

Sparse Spectral Methods on the Simplex

Sachin Natesh

August 25, 2022

Contents

1	Jacobi ladders for sparse discrete operators in 1D	3
1.1	Jacobi polynomials	3
1.2	Ladder operators	4
1.3	Promotion operators	4
1.4	Modal multiplication operator	5
1.4.1	Multiplication by x	5
1.4.2	Multiplication by $g(x)$	6
1.5	Sparsity patterns of the multiplication operator	7
1.6	Constructing the sparse differential operator	7
1.7	Evaluating the Jacobi polynomial expansion coefficients	9
1.7.1	Weighted L_2 approximation	9
1.7.2	Direct inversion of the Vandermonde matrix	10
1.8	Numerical experiments	10
2	Jacobi Ladders for Sparse Discrete Operators on the Triangle	13
2.1	Koornwinder polynomials	13
2.2	Modal multiplication operator	14
2.2.1	Multiplication by x or y	14
2.2.2	Multiplication by $g(x, y)$	15
2.2.3	Gaussian-like quadrature	15
2.2.4	Numerical algorithm to generate Gaussian-like quadrature	17
2.2.5	Quadrature verification	17
2.2.6	Generalization for higher dimensions	19
2.3	Ladder, lowering and promotion operators for Koornwinder polynomials	20
2.3.1	Promotion operators	20
2.3.2	Ladder operators	21
2.3.3	Weighted ladder operators	22
2.3.4	Lowering operators	22
2.4	Sparse partial differential operator on $\mathcal{T}_{\text{ref}}^2$	23
2.5	Sparse partial differential operator on general \mathcal{T}^2	26
2.6	Numerical experiments	26
3	General theory of orthogonal polynomials	29
3.1	Single variable	30
3.1.1	Recurrence relations	30
3.1.2	The Jacobi matrix	31

3.1.3 Gaussian quadrature	32
3.1.4 Reproducing kernel	33
References	34

1 Jacobi ladders for sparse discrete operators in 1D

We construct a sparse discrete differential operator for general, linear, two-point boundary value problems on $[-1, 1]$. Our modal discretization will approximate the coefficients of the solution expanded in the Jacobi polynomial basis. The problem we will consider is

$$p(x)u''(x) + r(x)u'(x) + q(x)u(x) = f(x), \quad x \in (-1, 1) \quad (1.1)$$

$$\alpha_1 u(-1) + \alpha_2 u'(1) = \alpha, \quad (1.2)$$

$$\beta_1 u(-1) + \beta_2 u'(1) = \beta, \quad (1.3)$$

with $\alpha, \alpha_i, \beta, \beta_i \in \mathbb{R}$.

1.1 Jacobi polynomials

Consider the Jacobi polynomials $P_k^{(a,b)}(x)$ of order k with parameters $a, b \in \mathbb{R}$ such that $a, b > -1$. These are orthogonal polynomials with respect to the weight function $w(x) = (1-x)^a(1+x)^b$ on the interval $[-1, 1]$. While the polynomials can be explicitly given in terms of the hypergeometric function, we list here only a few special cases, recurrence relations, and properties of relevance to our goal of constructing sparse discretized operators in 1D.

Let scalars q, r, s, t be defined in terms of the order and parameters (k, a, b) by

$$\begin{aligned} q &= k^2(k+a+b)(k^2+a+b-2), \\ r &= (k^2+a+b-1)(a^2-b^2), \\ s &= (k^2+a+b-2)(k^2+a+b-1)(k^2+a+b), \\ t &= 2(k+a-1)(k+b-1)(k^2+a+b). \end{aligned} \quad (1.4)$$

Then, we can completely specify $P_k^{(a,b)}$ for any k :

$$\begin{aligned} P_0^{(a,b)}(x) &= 1, \\ P_1^{(a,b)}(x) &= \frac{1}{2}((a+b+2)(x-1) + 2(a+1)), \\ P_{k \geq 2}^{(a,b)}(x) &= ((r+sx)P_{k-1}^{(a,b)}(x) - tP_{k-2}^{(a,b)}(x))/q. \end{aligned} \quad (1.5)$$

The polynomials satisfy the following orthogonality relation

$$\int_{-1}^1 (1-x)^a(1+x)^b P_i^{(a,b)}(x) P_j^{(a,b)}(x) dx = \frac{2^{a+b+1}}{2j+a+b+1} \frac{\Gamma(j+a+1)\Gamma(j+b+1)}{\Gamma(j+a+b+1)j!} \delta_{ij}, \quad (1.6)$$

and for $i = j$, the right hand side of (1.6) gives us the squared normalization factor for the polynomials. For the orthonormal polynomials expressed in the recurrence form of (3.8), the coefficients are

$$a_{k>0} = \frac{2}{a+b+2k+2} \left(\frac{(a+k+1)(b+k+1)(k+1)(a+b+k+1)}{(a+b+2k+1)(a+b+2k+3)} \right)^{1/2}, \quad (1.7)$$

$$a_0 = \frac{2}{a+b+2} \left(\frac{(a+1)(b+1)}{a+b+3} \right)^{1/2}, \quad (1.8)$$

$$b_{k>0} = -B_k/A_k, \quad b_0 = \frac{\frac{1}{2}(b-a)}{\frac{1}{2}(a+b)+1}, \quad (1.9)$$

where A_k, B_k are given in [1, Eq. 18.9.2]. We also note the following properties:

- (i) $P_k^{(a,b)}(-1) = (-1)^k \binom{k+b}{k}$
- (ii) $P_k^{(a,b)}(1) = \binom{k+a}{k}$
- (iii) $P_{-1}^{(a,b)}(x) = 0$ (convention we will adopt)

1.2 Ladder operators

There exist so called ladder operators which relate $P_k^{(a,b)}$ to another Jacobi polynomial with either the order k or parameters a, b incremented or decremented [2]. Here, we will consider the first ladder operation \mathcal{L}_1 , given by

$$\mathcal{L}_1 P_k^{(a,b)} = \frac{1}{2}(k+a+b+1)P_{k-1}^{(a+1,b+1)} = \frac{d}{dx}P_k^{(a,b)}. \quad (1.10)$$

That is, \mathcal{L}_1 relates the (k, a, b) polynomial to its derivative in terms of $(k-1, a+1, b+1)$ polynomial. Applying \mathcal{L}_1 to $P_k^{(a,b)}$ gives a similar relation for its second derivative;

$$\frac{d^2}{dx^2}P_k^{(a,b)} = \frac{1}{4}(k+a+b+1)(k+a+b+2)P_{k-2}^{(a+2,b+2)}, \quad k \geq 2 \quad (1.11)$$

$$\frac{d^2}{dx^2}P_k^{(a,b)} = 0, \quad k = 0, 1 \quad (1.12)$$

where now the order and parameters have been, respectively, decremented and incremented by 2.

1.3 Promotion operators

Consider an ODE with highest derivative order l , and suppose we represent the solution, RHS and coefficient functions by Jacobi polynomials with parameters (a, b) . We showed in the next section that these representations must be promoted to $(a+l, b+l)$ to match the coefficients of the highest derivative obtained by the ladder operation. In Section 1.8, we numerically demonstrate that greater Jacobi parameters lead to instability for increasing polynomial order. However, one can derive by L^2 -projection the promotion operators relating the Jacobi polynomial $P_k^{(a,b)}$ to $P_k^{(a+1,b+1)}$, as done in [3].

That is, if $\mathbf{f}^{a,b} = [f_0, \dots, f_N]^T$ are the expansion coefficients of $f(x)$ in the basis given by rows of $\mathbf{P}^{(a,b)}(x) = [P_0^{(a,b)}(x), \dots, P_N^{(a,b)}(x)]^T$, i.e.

$$f(x) \approx [\mathbf{P}^{(a,b)}(x)]^T \mathbf{f}^{(a,b)},$$

then there exists a matrix $\mathcal{K}_{(a,b)}^{(a+1,b+1)}$ such that

$$\mathbf{f}^{(a+1,b+1)} = \mathcal{K}_{(a,b)}^{(a+1,b+1)} \mathbf{f}^{(a,b)}. \quad (1.13)$$

In particular, $\mathcal{K}_{(a,b)}^{(a+1,b+1)}$ is the upper triangular matrix

$$\mathcal{K}_{(a,b)}^{(a+1,b+1)} = \begin{bmatrix} \lambda_1^{(a,b,0)} & \lambda_2^{(a,b,0)} & \lambda_3^{(a,b,0)} & & \\ & \lambda_1^{(a,b,1)} & \lambda_2^{(a,b,1)} & \ddots & \\ & & \ddots & \ddots & \lambda_3^{(a,b,N-2)} \\ & & & \ddots & \lambda_2^{(a,b,N-1)} \\ & & & & \lambda_1^{(a,b,N)} \end{bmatrix} \quad (1.14)$$

with diagonal entries

$$\begin{aligned}\lambda_1^{(a,b,i)} &= \frac{(a+b+i+1)(a+b+i+2)}{(a+b+2i+1)(a+b+2i+2)} \\ \lambda_2^{(a,b,i)} &= \frac{(a-b)(a+b+i+2)}{(a+b+2i+2)(a+b+2i+4)} \\ \lambda_3^{(a,b,i)} &= -\frac{(a+i+2)(b+i+2)}{(a+b+2i+4)(a+b+2i+5)}.\end{aligned}\tag{1.15}$$

Hence, we can construct an upper triangular promotion operator of order l through

$$\mathcal{K}_{(a,b)}^{(a+l,b+l)} = \mathcal{K}_{(a+l-1,b+l-1)}^{(a+l,b+l)} \cdots \mathcal{K}_{(a+1,b+1)}^{(a+2,b+2)} \cdot \mathcal{K}_{(a,b)}^{(a+1,b+1)},\tag{1.16}$$

which will promote $\mathbf{f}^{(a,b)}$ to $\mathbf{f}^{(a+l,b+l)}$

1.4 Modal multiplication operator

Our model discretization of (1.1) will require us to represent multiplications involving ODE coefficient functions in the modal basis. For example, we need to construct an operator which will map the coefficients of u to those of $q(x)u(x)$.

1.4.1 Multiplication by x

Recurrence relations of the form of (3.7) naturally provide a “multiplication by x ” operator, which, given coefficients of f , maps to the coefficients of the product xf . For $\mathbf{P}(x) \in \mathbb{R}^n$ we have

$$x\mathbf{P}(x) = \mathcal{J}_n\mathbf{P}(x) + a_{n-1}P_n(x)\mathbf{e}_n,\tag{1.17}$$

where \mathcal{J}_n is the tridiagonal Jacobi matrix encoding the recurrence (see (3.9) for the orthonormal case) given by

$$\mathcal{J}_n = \begin{bmatrix} b_0 & a_0 & & & \\ c_1 & b_1 & a_1 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & b_{n-2} & a_{n-2} \\ & & & c_{n-2} & b_{n-1} \end{bmatrix},\tag{1.18}$$

and \mathbf{e}_n is the n^{th} standard basis vector. Consider the Jacobi expansion of $f(x) \approx \mathbf{f}^T\mathbf{P}(x)$, and the product $xf(x)$:

$$xf(x) \approx x\mathbf{P}(x)^T\mathbf{f} = (\mathbf{P}(x)^T\mathcal{J}_n^T + [0 \ \cdots \ 0 \ a_{n-1}P_n(x)])\mathbf{f}\tag{1.19}$$

If we evaluate the above expression for \mathbf{x} a discrete n point grid from -1 to 1 (row vector), we see that $[\mathbf{P}(\mathbf{x})]^T$ is the Vandermonde matrix, and the component-wise product is

$$[\mathbf{x} \odot f(\mathbf{x})]^T \approx \mathbf{P}(\mathbf{x})^T(\mathcal{J}_x\mathbf{f}), \text{ where } \mathcal{J}_x = \left(\mathcal{J}_n^T + \mathbf{P}(\mathbf{x})^{-T} [\mathbf{0}_{n \times n-1} \ a_{n-1}P_n(\mathbf{x}^T)] \right)\tag{1.20}$$

Markedly, \mathcal{J}_n^T is the **orthogonal projection operator composed with the multiplication operator. It takes the modes of $f(x)$ and computes the modes of the projection of $xf(x)$ in the span of $\mathbf{P}(x)$.** \mathcal{J}_x is our modal multiplication-by- x operator, and \mathcal{J}_x^l represents multiplication by \mathbf{x}^{ol} (Hadamard power), as in

$$[\mathbf{x}^{\text{ol}} \odot f(\mathbf{x})]^T \approx \mathbf{P}(\mathbf{x})^T(\mathcal{J}_x^l\mathbf{f})\tag{1.21}$$

1.4.2 Multiplication by $g(x)$

We can use the multiplication operators to construct a general operator which can map the coefficients of f to those of the product fg for any function g . To see this, suppose f and g are represented in Jacobi bases of possibly different degrees $n-1, m-1$. Then, by the equivalent representation of g in the monomial basis $\tilde{\mathbf{P}}(x) = [1, x, \dots, x^{m-1}]^T$ with coefficients $\tilde{\mathbf{g}}$,

$$\begin{aligned} [f(\mathbf{x}) \odot g(\mathbf{x})]^T &\approx f(\mathbf{x})^T \odot (\mathbf{P}(\mathbf{x})^T \mathbf{g}) = f(\mathbf{x})^T \odot (\tilde{\mathbf{P}}(\mathbf{x})^T \tilde{\mathbf{g}}) \\ &= \left(\sum_{j=0}^{m-1} \tilde{g}_j [\mathbf{x}^{\odot j} \odot f(\mathbf{x})]^T \right) = \mathbf{P}(\mathbf{x})^T \left(\sum_{j=0}^{m-1} \tilde{g}_j \mathcal{J}_x^j \right) \mathbf{f} \\ &\approx \mathbf{P}(\mathbf{x})^T (g(\mathcal{J}_x) \mathbf{f}), \end{aligned}$$

where $g(\mathcal{J}_x)$ is approximated in terms of the Jacobi polynomial expansion given matrix input. Thus, given an $n \times n$ multiplication-by- x operator \mathcal{J}_x for approximating the n coefficients of $xf(x)$, the multiplication-by- g operator for f is simply

$$\mathcal{M}_g = g(\mathcal{J}_x), \quad (1.22)$$

and allows us to approximate the n expansion coefficients of $f(x)g(x)$ by $M_g \mathbf{f}$.

We can generalize Clenshaw's algorithm [4] for operator valued input to evaluate $g(\mathcal{J}_x)$ given the Jacobi expansion coefficients of g . The scalar-valued Clenshaw algorithm is derived by first recasting the recurrence (3.7) as the set of equations $\{P_0 = 1, a_n P_{n+1} + (b_n - x)P_n + c_n P_{n-1} = 0\}$. Then, as a matrix equation with $\mathbf{e}_1 \in \mathbb{R}^n$ the first standard basis vector, we have

$$\mathcal{C} \mathbf{P}(x) = \mathbf{e}_1, \quad \mathcal{C} = \begin{bmatrix} 1 & & & & & \\ b_0 - x & a_0 & & & & \\ c_1 & b_1 - x & a_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & c_{n-3} & b_{n-2} - x & a_{n-2} & \end{bmatrix}, \quad (1.23)$$

so that

$$g(x) = \mathbf{P}(x)^T \mathbf{g} = \mathbf{e}_1^T (\mathcal{C}^{-T} \mathbf{g}) \quad (1.24)$$

The operator valued algorithm follows analogously. Let I be the $n \times n$ identity matrix and $\mathbf{e}_1 \in \mathbb{R}^m$ the first standard basis vector (of length m). We know that $\mathbf{P}(\mathcal{J}_x)$ is an m -length block column vector with $n \times n$ blocks. Then,

$$\mathcal{C} \mathbf{P}(\mathcal{J}_x) = (\mathbf{e}_1 \otimes I), \quad \mathcal{C} = \begin{bmatrix} I & & & & & \\ b_0 I - \mathcal{J}_x & a_0 I & & & & \\ c_1 I & b_1 I - \mathcal{J}_x & a_1 I & & & \\ & \ddots & \ddots & \ddots & & \\ & & c_{n-3} I & b_{n-2} I - \mathcal{J}_x & a_{n-2} I & \end{bmatrix}, \quad (1.25)$$

so that

$$\mathcal{M}_g = g(\mathcal{J}_x) = \mathbf{P}(\mathcal{J}_x)^T (\mathbf{g} \otimes I) = (\mathbf{g} \otimes I)^T [\mathcal{C}^{-1} (\mathbf{e}_1 \otimes I)]. \quad (1.26)$$

Here, $\mathcal{C} \in \mathbb{R}^{mn \times mn}$ and contains m blocks of size $n \times n$ on the main diagonal. We have actually revealed a multiplication-by- \star operator, i.e.

$$\mathcal{M}_\star = \mathcal{C}^{-1} (\mathbf{e}_1 \otimes I), \quad \mathcal{M}_g = (\mathbf{g} \otimes I)^T \mathcal{M}_\star, \quad (1.27)$$

which can be precomputed given the expansion of f , and reused to construct \mathcal{M}_g for an arbitrary function g .

