

Kent Academic Repository

Full text document (pdf)

Citation for published version

Xu, Kuan and Hale, Nicholas (2015) Explicit construction of rectangular differentiation matrices. IMA Journal of Numerical Analysis, 36 (2). pp. 618-632. ISSN 0272-4979.

DOI

<http://doi.org/10.1093/imanum/drv013>

Link to record in KAR

<http://kar.kent.ac.uk/58496/>

Document Version

UNSPECIFIED

Copyright & reuse

Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research

The version in the Kent Academic Repository may differ from the final published version.

Users are advised to check <http://kar.kent.ac.uk> for the status of the paper. **Users should always cite the published version of record.**

Enquiries

For any further enquiries regarding the licence status of this document, please contact:

researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at <http://kar.kent.ac.uk/contact.html>

Explicit construction of rectangular differentiation matrices

KUAN XU

Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK
kuan.xu@maths.ox.ac.uk

AND

NICHOLAS HALE*

Department of Mathematical Sciences, University of Stellenbosch, Stellenbosch, South Africa

*Corresponding author: nickhale@sun.ac.za

[Received on 22 August 2014; revised on 16 January 2015]

Spectral discretizations based on rectangular differentiation matrices have recently been demonstrated to be a convenient means of solving linear and nonlinear ordinary differential equations with general boundary conditions and other side constraints. Here, we present explicit formulae for such matrices.

Keywords: Chebyshev; spectral method; collocation; differentiation matrix.

1. Introduction

In a recent paper Driscoll and Hale introduce *rectangular differentiation matrices*, which they demonstrate to be a novel and convenient approach for implementing boundary conditions in Chebyshev spectral collocation (Driscoll & Hale, 2015). The standard approach in Chebyshev spectral methods for many years has been to form square $n \times n$ systems, and in, say, the case of two-point boundary value problems (TBVPs), replace the top and bottom rows with others which enforce the required boundary constraints (see, for example, Trefethen, 2000, p. 135; Boyd, 2001, Section 6.4). Whilst for TBVPs this is clear and unambiguous, the row replacement or ‘boundary bordering’ strategy quickly becomes *ad hoc* when applied to more complicated situations, such as third- or higher-order problems or integral-type constraints. From the rectangular point of view, a p th-order differential operator is naturally discretized by an $n \times (n + p)$ matrix, allowing p boundary constraints to be appended to form an invertible $(n + p) \times (n + p)$ system.¹

Driscoll and Hale define the first-order rectangular differentiation matrix, which in this paper we denote by $D_{n,n+1} \in \mathbb{R}^{n \times (n+1)}$, as that which maps function values of a polynomial on an $(n + 1)$ -point Chebyshev grid of the second kind to the derivative of that polynomial on an n -point Chebyshev grid of the first kind. Since the traditional Chebyshev differentiation matrix, D_{n+1} , maps function values of a polynomial on a second-kind Chebyshev grid to the derivative of the polynomial on that same grid (Trefethen, 2000, Chapter 6), they achieve this *downsampling* by simply pre-multiplying by the rectangular matrix which interpolates between the $(n + 1)$ - and n -point grids. They call this matrix the *downsampling matrix*, which we shall denote by $P_{n,n+1}$, and show that it can be efficiently computed using

¹ The reason for considering an $n \times (n + p)$ rectangular matrix rather than an $(n - p) \times n$ one is that the former is more convenient when extending to coupled systems of equations. See (Driscoll & Hale, 2015, Section 4.4) for further details.

the barycentric interpolation formula (Berrut & Trefethen, 2004). Thus, the rectangular differentiation matrix is constructed via $D_{n,n+1} = P_{n,n+1}D_{n+1}$, and the idea extends naturally to higher-order derivatives so that $D_{n,n+p}^{(p)} = P_{n,n+p}D_{n+p}^{(p)}$. The combined use of first- and second-kind Chebyshev grids has also been exploited by Kopriva & Koliás (1996) and Kopriva (1998).

The main result of this paper is to show how $D_{m,n}$ can be constructed explicitly, that is, without first forming D_n and downsampling, for any m and n . We also derive a simple recurrence for computing $D_{m,n}^{(p)}$. The explicit construction has the advantage that it requires $\mathcal{O}(n^2)$ floating point operations, rather than $\mathcal{O}(n^3)$ for the matrix–matrix multiplication (although this can be reduced by applying $P_{m,n}$ using fast Fourier transforms (FFTs)—see Section 6), and may also be useful in providing more theoretical insight into rectangular spectral collocation.

The outline is as follows. In Section 2 we give some preliminaries before deriving the explicit construction of the first-order rectangular differentiation matrix in Section 3. In Section 4 we show that recurrence relations similar to those used in the standard square case can be used to extend the construction to higher-order derivatives, and in Section 5 we make some remarks regarding accurate implementation for both the first- and higher-order cases. Section 6 introduces some alternative constructions of rectangular differentiation matrices, and Section 7 compares these numerically with the explicit construction.

2. Preliminaries

In this paper we denote the Chebyshev points of the first kind (sometimes known as Chebyshev ‘nodes’ or ‘roots’, or more formally as ‘Gauss–Chebyshev points’) by

$$\tau_{k,n} = \cos \frac{(2k+1)\pi}{2n}, \quad k = 0, \dots, n-1, \quad (2.1)$$

and the Chebyshev points of the second kind (referred to as Chebyshev ‘extrema’ or more formally ‘Gauss–Chebyshev–Lobatto points’) by

$$t_{k,n} = \cos \frac{k\pi}{n-1}, \quad k = 0, \dots, n-1. \quad (2.2)$$

These points are, respectively, the roots and extrema of $T_n(x)$ and $T_{n-1}(x)$, where $T_n(x) = \cos(n \arccos x)$ is the degree- n Chebyshev polynomial of the first kind. The second-kind points (2.2) are also the roots of $(1-x^2)U_{n-2}(x)$, where $U_n(x) = \sin((n+1) \arccos x) / \sin(\arccos x)$ is the degree- n Chebyshev polynomial of the second kind. For convenience, we write $\underline{\tau}_n = \{\tau_{k,n}\}_{k=0}^{n-1}$ and $\underline{t}_n = \{t_{k,n}\}_{k=0}^{n-1}$.

