MATH 5750/6880 COURSE NOTES
ORTHOGONAL POLYNOMIALS AND SPECTRAL METHODS FOR PDES
FALL 2018

Contents

1. Orthogonal polynomials: basic properties

1.1. **Notation.** The real numbers and natural numbers are denoted $\mathbb{R}$ and $\mathbb{N}$, respectively. We use the notation

$$\mathbb{N}_0 := \{0\} \bigcup \mathbb{N}, \qquad\qquad [N] = \{0, 1, 2, \ldots, N\} \subset \mathbb{N}_0,$$

where $N \in \mathbb{N}_0$. We will use $w : \mathbb{R} \to [0, \infty)$ to denote a positive, Borel-measurable weight function on $\mathbb{R}$. The space of univariate algebraic polynomials of degree n or less is P_n, given by

$$P_n = \operatorname{span}\left\{1, x, x^2, \ldots, x^n\right\}.$$

The notation $\int f(x)\mathrm{d}x$ denotes integration over the entire real line. If $w(x)$ has compact support, then $\int f(x)w(x)\mathrm{d}x$ is equivalent to an integral over the compact support. We will make use of w-weighted L^2 spaces, defined as

$$L^2_w := \left\{ f : \mathbb{R} \to \mathbb{R} \;\middle|\; \|f\|^2_w < \infty \right\},$$

with the inner product and norm

$$\langle f, g\rangle_w := \int f(x)g(x)w(x)\mathrm{d}x, \qquad\qquad \|f\|^2_w := \langle f, g\rangle_w .$$

1.2. **Existence and uniqueness.** Given w, we make the following assumptions:

- $w(x)$ has finite polynomial moments of all orders,

$$\int |x|^n\, w(x)\mathrm{d}x < \infty \tag{1a}$$

- For any nontrivial polynomial p on $\mathbb{R}$,

$$\int p^2(x)w(x)\mathrm{d}x > 0. \tag{1b}$$

Date: Last updated: September 8, 2018.

This condition is equivalent, for example, to the requirement

$$\int x^{2n} w(x) \mathrm{d}x > 0, \qquad n \in \mathbb{N}_0. \tag{1c}$$

Under these assumptions, a(n essentially unique) sequence of orthogonal polynomials exists.

Theorem 1.1. *If* (1) *hold, then there exist an infinite sequence of polynomials,* $p_0, p_1, \ldots,$ *where* $\deg p_j = j$*, such that* $\langle p_j, p_k \rangle_w = \delta_{j,k}$ *for* $j, k \in \mathbb{N}_0$*. These polynomials are unique up to a multiplicative sign.*

Proof. We proceed by a Gram-Schmidt induction argument. The existence and essential uniqueness of p_0 follows immediately from the $n = 0$ version of (1c). For some $n \in \mathbb{N}_0$, suppose that $\{p_0, \ldots, p_n\}$ have been determined such that

$$\langle p_j, p_k \rangle_w = \delta_{j,k}, \qquad j, k \in [n].$$

Consider $g_{n+1}(x) = x^{n+1}$. Then $\{p_0, \ldots, p_n, g_{n+1}\}$ are $n+2$ linearly independent functions; if this were not true, then we can violate (1b) by a suitable choice of p. In this case, an orthogonalization procedure can be carried out, such that

$$\widetilde{p}_{n+1}(x) = g_{n+1}(x) - \sum_{j=0}^{n} \langle g_{n+1}, p_j \rangle_w p_j(x) \in P_{n+1}.$$

Finally, we define

$$p_{n+1}(x) = \frac{\widetilde{p}_{n+1}(x)}{\|\widetilde{p}_{n+1}\|_w}.$$

By construction, we now have

$$\langle p_j, p_k \rangle_w = \delta_{j,k}, \qquad j, k \in [n+1],$$

which completes the inductive step. This procedure produces a polynomial p_{n+1} that is unique up to a sign. □

With the establishment of an orthonormal polynomial sequence, we immediately have that there exist some constants $c_{n,j}$ such that for each $n \in \mathbb{N}_0$:

$$x^n = \sum_{j=0}^{n} c_{n,j} p_j(x).$$

Furthermore, we have for any $n \in \mathbb{N}_0$:

$$\langle p_n, p \rangle_w = 0, \qquad p \in P_{n-1} \tag{2}$$

1.3. **The three-term recurrence relation.** One of the foundational results in orthogonal polynomials is their satisfaction of a three-term recurrence relation.

Theorem 1.2. *Let* $\{p_n\}_{n \in \mathbb{N}_0}$ *be a sequence of orthonormal polynomials as identified in Theorem 1.1. Then there exist constants* $(a_n, b_n) \in \mathbb{R} \times (0, \infty)$ *for* $n \in \mathbb{N}$ *such that*

$$x p_n(x) = b_n p_{n-1}(x) + a_{n+1} p_n(x) + b_{n+1} p_{n+1}(x), \qquad n \in \mathbb{N}_0, \tag{3}$$

where we define $p_{-1} \equiv 0$*. This determines the polynomials* p_j*,* $j \geq 1$*, and we define* $b_0 = 1/p_0$*.*

Proof. The relation is clearly true for $n = 0$ for some numbers a_1, b_1. Now fix $n \geq 1$. First, we note that clearly there are constants a_{n+1}, b_{n+1}, c_{n+1} and $d_{n,j}$ so that

$$xp_n(x) = a_{n+1}p_n(x) + b_{n+1}p_{n+1}(x) + c_{n+1}p_{n-1}(x) + \sum_{j=0}^{n-2} d_{n,j}p_j(x).$$

We have

$$\langle xp_n, p_k \rangle_k = \begin{cases} a_{n+1}, & k = n \\ b_{n+1}, & k = n+1 \\ c_{n+1}, & k = n-1, \\ d_{n,j}, & k = j \in [n-2]. \end{cases}$$

Multiplication by x is self-adjoint in L^2_w; therefore for any $j \in [n-2]$, we have

$$\langle xp_n, p_j \rangle_w = \langle p_n, xp_j \rangle_w \overset{(2)}{=} 0.$$

This shows that all the $d_{n,j}$ coefficients vanish. Furthermore,

$$c_{n+1} = \langle xp_n, p_{n-1} \rangle_w = \langle xp_{n-1}, p_n \rangle_w = b_n.$$

This shows the formula (3). By this construction, the b_n must be positive: the construction in Theorem 1.1 shows that the leading coefficient of each p_n is positive. By inspection of (3), this equation cannot hold unless b_{n+1} is positive for all n. □

1.4. **The Christoffel-Darboux identity.** One useful relation for orthogonal polynomials is the following identity.