1.5 Sparsity patterns of the multiplication operator

For variable coefficient problems, it is desirable that the coefficients can be represented by a polynomial of degree $M \leq N$, where N is the representation order for the solution. If $M = N$, then \mathcal{M}_g is completely filled in. As M decreases, \mathcal{M}_g develops a banded structure. Hence, the max polynomial degree for representing the coefficient functions should be carefully selected. Moreover, if $M > N$, the multiplication operators will not resolve the products well. In Figure 1, we show the sparsity pattern of \mathcal{M}_g for $g(x) = \{x^2, x^3, x^4\}$ given $f(x) = e^{x \cos(x)}$. We need $M + 1$ points to represent a polynomial of degree M , and each extra degree adds 2 to the bandwidth of the block diagonal part of \mathcal{M}_g .

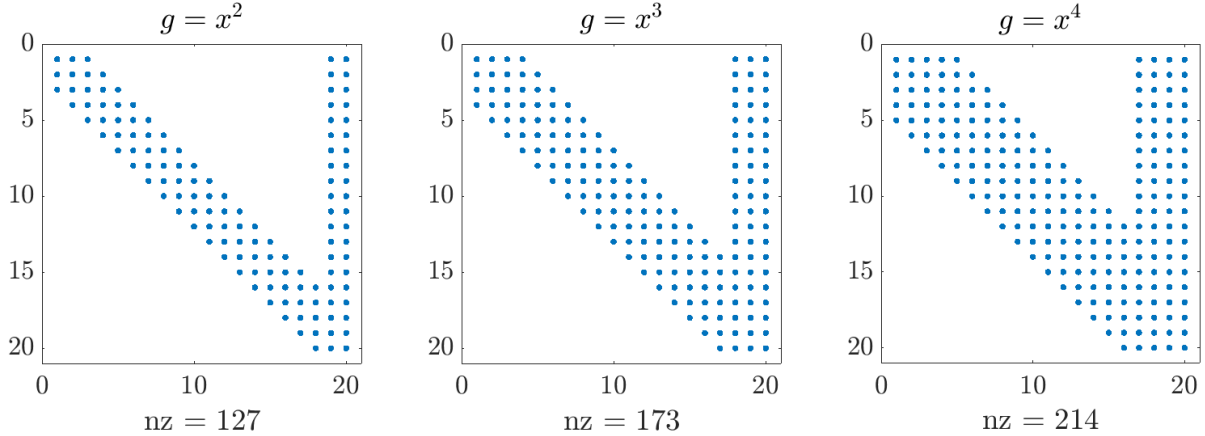


Figure 1: Sparsity pattern for \mathcal{M}_g given $f(x) = e^{x \cos(x)}$. We used $N = 20$ points to represent f (degree $N - 1$).

1.6 Constructing the sparse differential operator

Equipped with our ladder, promotion and multiplication operators, we can now construct a sparse differential operator for (1.1)-(1.3). First, we consider an N^{th} degree polynomial approximation for u , and M_q, M_r, M_p degree approximations for the coefficient functions p, r, q . For example,

$$u(x) \approx \sum_{k=0}^N a_k P_k^{(a,b)}(x). \quad (1.28)$$

Inserting (1.28) into (1.1) and applying (1.10)-(1.12), we find

$$\begin{aligned} u''(x) &\approx \sum_{k=2}^N \left(\frac{a_k}{4} (k+a+b+1)(k+a+b+2) \right) P_{k-2}^{(a+2,b+2)}(x), \\ u'(x) &\approx \sum_{k=1}^N \frac{a_k}{2} (k+a+b+1) P_{k-1}^{(a+1,b+1)}(x) \end{aligned} \quad (1.29)$$

According to (1.29), there exists linear mappings, which we denote $\mathcal{D}^2 : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^{N+1}$, and \mathcal{D} that relates the Jacobi coefficients of u to those of u'' , and u' respectively. Explicitly, for $i, j = 0, 1, \dots, N$,

the entries of \mathcal{D}^2 are given by

$$\mathcal{D}_{ij}^2 = \begin{cases} \frac{1}{4}(a+b+j+1)(a+b+j+2) & , \quad j = i+2 \\ 0 & , \quad j \neq i+2, \end{cases} \quad (1.30)$$

and those of \mathcal{D} are given by

$$\mathcal{D}_{ij} = \begin{cases} \frac{1}{2}(a+b+j+1) & , \quad j = i+1 \\ 0 & , \quad j \neq i+1, \end{cases} \quad (1.31)$$

We see that $\mathcal{D}^2 \mathbf{a}$ gives the Jacobi coefficients of u'' in the basis of $\mathbf{P}^{(a+2,b+2)}$. That is,

$$u''(x) \approx (\mathcal{D}^2 \mathbf{a})^T \mathbf{P}^{(a+2,b+2)}(x). \quad (1.32)$$

Assuming the coefficients of p, r, q are appropriately promoted, we can use the multiplication operator to explicitly write the discrete modal operator:

$$\begin{aligned} \mathcal{L} &= \mathcal{M}_p \mathcal{D}^2 + \mathcal{M}_r \mathcal{K}_{(a+1,b+1)}^{(a+2,b+2)} \mathcal{D} + \mathcal{M}_q \mathcal{K}_{(a,b)}^{(a+2,b+2)}, \\ \mathcal{L} \mathbf{a} &= \mathcal{K}_{(a,b)}^{(a+2,b+2)} \mathbf{f}. \end{aligned} \quad (1.33)$$

As is, \mathcal{L} is not invertible. However, we can add the independent boundary conditions (1.2)-(1.3) to eliminate the rank deficiency. By properties (ii) and (iii) in Section 1.1, and using the first ladder operator, we have that

$$\begin{aligned} u(-1) &\approx \sum_{k=0}^N (-1)^k \binom{k+b}{k} a_k, \quad u(1) \approx \sum_{k=0}^N \binom{k+a}{k} a_k \\ u'(-1) &\approx \sum_{k=1}^N \frac{k+a+b+1}{2} (-1)^k \binom{k+b+1}{k} a_k, \quad u'(1) \approx \sum_{k=1}^N \frac{k+a+b+1}{2} \binom{k+a+1}{k} a_k \end{aligned} \quad (1.34)$$

Hence, we can define a modified operator $\hat{\mathcal{L}}$ with same entries as \mathcal{L} , except

$$\hat{\mathcal{L}}_{ij} = \begin{cases} \mathcal{L}_{ij} & , \quad i \leq N-2 \\ \alpha_1 (-1)^j \binom{j+b}{j} & , \quad i = N-1, j = 0 \\ \alpha_1 (-1)^j \binom{j+b}{j} + \alpha_2 \frac{j+a+b+1}{2} (-1)^j \binom{j+b+1}{j} & , \quad i = N-1, j \neq 0 \\ \beta_1 \binom{j+a}{j} & , \quad i = N, j = 0 \\ \beta_1 \binom{j+a}{j} + \beta_2 \frac{j+a+b+1}{2} \binom{j+a+1}{j} & , \quad i = N, j \neq 0 \end{cases} \quad (1.35)$$

Now, if we expand $f(x) \approx \sum_{k=0}^N c_k P_k^{(a+2,b+2)}(x)$ (obtained by the promotion of \mathbf{f}), then we can approximate a solution to (1.1)-(1.3) by solving

$$\hat{\mathcal{L}} \mathbf{a} = \begin{pmatrix} \tilde{\mathbf{c}} \\ \alpha \\ \beta \end{pmatrix}, \quad (1.36)$$

where $\tilde{\mathbf{c}} = (c_0, c_1, \dots, c_{N-2})^T$. Note, we have to expand \mathbf{f} in the basis of $\mathbf{P}^{(a+2,b+2)}$ in order to solve the modal system, while \mathbf{a} are the coefficients of u in the basis of $\mathbf{P}^{(a,b)}$. More generally, we can

construct the operator \mathcal{D}^l such that $\mathcal{D}^l \mathbf{a}$ gives the Jacobi coefficients of $\partial^l u$ in the basis of $\mathbf{P}^{(a+l, b+l)}$ by

$$\mathcal{D}_{ij}^l = \begin{cases} \frac{1}{2^l} \prod_{k=1}^l (a + b + j + k) & , \quad j = i + l \\ 0 & , \quad j \neq i + l, \end{cases} \quad (1.37)$$

and l boundary conditions must be added in a manner similar to (1.35) in order to eliminate the rank deficiency.

1.7 Evaluating the Jacobi polynomial expansion coefficients

If nodal exactness in the approximation for f is not a concern, we will use a best least squares approximation in the weighted L_2 sense to compute the expansion coefficients. If nodal exactness is desired, we can directly invert the Jacobi Vandermonde matrix. Similarly, if f is already a high order, accurate polynomial interpolant in a different basis, direct inversion gives an identical interpolant in the Jacobi basis, owed to the uniqueness of polynomial interpolation.

1.7.1 Weighted L_2 approximation

Consider the polynomial expansion in the weighted L_2 sense of $f \in \Pi_N([-1, 1])$ in the span of $\{P_j^{(a,b)}\}_{j=0}^N$:

$$f(x) = \sum_{j=0}^N c_j P_j^{(a,b)}(x), \quad c_j = \frac{1}{\sqrt{h_j}} \int_{-1}^1 f(x) p_j^{(a,b)}(x) w(x) dx, \quad (1.38)$$

where $w(x)$ is the Jacobi weight function, $p_j^{(a,b)}$ are the normalized polynomials, and h_j is the structure factor (as in (1.6)) for $P_j^{(a,b)}$. This generalized Fourier series follows from the orthogonality of the Jacobi polynomials, and equality in the first term becomes \approx if f is not polynomial.

We can compute the coefficients c_j using the Jacobi Vandermonde matrix. To see this, suppose that we have an $M + 1$ point quadrature rule on $[-1, 1]$, capable of exactly integrating polynomials up to degree $2N$. For example, a Clenshaw-Curtis-Jacobi (CCJ) rule will require $M \geq 2N$, while a Gauss-Jacobi (GJ) rule requires $M \geq N$. Let $\{x_k\}_{k=0}^M$ with $-1 \leq x_0 < x_k < x_M \leq 1$ be the quadrature nodes, $\mathbf{w} = (w_0, w_1, \dots, w_M)^T$ the weights, and $\mathbf{f} = (f(x_0), f(x_1), \dots, f(x_M))^T$ the function values at the quadrature nodes.

Using the recurrence relation in (1.5), we can construct the $(M + 1) \times (N + 1)$ Vandermonde matrix V with entries $V_{kj} = P_j^{(a,b)}(x_k)$. We also incorporate the structure factors from (1.6) into a vector \mathbf{h} . Noting that $\deg(f P_j^{(a,b)}) \leq 2N$,

$$\begin{aligned} c_j &= \frac{1}{\sqrt{h_j}} \int_{-1}^1 \frac{P_j^{(a,b)}(x)}{\sqrt{h_j}} f(x) w(x) dx = \frac{1}{h_j} \sum_{k=0}^M (P_j^{(a,b)}(x_k) f(x_k)) w_k \\ \Rightarrow \mathbf{c} &= \left(V^T (\mathbf{f} \odot \mathbf{w}) \right) \oslash \mathbf{h}, \end{aligned} \quad (1.39)$$

where \odot, \oslash are component-wise multiplication and division, respectively. If $\deg(f) > N + 1$, or if f is not a polynomial, then the equality in (1.39) changes to \approx . **In Chebfun, it seems that they take this approach for converting between Chebyshev and Jacobi coefficients. However, this does not preserve the nodal exactness of the Chebyshev interpolant (easy to check). Moreover, we show in the next section that Clenshaw-Curtis rules are unstable for larger N and a, b .**

1.7.2 Direct inversion of the Vandermonde matrix

The coefficients for the interpolant in the Jacobi polynomial basis are given by

$$\mathbf{c} = V^{-1} \mathbf{f}, \quad (1.40)$$

where V is $(N + 1) \times (N + 1)$, and need not include any normalization factors.

1.8 Numerical experiments

Here, we provide convergence results for both Vandermonde inversion and weighted L_2 projection in approximating solutions to (1.1)-(1.3) with $f(x)$ taken from $\{x, x^2, x^3, e^x \sin(x), \arctan(x)\}$. The solution nodes are the Chebyshev points $x_j = -\cos(j\pi/N)$ for $j = 0, 1, \dots, N$. In Figure 2, we show approximate solutions for each right-hand side $f(x)$.

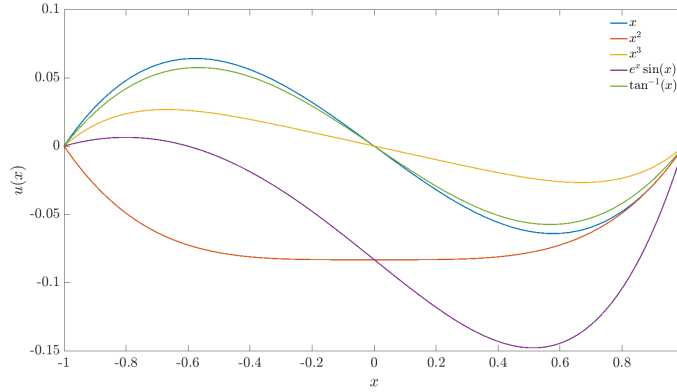


Figure 2: Solutions to (1.1)-(1.3) for several right-hand sides.

The error we report in Figure 3 is the approximate interpolation error in the weighted L_2 norm. Given the solution coefficients \mathbf{a} , we compute

$$E^2(u) = \left\| u(x) - \sum_{j=0}^N a_j P_j^{(a,b)}(x) \right\|_{L_2^w([-1,1])}^2 \approx \sum_{k=0}^M \left(u(x_k) - \sum_{j=0}^N a_j P_j^{(a,b)}(x_k) \right)^2 w_k, \quad (1.41)$$

where $\{x_k, w_k\}_{k=0}^M$ is an $M + 1$ point quadrature rule ($M \gg N$). In words, we interpolate the approximate solution onto a fine $M + 1$ point grid, and use the associated Gauss-Jacobi quadrature weights to approximate interpolation error. For these tests, we take $M = 50$, and Jacobi parameters $(a, b) = (3, 4)$.

