Chapter 3. Approximation of Functions

Approximation of functions is one of the most important tasks in Numerical Analysis.

Most functions encountered in mathematical problems and applications cannot be evaluated exactly, even though we usually handle them as if they where completely known quantities. The simplest and most important of these are $\sqrt{x}, e^x, \log x$, and the trigonometric functions; and there are many other functions that occur commonly in physics, engineering, and other disciplines. In evaluating functions, by hand or using a computer, we are essentially limited to the elementary arithmetic operations $+, -, \times$ and $\div$. Combining these operations means that we can evaluate polynomials and rational functions, which are polynomials divided by polynomials. All other functions must be evaluated by using approximations based on polynomials or rational functions, including piecewise variants of them (e.g., spline functions). Although rational functions generally give slightly more efficient approximations, polynomials are adequate for most problems and their theory is much easier to handle.

Interpolation is the process of finding and evaluating a function whose graph goes through a set of given points. The points may arise as measurements in a physical problem, or they may be obtained from a known function. The interpolating function is usually chosen from a restricted class of functions and *polynomials* are the most commonly used class.

Interpolation is an important tool in producing computable approximations to commonly used functions. Moreover, to numerically integrate or differentiate a function, we often replace the function with a simpler approximating expression, which is then integrated or differentiated. These simpler expressions are almost always obtained by interpolation. Also, some of the most widely used numerical methods for solving differential equations are obtained from interpolating approximations. Finally, interpolation is widely used in computer graphics, to produce smooth curves and surfaces when the geometric object of interest is given at only a discrete set of data points.

Later on, we will briefly consider other forms of function approximations.

1 Polynomial Interpolation

1.1 Lagrange Interpolation

Interpolation problem. Given $n+1$ distinct points – called *nodes (or knots)* – $x_i \in [a,b], i = \overline{0,n}$, and the values $f(x_i) = y_i$ of an unknown function $f : [a,b] \to \mathbb{R}$, find a polynomial $P(x)$ of

minimum degree, satisfying

$$P(x_i) \;=\; f(x_i),\ i = \overline{0,n}, \tag{1.1}$$

called *interpolation conditions*. This polynomial approximates function f.

Linear interpolation

We start with a simple case: consider two interpolation nodes, $(x_0, y_0), (x_1, y_1), x_0 \neq x_1$.
We know that there is a unique *line* passing through these points. That means we can find a polynomial of degree 1 that interpolates the data. Let us find it.

The slope of the line is

$$m \;=\; \frac{y_1 - y_0}{x_1 - x_0}$$

and its equation is

$$y - y_0 \;= \frac{y_1 - y_0}{x_1 - x_0}(x - x_0).$$

We find the linear interpolation polynomial as

$$\begin{aligned} P_1(x) &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \\ &= \left(1 - \frac{x - x_0}{x_1 - x_0}\right) y_0 + \frac{x - x_0}{x_1 - x_0} y_1 \\ &= \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1. \end{aligned}$$

Example 1.1. Consider the function $f(x) = \sqrt{x}$ and the nodes $x_0 = 1, x_1 = 4$, i.e. the data $(1, 1), (4, 2)$.

Solution. The linear interpolation polynomial is

$$P_1(x) \;=\; \frac{x - 4}{1 - 4} \cdot 1 + \frac{x - 1}{4 - 1} \cdot 2 \;=\; \frac{1}{3}x + \frac{2}{3}.$$

The graphs of f and P_1 on the interval $[0, 15]$ are shown in Figure 1.

