularity, it is not surprising that the error curves are asymptotically the same. Changing the character of the singularity in one of the two example functions separates the two curves. In fact, either moving the location of the singularity closer to $x = -1$, or increasing its strength, makes the errors worse. Thus, the growth of the derivative error defines a lower bound on the accuracy of the computed derivative, independent of the number of modes required to resolve the function. In the remainder of the discussion, we shall concentrate solely on the simpler function (Eq. (12)) so that we may compare results over a wide range of $N$.

## 3. SOURCES OF ERROR

Having determined the nature of the errors in computing derivatives, we now address the issue of their source. For this purpose, we shall mainly restrict our discussion to the errors observed in computing the first derivative.

Evaluation of each derivative requires two FFTs: one to obtain the Chebyshev coefficients $a_k$ of $u(x)$ and another to recover the derivative from the coefficients $b_k$. The errors associated with the transform operations alone were assessed by reconstructing $\hat{u}(x)$ in physical space from the numerically calculated $a_k$. That is, we calculate coefficients $a_k$ from $u(x)$ by Chebyshev transform and immediately perform the inverse transform to obtain $\hat{u}(x)$. The two lower lines in Fig. 1 show that the FFT contributes an error comparable with the machine precision. (Machine precision is the smallest number, $\delta$, such that $1 + \delta$ can be distinguished from 1. For 64-bit floating point arithmetic $\delta \approx 10^{-16}$.) The transform error grows slowly with $N$ indicating that, in addition to the introduction of $O(\delta)$ error by the arithmetic operations of the transforms (which is independent of $N$), the error is also magnified by the number of operations during the transform back to physical space. To quantify this more precisely, the magnitude of the error in the coefficients was artificially raised by adding random noise, uniformly distributed between $\pm 10^{-10}$, to the $a_k$, performing the inverse transform and computing the $L_\infty$ and $L_2$ errors. In this case, it was found that both the $L_\infty$ and $L_2$ errors grew approximately like the square root of $N$. In a subsequent test the noise was distributed between 0 and $10^{-10}$ in which case the $L_\infty$ error was found to grow linearly with $N$. Linear growth was also found for noise introduced such that the random numbers added to adjacent coefficients had opposite signs (note that the recursion relation (9) links every other coefficient not adjacent coefficients). Since the transform error in Fig. 1 is also observed to grow approximately like $N^{1/2}$, we can conclude that the roundoff error in the transforms themselves has random sign. Clearly, the errors apparent in the derivatives arise from the process of numerical differentiation, and not solely from the FFT.

Figure 4 shows the errors associated with the first derivative of the example function for three different machine precisions. Three pairs of curves are plotted, corresponding to 32-, 64-, and 128-bit floating point calculations. (The 128-bit calculations were performed using double precision arithmetic on a Cray-YMP.) As before, both the $L_\infty$- and $L_2$-errors are displayed for each precision. The most striking feature of this graph is the magnitude of the error in single precision. Even for small values of $N$, the error is quite unacceptable, never falling below $10^{-5}$, and rising to $O(1)$ by $N = 2048$. That the different machine precisions result in different levels of error is not an unexpected result, but it does confirm the suspicion that these problems originate from roundoff error incurred during the computation. We have already determined that the error associated with the FFT is not the primary source of the errors, which leads us to the conclusion that the error is generated in the calculation of the coefficients of $u'(x)$.

We can examine the error in the calculated coefficients of the first derivative by comparing them with those obtained directly from the Chebyshev transform of $u'$. We find that the magnitude of the error in the coefficients, $e_k = |\hat{b}_k - b_k|$, is fairly uniform, independent of both $k$ and the magnitude of $b_k$. An odd/even decoupling is also exhibited by the $e_k$, reflecting the structure of the recursion formula (Eq. (9)). The variation of $e_k$ with $N$ is, however, more significant. The

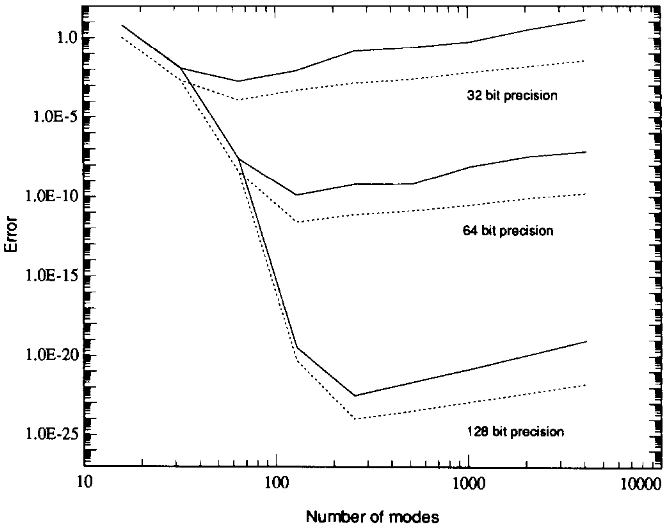

FIG. 4. Maximum (solid line) and rms (dotted line) errors for the first derivative of the example function, $u(x)$ (Eq. (12)) computed using transform techniques. Pairs of lines represent 32-, 64-, and 128-bit precision floating-point calculations.

magnitude of $e_k$ scales linearly with $N$, ranging from $e_0 \approx 2.5 \times 10^{-15}$ for $N=128$ up to $e_0 \approx 6 \times 10^{-13}$ at $N=2048$. These magnitudes are $O(N\delta)$, indicating that the source of the $e_k$ derives from the magnification by the recursion operation of the roundoff error incurred during the Chebyshev transform. This is supported by observing the form of the recursion relation (Eq. (9)) which includes a $ka_k$ term. Thus, if there is any roundoff error present in the $\hat{a}_k$, that error will be multiplied by $k$ during the computation of $\hat{b}_k$. Although it is only the high-frequency coefficients that are multiplied by factors of $O(N)$, the tight coupling of the $b_k$ implied by the recursion relation dictates that any error present in a high-frequency coefficient will be transmitted to all of the lower-frequency coefficients, thus explaining the uniform nature of $|e_k|$.