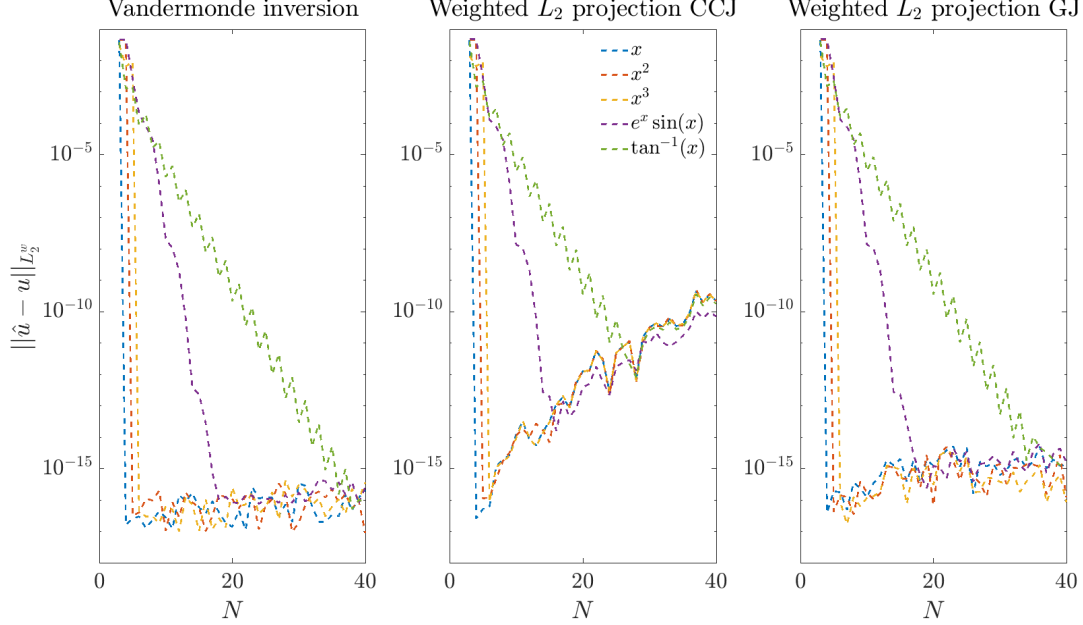


Figure 3: Weighted L_2 errors in approximate solutions to (1.1)-(1.3) for several right-hand sides.

We see that a polynomial right-hand side of degree k requires $k + 3$ points to approximate the solution to machine precision. Spectral convergence is observed for all f considered. Moreover, Vandermonde inversion does not lose correct digits as $N \rightarrow \infty$, while the L_2 projection with CCJ quadrature is not as stable. On the other hand, GJ quadrature behaves more stably, like Vandermonde inversion.

The instability is related to the Jacobi parameters (a, b) . For the problem of finding the best weighted L_2 coefficients (using the Clenshaw-Curtis type method in (1.39)) of a function f , larger values of (a, b) lead to loss in digits for fixed N large enough. This is illustrated in Figure 4, where with $N = 20$, we compute the weighted L_2 error in the projection of $f(x) = e^x \sin(x)$ onto $\mathbf{P}^{(a,b)}$ for a range of (a, b) . Notice, for $a + b \in (-2, 2)$, a regime which includes the Chebyshev and Legendre polynomials, the error decreases and is well controlled within 10^{-10} as a and b are incremented. On the other hand, the $a = b$ regime, corresponding to the ultraspherical (i.e. Gegenbauer) polynomials, and as well the $a, b > 1$ regime, display an increase in error as a and b are incremented. For comparison, the right panel shows the same test with Gauss-Jacobi quadrature.

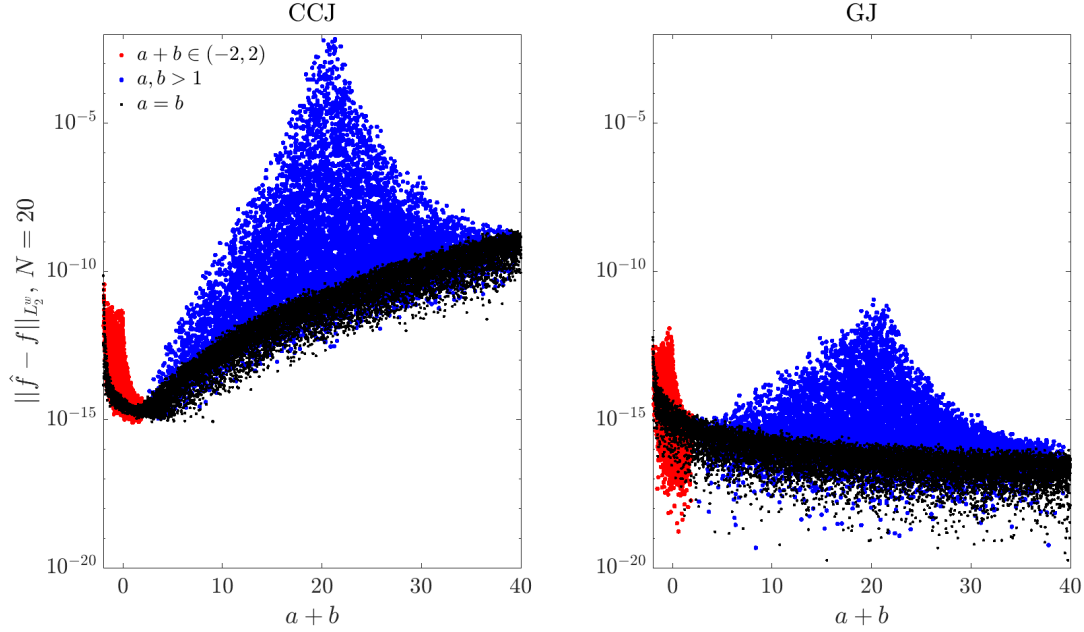


Figure 4: Weighted L_2 errors in the projection of $f(x) = e^x \sin(x)$ onto $\mathbf{P}^{(a,b)}$ for various regimes in (a, b) and $N = 20$. CCJ is the Clenshaw-Curtis-Jacobi quadrature method, while GJ is Gauss-Jacobi.

Hence, care must be taken when selecting the (a, b) parameters to represent the solution of the differential equation, if the RHS f is expressed via L_2 projection. Given that an order k derivative will increment the parameters by k , the handling of high order equations is limited - unless a more stable quadrature is chosen to evaluate the integral in (1.39).

2 Jacobi Ladders for Sparse Discrete Operators on the Triangle

Here, we construct sparse differential operators for BVPs on the triangle

$$\mathcal{T}_{\text{ref}}^2 := \{(x, y) \mid 0 \leq x, y, x + y \leq 1\}$$

using the classical orthogonal polynomials on this domain, due to Koornwinder. For starters, we want to solve Poisson's equation with Dirichlet data:

$$\begin{aligned} \Delta u(x, y) &= f(x, y), & (x, y) &\in \mathcal{T}_{\text{ref}}^2 \\ u(x, y) &= g(x, y), & (x, y) &\in \partial \mathcal{T}_{\text{ref}}^2. \end{aligned} \quad (2.1)$$

2.1 Koornwinder polynomials

To simplify notation in this section, P_k^n denotes the k^{th} Koornwinder polynomial of degree n with parameters (a, b, c) . When the parameters must be specified, we will refer to the same polynomial by $P_{k,n}^{(a,b,c)}$. The weight function is

$$\begin{aligned} W(x, y) &= x^{a-\frac{1}{2}} y^{b-\frac{1}{2}} (1-x-y)^{c-\frac{1}{2}}, \\ w_{a,b,c} &= \left(\int_{\mathcal{T}_{\text{ref}}^2} W(x, y) dx dy \right)^{-1} = \frac{\Gamma(|\kappa| + \frac{3}{2})}{\Gamma(a + \frac{1}{2}) \Gamma(b + \frac{1}{2}) \Gamma(c + \frac{1}{2})} \end{aligned} \quad (2.2)$$

with $a, b, c > -1/2$ the Jacobi parameters, and $|\kappa| = a + b + c$. The orthonormal polynomials are given in terms of the 1D Jacobi polynomials by

$$\begin{aligned} P_k^n(x, y) &= (h_{k,n})^{-1} P_{n-k}^{2k+b+c, a-1/2}(2x-1)(1-x)^k P_k^{c-1/2, b-1/2}\left(\frac{2y}{1-x} - 1\right), \\ (h_{k,n})^2 &= \frac{w_{a,b,c}}{(2n + |\kappa| + \frac{1}{2})(2k + b + c)} \\ &\times \frac{\Gamma(n+k+b+c+1)\Gamma(n-k+a+\frac{1}{2})\Gamma(k+b+\frac{1}{2})\Gamma(k+c+\frac{1}{2})}{(n-k)!k!\Gamma(n+k+|\kappa|+\frac{1}{2})\Gamma(k+b+c)}, \end{aligned} \quad (2.3)$$

and they satisfy a mutual orthonormality condition under the weighted inner product:

$$\langle P_j^m, P_k^n \rangle = \delta_{jk} \delta_{mn}. \quad (2.4)$$

That is to say, a given polynomial is not only orthonormal to those with different total degree, but also to those of the same degree. Recall, in 1D we have convenient explicit evaluations of the polynomials at the endpoints. On the triangle, we have.

- (i) $P_k^n(0, y) = (h_{k,n})^{-1} (-1)^{n-k} \binom{n-k+a-1/2}{n-k} P_k^{c-1/2, b-1/2}(2y-1)$
- (ii) $P_k^n(x, 0) = (h_{k,n})^{-1} P_{n-k}^{2k+b+c, a-1/2}(2x-1)(1-x)^k (-1)^k \binom{k+b-1/2}{k}$
- (iii) $P_k^n(x, 1-x) = (h_{k,n})^{-1} P_{n-k}^{2k+b+c, a-1/2}(2x-1)(1-x)^k \binom{k+c-1/2}{k}$

While we do not have closed-form expressions for boundary evaluation, these properties are nonetheless useful. They allow us to compute more general boundary conditions without modifying the basis.

2.2 Modal multiplication operator

We will derive the multiplication operator associated with g for computing the modes of $f(x, y)g(x, y)$ given the modes of f . As in 1D, we begin with the multiplication by x, y operators and use them to construct the multiplication by g operator.

2.2.1 Multiplication by x or y

P_k^n , $0 \leq k \leq n$ forms a basis of the space of homogeneous polynomials of degree n . By considering this subspace instead of individual polynomials, our recurrence relations from 1D are generalized to matrix relations. Let

$$\mathbb{P}_n(x_1, x_2) = (P_0^n(x_1, x_2), \dots, P_n^n(x_1, x_2))^T.$$

The orthonormal 3-term relation is given by

$$x_i \mathbb{P}_n = A_{n,i} \mathbb{P}_{n+1} + B_{n,i} \mathbb{P}_n + A_{n-1,i}^T \mathbb{P}_{n-1}, \quad i = 1, 2, \quad (2.5)$$

$$A_{n,i} = \langle x_i \mathbb{P}_n, \mathbb{P}_{n+1}^T \rangle \in \mathbb{R}^{(n+1) \times (n+2)}, \quad (2.6)$$

$$B_{n,i} = \langle x_i \mathbb{P}_n, \mathbb{P}_n^T \rangle \in \mathbb{R}^{(n+1) \times (n+1)} \quad (2.7)$$

where $\langle \cdot \rangle$ is the weighted L^2 inner product. Explicitly, the coefficient matrices are

$$A_{n,1} = \begin{bmatrix} a_{0,n} & & & 0 \\ & a_{1,n} & & 0 \\ & & \ddots & \vdots \\ & & & a_{n,n} & 0 \end{bmatrix}, \quad B_{n,1} = \begin{bmatrix} b_{0,n} & & & \\ & b_{1,n} & & \\ & & \ddots & \\ & & & b_{n,n} \end{bmatrix}, \quad (2.8)$$

$$A_{n,2} = \begin{bmatrix} e_{0,n} & d_{0,n} & & & 0 \\ c_{1,n} & e_{1,n} & d_{1,n} & & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ & & c_{n,n} & e_{n,n} & d_{n,n} \end{bmatrix}, \quad B_{n,2} = \begin{bmatrix} f_{0,n} & g_{0,n} & & & \\ g_{0,n} & f_{1,n} & g_{1,n} & & \\ & \ddots & \ddots & \ddots & \\ & & & g_{n-1,n} & f_{n,n} \end{bmatrix}, \quad (2.9)$$

and the formulas for $a_{i,n}, b_{i,n}, \dots$ can be found on pages 80-81 of [5].

The matrix recurrence generates an analog of the Jacobi matrix from 1D, except now there are two of them - one for each dimension:

$$\mathcal{J}_{n,i} = \begin{bmatrix} B_{0,i} & A_{0,i} & & & \\ A_{0,i}^T & B_{1,i} & A_{1,i} & & \\ & A_{1,i}^T & \ddots & \ddots & \\ & & \ddots & B_{n-2,i} & A_{n-2,i} \\ & & & A_{n-2,i}^T & B_{n-1,i} \end{bmatrix}, \quad i = 1, 2 \quad (2.10)$$

with $\mathcal{J}_{n,i} \in \mathbb{R}^{N \times N}$ symmetric, and $N = \dim \Pi_{n-1}^d = \binom{n-1+d}{n-1}$. As we did in 1D, we can recast (2.5) by taking

$$\mathbf{P} = (\mathbb{P}_0^T, \mathbb{P}_1^T, \dots, \mathbb{P}_{n-1}^T)^T$$

and writing

$$x_i \mathbf{P} = \mathcal{J}_{n,i} \mathbf{P} + \begin{bmatrix} \mathbf{0}_{(N-n) \times 1} \\ A_{n-1,i} \mathbb{P}_n^T \end{bmatrix}, \quad i = 1, 2. \quad (2.11)$$

If we consider the Jacobi expansion of $f(x_1, x_2) \approx \mathbf{f}^T \mathbf{P}(x_1, x_2)$, then the products $x_i f(x, y)$ are given by

$$x_i f(x_1, x_2) \approx x_i \mathbf{P}(x_1, x_2)^T \mathbf{f} = \left(\mathbf{P}(x_1, x_2)^T \mathcal{J}_{n,i} + [\mathbf{0}_{1 \times (\mathbf{N}-n)} \quad \mathbb{P}_n(x_1, x_2)^T A_{n-1,i}^T] \right) \mathbf{f}, \quad i = 1, 2 \quad (2.12)$$

We can evaluate the above expression for $(\mathbf{x}_1, \mathbf{x}_2)$ a set of N points in $\mathcal{T}_{\text{ref}}^2$ (viewed as a row of N pairs), and see that $[\mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)]^T$ is the Vandermonde matrix. The component-wise product can now be given as

$$[\mathbf{x}_i \odot f(\mathbf{x}_1, \mathbf{x}_2)]^T \approx \mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)^T (\mathcal{J}_{x_i} \mathbf{f}), \quad i = 1, 2 \quad (2.13)$$

where $\mathcal{J}_{x_i} = \left(\mathcal{J}_{n,i} + \mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)^{-T} [\mathbf{0}_{\mathbf{N} \times \mathbf{N}-n} \quad \mathbb{P}_n(\mathbf{x}_1^T, \mathbf{x}_2^T)^T A_{n-1,i}^T] \right)$

If only M instead of N points are available so that $\mathbf{P}(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^{N \times M}$, we can evaluate (2.13) in the least-squares sense:

$$\mathcal{J}_{x_i} = \left(\mathcal{J}_{n,i} + [\mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)^T]^+ [\mathbf{0}_{\mathbf{M} \times \mathbf{M}-n} \quad \mathbb{P}_n(\mathbf{x}_1^T, \mathbf{x}_2^T)^T A_{n-1,i}^T] \right) \quad (2.14)$$

where $[\mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)^T]^+ = (\mathbf{P}(\mathbf{x}_1, \mathbf{x}_2) \mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)^T)^{-1} \mathbf{P}(\mathbf{x}_1, \mathbf{x}_2)$ is the pseudo-inverse. Note, like in 1D, $\mathcal{J}_{n,i}$ are both orthogonal projection operators composed with a multiplication operator .

