Part III

Functions

CHAPTER 9

Polynomial Interpolation

A fundamental mathematical technique is to approximate something complicated by something simple, or at least less complicated, in the hope that the simple can capture some of the essential information in the complicated. This is the core idea of approximation with Taylor polynomials, a tool that has been central to mathematics since the calculus was first discovered.

The wide-spread use of computers has made the idea of approximation even more important. Computers are basically good at doing very simple operations many times over. Effective use of computers therefore means that a problem must be broken up into (possibly very many) simple sub-problems. The result may provide only an approximation to the original problem, but this does not matter as long as the approximation is sufficiently good.

The idea of approximation is often useful when it comes to studying functions. Most mathematical functions only exist in quite abstract mathematical terms and cannot be expressed as combinations of the elementary functions we know from school. In spite of this, virtually all functions of practical interest can be approximated arbitrarily well by simple functions like polynomials, trigonometric or exponential functions. Polynomials in particular are very appealing for use on a computer since the value of a polynomial at a point can be computed by utilising simple operations like addition and multiplication that computers can perform extremely quickly.

A classical example is Taylor polynomials which is a central tool in calculus. A Taylor polynomial is a simple approximation to a function that is based on information about the function at a single point only. In practice, the degree of a Taylor polynomial is often low, perhaps only degree one (linear), but by increasing the degree the approximation can in many cases become arbitrarily good

over large intervals.

In this chapter we first give a review of Taylor polynomials. We assume that you are familiar with Taylor polynomials already or that you are learning about them in a parallel calculus course, so the presentation is brief, with few examples. We do however take the time to derive the Taylor polynomial as this illustrates the importance of writing polynomials in an appropriate form. We also include a description of Taylor polynomials for functions of two variables as this is needed for the analysis of numerical methods for solving differential equations in chapter 12.

The second topic in this chapter is a related procedure for approximating general functions by polynomials. The polynomial approximation will be constructed by forcing the polynomial to take the same values as the function at a few distinct points; this is usually referred to as *interpolation*. Although polynomial interpolation can be used for practical approximation of functions, we are mainly going to use it in later chapters for constructing various numerical algorithms for approximate differentiation and integration of functions, and numerical methods for solving differential equations.

An important additional insight that should be gained from this chapter is that the form in which we write a polynomial is important. We can simplify algebraic manipulations greatly by expressing polynomials in the right form, and the accuracy of numerical computations with a polynomial is also influenced by how the polynomial is represented.

9.1 Taylor polynomials

Taylor polynomials are discussed extensively in all calculus books, so the description here is brief. The essential feature of a Taylor polynomial is that it approximates a given function well at a single point.

Definition 9.1 (Taylor polynomial)**.** *Suppose that the first n derivatives of the function f exist at $x = a$. The Taylor polynomial of f of degree n at a is written $T_n(f;a)$ and satisfies the conditions*

$$T_n(f;a)^{(i)}(a) = f^{(i)}(a), \quad \text{for } i = 0, 1, \ldots, n. \tag{9.1}$$

The conditions in (9.1) mean that $T_n(f;a)$ and f have the same value and first n derivatives at a. This makes it quite easy to derive an explicit formula for the Taylor polynomial.

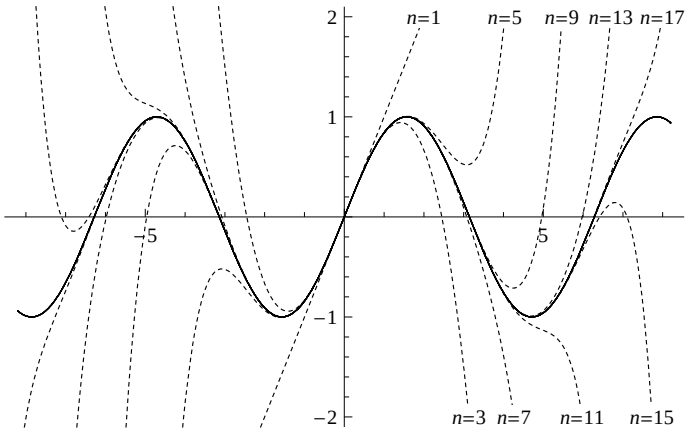


Figure 9.1. The Taylor polynomials of $\sin x$ (around $a = 0$) for degrees 1 to 17.

Theorem 9.2. *The Taylor polynomial of f of degree n at a is unique and can be written as*

$$T_n(f;a)(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''(a) + \cdots + \frac{(x-a)^n}{n!}f^{(n)}(a). \quad (9.2)$$

Figure 9.1 shows the Taylor polynomials of $\sin x$, generated about $a = 0$, for degrees up to 17. Note that the even degree terms for these Taylor polynomials are 0, so there are only 9 such Taylor polynomials. We observe that as the degree increases, the approximation improves on a larger interval.

Formula (9.2) is a classical result of calculus which is proved in most calculus books. We will include a proof here just to illustrate the usefulness of writing polynomials on nonstandard form.

9.1.1 Derivation of the Taylor formula

Suppose we are to prove (9.2) in the quadratic case. We are looking for a general quadratic polynomial

$$p_2(x) = c_0 + c_1 x + c_2 x^2 \quad (9.3)$$

that satisfies the conditions

$$p_2(a) = f(a), \quad p_2'(a) = f'(a), \quad p_2''(a) = f''(a). \quad (9.4)$$

The derivatives of p_2 are given by $p_2'(x) = c_1 + 2c_2x$ and $p_2''(x) = 2c_2$. If we insert these formulas and p_2 itself in the interpolation conditions (9.1), we obtain

$$\begin{aligned} p_2(a) &= c_0 + c_1 a + c_2 a^2 = f(a), \\ p_2'(a) &= c_1 + 2c_2 a = f'(a), \\ p_2''(a) &= 2c_2 = f''(a). \end{aligned} \tag{9.5}$$

This system is easy to solve if we start with the last equation. The result is

$$\begin{aligned} c_2 &= f''(a)/2, \\ c_1 &= f'(a) - 2c_2 a = f'(a) - f''(a)a, \\ c_0 &= f(a) - c_1 a - c_2 a^2 = f(a) - \big(f'(a) - f''(a)a\big)a - f''(a)a^2/2. \end{aligned}$$

If we insert these expression in (9.3) and collect the terms that multiply $f(a)$, $f'(a)$ and $f''(a)$, we obtain

$$\begin{aligned} p_2(x) &= f(a) - \big(f'(a) - f''(a)a\big)a - f''(a)a^2/2 + \big(f'(a) - f''(a)a\big)x + f''(a)x^2/2 \\ &= f(a) + (x-a)f'(a) + (x^2 - 2ax + a^2)f''(a)/2 \\ &= f(a) + (x-a)f'(a) + (x-a)^2 f''(a)/2, \end{aligned}$$

which agrees with (9.2).

The work involved in this derivation is not so much, but if we try the same strategy for general degree n, the algebra becomes quite involved. An alternative is to write the quadratic polynomial in the form

$$p_2(x) = b_0 + b_1(x-a) + b_2(x-a)^2 \tag{9.6}$$

straightaway. The derivatives can then be written as $p_2'(x) = b_1 + 2b_2(x-a)$ and $p_2''(x) = 2b_2$. If we insert the conditions (9.4) we now obtain

$$\begin{aligned} p_2(a) &= b_0 = f(a), \\ p_2'(a) &= b_1 = f'(a), \\ p_2''(a) &= 2b_2 = f''(a). \end{aligned} \tag{9.7}$$

From this it is easy to find the coefficients, and if we insert this in (9.6) we immediately obtain

$$p_2(x) = f(a) + (x-a)f'(a) + \frac{f''(a)}{2}(x-a)^2.$$

The main difference is that by writing the polynomial in the form (9.6), the system of equations (9.7) becomes extremely simple, and we can just read off the

coefficients. When we used the traditional form of the polynomial (9.3), we obtained a more complicated system that required some algebraic manipulations when the solution was inserted in (9.3).