Theorem 1.3 (Christoffel-Darboux Identity)**.** *Let $\{p_n\}_{n \in \mathbb{N}_0}$ be an orthonormal polynomial family with recurrence coefficients (a_n, b_n). Then for all $n \in \mathbb{N}_0$:*

$$\sum_{j=0}^{n} p_j(x)p_j(y) = b_{n+1}\frac{p_{n+1}(x)p_n(y) - p_{n+1}(y)p_n(x)}{x-y} \qquad x \neq y \tag{4a}$$

$$\sum_{j=0}^{n} p_j^2(x) = b_{n+1}\left[p'_{n+1}(x)p_n(x) - p_{n+1}(x)p'_n(x)\right] \tag{4b}$$

Proof. The result is a fairly straightforward proof by induction. For $n = 0$ and $x \neq y$, we have

$$\begin{aligned} p_0(x)p_0(y) = p_0^2 = \frac{(x-y)p_0^2}{x-y} &= \frac{[(x-a_1)p_0]\,p_0 - [(y-a_0)p_0]\,p_0}{(x-y)} \\ &= \frac{b_1p_1(x)p_0(y) - b_1p_1(y)p_0(x)}{x-y}, \end{aligned}$$

proving the relation for $n = 0$. Now if the relation holds for a value n, then

$$\sum_{j=0}^{n+1} p_j(x)p_j(y) = \frac{b_{n+1}p_{n+1}(x)p_n(y) - b_{n+1}p_{n+1}(y)p_n(x)}{x-y} + p_{n+1}(x)p_{n+1}(y)$$

$$= \frac{b_{n+1}p_{n+1}(x)p_n(y) - b_{n+1}p_{n+1}(y)p_n(x) + xp_{n+1}(x)p_{n+1}(y) - yp_{n+1}(y)p_{n+1}(y)}{x-y} \tag{5}$$

We now use the three-term recurrence relation for the numerator on the right-hand side,

$$\begin{aligned}
&b_{n+1}p_{n+1}(x)p_n(y) - b_{n+1}p_{n+1}(y)p_n(x) + xp_{n+1}(x)p_{n+1}(y) - yp_{n+1}(y)p_{n+1}(x) \\
=&b_{n+1}p_{n+1}(x)p_n(y) - b_{n+1}p_{n+1}(y)p_n(x) + p_{n+1}(y)\left[b_{n+2}p_{n+2}(x) + a_{n+2}p_{n+1}(x) + b_{n+1}p_n(x)\right] \\
&- p_{n+1}(x)\left[b_{n+2}p_{n+2}(y) + a_{n+2}p_{n+1}(y) + b_{n+1}p_n(y)\right] \\
=&b_{n+2}p_{n+2}(x)p_{n+1}(y) - b_{n+2}p_{n+2}(y)p_{n+1}(x)
\end{aligned}$$

Combining this with (5) completes the inductive proof. For relation (4b) one can, e.g., use L'Hôpital's rule on (4a) as $x \to y$. □

1.4.1. *Examples.* We list here a few examples of well-known orthogonal polynomial families and their associated recurrence coefficients. For general weight functions computing these coefficients is a difficult problem with no panacea. We will see later how the coefficients for these families are computed analytically.

- *Legendre polynomials* – We define

$$w(x) = \begin{cases} 1, & x \in [-1,1] \\ 0, & x \in \mathbb{R}\backslash[-1,1] \end{cases} \tag{6a}$$

The associated family of polynomials $\{p_n\}_{n=0}^{\infty}$ have recurrence coefficients given by

$$b_0 = \frac{1}{\sqrt{2}}, \qquad a_n = 0, \qquad b_n = \frac{n}{\sqrt{4n^2-1}}, \qquad n \in \mathbb{N} \tag{6b}$$

- *Hermite polynomials* – We define

$$w(x) = \exp(-x^2), \qquad x \in \mathbb{R} \tag{7a}$$

The associated family of polynomials $\{p_n\}_{n=0}^{\infty}$ have recurrence coefficients given by

$$b_0 = \frac{1}{\pi^{1/4}}, \qquad a_n = 0, \qquad b_n = \sqrt{\frac{n}{2}} \tag{7b}$$

2. Direct consequences of the three-term recurrence

2.1. **Derivatives.** Differentiation of (3) yields a recurrence formula for derivatives of orthogonal polynomials:

$$xp_n'(x) = b_n p_{n-1}'(x) + a_{n+1}p_n'(x) + b_{n+1}p_{n+1}'(x) - p_n(x), \qquad n \in \mathbb{N}, \tag{8}$$

with the initial conditions $p_0' = 0$ and $p_1' = c_{1,1} = 1/(b_0 b_1)$. This procedure can be repeated, resulting in a general recurrence formula for the d'th derivative of p_n:

$$\begin{aligned}
xp_n^{(d)}(x) &= b_n p_{n-1}^{(d)}(x) + a_{n+1}p_n^{(d)}(x) + b_{n+1}p_{n+1}^{(d)}(x) - dp_n^{(d-1)}(x), && n \geq d, \\
p_d^{(d)}(x) &= \frac{d!}{\prod_{j=0}^{d} b_j}, \\
p_j^{(d)}(x) &= 0, && j < d.
\end{aligned}$$

Note that this yields a simple algorithm to compute derivatives of orthogonal polynomials of any order: one needs only first compute lower-order derivatives first.

2.2. **Alternative normalizations.** One will encounter many normalizations for orthogonal polynomials in the literature. Given a sequence of positive numbers $\{h_n\}_{n=0}^{\infty}$, we can define a new sequence of polynomials,

$$q_n = h_n p_n.$$

The polynomial family $\{q_n\}_{n=0}^{\infty}$ is still an L^2_w-orthogonal family, but no longer orthonormal in general. A simple exercise shows that these new polynomials satisfy a three-term recurrence relation whose coefficients can be expressed in terms of the original coefficients (a_n, b_n) for $n \geq 1$:

$$x q_n(x) = A_{n+1} q_n(x) + B_{n+1} q_{n+1}(x) + C_{n+1} q_{n-1}(x),$$
$$A_{n+1} = a_{n+1}, \quad B_{n+1} = b_{n+1}\frac{h_n}{h_{n+1}}, \quad C_{n+1} = \frac{b_n h_n}{h_{n-1}}.$$

The reverse map may also computed so that, given $(A_n, B_n, C_n)_{n\geq 1}$ and h_0, one may compute the orthonormal recurrence coefficients,

$$a_n = A_n, \qquad b_n = \sqrt{B_n C_{n+1}}, \qquad n \in \mathbb{N},$$

and $b_0 = 1/p_0 = h_0/q_0$. The normalization coefficients can likewise be recovered sequentially:

$$h_n = h_{n-1}\sqrt{\frac{C_{n+1}}{B_n}}.$$

2.3. **Affine mappings.** Let $w(x)$ be a weight function with an associated orthonormal polynomial family $\{p_n\}_{n\in\mathbb{N}_0}$ and known recurrence coefficients (a_n, b_n). Suppose we are given a bijective affine map $A : \mathbb{R} \to \mathbb{R}$ of the form

$$A(x) = bx + a, \qquad a, b \in \mathbb{R},$$

where a and b are arbitrary ($b \neq 0$) but fixed. It is frequently desireable to generate a new sequence of polynomials that are orthonormal under the same weight function composed with A, i.e., to find polynomials $\{\pi_n\}_{n=0}^{\infty}$ satisfying

$$\int \pi_n(x)\pi_m(x) w(A(x))\mathrm{d}x = \delta_{m,n}, \qquad m, n \in \mathbb{N}_0. \tag{9}$$