2.2.2 Multiplication by $g(x, y)$

TODO - derive the correction for truncating the operator in equation (2.12) of [6] - analogous to the 1D case.

2.2.3 Gaussian-like quadrature

The hope here is that we can generalize the Golub-Welsch algorithm described in Section 3.1.3 for the bi-variate case on the triangle. We say that $\Lambda = (\lambda_1, \lambda_2)$ is a *joint eigenvalue* of $J_{n,1}, J_{n,2}$ if there is a vector $\xi \neq 0$ such that $J_{n,i} \xi = \lambda_i \xi$ for $i = 1, 2$. The vector ξ is the *joint eigenvector*. If there are N distinct joint eigenvalues, then the weights and abscissa for the Gaussian quadrature are as in 1D - the nodes are the joint eigenvalues, and the weights are the square of the first component of the corresponding joint eigenvector. However, Theorem 3.7.5 of [5] states that a necessary and sufficient condition for this to occur is the following commutativity relation

$$A_{n-1,i} A_{n-1,j}^T = A_{n-1,j} A_{n-1,i}^T, \quad 1 \leq i, j \leq d \quad (2.15)$$

This is equivalent to stating that $J_{n,i}$ are a commuting family, and hence, simultaneously diagonalizable. By inspection, we see that the coefficient matrices in the recurrence for Jacobi polynomials on the triangle do not, in general, satisfy the commutativity relation. Thus, *there do not exist minimal Gaussian quadrature rules on the Triangle using the classic orthogonal polynomials*.

We need an accurate quadrature on the triangle, as the convergence of the solver depends highly on accurate representations of the right-hand-side in the modal basis. While $J_{n,i}$ may not commute, there is a connection between their spectrums and non-minimal Gaussian-like quadrature rules on $\mathcal{T}_{\text{ref}}^2$. Vioreanu and Rokhlin show in [7] that the joint eigenvalues of the combined complex operator

$$\mathcal{J}_n = \mathcal{J}_{n,1} + \iota \mathcal{J}_{n,2} \quad (2.16)$$

lie within $\mathcal{T}_{\text{ref}}^2$ (Theorem 3.4 in [7]). The algorithm described in their work uses the constant unit weight function and gives an alternate way to evaluate \mathcal{J}_n . For the case of the weighted space $L^2(\mathcal{T}, w_{a,b,c}W(x, y))$ with normalized weight (see (2.2)), the entries are

$$(\mathcal{J}_n)_{i,j} = \int_{\mathcal{T}} (x_1 + \iota x_2) P_j(x, y) P_i(x, y) w_{a,b,c} W(x, y) dx dy, \quad 0 \leq i, j \leq N-1, \quad (2.17)$$

where P_j is the j^{th} component of $\mathbf{P} = (\mathbb{P}_0^T, \mathbb{P}_1^T, \dots, \mathbb{P}_{n-1}^T)^T = (P_0^0, P_0^1, P_1^1, \dots, P_n^n)^T \in \mathbb{R}^N$. Noting the correspondence between (2.6)-(2.7) and (2.17), we see that our construction for J_n in (2.16) is precisely the same operator, except the integrals have been evaluated analytically in the formulas like (2.8). This gives us a strategy for finding near optimal quadratures on $\mathcal{T}_{\text{ref}}^2$ by way of Newton's method. We want to integrate the M polynomials up to total degree $m-1$ exactly with $m \geq n$ using only N nodes and weights. That is, with $M = \dim \Pi_{m-1}^d = \binom{m-1+d}{m-1}$, the integrals for which we want an exact quadrature are

$$\begin{aligned} \mathcal{I}_j &= \int_{\mathcal{T}_{\text{ref}}^2} P_j(x, y) w_{a,b,c} W(x, y) dx dy, \quad 0 \leq j \leq M-1 \\ &= \langle P_j, P_0 \rangle = \delta_{j0}, \end{aligned} \quad (2.18)$$

where we use mutual orthonormality under $w_{a,b,c}W(x, y)$ (2.4) to evaluate them. If $\mathbf{x}, \mathbf{y}, \mathbf{w} \in \mathbb{R}^N$ are the nodes and weights, $\mathcal{I} \in \mathbb{R}^M$ the above integrals, and $\mathbf{P}(x, y) \in \mathbb{R}^M$ our M polynomials, the objective function we aim to find a zero of is $\mathbf{F} : \mathbb{R}^{3N} \rightarrow \mathbb{R}^M$ given by

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \mathbf{P}(\mathbf{x}, \mathbf{y})\mathbf{w} - \mathcal{I}, \quad \mathcal{I} \in \mathbb{R}^M, \quad \mathbf{P}(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{M \times N} \quad (2.19)$$

For the Newton iteration

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \alpha_k [\nabla \mathbf{F}(\mathbf{z}_k)]^+ \mathbf{F}(\mathbf{z}_k), \quad (2.20)$$

the choice of initial point $\mathbf{z}_0 = (\mathbf{x}_0, \mathbf{y}_0, \mathbf{w}_0)$ and control of the step length α_k are vital for its convergence. Moreover, we'd like to incorporate inequality constraints on \mathbf{z}_k to enforce that quadrature nodes are interior to $\mathcal{T}_{\text{ref}}^2$, and we want the weights to be positive with unit sum. By casting the problem of finding roots of (2.19) into a minimization problem followed by a relaxation of \mathbf{F} , we can use techniques from constrained convex optimization to find acceptable quadratures. The optimization problem we would like to solve is

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^{3N}} \|\mathbf{F}(\mathbf{z})\|_2^2 \quad \text{such that} \quad G\mathbf{z} < g, \quad [\mathbf{0}_{1 \times 2N} \quad \mathbf{1}_{1 \times N}] \mathbf{z} = 1 \quad \text{where} \\ G = \begin{bmatrix} -I & \times & \times \\ \times & -I & \times \\ \times & \times & -I \\ I & \times & \times \\ I & I & \times \end{bmatrix} \in \mathbb{R}^{5N \times 3N}, \quad g = \begin{bmatrix} \mathbf{0}_{3N \times 1} \\ \mathbf{1}_{2N \times 1} \end{bmatrix} \in \mathbb{R}^{5M}. \end{aligned} \quad (2.21)$$

The zero entries in the constraint matrix G are denoted by \times , and I is the $N \times N$ identity matrix. Specifically, each block row of G corresponds, respectively, to the following inequality constraints: (i) $\mathbf{x} > 0$, (ii) $\mathbf{y} > 0$, (iii) $\mathbf{w} > 0$, (iv) $\mathbf{x} < 1$, (v) $\mathbf{x} + \mathbf{y} < 1$, and the equality constraint is $\sum_{i=1}^N w_i = 1$. If a local minimum \mathbf{z} is found, it is not necessarily a zero of $\mathbf{F}(\mathbf{z})$. We have only satisfied the necessary KKT conditions. However, we may be closer to an actual root and will have maintained interior nodes and positive weights. Hence, we can use this new estimate \mathbf{z} as an initial point in a Newton iteration that will relax $\mathbf{F}(\mathbf{z})$ to zero quadratically. There exist several methods

for solving the nonlinear program (2.21), such as sequential quadratic programming described by Schittkowski [8] and faithfully implemented within the `fmincon` package of MATLAB. Step length control, like that described in [9], is required to prevent $\|\mathbf{F}\|$ from blowing up at the next Newton step, and is easy to incorporate into the Newton iteration.

2.2.4 Numerical algorithm to generate Gaussian-like quadrature

Given an input total degree n , and desired total degree $m \geq n$, the following procedure generates an $N = \dim \Pi_{n-1}^d$ point quadrature capable of exactly integrating the first $M = \dim \Pi_{m-1}^d$ polynomials (up to degree $m - 1$) on $\mathcal{T}_{\text{ref}}^2$. By specializing the algorithm in [7] to the Koornwinder polynomials, we alleviate the need for an initial quadrature rule to compute the multiplication operators.

- (i) Create the Jacobi matrices as in (2.10)
- (ii) The eigenvalues of \mathcal{J}_n projected in \mathbb{R}^2 serve as an initial node set $(\mathbf{x}_0, \mathbf{y}_0)$ for the Newton iteration. The QR algorithm is well suited for finding these eigenvalues. The weights can be initialized by solving in the least-squares sense

$$\mathbf{P}(\mathbf{x}_0, \mathbf{y}_0) \mathbf{w}_0 = \mathcal{I}.$$

In words, the weights are the first column of the pseudoinverse $[\mathbf{P}(\mathbf{x}_0, \mathbf{y}_0)]^+$.

- (iii) Solve the nonlinear problem (2.21) using a sequential quadratic programming method with $\mathbf{z}_0 = (\mathbf{x}_0, \mathbf{y}_0, \mathbf{w}_0)$ as input. If desired, enforce stricter interior node conditions. This produces the current best guess for a root $\tilde{\mathbf{z}}$.
- (iv) Use $\tilde{\mathbf{z}}$ to initialize Newton's method, and iterate until $\|\mathbf{F}\|$ is comparable to machine precision.
- (v) Check if any output weights are negative. If some are, flip their sign and return to step (iii) with the modified \mathbf{z} as input. **This is just a heuristic to perturb the current state and try searching again. It only works sometimes, and the positivity check typically passes.**
- (vi) Terminate the outer iteration when $\|\mathbf{F}\|$ is within some tolerance, or total number of outer iterations (back and forth between (iii) and (v)) exceeds a maximum.

2.2.5 Quadrature verification

Given a polynomial degree n , we run the algorithm to obtain an $N = \dim \Pi_n^2$ point quadrature rule of order m and capable of exactly integrating $M = \dim \Pi_m^2$ polynomials with positive weights and interior nodes. Table 1 provides the maximal order m obtainable given the input n for the analog of the Legendre weight function on $\mathcal{T}_{\text{ref}}^2$. These are the Koornwinder polynomials with $a = b = c = 1/2$.

Table 1: For each initial polynomial degree n from 1 to 10, we list the corresponding maximum order m of a quadrature having positive weights. The number of nodes of the quadrature is $N = (n+1)(n+2)/2$ and the number of polynomials which can be integrated exactly is $M = \dim \Pi_m^2 = (m+1)(m+2)/2$. The condition number of the associated interpolation matrix is listed in the row labeled CN. We use the Legendre analog weight function (unit weight) on $\mathcal{T}_{\text{ref}}^2$, i.e with Koornwinder parameters $a = b = c = 1/2$.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m	2	4	5	7	9	11	12	14	16	17	19	21	22	23	26
N	3	6	10	15	21	28	36	45	55	66	78	91	105	120	136
M	6	15	21	36	55	78	91	120	153	171	210	253	276	300	378
CN	1.2	1.75	1.86	2.39	2.99	4.54	4.81	5.34	5.5	6.24	8.15	12.1	8.85	15.7	8.0

We obtain the same efficiency as defined and observed in [7], and the interpolation matrices are well conditioned. In Figure 5 we plot the quadrature nodes for each of the rules in Table 1, excluding the last column. Notably, all nodes are interior to $\mathcal{T}_{\text{ref}}^2$.

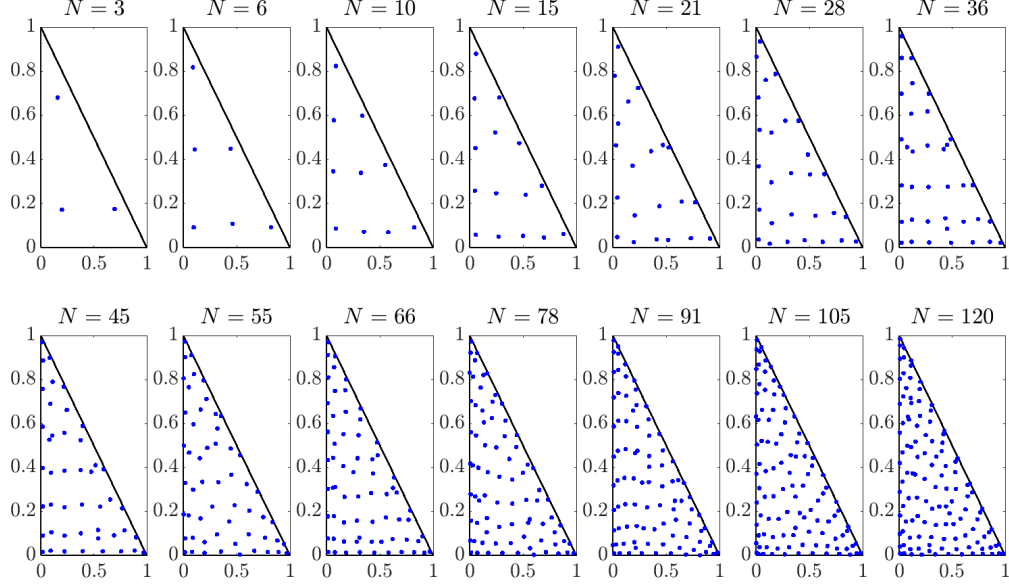


Figure 5: Quadrature nodes on $\mathcal{T}_{\text{ref}}^2$, corresponding to Table 1

In Figure 6, we plot the relative error in approximating $I = \int_{\mathcal{T}_{\text{ref}}^2} f(x, y) dx dy$ as N the size of the quadrature rule is increased. The reference integrals I are approximated with the quadrature rule of order $m = 26$ (last column in Table 1).

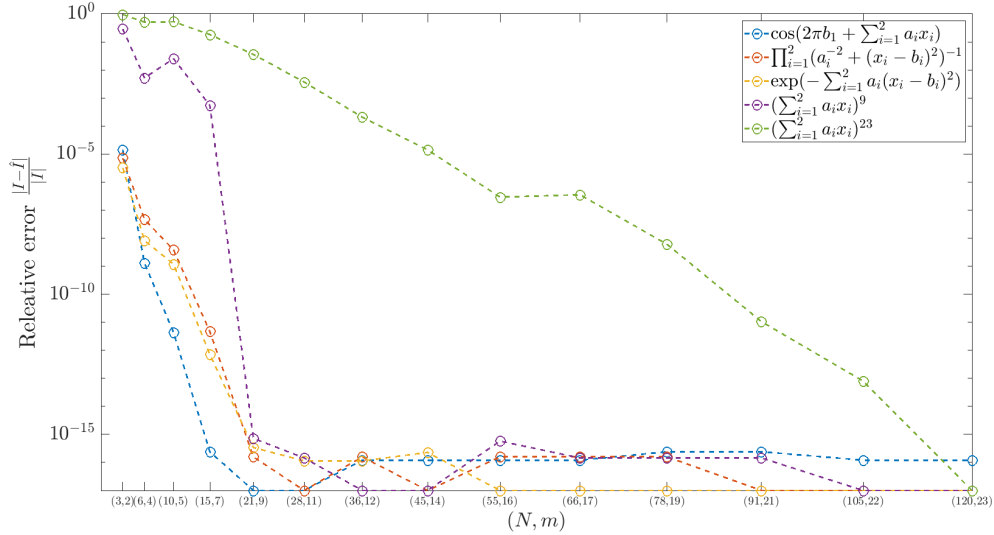


Figure 6: Relative error for integrating several test functions, including some of those due to [10], as a function of the number of quadrature nodes N and quadrature order m .

We choose several standard test functions, including a few due to Genz [10]. The function parameters $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ are drawn from the standard normal distribution. Polynomials of total degree k are seen to be integrated to near machine precision by quadrature rules of order $m \geq k$. The rules are stable as (N, m) are increased. Moreover, the decay is at least linear on a scale logarithmic only in y , which implies that the rules exhibit spectral convergence for these functions.