For general degree we write the polynomial as

$$p_n(x) = b_0 + b_1(x-a) + \cdots b_i(x-a)^i + \cdots + b_n(x-a)^n. \tag{9.8}$$

The conditions to be satisfied by p_n are

$$p_n(a) = f(a), \quad p_n'(a) = f'(a), \quad ,\ldots, \quad p_n^{(i)}(a) = f^{(i)}(a), \quad \ldots, \quad p_n^{(n)}(a) = f^{(n)}(a).$$

The derivatives of p_n are

$$\begin{aligned}
p'(x) &= b_1 + 2b_2(x-a) + \cdots + ib_i(x-a)^{i-1} + \cdots + nb_n(x-a)^{n-1},\\
&\vdots\\
p^{(i)}(x) &= i!b_i + \cdots + n(n-1)\cdots(n-i+1)(x-a)^{n-i},\\
p^{(n)}(x) &= n!b_n.
\end{aligned}$$

If we insert this in the conditions to be satisfied by p_n we obtain

$$b_0 = f(a), \quad b_1 = f'(a), \quad \ldots, \quad b_i = f^{(i)}(a)/i!, \quad \ldots, \quad b_n = f^{(n)}(a)/n!,$$

with hardly any work, which confirms formula (9.2).

The polynomial form (9.8) simplifies the computations because of the factor $(x-a)$ which cancels almost all of $p_n(x)$ and its derivatives when $x = a$. This illustrates that there is much to be gained by choosing carefully how the polynomial is written.

Observation 9.3. *In the derivation of the Taylor polynomial, the manipulations simplify if polynomials of degree n are written as*

$$p_n(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + \cdots + c_n(x-a)^n.$$

9.1.2 The remainder

The Taylor polynomial $T_f(f)$ is an approximation to f, and in many situations it will be important to control the error in the approximation. The error can be expressed in a number of ways, and the following two are the most common.

Theorem 9.4. *Suppose that f is a function whose derivatives up to order $n+1$ exist and are continuous. Then the remainder in the Taylor expansion $R_n(f;a)(x) = f(x) - T_n(f;a)(x)$ is given by*

$$R_n(f;a)(x) = \frac{1}{n!}\int_a^x f^{(n+1)}(t)(x-t)^n\,dt. \tag{9.9}$$

The remainder may also be written as

$$R_n(f;a)(x) = \frac{(x-a)^{n+1}}{(n+1)!}f^{(n+1)}(\xi), \tag{9.10}$$

where ξ is a number in the interval (a,x) (the interval (x,a) if $x<a$).

The proof of this theorem is based on the fundamental theorem of calculus and integration by parts, and can be found in any standard calculus text.

We are going to make use of Taylor polynomials with remainder in future chapters to analyse the error in a number of numerical methods. Here we just consider one example of how we can use the remainder to control how well a function is approximated by a polynomial.

Example 9.5. We want to determine a polynomial approximation of the function $\sin x$ on the interval $[-1,1]$ with error smaller than 10^{-5}. We want to use Taylor polynomials about the point $a=0$; the question is how high the degree needs to be in order to get the error to be small.

If we look at the error term (9.10), there is one factor that looks rather difficult to control, namely $f^{(n+1)}(\xi)$: Since we do not know the degree, we do not really know what this derivative is, and on top of this we do not know at which point it should be evaluated either. The solution is not so difficult if we realise that we do not need to control the error exactly, it is sufficient to make sure that the error is *smaller* than 10^{-5}.

The easiest error term to work with is (9.10), and what we want is to find the smallest n such that

$$\left|\frac{x^{n+1}}{(n+1)!}f^{(n+1)}(\xi)\right| \leq 10^{-5}, \tag{9.11}$$

where the function $f(x) = \sin x$ in our case and ξ is a number in the interval $(0,x)$. Here we demand that the absolute value of the error should be smaller than 10^{-5}. This is important since otherwise we could make the error small by making it negative, with large absolute value. The main ingredient in achieving what we want is the observation that since $f(x) = \sin x$, any derivative of f is either $\cos x$ or $\sin x$ (possibly with a minus sign which disappears when we take

absolute values). But then we certainly know that

$$\left| f^{(n+1)}(\xi) \right| \leq 1. \tag{9.12}$$

This may seem like a rather crude estimate, which may be the case, but it was certainly very easy to derive; to estimate the correct value of ξ would certainly be much more difficult. If we insert the estimate (9.12) on the left in (9.11), we can also change our required inequality,

$$\left| \frac{x^{n+1}}{(n+1)!} f^{(n+1)}(\xi) \right| \leq \frac{|x|^{n+1}}{(n+1)!} \leq 10^{-5}.$$

If we manage to find an n such that this last inequality is satisfied, then (9.11) will also be satisfied. Since $x \in [-1,1]$ we know that $|x| \leq 1$ so this last inequality will be satisfied if

$$\frac{1}{(n+1)!} \leq 10^{-5}. \tag{9.13}$$

The left-hand side of this inequality decreases with increasing n, so we can just determine n by computing $1/(n+1)!$ for the first few values of n, and use the first value of n for which the inequality holds. If we do this, we find that $1/8! \approx 2.5 \times 10^{-5}$ and $1/9! \approx 2.8 \times 10^{-6}$. This means that the smallest value of n for which (9.13) will be satisfied is $n = 8$. The Taylor polynomial we are looking for is therefore

$$p_8(x) = T_8(\sin; 0)(x) = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040},$$

since the term of degree 8 is zero. If we evaluate this we find $p_8(1) \approx 0.841468$ with error roughly 2.73×10^{-6} which is close to the estimate $1/9!$ which we computed above.

Figure 9.1 shows the Taylor polynomials of $\sin x$ about $a = 0$ of degree up to 17. In particular we see that for degree 7, the approximation is indistinguishable from the original in the plot, at least up to $x = 2$. ∎

The error formula (9.10) will be the most useful one for us, and for easy reference we record the complete Taylor expansion in a corollary.

Corollary 9.6. *Any function f whose first $n+1$ derivatives are continuous at $x = a$ can be expanded in a Taylor polynomial of degree n at $x = a$ with a corresponding error term,*

$$f(x) = f(a) + (x-a)f'(a) + \cdots + \frac{(x-a)^n}{n!} f^{(n)}(a) + \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi_x), \tag{9.14}$$

where ξ_x is a number in the interval (a,x) (the interval (x,a) if $x < a$) that depends on x. This is called a Taylor expansion *of f.*

The remainder term in (9.14) lets us control the error in the Taylor approximation. It turns out that the error behaves quite differently for different functions.

Example 9.7 (Taylor polynomials for $f(x) = \sin x$)**.** If we go back to figure 9.1, it seems like the Taylor polynomials approximate $\sin x$ well on larger intervals as we increase the degree. Let us see if this observation can be derived from the error term

$$e(x) = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi). \tag{9.15}$$

When $f(x) = \sin x$ we know that $|f^{(n+1)}(\xi)| \leq 1$, so the error is bounded by

$$|e(x)| \leq \frac{|x|^{n+1}}{(n+1)!}$$

where we have also inserted $a = 0$ which was used in figure 9.1. Suppose we want the error to be small on the interval $[-b, b]$. Then $|x| \leq b$, so on this interval the error is bounded by

$$|e(x)| \leq \frac{b^{n+1}}{(n+1)!}.$$

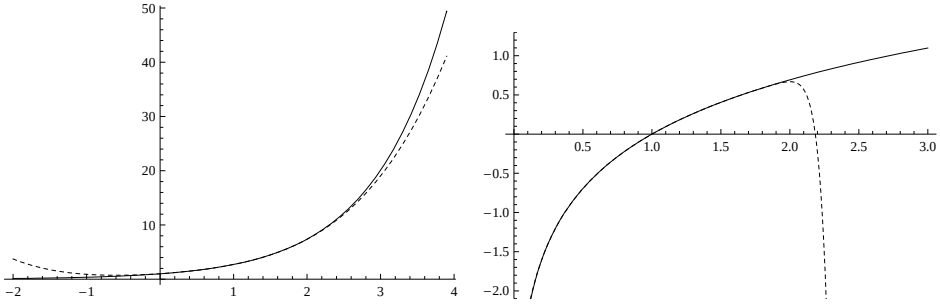
The question is what happens to the expression on the right when n becomes large; does it tend to 0 or does it not? It is not difficult to show that regardless of what the value of b is, the factorial $(n+1)!$ will tend to infinity more quickly, so

$$\lim_{n\to\infty} \frac{b^{n+1}}{(n+1)!} = 0$$

In other words, if we just choose the degree n to be high enough, we can get the Taylor polynomial to be an arbitrarily good approximation to $\sin x$ on an interval $[-b, b]$, regardless of the value of b. ∎

