2.2 Cubic Splines

Recall the **the space of polynomial spline functions** of degree m and class k on Δ

$$\begin{aligned} \mathbb{S}_m^k(\Delta) &= \{s \mid s \in C^k[a,b],\ s|_{[x_i,x_i+1]} \in \mathbb{P}_m,\ i = 1,2,\ldots,n-1\}, \\ \Delta &: a = x_1 < x_2 < \cdots < x_{n-1} < x_n = b. \end{aligned} \tag{2.1}$$

Now we focus on the case $m = 3$.

Cubic splines are the most widely used. In general, cubic splines are fairly smooth functions that are convenient to work with. Cubic spline interpolation is a useful tool in mathematical modeling of curves and surfaces of complex geometric shapes in aircraft construction, shipbuilding, production of hydro turbines and many more areas of science and technology. They have also come to be widely used in the past several decades in computer graphics. There are several types of cubic spline functions, depending on the smoothness conditions they satisfy.

Interpolation with cubic splines $s \in \mathbb{S}_3^1(\Delta)$

We impose the continuity of the first order derivative of $s_3(f;\cdot)$ by prescribing the values of the first derivative at each node $x_i, i = 1,2,\ldots,n$. Given n arbitrary numbers $m_1, m_2, \ldots, m_n$, we seek a function $s_3(f;\cdot)$ that satisfies the conditions

$$\begin{aligned} s_3|_{[x_i,x_{i+1}]} &= p_i(x) \in \mathbb{P}_3,\ i = 1,2,\ldots,n-1, \\ s_3(f;x_i) &= f_i,\ i = 1,2,\ldots,n, \\ s_3'(f;x_i) &= m_i,\ i = 1,2,\ldots,n. \end{aligned} \tag{2.2}$$

This means that on each subinterval $[x_i, x_{i+1}]$, $s_3(f;\cdot)$ is the unique solution of the Hermite interpolation problem

$$\begin{aligned} p_i(x_i) &= f_i, \quad p_i(x_{i+1}) = f_{i+1}, \\ p_i'(x_i) &= m_i, \quad p_i'(x_{i+1}) = m_{i+1},\ i = \overline{1,n-1}. \end{aligned} \tag{2.3}$$

The divided differences are computed from the table

$$
\begin{array}{c|ccccccc}
x_i & \boldsymbol{f_i} & \longrightarrow & \boldsymbol{m_i} & \longrightarrow & \dfrac{\boldsymbol{f[x_i, x_{i+1}] - m_i}}{\boldsymbol{h_i}} & \longrightarrow & \dfrac{\boldsymbol{m_{i+1} - 2f[x_i, x_{i+1}] + m_i}}{\boldsymbol{h_i^2}} \\
 & & \nearrow & & \nearrow & & \nearrow & \\
x_i & f_i & \longrightarrow & f[x_i, x_{i+1}] & \longrightarrow & \dfrac{m_{i+1} - f[x_i, x_{i+1}]}{h_i} & & \\
 & & \nearrow & & \nearrow & & & \\
x_{i+1} & f_{i+1} & \longrightarrow & m_{i+1} & & & & \\
 & & \nearrow & & & & & \\
x_{i+1} & f_{i+1}, & & & & & &
\end{array}
$$

Using the Newton form of the Hermite polynomial, we have

$$
\begin{aligned}
p_i(x) &= f_i + m_i(x - x_i) + \frac{f[x_i, x_{i+1}] - m_i}{h_i}(x - x_i)^2 \\
&+ \frac{m_{i+1} - 2f[x_i, x_{i+1}] + m_i}{h_i^2}(x - x_i)^2(x - x_{i+1}).
\end{aligned}
$$

Alternatively, we can write it in Taylor's form around x_i. Considering that $x - x_{i+1} = x - x_i - h_i$, for $x \in [x_i, x_{i+1}]$, we get

$$
p_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3, \tag{2.4}
$$

with

$$
\begin{aligned}
c_{i,0} &= f_i, \\
c_{i,1} &= m_i, \\
c_{i,2} &= \frac{f[x_i, x_{i+1}] - m_i}{h_i} - c_{i,3}h_i = \frac{3f[x_i, x_{i+1}] - 2m_i - m_{i+1}}{h_i}, \\
c_{i,3} &= \frac{m_{i+1} - 2f[x_i, x_{i+1}] + m_i}{h_i^2}.
\end{aligned} \tag{2.5}
$$

Hence, to compute $s_3(f; x)$ at a point $x \in [a, b]$ that is not a node, we first identify the interval $[x_i, x_{i+1}]$ that contains x, then compute the coefficients in (2.5) and evaluate the spline using (2.4).

Next, we discuss some possible choices for the parameters $m_1, m_2, \ldots, m_n$.

Piecewise cubic Hermite interpolation

Assuming that the derivatives $f'(x_i),\ i = 1, \ldots, n$, are known, we choose $m_i = f'(x_i)$. This way, we obtain a strictly local scheme, where the polynomial on each subinterval $[x_i, x_{i+1}]$ is completely determined by the interpolation data at node points inside, independently of the other pieces. The error in this case (see Example 1.4, in Lecture 6) is

$$||f(\cdot) - s_3(f, \cdot)||_\infty \ \leq \ \frac{1}{384}\,|\Delta|^4\,||f^{(4)}||_\infty. \tag{2.6}$$

For equally spaced nodes, we have

$$|\Delta| \ = \ (b-a)/(n-1)$$

and, therefore,

$$||f(\cdot) - s_3(f, \cdot)||_\infty \ = \ O(n^{-4}),\ n \to \infty. \tag{2.7}$$

Interpolation with cubic splines $s \in \mathbb{S}_3^2(\Delta)$

To have $s_3(f; \cdot) \in S_3^2(\Delta)$, we require continuity of the second derivatives at the nodes, i.e.

$$p''_{i-1}(x_i) \ = \ p''_i(x_i),\ i = 2, \ldots, n-1,$$

which, for the Taylor coefficients in (2.4), means

$$2c_{i-1,2} \ + \ 6c_{i-1,3}h_{i-1} \ = \ 2c_{i,2},\ i = 2, \ldots, n-1. \tag{2.8}$$

Substituting in (2.5), we obtain the linear system

$$h_i m_{i-1} \ + \ 2(h_{i-1} + h_i)m_i \ + \ h_{i-1}m_{i+1} \ = \ b_i,\ i = 2, \ldots, n-1, \tag{2.9}$$

where

$$b_i \ = \ 3\Big(h_i f[x_{i-1}, x_i] + h_{i-1} f[x_i, x_{i+1}]\Big). \tag{2.10}$$

Thus, we have a system of $n-2$ linear equations with n unknowns, $m_1, m_2, \ldots, m_n$. Once m_1 and m_n are chosen, the system is *tridiagonal* and can be solved efficiently by several methods.

