JOURNAL OF COMPUTATIONAL PHYSICS 116, 380–383 (1994)

NOTE

Roundoff Error in Computing Derivatives Using the Chebyshev Differentiation Matrix*

1. INTRODUCTION

Differentiation of a function f using the Chebyshev pseudo-spectral method is a linear operator, so that differentiation can be reduced to multiplication by a matrix D. When the collocation points are the commonly used Gauss–Lobatto points, an explicit formula for D can be obtained [4]. A procedure to derive the matrix for more general collocation points is presented in [5].

Ill-conditioning of D and of matrices related to D which account for the boundary conditions, has been identified as a potential problem for the implementation of Chebyshev pseudo-spectral methods for the solution of partial differential equations [1–3, 6, 7]. If there are $N + 1$ collocation points and boundary conditions are not incorporated into D, then it can easily be seen that the only eigenvalue of D is 0, with algebraic multiplicity $N + 1$. The null eigenvector corresponds to differentiation of the constant function. When D is modified to account for boundary conditions, the modified matrix possesses eigenvalues which are $O(N^2)$ [2], which accounts for the severe timestep restriction in using Chebyshev pseudo-spectral methods.

Aspects of the ill-conditioning of D were studied in [6, 7]. In [6] it was shown that there could be severe numerical difficulties in computing the large (in absolute value) eigenvalues and corresponding eigenvectors of the modified matrix. This led to the conclusion that stability bounds can be dependent on machine precision. However, this conclusion was not directly related to the accuracy of the approximation. In [7] the spectrum of a matrix corresponding to a second-order differential operator with Dirichlet boundary conditions was analyzed. It was shown that a fraction $((\pi - 2)/\pi)$ of the eigenvalues did not approximate the exact eigenvalues, and that these large eigenvalues were $O(N^4)$, leading to poorly conditioned matrices. These conclusions were not directly related to the accuracy of approximating the derivatives for any particular function.

The relationship between ill-conditioning of D and the accuracy of derivative calculations was studied in [1, 3]. In practice Chebyshev pseudo-spectral differentiation can be implemented either by matrix operations (i.e., multiplication by a matrix) or by transform techniques, together with a recursion. In [1, 3] it was shown that the transform-recursion techniques were generally less susceptible to roundoff error than implementations employing matrix operations. In [1] roundoff error associated with the matrix D was attributed to inaccurate computation of certain matrix elements, particularly certain large elements in the upper left and lower right corners of the matrix. A technique to reduce the influence of these large elements by modifying the vector on which D operates was proposed. In [3] other implementations of the matrix multiply algorithm were considered. It was shown that the susceptibility to roundoff error could be reduced by reformulating the elements of D employing trigonometric identities. This was combined with a flipping technique whereby symmetries of certain matrix elements were exploited to avoid calculating half of the reformulated matrix (the bottom half in the ordering in [3]). The resulting reformulated matrix combined with the flipping technique, was shown to yield comparable accuracy to that obtained by transform techniques in computing pseudo-spectral derivatives.

In this note we identify one cause of roundoff error in using the Chebyshev differentiation matrix as the failure of the computed matrix to exactly preserve the constant null vector of the differentiation matrix, or equivalently, the row sum of the elements of D. We show that a simple procedure to compute the diagonal elements of the matrix, so that the constant null vector is preserved, allows for a dramatic reduction in roundoff error in the computation of high-order derivatives. For the cases we have tested, we find that the accuracy obtained from the matrix multiply approach is comparable to the accuracy obtained from transform techniques.

2. NUMERICAL METHOD

We assume that there are $N + 1$ collocation points and that the indices i and j run from 0 to $N + 1$ unless otherwise stated. An explicit formula for the matrix D is [2, 4, 5]

$$
\begin{aligned}
&D_{ij} = c_i(-1)^{i+j}/(c_j(x_i - x_j)), \quad i \neq j,\\
&D_{ii} = -x_i/(2(1 - x_i^2)), \quad 1 \leq i \leq N - 1,\\
&D_{00} = (2N^2 + 1)/6, \quad D_{NN} = -D_{00},
\end{aligned}
\tag{1}
$$

* Supported in part by NSF Grants MMS 91-02981 and DMS 93-01635 and DOE Grant DEFG02ER25027.