Example 9.8 (Taylor polynomials for $f(x) = e^x$)**.** Figure 9.2 (a) shows a plot of the Taylor polynomial of degree 4 for the exponential function $f(x) = e^x$, expanded about the point $a = 1$. For this function it is easy to see that the Taylor polynomials will converge to e^x on any interval as the degree tends to infinity, just like we saw for $f(x) = \sin x$ in example 9.7. ∎

Example 9.9 (Taylor polynomials for $f(x) = \ln x$)**.** The plot in figure 9.2 shows the function $f(x) = \ln x$ and its Taylor polynomial of degree 20 expanded at $a = 1$. The Taylor polynomial seems to be very close to $\ln x$ as long as x is a bit smaller than 2, but for $x > 2$ it seems to diverge quickly. Let us see if this can be deduced from the error term.



(a) (b)

Figure 9.2. In (a) the Taylor polynomial of degree 4 about the point $a = 1$ for the function $f(x) = e^x$ is shown. Figure (b) shows the Taylor polynomial of degree 20 for the function $f(x) = \log x$, also about the point $a = 1$.

The error term involves the derivative $f^{(n+1)}(\xi)$ of $f(x) = \ln x$, so we need a formula for this. Since $f(x) = \ln x$, we have

$$f'(x) = \frac{1}{x} = x^{-1}, \quad f''(x) = -x^{-2}, \quad f'''(x) = 2x^{-3}$$

and from this we find that the general formula is

$$f^{(k)}(x) = (-1)^{k+1}(k-1)!\,x^{-k}. \tag{9.16}$$

From this we find that the general term in the Taylor polynomial is

$$\frac{(x-1)^k}{k!} f^{(k)}(1) = (-1)^{k+1}\frac{(x-1)^k}{k}$$

so the Taylor expansion (9.14) becomes

$$\ln x = \sum_{k=1}^{n} (-1)^{k+1}\frac{(x-1)^k}{k} + \frac{(x-1)^{n+1}}{n+1}\xi^{-n-1},$$

where ξ is some number in the interval $(1, x)$ (in $(x, 1)$ if $0 < x < 1$). The problematic area seems to be to the right of $x = 1$, so let us assume that $x > 1$. In this case $\xi > 1$ so $\xi^{-n-1} < 1$ so the error is bounded by

$$\left|\frac{(x-1)^{n+1}}{n+1}\xi^{-n-1}\right| \leq \frac{(x-1)^{n+1}}{n+1}.$$

When $x - 1 < 1$, i.e., when $x < 2$, we know that $(x-1)^{n+1}$ will tend to zero when n tends to infinity, and the denominator $n+1$ will just contribute to this happening even more quickly.

For $x > 2$, one can try and analyse the error term, and if one uses the integral form of the remainder (9.9) it is in fact possible to find an exact formula for the error. However, it is much simpler to consider the Taylor polynomial directly,

$$p_n(x) = T(\ln; 1)(x) = \sum_{k=1}^{n} (-1)^{k+1} \frac{(x-1)^k}{k}.$$

Note that for $x > 2$, the absolute value of the terms in the sum will become arbitrarily large since

$$\lim_{k\to\infty} \frac{c^k}{k} = \infty$$

when $c > 0$. This means that the sum will jump around more and more, so there is no way it can converge for $x > 2$, and it is this effect we see in figure 9.2 (b). ∎

9.2 Interpolation

A Taylor polynomial based at a point $x = a$ usually provides a very good approximation near a, but as we move away from this point, the error will increase. If we want a good approximation to a function f across a whole interval, it seems natural that we ought to utilise information about f from different parts of the interval. Polynomial interpolation lets us do just that.

9.2.1 Polynomial interpolation is possible

The idea behind polynomial interpolation is simple: We approximate a function f by a polynomial p by forcing p to have the same function values as f at a number of points. Suppose for instance that we have $n+1$ distinct points $(x_i)_{i=0}^n$ scattered throughout the interval $[a, b]$ where f is defined. Since a polynomial of degree n has $n+1$ free coefficients it is natural to try and find a polynomial of degree n with the same values as f at these points.

Problem 9.10 (Polynomial interpolation)**.** *Let f be a given function defined on an interval $[a, b]$, and let $(x_i)_{i=0}^n$ be $n+1$ distinct points in $[a, b]$. The polynomial interpolation problem is to find a polynomial $p_n = P(f; x_0, \ldots, x_n)$ of degree n that matches f at the given points,*

$$p_n(x_i) = f(x_i), \quad \text{for } i = 0, 1, \ldots, n. \tag{9.17}$$

The points $(x_i)_{i=0}^n$ are called interpolation points*, the conditions* (9.17) *are called the* interpolation conditions*, and the polynomial $p_n = P(f; x_0, \ldots, x_n)$ is called a polynomial* interpolant*.*

The notation $P(f; x_0, \ldots, x_n)$ for the polynomial interpolant is similar to the notation $T_n(f; a)$ for the Taylor polynomial. However, it is a bit cumbersome, so we will often use p_n when no confusion is possible.

There are at least three questions raised by problem 9.10: Is there a polynomial of degree n that satisfies the interpolation conditions (9.17)? How many such polynomials are there? How can we find one such polynomial? As always it is instructive to consider the simplest cases to gain some insight into the problem.

Example 9.11 ($n = 0$)**.** The case $n = 0$ corresponds to the situation where we only have one interpolation point x_0. Then we are looking for a polynomial p_0 of degree 0, i.e. a constant, such that $p_0(x_i) = f(x_i)$. This is easy to solve; we see that we must have $p_0(x) = f(x_0)$. ∎

Example 9.12 ($n = 1$)**.** The next case is $n = 1$, i.e., we have two interpolation points x_0 and x_1. Now we are looking for a straight line $p_1(x) = c_0 + c_1 x$ such that $p_1(x_0) = f(x_0)$ and $p_1(x_1) = f(x_1)$. From school we know that this is possible, and one formula for the solution is

$$p_1(x) = \frac{x_1 - x}{x_1 - x_0} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1). \quad \blacksquare \tag{9.18}$$

Example 9.13 ($n = 2$)**.** In this case we are given 3 points x_0, x_1 and x_2 and we are looking for a quadratic polynomial p_2 that may take the same values as f at these points. The representation in (9.18) is appealing since the dependence on the two function values is so explicit, so let us try and generalise this. In other words, we want to write p_2 in the form

$$p_2(x) = \ell_0(x) f(x_0) + \ell_1(x) f(x_1) + \ell_2(x) f(x_2) \tag{9.19}$$

where $\{\ell_i\}_{i=0}^2$ are quadratic polynomials. If we insert the two interpolation conditions at $x = x_1$ and $x = x_2$ we obtain the equations

$$\begin{aligned} f(x_1) = p_2(x_1) &= \ell_0(x_1) f(x_0) + \ell_1(x_1) f(x_1) + \ell_2(x_1) f(x_2), \\ f(x_2) = p_2(x_2) &= \ell_0(x_2) f(x_0) + \ell_1(x_2) f(x_1) + \ell_2(x_2) f(x_2). \end{aligned}$$