■

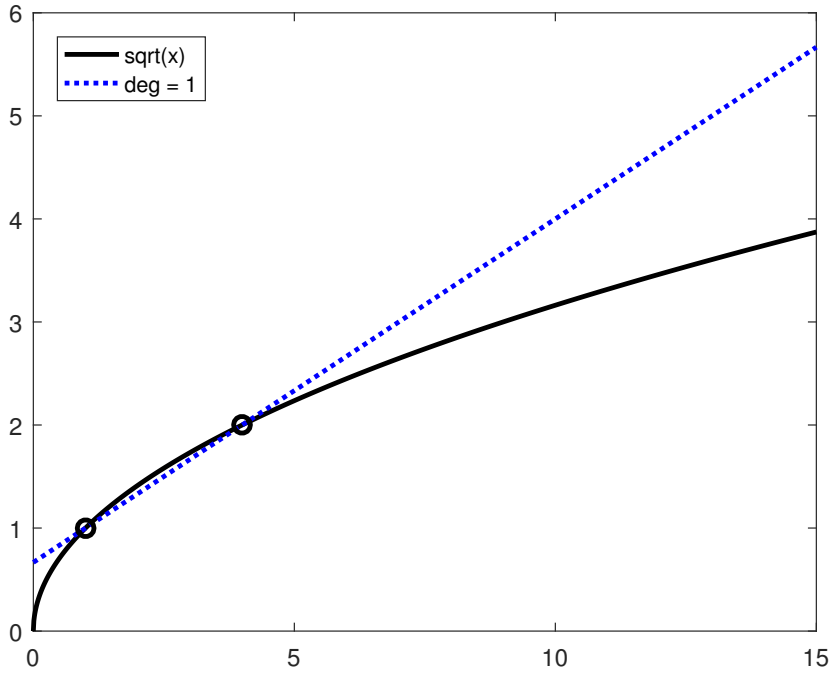

Fig. 1: Linear interpolation of function $\sqrt{x}$

Quadratic interpolation

We go further and consider 3 distinct nodes $(x_0, y_0), (x_1, y_1), (x_2, y_2)$. It can easily be checked that the quadratic polynomial

$$P_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2$$

interpolates these data.

Example 1.2. In Example 1.1 we add the node $(9, 3)$.

Solution. With nodes $(1, 1), (4, 2)$ and $(9, 3)$, the quadratic interpolation polynomial is given by

$$\begin{aligned} P_2(x) &= \frac{(x-4)(x-9)}{(1-4)(1-9)} \cdot 1 + \frac{(x-1)(x-9)}{(4-1)(4-9)} \cdot 2 + \frac{(x-1)(x-4)}{(9-1)(9-4)} \cdot 3 \\ &= -\frac{1}{60}x^2 + \frac{5}{12}x + \frac{3}{5}. \end{aligned}$$

The graphs of f and the two interpolation polynomials P_1 and P_2 are plotted in Figure 2. ■

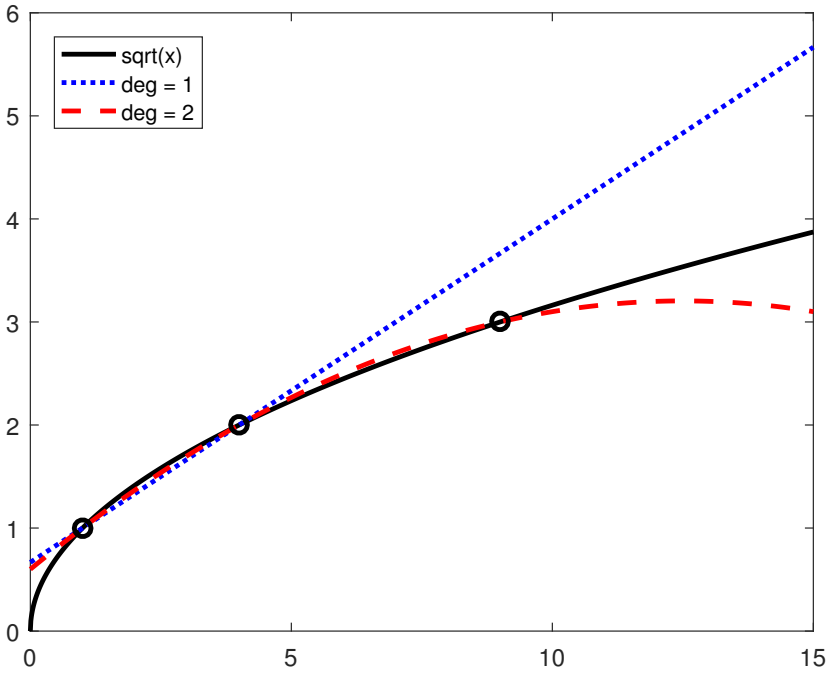

Fig. 2: Linear and quadratic interpolation of function $\sqrt{x}$

General case

Consider the interval $[a,b] \subset \mathbb{R}$, a function $f : [a,b] \to \mathbb{R}$ and a set of $n+1$ distinct nodes $\{x_0, x_1, \ldots, x_n\} \subset [a,b]$.
Recall the notations

$$\begin{aligned} u(x) &= \prod_{j=0}^{n}(x - x_j) = (x - x_0)(x - x_1)\ldots(x - x_n), \\ u_j(x) &= \frac{u(x)}{x - x_j},\ j = 0, 1, \ldots, n. \end{aligned} \tag{1.2}$$

