Chapter 3

Polynomial interpolation

We are getting back to the topic of (polynomial) interpolation from Section 2.1. First, a short summary of what we learned in that section. Given interpolation data (x_j, y_j) with $x_j \in \mathbb{R}$ and $y_j \in \mathbb{R}$ for $j = 0, \dots, n$, we have shown that there is a unique polynomial p_n of degree at most n such that

$$p_n(x_j) = y_j, \quad j = 0, \dots, n,$$

if (and only if) the interpolation nodes x_j are pairwise distinct. The Lagrange representation of p_n is given by

$$p_n(x) = \sum_{j=0}^{n} y_j \ell_j(x), \quad \ell_j(x) := \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}. \tag{3.1}$$

In the following, we let $\|\cdot\|_\infty$ denote the uniform (or L^∞) norm of a function $g : [a, b] \to \infty$, that is, $\|g\|_\infty := \sup_{x\in[a,b]} |g(x)|$. Then Theorem 2.3 implies the error bound

$$\|f - p_n\|_\infty \le \frac{1}{(n+1)!} \|\omega_{n+1}\|_\infty \|f^{(n+1)}\|_\infty \tag{3.2}$$

when $y_j = f(x_j)$ for an $n + 1$ times continuously differentiable function f. Apart from properties of the function f, the norm of $\omega_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$ on $[a, b]$ also enters this error bound.

By the Weierstrass approximation theorem, for any *continuous* function f there is a sequence of polynomials $\tilde{p}_n$ such that $\|f-\tilde{p}_n\|_\infty$ converges to zero as $n \to \infty$. This, however, does *not* necessarily hold for the polynomials p_n defined in (3.1), even when assuming that f is infinitely often differentiable. To see this, consider the function $f(x) = 1/\left(1 + 25x^2\right)$ and equidistant nodes on $[-1, 1]$, that is, $x_j = -1 + 2j/n$ for $j = 0, \dots, n$. As n increases, the accuracy of the polynomial interpolation deteriorates at the neighborhoods of the end points of the interval; see Figure 3.1. In particular, the $\|f - p_n\|_\infty$ *grows* (quite quickly) instead of converging to zero. This problem has been observed for the first time in 1901 by Runge and, for this reason, is called *Runge phenomenon.* Both, the growth of the derivatives of f and the strong oscillations of ω_{n+1} near the end points of the interval contribute to this phenomenon.

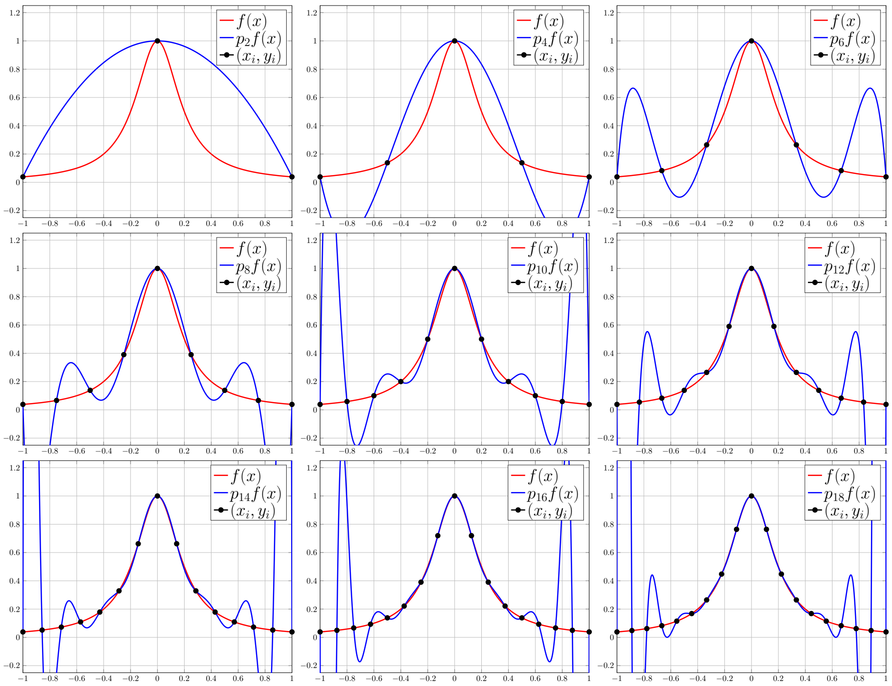


Figure 3.1: The Runge phenomenon for $f(x) = \frac{1}{1+25x^2}$.