As usual, finding recurrence coefficients is the goal so that we can perform computations with the π_n. Since the π_n are orthonormal polynomials, then they have their own set of recurrence coefficients $(\alpha_n, \beta_n)_{n\in\mathbb{N}}$:

$$x\pi_n(x) = \beta_{n+1}\pi_{n+1}(x) + \alpha_{n+1}\pi_n(x) + \beta_n \pi_{n-1}(x), \qquad n \in \mathbb{N}, \tag{10}$$

with $\pi_0 \equiv 1/\beta_0$. The goal is to compute the unknown (α_n, β_n) recurrence coefficients for π_n from the known (a_n, b_n) recurrence coefficients for p_n. As before, β_0 has an explicit definition:

$$\beta_0^2 = \int w(A(x))\mathrm{d}x \overset{u=A(x)}{=} \frac{1}{|b|}\int w(u)\mathrm{d}u = \frac{b_0^2}{|b|},$$

which yields $\beta_0 = b_0/\sqrt{|b|}$. To determine the rest of the recurrence coefficients, we first observe that

$$\int p_n(x)p_m(x)w(x)\mathrm{d}x = \delta_{m,n}$$

$$\Updownarrow \; (x = A(u))$$

$$|b| \int p_n(A(u))p_m(A(u))w(A(u))\mathrm{d}u = \delta_{n,m}.$$

Since A is affine, then $p_n(A(x))$ is a polynomial of degree n. Thus, if we define

$$\pi_n(x) \coloneqq \sqrt{|b|}(\operatorname{sign} b)^n p_n(A(x)),$$

then $\{\pi_n\}_{n\in\mathbb{N}_0}$ satisfies (9) and has positive leading coefficient. We now determine which recurrence equation the π_n satisfy. Evaluating (3) at $x \leftarrow A(x)$, we have

$$(bx + a)p_n(A(x)) = b_{n+1}p_{n+1}(A(x)) + a_{n+1}p_n(A(x)) + b_n p_{n-1}(A(x)),$$

$$xp_n(A(x)) = \frac{b_{n+1}}{b}p_{n+1}(A(x)) + \frac{a_{n+1} - a}{b}p_n(A(x)) + \frac{b_n}{b}p_{n-1}(A(x))$$

Multiplying the last equation above by $\sqrt{|b|}(\operatorname{sign} b)^n$ and using $|b| = b \operatorname{sign} b = b/(\operatorname{sign} b)$ for $b \neq 0$, we have

$$x\pi_n(x) = \frac{b_{n+1}}{|b|}\pi_{n+1}(x) + \frac{a_{n+1} - a}{b}\pi_n(x) + \frac{b_n}{|b|}\pi_{n-1}(x)$$

By matching the above equation with (10), we conclude:

$$\beta_0 = \frac{b_0}{\sqrt{|b|}}, \qquad \beta_n = \frac{b_n}{|b|} \qquad \alpha_n = \frac{a_n - a}{b}, \quad (n \in \mathbb{N}). \tag{11}$$

We have therefore shown the following result.

Theorem 2.1. *Assume $w(x)$ is a weight function with an orthonormal polynomial family $\{p_n\}_{n\in\mathbb{N}_0}$ and recurrence coefficients (a_n, b_n). Let $A(x) = bx + a$ be an invertible affine map ($a \in \mathbb{R}$, $b \in \mathbb{R}\backslash\{0\}$). If π_n are the unique polynomials with positive leading coefficient that are orthonormal under $w(A(x))$, then they satisfy the three-term recurrence* (10) *with coefficients α_n and β_n given by* (11).

As an example, we consider the problem of computing recurrence coefficients for the so-called *shifted* Legendre polynomials. These are polynomials π_n that satisfy the orthogonality relation

$$\int_0^1 \pi_n(x)\pi_m(x)\mathrm{d}x = \delta_{m,n}.$$

That is, they are polynomials orthogonal under the weight function $\omega(x)$,

$$\omega(x) = \begin{cases} 1, & x \in [0, 1] \\ 0, & x \in \mathbb{R}\backslash[0, 1] \end{cases}$$

Note that $\omega(x)$ can be expressed as the composition of $w(x)$ for Legendre polynomials with an affine map. Letting $w(x)$ be as in (6a), we have

$$\omega(x) = w(A(x)), \qquad A(x) = 2x - 1.$$

Therefore, let $b = 2$, $a = -1$ be the coefficients of the affine map A. Since the Legendre polynomials have recurrence coefficients (a_n, b_n) defined in (6b), then by theorem 2.1 the shifted Legendre polynomials satisfy a three-term recurrence relation of the form (10) with coefficients

$$\alpha_n = \frac{a_n - a}{b} = \frac{1}{2}, \qquad\qquad \beta_n = \frac{b_n}{|b|} = \frac{n}{2\sqrt{4n^2-1}},$$

for $n \geq 1$, with $\beta_0 = b_0/\sqrt{|b|} = \frac{1}{2}$.

2.4. **General connection coefficients.** Let $w(x)$ be a weight function with orthonormal polynomial family $\{p_n\}_{n\in\mathbb{N}_0}$ and recurrence coefficients (a_n, b_n). Let $\omega(x)$ be another weight function with orthonormal polynomial family $\{\pi_n\}_{n\in\mathbb{N}_0}$ and recurrence coefficients (α_n, β_n). There is a unique triangular array of numbers $d_{n,j}$ defined as

$$p_n(x) = \sum_{j=0}^{n} d_{n,j}\pi_j(x) \tag{12}$$

These numbers are called *connection coefficients*. Connection coefficients between two polynomial families can be computed using the recurrence coefficients from both families. Note that

$$p_0 = 1/b_0, \quad \pi_0 = 1/\beta_0 \qquad \Longrightarrow \qquad d_{0,0} = \frac{\beta_0}{b_0}. \tag{13}$$