Theorem 1.3. *There is a unique polynomial $L_n f$ of degree at most n, satisfying the interpolation conditions* (1.1)*. This polynomial can be written as*

$$L_n f(x) = \sum_{i=0}^{n} l_i(x) f(x_i), \tag{1.3}$$

where

$$l_i(x) \;=\; \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j} \;=\; \frac{u_i(x)}{u_i(x_i)} \;=\; \frac{u_i(x)}{u'(x_i)}. \tag{1.4}$$

*L_nf is called the **Lagrange interpolation polynomial** of f at the nodes $x_0, x_1, \ldots, x_n$. The functions $l_i(x), i = \overline{0,n}$ are called **Lagrange fundamental (basis) polynomials** associated with these points.*

Proof. It can easily be checked that l_i is a polynomial of degree at most n and that

$$l_i(x_j) \;= \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}.$$

Hence, the polynomial L_nf defined in(1.3) is also a polynomial of degree at most n and it satisfies conditions (1.1).

To prove uniqueness, assume there exists another polynomial P_n^* (of degree at most n) satisfying conditions (1.1) and consider

$$Q_n \;=\; L_n - P_n^*.$$

By (1.1), $Q_n(x_i) = 0, i = 0, \ldots, n$, which means Q_n, a polynomial of degree at most n, has $n+1$ distinct roots. By the Fundamental Theorem of Algebra, Q_n must be identically zero, thus proving the uniqueness of L_n.

□

So, given $n+1$ distinct points, we can find a unique polynomial of degree *at most* n, interpolating the data. It is possible that the degree of the interpolation polynomial to be actually *less* than n.

Example 1.4. Find the polynomial of minimum degree that interpolates the data

x	-2	-1	0	1	2	3
y	-5	1	1	1	7	25

.

Solution. Given 6 points, we find, by (1.3)-(1.4) (after simplifying), the polynomial

$$L_5f(x) \;=\; \sum_{i=0}^{5} l_i(x)y_i \;=\; x^3 - x + 1.$$

which, actually, has degree $3 < 5$.

■

Error and convergence

First of all, for any set of distinct nodes, the interpolation problem is *well-posed*, meaning that it has a unique solution that depends continuously on the data. Moreover, it can be expressed in terms of basis polynomials in the form (1.3).

We want to use the approximation

$$f(x) \approx L_n f(x),\ x \in [a,b].$$

To this end, we must assess (bound) the error (the remainder)

$$(R_n f)(x) = f(x) - (L_n f)(x),\ x \in [a,b]. \tag{1.5}$$

Theorem 1.5. *Let* $[a,b] \subset \mathbb{R}$*,* $f : [a,b] \to \mathbb{R}$ *a function of class* $C^{n+1}[a,b]$ *and consider the distinct nodes* $\{x_0, x_1, \ldots, x_n\} \subset [a,b]$*. Then there exists* $\xi \in (a,b)$ *such that*

$$(R_n f)(x) = \frac{u(x)}{(n+1)!} f^{(n+1)}(\xi), \tag{1.6}$$

where $u(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$*.*

Example 1.6. For linear and quadratic interpolation, the remainders are given by

$$\begin{aligned}(R_1 f)(x) &= \frac{(x-x_0)(x-x_1)}{2} f''(\xi),\\ (R_2 f)(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f'''(\xi).\end{aligned}$$

For $f(x) = \sqrt{x} = x^{1/2}$, the derivatives are

$$\begin{aligned}f'(x) &= \frac{1}{2}x^{-1/2} = \frac{1}{2}\cdot\frac{1}{\sqrt{x}},\\ f''(x) &= \frac{1}{2}\left(-\frac{1}{2}\right)x^{-3/2} = -\frac{1}{4}\cdot\frac{1}{x\sqrt{x}},\\ f'''(x) &= -\frac{1}{4}\left(-\frac{3}{2}\right)x^{-5/2} = \frac{3}{8}\cdot\frac{1}{x^2\sqrt{x}}.\end{aligned}$$

So, for the remainders, we have

$$\begin{aligned} |(R_1 f)(x)| &= \frac{|(x-x_0)(x-x_1)|}{8} \cdot \frac{1}{\xi\sqrt{\xi}}, \\ |(R_2 f)(x)| &= \frac{3}{8}\frac{|(x-x_0)(x-x_1)(x-x_2)|}{6} \cdot \frac{1}{\xi^2\sqrt{\xi}} = \frac{|(x-x_0)(x-x_1)(x-x_2)|}{16} \cdot \frac{1}{\xi^2\sqrt{\xi}}. \end{aligned}$$

