Applied Numerical Mathematics 33 (2000) 143–149
APPLIED NUMERICAL MATHEMATICS
www.elsevier.nl/locate/apnum

Improving the accuracy of the matrix differentiation method for arbitrary collocation points

Richard Baltensperger [1]
Institut de Mathématiques, Université de Fribourg, CH-1700 Fribourg/Pérolles, Switzerland

Abstract

We discuss the errors incurred when using the standard formula for calculating differentiation matrices in spectral methods and suggest more precise ways of calculating the derivatives and their matrices. © 2000 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Spectral methods; Differentiation matrix; Barycentric formula; Roundoff error

0. Introduction

In the matrix version of the collocation methods one expands the approximate solution u_n as an interpolating polynomial in its Lagrangian form

$$u_n(x) := \sum_{k=0}^{n} u_n(x_k) L_k(x), \tag{1}$$

where the $L_k(x)$ are the Lagrange polynomials corresponding to distinct interpolation points x_k, $k = 0(1)n$, [6]. The derivatives of the approximate solution u_n are then estimated at the collocation points by differentiating (1) and evaluating the resulting expression, see [4]. This yields

$$u_n^{(p)}(x_j) = \sum_{k=0}^{n} u_n(x_k) L_k^{(p)}(x_j), \quad p = 1, 2, \ldots, \tag{2}$$

or in matrix notation

$$\boldsymbol{u}^{(p)} = \boldsymbol{D}^{(p)} \boldsymbol{u}, \tag{3}$$

where

$$\boldsymbol{u} := \left[u_n(x_0), \ldots, u_n(x_n)\right]^{\mathrm{T}}, \qquad \boldsymbol{u}^{(p)} := \left[u_n^{(p)}(x_0), \ldots, u_n^{(p)}(x_n)\right]^{\mathrm{T}}, \tag{4}$$

[1] E-mail: richard.baltensperger@unifr.ch

PII: S0168-9274(99)00077-X

and where $\boldsymbol{D}^{(p)}$ is the $(n+1)\times(n+1)$ matrix whose entries are given by

$$D_{jk}^{(p)} := L_k^{(p)}(x_j), \quad j,k = 0(1)n. \tag{5}$$

Such derivatives are calculated a great many times when solving time independent problems with iterative methods or time evolution problems with the method of lines, see [4].

The authors of [1–3,5,7] have presented different ideas for alleviating the effects of roundoff-errors in the calculation of derivatives and differentiation matrices for Čebyšev–Gauss–Lobatto points (i.e., $x_k = \cos(k\pi/n)$, $k = 0(1)n$).

In the present work we will briefly review the results of [1] in a more general setting (in Section 1) and apply this technique to Gauss–Lobatto points (in Section 2) and to Čebyšev–Gauss–Radau points (in Section 3), for which explicit formulas for the differentiation matrices are known but where no fast transform method (independent of the machine precision) is known for calculating approximate derivatives.

1. Sources and alleviation of errors

In [1,2] the relation

$$D_{jj}^{(p)} = -\sum_{k=0,\ k\neq j}^{n} D_{jk}^{(p)}, \quad p = 1,2,\ldots, \tag{6}$$

is used to calculate the diagonal elements of the differentiation matrices. Relation (6) arises from the fact that, if the interpolated function u takes the value one everywhere ($u(x)\equiv 1$), then the polynomial (1) interpolates u exactly, so that $\sum_{k=0}^{n} L_k(x) \equiv 1$. For the derivatives of this sum this implies, in particular,

$$\sum_{k=0}^{n} L_k^{(p)}(x_j) = 0 \quad (p = 1,2,\ldots),$$

or (6) for $\boldsymbol{D}^{(p)}$. Every diagonal element of the differentiation matrix thus should equal the negative sum of all other elements on its row.

Experiments show that the maximum error incurred in calculating the pth ($p = 1,2$) derivative of u grows at the same rate as the maximum value (M) of the sum of the elements of the row (see Tables 1 and 3 or the numerical experiments in [1]). The results in [1,2] and in the next two sections of the present paper demonstrate that preserving the relation (6) is more important than calculating some elements of the differentiation matrices exactly.

