# 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.

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 The Taylor polynomial with remainder

A discussion of Taylor polynomials involves two parts: The Taylor polynomial itself, and the error, the remainder, committed in approximating a function by a polynomial. Let us consider each of these in turn.

### 9.1.1 The Taylor polynomial

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 $T_n(f;a)$ (sometimes shortened to $T_n(x)$) and satisfies the conditions*

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

The conditions (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.

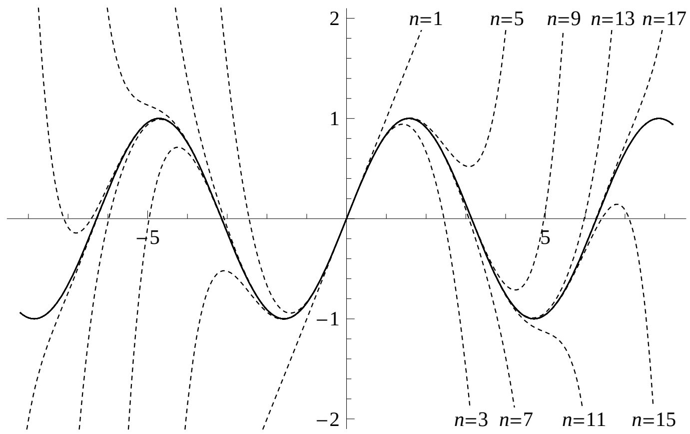



**Figure 9.1**. The Taylor polynomials of $\sin x$ (around $a = 0$) for degrees 1 to 17.

**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 an ever larger interval.

Formula (9.2) is a classical result of calculus which is proved in most calculus books. Note however that the polynomial in (9.2) is written in non-standard form.

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

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

This is an important observation: It is wise to adapt the form of the polynomial to the problem that is to be solved. We will see another example of this when we discuss interpolation below.

The elementary exponential and trigonometric functions have very simple and important Taylor polynomials.

**Example 9.4** (The Taylor polynomial of $e^x$)**.** The function $f(x) = e^x$ has the nice property that $f^{(n)}(x) = e^x$ for all integers $n \geq 0$. The Taylor polynomial about $a = 0$ is therefore very simple since $f^{(n)}(0) = 1$ for all $n$. The general term in the Taylor polynomial then becomes

$$\frac{(x-a)^k f^{(k)}(a)}{k!} = \frac{x^k}{k!}.$$

This means that the Taylor polynomial of degree $n$ about $a = 0$ for $f(x) = e^x$ is given by

$$T_n(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}.$$

For the exponential function the Taylor polynomials will be very good approximations for large values of $n$. More specifically, it can be shown that for any value of $x$, the difference between $T_n(x)$ and $e^x$ can be made as small as we wish if we just let $n$ be big enough. This is often expressed by writing

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots.$$

It turns out that the Taylor polynomials of the trigonometric functions $\sin x$ and $\cos x$ converge in a similar way. In exercise 4 these three Taylor polynomials are linked together via a classical formula. ∎

### 9.1.2 The remainder

The Taylor polynomial $T_n(f)$ 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 9.5.** *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*

$$R_n(f;a)(x) = \frac{1}{n!}\int_a^x f^{(n+1)}(t)(x-t)^n\,dt. \tag{9.3}$$

*The remainder may also be written as*

$$R_n(f;a)(x) = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi), \tag{9.4}$$

*where $\xi$ 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 9.6.** 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 (9.4), 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 $10^{-5}$.

We want to find the smallest $n$ such that

$$\left|\frac{x^{n+1}}{(n+1)!} f^{(n+1)}(\xi)\right| \le 10^{-5}, \tag{9.5}$$

where the function $f(x) = \sin x$ and $\xi$ 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

$$\left|f^{(n+1)}(\xi)\right| \le 1. \tag{9.6}$$

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 $\xi$ would be much more difficult. If we insert the estimate (9.6) on the left in (9.5), we can also change our required inequality,

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

If we manage to find an $n$ such that this last inequality is satisfied, then (9.5) will also be satisfied. Since $x \in [-1,1]$ we know that $|x| \le 1$ so this last inequality will be satisfied if

$$\frac{1}{(n+1)!} \le 10^{-5}. \tag{9.7}$$

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 (9.7) 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 check the approximation at $x = 1$, we find $p_8(1) \approx 0.8414682$. Comparing with the exact value $\sin 1 \approx 0.8414710$, we find that the error is roughly $2.73 \times 10^{-6}$, which is close to the upper bound $1/9!$ 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 (9.4) will be most useful for us, and for easy reference we record the complete Taylor expansion in a corollary.