Remark 1.7. In general, an upper bound of the interpolation error is given by

$$|(R_n f)(x)| \leq \frac{|u(x)|}{(n+1)!} M_{n+1}(f), \tag{1.7}$$

where

$$M_{n+1}(f) = \sup_{t\in[a,b]} |f^{(n+1)}(t)|.$$

Regarding the convergence of the Lagrange polynomial $L_n f$ to f, this *does not* happen, in general. The polynomial does converge, if, for instance, $f \in C^\infty[a,b]$, with $|f^{(k)}(x)| \leq M_k,\ \forall x \in [a,b], k = 0,1,2,\ldots$ and satisfies

$$\lim_{k\to\infty} \frac{(b-a)^k}{k!} M_k = 0.$$

In the early 1900's, it was proved (by Bernstein and Faber) that for each triangular array of nodes

$$\begin{array}{ccccc} x_0^{(0)} & & & & \\ x_0^{(1)} & x_1^{(1)} & & & \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & & \\ \vdots & \vdots & \vdots & \ddots & \\ x_0^{(n)} & x_1^{(n)} & x_2^{(n)} & \ldots & x_n^{(n)} \end{array}$$

in $[a,b]$ there exists a function $f \in C[a,b]$, such that the sequence of Lagrange interpolation polynomials

$$L_n f = L_n(f, x_0^{(n)}, x_1^{(n)}, \ldots, x_n^{(n)}; x)$$

does not converge uniformly to f on $[a,b]$. Moreover, they proved that for any array of nodes as above, there exists a function $f \in C[a,b]$, such that the corresponding sequence $\{L_n f\}_n$ is *divergent*.

Example 1.8 (Bernstein's Example). Let

$$f(x) \quad = \quad |x|, \; x \in [-1, 1]$$

and consider the equidistant nodes

$$x_k^{(n)} \; = \; -1 + \frac{2k}{n}, \; k = \overline{0, n}.$$

One can show that

$$\lim_{n \to \infty} |f(x) - L_n f(x)| \quad = \quad \infty,$$

for every $x \in [-1, 1]$, except $x = -1, x = 0$ and $x = 1$ (see Figure 3). Convergence in $x = \pm 1$ is trivial, since they are interpolation nodes (where the error is zero). The same is true for $x = 0$, when n is even.

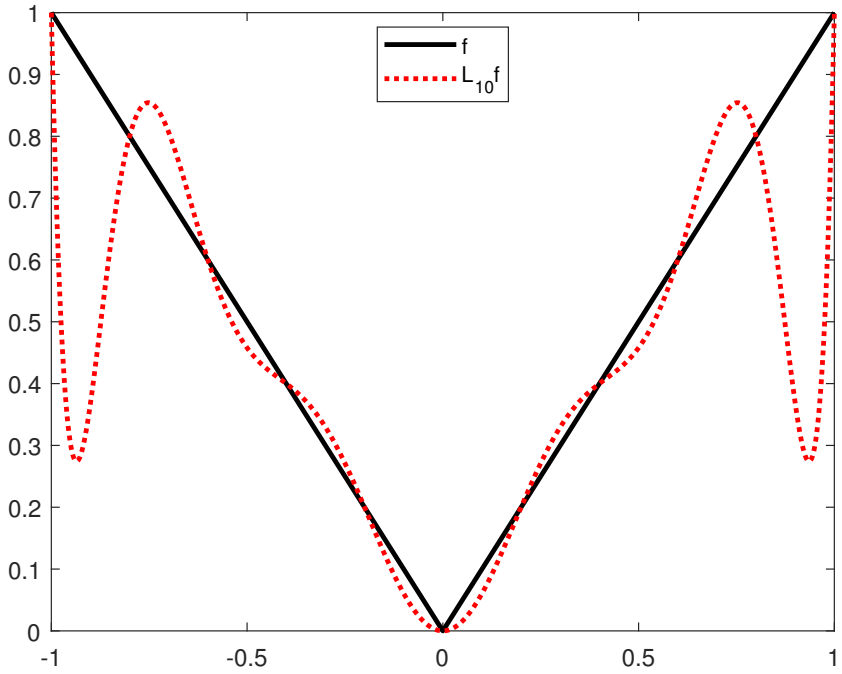

Fig. 3: Bernstein's Example, equidistant nodes, $n = 10$

Example 1.9 (**Runge's Example**). Consider the function

$$f(x) \;=\; \frac{1}{1+x^2},\; x \in [-5,5]$$

and the equally spaced nodes

$$x_k^{(n)} \;=\; -5 + 10\,\frac{k}{n},\; k = \overline{0,n}.$$

It can be shown that

$$\lim_{n\to\infty} |f(x) - L_n f(x)| \;=\; \begin{cases} 0, & \text{if} \quad |x| < 3.633\ldots \\ \infty, & \text{if} \quad |x| > 3.633\ldots \end{cases}$$