0021-9991/94 $6.00

where $c_j = 1$ for $1 \leq j \leq N-1$, $c_0 = c_N = 2$ and x_j are the Gauss–Lobatto points

$$x_j = \cos(j\pi/J) \quad (j = 0, ..., N).$$

If f is any function and $\hat{f}$ denotes the vector $f_j = f(x_j)$ then the matrix product $D\hat{f}$ is the Chebyshev pseudo-spectral approximation to $f'(x_j)$. It is known that this matrix product can be computed in $O(N \log N)$ operations by use of fast Fourier transform techniques coupled to a recursion [2]. However, for many applications the matrix multiply approach can be more efficient, as (i) it is vectorizable, (ii) it is readily parallelizable, and (iii) it can be more amenable to the implementation of implicit time differencing schemes.

It can easily be seen that for each row of D we have

$$\sum_{j=0}^{j=N} D_{ij} = 0. \tag{2}$$

This is simply a statement that the constant vector is a null vector of D. Straightforward implementation of (1) leads to a matrix which fails to satisfy (2). For example, in single precision on a Cray C90 computation of D with $N = 16$ leads to values for the absolute value of the left-hand side of (2) varying from 3×10^{-11} to 6×10^{-14}. The largest errors occur near the boundaries and the smallest errors occur in the interior. Due to accumulation of roundoff error, the failure to satisfy (2) becomes more pronounced for larger values of N. For example, increasing N to 97 leads to values for the absolute value of the left-hand side of (2) ranging from 3×10^{-8} near the boundaries to 6×10^{-14} in the interior. We show below that this can lead to significant errors, particularly in approximating higher derivatives.

We can enforce (2) by modifying the calculation of the diagonals. Specifically, the matrix D can be constructed by defining the off-diagonal entries as above and then defining the diagonal entries by

$$D_{ii} = -\sum_{j=0, j\neq i}^{j=N} D_{ij}. \tag{3}$$

We have found that the use of (3) can lead to significantly greater accuracy in the computation of higher derivatives for a wide range of functions. We have tested many functions, and we present results for two such functions below. We note that it is possible to use a similar technique to correct the maximum entry (in absolute value) in each row of D, rather than the diagonal entry. We have tested this and find that the results change only slightly.

3. EXAMPLES

We consider the approximation of the following functions defined for $-1 \leq x \leq 1$:

$$f(x) = \sin(x\pi/2), \tag{4}$$

$$f(x) = x^8. \tag{5}$$

These functions are gradually varying and, for the values of N that we consider, the approximation of these functions by a Chebyshev interpolant is essentially exact (it is exact for (5)). Errors in approximating the higher derivatives for (4) and (5) can be attributed to roundoff errors due to ill-conditioning of the matrix D. We note that (4) satisfies the boundary conditions

$$f'(\pm 1) = f'''(\pm 1) = 0, \tag{6}$$

allowing us to test the effect of including boundary conditions in the matrix.

We consider the error in approximating (4) and (5) for various values of N and for three techniques to implement Chebyshev pseudo-spectral differentiation. In method 1, D is constructed using a straightforward implementation of (1). In method 2 we modify the calculation of the diagonals as indicated in (3). In method 3 we employ a fast Fourier transform, together with a recursion, so that the matrix D is not explicitly constructed. For methods 1 and 2 the Cray routine MXV is employed to compute the product of a matrix and a vector. Higher derivatives are computed by successive matrix multiplications. Each product of a matrix and a vector is thus performed with the corrected matrix. If higher derivatives are computed by first performing a matrix product and then computing the product of a matrix and a vector (e.g., computing D^2 first, in order to compute second derivatives), it may be necessary to correct the diagonals of the resulting matrix product. In addition, it may be useful to employ a similar correction to matrices obtained from matrix inversion procedures, which arise in implicit time differencing schemes. For method 3 we employed a hand-coded fast Fourier transform. We measure errors in the second and fourth derivatives. E_2 is the maximum error in the second derivative, where the maximum is taken over all collocation points employed, while E_4 is the maximum relative error for the fourth derivative. Maximum relative errors are taken for the fourth derivatives since the values of the fourth derivative are significantly larger than the values of the second derivatives for the functions that are tested here. We point out that errors in the second derivative are crucially important for the calculation of diffusion, while errors in the fourth derivative are important for the study of fourth-order equations, such as the Kuramoto–Sivashinsky equation, both of which are important in the modeling of nonlinear phenomena in a variety of physical applications. In both cases a large number of collocation points may be needed in order to adequately resolve the solution.