Much of this chapter is concerned with finding better interpolation nodes that avoid the Runge phenomenon for sufficiently nice functions. We will also discuss links to stability and best approximation.

3.1 Chebyshev nodes

Motivated by the error *bound* (3.2) we now aim at determining interpolation nodes $x_0, \ldots, x_n$ that minimize

$$\|\omega_{n+1}\|_\infty = \max_{x\in[a,b]} |(x-x_0)(x-x_1)\cdots(x-x_n)|.$$

By an affine linear transformation of the interval and the interpolation nodes, we may assume without loss of generality that $[a,b] = [-1,1]$. Minimizing $\|\omega_{n+1}\|_\infty$ directly by brute force would be a daunting task. We therefore approach it indirectly via Chebyshev polynomials.

Definition 3.1 *Given* $n \in \mathbb{N}$, *the* n*th* Chebyshev polynomial *is defined as*

$$T_n(x) = \cos(n \arccos x) \qquad \forall\ x \in [-1,1].$$

Note that for every $n \in \mathbb{N}$ and $x \in [-1,1]$ we have $|T_n(x)| \leq 1$. Moreover, in spite of the unusual definition, it will follow from Lemma 3.2 below that $T_n \in \mathbb{P}_{n+1}$ for every $n \in \mathbb{N}$. By computing the first few examples, we obtain indeed:

$$\begin{aligned}
n = 0: \quad & T_0(x) = \cos(0 \arccos x) = \cos(0) = 1, \\
n = 1: \quad & T_1(x) = \cos(1 \arccos x) = x, \\
n = 2: \quad & T_2(x) = \cos(2 \arccos x) = 2\cos^2(\arccos x) - 1 = 2x^2 - 1.
\end{aligned}$$

Theorem 3.2 *The Chebyshev polynomials satisfy the following* recurrence relation*:*

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad n \in \mathbb{N}, n \geq 1.$$

Proof. For every $n \in \mathbb{N}$, $n \geq 1$, the following trigonometric identity holds:

$$\cos\left((n+1)\,\varphi\right) + \cos\left((n-1)\,\varphi\right) = 2\cos\varphi\cos\left(n\varphi\right) \qquad \forall\ \varphi \in \mathbb{R}.$$

Setting $\varphi = \arccos x$, this implies

$$T_{n+1}(x) + T_{n-1}(x) = 2T_1(x)T_n(x) = 2xT_n(x) \qquad \forall\ x \in [-1,1],$$

which completes the proof. □

Note that the result of Theorem 3.2 also implies that the leading coefficient of T_{n+1} is 2^n.

Lemma 3.3 *The roots of T_n are*

$$x_k = \cos\left(\frac{(2k+1)\,\pi}{2n}\right), \qquad k = 0, \ldots, n-1,$$

which are called Chebyshev nodes.

Proof. A direct calculation yields, for every $k \in \mathbb{N}$, $0 \leq k \leq n-1$,

$$T_n\left(x_k\right) = \cos\left(n \arccos\cos\left(\frac{(2k+1)\,\pi}{2n}\right)\right) = \cos\left(\frac{(2k+1)\,\pi}{2}\right) = 0.$$

These are all the roots of T_n because, by Theorem 3.2, T_n is a (nonzero) polynomial of degree n. □

Lemma 3.3 shows that T_n has n real, pairwise distinct roots in the open interval $(-1,1)$. Moreover, it can be seen from Figure 3.2 that the roots tend to cluster at the end points of $[-1,1]$.[9]

By looking at the extremal properties of T_n, we are getting closer to our goal of optimizing $\|\omega_{n+1}\|_\infty$.