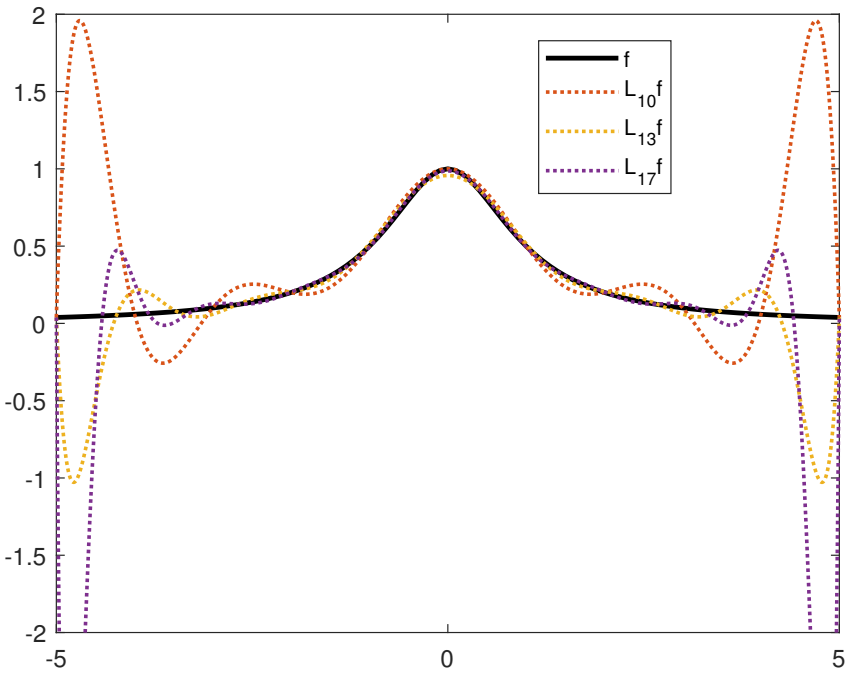

Fig. 4: Runge's Example, $n = 10, 13, 17$

Optimal choice of nodes

It may seem that choosing equally spaced nodes is beneficial, because it makes computations easier. However, as the last two examples showed, that is not the case. In fact, polynomial interpolation in equally spaced points is highly *ill-conditioned*: small changes in the data may cause huge changes

in the interpolant. This is known as **Runge's phenomenon**: a problem of oscillation at the edges of an interval, that occurs when using polynomial interpolation with polynomials of high degree over a set of *equidistant* nodes.

For polynomial interpolation to be a well-conditioned process, unless n is rather small, one must dispense with equally spaced points. The alternative is to use point sets that are clustered at the endpoints of the interval. Such families of nodes will minimize the term $|u(x)|$ in the error (1.7).

The simplest examples of clustered point sets are the families of *Chebyshev points*, obtained by projecting equally spaced points on the unit circle down to the unit interval $[-1, 1]$.
Assume the interval is $[-1, 1]$. Then, for a general interval $[a, b]$, we use the linear change of variables

$$x = \frac{b-a}{2}t + \frac{b+a}{2},\ t \in [-1, 1],\ x \in [a, b].$$

Chebyshev points of the first kind

An optimal choice of nodes are the roots of the **Chebyshev polynomial of the first kind**:

$$T_n(x) \ = \ \cos\left(n \arccos x\right),\ x \in [-1, 1]. \tag{1.8}$$

With the change of variables $x = \cos t, t \in [0, \pi]$, we get

$$\begin{aligned} T_n(x) &= \cos(nt) = \frac{1}{2}\left(e^{int} + e^{-int}\right) \\ &= \frac{1}{2}\left[(\cos t + i \sin t)^n + (\cos t - i \sin t)^n\right] \\ &= \frac{1}{2}\left[\left(x + i\sqrt{1-x^2}\right)^n + \left(x - i\sqrt{1-x^2}\right)^n\right]. \end{aligned}$$

The odd powers of the radical will be canceled, resulting in a polynomial of degree n in x, with leading coefficient 2^{n-1}.