We first consider the case where boundary conditions are not explicitly imposed. Errors for the different methods are presented in Tables I and II for (4) and (5), respectively. We note that the specific numbers obtained depend slightly on the particular hardware employed and on the way that the

TABLE I

Results for Approximating Derivatives of $f(x) = \sin(x\pi/2)$

Method	N	E_2	E_4
1	17	3.39×10^{-9}	1.24×10^{-6}
2	17	4.96×10^{-11}	1.32×10^{-8}
3	17	2.06×10^{-10}	4.27×10^{-8}
1	49	2.77×10^{-7}	9.55×10^{-3}
2	49	7.92×10^{-9}	2.47×10^{-4}
3	49	1.21×10^{-8}	2.27×10^{-4}
1	97	1.37×10^{-4}	6.03×10^{1}
2	97	1.18×10^{-7}	6.37×10^{-2}
3	97	3.70×10^{-6}	1.37×10^{0}
1	257	2.86×10^{-2}	6.13×10^{5}
2	257	1.03×10^{-5}	2.66×10^{2}
3	257	5.92×10^{-4}	1.16×10^{4}

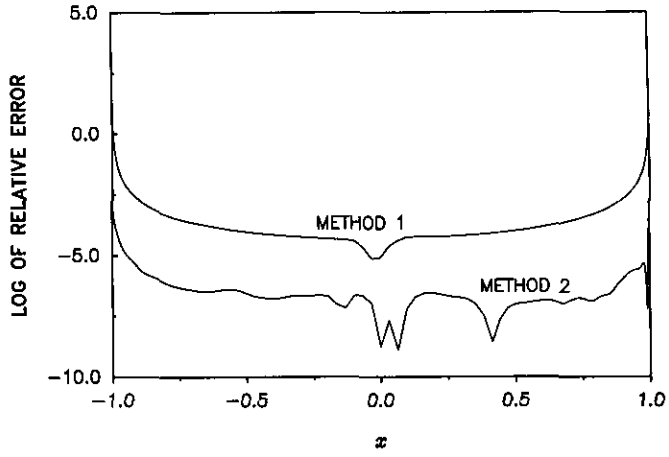

FIG. 1. Logarithm of relative error for approximation of fourth derivative of (4) using 97 collocation points.

summation in (3) is implemented. However, we have found that these differences are very small compared to the differences between the proposed method and the other two methods. The results presented here were obtained on a Cray C90, using a straightforward implementation of (3).

The errors in both cases can be directly attributed to roundoff in computing the matrix elements and in performing the matrix or transform operations. Indeed, computations for (5) should be exact as it is a polynomial. The results demonstrate the dramatic increase in ill-conditioning as N increases, as well as the significant improvement in accuracy when (3) is employed to construct the matrix. We note that the errors from the fast Fourier transform approach are roughly comparable to those obtained from the matrix multiply approach employing (3). It was shown in [1, 3] that this approach is generally more accurate than matrix multiplication techniques; however, this does not show up for our implementation of the fast Fourier transform approach.

The spatial distribution for the error can be seen in Fig. 1, where we plot the logarithm of the relative error for the fourth derivative of (4) employing methods 1 and 2 for $N = 97$. We note that while the error is largest at the boundaries, where the diagonal terms are largest (an effect also observed in [1]), a significant reduction in the error is realized for all values of x. The results presented for $f(x) = x^8$ are typical. This is illustrated in Fig. 2, where we plot the logarithm of the maximum error for the second derivative of $f(x) = x^j$, $j = 1,..., 95$, with $N = 97$. In addition, similar results have been found for other values of N and for other smooth functions.

Finally, we consider the effect of incorporating boundary conditions in the matrix. We consider the approximation of (4)

TABLE II

Results for Approximating Derivatives of $f(x) = x^8$

Method	N	E_2	E_4
1	17	3.50×10^{-9}	7.40×10^{-9}
2	17	3.71×10^{-10}	3.73×10^{-10}
3	17	4.28×10^{-10}	2.53×10^{-10}
1	49	3.14×10^{-7}	3.39×10^{-5}
2	49	5.87×10^{-8}	4.09×10^{-6}
3	49	7.33×10^{-8}	4.61×10^{-6}
1	97	1.37×10^{-4}	2.18×10^{-1}
2	97	3.07×10^{-7}	3.49×10^{-4}
3	97	1.01×10^{-6}	9.36×10^{-4}
1	257	2.86×10^{-2}	2.22×10^{3}
2	257	3.90×10^{-5}	1.25×10^{0}
3	257	2.31×10^{-4}	1.46×10^{1}