To alleviate the error, we propose (as in [1]) to use the barycentric representation of u_n [6]:

$$u_n(x) = \frac{\sum_{k=0}^{n}(\lambda_k/(x-x_k))u_n(x_k)}{\sum_{k=0}^{n}\lambda_k/(x-x_k)}, \tag{7}$$

where

$$\lambda_k^{-1} := \prod_{j=0,\ j\neq k}^{n}(x_k - x_j).$$

In [8], Schneider and Werner have given a *formula* for differentiating rational functions written in their barycentric form. They have also suggested an *algorithm* for calculating all derivatives of such functions;

this algorithm involves divided differences, which account for the better stability in the Čebyšev–Gauss–Lobatto case [1]. As in the latter article, we use this algorithm to calculate the derivative of the polynomial u_n. It should however be noted that this algorithm does not produce the differentiation matrices.

For the first and second order differentiation matrices, Schneider and Werner's formula reads

$$D_{jk}^{(1)} = \begin{cases} \dfrac{\lambda_k}{\lambda_j} \dfrac{1}{x_j - x_k}, & \text{if } j \neq k, \\ -\displaystyle\sum_{i=0, i\neq j}^{n} \dfrac{\lambda_i}{\lambda_j} \dfrac{1}{x_j - x_i}, & \text{if } j = k, \end{cases} \tag{8}$$

and

$$D_{jk}^{(2)} = \begin{cases} 2D_{jk}^{(1)}\left(D_{jj}^{(1)} - \dfrac{1}{x_j - x_k}\right), & \text{if } j \neq k, \\ 2\left(D_{jj}^{(1)}\right)^2 + 2\displaystyle\sum_{i=0, i\neq j}^{n} D_{ji}^{(1)} \dfrac{1}{x_j - x_i}, & \text{if } j = k. \end{cases} \tag{9}$$

Moreover, to diminish the errors due to smearing [6] in the calculation of every diagonal element of the first order differentiation matrix, we rearrange the summation in (6) from the smallest to the largest term in absolute value (as already done in [1]).

And also for the second order differentiation matrix, instead of the second line of (9), we use the relation (6) and the rearrangement proposed above.

Note that the formulas (8) and (9) can be used for any set of points: If we want to approximate the derivative of a function by its interpolating polynomial, we can use these formulas and the relation (6).

In the next two sections we give errors in calculating the derivatives for Gauss–Lobatto points and for Čebyšev–Gauss–Radau points of two different example functions, $u(x) := \sin(x)$ and $v(x) := 1/(1 + x^2)$ on $[-1, 1]$. We measure the error in the numerical approximation u_n with the maximum- or L_∞-error

$$E_\infty := \max_{0\leqslant k\leqslant n} \left|u_n(x_k) - u(x_k)\right|.$$

The computations were performed on an AlphaServer 2100A 5/300 using MATLAB. The results are presented in Tables 1–4. There (1) denotes the results with the standard formula, (2) with the algorithm of Schneider and Werner, (3) with the explicit formulas (8) and (9) with rearrangement (and relation (6) for (9)) and (M) the maximum row sum of the elements of the matrix used in methods (1) and (3).

2. Computational examples for Gauss–Lobatto points

The first order differentiation matrix for (Legendre–)Gauss–Lobatto points can be given explicitly (see [4,10])

$$D_{jk}^{(1)} = \begin{cases} \dfrac{(n+1)n}{4}, & \text{if } j = k = 0, \\ -\dfrac{(n+1)n}{4}, & \text{if } j = k = n, \\ 0, & \text{if } j = k \neq 0,\ n, \\ \dfrac{L_n(x_j)}{L_n(x_k)} \dfrac{1}{x_j - x_k}, & \text{if } j \neq k, \end{cases} \tag{10}$$

Table 1
Errors in approximating the first (top) and second (bottom) derivatives of $u(x)$ using the three different techniques