Chebyshev polynomials of the first kind have some remarkable properties:

1. Polynomials of degree $0, 1, 2$ and 3 are easily computable using trigonometric identities. They

are

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

2. Higher degree polynomials can be obtained from the recurrence relation

$$T_{k+1}(x) \;=\; 2xT_k(x) - T_{k-1}(x),\; k = 1, 2, \ldots.$$

For example, the next polynomial is

$$T_4(x) = 8x^4 - 8x^2 + 1.$$

3. $\{T_n(x)\}_{n\in\mathbb{N}}$ is a sequence of *orthogonal* polynomials on $(-1, 1)$ with respect to the weight function $w(x) = \dfrac{1}{\sqrt{1-x^2}}$, i.e.

$$\int_{-1}^{1} \frac{T_n(x)T_m(x)}{\sqrt{1-x^2}}\, dx \;=\; \begin{cases} 0, & n \neq m \\ \pi, & n = m = 0 \\ \dfrac{\pi}{2}, & n = m \neq 0 \end{cases}.$$

To minimize the term $|u(x)|$ in the error (1.7), on the interval $[-1, 1]$, we choose

$$u(x) = \widetilde{T}_{n+1}(x) = \frac{1}{2^n} T_{n+1}(x),$$

i.e. the nodes

$$x_k \;=\; \cos\left(\frac{2k+1}{2n+2}\pi\right),\; k = 0, \ldots, n, \tag{1.9}$$

the roots of the Chebyshev polynomial T_{n+1}. In this case, we have

$$||R_n f|| \;\leq\; \frac{1}{2^n(n+1)!}||f^{(n+1)}||.$$

For the general case, on the interval $[a, b]$, we take

$$u(x) = \widetilde{T}_{n+1}(x; a, b) = \frac{(b-a)^{n+1}}{2^{2n+1}} \cos\left((n+1)\arccos\frac{2x-a-b}{b-a}\right)$$

and for the remainder, we have

$$||R_n f|| \quad \leq \quad \frac{(b-a)^{n+1}}{2^{2n+1}(n+1)!}||f^{(n+1)}||.$$

Example 1.10. Let us revisit Bernstein's Example, i.e. the function

$$f(x) \quad = \quad |x|, \ x \in [-1, 1],$$

only with Chebyshev nodes, this time, given by (1.9). Figure 5 shows a much better behaviour of the Lagrange polynomial.

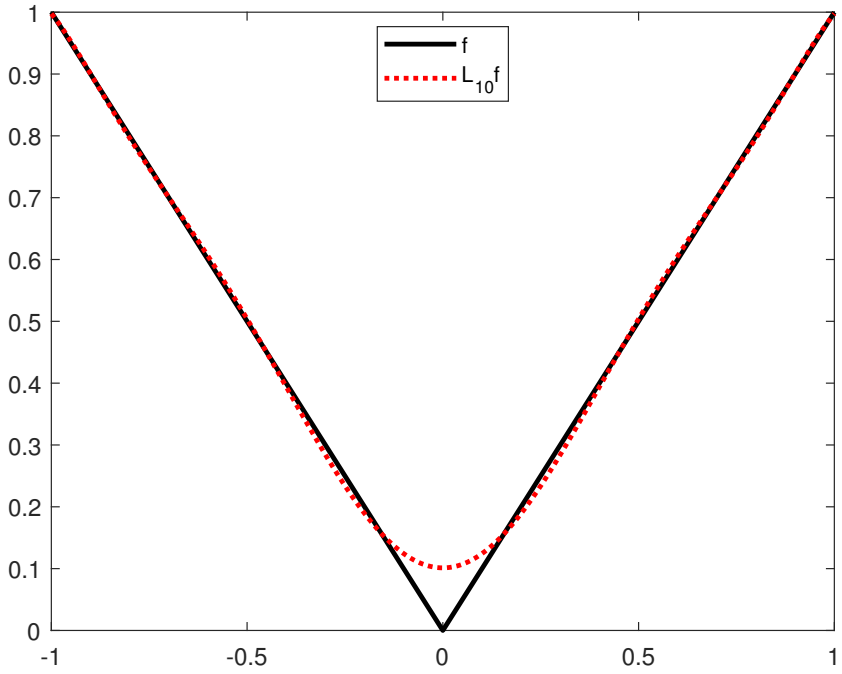

Fig. 5: Bernstein's Example, Chebyshev nodes, $n = 10$

Chebyshev points of the second kind

Chebyshev polynomials of the second kind are defined by

$$Q_n(x) \;=\; \frac{\sin\big((n+1)\arccos x\big)}{\sqrt{1-x^2}},\ x\in(-1,1). \tag{1.10}$$

These polynomials are orthogonal on $[-1,1]$ with respect to the weight function $w(x)=\sqrt{1-x^2}$:

$$\int_{-1}^{1}\sqrt{1-x^2}\,Q_n(x)Q_m(x)\,dx \;=\; \begin{cases} 0, & n\neq m \\ \dfrac{\pi}{2}, & n=m \end{cases}.$$