**Corollary 9.7.** *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,*

$$f(x) = f(a) + (x-a)f'(a) + \cdots + \frac{(x-a)^n}{n!} f^{(n)}(a) + \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi_x), \tag{9.8}$$

*where $\xi_x$ 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 (9.8) lets us control the error in the Taylor approximation. It turns out that the error behaves quite differently for different functions.

**Example 9.8** (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

$$e(x) = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi). \tag{9.9}$$

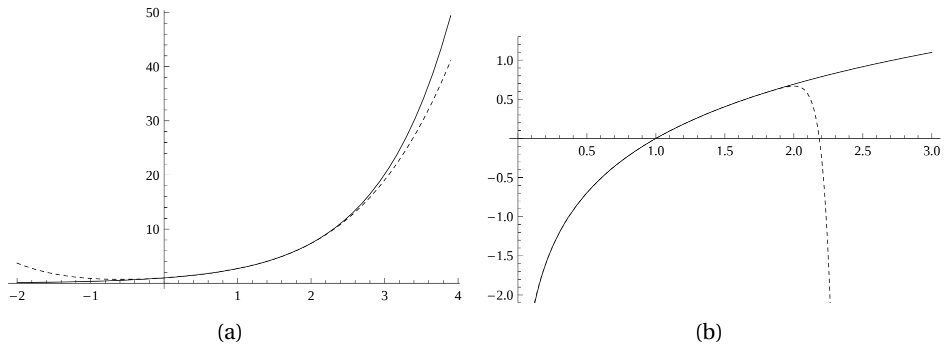



**Figure 9.2**. In (a) the Taylor polynomial of degree 4 about the point $a = 1$ for the function $f(x) = e^x$ is shown. Figure (b) shows the Taylor polynomial of degree 20 for the function $f(x) = \log x$, also about the point $a = 1$.

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 what the value of $b$ is. ∎

**Example 9.9** (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 9.8. ∎

**Example 9.10** (Taylor polynomials for $f(x) = \ln x$)**.** The plot in figure 9.2 shows the logarithm 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.

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

$$f^{(k)}(x) = (-1)^{k+1}(k-1)!\,x^{-k}, \quad k \geq 1. \tag{9.10}$$

Since $a = 1$, this means that the general term in the Taylor polynomial is

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

The Taylor expansion (9.8) therefore 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 $\xi$ 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 therefore $\xi^{-n-1} < 1$. The error is then 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 (9.3) 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 $c > 1$. 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). ∎

## Exercises

**1** In this exercise we are going to see that the calculations simplify if we adapt the form of a polynomial to the problem to be solved. The function $f$ is a given function to be approximated by a quadratic polynomial near $x = a$, and it is assumed that $f$ can be differentiated twice at $a$.

- **a)** Assume that the quadratic Taylor polynomial is in the form $p(x) = b_0 + b_1 x + b_2 x^2$, and determine the unknown coefficients from the three conditions $p(a) = f(a)$, $p'(a) = f'(a)$, $p''(a) = f''(a)$.
- **b)** Repeat (a), but write the unknown polynomial in the form $p(x) = b_0 + b_1(x-a) + b_2(x-a)^2$.

**2** Find the second order Taylor approximation of the following functions at the given point $a$.

- **a)** $f(x) = x^3$, $a = 1$
- **b)** $f(x) = 12x^2 + 3x + 1$, $a = 0$
- **c)** $f(x) = 2^x$, $a = 0$

**3** In many contexts, the approximation $\sin x \approx x$ is often used.

- **a)** Explain why this approximation is reasonable.
- **b)** Estimate the error in the approximation for $x$ in the interval $[0, 0.1]$

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

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

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 Interpolation and the Newton form

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.