From this we see that ℓ_0 should satisfy the equations $\ell_0(x_1) = \ell_0(x_2) = 0$ to make sure that $f(x_0)$ is cancelled out at x_1 and x_2. For this to happen, the quadratic polynomial p_2 should have two roots at x_1 and x_2, i.e., it should be on the form $\ell_0(x) = d_0(x - x_1)(x - x_2)$. The value of d_0 we can find from the interpolation condition at x_0,

$$f(x_0) = p_2(x_0) = d_0 \ell_0(x_0) f(x_0) = d_0(x_0 - x_1)(x_0 - x_2) f(x_0), \tag{9.20}$$

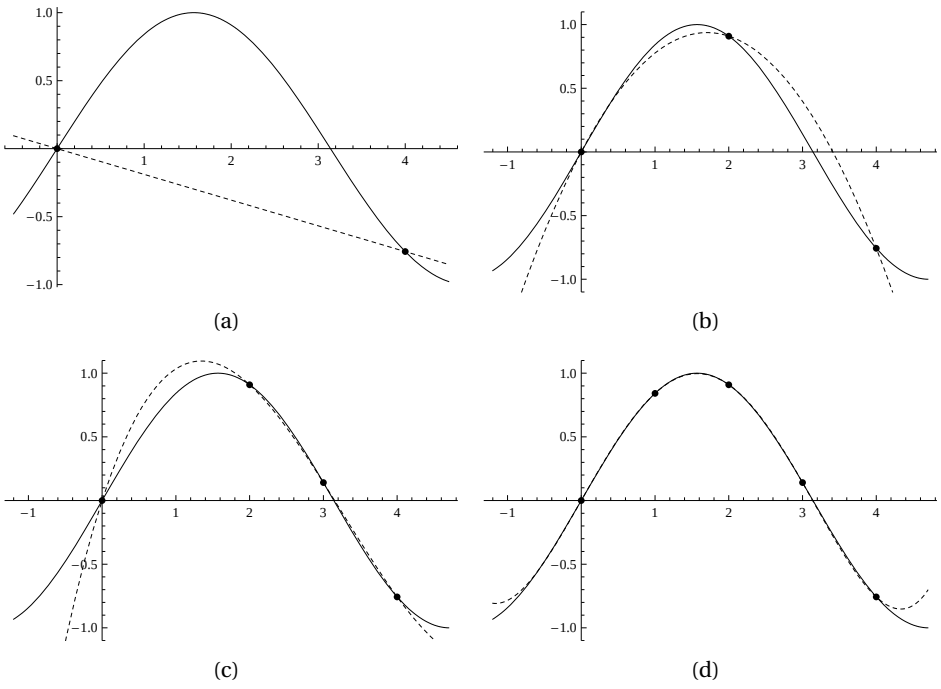


Figure 9.3. Interpolation of $\sin x$ with a line (a), a parabola (b), a cubic (c), and a quartic polynomial (d).

where we have used the fact that we must also have $\ell_1(x_0) = \ell_2(x_0) = 0$. From (9.20) we see that $d_0 = 1/\big((x_0 - x_1)(x_0 - x_2)\big)$ and therefore

$$\ell_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}.$$

We can construct ℓ_1 and ℓ_2 in the same way and find

$$\ell_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \quad \ell_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

If we insert these polynomials in (9.19) it is easy to verify that the resulting polynomial p_2 indeed satisfies the 3 interpolation conditions. ∎

Some examples of interpolation are shown in figure 9.3. Note how the quality of the approximation improves with increasing degree.

The crucial property possessed by the quadratic polynomials $\{\ell_i(x)\}_{i=0}^2$ at the end of example 9.13 is that they satisfy the relations

$$\ell_i(x_j) = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases} \tag{9.21}$$

for any pair of integers in the range $0 \le i, j \le 2$. The two polynomials in the linear case (9.18) satisfy the same kind of relation. This kind of relation is so common that the right-hand side of (9.21) has been given a special name.

Definition 9.14 (Kronecker delta)**.** *The* Kronecker delta $\delta_{i,j}$ *is defined as*

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases}$$

for any pair of integers (i, j)*.*

The general idea for showing that the interpolation problem is solvable follows exactly as in the quadratic case. For degree n we have to construct $n+1$ polynomials $\{\ell_{i,n}\}_{i=0}^{n}$ of degree n such that the interpolant p_n can be written

$$p_n(x) = \ell_{0,n}(x) f(x_0) + \ell_{1,n}(x) f(x_1) + \cdots + \ell_{n,n}(x) f(x_n)$$

where we have added a second subscript to the ℓ_i-polynomials to indicate the degree. This will work if the polynomial $\ell_{i,n}(x)$ is one at x_i and zero at all the other x_is, i.e., if $\ell_{i,n}(x_j) = \delta_{i,j}$. This is accomplished if

$$\ell_{i,n}(x) = \frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}. \tag{9.22}$$

We have therefore proved most of the following basic theorem on interpolation.

Theorem 9.15. *Let* f *be a given function defined on an interval* $[a, b]$*, and let* $(x_i)_{i=0}^{n}$ *be* $n+1$ *distinct points in* $[a, b]$*. Then the polynomial defined by*

$$p_n(x) = \ell_{0,n}(x) f(x_0) + \ell_{1,n}(x) f(x_1) + \cdots + \ell_{n,n}(x) f(x_n)$$

is the unique polynomial of degree n *that satisfies the interpolation conditions*

$$p_n(x_i) = f(x_i), \quad \text{for } i = 0, 1, \ldots, n.$$

Here $\{\ell_{i,n}\}_{i=0}^{n}$ *are the polynomials of degree* n *defined by* (9.22) *which are usually referred to as the* Lagrange polynomials *of degree* n*.*

Proof. The fact that p_n satisfies the interpolation conditions follows from the argument preceding the theorem, just like in the quadratic case. What remains

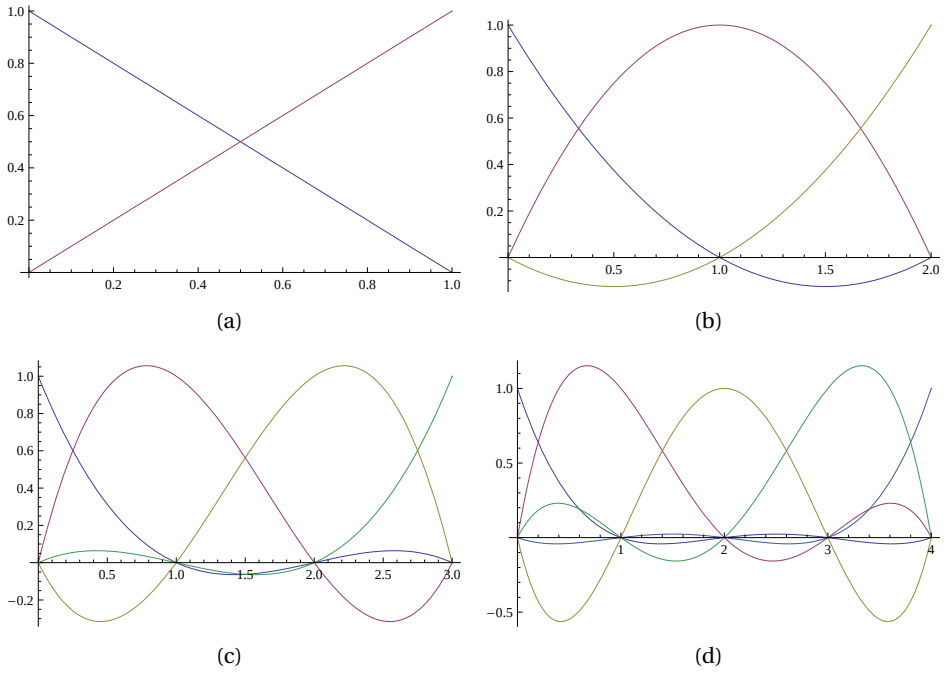


Figure 9.4. The Lagrange polynomials of degree 1 (in (a)), 2 (in (b)), 3 (in (c)), and 4 (in (d)). For degree n the interpolation points are $x_i = i$, for $i = 0, \ldots, n$.