With the same change of variables as before, $x=\cos t, t\in[0,\pi]$, we find

$$Q_n(t) \;=\; \frac{\sin\big((n+1)t\big)}{\sin t},\ t\in[0,\pi].$$

The roots of Q_n are

$$x_k \;=\; \cos\left(\frac{k}{n+1}\pi\right),\ k=1,\ldots,n. \tag{1.11}$$

$Q_n(x)$ can be generated using the recurrence relations

$$\begin{aligned} Q_{k+1}(x) &= 2xQ_k(x)-Q_{k-1}(x),\ k=1,2,\ldots \\ Q_0(x) &= 1,\ \ Q_1(x)\;=\;2x. \end{aligned}$$

The first few Chebyshev polynomials of the second kind are

$$\begin{aligned} Q_0(x) &= 1, \\ Q_1(x) &= 2x, \\ Q_2(x) &= 4x^2-1, \\ Q_3(x) &= 8x^3-4x. \end{aligned}$$

Remark 1.11. The Chebyshev polynomials of the first and second kinds are closely related. It can

be easily checked that they satisfy a pair of mutual recurrence relations:

$$\begin{aligned} T_{n+1}(x) &= x\,T_n(x) - (1-x^2)\,Q_{n-1}(x), \\ Q_{n+1}(x) &= x\,Q_n(x) + T_{n+1}(x). \end{aligned}$$

Orthogonal polynomials are used extensively in many problems in Numerical Analysis, so we will revisit them (these and other families) later on.

1.2 Efficient Computation of Interpolation Polynomials

Recall the Lagrange interpolation polynomial of a function f, at the distinct nodes $(x_i, f_i), f_i = f(x_i),\ i = \overline{0,n}$:

$$L_n f(x) = \sum_{i=0}^{n} l_i(x) f_i, \tag{1.12}$$

where

$$\begin{aligned} l_i(x) &= \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j} = \frac{u_i(x)}{u_i(x_i)} = \frac{u_i(x)}{u'(x_i)}, \\ u(x) &= \prod_{j=0}^{n}(x-x_j), \quad u_j(x) = \frac{u(x)}{x-x_j},\ j=0,1,\ldots,n. \end{aligned}$$

This formula is well-suited for many theoretical uses of interpolation, but it is less desirable for practical computations. Among its shortcomings:
$-$ for each value of x, all of the basis functions l_i must be evaluated at x, which requires a product of n terms; thus, the total work is $O(n^2)$ flops (additions and multiplications) for every value of x;
$-$ adding a new node (x_{n+1}, f_{n+1}) requires a new computation *from scratch* and knowing $L_n f(x)$ does not lead to a less expensive way to evaluate $L_{n+1} f(x)$;
$-$ the computation is numerically unstable.
For these reasons, we need alternative and more easily computable formulations and expressions for interpolation polynomials.

1.2.1 Barycentric interpolation

The Lagrange formula (1.12) can be rewritten in such a way that it can be evaluated and updated in $O(n)$ flops.

$$L_n f(x) \;=\; \sum_{i=0}^{n} \frac{u_i(x)}{u'(x_i)} f_i \;= \sum_{i=0}^{n} \frac{u(x)}{(x-x_i)u'(x_i)} f_i \;=\; u(x) \sum_{i=0}^{n} \frac{\dfrac{1}{u'(x_i)}}{x-x_i} f_i.$$

Let

$$w_i \;=\; \frac{1}{u'(x_i)} \;=\; \frac{1}{\displaystyle\prod_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j)},\; i = 0, 1, \ldots, n. \tag{1.13}$$

These are called **barycentric weights**. With these, the Lagrange interpolation polynomial can be written as

$$L_n f(x) \;=\; u(x) \sum_{i=0}^{n} \frac{w_i}{x-x_i} f_i. \tag{1.14}$$

Formula (1.14) is called the **first barycentric formula** (also known as the *modified Lagrange interpolation formula*).

Now, Lagrange interpolation is a formula requiring $O(n^2)$ operations for calculating some quantities independent of x, the weights w_i, followed by $O(n)$ flops for evaluating $L_n f(x)$, once these numbers are known. Incorporating a new node x_{n+1} entails two calculations:
$-$ dividing each $w_i, i = 0, \ldots, n$ by $x_i - x_{n+1}$, for a cost of $n+1$ flops,
$-$ computing a new weight w_{n+1} using formula (1.13), for another $n+1$ flops.