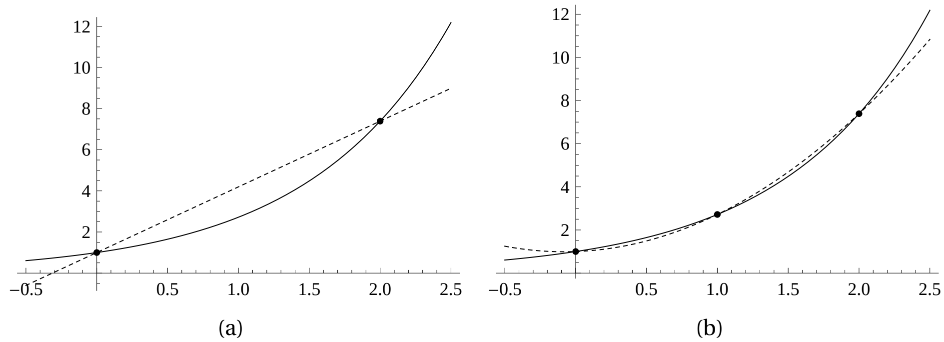



**Figure 9.3.** Interpolation of $e^x$ at two points with a secant (a), and at three points with a parabola (b).

### 9.2.1 The interpolation problem

Just like Taylor approximation is a generalisation of the tangent, interpolation is a generalisation of the secant, see figure 9.3.

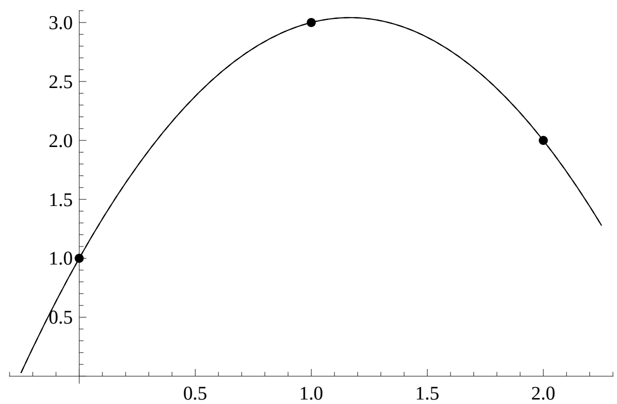
The idea behind polynomial interpolation is simple: We approximate a function $f$ by a polynomial $p$ by forcing $p$ to have the same function values as $f$ at a number of points. A general parabola has three free coefficients, and we should therefore expect to be able to force a parabola through three arbitrary points. More generally, suppose we have $n+1$ distinct numbers $\{x_i\}_{i=0}^n$ scattered throughout an 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 the numbers $\{x_i\}_{i=0}^n$.

**Problem 9.11** (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 numbers in $[a,b]$. The polynomial interpolation problem is to find a polynomial $p_n = P(f;x_0,\dots,x_n)$ of degree $n$ that matches $f$ at each $x_i$,*

$$p_n(x_i) = f(x_i), \quad \text{for } i = 0, 1, \dots, n. \tag{9.11}$$

*The numbers $\{x_i\}_{i=0}^n$ are called* interpolation points*, the conditions* (9.11) *are called the* interpolation conditions*, and the polynomial $p_n = P(f;x_0,\dots,x_n)$ is called a polynomial* interpolant*.*

The notation $P(f;x_0,\dots,x_n)$ for a 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 just use $p_n$ when no confusion is possible.




(a)

**Figure 9.4**. Three interpolation points and the corresponding quadratic interpolating polynomial.

In many situations the function $f$ may not be known, just its function values at the points $\{x_i\}_{i=0}^n$, as in the following example.