is to prove is that this is the only polynomial that satisfies the interpolation conditions. For this, suppose that $\hat{p}_n$ is another polynomial that satisfies the interpolation conditions. Then the difference $e = p_n - \hat{p}_n$ is a polynomial of degree n that satisfies the conditions

$$e(x_i) = p_n(x_i) - \hat{p}_n(x_i) = f(x_i) - f(x_i) = 0, \quad \text{for } i = 0, 1, \ldots, n.$$

In other words, the polynomial e is of degree n and has $n+1$ roots. The only polynomial of degree n that can have this many zeros is the polynomial which is identically zero. But if $e = p_n - \hat{p}_n$ is identically zero, then $\hat{p}_n$ must be identically equal to p_n, so there is only one polynomial of degree n that solves the interpolation problem. ∎

Some examples of Lagrange polynomials of different degrees are shown in figure 9.4.

Theorem 9.15 answers two of the questions raised above: Problem 9.10 has a solution and it is unique. The theorem itself does not tell us directly how to find the solution, but in the text preceding the theorem we showed how it could be constructed. One concrete example will illustrate the procedure.

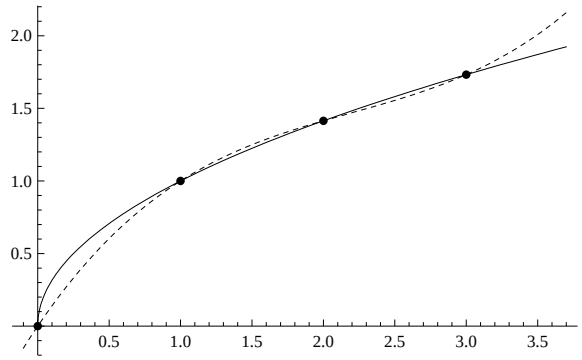


Figure 9.5. The function $f(x) = \sqrt{x}$ (solid) and its cubic interpolant at the four points 0, 1, 2, and 3 (dashed).

Example 9.16. Suppose we have the four points $x_i = i$, for $i = 0, \ldots, 3$, and we want to interpolate the function $\sqrt{x}$ at these points. We first construct the cubic Lagrange polynomials for the points $(x_i)_{i=0}^3$. For the first one, ℓ_0, we find (we drop the second subscript),

$$\ell_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = -\frac{(x-1)(x-2)(x-3)}{6}.$$

The other three Lagrange polynomials can be constructed in the same way,

$$\ell_1(x) = \frac{x(x-2)(x-3)}{2}, \quad \ell_2(x) = -\frac{x(x-1)(x-3)}{2}, \quad \ell_3(x) = \frac{x(x-1)(x-2)}{6}.$$

When the Lagrange polynomials are known the interpolant is simply

$$p_3(x) = \ell_1(x) + \ell_2(x)\sqrt{2} + \ell_3(x)\sqrt{3}$$

(the first term disappears since $\sqrt{0} = 0$). Figure 9.5 shows a plot of this interpolant. ∎

9.2.2 The Newton form

Representation of polynomial interpolants provides another example of the fact that it may be worthwhile to write polynomials in alternative forms. The representation in terms of Lagrange polynomials makes it simple to write down the interpolant since no equations need to be solved.

Sometimes, it is desirable to improve the interpolating polynomial by adding one more interpolation point. If the Lagrange form is used, it is far form obvious how the earlier constructed interpolant can be exploited when computing the new one which uses one extra point. For this, the Newton form of the interpolating polynomial is more convenient.

Definition 9.17. *Let $(x_i)_{i=0}^n$ be n distinct real numbers. The Newton form of the interpolating polynomial of degree n is written on the form*

$$p_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}). \quad (9.23)$$

Note that there is no mention of the function f to be interpolated in definition 9.17. This information is contained in the coefficients $(c_i)_{i=0}^n$ which are not specified in the definition, but which must be computed. As usual, some examples are instructive.

Example 9.18 (Newton form for $n = 0$)**.** Suppose we have only one interpolation point x_0. Then the Newton form is just $p_0(x) = c_0$. To interpolate f at x_0 we just have to choose $c_0 = f(x_0)$,

$$p_0(x) = f(x_0). \quad \blacksquare$$

Example 9.19 (Newton form for $n = 1$)**.** With two points x_0 and x_1 the Newton form is $p_1(x) = c_0 + c_1(x - x_0)$. Interpolation at x_0 means that $f(x_0) = p_1(x_0) = c_0$ and interpolation at x_1,

$$f(x_1) = p_1(x_1) = f(x_0) + c_1(x_1 - x_0),$$

which means that

$$c_0 = f(x_0), \quad c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \quad (9.24)$$

Therefore c_0 remains the same as in the case $n = 0$. $\blacksquare$

Example 9.20 (Newton form for $n = 2$)**.** We add another point and consider interpolation with a quadratic polynomial

$$p_2(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1).$$

at the three points x_0, x_1, x_2. Interpolation at x_0 and x_1 gives the equations

$$f(x_0) = p_2(x_0) = c_0,$$
$$f(x_1) = p_2(x_1) = c_0 + c_1(x_1 - x_0),$$

which we note are the same equations as we solved in the case $n = 1$. From the third condition

$$f(x_2) = p(x_2) = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1),$$

we obtain

$$c_2 = \frac{f(x_2) - f(x_0) - \frac{f(x_1)-f(x_0)}{x_1-x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}.$$

Playing around a bit with this expression one finds that it can also be written as

$$c_2 = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0}. \quad \blacksquare \tag{9.25}$$

It is easy to see that what happened in the quadratic case, happens in the general case: The equation that results from the interpolation condition at x_k involves only the points $\big(x_0, f(x_0)\big)$, $\big(x_1, f(x_1)\big)$, …, $\big(x_k, f(x_k)\big)$. If we write down all the equations we find

$$\begin{aligned}
f(x_0) &= c_0, \\
f(x_1) &= c_0 + c_1(x_1 - x_0), \\
f(x_2) &= c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1), \\
&\vdots \\
f(x_k) &= c_0 + c_1(x_k - x_0) + c_2(x_k - x_0)(x_k - x_1) + \cdots \\
&\qquad + c_{k-1}(x_k - x_0)\cdots(x_k - x_{k-2}) + c_k(x_k - x_0)\cdots(x_k - x_{k-1}).
\end{aligned} \tag{9.26}$$

This is an example of a *triangular system* where each new equation introduces one new variable and one new point. This means that each coefficient c_k only depends on the data $\big(x_0, f(x_0)\big)$, $\big(x_1, f(x_1)\big)$, …, $\big(x_k, f(x_k)\big)$, and the following lemma is immediate.

Lemma 9.21. *Let f be a given function and x_0, …, x_n given interpolation points. If the interpolating polynomial of degree n is written in Newton form,*

$$p_n(x) = c_0 + c_1(x - x_0) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}), \tag{9.27}$$

then c_k is only dependent on $\big(x_0, f(x_0)\big)$, $\big(x_1, f(x_1)\big)$, …, $\big(x_k, f(x_k)\big)$ and is written

$$c_k = f[x_0, \ldots, x_k] \tag{9.28}$$

for $k = 0, 1, \ldots, n$. The interpolating polynomials p_n and p_{n-1} are related by

$$p_n(x) = p_{n-1}(x) + f[x_0, \ldots, x_n](x - x_0)\cdots(x - x_{n-1}).$$

Note that with the notation (9.28) for the coefficients, the interpolation formula (9.27) becomes

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \ldots, x_n](x - x_0) \cdots (x - x_{n-1}). \quad (9.29)$$

The coefficients (c_k) can be computed from the system (9.26). The big advantage compared with the Lagrange form is the simplicity of adding an extra point: An extra point adds an extra equation to (9.26) from which the additional coefficient can be computed, while the remaining coefficients are the same as in the case with one point less.