Having established this, it is quite easy to understand how the overall $O(N^2\delta)$ error that is observed in $u'$ occurs. The transform, from physical to Chebyshev space introduces an $O(\delta)$ roundoff error into each Chebyshev coefficient. The process of calculating $\hat{b}_k$ magnifies this error by $O(N)$, which is communicated to each mode by the tight interdependence that the Chebyshev coefficients exhibit. To make matters worse, the error will *always* be either of uniform or alternating sign due to the odd/even coupling of the recursion relation. This removes the possibility of error cancellation due to random sign, resulting in an additional $O(N)$ error incurred during the backward transform, for an overall $O(N^2\delta)$ error in $u'$.

To be certain that the particular numerical scheme was not the source of the observed problems, several other numerical techniques for calculating $b_k$ from $a_k$ in Chebyshev space were investigated. These alternative schemes included re-writing the recursion equation in various forms so that each of the terms of the equation were of equal magnitude, computing the $b_k$ by direct summation (Eq. (8)), solving the matrix system defined by the recursion equation by a variety of numerical methods (Gaussian elimination, LU and SVD decompositions with forward- and backward-substitution), and re-formulating the matrix system to be diagonally-dominant (see Gottlieb and Orszag, p. 120 of [1]). The results for all of these methods were identical to those for the standard recursion procedure.

### 3.1. *Comparison with Fourier Series*

The conclusion that the coupling between the Chebyshev modes contributes to the large magnitude of the errors observed in $\hat{u}'$ leads us to consider the relationship between the Chebyshev series and the Fourier series. In the latter case, it is well known that if a function is represented by a Fourier series,

$$u(x) = \sum_{k=0}^{N} a_k e^{ikx}, \tag{14}$$

then the derivative of that function is approximated as

$$u'(x) = \sum_{k=0}^{N} b_k e^{ikx}, \tag{15}$$

in which $b_k = ika_k$. Unlike the Chebyshev series, each coefficient is calculated independently and does not depend on the value of any other coefficient. Therefore, errors that accumulate in an individual mode are not transmitted to any other mode. Following the steps outlined above in tracing the growth of errors, we quickly see that if a uniform roundoff error $O(\delta)$ is introduced into $a_k$ by the forward Fourier transform, then that error will be magnified by a factor of $O(N)$ at the highest frequencies, but the factor decreases as $k$ decreases. Due to the independence of the Fourier modes, the error in $\hat{b}_k$ will not be uniform for all $k$ and, since the high-frequency modes generally contain less energy than the low-frequency modes, the larger errors incurred at the high-$k$ will be less detrimental to the overall accuracy of $\hat{u}'$.

In order to compare the Fourier with the Chebyshev series, a calculation was performed in which the derivatives of an example function were computed using a Fourier series approximation. Since periodicity must be enforced, a different example function was used,

$$u(x) = e^{-\sigma(x-x_0)^2}, \qquad x \in [-1, 1], \tag{16}$$

where $x_0$ is a small offset, chosen so that the function is not symmetric about $x=0$, while $\sigma$ is chosen so that the function is sufficiently close to zero at the edges of the boundary. Here, $x_0 = 0.1$ and $\sigma = 10.0$.

The results of this calculation are summarized in Fig. 5. This figure, like Fig. 1, shows both the maximum error, $E_\infty$, and the root-mean-square error, $E_2$, for the first and second derivatives, together with the error associated with the Fourier transform alone. When compared with Fig. 1, it is clear that, while the errors do grow as $N$ increases, they are considerably smaller than those associated with the Chebyshev series. These errors grow slightly faster than linearly with $N$ (rather than quadratically), confirming the above analysis. This calculation was repeated with more complicated functions, and it was found that, as was the case for the Chebyshev errors, this behavior is independent of the example function used.

### 3.2. *Spatial Characteristics*

An alternative approach in analyzing the source of the errors in computing the derivatives of a function approximated by a Chebyshev series is to investigate the spatial distribution of the error. For this, we restrict ourselves to the $N=1024$ case, and only remark that the results presented here apply for all $N$ within the range tested ($N \leqslant 4096$).

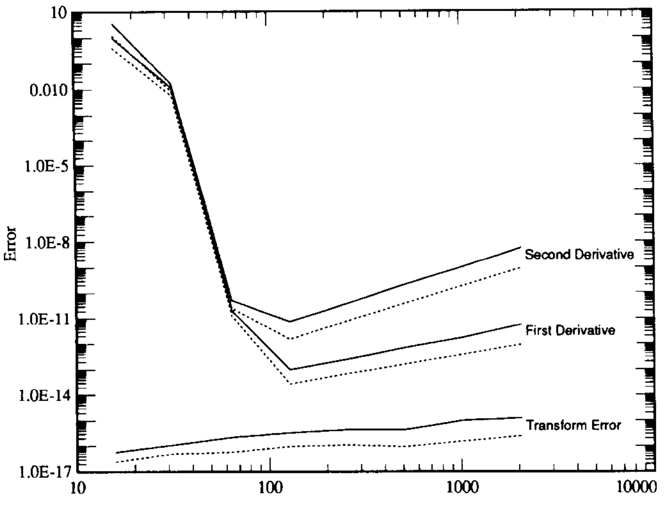

FIG. 5. Maximum (solid line) and rms (dotted line) errors for the first and second derivatives of the example function, $u(x) = \exp(-\sigma(x + x_0)^2)$, approximated using a Fourier series. The derivatives are calculated using transform techniques. The lower two curves represent the errors from the transform operations only.

Figure 6 shows the spatial distribution of the error, defined as

$$E(x) = |u'(x) - \hat{u}'(x)|, \tag{17}$$

plotted versus $(x+1)$ on logarithmic axes so as to allow detailed examination of the behavior near one of the boundaries. Two features deserve comment here. The first observation is that the distribution of $E(x)$ is by no means uniform throughout the domain. The error is somewhat larger on the left side of the domain, corresponding to the region where $u'$ becomes large (due to the $(x + 1.1)^{3/2}$ term in the denominator of the example function).