n	16	32	64	128	256	512
(1)	$3.21 \cdot 10^{-12}$	$8.05 \cdot 10^{-11}$	$6.68 \cdot 10^{-10}$	$1.10 \cdot 10^{-8}$	$1.09 \cdot 10^{-7}$	$2.35 \cdot 10^{-5}$
(M)	$3.78 \cdot 10^{-12}$	$9.58 \cdot 10^{-11}$	$7.95 \cdot 10^{-10}$	$1.31 \cdot 10^{-8}$	$1.29 \cdot 10^{-7}$	$2.79 \cdot 10^{-5}$
(2)	$3.77 \cdot 10^{-15}$	$1.88 \cdot 10^{-14}$	$3.45 \cdot 10^{-14}$	$5.03 \cdot 10^{-13}$	$1.71 \cdot 10^{-12}$	$5.14 \cdot 10^{-12}$
(3)	$7.99 \cdot 10^{-15}$	$1.38 \cdot 10^{-14}$	$4.10 \cdot 10^{-14}$	$1.18 \cdot 10^{-12}$	$1.63 \cdot 10^{-12}$	$2.04 \cdot 10^{-12}$
(M)	$2.00 \cdot 10^{-15}$	$5.69 \cdot 10^{-15}$	$1.05 \cdot 10^{-13}$	$4.47 \cdot 10^{-13}$	$9.77 \cdot 10^{-13}$	$4.28 \cdot 10^{-12}$
(1)	$1.34 \cdot 10^{-10}$	$1.41 \cdot 10^{-8}$	$3.97 \cdot 10^{-7}$	$2.50 \cdot 10^{-5}$	$1.22 \cdot 10^{-3}$	$9.33 \cdot 10^{-1}$
(M)	$1.51 \cdot 10^{-10}$	$1.67 \cdot 10^{-8}$	$4.73 \cdot 10^{-7}$	$2.97 \cdot 10^{-5}$	$1.45 \cdot 10^{-3}$	$1.11 \cdot 10^{0}$
(2)	$4.87 \cdot 10^{-13}$	$7.24 \cdot 10^{-12}$	$5.49 \cdot 10^{-11}$	$2.46 \cdot 10^{-9}$	$3.70 \cdot 10^{-8}$	$1.69 \cdot 10^{-7}$
(3)	$1.22 \cdot 10^{-12}$	$6.91 \cdot 10^{-12}$	$6.59 \cdot 10^{-11}$	$1.93 \cdot 10^{-9}$	$5.78 \cdot 10^{-8}$	$4.78 \cdot 10^{-7}$
(M)	$2.13 \cdot 10^{-13}$	$2.01 \cdot 10^{-12}$	$4.33 \cdot 10^{-11}$	$4.66 \cdot 10^{-10}$	$9.30 \cdot 10^{-9}$	$2.33 \cdot 10^{-7}$

Table 2
Errors in approximating the first (top) and second (bottom) derivatives of $v(x)$ using the three different techniques

n	16	32	64	128	256	512
(1)	$3.47 \cdot 10^{-5}$	$8.15 \cdot 10^{-11}$	$3.98 \cdot 10^{-10}$	$6.54 \cdot 10^{-9}$	$6.46 \cdot 10^{-8}$	$1.40 \cdot 10^{-5}$
(2)	$3.47 \cdot 10^{-5}$	$7.14 \cdot 10^{-11}$	$3.49 \cdot 10^{-14}$	$1.62 \cdot 10^{-13}$	$4.51 \cdot 10^{-13}$	$5.47 \cdot 10^{-12}$
(3)	$3.47 \cdot 10^{-5}$	$7.14 \cdot 10^{-11}$	$2.13 \cdot 10^{-14}$	$4.55 \cdot 10^{-13}$	$1.82 \cdot 10^{-12}$	$7.27 \cdot 10^{-12}$
(1)	$4.71 \cdot 10^{-3}$	$3.85 \cdot 10^{-8}$	$2.37 \cdot 10^{-7}$	$1.48 \cdot 10^{-5}$	$7.24 \cdot 10^{-4}$	$5.54 \cdot 10^{-1}$
(2)	$4.71 \cdot 10^{-3}$	$3.77 \cdot 10^{-8}$	$6.53 \cdot 10^{-11}$	$1.49 \cdot 10^{-9}$	$4.46 \cdot 10^{-9}$	$6.47 \cdot 10^{-7}$
(3)	$4.71 \cdot 10^{-3}$	$3.77 \cdot 10^{-8}$	$1.16 \cdot 10^{-10}$	$1.86 \cdot 10^{-9}$	$3.78 \cdot 10^{-9}$	$6.95 \cdot 10^{-7}$

where $L_n(x)$ is the Legendre polynomial of degree n, $x_0 = -1$, $x_n = 1$ and x_j, $j = 1(1)n-1$, are the zeros of $L'_n(x)$.

The second derivative matrix can also be given explicitly (see [10]), but for the direct method we use here the relation $\boldsymbol{D}^{(2)} = (\boldsymbol{D}^{(1)})^2$.

In Table 1 the results for the first two derivatives of the function $u(x) = \sin(x)$ are listed.

We see in Table 1 that the error committed in calculating the derivative is close to the maximum value of the sum of the elements of the row (M). For $n = 512$ we gain seven powers of ten in the approximation of the first derivative of u and six powers for the approximation of the second derivative with methods (2) and (3).