Next, we discuss possible choices for m_1 and m_n.

1. **Complete (clamped) splines.** We take

$$m_1 = f'(a),\ m_n = f'(b).$$

For this type of spline, it can be shown that, if $f \in C^4[a,b]$, then

$$||f^{(r)}(\cdot) - s_3^{(r)}(f,\cdot)||_\infty \ \leq \ C_r|\Delta|^{4-r}\ ||f^{(4)}||_\infty,\ r = 0,1,2,3, \tag{2.11}$$

where

$$C_0 \ = \frac{5}{384},\ C_1 \ = \ \frac{1}{24},\ C_2 \ = \ \frac{3}{8},$$

and C_3 depends on the ratio $|\Delta|/\min\limits_i h_i$.

2. **Endpoint second derivative splines.** We require

$$s_3''(f,a) \ = \ f''(a),\ s_3''(f,b) \ = \ f''(b).$$

These lead to two more equations,

$$\begin{aligned} 2m_1 \ + \ m_2 \qquad\qquad\quad &= \ 3f[x_1,x_2] - \frac{1}{2}f''(a)h_1, \\ m_{n-1} \ + \ 2m_n \ &= \ 3f[x_{n-1},x_n] - \frac{1}{2}f''(b)h_{n-1}. \end{aligned} \tag{2.12}$$

We place the first equation at the beginning of the system (2.9) and the second at the end of it, thus preserving the tridiagonal structure of the system.

3. **Natural cubic splines.** Imposing

$$s_3''(f;a) \ = \ s_3''(f;b) \ = \ 0,$$

we get the same two equations as above, with $f''(a) = f''(b) = 0$:

$$\begin{aligned} 2m_1 \ + \ m_2 \qquad\qquad\quad &= \ 3f[x_1,x_2], \\ m_{n-1} \ + \ 2m_n \ &= \ 3f[x_{n-1},x_n]. \end{aligned} \tag{2.13}$$

Motivation for these boundary conditions can be given by looking at the physics of *bending thin beams of flexible materials* to pass thru the given data. To the left of x_1 and to the right of x_n, the beam is straight and therefore the second derivatives are zero at the transition points x_1 and x_n.

The advantage of this type of spline is that it requires only the function values of f – no derivatives – but the price paid is a decrease in the accuracy to $O(|\Delta|^2)$ near the endpoints (unless indeed $f''(a) = f''(b) = 0$).

4. **"Not-a-knot" (deBoor) splines.** Here we impose the conditions that the first two pieces and the last two, coincide, i.e.

$$p_1(x) \equiv p_2(x),\ p_{n-2}(x) \equiv p_{n-1}(x).$$

This means that the first and last interior nodes, x_2 and x_{n-1}, are both inactive (hence, the name). We get two more equations expressing the continuity of $s_3'''(f;x)$ at $x = x_2$ and $x = x_{n-1}$. This comes down to the equality of the leading coefficients $c_{1,3} = c_{2,3}$ and $c_{n-2,3} = c_{n-1,3}$. Thus, we get

$$\begin{array}{rcrcrcl} h_2^2\, m_1 & + & \big(h_2^2 - h_1^2\big)m_2 & - & h_1^2\, m_3 & = & \beta_1, \\ h_{n-1}^2\, m_{n-2} & + & \big(h_{n-1}^2 - h_{n-2}^2\big)m_{n-1} & - & h_{n-2}^2\, m_n & = & \beta_2, \end{array} \tag{2.14}$$

where

$$\begin{aligned} \beta_1 &= 2\Big(h_2^2 f[x_1,x_2] - h_1^2 f[x_2,x_3]\Big), \\ \beta_2 &= 2\Big(h_{n-1}^2 f[x_{n-2},x_{n-1}] - h_{n-2}^2 f[x_{n-1},x_n]\Big). \end{aligned}$$

Again, we place the first equation at the beginning of the system (2.9) and the second at the end of it. Even so, the resulting system is *no longer* tridiagonal, but it can be transformed into a tridiagonal one, by combining equations 1 and 2, and $n-1$ and n, respectively. Consequently, the first and the last equations become

$$\begin{array}{rcrcrcl} h_2\, m_1 & + & \big(h_2 + h_1\big)m_2 & & & = & \gamma_1, \\ & & & & \big(h_{n-1} - h_{n-2}\big)m_{n-1} & + & h_{n-2}\, m_n \;=\; \gamma_2, \end{array} \tag{2.15}$$

where

$$\begin{aligned} \gamma_1 &= \frac{1}{h_2 + h_1}\left[f[x_1,x_2]h_2\Big(h_1 + 2\big(h_1+h_2\big)\Big) + h_1^2 f[x_2,x_3]\right], \\ \gamma_2 &= \frac{1}{h_{n-1}+h_{n-2}}\left[h_{n-1}^2 f[x_{n-2},x_{n-1}] + \Big(2\big(h_{n-1}+h_{n-2}\big) + h_{n-1}\Big)h_{n-2} f[x_{n-1},x_n]\right]. \end{aligned}$$

Finding cubic splines using the second derivatives

Computational formulas for finding cubic splines $s \in \mathbb{S}_3^2(\Delta)$ can be derived (in a similar way) when the arbitrary numbers $M_1, M_2, \ldots, M_n$ are given and forced to satisfy the conditions

$$\begin{array}{rcl} s_3|_{[x_i,x_{i+1}]} & = & p_i(x) \in \mathbb{P}_3,\ i = 1, 2, \ldots, n-1, \\ s_3(f; x_i) & = & f_i,\ i = 1, 2, \ldots, n, \\ s_3''(f; x_i) & = & M_i,\ i = 1, 2, \ldots, n. \end{array} \tag{2.16}$$

Since s_3 is a cubic polynomial, its second derivative is linear. Hence, on $[x_i, x_{i+1}]$, we have

$$s_3''(f; x) = ax + b,$$

satisfying the conditions

$$s_3''(f; x_i) = M_i,\ s_3''(f; x_{i+1}) = M_{i+1},\ \ i = 1, 2, \ldots, n-1.$$

The values a and b are determined from the system

$$\left\{ \begin{array}{rcrcl} ax_i & + & b & = & M_i \\ ax_{i+1} & + & b & = & M_{i+1} \end{array} \right. .$$

Integrating successively, then imposing (2.16) and the continuity conditions at the nodes, $s_3'(f; x_i) = s_3'(f; x_{i+1}),\ i = \overline{1, n-1}$, we get the linear system

$$h_{i-1}\, M_{i-1} \ + \ 2\,(h_{i-1} + h_i)\, M_i \ + \ h_i\, M_{i+1} \ = \ 6\,(f[x_i, x_{i+1}] - f[x_{i-1}, x_i])\,, \tag{2.17}$$

for $i = \overline{2, n-1}$.