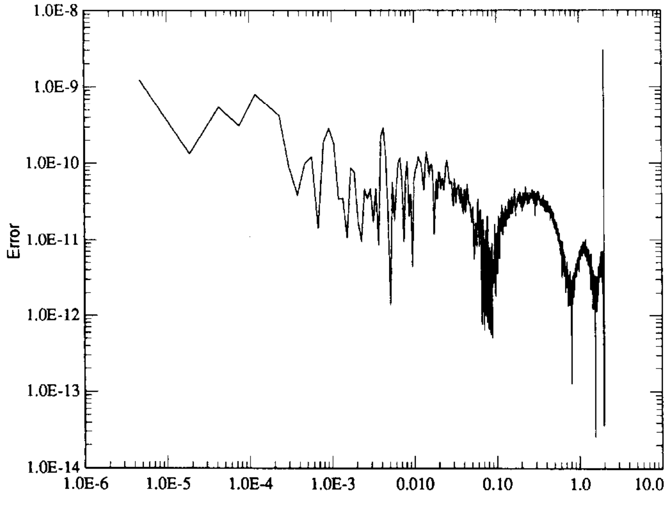

FIG. 6. Spatial distribution of the error in the first derivative calculated with Chebyshev polynomials. $N = 1024$. The error is plotted vs $x+1$ and in logarithmic coordinates to emphasize the $(1-x^2)^{-1/2}$ behavior near the boundary.

The second comment concerns the accuracy of $\hat{u}'(x)$ near the boundaries. It is clear that the approximation to $u'$ by $\hat{u}'$ is worst near the boundaries of the domain, $x \approx \pm 1$. As Fig. 6 indicates, the error rises dramatically near the boundary, gaining two orders of magnitude over the error in the interior of the domain. The behavior, as $x + 1$ approaches 0 is well approximated by a line with slope $-\frac{1}{2}$, implying that $E \propto (1-x^2)^{-1/2}$ as $x \to -1$. The large spike as $x + 1 \to 2$ corresponds to similar behavior at the right edge of the domain. This is somewhat counter-intuitive given that the Chebyshev polynomials are generally regarded to have maximum accuracy near the boundaries [5]. However, while the representation of a function by a Chebyshev series may be most accurate near $x = \pm 1$, these results indicate that the derivatives computed from a Chebyshev series are *least accurate* at the edges of the domain.

This dramatic growth of $E(x)$ in thin regions near the boundaries explains the marked difference in Fig. 1 between the $L_\infty$ error, which is determined by the error near the boundaries, and the $L_2$ error, which is an average over the whole domain. However, it is also a somewhat disturbing result. The boundary regions are often the regions of the problem in which accuracy is most crucial (wall bounded turbulent flows, for example) and in many computations, Chebyshev polynomials are used in large part because of the ease with which boundary conditions can be enforced, and also because the Chebyshev approximation of functions is generally most accurate near the boundaries [5]. The observation that derivatives computed from Chebyshev approximations are *least* accurate in precisely these areas is a disconcerting discovery.

### 3.3. *Weighted Sine Transform*

The observed spatial characteristics and the contrasts with Fourier expansions are unified by writing the Chebyshev transform as a weighted sine transform. The Chebyshev series may be written

$$u(x) = \sum_{k=0}^{N} a_k \cos(k\xi), \tag{18}$$

in which $x$ and $\xi$ are related by

$$x = \cos(\xi), \qquad \xi \in [0, \pi]. \tag{19}$$

By taking the derivative of this expression with respect to $x$, one obtains the following expression for $u'(x)$:

$$u'(x) = \sum_{k=0}^{N} \frac{k a_k \sin(k\xi)}{\sin(\xi)}. \tag{20}$$

The denominator is independent of $k$ and may be taken outside the summation and re-written in terms of $x$, yielding

$$u'(x) = \frac{1}{(1-x^2)^{1/2}} \sum_{k=0}^{N} k a_k \sin(k\xi). \tag{21}$$

When written in this form, we see that the Chebyshev derivative may be expressed as a weighted sine transform whose coefficients are related to the original function's Chebyshev coefficients by $\hat{b}_k = k a_k$. This formulation has the property that we desired from the Fourier series, namely that the coefficients $\hat{b}_k$ are not coupled to each other. Thus, only the high-frequency modes will suffer from the $O(N)$ magnification of the roundoff error while the low-frequency modes should retain their accuracy (within the limits of $\delta$). The weighting function, $w(x) = (1-x^2)^{-1/2}$ is just the familiar Chebyshev weighting function. At $x = \pm 1$, $w(x)$ is singular and so we must expand the full sine series (Eq. (20)) for $\xi = 0$, yielding simple expressions for the values of $\hat{u}'(1)$ and $\hat{u}'(-1)$:

$$\hat{u}'(1) = \sum_{k=0}^{N} k^2 a_k \tag{22a}$$

and

$$\hat{u}'(-1) = \sum_{k=0}^{N} (-1)^k k^2 a_k. \tag{22b}$$

The form of the second derivative may be derived by utilizing the Chebyshev differential equation:

$$T_k'' - \frac{x}{1-x^2} T_k' + \frac{k^2}{1-x^2} T_k = 0. \tag{23}$$

From this we obtain

$$\begin{aligned} u''(x) &= \sum_{k=0}^{N} a_k T_k'' \\ &= \frac{x}{(1-x^2)^{3/2}} \sum_{k=0}^{N} k a_k \sin(k\xi) \\ &\quad - \frac{1}{1-x^2} \sum_{k=0}^{N} k^2 a_k \cos(k\xi). \end{aligned} \tag{24}$$

In this instance, the equation for $u''$ is slightly more complicated than for the first derivative, but the overall form remains the same. Here, there are both Fourier sine and cosine transforms, each with their respective weight functions, both of which are singular at the boundaries. By expanding the equation for the second derivative, (24), we obtain expressions for $u''$ at $x = \pm 1$ ($\xi = 0, \pi$):

$$u''(1) = \sum_{k=0}^{N} \frac{k^2(k^2-1)}{3} a_k \tag{25a}$$

and

$$u''(-1) = \sum \frac{k^2(k^2-1)}{3} (-1)^k a_k. \tag{25b}$$

This procedure was employed in the computation of $u'(x)$ and $u''(x)$, but the results were no different from those shown in Fig. 1. The errors in $u'$ and $u''$ grew in exactly the same manner as for the standard recursion procedure, yielding no improvement. However, the effort was not completely wasted, as Fig. 7 indicates. In this figure, the spatial distribution of the *unweighted* error for the first derivative is plotted in a similar fashion to Fig. 6. Here, the unweighted error is defined as

$$E_{\text{un}}(x) = \left| \frac{u'(x)}{w(x)} - \sum_{k=0}^{N} k a_k \sin(k\xi) \right|. \tag{26}$$

In this form, there are no singular weight functions to cause concern, and the Chebyshev derivative has the form of a pure sine transform, with all the advantages of the independence of modes that the Fourier series enjoy.