Defining the nodal polynomial $\ell(x)$ for an arbitrary set of n points $\underline{x}_n = \{x_k\}_{k=0}^{n-1}$ by

$$\ell(x) = \prod_{k=0}^{n-1} (x - x_k), \quad (2.3)$$

the Lagrange polynomials may be written as

$$\ell_j(x) = \frac{\ell(x)}{\ell'(x_j)(x - x_j)}, \quad j = 0, \dots, n-1, \quad (2.4)$$

so that $\ell_j(x)$ is the unique polynomial of degree $n - 1$ satisfying

$$\ell_j(x_k) = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases} \tag{2.5}$$

LEMMA 2.1 If the nodes x_n are the Chebyshev points of the second kind, t_n , then

$$\ell(x) = 2^{-n+2}(x^2 - 1)U_{n-2}(x) = 2^{-n+1}(T_n(x) - T_{n-2}(x)). \tag{2.6}$$

Proof. The first equality follows from the observation that $\ell(x)$ and $2^{-n+2}(x^2 - 1)U_{n-2}(x)$ are both monic polynomials of degree n with the same zeros. The second follows directly from [Olver et al. \(2010, \(18.9.10\)\)](#). \square

LEMMA 2.2 ([Trefethen, 2013, p. 40](#)) If $\ell(x)$ is the nodal polynomial corresponding to the points t_n , then

$$\ell'(t_{j,n}) = 2^{-n+1}(T'_n(t_{j,n}) - T'_{n-2}(t_{j,n})) = \begin{cases} 2^{-n+2}(n-1)(-1)^j, & 1 \leq j \leq n-2, \\ 2^{-n+3}(n-1)(-1)^j, & j = 0, n-1. \end{cases} \tag{2.7}$$

Proof. By definition of $T_n(x)$ and $T_{n-2}(x)$,

$$T'_n(t_{j,n}) - T'_{n-2}(t_{j,n}) = \frac{2 \sin(n-1)\theta_j \cos \theta_j + 2(n-1) \sin \theta_j \cos(n-1)\theta_j}{\sin \theta_j}, \quad j = 0, 1, \dots, n-1, \tag{2.8}$$

where $\theta_j = j\pi/(n-1)$. Simplification of this expression, along with the second equality in (2.6), leads to (2.7). \square

LEMMA 2.3 If $t_{j,n}$ is a Chebyshev point of the second kind, then

$$T_n(t_{j,n}) = (-1)^j t_{j,n} = T_{n-2}(t_{j,n}). \tag{2.9}$$

Proof. The first equality follows immediately from expanding $T_n(t_{j,n}) = \cos(j\pi + j\pi/(n-1))$ using the standard cosine double angle formula. The second equality is essentially the same. \square

LEMMA 2.4 If $\ell(x)$ is the nodal polynomial corresponding to the points t_n , then

$$\ell''(t_{j,n}) = 2^{-n+2}(-1)^{j+1}(n-1) \frac{t_{j,n}}{1-t_{j,n}^2}, \quad 1 \leq j \leq n-2. \tag{2.10}$$

Proof. By the second equality in (2.6) we have

$$\ell''(t_{j,n}) = 2^{-n+1}(T''_n(t_{j,n}) - T''_{n-2}(t_{j,n})). \tag{2.11}$$

Using the differential equations satisfied by $T_n(x)$ and $T_{n-2}(x)$ ([Olver et al., 2010, \(18.8.1\)](#)), we find

$$\ell''(t_{j,n}) = 2^{-n+1} \left(\frac{t_{j,n}T'_n(t_{j,n}) - n^2T_n(t_{j,n})}{1-t_{j,n}^2} - \frac{t_{j,n}T'_{n-2}(t_{j,n}) - (n-2)^2T_{n-2}(t_{j,n})}{1-t_{j,n}^2} \right) \tag{2.12}$$

$$= \frac{2^{-n+1}}{1-t_{j,n}^2} (t_{j,n}(T'_n(t_{j,n}) - T'_{n-2}(t_{j,n})) - n^2T_n(t_{j,n}) + (n-2)^2T_{n-2}(t_{j,n})). \tag{2.13}$$

Using (2.7) to remove the $T'_n(t_{j,n})$ and $T'_{n-2}(t_{j,n})$ terms, substituting (2.9) for the $T_n(t_{j,n})$ and $T_{n-2}(t_{j,n})$ and then rearranging gives the required result. \square

LEMMA 2.5 For any $x \in [-1, 1]$,

$$\sum_{k=0}^{n-1} \frac{1}{x - t_{k,n}} = \frac{U'_{n-2}(x)}{U_{n-2}(x)} - \frac{2x}{1 - x^2}. \quad (2.14)$$

Proof. By Lemma 2.1,

$$U_{n-2}(x) = 2^{n-2} \frac{\ell(x)}{x^2 - 1} = 2^{n-2} \prod_{k=1}^{n-2} (x - t_{k,n}). \quad (2.15)$$

Taking the derivatives of both sides we find

$$U'_{n-2}(x) = 2^{n-2} \prod_{j=1}^{n-2} (x - t_{j,n}) \sum_{k=1}^{n-2} \frac{1}{x - t_{k,n}} = U_{n-2}(x) \sum_{k=1}^{n-2} \frac{1}{x - t_{k,n}}, \quad (2.16)$$

from which we have

$$\sum_{k=1}^{n-2} \frac{1}{x - t_{k,n}} = \frac{U'_{n-2}(x)}{U_{n-2}(x)}. \quad (2.17)$$

Together with the terms corresponding to $t_{0,n} = 1$ and $t_{n-1,n} = -1$, this amounts to (2.14). \square

3. Rectangular differentiation matrix

Suppose a function $f(x)$ is approximated by the Lagrange interpolant of the n sampled values $\{(x_j, f_j)\}_{j=0}^{n-1}$. The Lagrange interpolant of f is

$$p(x) = \sum_{j=0}^{n-1} f_j \ell_j(x), \quad (3.1)$$

and the derivative of $f(x)$ can be approximated by $p'(x)$:

$$p'(x) = \sum_{j=0}^{n-1} f_j \ell'_j(x). \quad (3.2)$$

Denoting the column vector consisting of f_j by \mathbf{f} , the vector of values $v_i = p'(y_i)$ at another set of points $\mathbf{y}_m = \{y_i\}_{i=0}^{m-1}$ can be computed by

$$\mathbf{v} = D_{m,n} \mathbf{f}, \quad (3.3)$$

where $D_{m,n}$ is an $m \times n$ differentiation matrix with (i, j) entry

$$[D_{m,n}]_{ij} = \ell'_j(y_i). \quad (3.4)$$

This idea forms the basis of spectral collocation methods (see, for example, Trefethen, 2000, Chapter 6; Boyd, 2001, Chapter 4).