Though not provided here, these results hold for alternative choices of (a, b, c) . However, the interpolation matrices tend to slowly grow more ill-conditioned as (N, a, b, c) are increased, and it becomes more difficult for our algorithm to find suitable quadratures. To the extent that these quadratures are used for computing expansion coefficients, this is not a problem. We can always convert coefficients from the Legendre basis to one with larger parameters (integer difference), as described in Section 2.3.

2.2.6 Generalization for higher dimensions

On the triangle, and in fact, for most higher dimensional orthogonal polynomials, $J_{n,i}$ are not exactly jointly diagonalizable. We exploited the isomorphism between \mathbb{R}^2 and the complex plane for $d = 2$ to match the eigenvalues of $J_{n,1}, J_{n,2}$. This structure does not exist for $d > 2$. However, there is still a connection between the approximate joint eigenvalues of $J_{n,i}$ and interior points of $\mathcal{T}_{\text{ref}}^2$. To be clear, we must specify what is meant by “approximate” joint eigenvalues.

The approximate joint diagonalization problem for a set of real-valued symmetric matrices $\{C^1, \dots, C^d\}$ is to find the orthogonal matrix U such that the matrices

$$F^k = UC^kU^T, \quad 1 \leq k \leq d \quad (2.22)$$

are approximately diagonal. Alternatively, we can find the diagonal matrices F^k such that C^k are approximately equal to $U^T F^k U$. The definition of “as diagonal as possible” leads to various formulations of the diagonalization problem in terms of minimizing an objective function. For example, solving

$$\min_{\substack{U \in \mathbb{R}^N \\ U^T U = U U^T = I}} \sum_{i \neq j} (F_{ij}^k)^2 \quad (2.23)$$

is equivalent to minimizing the off-diagonal entries of F^k by orthogonal similarity transformations of C^k . On the other hand, solving

$$\min_{\substack{U, F^1, \dots, F^d \\ U^T U = U U^T = I}} \sum_{k=1}^d \|C^k - U^T F^k U\|_F^2 \quad (2.24)$$

is finding the best representation of C^k in the basis of U - a subspace fitting approach. For both problems, we require orthogonality of U , as we want it to represent the “average eigenstructure” of C^k . For the Jacobi matrices $J_{n,i}$, this requirement is natural. We want a decomposition close to the optimal quadrature case of exact joint diagonalization. In fact, (I tested this using the methods described in [11] and [12]) if the orthogonality requirement is removed, the eigenvalues (diagonals of F^k) will not necessarily be contained in $\mathcal{T}_{\text{ref}}^2$. *If we can show that, under some assumption of the diagonality measure and the orthogonality requirement for U , the resulting solution (if found) gives eigenvalues within $\mathcal{T}_{\text{ref}}^2$ (or a more general convex space), then we have a nice way to bootstrap Netwon-like methods for adaptive quadrature in d dimensions. This is how we can generalize the Vioreanu-Rokhlin quadrature to 3D.*

2.3 Ladder, lowering and promotion operators for Koornwinder polynomials

In 1D, we obtained promotion operators which could map coefficients under $P^{(a,b)}$ to those under $P^{(a+1,b+1)}$ for Jacobi polynomial $P^{(a,b)}$, and ladder operators for the derivatives. For the Koornwinder polynomials in 2D, there exist similar relationships ([2],[6]). Simultaneous promotion is obtained by composing partial promotion operators, and differentiation composes partial ladder operators with promotion operators.

Suppose we have a Koornwinder polynomial expansion of $f(x, y)$;

$$f(x, y) \approx \mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f}.$$

2.3.1 Promotion operators

There exist banded operators which allow us to increment the parameters of a Koornwinder polynomial by one. We can equivalently express the expansion as

$$\begin{aligned} f(x, y) &\approx \mathbf{P}^{(a+1,b,c)}(x, y)^T \mathcal{K}_{(a,b,c)}^{(a+1,b,c)} \mathbf{f} \\ &\approx \mathbf{P}^{(a,b+1,c)}(x, y)^T \mathcal{K}_{(a,b,c)}^{(a,b+1,c)} \mathbf{f} \\ &\approx \mathbf{P}^{(a,b,c+1)}(x, y)^T \mathcal{K}_{(a,b,c)}^{(a,b,c+1)} \mathbf{f}, \end{aligned}$$

where $\mathcal{K}_{(a,b,c)}^{(a+i,b+j,c+k)}$ is the promotion operator which maps the coefficients \mathbf{f} of $f(x, y)$ in the basis of $\mathbf{P}^{(a,b,c)}$ to those in the basis of $\mathbf{P}^{(a+i,b+j,c+k)}$. For example, to promote the coefficients of \mathbf{f} from $\mathbf{P}^{(a,b,c)}$ to $\mathbf{P}^{(a+1,b+1,c+1)}$, we would construct

$$\mathcal{K}_{(a,b,c)}^{(a+1,b+1,c+1)} = \mathcal{K}_{(a,b+1,c+1)}^{(a+1,b+1,c+1)} \mathcal{K}_{(a,b,c+1)}^{(a,b+1,c+1)} \mathcal{K}_{(a,b,c)}^{(a,b,c+1)} \quad (2.25)$$

The entries of the single-parameter promotion operator are given by the following relations of $P_{k,n}^{(a,b,c)}$, where $h_{k,n}^{(a,b,c)}$ are the normalization constants from (2.3):

$$\begin{aligned} (2n + a + b + c + 2)h_{k,n}^{(a,b,c)} P_{k,n}^{(a,b,c)} &= (n + k + a + b + c + 2)h_{k,n}^{(a+1,b,c)} P_{k,n}^{(a+1,b,c)} \\ &\quad + (n + k + b + c + 1)h_{k,n-1}^{(a+1,b,c)} P_{k,n-1}^{(a+1,b,c)} \end{aligned} \quad (2.26)$$

$$\begin{aligned} (2n + a + b + c + 2)(2k + b + c + 1)h_{k,n}^{(a,b,c)} P_{k,n}^{(a,b,c)} &= (n + k + a + b + c + 2)(k + b + c + 1)h_{k,n}^{(a,b+1,c)} P_{k,n}^{(a,b+1,c)} \\ &\quad - (n - k + a)(k + b + c + 1)h_{k,n-1}^{(a,b+1,c)} P_{k,n-1}^{(a,b+1,c)} \\ &\quad + (k + c)(n + k + b + c + 1)h_{k-1,n-1}^{(a,b+1,c)} P_{k-1,n-1}^{(a,b+1,c)} \\ &\quad - (k + c)(n - k + 1)h_{k-1,n}^{(a,b+1,c)} P_{k-1,n}^{(a,b+1,c)} \end{aligned} \quad (2.27)$$

$$\begin{aligned} (2n + a + b + c + 2)(2k + b + c + 1)h_{k,n}^{(a,b,c)} P_{k,n}^{(a,b,c)} &= (n + k + a + b + c + 2)(k + b + c + 1)h_{k,n}^{(a,b,c+1)} P_{k,n}^{(a,b,c+1)} \\ &\quad - (n - k + a)(k + b + c + 1)h_{k,n-1}^{(a,b,c+1)} P_{k,n-1}^{(a,b,c+1)} \\ &\quad - (k + b)(n + k + b + c + 1)h_{k-1,n-1}^{(a,b,c+1)} P_{k-1,n-1}^{(a,b,c+1)} \\ &\quad + (k + b)(n - k + 1)h_{k-1,n}^{(a,b,c+1)} P_{k-1,n}^{(a,b,c+1)} \end{aligned} \quad (2.28)$$

The band structures of the promotion matrices are visualized in Figure 7 for expansion order $n = 6$. Each promotion matrix is in $\mathbb{R}^{N \times N}$ where $N = \dim \Pi_n^2$. Note, $\mathcal{K}_{(a,b,c)}^{(a,b+1,c)}$ and $\mathcal{K}_{(a,b,c)}^{(a,b,c+1)}$ have the same structure.

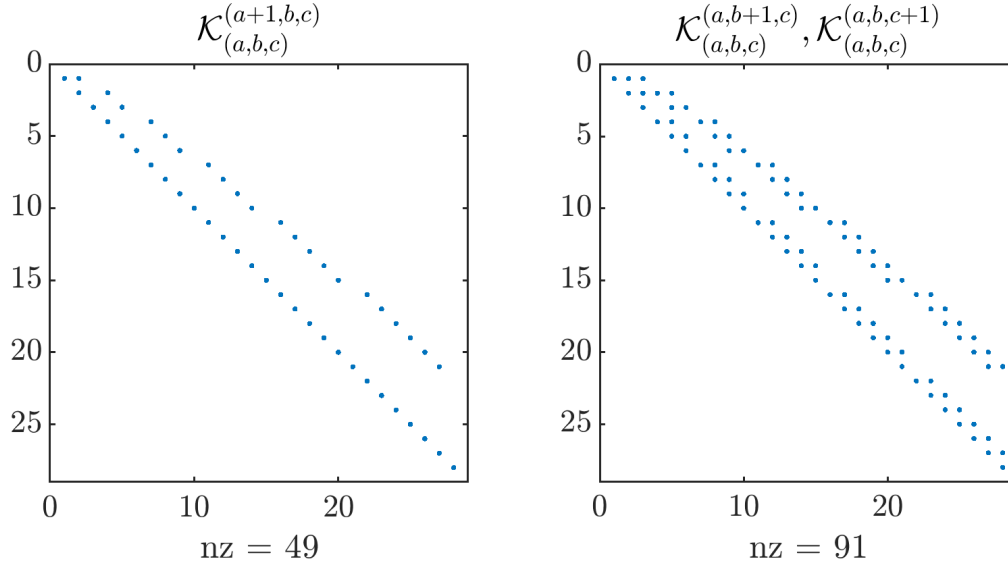


Figure 7: Sparsity and band structure of the promotion matrices for $n = 7$, $N = 28$.

To verify correctness of the implementation of these operators, we use the following procedure:

- (i) Use the highest order quadrature available (last column of [Table 1](#)) to compute the coefficients of a function $f(x, y)$, as in (1.39). That is,

$$\mathbf{c} = \mathbf{P}^{(a,b,c)}(\mathbf{x}, \mathbf{y})(f(\mathbf{x}, \mathbf{y}) \odot \mathbf{w})$$

- (ii) Generate $K_{(a,b,c)}^{(a+1,b,c)}, \mathcal{K}_{(a,b,c)}^{(a,b+1,c)}, \mathcal{K}_{(a,b,c)}^{(a,b,c+1)}$ and compute

$$\mathbf{c}_{a+1} = \mathcal{K}_{(a,b,c)}^{(a+1,b,c)} \mathbf{c}, \quad \mathbf{c}_{b+1} = \mathcal{K}_{(a,b,c)}^{(a,b+1,c)} \mathbf{c}, \quad \mathbf{c}_{c+1} = \mathcal{K}_{(a,b,c)}^{(a,b,c+1)} \mathbf{c}.$$

- (iii) Check that

$$\|\mathbf{P}^{(a+1,b,c)}(\mathbf{x}, \mathbf{y})\mathbf{c}_{a+1} - f(\mathbf{x}, \mathbf{y})\|, \quad \|\mathbf{P}^{(a,b+1,c)}(\mathbf{x}, \mathbf{y})\mathbf{c}_{b+1} - f(\mathbf{x}, \mathbf{y})\|, \quad \text{and} \quad \|\mathbf{P}^{(a,b,c+1)}(\mathbf{x}, \mathbf{y})\mathbf{c}_{c+1} - f(\mathbf{x}, \mathbf{y})\|$$

are near machine precision.

2.3.2 Ladder operators

Similar to 1D, there exist sparse relationships relating the partial derivatives of $P_{k,n}^{(a,b,c)}(x, y)$ to another Koornwinder polynomial with either the order (n, k) or parameters (a, b, c) incremented or decremented. In particular, the following recurrence relations can be used to construct partial derivatives to any order:

$$(2k + b + c + 1)h_{k,n}^{(a,b,c)} \partial_x P_{k,n}^{(a,b,c)} = (n + k + a + b + c + 2)(k + b + c + 1)h_{k,n-1}^{(a+1,b,c+1)} P_{k,n-1}^{(a+1,b,c+1)} + (k + b)(n + k + b + c + 1)h_{k-1,n-1}^{(a+1,b,c+1)} P_{k-1,n-1}^{(a+1,b,c+1)}, \quad (2.29)$$

$$h_{(k,n)}^{(a,b,c)} \partial_y P_{k,n}^{(a,b,c)} = (k + b + c + 1)h_{k-1,n-1}^{(a,b+1,c+1)} P_{k-1,n-1}^{(a,b+1,c+1)}, \quad (2.30)$$

so that there exist sparse banded matrices $\mathcal{D}_{x,(a,b,c)}^{(a+1,b,c+1)}, \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)}$ such that

$$\partial_x f(x, y) \approx \partial_x \mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f} = \mathbf{P}^{(a+1,b,c+1)}(x, y)^T \mathcal{D}_{x,(a,b,c)}^{(a+1,b,c+1)} \mathbf{f} \quad (2.31)$$

$$\partial_y f(x, y) \approx \partial_y \mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f} = \mathbf{P}^{(a,b+1,c+1)}(x, y)^T \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)} \mathbf{f} \quad (2.32)$$

2.3.3 Weighted ladder operators

The weighted basis $\tilde{\mathbf{P}}^{(a,b,c)}(x, y) = x^{a-\frac{1}{2}} y^{b-\frac{1}{2}} (1-x-y)^{c-\frac{1}{2}} \mathbf{P}^{(a,b,c)}$ will prove useful for incorporating boundary conditions to the discrete operator, while preserving its sparsity. This is accomplished using the following weighted ladder operators corresponding to differentiation of the weighted basis:

$$\begin{aligned} -(2k+b+c+1)h_{k,n}^{(a,b,c)} \partial_x \tilde{P}_{k,n}^{(a,b,c)} &= (k+c)(n-k+1)h_{k,n+1}^{(a-1,b,c-1)} \tilde{P}_{k,n+1}^{(a-1,b,c-1)} \\ &\quad + (k+1)(n-k+a)h_{k+1,n+1}^{(a-1,b,c-1)} \tilde{P}_{k+1,n+1}^{(a-1,b,c-1)} \end{aligned} \quad (2.33)$$

$$h_{k,n}^{(a,b,c)} \partial_y \tilde{P}_{k,n}^{(a,b,c)} = -(k+1)h_{k+1,n+1}^{(a,b-1,c-1)} \tilde{P}_{k+1,n+1}^{(a,b-1,c-1)}. \quad (2.34)$$

If \mathbf{f} are the coefficients of f under the weighted basis, then we have sparse banded matrices $\mathcal{W}_{x,(a,b,c)}^{(a-1,b,c-1)}, \mathcal{W}_{y,(a,b,c)}^{(a,b-1,c-1)}$ such that

$$\partial_x f(x, y) = \partial_x (\tilde{\mathbf{P}}^{(a,b,c)}(x, y)^T \mathbf{f}) = \tilde{\mathbf{P}}^{(a-1,b,c-1)}(x, y)^T \mathcal{W}_{x,(a,b,c)}^{(a-1,b,c-1)} \mathbf{f}, \quad (2.35)$$

$$\partial_y f(x, y) = \partial_y (\tilde{\mathbf{P}}^{(a,b,c)}(x, y)^T \mathbf{f}) = \tilde{\mathbf{P}}^{(a,b-1,c-1)}(x, y)^T \mathcal{W}_{y,(a,b,c)}^{(a,b-1,c-1)} \mathbf{f} \quad (2.36)$$