9.2.3 Divided differences

The coefficients $c_k = f[x_0, \ldots, x_k]$ have certain properties that are useful both for computation and understanding. When doing interpolation at the points $x_0, \ldots, x_k$ we can consider two smaller problems, namely interpolation at the points $x_0, \ldots, x_{k-1}$ as well as interpolation at the points $x_1, \ldots, x_k$.

Suppose that the polynomial q_0 interpolates f at the points $x_0, \ldots, x_{k-1}$ and that q_1 interpolates f at the points $x_1, \ldots, x_k$, and consider the polynomial defined by the formula

$$p(x) = \frac{x_k - x}{x_k - x_0} q_0(x) + \frac{x - x_0}{x_k - x_0} q_1(x). \quad (9.30)$$

Our claim is that $p(x)$ interpolates f at the points $x_0, \ldots, x_k$, which means that $p = p_k$ since a polynomial interpolant of degree $k-1$ which interpolates $k+1$ points is unique.

We fist check that p interpolates f at an interior point x_i with $0 < i < k$. In this case $q_0(x_i) = q_1(x_i) = f(x_i)$ so

$$p(x_i) = \frac{x_k - x}{x_k - x_0} f(x_i) + \frac{x - x_0}{x_k - x_0} f(x_i) = f(x_i).$$

At $x = x_0$ we have

$$p(x_0) = \frac{x_k - x_0}{x_k - x_0} q_0(x_0) + \frac{x_0 - x_0}{x_k - x_0} q_1(x_0) = q_0(x_0) = f(x_0),$$

as required, and in a similar way we also find that $p(x_k) = f(x_k)$.

Let us rewrite (9.30) in a more explicit way.

Lemma 9.22. *Let $P(f; x_0, \ldots, x_k)$ denote the polynomial of degree $k-1$ that interpolates the function f at the points $x_0, \ldots, x_k$. Then*

$$P(f; x_0, \ldots, x_k)(x) = \frac{x_k - x}{x_k - x_0} P(f; x_0, \ldots, x_{k-1})(x) + \frac{x - x_0}{x_k - x_0} P(f; x_1, \ldots, x_k)(x).$$

From this lemma we can deduce a useful formula for the coefficients of the interpolating polynomial. (Recall that the leading coefficient of a polynomial is the coefficient of the highest degree term.)

Theorem 9.23. *Let $c_k = f[x_0,\ldots,x_k]$ denote the leading coefficient of the interpolating polynomial $P(f;x_0,\ldots,x_k)$. This is called a* kth order divided difference *of f and satisfies the relations $f[x_0] = f(x_0)$, and*

$$f[x_0,\ldots,x_k] = \frac{f[x_1,\ldots,x_k] - f[x_0,\ldots,x_{k-1}]}{x_k - x_0} \tag{9.31}$$

for $k > 0$.

Proof. The relation (9.31) follows from the relation in lemma 9.22 if we consider the leading coefficients on both sides. On the left the leading coefficient is $f[x_0,\ldots,x_k]$. The right-hand side has the form

$$\begin{aligned}\frac{x_k - x}{x_k - x_0}\big(f[x_0,\ldots,x_{k-1}]x^{k-1} + \text{lower degree terms}\big) + \\ \frac{x - x_0}{x_k - x_0}\big(f[x_1,\ldots,x_k]x^{k-1} + \text{lower degree terms}\big) \\ = \frac{f[x_1,\ldots,x_k] - f[x_0,\ldots,x_k]}{x_k - x_0}x^k + \text{lower degree terms}\end{aligned}$$

and from this (9.31) follows. ∎

The significance of theorem 9.23 is that it provides a simple formula for computing the coefficients of the interpolating polynomial in Newton form. The relation (9.31) also explains the name 'divided difference', and it should not come as a surprise that $f[x_0,\ldots,x_k]$ is related to the kth derivative of f, as we will see below.

It is helpful to organise the computations of divided differences in a table,

$$\begin{array}{llllll} x_0 & f[x_0] & & & & \\ x_1 & f[x_1] & f[x_0,x_1] & & & \\ x_2 & f[x_2] & f[x_1,x_2] & f[x_0,x_1,x_2] & & \\ x_3 & f[x_3] & f[x_2,x_3] & f[x_1,x_2,x_3] & f[x_0,x_1,x_2,x_3] & \\ \vdots & & & & & \end{array} \tag{9.32}$$

Here an entry in the table (except for the first two columns) is computed by subtracting the entry to the left and above from the entry to the left, and dividing by the last minus the first x_i involved. Then all the coefficients of the Newton form can be read off from the diagonal. An example will illustrate how this is used.

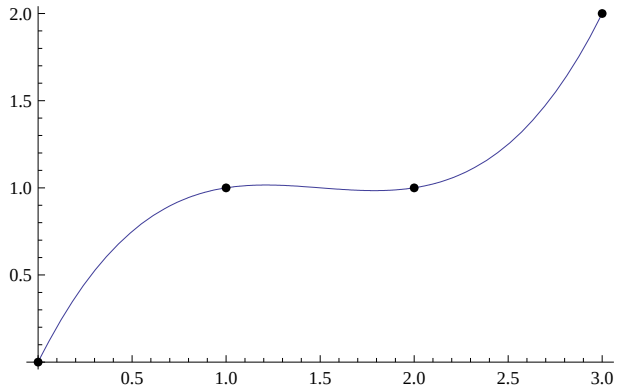


Figure 9.6. The data points and the interpolant in example 9.24.

Example 9.24. Suppose we have the data

x	0	1	2	3
$f(x)$	0	1	1	2

We want to compute the divided differences using (9.31) and organise the computations as in (9.32),

$$\begin{array}{llllll} x & f(x) & & & \\ 0 & 0 & & & \\ 1 & 1 & 1 & & \\ 2 & 1 & 0 & -1/2 & \\ 3 & 2 & 1 & 1/2 & 1/3 \end{array}$$

This means that the interpolating polynomial is

$$\begin{aligned} p_3(x) &= 0 + 1(x-0) - \frac{1}{2}(x-0)(x-1) + \frac{1}{3}(x-0)(x-1)(x-2) \\ &= x - \frac{1}{2}x(x-1) + \frac{1}{3}x(x-1)(x-2). \end{aligned}$$

A plot of this polynomial with the interpolation points is shown in figure 9.6. ∎

There is one more important property of divided differences that we need to discuss. If we look back on equation (9.24), we see that

$$c_1 = f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

From the mean value theorem for derivatives we can conclude from this that $f[x_0, x_1] = f'(\xi)$ for some number ξ in the interval (x_0, x_1), provided f' is contin-

uous in this interval. The relation (9.31) shows that higher order divided differences are built form lower order ones in a similar way, so it should come as no surprise that divided differences can be related to derivatives in general.

Theorem 9.25. *Let f be a function whose first k derivatives are continuous in the smallest interval $[a,b]$ that contains all the numbers $x_0,\ldots,x_k$. Then*

$$f[x_0,\ldots,x_k] = \frac{f^{(k)}(\xi)}{k!} \tag{9.33}$$

where ξ is some number in the interval (a,b).