LEMMA 3.1 The derivative of the j th Lagrange polynomial corresponding to the nodes x_n may be written as

$$\ell'_j(x) = \ell_j(x) \sum_{\substack{k=0 \\ k \neq j}}^{n-1} \frac{1}{x - x_k}. \tag{3.5}$$

Proof. Rewriting (2.4) as $\ell'(x_j)\ell_j(x)(x - x_j) = \ell(x)$, and then taking the logarithm and differentiating yields

$$\frac{\ell'_j(x)}{\ell_j(x)} + \frac{1}{x - x_j} = \frac{\ell'(x)}{\ell(x)}.$$

Noting that

$$\frac{\ell'(x)}{\ell(x)} = \sum_{k=0}^{n-1} \frac{1}{x - x_k}, \tag{3.6}$$

we obtain (3.5). □

COROLLARY 3.2 If the nodes x_n in Lemma 3.1 are the second-kind Chebyshev grid t_n , then

$$\ell'_j(x) = \ell_j(x) \left(\frac{U'_{n-2}(x)}{U_{n-2}(x)} - \frac{2x}{1 - x^2} - \frac{1}{x - t_{j,n}} \right). \tag{3.7}$$

Proof. Simply replace x_k by $t_{k,n}$ in (3.5) and replace the summation with the expression from (2.14). □

THEOREM 3.3 Denote by $D_{m,n}$ the $m \times n$ rectangular differentiation matrix which maps from the n -point Chebyshev grid of the second kind t_n to the m -point Chebyshev grid of the first kind τ_m . If $m < n$ and $\tau_{i,m} \neq t_{j,n}$, then the entries of $D_{m,n}$ are given by

$$[D_{m,n}]''_{ij} = \frac{(-1)^j T_{n-1}(\tau_{i,m})}{\tau_{i,m} - t_{j,n}} + \frac{(-1)^j U_{n-2}(\tau_{i,m})(1 - \tau_{i,m}t_{j,n})}{(n - 1)(\tau_{i,m} - t_{j,n})^2}, \tag{3.8}$$

where the double prime indicates that the first and last columns are halved. If $\tau_{i,m}$ and $t_{j,n}$ coincide, then

$$[D_{m,n}]_{ij} = -\frac{\tau_{i,m}}{2(1 - \tau_{i,m}^2)}. \tag{3.9}$$

Proof. For $j = 1, \dots, n - 2$, substituting (2.6) from Lemma 2.1 and (2.7) from Lemma 2.2 into (2.4) gives

$$\ell_j(\tau_{i,m}) = \frac{\ell(\tau_{i,m})}{\ell'(t_{j,n})(\tau_{i,m} - t_{j,n})} \tag{3.10}$$

$$= \frac{2^{-n+2}(\tau_{i,m}^2 - 1)U_{n-2}(\tau_{i,m})}{2^{-n+2}(n - 1)(-1)^j(\tau_{i,m} - t_{j,n})} \tag{3.11}$$

$$= \frac{(-1)^{j+1}(1 - \tau_{i,m}^2)U_{n-2}(\tau_{i,m})}{(n - 1)(\tau_{i,m} - t_{j,n})}, \tag{3.12}$$

with half of this value for $j = 0$ and $j = n - 1$. Setting $x = \tau_{i,m}$ in (3.7) and substituting (3.12) leads to

$$[D_{m,n}]_{ij} = \ell'_j(\tau_{i,m}) = \frac{(-1)^{j+1}(1 - \tau_{i,m}^2)U_{n-2}(\tau_{i,m})}{(n-1)(\tau_{i,m} - t_{j,n})} \left(\frac{U'_{n-2}(\tau_{i,m})}{U_{n-2}(\tau_{i,m})} - \frac{2\tau_{i,m}}{1 - \tau_{i,m}^2} - \frac{1}{\tau_{i,m} - t_{j,n}} \right), \quad (3.13)$$

for $j = 1, \dots, n - 2$, again with the corresponding values for $j = 0$ and $j = n - 1$ being halved. Substituting the relation

$$U'_{n-2}(\tau_{i,m}) = \frac{-(n-1)T_{n-1}(\tau_{i,m}) + \tau_{i,m}U_{n-2}(\tau_{i,m})}{1 - \tau_{i,m}^2} \quad (3.14)$$

into (3.13) and rearranging yields (3.8).

For (3.9), observe that differentiating (2.4) and applying l'Hôpital's rule to evaluate $\ell'_j(x_j)$ leads to $\ell'_j(x_j) = \ell''(x_j)/(2\ell'(x_j))$. Setting $x_j = t_{j,n}$ and substituting (2.7) and (2.10) gives the required result. \square

COROLLARY 3.4 When $m = n - 1$, (3.8) simplifies to²

$$[D_{n-1,n}]''_{ij} = \frac{(-1)^{i+j}(1 - \tau_{i,n-1}t_{j,n})}{(n-1)(1 - \tau_{i,n-1}^2)^{1/2}(\tau_{i,n-1} - t_{j,n})^2}. \quad (3.15)$$

Proof. When $m = n - 1$, the first term in (3.8) vanishes as τ_m are the roots of T_{n-1} . For the second term, we observe that

$$U_{n-2}(\tau_{i,n-1}) = \frac{\sin((2i+1)\pi/2)}{\sin((2i+1)\pi/(2n-2))} = \frac{(-1)^i}{(1 - \tau_{i,n-1}^2)^{1/2}}. \quad (3.16)$$

\square

Analogous results for rectangular differentiation matrices which map from function values on an n -point first-kind Chebyshev grid to an m -point grid also of the first kind can be found in the Appendix. See Section 5 for a discussion of how to accurately evaluate the expressions in (3.3) and (3.15).