For convenience in what follows we define $d_{n,j}$ outside the triangular constraints $0 \leq j \leq n$ to be 0:

$$d_{-1,j} = 0, \quad (j \geq -1), \qquad\qquad d_{n,j} = 0, \quad (j > n)$$

We use the recurrence relations for the families to express $d_{n+1,j}$ in terms of $d_{n,j}$ and $d_{n,j-1}$. First note from (12) that

$$d_{n,j} = \langle p_n, \pi_j \rangle_\omega . \tag{14}$$

Now multiplying (12) by $x\pi_k$ and integrating with respect the weight $\omega(x)$, we have

$$\langle xp_n, \pi_k \rangle_\omega = \sum_{j=0}^{n} d_{n,j} \langle x\pi_j, \pi_k \rangle_\omega ,$$

$$\Downarrow \text{(three-term recurrence)}$$

$$\langle b_{n+1}p_{n+1} + a_{n+1}p_n + b_n p_{n-1}, \pi_k \rangle_\omega = \sum_{j=0}^{n} d_{n,j} \langle \beta_{j+1}\pi_{j+1} + \alpha_j\pi_j + \beta_j\pi_{j-1} \rangle_\omega$$

Using the L^2_ω-orthogonality of the π_j and (14), the above results in the equation

$$b_{n+1}d_{n+1,k} = -a_{n+1}d_{n,k} - b_n d_{n-1,k} + \sum_{j=0}^{n} d_{n,j}\left[\beta_{j+1}\delta_{j+1,k} + \alpha_{j+1}\delta_{j,k} + \beta_j\delta_{j-1,k}\right],$$

where $\delta_{j,k}$ is the Kronecker delta function. For $n \geq 0$, we can now express $d_{n+1,k}$ in terms of $d_{n,k}$ and $d_{n-1,k}$:

$$\begin{aligned} d_{n+1,0} &= \frac{1}{b_{n+1}}\left[(\alpha_1 - a_{n+1})d_{n,0} + \beta_1 d_{n,1} - b_n d_{n-1,0}\right], \\ d_{n+1,k} &= \frac{1}{b_{n+1}}\left[(\alpha_{k+1} - a_{n+1})d_{n,k} + \beta_k d_{n,k-1} + \beta_{k+1} d_{n,k+1} - b_n d_{n-1,k}\right], \quad 1 \leq k \leq n-1 \\ d_{n+1,n} &= \frac{1}{b_{n+1}}\left[(\alpha_{n+1} - a_{n+1})d_{n,n} + \beta_n d_{n,n-1}\right], \\ d_{n+1,n+1} &= \frac{\beta_{n+1}}{b_{n+1}} d_{n,n}, \end{aligned}$$

Pairing the equations above with the $n = 0$ initial condition (13) yields a simple algorithm for computing the connection coefficients in (12).

2.5. **The canonical connection.** It is occasionally helpful to have coefficients for an expansion of p_n in monomials. The goal of this section is to compute these coefficients in terms of the recurrence coefficients for the orthonormal family. We define these monomial expansion coefficients as the triangular array of numbers $c_{n,j}$ defined as

$$p_n(x) = \sum_{j=0}^{n} c_{n,j} x^j, \tag{15}$$

and refer to this equality the connection to the canonical (monomial) basis. For $n = 0$, we have $c_{0,0} = p_0 = 1/b_0$, and for convenience define $c_{-1,j} = 0$ for all j. A manipulation of the three-term recurrence relation (3) yields

$$p_{n+1}(x) = \frac{1}{b_{n+1}}\left[(x - a_{n+1})p_n(x) - b_n p_{n-1}(x)\right],$$

and using (15) we obtain

$$\begin{aligned} p_{n+1}(x) = \frac{1}{b_{n+1}}\Big[&\left(-a_{n+1}c_{n,0} - b_n c_{n-1,0}\right) x^0 + c_{n,n}x^{n+1} + \left(c_{n,n-1} - a_{n+1}c_{n,n}\right) x^n \\ &+ \sum_{j=1}^{n-1} \left(-a_{n+1}c_{n,j} - b_n c_{n-1,j} + c_{n,j-1}\right) x^j \Big]. \end{aligned}$$

This therefore provides the following recurrence for computing the monomial expansion coefficients $c_{n,j}$ from the recurrence coefficients. Start with the initial conditions:

$$\begin{aligned} &c_{0,0} = 1/b_0, \\ &c_{1,1} = c_{0,0}/b_1, \qquad\qquad c_{1,0} = \frac{1}{b_1}\left(-a_1 c_{n,0}\right), \end{aligned}$$

and use the following recurrence to compute $c_{n+1,j}$ from $c_{n,j}$ and $c_{n-1,j}$:

$$c_{n+1,j} = \begin{cases} (-a_{n+1}c_{n,0} - b_n c_{n-1,0})/b_{n+1}, & j = 0 \\ (c_{n,j-1} - a_{n+1}c_{n,j} - b_n c_{n-1,j})/b_{n+1}, & 1 \leq j \leq n-1 \\ (c_{n,n-1} - a_{n+1}c_{n,n})/b_{n+1}, & j = n \\ c_{n,n}/b_{n+1}, & j = n+1 \end{cases}$$

Note in particular that we have an explicit form for the leading coefficients $c_{n,n}$ of p_n:

$$c_{n,n} = \frac{1}{\prod_{j=0}^{n} b_j}, \qquad\qquad p_n(x) = c_{n,n}x^n + \cdots \tag{16}$$

The related expansion that accompanies (15) is

$$x^n = \sum_{j=0}^{n} C_{n,j} p_j(x). \tag{17}$$

First we note that the arrays $\{c_{n,j}\}_{n,j}$ and $\{C_{n,j}\}_{n,j}$ are related. Define $N \times N$ matrices $\boldsymbol{C}$ and $\boldsymbol{c}$ with entries:

$$(\boldsymbol{C})_{n,j} = C_{n,j}, \qquad (\boldsymbol{c})_{n,j} = c_{n,j}, \qquad 0 \le j \le n \le N-1$$

Now let $x \in \mathbb{R}$ be arbitrary and note that

$$\begin{pmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{N-1}(x) \end{pmatrix} = \boldsymbol{c} \begin{pmatrix} x^0 \\ x^1 \\ \vdots \\ x^{N-1} \end{pmatrix}, \qquad \begin{pmatrix} x^0 \\ x^1 \\ \vdots \\ x^{N-1} \end{pmatrix} = \boldsymbol{C} \begin{pmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{N-1}(x) \end{pmatrix}.$$

Thus, $\boldsymbol{C} = \boldsymbol{c}^{-1}$ and this provides one way to compute $\boldsymbol{C}$. For completeness, we provide another strategy that uses a recurrence formula to compute the entries of $\boldsymbol{C}$.

First we note that through explicit computation we have

$$C_{0,0} = b_0, \qquad C_{1,0} = a_1 b_0, \qquad C_{1,1} = b_0 b_1$$

Subsequently, we compute

$$\begin{aligned} C_{n,j} &\overset{(17)}{=} \langle x^n, p_j \rangle_w = \left\langle x^{n-1}, x p_j \right\rangle_w \\ &\overset{(3)}{=} b_{j+1} \left\langle x^{n-1}, p_{j+1} \right\rangle_w + a_{j+1} \left\langle x^{n-1}, p_j \right\rangle_w + b_j \left\langle x^{n-1}, p_{j-1} \right\rangle_w \\ &= b_{j+1} C_{n-1,j+1} + a_{j+1} C_{n-1,j} + b_j C_{n-1,j-1}, \end{aligned}$$

which provides a recursion that can be used to compute the coefficients $C_{n,j}$.