[9] EFY: Show that the Chebyshev nodes are the real parts of points *uniformly* distributed on the upper part of the unit circle.

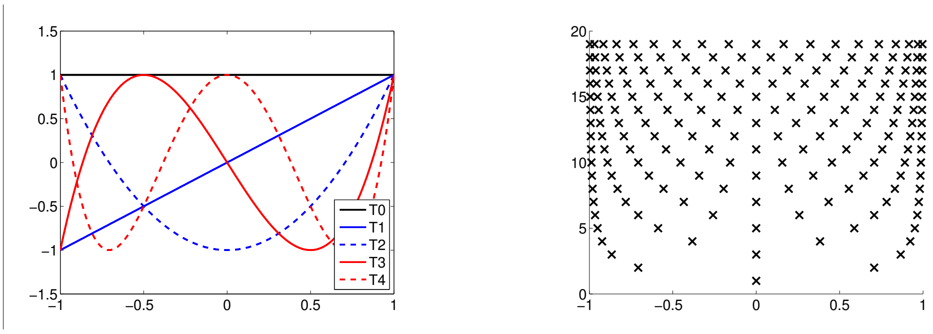


Figure 3.2: *Left:* Chebyshev polynomials $T_0, \ldots, T_4$. *Right:* Roots of Chebyshev polynomials $T_1, \ldots, T_{20}$.

Lemma 3.4 *For $n \geq 1$ the Chebyshev polynomial T_n takes alternatingly the values $+1$ and -1 exactly $n+1$ times:*

$$T_n\left(\cos\left(\frac{k\pi}{n}\right)\right) = (-1)^k, \qquad \forall\ k = 0, \ldots, n.$$

Proof. By differentiation, it is easy to show that T_n attains at the points $\cos\left(\frac{k\pi}{n}\right)$, $k = 0, \ldots n$, a local minimum for k odd and a local maximum for k even. Since, by construction, $\|T_n\|_\infty \leq 1$, these are global extrema. □

We are now ready to solve our optimization problem.

Lemma 3.5 *Among all polynomials of degree $n+1$ with leading coefficient 1, the rescaled Chebyshev polynomial $\tilde{T}_{n+1} := 2^{-n}T_{n+1}$ minimizes the uniform norm on $[-1, 1]$.*

Proof. From Lemma 3.4 we have that $\tilde{T}_{n+1}$ alternates at the values $+2^{-n}$ and -2^{-n} exactly $n+2$ times. We assume, in contradiction to the statement of the lemma, that there exists $q \in \mathbb{P}_{n+1}$ with leading coefficient 1 such that $\|q\|_\infty < 2^{-n}$ and define $p(x) := q(x) - 2^{-n}T_{n+1}(x)$ for every $x \in [-1, 1]$. Note that $p \in \mathbb{P}_n$ since the terms x^{n+1} cancel. Note, also, that p is nonzero because otherwise $\|q\|_\infty = 2^{-n}$. Moreover, p changes sign in each interval (z_i, z_{i+1}) for $i = 0, \ldots, n$, where $z_0, \ldots, z_{n+1}$ are the points where T_{n+1} attains its extreme values. In turn, by the intermediate value theorem and continuity of polynomials, p admits $n+1$ distinct roots, which is not possible for a nonzero polynomial of degree n. □

The result of Lemma 3.5 tells us that choosing the Chebyshev nodes (that is, the roots of T_{n+1}) as interpolation nodes leads to the ω_{n+1} of smallest uniform norm. Mapping the Chebyshev nodes to a general interval $[a, b]$, we arrive at the following result.