4. Recursion for high-order differentiation matrices

Welfert (1997) derives an expression for the entries of a $(p+1)$ th-order square differentiation matrix in terms of its p th-order counterpart, and based on this gives a recursive algorithm to compute higher-order square differentiation matrices in $\mathcal{O}(n^2)$ operations. Following the same approach we derive a similar recursive method for high-order rectangular differentiation matrices.

THEOREM 4.1 (Welfert, 1997, Theorem 2.1) For an arbitrary grid x_n , the p th and $(p+1)$ th derivatives of the Lagrange polynomial $\ell_j(x)$ are related by

$$\ell^{(p+1)}(x) = \ell'(x_j)((x - x_j)\ell_j^{(p+1)}(x) + (p+1)\ell_j^{(p)}(x)), \quad j = 0, \dots, n-1, \quad (4.1)$$

where $\ell(x)$ is the nodal polynomial defined by (2.3).

² The corresponding formula for (3.9) when $m = n - 1$ is not required as an $(n-1)$ -point first-kind grid and an n -point second-kind grid never have a point in common.

Proof. Another form of the nodal polynomial can be obtained by writing (2.4) as

$$\ell(x) = \ell_j(x)\ell'(x_j)(x - x_j). \tag{4.2}$$

Differentiating both sides of the above by the Leibniz rule, (4.1) follows immediately. □

THEOREM 4.2 Denote by $D_{m,n}^{(p)}$ the $m \times n$ rectangular differentiation matrix of order p which maps from an n -point Chebyshev grid of the second kind t_n to an m -point Chebyshev grid of the first kind τ_m . If $\tau_{i,m} \neq t_{j,n}$, the entries of $D_{m,n}^{(p+1)}$ are given by

$$[D_{m,n}^{(p+1)}]_{ij}'' = \frac{1}{\tau_{i,m} - t_{j,n}} \left[\frac{(-1)^j}{2(n-1)} (T_n^{(p+1)}(\tau_{i,m}) - T_{n-2}^{(p+1)}(\tau_{i,m})) - (p+1)[D_{m,n}^{(p)}]_{ij} \right], \tag{4.3}$$

where the double prime indicates that, for $j = 0$ and $j = n - 1$, the first term in the square brackets on the right-hand side is halved. If $\tau_{i,m}$ and $t_{j,n}$ coincide, the corresponding entry becomes

$$[D_{m,n}^{(p+1)}]_{ij} = (-1)^j \frac{T_n^{(p+2)}(\tau_{i,m}) - T_{n-2}^{(p+2)}(\tau_{i,m})}{2(n-1)(p+2)}. \tag{4.4}$$

Proof. Recalling that $\ell_j^{(p)}(\tau_{i,m}) = [D_{m,n}^{(p)}]_{ij}$, substituting this into (4.1) and rearranging, we find

$$[D_{m,n}^{(p+1)}]_{ij} = \frac{1}{\tau_{i,m} - t_{j,n}} \left(\frac{\ell^{(p+1)}(\tau_{i,m})}{\ell'(t_{j,n})} - (p+1)[D_{m,n}^{(p)}]_{ij} \right). \tag{4.5}$$

From (2.6) in Lemma 2.1 and (2.7) in Lemma 2.2 we have

$$\frac{\ell^{(p+1)}(\tau_{i,m})}{\ell'(t_{j,n})} = \frac{2^{-n+1}(T_n^{(p+1)}(\tau_{i,m}) - T_{n-2}^{(p+1)}(\tau_{i,m}))}{2^{-n+2}(n-1)(-1)^j}, \tag{4.6}$$

which when substituted into (4.5) gives (4.3).

For the case when $\tau_{i,m}$ and $t_{j,n}$ coincide we apply l'Hôpital's rule to (2.4) in a similar way as in the proof of Theorem 3.3, giving

$$[D_{m,n}^{(p+1)}]_{ij} = \frac{\ell^{(p+2)}(t_{j,n})}{(p+2)\ell'(t_{j,n})}. \tag{4.7}$$

Using Lemmas 2.1 and 2.2 in the same way as above, we obtain (4.4). □

5. Implementation details

Much as in the square case, there are a number of useful identities which can be used to ensure that the rectangular differentiation matrices are constructed accurately. In the literature these are often referred to as 'tricks', and we briefly review those which are relevant here.

5.1 Trigonometric identities

Following [Canuto \(1988\)](#) and [Tang & Trummer \(1996\)](#) we evaluate $\tau_{i,m} - t_{j,n}$ in (3.8) and (4.3) using the trigonometric identity

$$\tau_{i,m} - t_{j,n} = -2 \sin\left(\frac{\theta_i + \phi_j}{2}\right) \sin\left(\frac{\theta_i - \phi_j}{2}\right), \quad (5.1)$$

where $\theta_i = (2i + 1)\pi/(2m) = \arccos(\tau_{i,m})$ and $\phi_j = j\pi/(n - 1) = \arccos(t_{j,n})$. As noted by others, this trick reduces the cancellation error and offers improvements in accuracy in the computation of differentiation matrices. Unfortunately, a similar trick is less forthcoming for the $1 - \tau_{i,m}t_{j,n}$ term in the numerators of (3.8) and (3.15). However, in the corner entries of the rectangular matrix this term cancels with the $\tau_{i,m} - t_{j,n}$ discussed above. In particular, the first entry of (3.15) simplifies to

$$[D_{n-1,n}]_{0,0} = -\left(4(n-1) \sin\left(\frac{\pi}{2(n-1)}\right) \sin^2\left(\frac{\pi}{4(n-1)}\right)\right)^{-1}, \quad (5.2)$$

which can be accurately evaluated. If required, the $1 - \tau_{i,m}t_{j,n}$ term can be accurately evaluated for $\tau_{i,m}t_{j,n} \approx 1$ by using a Taylor series expansion in m and n .

Additionally, terms of the form $(1 - \tau_{i,n})^{1/2}$, such as in (3.15), should be evaluated as $\sin((2i + 1)\pi/(2n))$ to avoid rounding error near ± 1 .