2.6. **Derivative expansions.** We consider the task of computing the coefficients $c_{n,j}^{(d)}$ satisfying

$$p_n^{(d)}(x) = \sum_{j=0}^{n-d} c_{n,j}^{(d)} p_j(x). \tag{18}$$

As before, our strategy for a fixed d will entail iteration over the index n. First we note from (16) that

$$p_d^{(d)} = \frac{d!}{\prod_{j=0}^{d} b_j}, \qquad\qquad p_n^{(d)} = 0, \quad (n < d).$$

Noting from (18) that $c_{n,j}^{(d)} = \left\langle p_n^{(d)}, p_j \right\rangle_w$, this provides the starting condition

$$c_{d,0}^{(d)} = \frac{d!}{\prod_{j=1}^{d} b_j}, \qquad\qquad c_{n,j}^{(d)} = 0, \quad (d > n) \tag{19}$$

Thus for $n \geq d$ we have

$$\begin{aligned}\left\langle x p_n^{(d)}, p_j \right\rangle_w &\overset{(8)}{=} b_{n+1} c_{n+1,j}^{(d)} + a_{n+1} c_{n,j}^{(d)} + b_n c_{n-1,j}^{(d)} - d c_{n,j}^{(d-1)}, \\ \left\langle x p_j, p_n^{(d)} \right\rangle_w &\overset{(3)}{=} b_{j+1} c_{n,j+1}^{(d)} + a_{j+1} c_{n,j}^{(d)} + b_j c_{n,j-1}^{(d)}.\end{aligned}$$

These two expressions must be equal, so setting the right-hand sides to be equal and solving for $c_{n+1,j}^{(d)}$ yields the iteration

$$c_{n+1,j}^{(d)} = \frac{1}{b_{n+1}} \left[b_{j+1} c_{n,j+1}^{(d)} + (a_{j+1} - a_{n+1}) c_{n,j}^{(d)} + b_j c_{n,j-1}^{(d)} - b_n c_{n-1,j}^{(d)} + d c_{n,j}^{(d-1)} \right]$$

Therefore, to compute the numbers $c_{n,j}^{(d)}$, one must first have $c_{n,j}^{(d-1)}$ and use the initial conditions (19). Subsequently, one can iterate on n via the above equation. Computing the initial $d = 0$ values is straightforward since

$$c_{n,j}^{(0)} = \langle p_n, p_j \rangle_w = \delta_{n,j}.$$

3. Quadrature and zeros of orthogonal polynomials

There is a very strong characterization of roots of orthogonal polynomials.

Theorem 3.1. *Let $\{p_n\}_{n \in \mathbb{N}_0}$ be a sequence of orthogonal polynomials. Then, for all $n \geq 1$:*

1. *p_n has n real-valued roots*
2. *The roots of p_n are all simple*
3. *p_n and p_{n+1} have no common roots*

Proof. Fix $n \geq 1$. Suppose that p_n has exactly r real-valued roots, $x_1, \ldots, x_r$, $0 \leq r \leq n-1$. Consider the polynomial

$$q(x) = p_n(x) \prod_{j=1}^{r} (x - x_j)$$

Then $q(x)$ is a non-trivial polynomial that is single-signed on $\mathbb{R}$. Therefore:

$$0 \neq \int q(x) w(x) \mathrm{d}x = \int p_n(x) \prod_{j=1}^{r} (x - x_j) w(x) \mathrm{d}x = \left\langle p_n, \prod_{j=1}^{r} (x - x_j) \right\rangle_w = 0,$$

where the last equality holds by orthogonality of p_n and since $r < n$. This contradiction shows that $r = n$, and therefore p_n has exactly n real-valued roots.

We can similarly show that all roots of p_n are simple. Suppose that there is a repeated root, x_0 of p_n. I.e.,

$$p_n(x) = \kappa_n (x - x_0)^2 \prod_{j=1}^{n-2} (x - x_j).$$

Then consider the polynomial

$$q(x) = p_n(x) \prod_{j=1}^{n-2} (x - x_j) = \kappa_n (x - x_0)^2 \prod_{j=1}^{n-2} (x - x_j)^2.$$

Since q is again single-signed on $\mathbb{R}$, the same argument as above shows that we can again achieve a contradiction. Thus, p_n cannot have a repeated root, so all its roots must be simple.

Finally, assume p_n and p_{n+1} share a root, say x_0. By the Christoffel-Darboux identity:

$$\sum_{j=0}^{n-1} p_j(x_0)p_j(x) = \sum_{j=0}^{n} p_j(x_0)p_j(x) = b_{n+1}\frac{p_{n+1}(x_0)p_n(x) - p_n(x_0)p_{n+1}(x)}{x_0 - x} = 0,$$

for all $x \neq x_0$. Similarly, (4b) can be used to show the same result for $x = x_0$. Therefore, we have

$$\sum_{j=0}^{n-1} p_j(x_0)p_j(x) = 0, \qquad x \in \mathbb{R}.$$

Since the p_j are a linearly independent basis, this implies that $p_j(x_0) = 0$ for $j = 0, \ldots, n-1$. But $p_0(x_0) = 0$ cannot happen. Thus, p_n and p_{n+1} cannot share a root. □

3.1. **Near-optimal quadrature.** Apart from curiosity, there appears to be no reason at face value to investigate roots of orthogonal polynomials. However, it turns out that zeros of such functions are roots of excellent quadrature rules. We first introduce some terminology.

Definition 3.1. *An n-point quadrature rule is a set of nodes and weights $(x_j, w_j)_{j=1}^n$. The node polynomial associated to a quadrature rule is the monic polynomial*

$$q(x) \coloneqq \prod_{j=1}^{n} (x - x_j).$$

We assume, without loss, that the nodes $\{x_j\}_j$ of a quadrature rule are distinct, and that all weights are non-zero. (For if nodes are repeated, we can merge identical nodes and sum their corresponding weights, and if weights are zero we can simply ignore any contribution associated to that node.)

Quadrature rules are built to approximate integrals, i.e., given some weight $w(x)$,

$$\sum_{j=1}^{n} w_j f(x_j) \approx \int f(x)w(x)\mathrm{d}x, \tag{20}$$

where the meaning of $\approx$ depends on the kind of quadrature rule. Quadrature rules that exactly integrate polynomials of as high a degree as possible are particularly useful. In particular given a weight function $w(x)$, we call a quadrature rule $(x_j, w_j)_{j=1}^n$ accurate to order k if

$$\int p(x)w(x)\mathrm{d}x = \sum_{j=1}^{n} p(x_j)w_j, \qquad p \in P_k$$

A first result characterizes the maximum order of accuracy of quadrature rules.