**Example 9.12.** Suppose we want to find a polynomial that passes through the three points $(0,1)$, $(1,3)$, and $(2,2)$. In other words, we want to find a polynomial $p$ such that

$$p(0) = 1, \quad p(1) = 3, \quad p(2) = 2. \tag{9.12}$$

Since there are three points it is natural to try and accomplish this with a quadratic polynomial, i.e., we assume that $p(x) = c_0 + c_1 x + c_2 x^2$. If we insert this in the conditions (9.12) we obtain the three equations

$$\begin{aligned} 1 &= p(0) = c_0, \\ 3 &= p(1) = c_0 + c_1 + c_2, \\ 2 &= p(2) = c_0 + 2c_1 + 4c_2. \end{aligned}$$

We solve these and find $c_0 = 1$, $c_1 = 7/2$, and $c_2 = -3/2$, so $p$ is given by

$$p(x) = 1 + \frac{7}{2}x - \frac{3}{2}x^2.$$

A plot of this polynomial and the interpolation points is shown in figure 9.4. ∎

There are at least four questions raised by problem 9.11: Is there a polynomial of degree $n$ that satisfies the interpolation conditions (9.11)? How many such polynomials are there? How can we find one such polynomial? What is a convenient way to write an interpolating polynomial?

### 9.2.2 The Newton form of the interpolating polynomial

We start by considering the last of the four questions above. We have already seen that by writing polynomials in a particular form, the computations of the Taylor polynomial simplified. This is also the case for interpolating polynomials.

**Definition 9.13** (Newton form)**.** *Let $\{x_i\}_{i=0}^n$ be $n+1$ distinct real numbers. The Newton form of a polynomial of degree $n$ is an expression in the form*

$$\begin{aligned} p_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \cdots \\ + c_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}). \end{aligned} \tag{9.13}$$

The advantage of the Newton form will become evident when we consider some examples.

**Example 9.14** (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 have to choose $c_0 = f(x_0)$,

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

**Example 9.15** (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 $f(x_0) = p_1(x_0) = c_0$, while interpolation at $x_1$ yields

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

Together this means that

$$c_0 = f(x_0), \quad c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{9.14}$$

We note that $c_0$ remains the same as in the case $n=0$. $\blacksquare$

**Example 9.16** (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

$$\begin{aligned} f(x_0) &= p_2(x_0) = c_0, \\ f(x_1) &= p_2(x_1) = c_0 + c_1(x_1 - x_0), \end{aligned}$$

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

$$c_2 = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0}. \quad \blacksquare \tag{9.15}$$

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 $\big(x_0, f(x_0)\big)$, $\big(x_1, f(x_1)\big)$, ..., $\big(x_k, f(x_k)\big)$. This becomes clear if we write down all the equations,

$$\begin{aligned}
f(x_0) &= c_0,\\
f(x_1) &= c_0 + c_1(x_1 - x_0),\\
f(x_2) &= c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1),\\
&\ \ \vdots\\
f(x_k) &= c_0 + c_1(x_k - x_0) + c_2(x_k - x_0)(x_k - x_1) + \cdots\\
&\qquad + c_{k-1}(x_k - x_0)\cdots(x_k - x_{k-2}) + c_k(x_k - x_0)\cdots(x_k - x_{k-1}).
\end{aligned} \tag{9.16}$$

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 $\big(x_0, f(x_0)\big)$, $\big(x_1, f(x_1)\big)$, ..., $\big(x_k, f(x_k)\big)$, so the following theorem is immediate.

**Theorem 9.17.** *Let $f$ be a given function and $x_0$, ..., $x_n$ given and distinct interpolation points. There is a unique polynomial of degree $n$ which interpolates $f$ at these points. If the interpolating polynomial is expressed in Newton form,*

$$p_n(x) = c_0 + c_1(x - x_0) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}), \tag{9.17}$$

*then $c_k$ depends only on $\big(x_0, f(x_0)\big)$, $\big(x_1, f(x_1)\big)$, ..., $\big(x_k, f(x_k)\big)$ which is indicated by the notation*