Theorem 3.6 *The expression* $\max_{x\in[a,b]} |(x-x_0)\dots(x-x_n)|$ *is minimized for the interpolation nodes*

$$x_k = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{(2k+1)\,\pi}{2n+2}\right), \qquad k = 0,\dots,n.$$

For this choice of interpolation nodes, the interpolating polynomial p_n *satisfies*

$$\|f-p_n\|_\infty \le \frac{1}{2^n(n+1)!}\left(\frac{b-a}{2}\right)^{n+1}\|f^{(n+1)}\|_\infty.$$

Proof. The proof proceeds, similarly to what has been used in Chapter 2, by an affine linear mapping from the reference interval to the interval of interest. In this case, the reference interval is $[-1,1]$ and the interval of interest is $[a,b]$, and the mapping takes the form

$$\varphi : [-1,1] \to [a,b], \qquad \varphi : x \mapsto \frac{a+b}{2} + \frac{b-a}{2}x.$$

Note that this map is invertible, with the inverse map given by

$$\varphi^{-1} : [a,b] \to [-1,1], \qquad \varphi^{-1} : y \mapsto \frac{2}{b-a}\left(y - \frac{a+b}{2}\right).$$

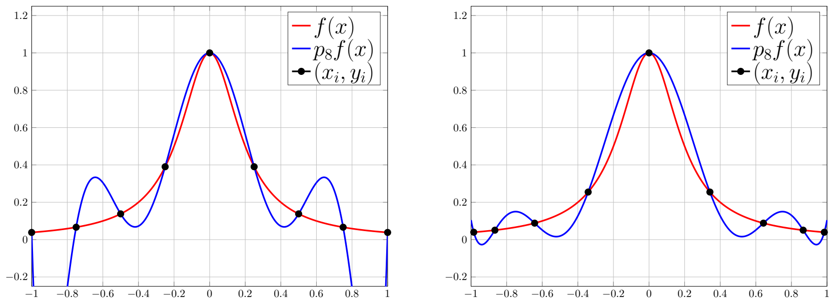
We can use φ to map interpolation nodes $\tilde{x}_0,\dots,\tilde{x}_n \in [-1,1]$, to the interval $[a,b]$: $x_k = \varphi(\tilde{x}_k)$ for $k = 0,\dots,n$. Setting $\tilde{x} = \varphi^{-1}(x) \in [-1,1]$, we obtain

$$\begin{aligned}
\omega_{n+1}(x) &= (x-x_0)(x-x_1)\cdots(x-x_n) \\
&= (\varphi(\tilde{x})-\varphi(\tilde{x}_0))(\varphi(\tilde{x})-\varphi(\tilde{x}_1))\cdots(\varphi(\tilde{x})-\varphi(\tilde{x}_n)) \\
&= \left(\frac{b-a}{2}\right)^{n+1} \underbrace{(\tilde{x}-\tilde{x}_0)(\tilde{x}-\tilde{x}_1)\cdots(\tilde{x}-\tilde{x}_n)}_{=:\tilde{\omega}_{n+1}(\tilde{x})}.
\end{aligned}$$

By Lemma 3.5, we know that the maximum norm of $\tilde{\omega}_{n+1}$ on $[-1,1]$ is minimized by choosing $\tilde{x}_k = \cos((2k+1)\pi/(2n+2))$, the roots of the Chebyshev polynomial T_{n+1}. The relation above shows that the maximum norm of ω_{n+1} on $[a,b]$ is minimized by setting $x_k = \varphi(\tilde{x}_k) = \varphi(\cos((2k+1)\pi/(2n+2))$, which proves the first part of the theorem. The second part follows directly from the error bound (3.2):

$$\begin{aligned}
\|f-p_n\|_\infty &\le \frac{1}{(n+1)!}\|\omega_{n+1}\|_\infty\|f^{(n+1)}\|_\infty \\
&= \frac{1}{(n+1)!}\left(\frac{b-a}{2}\right)^{n+1}\|\tilde{\omega}_{n+1}\|_\infty\|f^{(n+1)}\|_\infty \\
&= \frac{1}{2^n(n+1)!}\left(\frac{b-a}{2}\right)^{n+1}\|f^{(n+1)}\|_\infty,
\end{aligned}$$

where the last inequality uses $\tilde{\omega}_{n+1}(\tilde{x}) = 2^{-n} T_{n+1}(\tilde{x})$. □



(a) Interpolating on equidistant points, $n = 8$, $f(x) = \frac{1}{1+25x^2}$.

(b) Interpolating on Chebyshev nodes, $n = 8$, $f(x) = \frac{1}{1+25x^2}$.