We skip the proof of this theorem, but return to the Newton form of the interpolating polynomial,

$$p_n = f[x_0] + f[x_0,x_1](x-x_0) + \cdots + \cdots f[x_0,\ldots,x_n](x-x_0)\cdots(x-x_{n-1}).$$

Theorem 9.25 shows that divided differences can be associated with derivatives, so this formula is very similar to the formula for a Taylor polynomial. In fact, if we let all the interpolation points x_i approach a common number z, it is quite easy to show that the interpolating polynomial p_n approaches the Taylor polynomial

$$T_n(f;z)(x) = f(z) + f'(z)(x-z) + \cdots + f^{(n)}(z)\frac{(x-z)^n}{n!}.$$

9.2.4 Computing with the Newton form

Our use of polynomial interpolation will primarily be as a tool for developing numerical methods for differentiation, integration, and solution of differential equations. For such purposes the interpolating polynomial is just a step on the way to a final computational formula, and is not computed explicitly. There are situations though where one may need to determine the interpolating polynomial explicitly, and then the Newton form is usually preferred.

To use the Newton form in practice, we need two algorithms: One for determining the divided differences involved in the formula (9.29), and one for computing the value $p_n(x)$ of the interpolating polynomial for a given number x. We consider each of these in turn.

The Newton form of the polynomial that interpolates a given function f at the $n+1$ points $x_0,\ldots,x_n$ is given by

$$p_n(x) = f[x_0] + f[x_0,x_1](x-x_0) + \cdots + f[x_0,\ldots,x_n](x-x_0)\cdots(x-x_{n-1}),$$

and to represent this polynomial we need to compute the divided differences $f[x_0]$, $f[x_0,x_1]$, ..., $f[x_0,\ldots,x_n]$. The obvious way to do this is as indicated in the table (9.32) which we repeat here for convenience,

$$\begin{array}{llllll} x_0 & f[x_0] & & & & \\ x_1 & f[x_1] & f[x_0,x_1] & & & \\ x_2 & f[x_2] & f[x_1,x_2] & f[x_0,x_1,x_2] & & \\ x_3 & f[x_3] & f[x_2,x_3] & f[x_1,x_2,x_3] & f[x_0,x_1,x_2,x_3] & \\ \vdots & & & & & \end{array}$$

The natural way to do this is to start with the interpolation points $x_0, \ldots, x_n$ and the function f, and then compute the values in the table column by column. Let us denote the entries in the table by $(d_{i,k})_{i=0,k=0}^{n}$, where i runs down the columns and k indicates the column number, starting with 0 for the column with the function values. The first column of function values is special and must be computed separately. Otherwise we note that a value $d_{i,k}$ in column k is given by the two neighbouring values in column $k-1$,

$$d_{i,k} = \frac{d_{i,k-1} - d_{i-1,k-1}}{x_i - x_{i-k}}, \tag{9.34}$$

for $i \geq k$, while the other entries in column k are not defined. We start by computing the first column of function values. Then we can use the formula (9.34) to compute the next column. It would be natural to start by computing the diagonal entry and then proceed down the column. However, it is better to start with the last entry in the column, and proceed up, to the diagonal entry. The reason is that once an entry $d_{i,k}$ has been computed, the entry $d_{i,k-1}$ immediately to the left is not needed any more. Therefore, there is no need to use a two dimensional array; we can just start with the one dimensional array of function values and let every new column overwrite the one to the left. Since no column contains values above the diagonal, we end up with the correct divided differences in the one dimensional array at the end, see exercise 3.

Algorithm 9.26 (Computing divided differences)**.** *Let f be a given function, and $x_0, \ldots, x_n$ given interpolation points for some nonnegative integer n. After the code*

for $i := 0, 1, \ldots, n$
    $f_i := f(x_i)$;
for $k := 1, 2, \ldots n$
    for $i := n, n-1, \ldots, k$
        $f_i := (f_i - f_{i-1})/(x_i - x_{i-k})$;

has been performed, the array $\boldsymbol{f}$ *contains the divided differences needed for the Newton form of the interpolating polynomial, so*

$$p_n = f_0 + f_1(x - x_0) + f_2(x - x_0)(x - x_1) + \cdots + f_n(x - x_0)\cdots(x - x_{n-1}). \quad (9.35)$$

Note that this algorithm has two nested for-loops, so the number of subtractions is

$$\sum_{k=1}^{n}\sum_{i=k}^{n} 2 = \sum_{k=1}^{n} 2(n-k+1) = 2\sum_{k=1}^{n} k = n(n+1) = n^2 + n$$

which follows from the formula for the sum on the first n integers. We note that this grows with the square of the degree n, which is a consequence of the double for-loop. This is much faster than linear growth, which is what we would have if there was only the outer for-loop. However, for this problem the quadratic growth is not usually a problem since the degree tends to be low — rarely more than 10. If one has more points than this the general advice is to use some other approximation method which avoids high degree polynomials since these are also likely to lead to large rounding-errors.

The second algorithm that is needed is evaluation of the interpolation polynomial (9.35). Let us consider a specific example,

$$p_3(x) = f_0 + f_1(x - x_0) + f_2(x - x_0)(x - x_1) + f_3(x - x_0)(x - x_1)(x - x_2). \quad (9.36)$$

Given a number x, there is an elegant algorithm for computing the value of the polynomial which is based on rewriting (9.36) slightly as

$$p_3(x) = f_0 + (x - x_0)\Big(f_1 + (x - x_1)\big(f_2 + (x - x_2)f_3\big)\Big). \quad (9.37)$$

To compute $p_3(x)$ we start from the inner-most parenthesis and then repeatedly multiply and add,

$$\begin{aligned} s_3 &= f_3, \\ s_2 &= (x - x_2)s_3 + f_2, \\ s_1 &= (x - x_1)s_2 + f_1, \\ s_0 &= (x - x_0)s_1 + f_0. \end{aligned}$$

After this we see that $s_0 = p_3(x)$. This can easily be generalised to a more formal algorithm. Note that there is no need to keep the different (s_i)-values; we can just use one variable s and accumulate the calculations in this.

Algorithm 9.27 (Horner's rule)**.** *Let $x_0, \ldots, x_n$ be given numbers, and let $(f_k)_{k=0}^n$ be the coefficients of the polynomial*

$$p_n(x) = f_0 + f_1(x - x_0) + \cdots + f_n(x - x_0)\cdots(x - x_{n-1}). \tag{9.38}$$

After the code

```
s := f_n;
for k := n − 1, n − 2, ...0
    s := (x − x_k) ∗ s + f_k;
```

the variable s will contain the value of $p_n(x)$.

The total number of arithmetic operations in this algorithm is n additions, subtractions and multiplications. On the other hand, the last term in (9.38) on its own requires n multiplications. The simple reformulation (9.37) is therefore almost like magic.

9.2.5 Interpolation error

The interpolating polynomial p_n is an approximation to f, but unless f itself is a polynomial of degree n, there will be a nonzero error $e(x) = f(x) - p_n(x)$, see exercise 4. At times it is useful to have an explicit expression for the error.

Theorem 9.28. *Suppose f is interpolated by a polynomial of degree n at $n+1$ distinct points $x_0, \ldots, x_n$. Let $[a,b]$ be the smallest interval that contains all the interpolation points as well as the number x, and suppose that the function f has continuous derivatives up to order $n+1$ in $[a,b]$. Then the error $e(x) = f(x) - p_n(x)$ is given by*

$$e(x) = f[x_0,\ldots,x_n,x](x - x_0)\cdots(x - x_n) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0)\ldots(x - x_n), \tag{9.39}$$

where ξ_x is a number in the interval (a,b) that depends on x.