The two extra conditions needed for a closed system can be imposed, e.g., on M_1 and M_n. If $M_1 = M_n = 0$, we get the natural cubic spline.

Other conditions can be enforced, such as the continuity of $s_3'''(f; x)$ at $x = x_2$ and $x = x_{n-1}$, which lead to deBoor cubic splines.

If the first and last equations are

$$\begin{array}{rcrcl} 2M_1 & + & M_2 & = & 6\,(f[x_1, x_2] - f_1')\,, \\ M_{n-1} & + & 2M_n & = & 6\,(f_n' - f[x_{n-1}, x_n])\,, \end{array} \tag{2.18}$$

where $f_1' = f'(a), f_n' = f'(b)$, then the resulting function is the complete cubic spline.

Example 2.1. Find the natural cubic spline that interpolates the data

x_i	1	2	4	5
f_i	3	5	9	10

Solution.

We have $n = 4$ nodes and $h_1 = 1, h_2 = 2, h_3 = 1$.

From (2.9)–(2.13), the linear system for the unknowns m_i, also called *slopes*, is

$$\begin{cases} 2m_1 + m_2 = 6 \\ 2m_1 + 6m_2 + m_3 = 18 \\ 2m_2 + 6m_3 + 2m_4 = 12 \\ m_3 + 2m_4 = 3 \end{cases}$$

with solution

$$m_1 = \frac{87}{46},\ m_2 = \frac{51}{23},\ m_3 = \frac{21}{23},\ m_4 = \frac{24}{23}.$$

The system (from (2.17) together with the conditions $M_1 = M_4 = 0$) for the *moments* M_i becomes

$$\begin{cases} M_1 = 0 \\ M_1 + 6M_2 + 2M_3 = 0 \\ 2M_2 + 6M_3 + M_4 = -6 \\ M_4 = 0 \end{cases}$$

whose solution is

$$M_1 = 0,\ M_2 = \frac{3}{8},\ M_3 = -\frac{9}{8},\ M_4 = 0.$$

Hence, both ways, we get the natural cubic spline function

$$s_3(x) = \begin{cases} \dfrac{x^3}{16} - \dfrac{3x^2}{16} + \dfrac{17x}{8} + 1, & x \in [1,2] \\ -\dfrac{x^3}{8} + \dfrac{15x^2}{16} - \dfrac{x}{8} + \dfrac{5}{2}, & x \in [2,4] \\ \dfrac{3x^3}{16} - \dfrac{45x^2}{16} + \dfrac{119x}{8} - \dfrac{35}{2}, & x \in [4,5] \end{cases}.$$

■

Minimality properties of cubic spline interpolants

Natural and complete splines have interesting optimality properties. Henceforth, we denote them by $s_{nat}(f;\cdot)$ and $s_{compl}(f;\cdot)$, respectively.

Theorem 2.2. *Let $g \in C^2[a,b]$ be any function that interpolates f on Δ. Then*

$$\int_a^b |s''_{nat}(f;x)|^2 dx \;\leq\; \int_a^b |g''(x)|^2 dx, \tag{2.19}$$

with equality if and only if $g(\cdot) = s_{nat}(f;\cdot)$.

For the next minimality result, we slightly change the subdivision Δ. Consider the grid

$$\Delta' \;:\; a = x_0 = x_1 < x_2 < \cdots < x_{n-1} < x_n = x_{n+1} = b, \tag{2.20}$$

where the endpoints are *double* nodes. That means that when we use Δ', we interpolate the function values at all interior points, and, both the functional and the derivative values, at the endpoints.

Theorem 2.3. *Let $g \in C^2[a,b]$ be any function that interpolates f on Δ'. Then*

$$\int_a^b |s''_{compl}(f;x)|^2 dx \;\leq\; \int_a^b |g''(x)|^2 dx, \tag{2.21}$$

with equality if and only if $g(\cdot) = s_{compl}(f;\cdot)$.

Remark 2.4. Taking $g(\cdot) = s_{compl}(f;\cdot)$ in Theorem 2.2, we get

$$\int_a^b |s''_{nat}(f;x)|^2 dx \;\leq\; \int_a^b |s''_{compl}(f;x)|^2 dx. \tag{2.22}$$

So, in a sense, the natural cubic spline is the "smoothest" interpolant.

Remark 2.5. These minimality properties are at the origin of the name "spline". A *spline* is a flexible strip of wood used in drawing curves (or a musical instrument in that shape).

Example 2.6. Consider the function $f(x) = \arctan x,\ x \in [-2,2]$ and the nodes $\{-2,-1,0,1,2\}$. Figure 1 shows the graphs of the function f, the nodes and the complete, natural, deBoor and piecewise Hermite cubic splines interpolating f. In Figure 2 we have the interpolation errors.