For v and $n = 16$ and 32, we see in Table 2 that the results are almost the same with all techniques: The discretization error dominates the calculation error. For larger n, the algorithm and the formulas of Schneider and Werner give the best results. As in Table 1, we gain almost seven powers of ten for the first derivative and almost seven powers for the second derivative and $n = 512$.

In this case, we can even calculate the approximate derivatives of a function in n^2 operations (instead of $2n^2$) by using Solomonoff's algorithm [9]. This algorithm uses the anticentrosymmetry property $D^{(1)}_{jk} = -D^{(1)}_{n-j,n-k}$ of the differentiation matrix. It has been used by Don and Solomonoff in [5] for calculating the differentiation matrix for Čebyšev–Gauss–Lobatto points.

3. Computational examples for Čebyšev–Gauss–Radau points

The first order differentiation matrix for Čebyšev–Gauss–Radau points can be given explicitly, see [10]:

$$D^{(1)}_{jk} = \begin{cases} \dfrac{(n+1)n}{3}, & \text{if } j = k = 0, \\ -\dfrac{1}{2(1-x_j^2)}, & \text{if } j = k \neq 0, \\ \dfrac{c_j}{c_k}\dfrac{1}{x_j - x_k}, & \text{if } j \neq k. \end{cases} \tag{11}$$

Here, the collocation points are given by

$$x_j := \cos\left(\frac{2j\pi}{2n+1}\right), \quad j = 0(1)n,$$

and

$$c_0 = 2, \qquad c_j = \sqrt{\frac{2}{1+x_j}}, \quad j = 1(1)n.$$

Table 3
Errors in approximating the first (top) and second (bottom) derivatives of $u(x)$ using the three different techniques

n	16	32	64	128	256	512
(1)	$7.27 \cdot 10^{-14}$	$6.35 \cdot 10^{-12}$	$5.82 \cdot 10^{-11}$	$1.20 \cdot 10^{-10}$	$2.42 \cdot 10^{-8}$	$1.70 \cdot 10^{-7}$
(M)	$9.95 \cdot 10^{-14}$	$7.45 \cdot 10^{-12}$	$6.91 \cdot 10^{-11}$	$1.44 \cdot 10^{-10}$	$2.88 \cdot 10^{-8}$	$2.02 \cdot 10^{-7}$
(2)	$2.08 \cdot 10^{-14}$	$6.00 \cdot 10^{-14}$	$7.42 \cdot 10^{-14}$	$2.94 \cdot 10^{-13}$	$1.14 \cdot 10^{-12}$	$7.03 \cdot 10^{-12}$
(3)	$9.10 \cdot 10^{-15}$	$1.29 \cdot 10^{-14}$	$2.37 \cdot 10^{-13}$	$4.06 \cdot 10^{-13}$	$3.04 \cdot 10^{-12}$	$1.34 \cdot 10^{-11}$
(M)	$3.94 \cdot 10^{-15}$	$9.68 \cdot 10^{-15}$	$8.26 \cdot 10^{-14}$	$1.14 \cdot 10^{-13}$	$2.32 \cdot 10^{-12}$	$5.04 \cdot 10^{-12}$
(1)	$5.68 \cdot 10^{-12}$	$1.90 \cdot 10^{-9}$	$6.72 \cdot 10^{-8}$	$7.38 \cdot 10^{-7}$	$4.27 \cdot 10^{-4}$	$1.47 \cdot 10^{-2}$
(M)	$7.44 \cdot 10^{-12}$	$2.26 \cdot 10^{-9}$	$7.97 \cdot 10^{-8}$	$8.79 \cdot 10^{-7}$	$5.05 \cdot 10^{-4}$	$1.74 \cdot 10^{-2}$
(2)	$2.12 \cdot 10^{-12}$	$1.75 \cdot 10^{-11}$	$1.93 \cdot 10^{-10}$	$3.71 \cdot 10^{-9}$	$2.79 \cdot 10^{-8}$	$7.37 \cdot 10^{-7}$
(3)	$1.88 \cdot 10^{-12}$	$9.40 \cdot 10^{-12}$	$5.20 \cdot 10^{-10}$	$3.70 \cdot 10^{-9}$	$6.02 \cdot 10^{-8}$	$8.41 \cdot 10^{-7}$
(M)	$4.33 \cdot 10^{-13}$	$3.51 \cdot 10^{-12}$	$6.23 \cdot 10^{-11}$	$1.22 \cdot 10^{-9}$	$1.49 \cdot 10^{-8}$	$5.71 \cdot 10^{-7}$