$$c_k = f[x_0, \ldots, x_k] \tag{9.18}$$

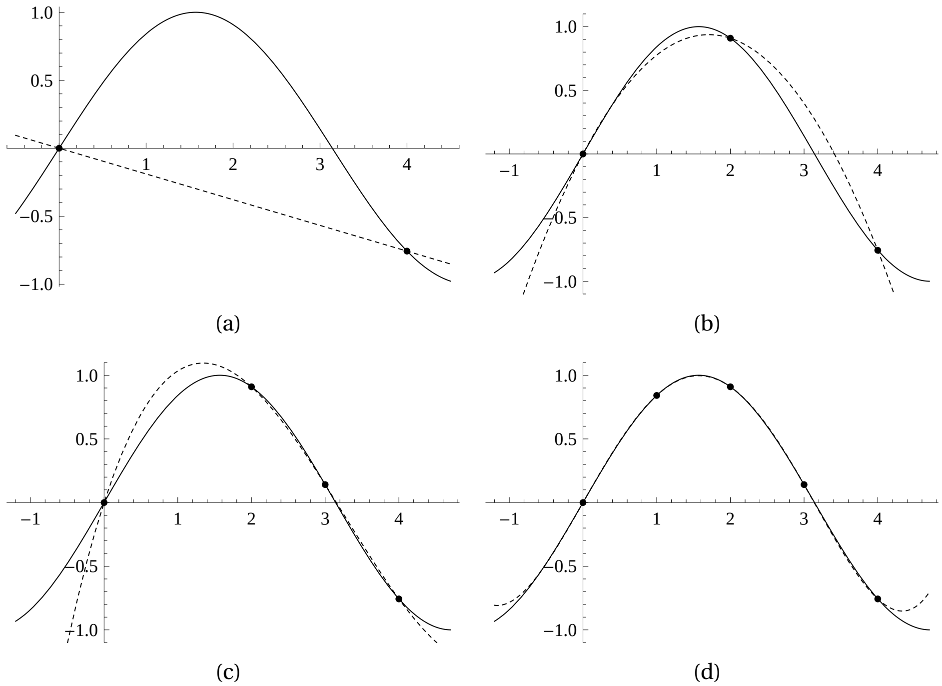



**Figure 9.5.** Interpolation of $\sin x$ with a line (a), a parabola (b), a cubic (c), and a quartic polynomial (d).

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

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

Some examples of interpolation are shown in figure 9.5. Note how the quality of the approximation improves with increasing degree.

**Proof.** Most of this theorem is a direct consequence of writing the interpolating polynomial in Newton form, which becomes

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \ldots, x_n](x - x_0)\cdots(x - x_{n-1}) \quad (9.19)$$

when we write the coeefficients as in (9.18). The coefficients can be computed, one by one, from the equations (9.16), starting with $c_0$. The uniqueness follows since there is no choice in solving the equations (9.16); there is one and only one solution. ∎

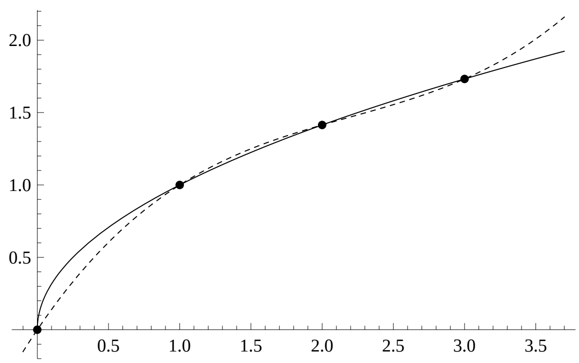



**Figure 9.6**. The function $f(x) = \sqrt{x}$ (solid) and its cubic interpolant at the four points 0, 1, 2, and 3 (dashed).

Theorem 9.17 answers the questions raised above: Problem 9.11 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.

**Example 9.18.** 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. In this case the Newton form becomes

$$p_3(x) = c_0 + c_1 x + c_2 x(x-1) + c_3 x(x-1)(x-2).$$

The interpolation conditions become

$$\begin{aligned} 0 &= c_0, \\ 1 &= c_0 + c_1, \\ \sqrt{2} &= c_0 + 2c_1 + 2c_2, \\ \sqrt{3} &= c_0 + 3c_1 + 6c_2 + 6c_3. \end{aligned}$$

Not surprisingly, the equations are triangular and we find

$$c_0 = 0, \quad c_1 = 1, \quad c_2 = -(1-\sqrt{2}/2), \quad c_3 = (3+\sqrt{3}-3\sqrt{2})/6$$

Figure 9.6 shows a plot of this interpolant. ∎

We emphasise that the Newton form is just one way to write the interpolating polynomial — there are many alternatives. One of these is the *Lagrange form* which is discussed in exercise 1 below.