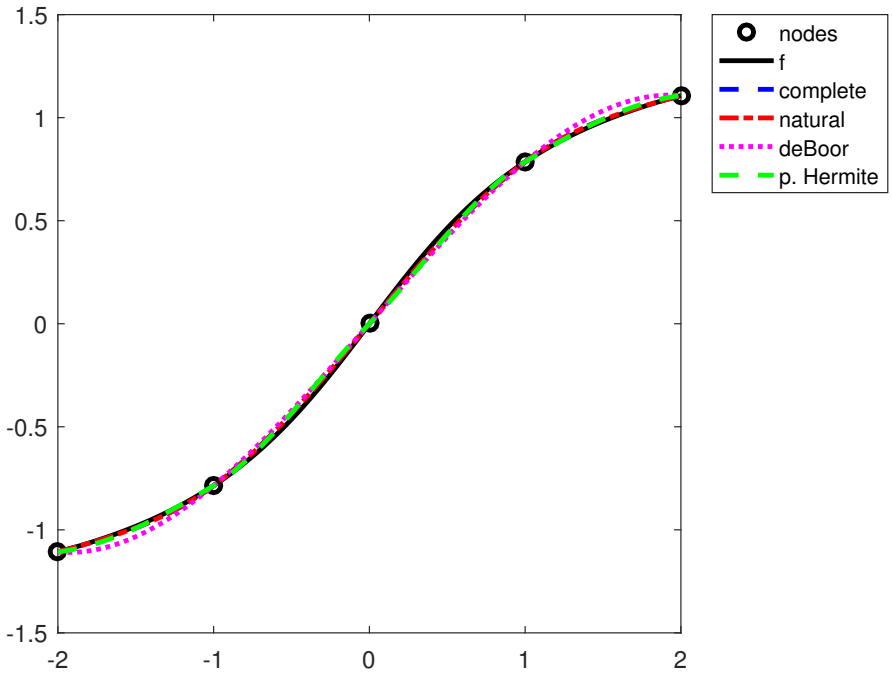

Fig. 1: Interpolation with cubic splines, $f(x) = \arctan x$

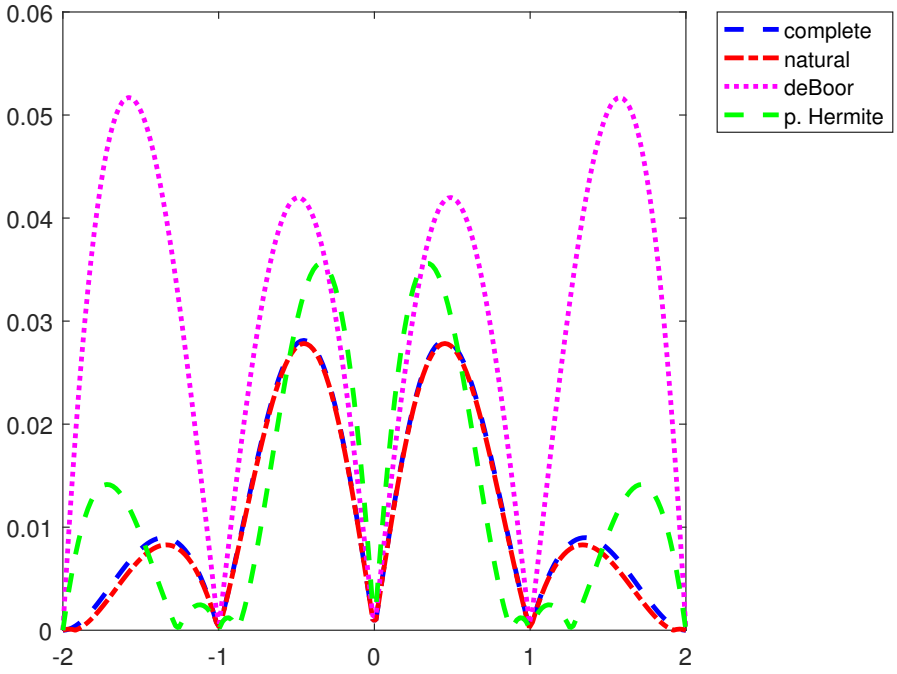

Fig. 2: Errors in cubic spline interpolation, $f(x) = \arctan x$

3 Least Squares Approximation

3.1 Best approximation problem

In general, an approximation problem can be described as follows: Let $f \in X$ be a function, Φ, a family of approximants and $||\cdot||$ a norm on X. We seek an approximation $\hat{\varphi} \in \Phi$ of f that approximates the given function "as well as possible".

$$||f - \hat{\varphi}|| \quad \leq \quad ||f - \varphi||, \ \forall \varphi \in \Phi. \tag{3.1}$$

This is called a *best approximation problem* of f with elements of Φ. The function $\hat{\varphi}$ is called a *best approximation* of f relative to the norm $||\cdot||$. Given a basis $\{\pi_j\}_{j=1}^m$ of Φ, we can write

$$\Phi \ = \ \Phi_m \quad = \quad \Big\{\varphi \mid \varphi(t) = \sum_{j=1}^{m} c_j \pi_j(t), \ c_j \in \mathbb{R}\Big\}. \tag{3.2}$$

Φ is a finite dimensional linear space or a subset of one.

In the preceding sections we gave a polynomial (i.e., we used $\Phi = \mathbb{P}_m$) or piecewise polynomial (with $\Phi = S_m^k(\Delta)$) approximation based on using interpolation at suitably chosen node points. Another approach is to seek an approximation with a small "average error" over the interval of approximation. That "average error" can be best expressed in terms of inner products and norms.

3.2 Scalar Products and Norms

The functions we want to approximate can be defined *continuously*, i.e., on an interval $[a, b]$, or *discreetly*, on a set of points $\{t_1, \ldots, t_N\}$. A *measure* will de defined accordingly, as an *integral* in the continuous case and as a *sum* for discrete functions.

Many measures also involve *weight functions*. An intuitive, physical justification for a *weighted measure* would be that some observations are more important than others, or they are more common, so they "weigh more".

Definition 3.1. *A **weight function** w is defined as follows:*
$-$ **continuous case**, *a function $w : [a, b] \to \mathbb{R}_+$, satisfying the conditions*

$$\text{(i)} \quad \int_a^b |x|^n w(x)\, dx \text{ exists and is finite }, \ \forall n \geq 0,$$
$$\text{(ii)} \quad \text{if} \int_a^b w(x)g(x)\, dx = 0, \ g(x) \geq 0, \text{ then } \ g \equiv 0;$$

$-$ **discrete case**, *$w_i \geq 0$, satisfying the conditions*

$$\text{(i)} \quad \sum_{i=1}^{N} |t_i|^n w(t_i) \text{ exists and is finite }, \ \forall n \geq 0,$$
$$\text{(ii)} \quad \text{if } \sum_{i=1}^{N} w_i g_i = 0, \ g_i \geq 0, \text{ then } \ g_i = 0, \forall i = \overline{1, N}.$$