Proposition 3.1. *Suppose an n-point quadrature rule is accurate to order k. Then $k \leq 2n - 1$.*

Proof. Suppose the quadrature rule can exactly integrate polynomials of degree $2n$ and less. Then $q^2(x)$ is a nontrivial polynomial that vanishes on the quadrature nodes, where q is the node polynomial. Thus,

$$0 < \int q^2(x)w(x)\mathrm{d}x = \sum_{j=1}^{n} q^2(x_j)w_j = 0,$$

Thus, we have achieved a contradiction since the quadrature rule cannot accurately integrate this degree-$2n$ polynomial. □

Therefore, an n-point quadrature can be called *optimal* if it exactly integrates polynomials of degree $2n-1$ and less. Before characterizing optimal quadrature rules, we consider the following well-known characterization of univariate polynomial interpolation.

Lemma 3.1 (Polynomial interpolation unisolvence)**.** *Let $C(\mathbb{R})$ denote the space of continuous real-valued functions on $\mathbb{R}$. If $\{x_1, \ldots, x_n\}$ are distinct points on $\mathbb{R}$, then there is a unique interpolation operator $I_n : C(\mathbb{R}) \to P_{n-1}$ such that*

$$[I_n f](x_j) = f(x_j), \qquad j = 1, \ldots, n, \quad f \in C(\mathbb{R}).$$

The interpolant $I_n f$ is unique and can be expressed as

$$I_n f = \sum_{j=1}^{n} f(x_j)\ell_j(x), \qquad \ell_k(x) = \prod_{\substack{k=1 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

The polynomials $\{\ell_k\}_{k=1}^{n}$ are the cardinal Lagrange polynomials associated to the nodal set $\{x_1, \ldots, x_n\}$, and satisfy $\ell_k(x_j) = \delta_{k,j}$.

In light of the above result, one strategy for approximating an integral given a finite number of function evaluations is to first construct an interpolant, and then to exactly integrate the interpolant. This generates an interpolatory quadrature rule.

Corollary 3.1. *Let $\{x_1, \ldots, x_n\}$ be distinct, defining cardinal Lagrange polynomials ℓ_j. Define weights*

$$w_j = \int \ell_j(x) w(x) \mathrm{d}x. \tag{21a}$$

Then the quadrature rule $(x_j, w_j)_{j=1}^{n}$ satisfies

$$\int p(x)w(x)\mathrm{d}x = \sum_{j=1}^{n} w_j p(x_j), \qquad p \in P_{n-1}. \tag{21b}$$

A quadrature rule whose weights satisfy (21a) *is called an interpolatory quadrature rule. A quadrature rule satisfies* (21a) *if and only if it satisfies* (21b)*.*

The result above shows that any n-point quadrature rule whose order of accuracy greater than or equal to $n-1$ must be interpolatory. The condition that separates interpolatory rules of accuracy $n-1$ from those of higher accuracy is related to orthogonality properties of the node polynomial.

Theorem 3.2. *Given an n-point quadrature rule $(x_j, w_j)_{j=1}^{n}$ and an integer m satisfying $0 \leq m \leq n$, the following two statements are equivalent:*

1. *The quadrature rule has order of accuracy $n - 1 + m$.*
2. *The quadrature rule is interpolatory, and the node polynomial q satisfies*

$$\int q(x)p(x)w(x)\mathrm{d}x = 0, \qquad p \in P_{m-1}, \tag{22}$$

where P_{-1} is the empty set. The relation (22) *is called a quasi-orthogonality relation.*

Proof. Suppose the quadrature rule has order of accuracy $n-1+m$. By Corollary 3.1, this rule must be interpolatory. Now if $p \in P_{m-1}$ is arbitrary, then $qp \in P_{n-1+m}$. Thus, we have

$$\int q(x)p(x)w(x)\mathrm{d}x = \sum_{j=1}^{n} q(x_j)p(x_j)w_j = 0,$$

where the last equality holds since q is the node polynomial so that $q(x_j) = 0$.

Now assume that the quadrature rule is interpolatory and the node polynomial satisfies (22). Let $p \in P_{n-1+m}$ be arbitrary. By Euclidean division of polynomials, we can express p in terms of the node polynomial q via the relation

$$p = qQ + R, \qquad\qquad Q \in P_{m-1}, \quad R \in P_{n-1}.$$

Note that the quadrature rule exactly integrates R since it is interpolatory. Thus,

$$\int R(x)w(x)\mathrm{d}x = \sum_{j=1}^{n} R(x_j)w_j.$$

The quasi-orthogonality condition implies that

$$\int q(x)Q(x)w(x)\mathrm{d}x = 0 = \sum_{j=1}^{n} q(x_j)Q(x_j)w_j,$$

where the second equality holds since $q(x_j) = 0$. Therefore, for any $p \in P_{n-1+m}$,

$$\begin{aligned}\int p(x)w(x)\mathrm{d}x &= \int q(x)Q(x)w(x)\mathrm{d}x + \int R(x)w(x)\mathrm{d}x \\ &= \sum_{j=1}^{n} q(x_j)Q(x_j)w_j + \sum_{j=1}^{n} R(x_j)w_j = \sum_{j=1}^{n} p(x_j)w_j.\end{aligned}$$

□

Corollary 3.2 (Guassian quadrature)**.** *If $\{x_1, \ldots x_n\}$ are the roots of the degree-n L^2_w orthogonal polynomial p_n and weights w_j are defined as* (21a)*, then the resulting quadrature rule has optimal order of accuracy $2n-1$. This rule is unique, and is called the Gauss quadrature rule.*

3.2. **Gaussian quadrature weights.** The weights w_j of the Gauss quadrature rule defiend in Corollary 3.2 can be explicitly evaluated in terms of the corresponding orthonormal polynomials.

Theorem 3.3. *Let $(x_j, w_j)_{j=1}^n$ be an n-point Gaussian quadrature rule. Then*

$$w_j = \frac{1}{b_n p_n'(x_j) p_{n-1}(x_j)} = \frac{1}{\sum_{k=0}^{n-1} p_k^2(x_j)}, \qquad j = 1, \ldots, n. \tag{23}$$

Proof. Recall that $\ell_j \in P_{n-1}$ is the jth cardinal Lagrange polynomial, satisfying $\ell_j(x_k) = \delta_{j,k}$. Since the $\{x_j\}_j$ are the roots of p_N, then there is a constant C such that

$$\ell_j(x) = C\frac{p_n(x)}{x - x_j}. \tag{24}$$

Enforcing $\ell_j(x_j) = 1$ and with L'Hôpital's rule, one can verify that this constant is $C = 1/p_n'(x_j)$.