## Exercises

**1** The data

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

are given.

**a)** Write the cubic interpolating polynomial in the form

$$p_3(x) = c_0(x-1)(x-2)(x-3) + c_1 x(x-2)(x-3) + c_2 x(x-1)(x-3) + c_3 x(x-1)(x-2),$$

and determine the coefficients from the interpolation conditions. This is called the *Lagrange form* of the interpolating polynomial.

**b)** Determine the Newton form of the interpolating polynomial.

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

**2** In this exercise we are going to consider an alternative proof that the interpolating polynomial is unique.

**a)** Suppose that there are two quadratic polynomials $p_1$ and $p_2$ that interpolate a function $f$ at the three points $x_0$, $x_1$ and $x_2$. Consider the difference $p = p_2 - p_1$. What is the value of $p$ at the interpolation points?

**b)** Use the observation in (a) to prove that $p_1$ and $p_2$ must be the same polynomial.

**c)** Generalise the results in (a) and (b) to polynomials of degree $n$.

## 9.3 Divided differences

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

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

$$p(x) = \frac{x_k - x}{x_k - x_0} q_0(x) + \frac{x - x_0}{x_k - x_0} q_1(x). \tag{9.20}$$

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

We first 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 (9.20) in a more explicit way.

**Lemma 9.19.** *Let $P(f; x_0, \dots, x_k)$ denote the polynomial of degree $k-1$ that interpolates the function $f$ at the points $x_0, \dots, x_k$. Then*

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

From this lemma we can deduce a useful formula for the coefficients of the interpolating polynomial. We first recall that the term of highest degree in a polynomial is referred to as the *leading term*, and the coefficient of the leading term is referred to as the *leading coefficient*. From equation (9.17) we see that the leading coefficient of the interpolating polyonomial $p_k$ is $f[x_0, \dots, x_k]$. This observation combined with Lemma 9.19 leads to a useful formula for $f[x_0, \dots, x_k]$.

**Theorem 9.20.** *Let $c_k = f[x_0, \dots, x_k]$ denote the leading coefficient of the interpolating polynomial $P(f; x_0, \dots, x_k)$. This is called a* $k$th order divided difference *of $f$ and satisfies the relations $f[x_0] = f(x_0)$, and*

$$f[x_0, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \tag{9.21}$$

*for $k > 0$.*

**Proof.** The relation (9.21) follows from the relation in lemma 9.19 if we consider the leading coefficients on both sides. On the left the leading coefficient is $f[x_0, \dots, x_k]$. The right-hand side has the form

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

and from this (9.21) follows. ∎

The significance of theorem 9.20 is that it provides a simple formula for computing the coefficients of the interpolating polynomial in Newton form. The relation (9.21) also explains the name 'divided difference', and it should not come as a surprise that $f[x_0,\dots,x_k]$ is related to the $k$th derivative of $f$, as we will see below.

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

$$\begin{array}{lllll}
x_0 & f[x_0] & & & \\
x_1 & f[x_1] & f[x_0,x_1] & & \\
x_2 & f[x_2] & f[x_1,x_2] & f[x_0,x_1,x_2] & \\
x_3 & f[x_3] & f[x_2,x_3] & f[x_1,x_2,x_3] & f[x_0,x_1,x_2,x_3] \\
\vdots & & & &
\end{array} \tag{9.22}$$

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.