Proof. The second equality in (9.39) follows from (9.33), so we need to prove the first equality. For this we add the (arbitrary) number x as an interpolation point and consider interpolation with a polynomial of degree $n+1$ at the points x_0, $\ldots$, x_n, x. We use t as the free variable to avoid confusion with x. Then we know that

$$p_{n+1}(t) = p_n(t) + f[x_0,\ldots,x_n,x](t - x_0)\cdots(t - x_n).$$

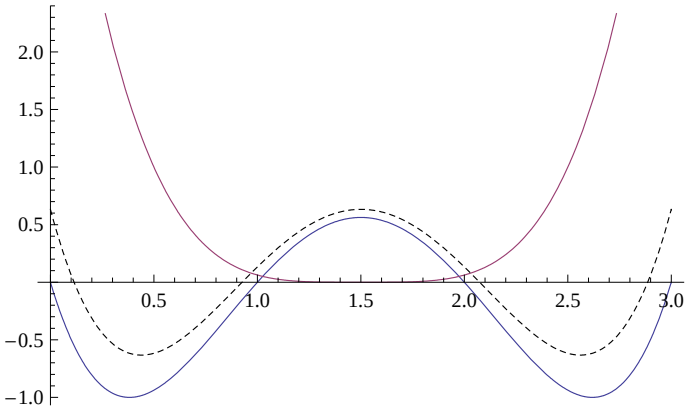


Figure 9.7. The solid, nonnegative graph is the polynomial factor $(x-3/2)^4$ in the error term for Taylor expansion of degree 3 about the $a = 3/2$, while the other solid graph is the polynomial part $x(x-1)(x-2)(x-3)$ of the error term for interpolation at 0, 1, 2, 3. The dashed graph is the smallest possible polynomial part of an error terms for interpolation at 4 points in $[0,3]$.

Since p_{n+1} interpolates f at $t = x$ we have $p_{n+1}(x) = f(x)$ so

$$f(x) = p_n(x) + f[x_0, \ldots, x_n, x](x - x_0) \cdots (x - x_n)$$

which proves the first relation in (9.39). ∎

Theorem 9.28 has obvious uses for assessing the error in polynomial interpolation and will prove very handy in later chapters.

The error term in (9.39) is very similar to the error term in the Taylor expansion (9.14). A natural question to ask is therefore: Which approximation method will give the smallest error, Taylor expansion or interpolation? Since the only essential difference between the two error terms is the factor $(x-a)^{n+1}$ in the Taylor case and the factor $(x-x_0)\cdots(x-x_n)$ in the interpolation case, a reasonable way to compare the methods is to compare the two polynomials $(x-a)^{n+1}$ and $(x-x_0)\cdots(x-x_n)$.

In reality, we do not just have two approximation methods but infinitely many, since there are infinitely many ways to choose the interpolation points. In figure 9.7 we compare the two most obvious choices in the case $n = 3$ for the interval $[0,3]$: Taylor expansion about the midpoint $a = 3/2$ and interpolation at the integers 0, 1, 2, and 3. In the Taylor case, the polynomial $(x-3/2)^4$ is nonnegative and small in the interval $[1,2]$, but outside this interval it grows quickly and soon becomes larger than the polynomial $x(x-1)(x-2)(x-3)$ corresponding to interpolation at the integers. We have also included a plot of a third polynomial which corresponds to the best possible interpolation points in the sense

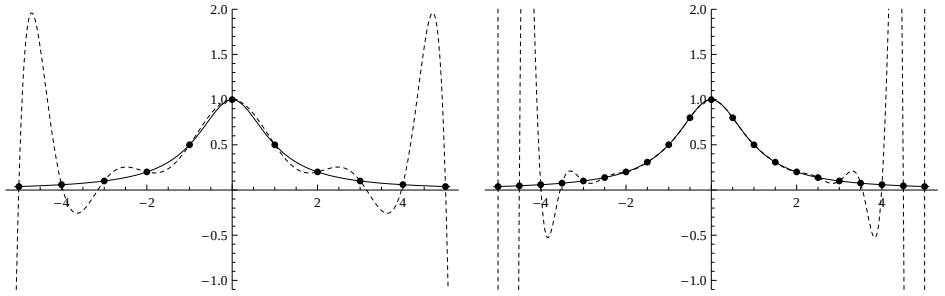


Figure 9.8. Interpolation of the function $f(x) = 1/(1+x^2)$ on the interval $[-5,5]$ with polynomials of degree 10 in (a), and degree 20 in (b). The points are uniformly distributed in the interval in each case.

that the maximum value of this polynomial is as small as possible in the interval $[0,3]$, given that its leading coefficient should be 1.

If used sensibly, polynomial interpolation will usually provide a good approximation to the underlying data. As the distance between the data points decreases, either by increasing the number of points or by moving the points closer together, the approximation can be expected to become better. However, we saw that there are functions for which Taylor approximation does not work well, and the same may happen with interpolation. As for Taylor approximation, the problem arises when the derivatives of the function to be approximated become large. A famous example is the so-called Runge function $1/(1+x^2)$ on the interval $[-5,5]$. Figure 9.8 shows the interpolants for degree 10 and degree 20. In the middle of the interval, the error becomes smaller when the degree is increased, but towards the ends of the interval the error becomes larger when the degree increases.

9.3 Summary

In this chapter we have considered two different ways of constructing polynomial interpolants. We first reviewed Taylor polynomials briefly, and then studied polynomial interpolation in some detail. Taylor polynomials are for the main part a tool that is used for various pencil and paper investigations, while interpolation is often used as a tool for constructing numerical methods, as we will see in later chapters. Both Taylor polynomials and polynomial interpolation are methods of approximation and so it is important to keep track of the error, which is why the error formulas are important.

In this chapter we have used polynomials all the time, but have written them in different forms. This illustrates the important principle that there are many

different ways to write polynomials, and a problem may simplify considerably by adapting the form of the polynomial to the problem at hand.

Exercises

9.1 The Taylor polynomials of e^x, $\cos x$ and $\sin x$ expanded around zero are

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \cdots$$
$$\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \cdots$$
$$\sin x = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \cdots$$

Calculate the Taylor polynomial of the complex exponential e^{ix}, compare with the Taylor polynomials above, and explain why Euler's formula $e^{ix} = \cos x + i \sin x$ is reasonable.

9.2 The data

x	0	1	3	4
$f(x)$	1	0	2	1

are given.

a) Determine the Lagrange form of the cubic polynomial that interpolates the data.

b) Determine the Newton form of the interpolating polynomial.

c) Verify that the solutions in (a) and (b) are the same.

9.3 Use algorithm 9.26 to compute the divided differences needed to determine the Newton form of the interpolating poynomial in exercise 2. Verify that no data are lost when variables are overwritten.

9.4 **a)** We have the data

x	0	1	2
$f(x)$	2	1	0

which have been sampled from the straight line $y = 2 - x$. Determine the Newton form of the quadratic, interpolating polynomial, and compare it to the straight line. What is the difference?

b) Suppose we are doing interpolation at $x_0, \ldots, x_n$ with polynomials of degree n. Show that if the function f to be interpolated is a polynomial p of degree n, then the interpolant p_n will be identically equal to p. How does this explain the result in (a)?

9.5 Suppose we have the data

$$(0, y_0), \quad (1, y_1), \quad (2, y_2), \quad (3, y_3) \tag{9.40}$$

where we think of $y_i = f(i)$ as values being sampled from an unknown function f. In this problem we are going to find formulas that approximate f at various points using cubic interpolation.

a) Determine the straight line p_1 that interpolates the two middle points in (9.40), and use $p_1(3/2)$ as an approximation to $f(3/2)$. Show that

$$f(3/2) \approx p_1(3/2) = \frac{1}{2}\big(f(1) + f(2)\big).$$

Find an expression for the error.

b) Determine the cubic polynomial p_3 that interpolates the data (9.40) and use $p_3(3/2)$ as an approximation to $f(3/2)$. Show that then

$$f(3/2) \approx p_3(3/2) = \frac{-y_0 + 9y_1 - 9y_2 + y_3}{16}.$$

What is the error?

c) Sometimes we need to estimate f outside the interval that contains the interpolation points; this is called *extrapolation*. Use the same approach as in (a), but find an approximation to $f(4)$. What is the error?

9.6 **a)** The data

x	0	1	2	3
$f(x)$	0	1	4	9

is sampled from the function $f(x) = x^2$. Determine the third order divided difference $f[0,1,2,3]$.

b) Explain why we always have $f[x_0,\ldots,x_n] = 0$ if f is a polynomial of degree at most $n-1$.