Now consider the Christoffel-Darboux formula (4a) with $y \leftarrow x_j$:

$$\sum_{k=0}^{n-1} p_k(x)p_k(x_j) = b_n\frac{p_n(x)p_{n-1}(x_j) - p_n(x_j)p_{n-1}(x)}{x - x_j} = b_n\frac{p_n(x)p_{n-1}(x_j)}{x - x_j}$$

If we integrate both sides of this equation against $w(x)$, we obtain

$$1 = \int p_0^2 w(x)\mathrm{d}x = b_n p_{n-1}(x_j) \int \frac{p_n(x)}{x - x_j} w(x)\mathrm{d}x \stackrel{(24)}{=} b_n p_{n-1}(x_j) p_n'(x_j) \int \ell_j(x) w(x)\mathrm{d}x \stackrel{(21a)}{=} w_j b_n p_{n-1}(x_j) p_n'(x_j),$$

which proves the first equality in (23). The second equality is due, again, to the Christoffel Darboux identity (4b),

$$\sum_{k=0}^{n-1} p_k^2(x_j) = b_n \left[p_n'(x_j) p_{n-1}(x_j) - p_n(x_j) p_{n-1}'(x_j)\right] = b_n p_n'(x_j) p_{n-1}(x_j).$$

□

The relation (23) immediately implies that the Gaussian quadrature rule has positive weights.

3.3. **Computation of Gaussian quadrature rules.** Gaussian quadrature rules are desirable beause of their high order of accuracy. However, the characterization in Corollary 3.2 requires that we compute roots of polynomials. This is, in general, a difficult and numerically ill-conditioned problem. However, that the polynomials whose roots we seek is an *orthogonal* polynomial provides a strong characterization that allows for stable and efficient computation. This characterization is the main topic of this section.

Definition 3.2. *Given recurrence coefficients $(a_j, b_j)_{j\geq 1}$ and a positive integer n, the $n \times n$ Jacobi matrix is the symmetric, tridigonal matrix defined as*

$$J_n = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & b_3 & \\ & & \ddots & \ddots & \\ & & & b_{n-1} & a_n \end{pmatrix}$$

The main result of this section is that n-point Gaussian quadrature rules can be derived from spectral information of J_n.

Theorem 3.4. *Let $(a_j, b_j)_{j\geq 1}$ and b_0, be the recurrence coefficients associated to the weight function $w(x)$. For any $n \geq 1$, consider the real-valued eigenvalue decomposition of J_n:*

$$J_n = V \Lambda V^T,$$

where Λ is a diagonal matrix containing the eigenvalues of J_n, and the columns of V are the corresponding orthonormal eigenvectors $\{v_j\}_{j=1}^n$,

$$\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}, \quad V = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} v_{1,1} & v_{2,1} & \cdots & v_{n,1} \\ v_{1,2} & v_{2,2} & \cdots & v_{n,2} \\ \vdots & \vdots & & \vdots \\ v_{1,n} & v_{2,n} & \cdots & v_{n,n} \end{pmatrix}$$

The eigenvalues $\{\lambda_j\}_{j=1}^n$ of J_n are all simple and are equal to the roots of p_n, hence are the nodes of the n-point Gaussian quadrature rule. Letting $w_1, \ldots, w_n$ be the weights of the n-point Gaussian quadrature rule, then

$$w_j = b_0^2 v_{j,1}^2. \tag{25}$$

Proof. Define $\boldsymbol{p}(x) \in \mathbb{R}^n$ as

$$\boldsymbol{p}(x)^T = \begin{pmatrix} p_0(x) & p_1(x) & \cdots & p_{n-1}(x) \end{pmatrix}^T \in \mathbb{R}^n.$$

Stacking the three-term recurrence relation (3) for index $j = 0, 1, \ldots, n-1$, yields the vector-valued equality

$$x\boldsymbol{p}(x) = J_n \boldsymbol{p}(x) + b_n p_n(x) \boldsymbol{e}_n, \qquad\qquad \boldsymbol{e}_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^n.$$

Thus x_j is an eigenvalue of J_n if and only if $p_n(x_j) = 0$, showing that the spectrum of J_n equals the roots of p_n. Since p_n has n distinct roots, then the $n \times n$ matrix J_n must have all simple eigenvalues.

To prove the statement about the weights, note that if $x = x_j$ is a root of p_n, then

$$x_j \boldsymbol{p}(x_j) = J_n \boldsymbol{p}(x_j),$$

so that $\boldsymbol{p}(x_j)$ is an eigenvector of J_n associated to eigenvalue x_j. Therefore, the (real-valued) eigenvector v_j satisfies $v_j = \pm \boldsymbol{p}(x_j)/\|\boldsymbol{p}(x_j)\|_2$. We therefore have

$$v_{j,1} = \pm \frac{p_0}{\|\boldsymbol{p}(x_j)\|_2} = \pm \frac{p_0}{\sqrt{\sum_{k=0}^{n-1} p_k^2(x_j)}} \overset{(23)}{=} \pm p_0 \sqrt{w_j}$$

Combining this with the relation $p_0 = 1/b_0$ yields (25). □

The characterization of Gaussian nodes and weights provided by the previous theorem elegantly characterizes a strategy for computing Gauss quadrature rules: compute the spectrum of the (sparse) symmetric tridiagonal matrix J_n. Since the numerical computation of such quantities is efficient and stable, this recipe for computing Gaussian rules is perhaps the most-used procedure for computing Gaussian quadrature nodes and weights.

4. Computing approximations with orthogonal polynomials

One of the major tasks in spectral methods is formation of an approximating polynomial to a given function $u(x)$. Typically, we assume that $u(x)$ is given, or can be evaluated easily at arbitrary locations x. This section concerns construction of approximations from orthogonal polynomials. Precisely, we will consider two approaches, interpolatory procedures and quadrature procedures. In some situations, these two procedures result in identical approximations.

The approximation techniques we consider in this section both fall under the following category: suppose we have M point-evaluations $\{u(x_j)\}_{j=1}^M$ of u, and we seek to compute the approximation

$$u(x) \approx u_N(x) \coloneqq \sum_{j=0}^{N-1} \widehat{u}_j p_j(x), \tag{26}$$

where $\{p_j\}_{j \geq 0}$ is an orthonormal polynomial family. When solving differential equations, we will typically manipulate and operate on u_N instead of u. The approximation defining

u_N can be recast as a linear algebra condition,

$$V\widehat{\boldsymbol{u}} \approx \boldsymbol{u}, \qquad \widehat{\boldsymbol{u}} = \begin{pmatrix} \widehat{u}_0 \\ \vdots \\ \widehat{u}_{N-1} \end{pmatrix}, \qquad \boldsymbol{u} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_M) \end{pmatrix}, \tag{27}$$

where the matrix V is a Vandermonde-like matrix,

$$V = \begin{pmatrix} p_0(x_1) & p_1(x_1) & \cdots & p_{N-1}(x_1) \\ p_0(x_2) & p_1(x_2) & \cdots & p_{N-1}(x_2) \\ \vdots & \vdots & & \vdots \\ p_0(x_M) & p_1(x_M) & \cdots & p_{N-1}(x_M) \end{pmatrix} \in \mathbb{R}^{M\times N}.$$

Clearly, the feasibility of enforcing equality in (27) depends on the relation of M to N. We will discuss in the next sections particular strategies for computing vectors $\widehat{\boldsymbol{u}}$ from (27) that define approximations u_N of the form (26). Later, we will see how to mathematically analyze the expected error committed by such approximations.