Formula (1.14) can be improved even further. Notice that for the constant function $f \equiv 1$, the Lagrange polynomial is f itself

$$L_n f \;\equiv\; 1,$$

by the uniqueness of the interpolation polynomial. Substituting in (1.14), we find

$$1 \;=\; u(x) \sum_{i=0}^{n} \frac{w_i}{x-x_i}$$

and further

$$u(x) = \frac{1}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}}.$$

Then the Lagrange polynomial can be written as

$$L_n f(x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i} f_i}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}}, \tag{1.15}$$

called the **second barycentric formula** (or, simply, the *barycentric formula*).

Example 1.12. Consider our previous example, the function $f(x) = \sqrt{x}$ and the nodes $x_0 = 1$ and $x_1 = 4$.

Solution. We have

$$w_0 = \frac{1}{x_0 - x_1} = \frac{1}{1-4} = -\frac{1}{3}, \quad w_1 = \frac{1}{x_1 - x_0} = \frac{1}{4-1} = \frac{1}{3}$$

and

$$\begin{aligned} L_1 f(x) &= \frac{\dfrac{w_0}{x - x_0} f(x_0) + \dfrac{w_1}{x - x_1} f(x_1)}{\dfrac{w_0}{x - x_0} + \dfrac{w_1}{x - x_1}} = \frac{-\dfrac{1/3}{x-1} \cdot 1 + \dfrac{1/3}{x-4} \cdot 2}{-\dfrac{1/3}{x-1} + \dfrac{1/3}{x-4}} \\ &= \frac{-\dfrac{1}{x-1} + \dfrac{2}{x-4}}{-\dfrac{1}{x-1} + \dfrac{1}{x-4}} = \frac{-x + 4 + 2x - 2}{-x + 4 + x - 1} = \frac{1}{3}x + \frac{2}{3}, \end{aligned}$$

as before.

■

Remark 1.13. Even though formula (1.15) *hardly* looks like a polynomial, it *actually* is, as seen in the example above. However, there is one troublesome aspect: the interpolation polynomial should agree with the function value at the nodes, but, it is technically undefined when x equals one of the

nodes. In fact, it can easily be shown (using L'Hôpital's rule) that

$$\lim_{x\to x_k} \frac{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x-x_i} f_i}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x-x_i}} \;=\; f_k, \;\; k = 0, 1, \ldots, n,$$

so a continuous extension to the nodes is justified. This aspect is particularly important in the implementation of the barycentric formula.

We see that the barycentric formula is a Lagrange formula, but one with a special and beautiful symmetry. The weights w_i appear in the denominator exactly as in the numerator, except without the function values f_i. This means that *any common factor in all the weights w_i may be canceled without affecting the value of $L_n f$*, something that will be very useful next.

Computation of the barycentric weights

For some special sets of nodes x_j, one can give explicit formulas for the barycentric weights w_j, using the identity

$$w_j \;=\; \frac{1}{u'(x_j)}.$$

- The obvious place to start is *equidistant nodes* with spacing $h = 2/n$ on the interval $[-1, 1]$. In this case,

  $$w_j \;=\; (-1)^{n-j} \left(\begin{array}{c} n \\ j \end{array} \right) / (h^n n!)$$

  Since any common factor that does not depend on j will be canceled in the numerator and denominator of (1.15), this can be simplified to

  $$w_j \;=\; (-1)^j \left(\begin{array}{c} n \\ j \end{array} \right), \; j = 0, \ldots, n. \tag{1.16}$$

  For a general interval $[a, b]$ we would multiply this formula by $2^n (b-a)^{-n}$, but this constant factor too can be dropped, so we end up with (1.16) again, regardless of a and b.

- For *Chebyshev points of the first kind*, after canceling factors independent of j, we find

$$w_j \;=\; (-1)^j \sin \frac{(2j+1)\pi}{2n+2},\ j = 0, \ldots, n. \tag{1.17}$$

  If n is large, the weights w_j in (1.16) for equispaced barycentric interpolation vary by exponentially large factors, of order approximately 2^n. The effect will be that even small data near the center of the interval are associated with large oscillations in the interpolant, of the order of 2^n times bigger, near the edge of the interval, i.e. Runge's phenomenon.
  In contrast, the weights in (1.17) vary by factors $\mathcal{O}(n)$, not exponentially, reflecting the good distribution of the points and making polynomial interpolation a well-conditioned problem.

- If *Chebyshev points of the second kind* are used, then the weights are given by

$$w_j \;=\; \begin{cases} (-1)^j \dfrac{1}{2}, & \text{if } \ j = 0 \text{ or } \ j = n, \\ (-1)^j, & \text{otherwise.} \end{cases} \tag{1.18}$$