2.3.4 Lowering operators

Additionally, we have lowering operators which relate $xP_{k,n}^{(a,b,c)}$ to $P_{k,n}^{(a-1,b,c)}$ and $yP_{k,n}^{(a,b,c)}$ to $P_{k,n}^{(a,b-1,c)}$. These are given by the following recurrences:

$$(2n+a+b+c+2)h_{k,n}^{(a,b,c)} xP_{k,n}^{(a,b,c)} = (n-k+a)h_{k,n}^{(a-1,b,c)} P_{k,n}^{(a-1,b,c)} + (n-k+1)h_{k,n+1}^{(a-1,b,c)} P_{k,n+1}^{(a-1,b,c)} \quad (2.37)$$

$$\begin{aligned} (2k+b+c+1)(2n+a+b+c+2)h_{k,n}^{(a,b,c)} yP_{k,n}^{(a,b,c)} &= (k+b)(n+k+b+c+1)h_{k,n}^{(a,b-1,c)} P_{k,n}^{(a,b-1,c)} \\ &\quad - (k+1)(n-k+a)h_{k+1,n}^{(a,b-1,c)} P_{k+1,n}^{(a,b-1,c)} \\ &\quad - (k+b)(n-k+1)h_{k,n+1}^{(a,b-1,c)} P_{k,n+1}^{(a,b-1,c)} \\ &\quad + (k+1)(n+k+a+b+c+2)h_{k+1,n+1}^{(a,b-1,c)} P_{k+1,n+1}^{(a,b-1,c)}. \end{aligned} \quad (2.38)$$

So, there are sparse banded matrices $\mathcal{L}_{(a,b,c)}^{(a-1,b,c)}$ and $\mathcal{L}_{(a,b,c)}^{(a,b-1,c)}$ corresponding to multiplication by x and y , respectively, such that

$$xf(x, y) = x\mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f} = \mathbf{P}^{(a-1,b,c)}(x, y)^T \mathcal{L}_{(a,b,c)}^{(a-1,b,c)} \mathbf{f}, \quad (2.39)$$

$$yf(x, y) = y\mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f} = \mathbf{P}^{(a,b-1,c)}(x, y)^T \mathcal{L}_{(a,b,c)}^{(a,b-1,c)} \mathbf{f}. \quad (2.40)$$

2.4 Sparse partial differential operator on $\mathcal{T}_{\text{ref}}^2$

In order to combine the discrete partial derivative operators into the Laplacian, we must promote the coefficients of the derivatives so that they are coordinates in the same basis. To be clear, we have

$$\begin{aligned}
\partial_{xx}f(x, y) &\approx \partial_x(\partial_x \mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f}) \\
&= \partial_x(\mathbf{P}^{(a+1,b,c+1)}(x, y)^T \mathcal{D}_{x,(a,b,c)}^{(a+1,b,c+1)} \mathbf{f}) \\
&= \mathbf{P}^{(a+2,b,c+2)}(x, y)^T \mathcal{D}_{x,(a+1,b,c+1)}^{(a+2,b,c+2)} \mathcal{D}_{x,(a,b,c)}^{(a+1,b,c+1)} \mathbf{f} \\
\partial_{yy}f(x, y) &\approx \partial_y(\partial_y \mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f}) \\
&= \partial_y(\mathbf{P}^{(a,b+1,c+1)}(x, y)^T \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)} \mathbf{f}) \\
&= \mathbf{P}^{(a,b+2,c+2)}(x, y)^T \mathcal{D}_{y,(a,b+1,c+1)}^{(a,b+2,c+2)} \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)} \mathbf{f} \\
\partial_{xy}f(x, y) &\approx \partial_x(\partial_y \mathbf{P}^{(a,b,c)}(x, y)^T \mathbf{f}) \\
&= \partial_x(\mathbf{P}^{(a,b+1,c+1)}(x, y)^T \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)} \mathbf{f}) \\
&= \mathbf{P}^{(a+1,b+1,c+2)}(x, y)^T \mathcal{D}_{x,(a,b+1,c+1)}^{(a+1,b+1,c+2)} \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)} \mathbf{f}
\end{aligned}$$

Since the coefficients of $\partial_{xx}f(x, y)$ are in the span of $\mathbf{P}^{(a+2,b,c+2)}$, while those of $\partial_{yy}f(x, y)$ are in the span of $\mathbf{P}^{(a,b+2,c+2)}$, we can promote both of them to $(a+2, b+2, c+2)$. Let $\Delta_{(a,b,c)}^{(a+2,b+2,c+2)}$ denote the discrete Laplacian which takes the modes of $f(x, y)$ in the basis of $\mathbf{P}^{(a,b,c)}$ and returns the coefficients of Δf in the basis of $\mathbf{P}^{(a+2,b+2,c+2)}$. Then, we can construct the modal operator through

$$\begin{aligned}
\Delta_{(a,b,c)}^{(a+2,b+2,c+2)} &= \mathcal{K}_{(a+2,b+1,c+2)}^{(a+2,b+2,c+2)} \mathcal{K}_{(a+2,b,c+2)}^{(a+2,b+1,c+2)} \mathcal{D}_{x,(a+1,b,c+1)}^{(a+2,b,c+2)} \mathcal{D}_{x,(a,b,c)}^{(a+1,b,c+1)} \\
&\quad + \mathcal{K}_{(a+1,b+2,c+2)}^{(a+2,b+2,c+2)} \mathcal{K}_{(a,b+2,c+2)}^{(a+1,b+2,c+2)} \mathcal{D}_{y,(a,b+1,c+1)}^{(a,b+2,c+2)} \mathcal{D}_{y,(a,b,c)}^{(a,b+1,c+1)}
\end{aligned} \tag{2.41}$$

All operators have been verified using a similar approach to that presented at the end of [Section 2.3.1](#). To eliminate the rank deficiency in the modal Laplacian and construct an operator which will solve (2.1), we can use properties (i)-(iii) listed in [Section 2.1](#). By evaluating the polynomials on the boundary, we can explicitly enforce the boundary conditions. For expansion order $m-1$, we always have

$$\text{rank}(\Delta_{(a,b,c)}^{(a+2,b+2,c+2)}) = (m-2)(m-1)/2.$$

So, there are $m(m+1)/2 - (m-2)(m-1)/2 = 2m-1$ rank deficient rows. To fill in the rank deficient rows with the boundary conditions, we must first select the appropriate number of interpolation nodes for the left, bottom and hypotenuse edge of $\mathcal{T}_{\text{ref}}^2$.

- For the left and bottom edges, we take

$$N_{\text{lb}} = \left\lfloor \frac{2m-1}{2+\sqrt{2}} \right\rfloor.$$

- For the hypotenuse, we take

$$N_{\text{h}} = (2m-1) - 2N_{\text{lb}}.$$

The nodes for this task come from three 1D Gauss-Jacobi quadrature rules with parameters a, b . For example, the complete operator with Dirichlet boundary conditions can be given as

$$\tilde{\Delta}_{(a,b,c)}^{(a+2,b+2,c+2)} = \begin{bmatrix} \Delta_{(a,b,c)}^{(a+2,b+2,c+2)} \\ \mathbf{P}^{(a,b,c)}(\mathbf{x}_b, 0)^T \\ \mathbf{P}^{(a,b,c)}(0, \mathbf{x}_l)^T \\ \mathbf{P}^{(a,b,c)}(\mathbf{x}_h, 1 - \mathbf{x}_h)^T \end{bmatrix}, \quad (2.42)$$

where $\mathbf{x}_b, \mathbf{x}_l$ and \mathbf{x}_h are Gauss-Jacobi nodes shifted to the interval $(0, 1)$ for the bottom, left and hypotenuse edges of $\mathcal{T}_{\text{ref}}^2$. The process for solving (2.1) is then as follows:

- (i) Expand the right-hand side $f(x, y)$ in the basis $\mathbf{P}^{(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})}$ by first computing the coefficients in $\mathbf{P}^{(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})}$ and promoting them by composing the three single-parameter promotion matrices.
- (ii) Assemble $\tilde{\Delta}$ and solve for the coefficients \mathbf{u} of $u(x, y)$ in the basis $\mathbf{P}^{(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})}$ through

$$\mathbf{u} = \tilde{\Delta}^{-1} \begin{bmatrix} \mathbf{f} \\ g(\mathbf{x}_b, 0)^T \\ g(0, \mathbf{x}_l)^T \\ g(\mathbf{x}_h, 1 - \mathbf{x}_h)^T \end{bmatrix},$$

where \mathbf{f} are the coefficients of the right-hand side under $\mathbf{P}^{(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})}$ obtained in step (i).

- (iii) Evaluate the approximation $\hat{u}(x, y)$ using the Vandermonde matrix as

$$\hat{u}(\mathbf{x}, \mathbf{y}) = \mathbf{P}^{(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})}(\mathbf{x}, \mathbf{y})^T \mathbf{u},$$

where \mathbf{x}, \mathbf{y} are the quadrature nodes used in step (i).

For the case of zero Dirichlet boundary conditions, we can use the weighted basis $\tilde{\mathbf{P}}^{(a,b,c)}$ which evaluates to $\mathbf{0}$ on the boundary and naturally satisfies homogeneous Dirichlet conditions. We will construct the modal Laplacian Δ_W which operates on coefficients under the basis $\tilde{\mathbf{P}}^{(3/2, 3/2, 3/2)}$ and produces coefficients under $\mathbf{P}^{(3/2, 3/2, 3/2)}$. Note, *the conversion from the weighted to regular Koornwinder basis depends on our choice of parameters $a = b = c = 3/2$.*

With \mathbf{f} the coefficients of $f(x, y)$ in the weighted basis,

$$\begin{aligned} \partial_x f(x, y) &= \tilde{\mathbf{P}}^{a-1, b, c-1}(x, y)^T \mathcal{W}_{x, (a, b, c)}^{(a-1, b, c-1)} \mathbf{f} \\ &= x^{a-\frac{3}{2}} y^{b-\frac{3}{2}} (1-x-y)^{c-\frac{3}{2}} (y \mathbf{P}^{(a-1, b, c-1)}(x, y))^T (\mathcal{W}_{x, (a, b, c)}^{(a-1, b, c-1)} \mathbf{f}) \\ &= x^{a-\frac{3}{2}} y^{b-\frac{3}{2}} (1-x-y)^{c-\frac{3}{2}} \mathbf{P}^{(a-1, b-1, c-1)}(x, y)^T (\mathcal{L}_{(a-1, b, c-1)}^{(a-1, b-1, c-1)} \mathcal{W}_{x, (a, b, c)}^{(a-1, b, c-1)} \mathbf{f}), \\ \partial_y f(x, y) &= \tilde{\mathbf{P}}^{(a, b-1, c-1)}(x, y)^T \mathcal{W}_{y, (a, b, c)}^{(a, b-1, c-1)} \mathbf{f} \\ &= x^{a-\frac{3}{2}} y^{b-\frac{3}{2}} (1-x-y)^{c-\frac{3}{2}} (x \mathbf{P}^{(a, b-1, c-1)}(x, y))^T (\mathcal{W}_{y, (a, b, c)}^{(a, b-1, c-1)} \mathbf{f}) \\ &= x^{a-\frac{3}{2}} y^{b-\frac{3}{2}} (1-x-y)^{c-\frac{3}{2}} \mathbf{P}^{(a-1, b-1, c-1)}(x, y)^T (\mathcal{L}_{(a, b-1, c-1)}^{(a-1, b-1, c-1)} \mathcal{W}_{y, (a, b, c)}^{(a, b-1, c-1)} \mathbf{f}) \end{aligned}$$

When $a = b = c = 3/2$, we have

$$\begin{aligned} \partial_x f(x, y) &= \mathbf{P}^{(a-1, b-1, c-1)}(x, y)^T (\mathcal{L}_{(a-1, b, c-1)}^{(a-1, b-1, c-1)} \mathcal{W}_{x, (a, b, c)}^{(a-1, b, c-1)} \mathbf{f}), \\ \partial_y f(x, y) &= \mathbf{P}^{(a-1, b-1, c-1)}(x, y)^T (\mathcal{L}_{(a, b-1, c-1)}^{(a-1, b-1, c-1)} \mathcal{W}_{y, (a, b, c)}^{(a, b-1, c-1)} \mathbf{f}), \end{aligned}$$

so that the second partials can be constructed with the promotion and unweighted ladder operator:

$$\partial_{xx}f(x, y) = \mathbf{P}^{(a,b,c)}(x, y)^T \left(\mathcal{K}_{(a,b-1,c)}^{(a,b,c)} \mathcal{D}_{x,(a-1,b-1,c-1)}^{(a,b-1,c)} \mathcal{L}_{(a-1,b,c-1)}^{(a-1,b-1,c-1)} \mathcal{W}_{x,(a,b,c)}^{(a-1,b,c-1)} \mathbf{f} \right), \quad (2.43)$$

$$\partial_{yy}f(x, y) = \mathbf{P}^{(a,b,c)}(x, y)^T \left(\mathcal{K}_{(a-1,b,c)}^{(a,b,c)} \mathcal{D}_{y,(a-1,b-1,c-1)}^{(a-1,b,c)} \mathcal{L}_{(a,b-1,c-1)}^{(a-1,b-1,c-1)} \mathcal{W}_{y,(a,b,c)}^{(a,b-1,c-1)} \mathbf{f} \right), \quad (2.44)$$

$$\partial_{xy}f(x, y) = \mathbf{P}^{(a,b,c)}(x, y)^T \left(\mathcal{K}_{(a,b-1,c)}^{(a,b,c)} \mathcal{D}_{x,(a-1,b-1,c-1)}^{(a,b-1,c)} \mathcal{L}_{(a,b-1,c-1)}^{(a-1,b-1,c-1)} \mathcal{W}_{y,(a,b,c)}^{(a,b-1,c-1)} \mathbf{f} \right) \quad (2.45)$$

Δ_W is then given by

$$\begin{aligned} \Delta_W &= \mathcal{K}_{(a,b-1,c)}^{(a,b,c)} \mathcal{D}_{x,(a-1,b-1,c-1)}^{(a,b-1,c)} \mathcal{L}_{(a-1,b,c-1)}^{(a-1,b-1,c-1)} \mathcal{W}_{x,(a,b,c)}^{(a-1,b,c-1)} \\ &\quad + \mathcal{K}_{(a-1,b,c)}^{(a,b,c)} \mathcal{D}_{y,(a-1,b-1,c-1)}^{(a-1,b,c)} \mathcal{L}_{(a,b-1,c-1)}^{(a-1,b-1,c-1)} \mathcal{W}_{y,(a,b,c)}^{(a,b-1,c-1)} \end{aligned} \quad (2.46)$$

The matrix Δ_W has full rank, as indicated by the fill-in the last rows and columns of the sparsity pattern in Figure 8. The process for solving (2.1) with zero Dirichlet conditions is as follows:

- (i) Expand the right-hand side $f(x, y)$ in the basis $\mathbf{P}^{(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})}$ by first computing the coefficients in $\mathbf{P}^{(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})}$ and promoting them by composing the three single-parameter promotion matrices.
- (ii) Assemble Δ_W and solve for the coefficients \mathbf{u} of $u(x, y)$ in the weighted basis $\tilde{\mathbf{P}}^{(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})}$ through

$$\mathbf{u} = \Delta_W^{-1} \mathbf{f},$$

where \mathbf{f} are the coefficients of the right-hand side under $\mathbf{P}^{(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})}$ obtained in step (i).

- (iii) Evaluate the approximation $\hat{u}(x, y)$ using the weighted Vandermonde matrix as

$$\hat{u}(\mathbf{x}, \mathbf{y}) = \tilde{\mathbf{P}}^{(\frac{3}{2}, \frac{3}{2}, \frac{3}{2})}(\mathbf{x}, \mathbf{y})^T \mathbf{u},$$

where \mathbf{x}, \mathbf{y} are the quadrature nodes used in step (i).