4.1. **Interpolatory constructions.** When $M = N$ and the nodes x_j are distinct points, then Lemma 3.1 implies that exact satisfaction of (27) can be achieved. In particular, the matrix $V \in \mathbb{R}^{N\times N}$ is invertible, and so the unique solution is

$$\widehat{\boldsymbol{u}} = V^{-1}\boldsymbol{u}, \tag{28}$$

which defines the approximation (26). The computational tractibility of this approach is typically limited by (a) the stability (conditioning) of the matrix V, and (b) the size of N. Inverting matrices typically requires $\mathcal{O}(N^3)$ operations, which can be expensive when N is large.

More concerning is the stability issue. For a square matrix V, the condition number $\kappa(V)$ of V is defined as

$$\kappa(V) \coloneqq \frac{\sigma_{\max}(V)}{\sigma_{\min}(V)},$$

where $\sigma_{\max}(\cdot)$ and $\sigma_{\min}(\cdot)$ are, respectively, the maximum and minimum singular values of an input matrix. A standard rule-of-thumb in numerical analysis is that, in finite-precision arithmetic, one loses approximately $\log_{10}\kappa(V)$ digits of accuracy when using (28) to compute $\widehat{\boldsymbol{u}}$. If the nodes x_j are chosen poorly, then $\kappa(V)$ can be quite large, sometimes growing exponentially with N.

At this level of generality we can say very little about accuracy of interpolatory approaches, and can only provide a consistency result that is immediate from Lemma 3.1.

Lemma 4.1. *For interpolatory approaches on distinct nodes, if $u \in P_{N-1}$, then $u_N = u$.*

4.2. **Quadrature constructions.** We consider two types of quadrature constructions: direct methods, and least-squares methods. Direct methods are applicable for any relationship between M and N, whereas least-squares methods require $M \geq N$. We assume in this section that we have access to an M-point quadrature rule $(x_j, w_j)_{j=1}^M$.

Direct methods use the following approach: we seek to compute the coefficients $\widehat{u}_j$ in (26). Taking the definition of u_N in that equation, we have

$$\widehat{u}_j = \langle u_N, p_j \rangle_w \approx \langle u, p_j \rangle_w \approx \sum_{k=1}^M w_k u(x_k) p_j(x_k), \tag{29}$$

where the second $\approx$ comes from the desideratum (20). Thus, one way to define the $\widehat{u}_j$ coefficients is as the right-hand side of the above expression; we say that the coefficients defined in this way are computed via a *direct qudarature* procedure. Rewriting this in terms of linear algebra, we have

$$\widehat{\boldsymbol{u}} = V^T W \boldsymbol{u}, \qquad W = \begin{pmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_M \end{pmatrix}. \tag{30}$$

We see that this is a relation that is dual to $V\widehat{\boldsymbol{u}} = \boldsymbol{u}$, but the two systems are not directly related in general. One suspects that the accuracy of this approach, even simply in terms of consistency results, depends heavily on the order of accuracy of the quadrature rule.

Lemma 4.2. *Suppose $u \in P_{N-1}$. The direct quadrature approach with M quadrature points recovers $u_N = u$ if the quadrature rule is accurate to order $2N-2$.*

The proof comes directly from the chain (29): Both $\approx$ relations become equality if u is a polynomial and if the quadrature rule can accurately integrate $up_j \in P_{2N-2}$. Note that, from Proposition 3.1, direct quadrature approaches can be consistent on P_{N-1} in the sense of Lemma 4.2 only if $M \geq N$, but this relation between M and N is not sufficient for consistency. However, direct quadrature can be applied to compute *some* approximation u_N no matter the relation between M and N.

A second quadrature approach is least-squares: given the rectangular system $V\widetilde{\boldsymbol{u}} = \boldsymbol{u}$, we seek the solution $\widetilde{\boldsymbol{u}}$ in the least-squares sense with respect to the quadrature rule. I.e., we seek coefficients $\widehat{u}_j$ that minimize the residual

$$\sum_{j=1}^{M} w_j \left(u_N(x_j) - u(x_j)\right)^2 = \sum_{j=1}^{M} w_j \left(\sum_{k=0}^{N-1} \widehat{u}_k p_k(x_j) - u(x_j)\right)^2.$$

In linear algebraic terms, we seek the solution to the problem

$$\underset{\widehat{\boldsymbol{u}} \in \mathbb{R}^{\boldsymbol{N}}}{\operatorname{argmin}} \left\| \sqrt{W} \left(V\widehat{\boldsymbol{u}} - \boldsymbol{u}\right)\right\|_2, \tag{31}$$

where W is the diagonal matrix defined in (30). Note that this is a weighted least-squares problem. We first codify some well-known conditions that ensure existence of a unique solution to this problem.

Lemma 4.3. *There is a unique solution to* (31) *for all $\boldsymbol{u}$ if and only if $M \geq N$.*

Proof. If a unique solution to (31) always exists, then $\sqrt{W}V \in \mathbb{C}^{M\times N}$ cannot have a kernel, which means that $M \geq N$. Now instead assume $M \geq N$. Recall from Definition 3.1 that all quadrature rules have distinct nodes and non-zero weights. If $M \geq N$, then all columns of V are linearly independent (by Lemma 3.1), so that V has full (column) rank. Since all weights are non-zero then W is full-rank, so that $\sqrt{W}V$ has full (column) rank. Therefore the least-squares problem has a unique solution. □

Note above that we need not assume positive weights, and the result above holds true even with negative weights.

4.3. **Equivalences.** For general function data $\boldsymbol{u}$, the interpolatory approach (28), the direct quadrature approach (30), and the least-squares method (31) all produce different approximations. However, there are some simple situations in which they provide the same approximation.

Proposition 4.1. *If the quadrature rule* $(x_j, w_j)_{j=1}^M$ *is accurate to order* $2N-2$*, then the direct quadrature method* (30) *and the least-squares approach* (31) *produce the same approximation* u_N*.*

Proposition 4.2. *If the quadrature rule* $(x_j, w_j)_{j=1}^M$ *has* $M = N$ *points and is accurate to order* $2N-2$*, then the interpolatory approach* (28)*, the direct quadrature method* (30)*, and the least-squares approach* (31) *all produce the same approximation* u_N*.*