A few commonly used continuous weights:

$$\begin{aligned} w(x) &\equiv 1 && \text{on} \quad [-1,\ 1], \\ w(x) &= \frac{1}{\sqrt{1-x^2}} && \text{on} \quad [-1,\ 1], \\ w(x) &= \sqrt{1-x^2} && \text{on} \quad [-1,\ 1], \\ w(x) &= e^{-x} && \text{on} \quad [0,\ \infty), \\ w(x) &= e^{-x^2} && \text{on} \quad (-\infty,\ \infty). \end{aligned}$$

Definition 3.2. *Let w be a weight function. The **scalar (inner) product** of two functions u and v is defined as*

$$< u, v > \ = \ \int_a^b w(x)u(x)v(x)\, dx \ \text{ for continuous functions and}$$
$$< u, v > \ = \ \sum_{i=1}^{N} w_i u_i v_i \ \text{ in the discrete case.}$$

*The **norm** of a function u is*

$$||u|| \ = \ \big(< u, u > \big)^{\frac{1}{2}}.$$

The most frequently used (discrete and continuous) norms are given in Table 1.

Discrete norm	Continuous norm
$\|\|u\|\|_p = \Big(\sum_{i=1}^{N} w_i \left\|u(t_i)\right\|^p\Big)^{1/p},\ p \geq 1$	$\|\|u\|\|_p = \Big(\int_a^b w(x)\left\|u(x)\right\|^p dx\Big)^{1/p},\ p \geq 1$
$\|\|u\|\|_\infty = \max_{i=\overline{1,m}} \|u(t_i)\|$	$\|\|u\|\|_\infty = \max_{x\in[a,b]} \|u(x)\|$

Table 1: Commonly used discrete and continuous norms

Let us recall the main properties of scalar products:

1. **Symmetry**: $< u, v > = < v, u >$;
2. **Homogeneity**: $< \alpha u, v > = < u, \alpha v > = \alpha < u, v >,\ \alpha \in \mathbb{R}$;
3. **Additivity**: $< u + v, z > = < u, z > + < v, z >$;
4. **Positive definiteness**: $< u, u > \geq 0$ and $< u, u > = 0 \Longleftrightarrow u = 0$;
5. **Cauchy–Bunyakovsky–Schwarz inequality**: $| < u, v > | \leq ||u|| \cdot ||v||$.

Definition 3.3. *We say that two functions u and v are **orthogonal** if*

$$< u, v > \ = \ 0.$$

*More generally, we say that a family of functions $\{u_k\}_{k=\overline{1,n}}$ is an **orthogonal system** if*

$$< u_i, u_j > \ = \ 0,\ i \neq j.$$

*They are called **orthonormal** if*

$$< u_i, u_j > \ = \ \delta_{ij}.$$

3.3 Least Squares Approximation

We will consider a particular case for problem (3.1) by choosing the (discrete or continuous) 2-norm. In what follows, $||\cdot||$ will mean $||\cdot||_2$. So, we want to minimize the *square of the error*

$$\begin{aligned} E^2(\varphi) &= ||\varphi - f||^2 = <\varphi - f, \varphi - f> \\ &= <\varphi, \varphi> -2<\varphi, f> + <f, f> \\ &= ||\varphi||^2 - 2<\varphi, f> + ||f||^2. \end{aligned} \tag{3.3}$$

This is then called a *least squares approximation problem* or *mean square approximation problem*. Its solution was given by Gauss and Legendre at the beginning of the 19th century.

3.3.1 Normal equations

Recall that we seek a function $\varphi \in \Phi_m$ that minimizes E^2 in (3.3). Then

$$\varphi(t) = \sum_{j=1}^{m} c_j \pi_j(t).$$

Substitute φ into (3.3) to get

$$\begin{aligned} E^2(\varphi) &= <\sum_{i=1}^{m} c_i\pi_i, \sum_{j=1}^{m} c_j\pi_j> -2<\sum_{j=1}^{m} c_j\pi_j, f> + ||f||^2 \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j <\pi_i, \pi_j> -2\sum_{j=1}^{m} c_j <\pi_j, f> + ||f||^2. \end{aligned}$$

We want to find the values $c_j \in \mathbb{R}$ that minimize the error above. We solve this problem (finding the minimum of a function) by taking partial derivatives with respect to each unknown c_i and setting them equal to 0. Thus, we get

$$\frac{\partial E^2}{\partial c_i} = 2\sum_{j=1}^{m} c_j <\pi_i, \pi_j> -2<\pi_i, f> = 0,\ i = 1, \ldots, m,$$

or

$$\sum_{j=1}^{m} <\pi_i, \pi_j> c_j = <\pi_i, f>,\ i = 1, \ldots, m. \tag{3.4}$$

The equations in (3.4) are called **normal equations** and they form a linear system

$$Ac \;=\; b, \tag{3.5}$$

with

$$a_{ij} \;=\; <\pi_i, \pi_j> \;\text{ and }\; b_i \;=<\pi_i, f>. \tag{3.6}$$

Now, since the scalar product is symmetric, so is matrix A. Also, for every $x \in \Phi, x \neq 0$,

$$\begin{aligned} x^T Ax \;&=\; \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}x_i x_j \;=\; \sum_{i=1}^{m}\sum_{j=1}^{m} x_i x_j <\pi_i, \pi_j> \\ &=\; \sum_{i=1}^{m}\sum_{j=1}^{m} <x_i\pi_i, x_j\pi_j> = \left|\left|\sum_{i=1}^{m} x_i\pi_i\right|\right|^2 \;>\; 0. \end{aligned}$$

So A is symmetric and positive definite, therefore, nonsingular. Thus, the system (3.5) has a unique solution $c_j^*,\ j = 1, \ldots, m$, and hence, so does the least squares approximation problem,

$$\varphi^*(t) \;=\; \sum_{j=1}^{m} c_j^*\pi_j(t). \tag{3.7}$$

Example 3.4. Find the linear least squares approximation of the function $f(x) = \cos \pi t$ on $[0, 1]$, using the canonical basis $\pi_j(t) \;=\; t^j,\ j \in \mathbb{N}$.