Figure 7 shows $E_{\text{un}}(x)$ plotted against $x+1$ and in logarithmic coordinates to allow detailed examination of the region near the boundary. Comparing Figs. 7 and 6, one sees that in the interior, from $x = -0.6$ to $x = +0.6$, the

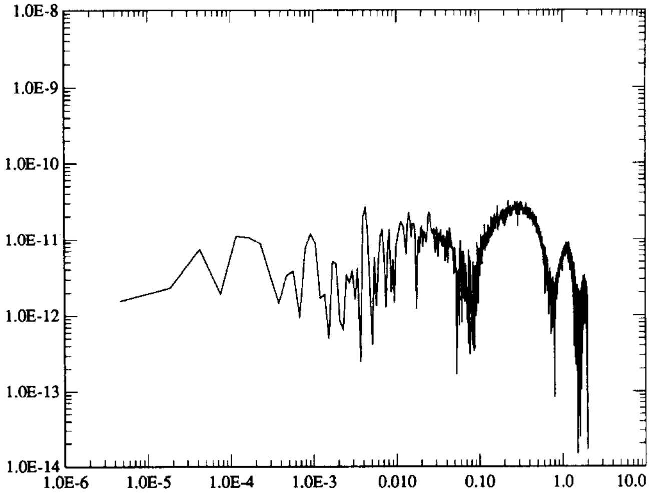

FIG. 7. Spatial distribution of the *unweighted* first derivative, calculated using the Fourier sine transform technique. $N = 1024$. The error is plotted against $x+1$ and in logarithmic coordinates so as to emphasize the behavior near the boundary.

errors appear identical and both the magnitude and the locations of the extrema correspond closely. However, closer to the edges of the domain, the two graphs diverge, and where we previously saw $E(x)$ increase by two orders of magnitude, we now see a slight drop in the magnitude of the error, $E_{\text{un}}(x)$. As before, the magnitude of the error is larger towards $x = -1$, but this may be attributed to the rise in the magnitude of $u'(x)$ in this region.

It should be commented that if one calculates the global errors, $E_\infty$ and $E_2$, for the unweighted sine transform (comparing the computed results with $u'(x)/w(x)$ and $u''(x)/w^3(x)$), one finds that they grow linearly with $N$, as was found to be the case with the Fourier series (Fig. 5). Given the structure of the Chebyshev derivative in this form, this result is not unexpected.

In Fig. 1, we noted that the maximum error, $E_\infty$, grows according to $N^2$, while the rms error, $E_2$, grows somewhat more slowly. This observation may now be explained in light of this discussion. The interior of the domain follows the $O(N\delta)$ growth exhibited by the Fourier series. However, in thin regions near each boundary, where $w(x)$ becomes singular, the error becomes large and increases further as $N$ increases and the collocation points (whose spacing decreases like $1/N^2$ near $x = \pm 1$) approach the singular points in $w$. Thus while the maximum error increases quadratically, the $L_2$ error increases more slowly since it also incorporates the interior error. A final comment concerns the earlier observation that the error at the boundary (Fig. 6) grows approximately as $(1-x^2)^{-1/2}$. This behavior now is fully accounted for by the form of the weight function, $w(x)$.

### 3.4. *Alleviation of the Errors*

Given that we now understand that the most dangerous portion of the errors associated with the computation of the derivative is located at the boundaries we can apply the following scheme in an attempt to alleviate the problem. If we write our example function, $u(x)$, as

$$u(x) = \frac{1+x}{2}\,u(1) + \frac{1-x}{2}\,u(-1) + (1-x^2)\,g(x) \quad (27)$$

we can express the first derivative as

$$u'(x) = \frac{u(1)-u(-1)}{2} + (1-x^2)\,g'(x) - 2xg(x). \quad (28)$$

If we compute $u'$ in this fashion, using standard recursion techniques to find $g'$, we can immediately see that the most serious errors accrued, namely, the errors in $g'(x)$ near the boundaries, will be damped by the weighting function and we might therefore expect to see some improvement in the fidelity of the computed derivative. The second derivative may be found by either applying this procedure iteratively, or by differentiating twice the expression for $u(x)$, (27), yielding an expression for the second derivative directly:

$$u''(x) = (1-x^2)\,g''(x) - 4xg'(x) - 2g(x). \quad (29)$$

This formulation is closely related to one recently proposed by Heinrichs [6] who showed rigorously that this form of pre-conditioning can be used to achieve an $O(N^2)$ condition number for second order Dirichlet problems instead of the usual $O(N^4)$.

One last problem remains to be solved. In computing $g(x)$, we cannot directly evaluate the boundary terms at $x = \pm 1$, since the weighting function is singular at those points. Two options present themselves. If we know the value of $u'$ at the boundaries (from boundary conditions or some other means) then we can find $g(\pm 1)$ by application of L'Hopital's rule:

$$g(\pm 1) = \pm\frac{1}{2}\left(\frac{u(1)-u(-1)}{2} - u'(\pm 1)\right). \quad (30)$$

If, however, the boundary conditions are not known, we can evaluate $g(\pm 1)$ in the following manner:

Since $u(x)$ is approximated by a polynomial of order $N$, we know that $g(x)$ must be a polynomial of order $N-2$. Using the orthogonality of Chebyshev polynomials, we can therefore write

$$\sum_{k=0}^{N} g(x_k)\,T_{N-1}(x_k) = 0 \quad (31a)$$

and

$$\sum_{k=0}^{N} g(x_k)\,T_N(x_k) = 0, \quad (31b)$$

where $x_k$ are the $N+1$ collocation points for the functions $u(x)$ and $g(x)$; $g(\pm 1)$ can readily be found from these conditions.