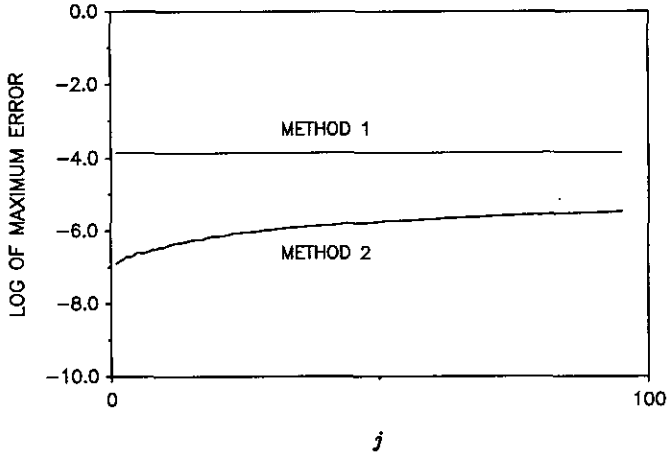

FIG. 2. Logarithm of maximum norm error for the computation of the second derivative of $f(x) = x^j$, $j = 1, ..., 95$, using 97 collocation points.

TABLE III

Results for Approximating Derivatives of $f(x) = \sin(x\pi/2)$ Incorporating Boundary Conditions in the Matrix

Method	N	E_2	E_4
1	17	8.42×10^{-10}	4.43×10^{-7}
2	17	1.29×10^{-11}	5.44×10^{-9}
3	17	5.13×10^{-11}	2.75×10^{-8}
1	49	8.73×10^{-8}	3.94×10^{-3}
2	49	1.91×10^{-9}	8.97×10^{-5}
3	49	4.79×10^{-9}	2.07×10^{-4}
1	97	2.83×10^{-5}	1.84×10^{1}
2	97	2.64×10^{-8}	2.01×10^{-2}
3	97	1.69×10^{-7}	1.14×10^{-1}
1	257	5.49×10^{-3}	1.74×10^{5}
2	257	2.49×10^{-6}	9.51×10^{1}
3	257	8.71×10^{-6}	2.99×10^{2}

with the boundary conditions (6) (for the error E_2 only the boundary conditions on the first derivative enter). The results are presented in Table III.

The results in Table III demonstrate that incorporating boundary conditions within the matrix, i.e., indirectly imposing the boundary conditions, can lead to a reduction in error for the higher derivatives employing all approaches. However, the basic ill-conditioning in progressively applying the differentiation operators is still apparent. A significant improvement in the computation of higher derivatives is still obtained by the use of (3). In all cases we note that the computation of fourth derivatives for problems where $O(100)$ collocation points are required to resolve the solution, can be inaccurate for Chebychev pseudo-spectral methods as currently employed.

ACKNOWLEDGMENTS

The authors thank Alex Solomonoff for his very useful comments.

REFERENCES

1. K. Breuer and R. Everson, *J. Comput. Phys.* **99,** 56 (1992).
2. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1987).
3. W. S. Don and A. Solomonoff, Accuracy and speed in computing the Chebyshev collocation derivative, preprint.
4. D. Gottlieb and E. Turkel, in *Lecture Notes in Mathematics,* Vol. 115 Springer-Verlag, New York/Berlin, 1985, 115.
5. A Solomonoff and E. Turkel, *J. Comput. Phys.* **81,** 239 (1989).
6. L. N. Trefethen and M. R. Trummer, *SIAM J. Numer. Anal.* **24,** 1008 (1987).
7. J. A. C. Weideman and L. N. Trefethen, *SIAM J. Numer. Anal.* **25,** 1279 (1988).

Received January 3, 1994; revised August 22, 1994

ALVIN BAYLISS
ANDREAS CLASS[1]
BERNARD J. MATKOWSKY

Department of Engineering Sciences and Applied Mathematics
Northwestern University
Evanston, Illinois 60208

[1] Permanent address: IATF, Kernforschungszentrum Karlsruhe GmbH, Karlsruhe, Germany.