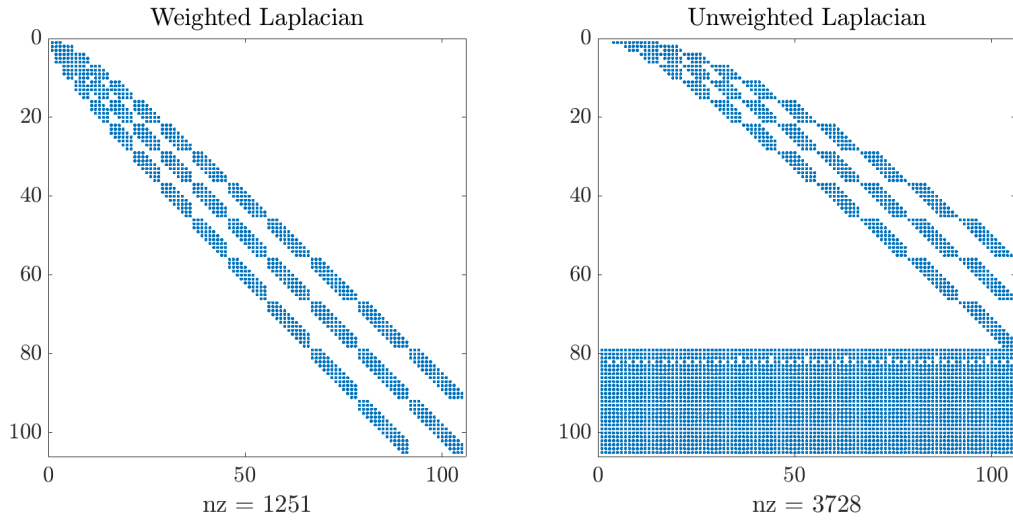


Figure 8: Sparsity and band structure of the weighted modal Laplacian (2.46) for expansion order $n = 13$.

2.5 Sparse partial differential operator on general \mathcal{T}^2

For general triangular elements \mathcal{T}^2 with vertices $(x_0, y_0), (x_1, y_1), (x_2, y_2)$, we can use the following affine transformation for $x, y \in \mathcal{T}^2$ and $r, s \in \mathcal{T}_{\text{ref}}^2$ to construct a map between \mathcal{T}^2 and $\mathcal{T}_{\text{ref}}^2$:

$$\begin{bmatrix} r \\ s \end{bmatrix} \mapsto A \begin{bmatrix} r \\ s \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad (2.47)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto A^{-1} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = \begin{bmatrix} r \\ s \end{bmatrix}, \quad (2.48)$$

where

$$A = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{bmatrix}.$$

For $(x, y) \in \mathcal{T}^2$, the Laplacian is augmented with Jacobian factors of the coordinate transformation:

$$\begin{aligned} \partial_{xx}u &= (\partial_x r)^2(\partial_{rr}u) + 2(\partial_x r)(\partial_x s)(\partial_{rs}u) + (\partial_x s)^2(\partial_{ss}u) + (\partial_{xx}r)(\partial_r u) + (\partial_{xx}s)(\partial_s u) \\ &= (\partial_x r)^2(\partial_{rr}u) + 2(\partial_x r)(\partial_x s)(\partial_{rs}u) + (\partial_x s)^2(\partial_{ss}u), \end{aligned} \quad (2.49)$$

$$\begin{aligned} \partial_{yy}u &= (\partial_y r)^2(\partial_{rr}u) + 2(\partial_y r)(\partial_y s)(\partial_{rs}u) + (\partial_y s)^2(\partial_{ss}u) + (\partial_{yy}r)(\partial_r u) + (\partial_{yy}s)(\partial_s u) \\ &= (\partial_y r)^2(\partial_{rr}u) + 2(\partial_y r)(\partial_y s)(\partial_{rs}u) + (\partial_y s)^2(\partial_{ss}u), \end{aligned} \quad (2.50)$$

from which we note we have to additionally compute the modal mixed partial operator \mathcal{D}_{rs} .

Note: We find that the condition number of the discrete operator with boundary conditions included is best controlled if the boundary nodes are distributed to the edges of \mathcal{T}^2 according to length fraction. In this way, longer sides get more nodes, and smaller sides get fewer. This avoids node coalescence when aspect ratio is large.

2.6 Numerical experiments

In Figure 9, we plot the relative solution error in solving (2.1) for several manufactured solutions (see the legend) satisfying the homogeneous Dirichlet B.C. $g(x, y) = 0$ on the reference triangle $\mathcal{T}_{\text{ref}}^2$. In the figure, $N = \dim \Pi_n^d = (n+1)(n+2)/2$ is the number of coefficients, and n is the expansion order. For each problem considered, we obtain at least one digit of accuracy as the expansion order n is increased by one. Hence, the method and implementation exhibits spectral convergence.

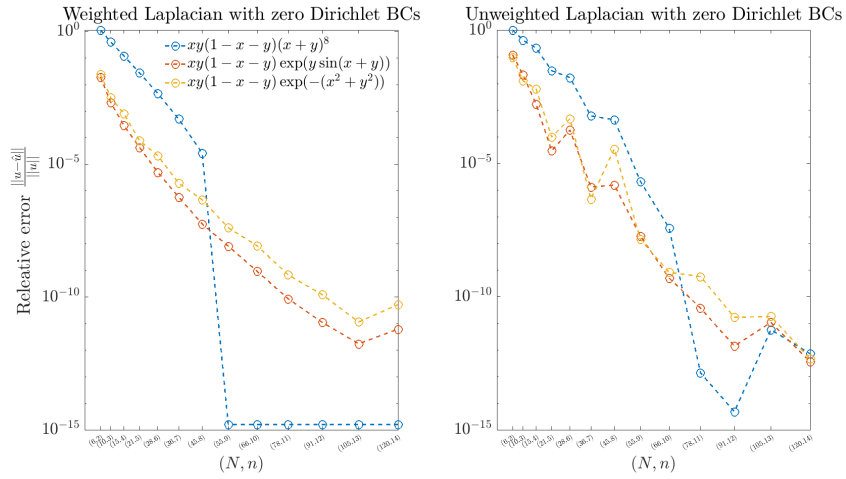


Figure 9: Relative error in solving (2.1) for several manufactured solutions u (see legend) with $g(x, y) = 0$ (homogeneous Dirichlet B.Cs) as the expansion order is increased

Now, we will consider a range of increasingly degenerate triangles by sampling the y_3 coordinate from $[0.001, 1]$ corresponding to aspect ratios between $[1, 500]$. That is, we shift the top corner of the reference triangle down, making each subsequent triangle thinner. We want to know how the methods (unweighted vs weighted) are affected by the aspect ratio of the triangle. In Figure 10, we plot the relative error in the solution to (2.1) on several (non-reference) triangles \mathcal{T} with aspect ratios between $[1, 500]$. We see that for the entire range of aspect ratios considered (very thin to normal triangles), both methods exhibit spectral convergence. The inset panels show the dependence of the conditioning of the discrete operator on the aspect ratio for a fixed order. We see that the operator grows illconditioned as the aspect ratio increases. This growth is better controlled in the weighted case.

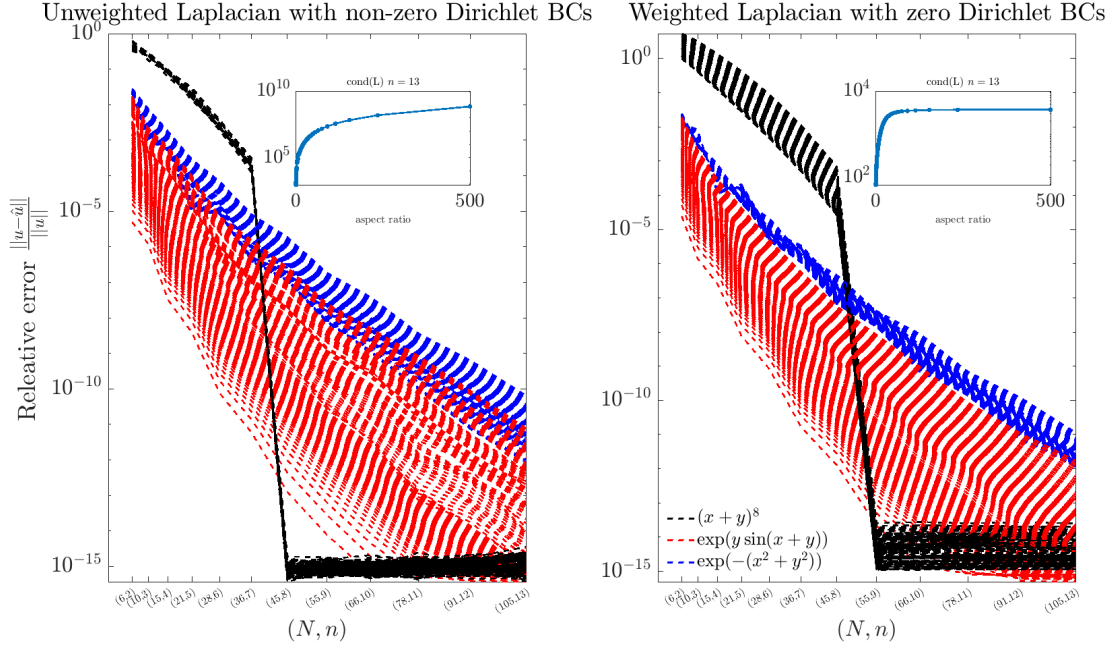


Figure 10: Relative error in solving (2.1) for several manufactured solutions u (see legend) with boundary condition $g(x, y) = u(x, y)$ for the unweighted Laplacian and $g(x, y) = 0$ for the weighted one as the expansion order is increased. Several aspect ratios between $[1, 500]$ are considered.

In Figure 11, we plot the condition number of the discrete operator as the expansion order is increased, for each aspect ratio and the weighted/unweighted methods. The weighted basis gives a much better conditioned operator, though both methods exhibit growth of the condition number in N . The width of the bands indicates control given a fixed N and growth in aspect ratio.

Hence, for the unweighted basis, to mitigate loss of accuracy as the aspect ratio increases, we need to find even better boundary nodes to keep the discrete operator reasonably conditioned as expansion order is increased.

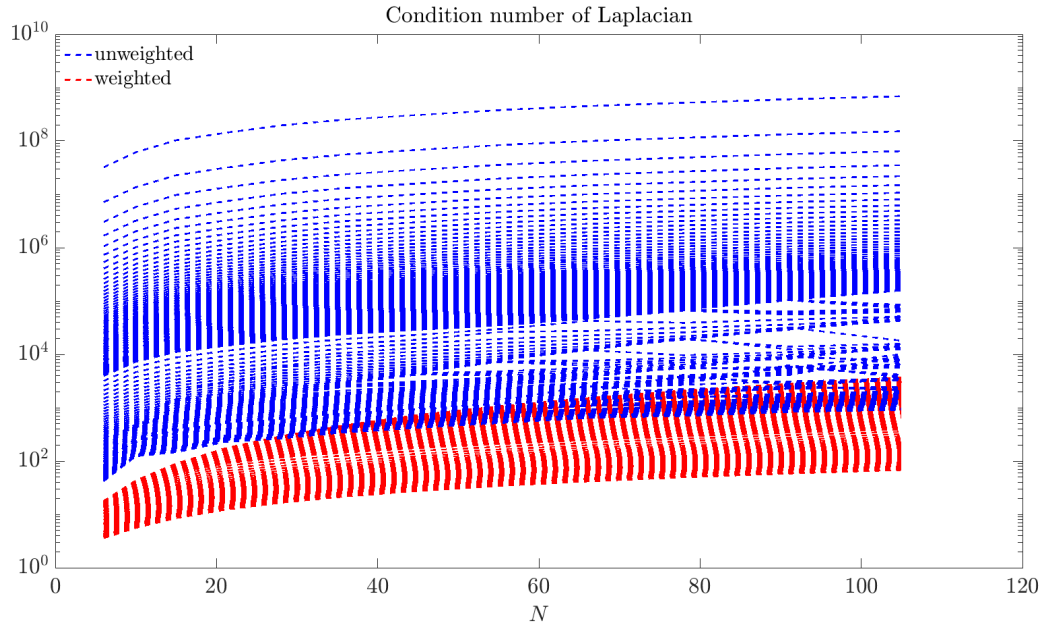


Figure 11: Condition number of weighted and unweighted discrete modal Laplacians for several expansion orders and aspect ratios.

3 General theory of orthogonal polynomials

In this section, we provide a general theory for orthogonal polynomials in 1D, and follow with the multivariable case. More details can be found in [5]. Let $X \subset \mathbb{R}^{d \geq 1}$, μ a probability measure on X , and $\{f_i(x)\}_{i \geq 1} = \mathbf{f}(x)$ a set of linearly independent functions in $L^2(X, \mu)$ arranged as a column vector. Denote the weighted L^2 inner product as $\langle f, g \rangle = \int_X f g d\mu$. Recall, the projection of $g \in L^2(X, \mu)$ onto $\text{span}\{f_i\}$ involves solving equations of the form $(Ga)_i = \langle f, f_i \rangle$, where G is the Gram matrix.

Definition 3.1. For linearly independent $\{f_i(x)\}_{i \geq 1} \in L^2(X, \mu)$, the Gram matrix G has entries

$$g_{ij} = \langle f_i, f_j \rangle.$$

Definition 3.2. For some $n \in \mathbb{N}$, let the moment determinants be given by

$$d_n = \det (g_{ij})_{i,j=1}^n, \quad (3.1)$$

with $d_0 = 1$, and define the polynomial $D_n(x)$ by

$$D_n(x) = \det \begin{bmatrix} (g_{ij})_{i,j=1}^{n-1,n} \\ \mathbf{f}(x)^T \end{bmatrix} \quad (3.2)$$

Proposition 3.1. We make the following observations:

- (i) $\text{span}\{(D_j(x))_{j=1}^n\} = \text{span}\{(f_j(x))_{j=1}^n\}$
- (ii) The functions $\{D_n(x)\}$ are orthogonal in $L^2(X, \mu)$.
- (iii) $\langle D_n, D_n \rangle = d_n d_{n-1}$.

Proof. Observe that for some $v \in \mathbb{R}^n$,

$$v^T G v = \int_X v^T \mathbf{f} \mathbf{f}^T v d\mu = \int_X \|\mathbf{f}^T v\|_2^2 d\mu \geq 0.$$

Since $\mathbf{f} = \{f_j\}_{j=1}^n$ are linearly independent, we can only have equality above if $v = 0$. Hence, $G \succ 0$, and $d_n = \det(G) > 0$. By method of minors, $D_n(x) = d_{n-1}f_n(x) + \sum_{j < n} c_j f_j(x)$ for principle minors c_j . Clearly, $\text{span}\{(D_j(x))_{j=1}^n\} = \text{span}\{(f_j(x))_{j=1}^n\}$. For orthogonality, note that

$$\langle D_n, D_j \rangle = \langle D_n, d_{j-1}f_j + \sum_{k < n} c_k f_k \rangle = d_{j-1} \langle D_n, f_j \rangle + \sum_{k < n} c_k \langle D_n, f_k \rangle.$$

Expanding $\langle D_n, f_j \rangle$, we see that

$$\langle D_n, f_j \rangle = \det \begin{bmatrix} (g_{ij})_{i,j=1}^{n-1,n} \\ \int_X f_j \mathbf{f}(x)^T d\mu \end{bmatrix} = \det \begin{bmatrix} (g_{ij})_{i,j=1}^{n-1,n} \\ (g_{jk})_{k=1}^n \end{bmatrix} = d_n \delta_{jn},$$

since for $j < n$, the second block is a dependent row. Thus, we can eliminate the sum in the orthogonality relation, yielding

$$\langle D_n, D_j \rangle = d_{j-1} d_n \delta_{jn}.$$

□

Definition 3.3. Given a member $P_n(x)$ of a family of orthogonal polynomials, the structural constant for P_n is defined as

$$h_n = \int_a^b P_n(x)^2 d\mu(x). \quad (3.3)$$

3.1 Single variable

Using the results from the previous section, we can now specialize to polynomials of a single variable $x \in [a, b]$. Let μ be a measure supported on $[a, b]$ with $\int_X |x|^n d\mu < \infty$, $\forall n$. Consider the monomial basis $\{x^{j-1}\}_{j \geq 1}$. The moment determinants are

$$g_{ij} = c_{i+j-2}, \quad c_n = \int_a^b x^n d\mu(x).$$

Since $\{D_n(x)\}$ generated with the monomial basis are orthogonal with structure factor $d_{n-1}d_n$, we can use them as our orthonormal polynomials. Let $P_{n \geq 0}(x) = D_{n+1}(x)$. Then,

$$p_{n \geq 0}(x) = (d_n d_{n+1})^{-1/2} P_n(x). \quad (3.4)$$

By construction $\langle p_n, q \rangle = 0$ for any $q \in \mathbb{P}_{k < n}$. Moreover, we can compute the leading coefficient k_n through

$$p_n(x) = (d_n d_{n+1})^{-1/2} \left(d_n x^n + \sum_{j=1}^n a_j x^{j-1} \right) \Rightarrow k_n = (d_n / d_{n+1})^{1/2} > 0, \quad (3.5)$$

where positivity of k_n follows from the fact that $d_n > 0$ (leading coefficient of normalized polynomial has the same sign as the original). The process used here to generate p_n is equivalent to applying the Gram-Schmidt process on the monomial basis with respect to the weight/measure $\mu(x)$.