Figure 8 shows the $L_\infty$ and $L_2$ errors for the first and second derivatives computed using this technique (computing $g(\pm 1)$ using Eq. (31)). For the first derivative the root-mean-square error is actually slightly larger than for the case of the recursion equation. However, the maximum error has decreased somewhat, indicating that the errors are more uniformly distributed over the domain and that the growth of errors near the boundaries has been attenuated. Inspection of the spatial distribution of the errors confirms this. The results for the second derivative also show some improvement in the magnitude of both the rms error and the maximum error.

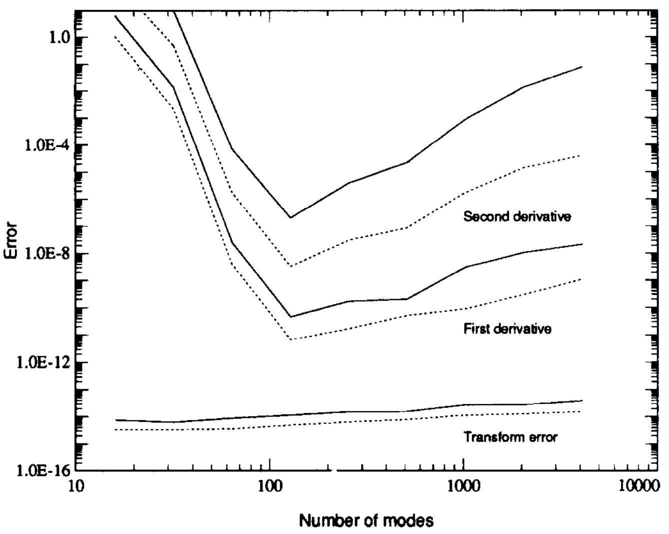

FIG. 8. Maximum (solid line) and rms (dotted line) errors for the first and second derivatives of the example function, $u(x)$, calculated using the weighting technique: $u(x) = (1 - x^2)\, g(x)$.

It is interesting to note that if the values of $u'(\pm 1)$ are known, and L'Hopital's rule is used to find $g(\pm 1)$, then the $L_2$ error remains the same as in Fig. 8, but the $L_\infty$ error rises to almost exactly the same level, indicating an almost completely uniform distribution of error within the domain. For the second derivative, the two curves for $E_\infty$ and $E_2$ are identical, growing somewhat slower than $N^4$.

## 4. MATRIX TECHNIQUES

Transformations into Chebyshev space and explicit determination of the Chebyshev coefficients can be avoided altogether by using a matrix multiplication formulation. If $\mathbf{u} = \{u(x_i)\}$, is the vector consisting of $u(x)$ evaluated at the $N+1$ collocation points and $\mathbf{u}' = \{u'(x_i)\}$ consists of the derivative at the collocation points, then the collocation derivative matrix, $D_N^{(1)}$, is the matrix mapping $\mathbf{u} \mapsto \mathbf{u}'$. For the Gauss–Lobatto collocation points, we have [7, 5]

$$(D_N^{(1)})_{ij} = \begin{cases} \dfrac{\bar{c}_i}{\bar{c}_j} \dfrac{(-1)^{i+j}}{x_i - x_j}, & i \neq j, \\ \dfrac{-x_j}{2(1-x_j)^2}, & 1 \leqslant i = j \leqslant N-1, \\ \dfrac{2N^2+1}{6}, & i = j = 0, \\ -\dfrac{2N^2+1}{6}, & i = j = N, \end{cases} \qquad (32)$$

where $\bar{c}_j = 2$ if $j = 0$, $N$, and is equal to 1 otherwise.

Two matrix multiplications yield $\mathbf{u}''$, the vector containing the second derivative evaluated at the collocation points. More efficiently, the matrix $D_N^{(2)}$ maps $\mathbf{u} \mapsto \mathbf{u}''$. Peyret [8] gives explicit formulae for the entries in $D_N^{(2)}$.

Matrix techniques, though asymptotically slower than the recursion technique, requiring $O(N^2)$ floating point operations, are often faster for small problems. Unlike transform methods, matrix multiplication is amenable to vectorization and the advent of parallel computation ensures their continued use. In this section we examine the errors incurred using matrix multiplication to calculate derivatives.

Unfortunately, the situation here is rather worse than with the recursion method. Figure 9 gives the results for our usual example function (Eq. (12)). As is evident from the graph, the $L_\infty$-error in the first derivative grows as $N^4$. The second derivative was computed using $D_N^{(2)}$, although the results using $D_N^{(1)} D_N^{(1)}$ were almost identical. However, the $L_\infty$ error for $u''$ is observed to grow as $N^6$ (compared with $N^4$ for the recursion technique).

In a similar fashion to the recursion results, the $L_2$-errors grow slightly more slowly than the maximum errors which reflects the spatial distribution of the error, which like the recursion error, is maximum at the domain boundaries, $x = \pm 1$. However, in contrast to the transform method, $E(x) = |u'(x) - \hat{u}'(x)|$ behaves like $1/(1 - |x|)$ as $x \to \pm 1$.