5.2 The ‘flipping’ trick

As pointed out by [Solomonoff \(1992\)](#) and [Don & Solomonoff \(1995\)](#), the $n \times n$ square differentiation matrix has the skew-symmetry property

$$[D_n]_{ij} = -[D_n]_{n-i-1, n-j-1}. \quad (5.3)$$

Since values of θ_i and ϕ_j nearer zero can be computed to a higher relative accuracy, this can lead to improved accuracy ([Baltensperger & Trummer, 2003](#)). The $m \times n$ rectangular differentiation matrices have an analogous property.

THEOREM 5.1 The rectangular differentiation matrices (3.8) and (4.3) satisfy the skew-symmetry properties

$$[D_{m,n}]_{ij} = -[D_{m,n}]_{m-i-1, n-j-1} \quad (5.4)$$

and

$$[D_{m,n}^{(p)}]_{ij} = (-1)^p [D_{m,n}^{(p)}]_{m-i-1, n-j-1}. \quad (5.5)$$

Proof. It suffices to note that $\tau_{m-i-1, m} = -\tau_{i, m}$, $t_{n-j-1, n} = -t_{j, n}$, $T_{n-1}(x) = (-1)^{n-1}T_{n-1}(-x)$ and $U_{n-2}(x) = (-1)^{n-2}U_{n-2}(-x)$. Substituting these expressions into (3.8) for $[D_{m,n}]_{m-i-1, n-j-1}$ readily verifies (5.4).

For (5.5) we can apply induction using (4.3), noting that differentiating $T_n(x)$, a function of definite parity, changes the parity if the differentiation is performed an odd number of times, while leaving it unchanged if performed an even number of times (see, for example, [Boyd, 2001](#), Theorem 25). \square

5.3 The ‘negative-sum’ trick

It has been standard practice to improve the accuracy in computing the square differentiation matrix by employing the ‘negative-sum trick’ (Bayliss *et al.*, 1995; Baltensperger & Berrut, 1999; Baltensperger & Trummer, 2003). That is, for the usual square differentiation matrix, the diagonal entry in each row is computed as the negated sum of all off-diagonal entries. This idea follows from the fact that constant functions should evaluate to zero when differentiated.

Constant functions must evaluate to zero when differentiated using rectangular matrices too, so we may employ a similar idea. For the rectangular differentiation matrix $D_{m,n}$ in (3.8), it is the entry in row i corresponding to the smallest value of $|\tau_{i,m} - t_{j,n}|$ which suffers most from cancellation error. Therefore, we assign it the negated sum of all other entries in the same row, i.e.,

$$[D_{m,n}]_{il} = \sum_{\substack{j=0 \\ j \neq l}}^{n-1} [D_{m,n}]_{ij}, \quad \text{where } |\tau_{i,m} - t_{l,n}| = \min_{j=0, \dots, n-1} |\tau_{i,m} - t_{j,n}|. \quad (5.6)$$

The exception is in the first and last row, where the corner entries on the diagonal can be accurately evaluated by (5.2). In this case we use the negative-sum trick to improve the value of the neighbouring entry.

The negative-sum trick can also be applied to the higher-order differentiation matrices $D_{m,n}^{(p)}$ in (4.3).

6. Other approaches

In this short section we outline other approaches for constructing rectangular differentiation matrices, and comment briefly on how they compare with the explicit constructions derived in Sections 3 and 4. Numerical comparisons are given in Section 7.

6.1 Downsampling via interpolation

In Driscoll & Hale (2015) the $m \times n$ rectangular differentiation matrix $D_{m,n}$ is computed by first forming the standard $n \times n$ square differentiation matrix D_n and then pre-multiplying by a *downsampling matrix* $P_{m,n}$.³ That is, $D_{m,n} = P_{m,n}D_n$, where $P_{m,n}$ is a rectangular matrix which interpolates function values from an n -point grid to an m -point grid. Barycentric interpolation (Berrut & Trefethen, 2004) is a natural choice for this operation, and the matrix $P_{m,n}$ mapping between $\{y_i\}_{i=0}^{m-1}$ and $\{x_j\}_{j=0}^{n-1}$ can be explicitly written as

$$[P_{m,n}]_{ij} = \begin{cases} \frac{w_k}{y_i - x_j} \left(\sum_{k=0}^{n-1} \frac{w_k}{y_i - x_k} \right)^{-1}, & y_i \neq x_j, \\ 1, & y_i = x_j, \end{cases} \quad (6.1)$$

where the w_k are the barycentric weights corresponding to the points $\{x_j\}_{j=0}^{n-1}$. Basic MATLAB code for computing $P_{m,n}$ can be found in Driscoll & Hale (2015, Fig. 3.1), but this can be improved upon by applying the trigonometric and flipping tricks from Section 5.

Though the costs for constructing $P_{m,n}$ and D_n are $\mathcal{O}(mn)$ and $\mathcal{O}(n^2)$ flops respectively, the multiplication of $P_{m,n}$ and D_n requires $\mathcal{O}(mn^2)$ flops, which dominates. However, the implied constant is small and the complexity is comparable to the $\mathcal{O}(n^3)$ cost of directly solving the linear system when,

³ In Driscoll & Hale (2015) the notation $P_{n,m-n}$ is used instead.

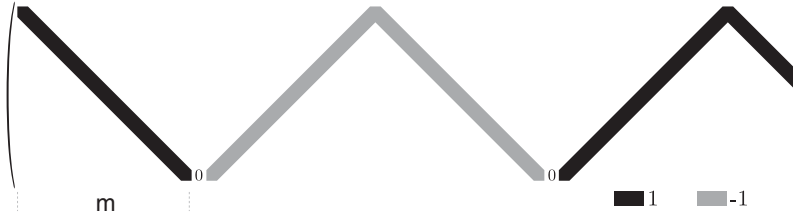


FIG. 1. The aliasing matrix, $A_{m,n} \in \mathbb{R}^{m \times n}$, which results from downsampling from an n -point Chebyshev grid of the second kind to an m -point Chebyshev grid of the first kind.

for example, solving a boundary value problem (BVP). As such, the additional overhead of constructing $D_{m,n}$ in this way is not usually significant. The additional motivation of this approach is that the discretized linear operators representing the BVP can be constructed in the traditional square sense, and the entire operator downsampled as a final step before applying boundary constraints.