Figure 3.3: Comparison of polynomial interpolation with equidistant and Chebyshev nodes.

Figure 3.3 confirms that Chebyshev nodes compare favorably with equidistant nodes for the Runge function.

It can be shown that the error $\|f - p_n\|_\infty$ converges to zero for any Lipschitz function f when choosing Chebyshev nodes. For a real analytic function f, the error converges exponentially fast, that is, it is bounded by a constant times ρ^{-n}, where $\rho > 1$ depends on the domain of analyticity of f. We highly recommend Trefethen's book [6] and the associated Chebfun software package for many more fascinating facts and uses of Chebyshev polynomials.[10]

3.2 Sensitivity and best uniform approximation

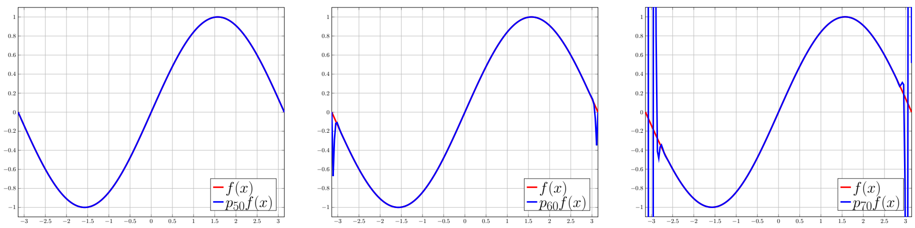
Building and evaluating interpolating polynomials is subject to roundoff error on a computer. Figures 3.4 and 3.5 show the results of interpolating $f(x) = \sin(x)$ on $[-\pi, \pi]$. Theoretically, for both equidistant and Chebyshev nodes, we expect that $\|f - p_n\|_\infty$ converges to zero as n increases. Instead, Figure 3.4 shows "wiggles" for large n in the case of equidistant nodes. These wiggles are due to roundoff error caused by numerical instabilities when using equidistant nodes.

Roundoff error already occurs when evaluating and storing the function values $y_i = f(x_i)$ on the computer. Let us investigate the effect of this error on the interpolating polynomial. For this purpose, suppose that

$$\hat{f}(x_i) = f(x_i)(1 + \delta_i), \qquad \text{where} \qquad |\delta_i| \leq \epsilon.$$

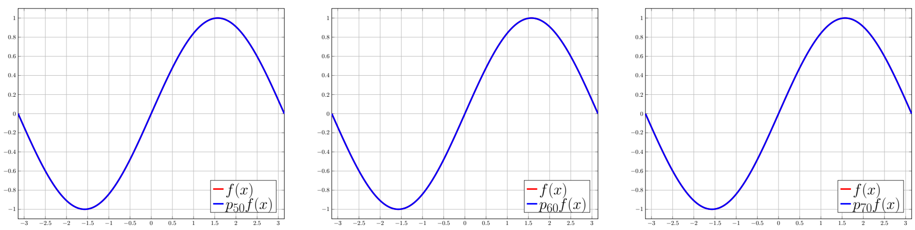
For elementary functions, we know that $\epsilon = u$, the unit roundoff. For more complex functions, this can be larger, but often one still has $\epsilon \approx 10^{-16}$. The interpolating

[10]EFY: Download Chebfun / ApproxFun / pychebfun and play with it!



(a) $n = 50$ (b) $n = 60$ (c) $n = 70$

Figure 3.4: Interpolation of $f(x) = \sin(x)$ on equidistant nodes.



(a) $n = 50$ (b) $n = 60$ (c) $n = 70$

Figure 3.5: Interpolating of $f(x) = \sin(x)$ on Chebyshev nodes.

polynomials for f and $\hat{f}$ are given by

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\,\ell_i(x), \qquad \hat{p}_n(x) = \sum_{i=0}^{n} \hat{f}(x_i)\,\ell_i(x). \tag{3.3}$$

respectively. Then for every $x \in [a, b]$ we have that

$$\begin{aligned} |p_n(x) - \hat{p}_n(x)| &= \Big| \sum_{i=0}^{n} \big(f(x_i) - \hat{f}(x_i)\big)\ell_i(x) \Big| \\ &\leq \sum_{i=0}^{n} \epsilon |f(x_i)|\,|\ell_i(x)| \leq \epsilon \, \|f\|_\infty \sum_{i=0}^{n} |\ell_i(x)|. \end{aligned} \tag{3.4}$$

Definition 3.7 *The quantity*

$$\Lambda_n := \max_{x \in [a,b]} \sum_{i=0}^{n} |\ell_i(x)|$$

is called Lebesgue constant *associated with the interpolation nodes* $x_0, \ldots, x_n$.