We would expect that the matrix methods should yield results no different from those of the transform techniques already discussed, namely $O(N^2)$ errors in $u'$. In determining the origins of this additional error, we shall concentrate primarily on $D_N^{(1)}$. Standard error analysis of matrix multiplication [9] suggests that an error of $\delta$ times the magnitude of the largest eigenvalue of $D_N^{(1)}$ may be incurred in accumulating such a product. Since the eigenvalues of

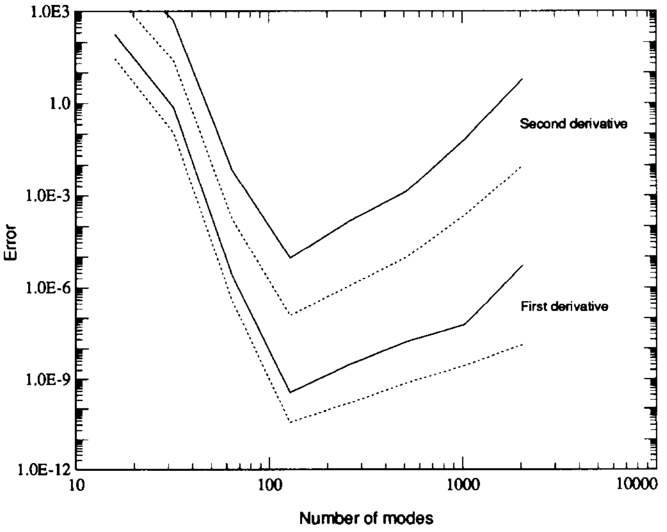

FIG. 9. Maximum (solid line) and rms (dotted line) errors for the first and second derivatives calculated using matrix multiplication techniques. The second derivative is calculated directly from u by multiplying by $D_N^{(2)}$ although the use of $(D_N^{(1)})^2$ yield almost identical results for $u''$.

$D_N^{(1)}$ are only $O(N^2)$ [3] the usual matrix multiplication error is insufficient to account for the $O(N^4\delta)$ observed error. Confirmation of this is found by performing the accumulation of the matrix multiplication in double precision and the remainder in single precision: in this case $E_\infty$ still grows as $N^4$.

The ill-conditioned nature of $D_N^{(1)}$ is widely known (see Orszag, [10]). Figure 10 shows the magnitudes of the elements for $N = 64$. Sizes are indicated by a logarithmic grey scale; the smallest elements ($O(1)$) are shown as white, and the largest elements ($O(N^2)$) are shown as black. This picture does not display the signs of the elements: alternate elements have opposite signs. The matrix is clearly not diagonally dominant; elements on the diagonal are small, whereas the off-diagonal elements are large. The largest elements are concentrated in the top-left and bottom-right corners and approximations to the derivative at collocation points close to the boundary are formed by the sums and differences of these very large coefficients. Since accumulation error has been eliminated as the cause of the large errors in the computed derivative, we are led to examine the calculation of the matrix elements themselves.

Figure 11 indicates the distribution of the errors in $D_{64}^{(1)}$ and is representative of all those we have examined ($N \leqslant 4096$). The error is calculated by subtracting $(D_N^{(1)})_{ij}$ computed in double precision ($\delta \approx 10^{-16}$, or $\approx 15$ significant figures) from $(D_N^{(1)})_{ij}$ computed to 35 decimal places (using an arbitrary precision arithmetic program). This is shown on a logarithmic grey scale with white representing machine precision and black indicating the maximum error. The distribution of errors evidently follows the distribution of the $|(D_N^{(1)})_{ij}|$ and the maximum error occurs in $(D_N^{(1)})_{01}$ and $(D_N^{(1)})_{N,N-1}$. Numerical calculation shows that the maximum errors grow like $N^4$, confirming the following analysis of the computation of the "most dangerous" element, $(D_N^{(1)})_{01}$.

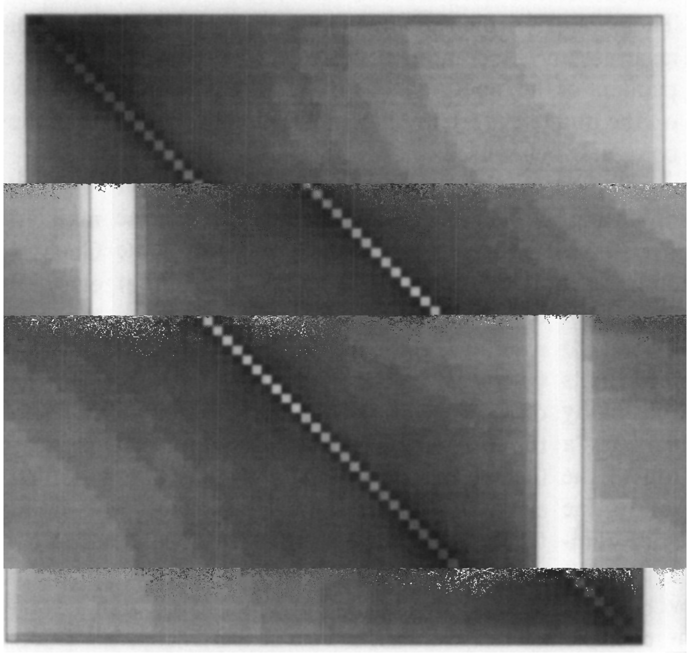

FIG. 10. Magnitude of the elements of $D_N^{(1)}$, $N = 64$, plotted on a logarithmic grey scale. The top left corner of the figure represents $(D_N^{(1)})_{00}$ while the lower right corner represents $(D_N^{(1)})_{NN}$. White represents $O(1)$ elements, while black indicates $O(N^2)$ elements.

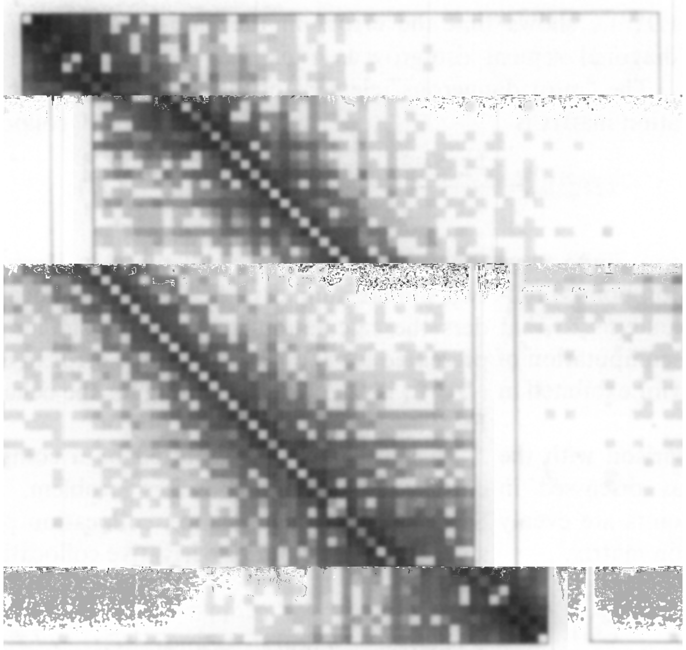

FIG. 11. Magnitude of errors in the elements of $D_N^{(1)}$, $N = 64$, plotted on a logarithmic grey scale. The top left corner of the figure represents $(D_N^{(1)})_{00}$ while the lower right corner represents $(D_N^{(1)})_{NN}$. White represents $O(\delta)$ errors, while black indicates $O(N^4\delta)$ errors.