6.2 Downsampling via aliasing

The downsampling described above is equivalent to a specific aliasing in the Chebyshev coefficients of the polynomial interpolant $p(x)$ from (3.1); see Driscoll & Hale (2015, Section 3.2). In particular, the $m \times n$ rectangular downsampling matrix $P_{m,n}$ can be constructed as the product of three matrices $P_{m,n} = V_m A_{m,n} C_n$, where $V_m \in \mathbb{R}^{m \times m}$ is the discrete cosine transform (DCT) of Type I which maps from Chebyshev coefficients to a first-kind Chebyshev grid, $C_n \in \mathbb{R}^{n \times n}$ is a scaled inverse DCT of Type I which maps from a second-kind Chebyshev grid to Chebyshev coefficients and $A_{m,n} \in \mathbb{R}^{m \times n}$ is the aliasing matrix

$$[A_{m,n}]_{ij} = \begin{cases} 1, & j = 4km \pm i, \\ -1, & j = 2(2k + 1)m \pm i, \\ 0, & \text{otherwise,} \end{cases} \quad (6.2)$$

where $i = 0, 1, \dots, m - 1$ and $k = 0, 1, \dots$ so that $0 \leq j \leq n - 1$. See Fig. 1 for a graphical illustration.

The connection to the DCT means that both V_m and C_n can be efficiently applied using FFTs in $\mathcal{O}(n^2 \log n)$ and $\mathcal{O}(mn \log m)$ flops, respectively. Taking advantage of its sparsity, multiplication by $A_{m,n}$ can be carried out in $\mathcal{O}(n^2)$ flops, making the total complexity $\mathcal{O}(n^2 \log n)$. Whilst this reduced complexity over the interpolation approach may not be significant when compared with solving the resulting linear system, the reduced number of operations may have a positive effect on the accumulation of rounding error.

In the usual case, where $(n - 1)/2 \leq m < n$, A takes the simpler form

$$[A_{m,n}]_{ij} = \begin{cases} 1, & j = i, \\ -1, & j = 2m - i, \\ 0, & \text{otherwise,} \end{cases} \quad (6.3)$$

where again $i = 0, 1, \dots, m - 1$. Furthermore, as pointed out in Driscoll & Hale (2015), if $m = n - 1$, then

$$[A_{n-1,n}]_{ij} = [I_n]_{ij}, \quad i = 0, \dots, n - 2, \quad j = 0, \dots, n - 1, \quad (6.4)$$

where I_n is the $n \times n$ identity matrix.

6.3 Constructing in coefficient space

In the previous section we started with the standard square Chebyshev spectral collocation differentiation matrix and moved to coefficient space using DCTs to achieve the downsampling via aliasing. An alternative construction can be achieved by forming the differentiation matrix in coefficient space and then using DCTs to map back to physical space. In particular, the required recurrence relation can be written in matrix form so that $\underline{\beta} = \mathcal{D}^{(p)} \underline{\alpha}$, where $f(x) = \sum_{k=0}^{n-1} \alpha_k T_k(x)$ and $f^{(p)}(x) = \sum_{k=0}^{n-1} \beta_k T_k(x)$. A convenient decomposition of $\mathcal{D}^{(1)}$ is given by the ultraspherical formulation (Olver & Townsend, 2013), $\mathcal{D}^{(1)} = \mathcal{S}_0^{-1} \mathcal{D}_0$, where

$$\mathcal{S}_0 = \begin{pmatrix} 1 & & & & & \\ & \frac{1}{2} & & & & \\ & & -\frac{1}{2} & & & \\ & & & -\frac{1}{2} & & \\ & & & & -\frac{1}{2} & \\ & & & & & \ddots \\ & & & & & & \ddots \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad \mathcal{D}_0 = \begin{pmatrix} 0 & 1 & & & & \\ & & 2 & & & \\ & & & 3 & & \\ & & & & \ddots & \\ & & & & & \ddots \end{pmatrix} \in \mathbb{R}^{n \times n}. \quad (6.5)$$

Hence, if $A_{m,n}$ is as in (6.2), the first-order rectangular differentiation matrix may be expressed as

$$D_{m,n} = V_m A_{m,n} \mathcal{S}_0^{-1} \mathcal{D}_0 C_n. \quad (6.6)$$

Higher-order derivatives can be expressed in a similar way, so that

$$D_{m,n}^{(p)} = V_m A_{m,n} (\mathcal{S}_{p-1} \dots \mathcal{S}_0)^{-1} \mathcal{D}_{p-1} C_n, \quad (6.7)$$

where $\mathcal{S}_0, \dots, \mathcal{S}_{p-1}$ and \mathcal{D}_{p-1} are defined in Olver & Townsend (2013, Section 3). The sparseness and structure of these matrices means they can be both applied and inverted efficiently.

We note also that $D_{m,n}^{(p)} = V_m A_{m,n} (\mathcal{S}_{p-1} \dots \mathcal{S}_0)^{-1} \mathcal{D}_{p-1}$ gives a convenient way of computing the values of the form $T_n^p(\tau_{i,m})$ appearing in Theorem 4.2.

7. Numerical results

In this section we test the accuracy of the explicit construction of the first-order rectangular differentiation matrices and compare with the other approaches outlined in the previous section. Results for higher-order derivatives are similar, and MATLAB code to reproduce these results can be found at Hale & Xu (2014). Implementations of the formulae can also be found in the `diffmat` function of Chebfun (Driscoll *et al.*, 2014, v5.1.0).

There are differing opinions in the literature as to whether one should test the accuracy of the matrix entries themselves (Breuer & Everson, 1992; Baltensperger & Trummer, 2003), or the action and inverse of the matrix when applied to a function (Bayliss *et al.*, 1995; Don & Solomonoff, 1995; Baltensperger & Berrut, 1999). Here we do both. We first construct $D_{n-1,n}$ in extended precision using the MATLAB symbolic toolbox and compute the elementwise maximum norm of the difference in the entries when constructing in double precision using the explicit construction from Section 3 and each of the approaches in Section 6. The left panel of Fig. 2 shows the absolute error and the right panel shows the elementwise relative error for a range of values of n . In the former case the standard square construction is included for reference, but this is omitted from the latter to avoid issues with zero entries.