Table 3.1: Lebesgue constant Λ_n for equidistant and Chebyshev nodes on $[-1,1]$.

n	equidistant	Chebyshev
5	3.106	2.104
10	29.89	2.489
15	512.05	2.728
20	10986.53	2.901

By the discussion above, we have

$$\|p_n - \hat{p}_n\|_\infty \le \epsilon \Lambda_n \|f\|_\infty .$$

Hence, Λ_n measures the sensitivity of the interpolating polynomial with respect to perturbations in the interpolation data. Table 3.1 shows that the Lebesgue constant grows much more rapidly for equidistant nodes than for Chebyshev nodes, explaining the effect observed in Figure 3.4. It has been shown that

$$\Lambda_n \sim \frac{2^n}{e(n-1)\ln n} \quad n \to +\infty$$

for equidistant nodes and

$$\Lambda_n \sim \frac{2}{\pi} \ln n \quad n \to +\infty \tag{3.5}$$

for Chebyshev nodes. There is a dramatic difference; exponential vs. logarithmic growth!

The Lebesgue constant also measures how far one is away from the best uniform approximation of a function.

Theorem 3.8 *Let $f \in C^0([a,b])$ and consider the interpolating polynomial p_n for f on interpolation nodes $x_0, \ldots, x_n \in [a,b]$. Then*

$$\inf_{q \in \mathbb{P}_n} \|f - q\|_\infty \le \|f - p_n\|_\infty \le (1 + \Lambda_n) \inf_{q \in \mathbb{P}_n} \|f - q\|_\infty .$$

Proof. The first inequality holds trivially because $p_n \in \mathbb{P}_n$. We first note the trivial fact that the interpolation of a polynomial p of degree at most n by a polynomial of degree n is the polynomial p itself. This implies

$$\begin{aligned} f(x) - p_n(x) &= f(x) - q(x) + q(x) - p_n(x) \\ &= f(x) - q(x) + \sum_{i=0}^{n} (q(x_i) - p_n(x_i)) \ell_i(x). \end{aligned}$$

for any $q \in \mathbb{P}_n$. Hence,

$$\|f - p_n\|_\infty \le \|f - q\|_\infty + \|f - q\|_\infty \left\| \sum_{j=0}^{n} |\ell_j(x)| \right\|_\infty = (1 + \Lambda_n)\|f - q\|_\infty .$$

Because this holds for arbitrary $q \in \mathbb{P}_n$, this completes the proof. ∎

Using Theorem 3.8 and (3.5), we see that Chebyshev interpolation attains nearly the best uniform approximation. In practice, being nearly best is usually sufficient. If one wants to achieve the truly best approximation in the L^∞ norm, one needs to resort to the so called Remez algorithm.

3.3 Best approximation in L^2 norm*

We now consider the best approximation in the L^2 norm on $[-1,1]$:

$$\|u\|_2 := \Big(\int_{-1}^{1} |u|^2 \, \mathrm{d}t \Big)^{1/2} \tag{3.6}$$

for a function $u : [-1,1] \to \infty$. More specifically, we aim at solving the following minimization problem. Given $f \in C^0([-1,1])$, determine

$$p^* = \underset{q_n \in \mathbb{P}_n}{\arg\min} \, \|f - q_n\|_2^2 \, . \tag{3.7}$$