In computing $(D_N^{(1)})_{01}$ we need to compute the values of two collocation points $x_0$ and $x_1$. The first collocation point, $x_0$, is the boundary point and is therefore calculated exactly, even with finite precision arithmetic. A roundoff error, $\delta$, however, is incurred in the calculation of $x_1 = \cos(1/N)$, so for large $N$,

$$\hat{x}_1 = 1 - \frac{1}{2N^2} + \delta + O(N^{-4}, \delta^2). \tag{33}$$

Using this expression, we may expand $(D_N^{(1)})_{01}$ as a series in $\delta$:

$$\widehat{(D_N^{(1)})}_{01} = \frac{-2}{\hat{x}_0 - \hat{x}_1} \tag{34}$$

$$\approx \frac{-2}{\delta + 1/(2N^2)} \tag{35}$$

$$= -4N^2 + O(N^4\delta). \tag{36}$$

Thus the error in the $(D_N^{(1)})_{01}$ grows like $N^4\delta$, dominating any $N^2$ inner-product accumulation error. Analysis of

$(D_N^{(1)})_{11}$ shows that the errors in the "most dangerous" diagonal element also grow as $N^4\delta$.

The "most dangerous" element of the second derivative collocation matrix is

$$(D_N^{(2)})_{01}=\frac{-2(2N^2+1)(1-x_1)-6}{(1-x_1)^2}. \qquad (37)$$

Here too, finite precision calculation of the difference in two similar numbers (the first two collocation points) renders the calculation imprecise. A similar analysis to that presented for $D_N^{(1)}$ shows that the error in a computation of $(D_N^{(2)})_{01}$ is $O(N^6\delta)$, accounting for the behavior exhibited in Fig. 9.

This reasoning is supported by a comparison with the corresponding Fourier series problem, as discussed in Section 3.1. In this instance, collocation points are evenly spaced and the Fourier derivative collocation matrix,

$$(D_N^{(2)})_{ij}=\begin{cases}\dfrac{(-1)^{i+j}}{2\tan(i-j)\pi/(N+1)}, & i\neq j \\ 0, & i=j,\end{cases} \qquad (38)$$

does not involve any small quantities, allowing its calculation to be accurate to machine precision. Its eigenvalues are uniformly distributed along the imaginary axis between $-i\pi N/2$ and $i\pi N/2$ and we may therefore expect the errors incurred in the Fourier collocation problem to arise primarily from the inner-product accumulation stage and to be bounded by $\pi N\delta/2$. Indeed, making a comparison calculation, we find that the errors in the first derivative grow like $N$ for the first derivative and $N^2$ for the second derivative.

Clearly, in any computation using collocation matrices, this source of error must be avoided, and we must therefore find a method for calculating the matrix elements accurately. One technique is to pre-calculate the matrix using enhanced precision routines, or a computer with the necessary additional precision. Although effective, this approach is somewhat inconvenient, and an alternative method to the direct calculation of the derivative collocation matrices is suggested by observing that if $u^{(i)}$ is the vector consisting of zeros except for a 1 in the $i$th position, then $D_N^{(1)}u^{(i)}$ is the $i$th column of $D_N^{(1)}$. (The functions $u^{(i)}$ are examples of *cardinal functions* (see Boyd [11]).) The collocation matrix may therefore by assembled by successively computing, using the transform-recursion-transform technique, the derivative of $u^{(i)}$ for $j=0, ..., N$. We may expect, from the results of Section 3, that errors in elements of $D_N^{(1)}$ will be $O(N^2)$, much smaller than if they are computed directly, and will be comparable with the inner-product accumulation errors. Indeed, this is found to be the case. A test calculation using a collocation matrix computed in this manner yielded first-derivative errors that grew slightly faster than $N^2$ and second-derivative errors growing slightly faster than $N^4$. Note, however, that the results are worse than the corresponding calculations using the recursion technique, because errors now derive both from those errors in the matrix elements and also the inner product accumulation error. We have not determined the precise cause of this slightly more rapid growth, but suggest that an additional logarithmic factor may be introduced by the FFT during the Chebyshev transforms.

In addition to taking care in the computation of the elements of $D_N^{(1)}$ and $D_N^{(2)}$, one can further reduce the error accumulated during the computation of the derivative by employing the technique presented by Heinrichs [6] and discussed in detail in the previous section with regard to the transform techniques. In common with the transform-recursion techniques, pre-conditioning the original function: $u(x)=(1-x^2)\,g(x)$, results in a more accurate evaluation of the derivative because errors near the boundaries are suppressed by the $(1-x^2)$ weighting.

## 5. CONCLUSIONS

We have determined that using standard techniques for the computation of derivatives of functions approximated by Chebyshev polynomials, large errors, deriving from finite numerical precision, become apparent as the number of polynomials used, $N+1$, increases. These errors scale with $N^2$ and are primarily located in thin regions near the boundaries of the Chebyshev domain at $x=\pm 1$. In the interior of the domain, the errors grow linearly with $N$. These errors, which define a lower bound on the level of accuracy in the computed derivative, are essentially independent of the function being differentiated, depending only on the number of Chebyshev modes and the precision of the computer being used.