We find that, when looking at the absolute error, the aliasing approach of Section 6.2 provides an accuracy comparable with the square case, with the barycentric and coefficient-based approaches

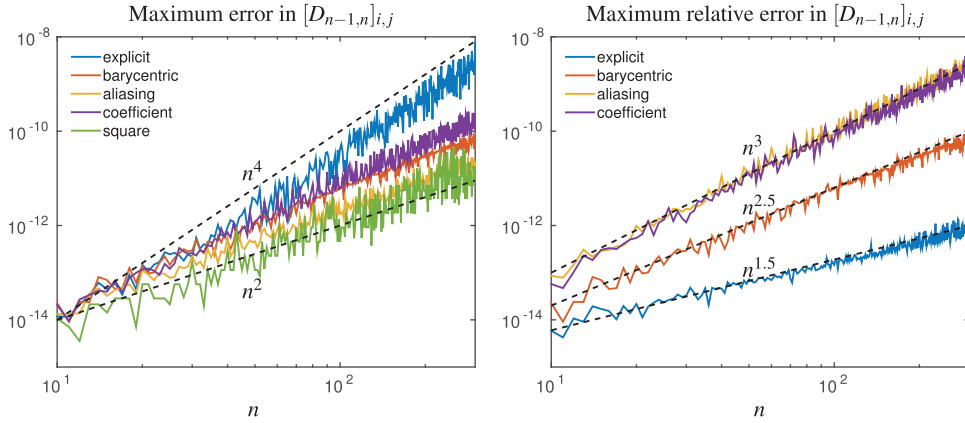


FIG. 2. Errors in $D_{n-1,n}$ for different methods, when compared with a construction using extended precision. Although the absolute error in the explicit construction (left) is worse than the other approaches outlined in Section 6, the relative error (right) is significantly improved.

being only slightly worse. The explicit construction performs poorly here, eventually growing at a rate proportional to n^4 , and this can be traced back to the difficulty of computing $(1 - \tau_i t_j)$ when τ_i and t_j are both close to ± 1 . It is worth noting that the coefficient-based approach obtains remarkable accuracy, almost full precision, along the matrix diagonal.

However, the situation is reversed when we look at the relative error. The explicit expression can be accurately evaluated away from the ‘diagonal’ (i.e., away from $\tau_i \approx t_j$), and in particular the small entries of $D_{n-1,n}$ near the boundaries of the matrix are evaluated to a good relative precision. Conversely, the DCT used in the aliasing and coefficient-based approaches means that the error is smeared out uniformly amongst all the matrix entries and a larger relative error is observed near the boundaries, leading to the observed $\mathcal{O}(n^3)$ growth in Fig. 2 (right). The barycentric approach, if applied using the tricks described in Section 5, performs somewhere between these two extremes.

For the second stage of testing we investigate the accuracy of the different rectangular differentiation matrix constructions when applied to a vector and when appended with a boundary condition and inverted. For simplicity, we consider the function $f(x) = \exp(x)$. We first consider the error $\|f'(\underline{\tau}_{n-1}) - D_{n-1,n}f(\underline{t}_n)\|_\infty$ for increasing values of n , and include $\|f'(\underline{t}_n) - D_n f(\underline{t}_n)\|_\infty$ for reference. In Fig. 3 (left) we see that error growth in each of the approaches follows the expected $\mathcal{O}(n^2)$ shown by the square case (Breuer & Everson, 1992).

The story is different in Fig. 3 (right), however, where we solve

$$u' = \exp(x), \quad x \in [-1, 1], \tag{7.1a}$$

$$u(-1) = \exp(-1), \tag{7.1b}$$

and compute the maximum norm error against the true solution, $u(x) = \exp(x)$, at resulting grid points. Here, although the error for each of the different constructions of each of the rectangular differentiation matrices grows roughly like $\mathcal{O}(n)$, the implied constant in the case of the aliasing and coefficient-based approaches is significantly worse than the barycentric and explicit approaches.

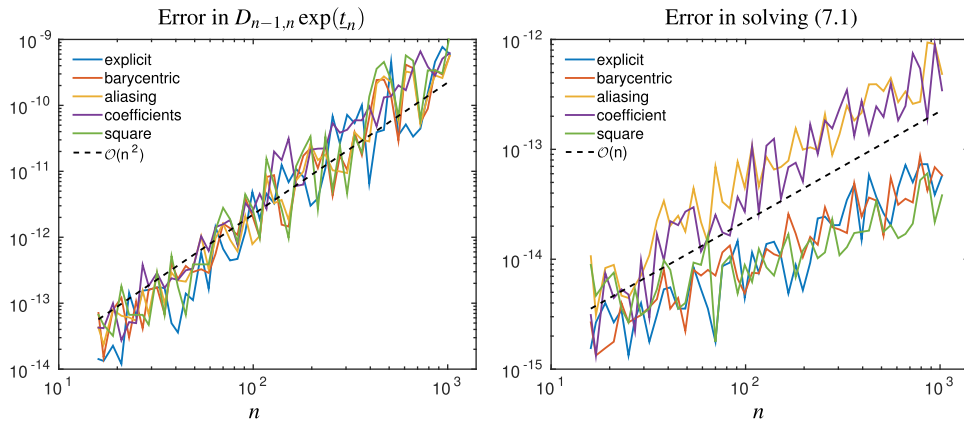


FIG. 3. Left: error in applying $D_{n-1,n}$ to $\exp(t_n)$. Right: error in solving (7.1). In both cases the maximum norm of the error on the resulting grid points is used. On the left we see little difference between the numerous approaches when $D_{n-1,n}$ is applied to a smooth function. However, on the right we see that when solving the BVP system the aliasing and coefficient-based approaches are less accurate.

8. Conclusion

We have shown how it is possible to derive explicit expressions for the entries in the rectangular differentiation matrix $D_{m,n}$, which maps from function values at an n -point Chebyshev grid of the second kind to derivative values on an m -point grid of the first kind. As well as being potentially useful for developing the theory underlying the rectangular spectral collocation as described in Driscoll & Hale (2015), it was demonstrated that the explicit formula could allow faster and, in some sense, more accurate construction of the matrix than existing approaches. A simple recurrence for higher-order rectangular differentiation matrices was also derived, and the analogous results for rectangular differentiation matrices that map from first-kind grids to first-kind grids can be found in the Appendix.