Solution. The function is given on an interval, so we want the *continuous* least squares approximation. Since we want a *linear* approximation, i.e., a polynomial of degree 1, we have the basis

$$\pi_1(t) \;=\; 1,\;\; \pi_2(t) \;=\; t$$

and we seek an approximation polynomial

$$\varphi(t) \;=\; c_1\pi_1(t) + c_2\pi_2(t) \;=\; a + bt,$$

with simplified notation $c_1 = a, c_2 = b$. The normal equations (3.4) are

$$\begin{aligned} <\pi_1, \pi_1> a \;&+\; <\pi_1, \pi_2> b \;&=\; <\pi_1, f>, \\ <\pi_2, \pi_1> a \;&+\; <\pi_2, \pi_2> b \;&=\; <\pi_2, f>, \end{aligned}$$

with

$$< \pi_1, \pi_1 > \;=\; \int_0^1 dt \;=\; 1, \;\; < \pi_1, \pi_2 > = \int_0^1 t dt \;=\; 1/2, \;\; < \pi_2, \pi_2 > = \int_0^1 t^2 dt \;=\; 1/3,$$

$$< \pi_1, f > \;=\; \int_0^1 \cos \pi t \; dt \;=\; \frac{1}{\pi} \sin \pi t \Big|_0^1 \;=\; 0,$$

$$< \pi_2, f > \;=\; \int_0^1 t \cos \pi t \; dt \;=\; \frac{1}{\pi} \, t \sin \pi t \Big|_0^1 - \frac{1}{\pi} \int_0^1 \sin \pi t \; dt \;=\; \frac{1}{\pi^2} \cos \pi t \Big|_0^1 \;=\; -\frac{2}{\pi^2},$$

the last one being integrated by parts ($u = t, \;\; dv = \cos \pi t \; dt$). Then we solve the system

$$\begin{cases} a \;+\; \dfrac{1}{2} b \;=\; 0, \\ \dfrac{1}{2} a \;+\; \dfrac{1}{3} b \;=\; -\dfrac{2}{\pi^2} \end{cases},$$

with solution $a = \dfrac{12}{\pi^2}, \;\; b = -\dfrac{24}{\pi^2}$.

So, we found the linear least squares approximation

$$\varphi^*(t) \;=\; \frac{12}{\pi^2}(1 - 2t).$$

Let us look at the error at some points:

t	$f(t)$	$\varphi^*(t)$	$\lvert f(t) - \varphi^*(t) \rvert$
0	1	$12/\pi^2$	$12/\pi^2 - 1 \;\approx\; 0.22$
$1/6$	$\sqrt{3}/2$	$8/\pi^2$	$\sqrt{3}/2 - 8/\pi^2 \;\approx\; 0.06$
$1/4$	$\sqrt{2}/2$	$6/\pi^2$	$\sqrt{2}/2 - 6/\pi^2 \;\approx\; 0.01$
$1/3$	$1/2$	$4/\pi^2$	$1/2 - 4/\pi^2 \;\approx\; 0.09$
$1/2$	0	0	0
1	-1	$-12/\pi^2$	$12/\pi^2 - 1 \;\approx\; 0.22$

■

When we seek the *discrete* least squares approximation of some scattered data, the problem is known as *data fitting*, and it arises in many applications.

Example 3.5. Find the least squares polynomial approximation that best fits the following data:

x_i	-5	-3	-1	1	3	5
y_i	4.8	3.0	2.0	2.8	3.2	10

Solution. The scatterplot is shown in Figure 3. We see from the graph that the best fit is given by a quadratic function,

$$\varphi(x) = a + bx + cx^2,$$

i.e., the basis is $1, x, x^2$ (the canonical basis again).

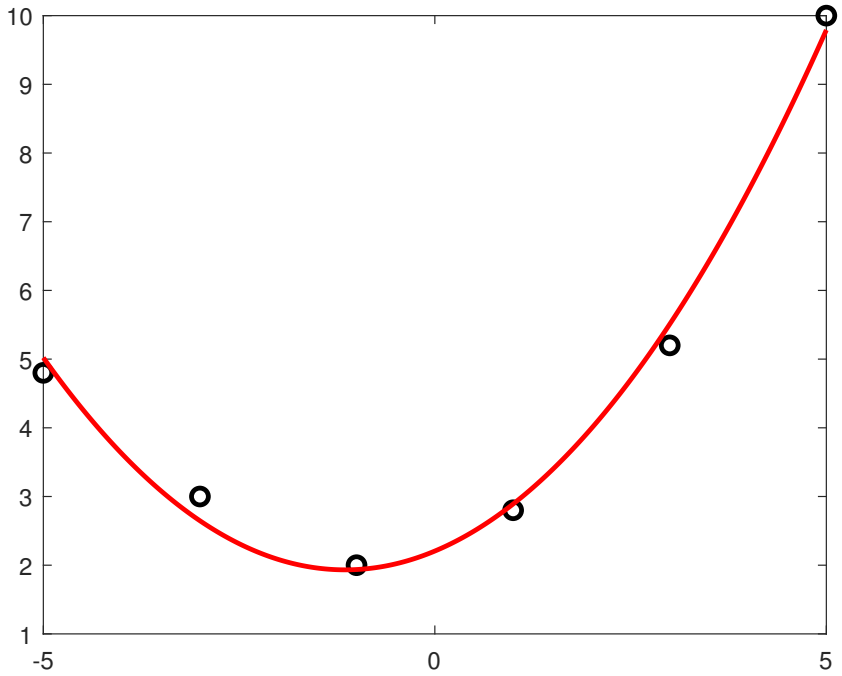

Fig. 3: Data fitting, Example 3.5

The normal equations (3.4) are

$$\begin{aligned}
&< \pi_1, \pi_1 > a \;+\; < \pi_1, \pi_2 > b \;+\; < \pi_1, \pi_3 > c \;=\; < \pi_1, y >,\\
&< \pi_2, \pi_1 > a \;+\; < \pi_2, \pi_2 > b \;+\; < \pi_2, \pi_3 > c \;=\; < \pi_2, y >,\\
&< \pi_3, \pi_1 > a \;+\; < \pi_3, \pi_2 > b \;+\; < \pi_3, \pi_3 > c \;=\; < \pi_3, y >,
\end{aligned}$$

with

$$< \pi_i, \pi_j > \ = \ \sum_{k=1}^{6} x_k^i x_k^j \ = \ \sum_{k=1}^{6} x_k^{i+j} \ \text{ and}$$
$$< \pi_i, y > \ = \ \sum_{k=1}^{6} x_k^i y_k, \ i, j = \overline{0, 2}, \ k = 1, \ldots, 6.$$