The source of the error may be viewed in two distinct manners: focusing either on the characteristics of the Chebyshev coefficients, or on the spatial structure of the error. Examining the Chebyshev coefficients, we find that the problem is initiated by the introduction of a small roundoff error during the transform from physical to Chebyshev space. The process of computing the coefficients of the derivative magnifies that error by $O(N)$ and finally the inverse transform, back to physical space, introduces a further magnification by $O(N)$. The magnification of the error during the computation of $\hat{b}_k$ is inherent in the Chebyshev problem. It is not dependent on how one computes $\hat{b}_k$, but only on there being $O(\delta)$ errors in the $a_k$s beforehand. The strong coupling between the Chebyshev coefficients that is characteristic of the Chebyshev differentiation serves to distribute this large error evenly amongst all of the coefficients, from high values of $k$, down to $k=0$. This is distinct from the equivalent problem using Fourier series, in which case the independence of the Fourier modes during the computation of the coefficients of $u'$ effectively isolates the large errors in the high-frequency modes. It is this essential distinction that results in $O(N)$ errors when using Fourier series but $O(N^2)$ errors when using Chebyshev series.

Matrix methods of computing the derivatives yield identical errors. From this perspective, we see that the errors result from the $O(N^2)$ eigenvalues of the $D_N^{(1)}$ differentiation matrix and are an unavoidable consequence of the finite-precision matrix multiplication. In addition, unless particular care is taken, the elements of $D_N^{(1)}$ can themselves contain $O(N^4\delta)$ errors, resulting in overall errors of $O(N^4\delta)$ in the computed first derivative and $O(N^6\delta)$ in the second derivative.

However, the picture is not all bleak. The problems encountered with both transform and matrix methods can be somewhat alleviated by pre-conditioning the function to be differentiated with a weighting function, $(1-x^2)$ [6]. This suppresses the growth of errors at the boundaries leading to a more uniform error in the first derivative and a reduction in the maximum error. In addition to this technique, other guidelines should be followed in order to minimize the growth of these errors. First, the computational procedure should be constructed so as to minimize the number of derivatives that must be computed. One way in which this can be achieved is to write the equations in integral form, thus replacing numerical differentiation with numerical integration which is often less prone to problems of this nature. This approach has been widely used in the solution of ordinary differential equations (see, for example, Fox and Parker [12], Zebib [13], and Greengard [4]). Boyd [11] also gives a comprehensive discussion of these approaches. An alternative way to minimize these errors is to avoid large values of $N$ by domain decomposition or spectral-element methods (for example, Korczak and Patera [14]).

We must emphasize that these roundoff error effects are not necessarily problems in the *solution* of differential equations, but in the computation of the derivatives of functions. Restated, the sensitivity of a system of equations to roundoff error is not at all the same as the demonstrated ill-conditioned nature of the numerical differentiation process. Indeed the accuracy of solutions to linear differential equations (by means of the Tau method, for example) does not appear to suffer as $N$ increases. However, this will not be true for nonlinear systems, in which case the equation is often solved by advancing in time using a marching scheme (e.g., Runge–Kutta), forced by a right-hand side assembled using the solution and its spatial derivatives at the previous time level. In view of the results presented here, one can see that for large $N$, the errors that arise in the derivatives can distort the forcing term, which may render the overall solution inaccurate.

The effect of these errors will not necessarily destroy the physical character of a numerical solution over a long integration time. However, it will introduce spurious "noise" into the solution which may force unintended or undesirable behavior in the system being solved. For example, when solving a system possessing unstable modes, the high level of background numerical noise may artificially trigger those instabilities (a numerical analogue to high levels of free-stream turbulence in a wind tunnel).

One final comment is that the over-resolution of a problem can lead to disastrous results. It is often the practice to allow more modes than are necessary in order to "be safe" regarding the spatial resolution. This can be dangerous since the use of too many modes can only aggravate this problem. In checking grid dependence, one should be aware that, although truncation error decreases as $N$ grows, the effect of roundoff error increases at the same time. For some value of $N$, one reaches a point of diminishing returns in which increased resolution will decrease rather than increase the accuracy of the computation. Thus, for complicated problems in which $N$ is reasonably large, the smallest number of modes necessary to sufficiently resolve the problem should be used.

## ACKNOWLEDGMENTS

We thank David Gottlieb for many helpful discussions and we are grateful to the referees for their comments. This work was supported by DARPA-URI Grant N00014-86-K074. Many of the computational results described here are based on work performed at the John von Neumann Supercomputing Center and the Pittsburgh Supercomputing Center, both of which are supported by the National Science Foundation.

## REFERENCES

1. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications* (SIAM, Philadelphia, 1977).
2. J. Kim, P. Moin, and R. Moser, *J. Fluid Mech.* **177**, 133 (1987).
3. L. N. Trefethen and M. R. Trummer, *SIAM J. Numer. Anal.* **24**, No. 5 1008 (1987).
4. L. Greengard, "Spectral Integration and Two-Point Boundary Value Problems, Yale University, Department of Computer Science Research Report, 1988, submitted.
5. A. Solomonoff and E. Turkel, *J. Comput. Phys.* **18**, 239 (1989).
6. W. Heinrichs, *Math. Comput.* **53**, 103 (1989).
7. D. Gottlieb, M. Y. Hussaini, and S. A. Orszag, in *Spectral Methods for Partial Differential Equations*, edited by D. G. Voigt, D. Gottlieb, and M. Y. Hussaini (SIAM-CMBS, Philadelphia, 1984), p. 1.
8. R. Peyret, *Introduction to Spectral Methods* (von Karman Institute Lecture Series, Rhode-St-Genese, 1986).
9. G. H. Golub and C. F. van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 1983).
10. S. A. Orszag, *J. Comput. Phys.* **37**, 70 (1980).
11. J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Lecture Notes in Engineering, Vol. 49, (Springer-Verlag, New York/Berlin, 1989).
12. L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analysis* (Oxford Univ. Press, London, 1968).
13. A. Zebib, *J. Comput. Phys.* **53**, 443 (1984).
14. K. Z. Korczak and A. T. Patera, *J. Comput. Phys.* **62**, 361 (1986).