Acknowledgement

The views expressed in this article are not those of the ERC or the European Commission, and the European Union is not liable for any use that may be made of the information contained here. The authors are grateful to Alex Townsend for suggesting the idea of computing the differentiation matrix in Chebyshev coefficient space, as outlined in Section 6.3, and to the anonymous referees for catching a number of $t \leftrightarrow \tau$ errors and suggesting a cleaner formulation of equation (3.8).

Funding

European Research Council/European Union Seventh Framework Programme ((FP7/2007-2013)/ERC grant agreement 291068 to K.X.).

REFERENCES

- BALTENSPERGER, R. & BERRUT, J.-P. (1999) The errors in calculating the pseudospectral differentiation matrices for Čebyšev–Gauss–Lobatto points. *Comput. Math. Appl.*, **37**, 41–48.

- BALTENSPERGER, R. & TRUMMER, M. R. (2003) Spectral differencing with a twist. *SIAM J. Sci. Comput.*, **24**, 1465–1487.
- BAYLISS, A., CLASS, A. & MATKOWSKY, B. J. (1995) Roundoff error in computing derivatives using the Chebyshev differentiation matrix. *J. Comput. Phys.*, **116**, 380–383.
- BERRUT, J.-P. & TREFETHEN, L. N. (2004) Barycentric Lagrange interpolation. *SIAM Rev.*, **46**, 501–517.
- BOYD, J. P. (2001) *Chebyshev and Fourier Spectral Methods*. New York: Courier Dover Publications.
- BREUER, K. S. & EVERSON, R. M. (1992) On the errors incurred calculating derivatives using Chebyshev polynomials. *J. Comput. Phys.*, **99**, 56–67.
- CANUTO, C. (1988) Spectral methods and a maximum principle. *Math. Comput.*, **51**, 615–629.
- DON, W. S. & SOLOMONOFF, A. (1995) Accuracy and speed in computing the Chebyshev collocation derivative. *SIAM J. Sci. Comput.*, **16**, 1253–1268.
- DRISCOLL, T. A. & HALE, N. (2015) Rectangular spectral collocation. *IMA J. Numer. Anal.* (to appear).
- DRISCOLL, T. A., HALE, N. & TREFETHEN, L. N. (2014) *Chebfun Guide*. Oxford: Pafnuty Publications.
- HALE, N. & XU, K. (2014) *Codes for Computing Rectangular Differentiation Matrices*. Available at <http://github.com/nickhale/rectdiff/>. Accessed 1 August 2014.
- KOPRIVA, D. A. (1998) A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations. *J. Comput. Phys.*, **143**, 125–158.
- KOPRIVA, D. A. & KOLIAS, J. H. (1996) A conservative staggered-grid Chebyshev multidomain method for compressible flows. *J. Comput. Phys.*, **125**, 244–261.
- OLVER, F. W. J., LOZIER, D. W., BOISVERT, R. F. & CLARK, C. W. (eds) (2010) *NIST Handbook of Mathematical Functions*. New York, NY: Cambridge University Press.
- OLVER, S. & TOWNSEND, A. (2013) A fast and well-conditioned spectral method. *SIAM Rev.*, **55**, 462–489.
- SOLOMONOFF, A. (1992) A fast algorithm for spectral differentiation. *J. Comput. Phys.*, **98**, 174–177.
- TANG, T. & TRUMMER, M. R. (1996) Boundary layer resolving pseudospectral methods for singular perturbation problems. *SIAM J. Sci. Comput.*, **17**, 430–438.
- TREFETHEN, L. N. (2000) *Spectral Methods in MATLAB*. Philadelphia, PA: SIAM.
- TREFETHEN, L. N. (2013) *Approximation Theory and Approximation Practice*. Philadelphia, PA: SIAM.
- WELFERT, B. D. (1997) Generation of pseudospectral differentiation matrices. I. *SIAM J. Numer. Anal.*, **34**, 1640–1657.

Appendix. Rectangular differentiation matrices for first-kind points

Here we give the analogous formulae to those in Theorems 3.3 and 4.2 for explicit construction of rectangular differentiation matrices which map between first-kind Chebyshev grids of different sizes. Proofs are omitted, since they are similar to those in Sections 3 and 4.

THEOREM A1 Denote by $D_{m,n}$ the $m \times n$ rectangular differentiation matrix which maps from the n -point Chebyshev grid of the first kind $\underline{\tau}_n$ to the m -point Chebyshev grid of the first kind $\underline{\tau}_m$. If $m < n$, then the entries of $D_{m,n}$ are given by

$$[D_{m,n}]_{ij} = \frac{1}{U_{n-1}(\tau_{j,n})(\tau_{i,m} - \tau_{j,n})} \left(U_{n-1}(\tau_{i,m}) - \frac{T_n(\tau_{i,m})}{n(\tau_{i,m} - \tau_{j,n})} \right), \quad (\text{A.1})$$

with the exception that if $\tau_{i,m}$ and $\tau_{j,n}$ coincide, then

$$[D_{m,n}]_{ij} = \frac{\tau_{i,m}}{2(1 - \tau_{i,m}^2)}. \quad (\text{A.2})$$

THEOREM A2 Denote by $D_{m,n}^{(p)}$ the $m \times n$ rectangular differentiation matrix of order p which maps from an n -point Chebyshev grid of the first kind $\underline{\tau}_n$ to an m -point Chebyshev grid of the first kind $\underline{\tau}_m$. The entries of $D_{m,n}^{(p+1)}$ are given by

$$[D_{m,n}^{(p+1)}]_{ij} = \frac{1}{\tau_{i,m} - \tau_{j,n}} \left[\frac{T_n^{(p+1)}(\tau_{i,m})}{nU_{n-1}(\tau_{j,n})} - (p+1)[D_{m,n}^{(p)}]_{ij} \right]. \quad (\text{A.3})$$

If $\tau_{i,m}$ and $\tau_{j,n}$ coincide, then

$$[D_{m,n}^{(p+1)}]_{ij} = \frac{T_n^{(p+2)}(\tau_{i,m})}{n(p+2)U_{n-1}(\tau_{i,m})}. \quad (\text{A.4})$$