So, the normal equations are

$$\begin{aligned} a\sum_{k=1}^{6} 1 + b\sum_{k=1}^{6} x_k + \ c\sum_{k=1}^{6} x_k^2 &= \sum_{k=1}^{6} y_k, \\ a\sum_{k=1}^{6} x_k + b\sum_{k=1}^{6} x_k^2 + c\sum_{k=1}^{6} x_k^3 &= \sum_{k=1}^{6} x_k y_k, \\ a\sum_{k=1}^{6} x_k^2 + b\sum_{k=1}^{6} x_k^3 + c\sum_{k=1}^{6} x_k^4 &= \sum_{k=1}^{6} x_k^2 y_k. \end{aligned}$$

The sums are computed in the following table (on the last row)

	x_k	y_k	x_k^2	x_k^3	x_k^4	$x_k y_k$	$x_k^2 y_k$
	-5	4.8	25	-125	625	-24.0	120
	-3	3.0	9	-27	81	-9.0	27
	-1	2.0	1	-1	1	2.0	2
	1	2.8	1	1	1	2.8	2.8
	3	5.2	9	27	81	15.6	46.8
	5	10.0	25	125	625	50	250
$\sum_{k=1}^{6}$	0	27.8	70	0	1414	33.4	448.6

The resulting linear system is

$$\begin{cases} 6a & & + & 70c & = & 27.8 \\ & 70b & & & = & 33.4 \\ 70a & & + & 1414c & = & 448.6 \end{cases},$$

with solution $a = 2.206, \ b = 0.477, c = 0.208$. The best fit approximation of this data is

$$\varphi^*(x) \ = \ 0.208x^2 + 0.477x + 2.206.$$

■

3.3.2 Orthogonal polynomials

Least square approximations can use *other functions*, not just polynomials, and bases *other than canonical* can be used, as well. The ideas and procedures described in the previous section still apply.

As we have seen in this chapter, polynomials or piecewise polynomials work quite well and it is rarely the case when other, more complicated approximating functions have to be used.

As far as the basis is concerned, in the continuous case, some choices are better than others and things *can be improved* there. Let us point out a few troublesome aspects:

- For continuous least squares approximations, the linear system $Ac = b$ in (3.5)-(3.6) can be ill-conditioned. If the canonical basis, $\pi_j(t) = t^{j-1}, j = \overline{1,m}$, is used on the interval $[0,1]$ (as in Example 3.4), then

$$a_{ij} \;=\; <\pi_i,\pi_j> = \int_0^1 t^{i+j-2}dt \;=\; \frac{1}{i+j-1},\;\; i,j=\overline{1,m},$$

  i.e., $A = H_m$, the Hilbert matrix, which is known to be very ill-conditioned. The basis functions become almost linearly dependent, as the exponent grows. The solution of the linear system (3.5) is extremely sensitive to small changes in the coefficients or right-hand constants and as a consequence, when $m \geq 4$, the solutions will be completely unsatisfactory.

- Another disadvantage is that all the coefficients c_j found this way depend on m, $c_j = c_j^{(m)}$. Increasing m will produce an enlarged system of normal equations with a *completely new* solution vector. There is no relation between $c_j^{(m)}$ and $c_j^{(m+n)}$, so no way of using previous computations.

Both these problems can be overcome if the basis $\{\pi_j\}_{j=1}^m$ is chosen to be **orthogonal**. If $<\pi_i,\pi_j> = 0,\; i \neq j$, then the coefficients $a_{ij} = 0,\; i \neq j$, which means the system $Ac = b$ is *diagonal* with solution

$$c_j^* \;=\; \frac{<\pi_j, f>}{<\pi_j,\pi_j>},\; j = 1,\ldots m \tag{3.8}$$

and the least squares approximation is

$$\varphi^*(t) \;=\; \sum_{j=1}^{m} \frac{<\pi_j, f>}{<\pi_j,\pi_j>}\pi_j(t). \tag{3.9}$$

Now, instead of solving a system of normal equations, we can use formula (3.8) directly. Obviously, the coefficients c_j^* are independent of m and once computed, they remain the same for any larger m. We now have what is called *permanence of the coefficients*.

Another aspect: recall from linear algebra that any linearly independent system can be orthogonalized using the *Gram-Schmidt procedure*. So using an orthogonal basis is *not* restrictive at all.

In fact, applying that procedure to the canonical basis $1, t, t^2, \ldots$ on an interval $[a, b]$, with respect to an appropriate weight function w, several well-known families of orthogonal polynomials can be obtained, including the Chebyshev polynomials (of the first and second kind) that were already discussed. Also, that procedure provides a linear recurrence relation between 3 consecutive such orthogonal polynomials:

$$\pi_{k+1}(t) \;=\; (t-\alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t),\ k=0,1,\ldots,\ \ \pi_{-1}(t) \;=\; 0,\ \ \pi_0(t) \;=\; 1, \quad (3.10)$$

where

$$\alpha_k \;=\; \frac{< t\pi_k, \pi_k >}{||\pi_k||^2},\ k=0,1,\ldots,\ \ \beta_k \;=\; \frac{||\pi_k||^2}{||\pi_{k-1}||^2},\ k=1,2,\ldots,\ \ \beta_0 \;=\; \mu_0. \quad (3.11)$$

Such examples are given in Table 2.

Name	**Notation**	**Polynomial**	**Weight fn.**	**Interval**	$\boldsymbol{\alpha_k}$	$\boldsymbol{\beta_k}$
Legendre	l_m	$\left[(x^2-1)^m\right]^{(m)}$	1	$[-1,1]$	0	$\beta_0 = 2,$ $\beta_k = (4-k^{-2})^{-1}, k \geq 1$
Chebyshev 1^{st}	T_m	$\cos(m \arccos x)$	$(1-x^2)^{-\frac{1}{2}}$	$[-1,1]$	0	$\beta_0 = \pi,$ $\beta_1 = \frac{1}{2},$ $\beta_k = \frac{1}{4}, k \geq 2$
Chebyshev 2^{nd}	Q_m	$\frac{\sin[(m+1)\arccos x]}{\sqrt{1-x^2}}$	$(1-x^2)^{\frac{1}{2}}$	$[-1,1]$	0	$\beta_0 = \frac{\pi}{2},$ $\beta_k = \frac{1}{4}, k \geq 1$
Laguerre	L_m^a	$x^{-a}e^x\left(x^{m+a}e^{-x}\right)^{(m)}$	$x^a e^{-x},\ a > -1$	$[0,\infty)$	$2k+a+1$	$\beta_0 = \Gamma(1+a),$ $\beta_k = k(k+a), k \geq 1$
Hermite	H_m	$(-1)^m e^{x^2}\left(e^{-x^2}\right)^{(m)}$	e^{-x^2}	$\mathbb{R}$	0	$\beta_0 = \sqrt{\pi},$ $\beta_k = \frac{k}{2}, k \geq 1$

Table 2: Orthogonal polynomials and recurrence coefficients

Example 3.6. Let $f : [-1,1] \to [0,\pi], f(t) = \arccos t$. Find the least squares polynomial approximation φ^* of f relative to the weight function $w(t) = \dfrac{1}{\sqrt{1-t^2}}$.