Table 4
Errors in approximating the first (top) and second (bottom) derivatives of $v(x)$ using the three different techniques

n	16	32	64	128	256	512
(1)	$5.38 \cdot 10^{-5}$	$1.53 \cdot 10^{-10}$	$3.47 \cdot 10^{-11}$	$7.12 \cdot 10^{-11}$	$1.44 \cdot 10^{-8}$	$1.01 \cdot 10^{-7}$
(2)	$5.38 \cdot 10^{-5}$	$1.57 \cdot 10^{-10}$	$1.12 \cdot 10^{-13}$	$3.02 \cdot 10^{-13}$	$1.07 \cdot 10^{-12}$	$6.71 \cdot 10^{-12}$
(3)	$5.38 \cdot 10^{-5}$	$1.57 \cdot 10^{-10}$	$1.93 \cdot 10^{-13}$	$6.46 \cdot 10^{-13}$	$1.76 \cdot 10^{-12}$	$7.74 \cdot 10^{-12}$
(1)	$6.03 \cdot 10^{-3}$	$6.63 \cdot 10^{-8}$	$3.98 \cdot 10^{-8}$	$4.41 \cdot 10^{-7}$	$2.52 \cdot 10^{-4}$	$8.71 \cdot 10^{-3}$
(2)	$6.03 \cdot 10^{-3}$	$6.75 \cdot 10^{-8}$	$1.16 \cdot 10^{-10}$	$1.23 \cdot 10^{-9}$	$7.97 \cdot 10^{-9}$	$4.46 \cdot 10^{-7}$
(3)	$6.03 \cdot 10^{-3}$	$6.74 \cdot 10^{-8}$	$8.37 \cdot 10^{-11}$	$9.24 \cdot 10^{-10}$	$1.75 \cdot 10^{-8}$	$6.00 \cdot 10^{-7}$

We have again computed approximate derivatives of the functions $u(x)$ and $v(x)$. The results are listed in Tables 3 and 4.

As in Table 1, we see in Table 3 that the error grows at the same rate as the maximum value of the sum of the elements of the row. For the approximation of the first and second derivatives, we gain five powers of ten with $n = 512$. For the same reason as in Table 2, the first two columns of Table 4 are almost the same and, again, the algorithm and the formulas of Schneider and Werner yield the best results for larger n.

Unfortunately, here the differentiation matrix is not anticentrosymmetric and Solomonoff's algorithm cannot be used.

4. Conclusion

We have displayed the errors incurred when calculating the pseudospectral differentiation matrices for different kinds of points and we have suggested ways of alleviating these errors. This leads to the following recommendations: To compute the derivative of a function one should use the algorithm of Schneider and Werner. On the other hand, the formulas (8), (9) and the relation (6) is to be preferred when computing the elements of the differentiation matrices which are necessary for solving time evolution problems with the method of lines or time independent problems with iterative methods.

Acknowledgements

The author is indebted to J.-P. Berrut for helpful comments on this work.

References

[1] R. Baltensperger, J.-P. Berrut, The errors in calculating the pseudospectral differentiation matrices for Čebyšev–Gauss–Lobatto points, Comput. Math. Appl. 37 (1999) 41–48. Errata: 38 (1999) 119.

[2] A. Bayliss, A. Class, B. Matkowsky, Roundoff error in computing derivatives using the Chebyshev differentiation matrix, J. Comput. Phys. 116 (1994) 380–383.

[3] K.S. Breuer, R.M. Everson, On the errors incurred calculating derivatives using Chebyshev polynomials, J. Comput. Phys. 99 (1992) 56–67.
[4] P. Canuto, M.Y. Hussaini, A. Quarteroni, T. Zang, Spectral Methods in Fluid Dynamics, Springer, New York, 1988.
[5] W.S. Don, A. Solomonoff, Accuracy and speed in computing the Chebyshev collocation derivative, SIAM J. Sci. Comput. 16 (1995) 1253–1268.
[6] P. Henrici, Essentials of Numerical Analysis, Wiley, New York, 1982.
[7] E.E. Rothman, Reducing round-off error in Chebyshev pseudospectral computations, in: M. Durand and F. El Dabaghi (Eds.), High Performance Computing II, Elsevier, North-Holland, 1991, pp. 423–439.
[8] C. Schneider, W. Werner, Some new aspects of rational interpolation, Math. Comp. 47 (1986) 285–299.
[9] A. Solomonoff, A fast algorithm for spectral differentiation, J. Comput. Phys. 98 (1992) 174–177.
[10] B.D. Welfert, A remark on pseudospectral differentiation matrices, Department of Mathematics, Arizona State University, Tempe, 1992.