What makes (3.7) fundamentally different (and simpler) compared to best uniform approximation is that $\|\cdot\|_2$ is induced by the L^2 inner product

$$(u, v)_2 = \int_{-1}^{1} u(t) v(t) \, \mathrm{d}t.$$

It is instructive to frame (3.7) in an abstract setting: Let V be a (possibly infinite-dimensional) real vector space with an inner product $(\cdot,\cdot)_V$. Letting U be a finite-dimensional subspace of V, we consider for given $v \in V$ the approximation problem

$$u^* = \underset{u \in U}{\arg\min} \|v - u\|_V^2. \tag{3.8}$$

Here, $\|\cdot\|_V^2$ denotes the norm induced by the inner product, that is, $\|w\|_V^2 = (w,w)_V$. We let v_U denote the orthogonal projection of $v \in V$ onto U, that is, the unique vector $v_U \in U$ in the decomposition

$$v = v_U + v_\perp, \quad v_U \in U, \quad (v_\perp, u)_V = 0 \ \forall u \in U. \tag{3.9}$$

Then for any vector $u \in U$ it holds that

$$\|v - u\|_V^2 = \|v_\perp + (v_u - u)\|_V^2 = \|v_\perp\|_V^2 + 2(v_\perp, v_u - u) + \|v_u - u\|_V^2 = \|v_\perp\|_V^2 + \|v_u - u\|_V^2.$$

The last expression is minimized by setting $u = v_u$. This yields the following theorem.

Theorem 3.9 *The unique solution to the minimization problem* (3.8) *is the orthogonal projection of v onto U.*

An explicit expression for $u^* = v_U$ is obtained when choosing an orthonormal basis $u_0, u_1, \dots, u_n$ of U, where $\dim U =: n+1$. Then

$$v_U = \sum_{k=0}^{n} (u_k, v) u_k. \tag{3.10}$$

To see this, we need to verify that (3.9) is satisfied for $v_\perp = v - v_U$. First, it directly follows that $v_u \in U$. Second, for every u_j orthonormality implies

$$(v_\perp, u_j) = (v - v_U, u_j) = (v, u_j) - \sum_{k=0}^{n} (u_k, v)(u_k, u_j) = (v, u_j) - (v, u_j) = 0.$$

Hence, $v_\perp$ is orthogonal to every $u \in U$.

To apply Theorem 3.8 and (3.10) to polynomial approximation on $[-1, 1]$, we recall that, by Definition 2.10, the Legendre polynomials $q_0, \dots, q_n$ form an *orthogonal* basis of Π_n in the L^2 inner product. It can be shown that

$$\|q_k\|_2^2 = \frac{2}{2k+1}.$$

Hence the scaled Legendre polynomials $\tilde{q}_0, \dots, \tilde{q}_n$ with $\tilde{q}_k := \sqrt{\frac{2}{2k+1}} q_k$ form an *orthonormal* basis of Π_n. As a corollary of Theorem 3.9 we now obtain the solution of (3.7).

Corollary 3.10 *Given $f \in C^0([-1, 1])$, the approximation problem* (3.7) *is solved by*

$$p^*(x) = \sum_{k=0}^{n} (\tilde{q}_k, f)_2 \tilde{q}_k,$$

with the scaled Legendre polynomials $\tilde{q}_0, \dots, \tilde{q}_n$.

In order to construct p^* we need to compute

$$(\tilde{q}_k, f)_2 = \int_{-1}^{1} f \tilde{q}_k \, \mathrm{d}t, \quad k = 0, \dots, n.$$

In general, one cannot calculate these quantities exactly . Instead, one can employ a Gauss quadrature with $n+1$ points in order to approximate them.

3.4 A note on piece-wise interpolation*

Similar to composite numerical quadrature, piece-wise interpolation partitions the interval into subintervals and applies polynomial interpolation to every subinterval. For $n = 1$ (that is, piece-wise linear interpolation) this will yield a continuous, polygonal curve when choosing on each subinterval the end points as interpolation nodes. For larger n, the additional degrees of freedom are *not* used to interpolate additional nodes in each subinterval but to achieve smoothness at the end points of subintervals. For $n = 3$, this yields the concept of (cubic) splines, which are two times differentiable on the whole interval. Constructing such splines involves the solution of linear systems; the details of the construction are beyond the scope of this lecture.