Solution. For this weight function, we use Chebyshev polynomials of the first kind, $\pi_j(t) = T_j(t)$. Recall that

$$< \pi_i, \pi_j > \;=\; \int_{-1}^{1} \frac{T_i(t)T_j(t)}{\sqrt{1-t^2}}\, dt \;=\; \begin{cases} 0, & i \neq j \\ \pi, & i = j = 0 \\ \dfrac{\pi}{2}, & i = j \neq 0 \end{cases},$$

so

$$< \pi_0, \pi_0 > = \pi \;\text{ and }\; < \pi_j, \pi_j > = \frac{\pi}{2},\; j \neq 0.$$

Now, compute the numerators of the coefficients c_j^*.

$$< \pi_j, f > \;=\; \int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} \cos\left(j \arccos t\right) \arccos t \; dt.$$

With the change of variables $t = \cos x$, we have $\arccos t = x,\; \sqrt{1-t^2} = \sin x,\; dt = -\sin x \; dx$ and the interval $[-1, 1]$ is mapped into $[\pi, 0]$. So,

$$< \pi_j, f > \;=\; \int_{0}^{\pi} \frac{1}{\sin x} x \cos\left(jx\right) \sin x \; dx \;=\; \int_{0}^{\pi} x \cos\left(jx\right) \, dx.$$

Then,

$$< \pi_0, f > \;=\; \int_{0}^{\pi} x \; dx \;=\; \frac{1}{2}x^2 \Big|_0^{\pi} \;=\; \frac{\pi^2}{2}.$$

For $j \neq 0$, we use integration by parts with $u = x,\; dv = \cos\left(jx\right) dx$ $\Big($so $du = dx,\; v = \dfrac{1}{j}\sin\left(jx\right)\Big)$:

$$\begin{aligned} < \pi_j, f > \;&=\; \int_{0}^{\pi} x \cos\left(jx\right) \, dx \;=\; \frac{1}{j} x \sin\left(jx\right) \Big|_0^{\pi} - \frac{1}{j} \int_{0}^{\pi} \sin\left(jx\right) \, dx \\ &=\; 0 + \frac{1}{j^2} \cos\left(jx\right) \Big|_0^{\pi} \;=\; \frac{1}{j^2}\Big(\cos\left(j\pi\right) - \cos 0 \Big) \;=\; \frac{1}{j^2}\Big((-1)^j - 1 \Big) \\ &=\; \begin{cases} 0, & j \text{ even} \\ -\dfrac{2}{j^2}, & j \text{ odd} \end{cases}. \end{aligned}$$

So,

$$c_0^* \;=\; \frac{\pi}{2},\;\; c_j^* \;=\; \begin{cases} 0, & j \neq 0 \;\text{ even} \\ -\dfrac{4}{j^2\pi}, & j \;\text{ odd} \end{cases} .$$

Then the least squares approximating polynomial is given by

$$\varphi_{2n+1}^*(t) \;=\; \frac{\pi}{2} - \sum_{i=0}^{n} \frac{4}{(2i+1)^2\pi} T_{2i+1}(t).$$

This is the solution of the least squares problem:

$$\int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} \big(f(t) - \varphi_n^*(t)\big)^2 \, dt \;=\; \min \left\{ \int_{-1}^{1} \frac{1}{\sqrt{1-t^2}} \big(f(t) - \varphi(t)\big)^2 \, dt \;:\; \varphi \in \mathbb{P}_n \right\}.$$

The approximation is quite good, even for small degrees, as seen in Figure 4.

■

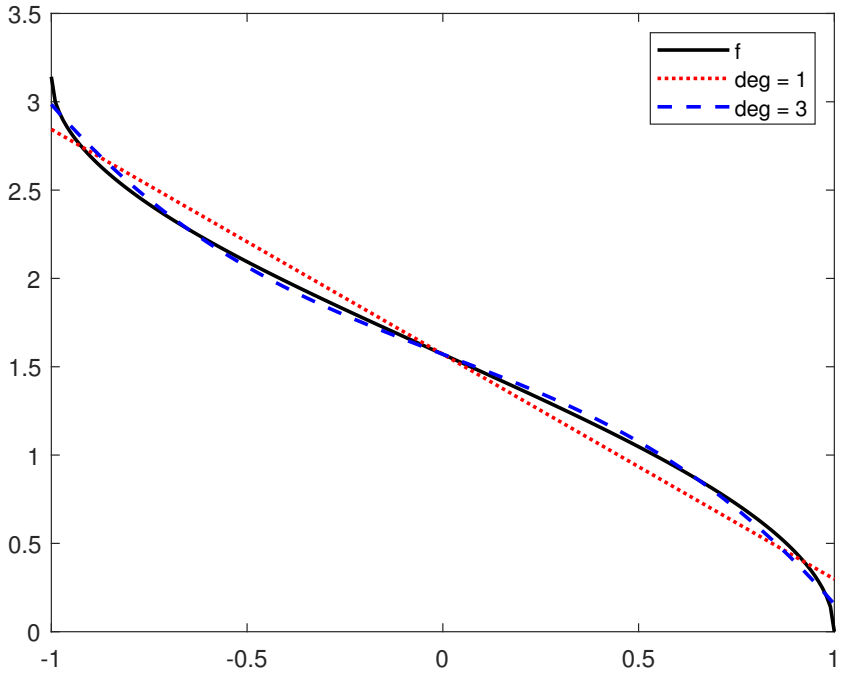

Fig. 4: Example 3.6