Theorem 3.1. $p_{n \geq 1}(x)$ has n distinct zeros on $[a, b]$.

Proof. First, decompose $p_n(x) = q(x)r(x)$. Take $q(x) \geq 0$, and for $m \leq n$ distinct roots $\{x_j\}_{j=1}^m$ of odd multiplicity, $r(x) = \prod_{j=1}^m (x - x_j)$. Notice, $r(x)$ changes sign wherever does $p_n(x)$. Then, $r(x)p_n(x) \geq 0$ on $[a, b]$, and $\langle r, p_n \rangle > 0$. Now, if $\deg(q) > 0$, then $\deg(r) < n$. By construction of orthogonal p_n , we would have $\langle r, p_n \rangle = 0$, a contradiction. So, $\deg(q) = 0$, and $\deg(r) = m = n$. Hence, $p_{n \geq 1}$ has n simple zeros. As we will show in the next section, the zeros are also real. \square

3.1.1 Recurrence relations

Here, we provide the general recurrence relation for a family of orthogonal polynomials. Then, we apply this to the construction of Gaussian quadrature. Let $\{P_{n \geq 0}\}$ be a family of orthogonal polynomials. Note, by independence, $\int_a^b x^j P_n(x) d\mu(x) = 0 \forall j < n$. We will denote the normalized polynomials as $p_n(x) = \pm h_n^{-1/2} P_n(x)$, where the sign depends on the leading coefficient k_n of P_n .

Proposition 3.2. The following recurrence relation holds given sequence $\{A_{n \geq 0}\}, \{B_{n \geq 0}\}, \{C_{n \geq 0}\}$ and leading coefficient k_n of $P_n(x)$;

$$P_{n+1}(x) = (A_n x + B_n) P_n(x) - C_n P_{n-1}(x), \text{ where} \quad (3.6)$$

$$A_n = \frac{k_{n+1}}{k_n}, \quad B_n = -\frac{k_{n+1}}{k_n h_n} \langle x P_n, P_n \rangle, \quad C_n = \frac{k_{n+1} k_{n-1} h_n}{k_n^2 h_{n-1}}.$$

Proof. Consider the expansion of $x P_n(x)$ onto $\text{span}\{P_{j \leq n+1}(x)\}$,

$$x P_n(x) = \sum_{j=0}^{n+1} a_j P_j(x), \quad a_j = \frac{1}{h_j} \langle x P_n, P_j \rangle.$$

From orthogonality and independence, we see that $a_j = 0$ whenever $|n - j| > 1$. So,

$$\begin{aligned} xP_n(x) &= a_{n-1}P_{n-1}(x) + a_nP_n(x) + a_{n+1}P_{n+1}(x) \\ \Rightarrow P_{n+1}(x) &= \left(\frac{x}{a_{n+1}} - \frac{a_n}{a_{n+1}}\right)P_n(x) - \frac{a_{n-1}}{a_{n+1}}P_{n-1}(x). \end{aligned}$$

Matching the leading coefficients on the degree $n + 1$ term, we find

$$k_{n+1} = \frac{k_n}{a_{n+1}} \Rightarrow A_n = \frac{k_{n+1}}{k_n}.$$

Then, the coefficient on P_n is

$$B_n = -\frac{a_n}{a_{n+1}} = -\frac{k_{n+1}}{k_n h_n} \langle xP_n, P_n \rangle.$$

To determine C_n , first note that shifting the index and applying self-adjointness of multiplication by x gives

$$\begin{aligned} a_{n+1} &= \frac{1}{h_{n+1}} \langle xP_n, P_{n+1} \rangle = k_n/k_{n+1} \\ \Rightarrow \langle xP_n, P_{n-1} \rangle &= h_n k_{n-1}/k_n. \end{aligned}$$

Then,

$$C_n = \frac{a_{n-1}}{a_{n+1}} = \frac{\langle xP_n, P_{n-1} \rangle}{h_{n-1}(k_n/k_{n+1})} = \frac{k_{n+1}k_{n-1}h_n}{k_n^2 h_{n-1}}.$$

□

From the arguments in the previous proof, we can restate the recurrence as

$$xP_n(x) = \frac{k_n}{k_{n+1}}P_{n+1}(x) + b_nP_n(x) + \frac{k_{n-1}h_n}{k_n h_{n-1}}P_{n-1}(x), \quad b_n = \frac{1}{h_n} \langle xP_n, P_n \rangle \quad (3.7)$$

Specializing for orthonormal polynomials, where $h_n = 1$, we see that

$$xp_n(x) = \alpha_n p_{n+1}(x) + b_n p_n(x) + \alpha_{n-1} p_{n-1}(x), \quad \alpha_n = k_n/k_{n+1} = (d_n d_{n+1}/d_{n+1}^2)^{1/2}, \quad (3.8)$$

where we used (3.5) to reformulate α_n .

3.1.2 The Jacobi matrix

The recurrences we have provided can be thought of as a linear operation on $\mathbf{p}(x)$, the vector of orthonormal polynomials. This operator is the Jacobi matrix.

Definition 3.4. The Jacobi matrix of order n is a tridiagonal $n \times n$ matrix encoding the three-term recurrence in (3.8). That is,

$$\mathcal{J}_n = \begin{bmatrix} b_0 & a_0 & & & \\ a_0 & b_1 & a_1 & & \\ & a_1 & \ddots & \ddots & \\ & & \ddots & b_{n-2} & a_{n-2} \\ & & & a_{n-2} & b_{n-1} \end{bmatrix}. \quad (3.9)$$

With $\mathbf{p}(x) \in \mathbb{R}^n$ for each x , and \mathbf{e}_n the n^{th} basis vector, the three-term recurrence can be expressed as

$$x\mathbf{p}(x) = \mathcal{J}_n\mathbf{p}(x) + a_{n-1}p_n(x)\mathbf{e}_n. \quad (3.10)$$

In fact, due to [Favard's theorem](#), any sequence of polynomials satisfying (3.10) are orthonormal under a suitably defined measure. We can immediately state some properties on the spectrum of \mathcal{J}_n .

Theorem 3.2. Let $\{x_j\}_{j=1}^n$ be the n distinct roots of $p_n(x)$, and $\mathbf{p} = [p_0, \dots, p_{n-1}]^T$. Then,

$$\mathcal{J}_n V = V\Lambda, \quad \mathbf{V}_j = \mathbf{p}(x_j), \quad \Lambda_{jj} = x_j. \quad (3.11)$$

Furthermore, defining the diagonal weight matrix $W_{jj} = w_j = 1/\|\mathbf{p}(x_j)\|_2^2$, and $n \times n$ identity matrix I_n ,

$$VWV^T = I_n. \quad (3.12)$$

Proof. Inserting the roots x_j into (3.10) shows

$$(x_j I_n - \mathcal{J}_n)\mathbf{p}(x_j) = 0, \quad j = 1, \dots, n.$$

That is, the eigenvalues of \mathcal{J}_n are the roots x_j of $p_n(x)$, and its eigenvectors are $\mathbf{p}(x_j)$. Since \mathcal{J}_n is symmetric tridiagonal, x_j are simple and real, and $\mathbf{p}(x_j)$ are pair-wise orthogonal. So, V is an orthogonal matrix, and $W^{1/2}$ normalizes each column of V and row of V^T . \square

3.1.3 Gaussian quadrature

Noting that $p_0(x) \equiv 1$, (3.12) yields a vital implication; namely,

$$\begin{aligned} \sum_{j=0}^{n-1} w_j p_0(x_j) p_{k \geq 0}(x_j) &= \sum_{j=0}^{n-1} w_j p_{k \geq 0}(x_j) = \delta_{0k} \\ &\Rightarrow V\mathbf{w} = \mathbf{e}_1 \end{aligned} \quad (3.13)$$

which allows us to state and prove Gaussian quadrature.

Theorem 3.3. Let $\{x_j\}_{j=1}^n$ be roots of the n^{th} orthonormal polynomial $p_n(x)$ under the normalized measure μ , and $w_j = 1/\|\mathbf{p}(x_j)\|^2 > 0$. Then, for every polynomial $p \in \Pi_{2n-1}$,

$$\int_a^b p(x) d\mu(x) = \sum_{j=1}^n w_j p(x_j) \quad (3.14)$$

Proof. Consider $p \in \Pi_{2n-1}$, and decompose it as $p(x) = p_n(x)q(x) + r(x)$, where $q, r \in \Pi_{n-1}$. Expand $r(x) = \sum_{k=0}^{n-1} r_k p_k(x)$. Then, by (3.13), orthogonality, and the fact $p_n(x_j) = 0$,

$$\begin{aligned} \int_a^b p(x) d\mu(x) &= \langle p_n, q \rangle + \langle r, 1 \rangle = \sum_{k=0}^{n-1} \langle r_k p_k, 1 \rangle = r_0, \\ \sum_{j=1}^n w_j p(x_j) &= \sum_{j=1}^n w_j (p_n(x_j)q(x_j) + r(x_j)) = \sum_{j=1}^n w_j r(x_j) = \sum_{k=0}^{n-1} r_k \sum_{j=1}^n w_j p_k(x_j) = r_0. \end{aligned}$$

\square

The converse also holds. Given weights and nodes such that (3.14) is true for $p \in \Pi_{2n-1}$, then they are as in Theorem 3.3. Furthermore, if we compute the normalized eigenvectors \hat{V}_j of \mathcal{J}_n , and note $p_0(x) \equiv 1$, then the weights are given by

$$w_j = 1/\|\mathbf{p}(x_j)\|_2^2 = (\hat{V}_{0j})^2, \quad (3.15)$$

the square of the first component of the j^{th} eigenvector. Thus, we can compute both the nodes and weights for Gaussian quadrature concurrently, as shown in the Golub-Welsch algorithm [13].

Practically speaking, if we have access to a_j, b_j , we can always evaluate the eigenvalues and eigenvectors with $\mathcal{O}(n^2)$ complexity - the QR algorithm for the tridiagonal symmetric Jacobi matrix, and Clenshaw's algorithm [4] for evaluating $\mathbf{p}(x_j)$ with $\mathcal{O}(n)$ complexity for each x_j .

3.1.4 Reproducing kernel

We now end this section with the Christoffel-Darboux formula, which provides the reproducing kernel for polynomials of degree $\leq n$, and an alternate definition for the weights.

Proposition 3.3 (Christoffel-Darboux). Let k_n be the leading coefficient of orthonormal member $p_{n \geq 1}(x)$, and take $p_{-1}(x) \equiv 0$. Then,

$$\begin{aligned} \sum_{j=0}^n p_j(x)p_j(y) &= \frac{k_n}{k_{n+1}} \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x - y}, \\ \sum_{j=0}^n p_j(x)^2 &= \frac{k_n}{k_{n+1}} \left(p'_{n+1}(x)p_n(x) - p'_n(x)p_{n+1}(x) \right) \end{aligned} \quad (3.16)$$

Proof. Using (3.8), the first sum can be computed explicitly;

$$\begin{aligned} (x - y)p_j(x)p_j(y) &= \left(\frac{k_j}{k_{j+1}}p_{j+1}(x) + b_jp_j(x) + \frac{k_{j-1}}{k_j}p_{j-1}(x) \right) p_j(y) \\ &\quad - \left(\frac{k_j}{k_{j+1}}p_{j+1}(y) + b_jp_j(y) + \frac{k_{j-1}}{k_j}p_{j-1}(y) \right) p_j(x) \\ &= \frac{k_j}{k_{j+1}} (p_{j+1}(x)p_j(y) - p_{j+1}(y)p_j(x)) \\ &\quad + \frac{k_{j-1}}{k_j} (p_{j-1}(x)p_j(y) - p_{j-1}(y)p_j(x)). \end{aligned}$$

The terms telescope, giving the first formula. For the second, note that

$$\sum_{j=0}^n p_j(x)^2 = \lim_{x \rightarrow y} \sum_{j=0}^n p_j(x)p_j(y),$$

and use L'Hospital's rule. □

Theorem 3.4. For $n \geq 1$, x_j the n distinct roots of $p_n(x)$, and $w_j = 1/\|\mathbf{p}(x_j)\|_2^2$,

$$w_j = \frac{k_n}{k_{n-1}p'_n(x_j)p'_{n-1}(x_j)} \quad (3.17)$$

Proof. In the second formula of (3.16), let $x = x_j$, and note $p_n(x_j) = 0$. Then,

$$\frac{1}{w_j} = \sum_{k=0}^{n-1} p_k(x_j)^2 = \frac{k_n - 1}{k_n} (p'_n(x_j)p_{n-1}(x_j))$$

□

References

- [1] *NIST Digital Library of Mathematical Functions*. <http://dlmf.nist.gov/>, Release 1.1.4 of 2022-01-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.
- [2] Sheehan Olver, Alex Townsend, and Geoffrey Vasil. *Recurrence Relations for a Family of Orthogonal Polynomials on a Triangle*, pages 79–92. 01 2020.
- [3] Jie Shen, Yingwei Wang, and Jianlin Xia. Fast structured jacobi-jacobi transforms. *Mathematics of Computation*, 88:1, 05 2018.
- [4] C. W. Clenshaw. A note on the summation of chebyshev series. *Mathematics of Computation*, 9:118–120, 1955.
- [5] Charles F. Dunkl and Yuan Xu. *Orthogonal Polynomials of Several Variables*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 2014.
- [6] Sheehan Olver, Alex Townsend, and Geoffrey Vasil. A sparse spectral method on triangles. *SIAM Journal on Scientific Computing*, 41(6):A3728–A3756, 2019.
- [7] Bogdan Vioreanu and Vladimir Rokhlin. Spectra of multiplication operators as a numerical tool. *SIAM J. Sci. Comput.*, 36, 2014.
- [8] Klaus Schittkowski. Sequential quadratic programming methods with distributed and non-monotone line search. 2004.
- [9] Boris Polyak and Andrey Tremba. Solving underdetermined nonlinear equations by newton-like method’03 2017.
- [10] Alan Genz. Testing multidimensional integration routines. 1984.
- [11] Andreas Ziehe, Pavel Laskov, and Klaus-Robert Mller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5:777–800, 07 2004.
- [12] Jean-François Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1):161–164, January 1996.
- [13] Gene H. Golub and John H. Welsch. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23(106):221–s10, 1969.