**Example 9.21.** Suppose we have the data

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

We want to compute the divided differences using (9.21) and organise the computations as in (9.22),

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

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 9.7. ∎

There is one more important property of divided differences that we need to discuss. If we look back on equation (9.14), we see that

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

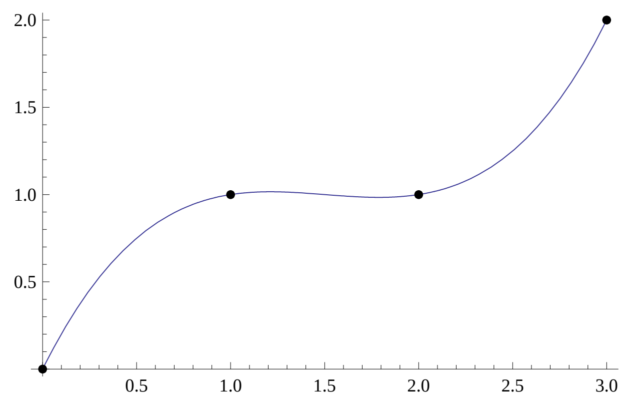



**Figure 9.7.** The data points and the interpolant in example 9.21.

From the mean value theorem for derivatives we can conclude from this that $f[x_0, x_1] = f'(\xi)$ for some number $\xi$ in the interval $(x_0, x_1)$, provided $f'$ is continuous in this interval. The relation (9.21) 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 9.22.** *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*

$$f[x_0, \ldots, x_k] = \frac{f^{(k)}(\xi)}{k!} \tag{9.23}$$

*where $\xi$ 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 9.22 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

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

## Exercises

**1** **a)** The data

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

are 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 $f[x_0,\dots,x_n] = 0$ if $f$ is a polynomial of degree at most $n-1$.

**2** **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 $x_0, \dots, x_n$ 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)?

**3** Suppose we have the data

$$(0, y_0), \quad (1, y_1), \quad (2, y_2), \quad (3, y_3) \tag{9.24}$$

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 (9.24), 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 (9.24) 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?

## 9.4 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 (9.19), 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 (9.22) which we repeat here for convenience,

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

We 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 $k-1$,

$$d_{i,k} = \frac{d_{i,k-1} - d_{i-1,k-1}}{x_i - x_{i-k}}, \tag{9.25}$$

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 (9.25) 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 1.

**Algorithm 9.23** (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** $i = n, n-1, \ldots, k$
    $f_i = (f_i - f_{i-1})\big/(x_i - x_{i-k})$;

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

$$p_n = f_0 + f_1(x - x_0) + f_2(x - x_0)(x - x_1) + \cdots + f_n(x - x_0)\cdots(x - x_{n-1}). \quad (9.26)$$

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 considerable rounding-errors.

The second algorithm that is needed is evaluation of the interpolation polynomial (9.26). Let us consider a specific example,

$$p_3(x) = f_0 + f_1(x - x_0) + f_2(x - x_0)(x - x_1) + f_3(x - x_0)(x - x_1)(x - x_2). \quad (9.27)$$

Given a number $x$, there is an elegant algorithm for computing the value of the polynomial which is based on rewriting (9.27) slightly as

$$p_3(x) = f_0 + (x - x_0)\Big(f_1 + (x - x_1)\big(f_2 + (x - x_2) f_3\big)\Big). \tag{9.28}$$

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 9.24** (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*

$$p_n(x) = f_0 + f_1(x - x_0) + \cdots + f_n(x - x_0)\cdots(x - x_{n-1}). \tag{9.29}$$

*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 $p_n(x)$.*

## Exercises

**1** Use algorithm 9.23 to compute the divided differences needed to determine the Newton form of the interpolating poynomial in exercise 9.3.1. Verify that no data are lost when variables are overwritten.

## 9.5 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 2. At times it is useful to have an explicit expression for the error.

**Theorem 9.25.** *Suppose $f$ is interpolated by a polynomial of degree $n$ at $n+1$ distinct points $x_0, \dots, x_n$. 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*

$$e(x) = f[x_0,\dots,x_n,x](x-x_0)\cdots(x-x_n) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x-x_0)\dots(x-x_n), \quad (9.30)$$

*where $\xi_x$ is a number in the interval $(a,b)$ that depends on $x$.*

**Proof.** The second equality in (9.30) follows from (9.23), so our job is 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 $x_0, \dots, x_n, x$. We use $t$ as the free variable to avoid confusion with $x$. Then we know that

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

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

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

which proves the first relation in (9.30). ∎

Theorem 9.25 has obvious uses for assessing the error in polynomial interpolation and will prove very handy in later chapters.

The error term in (9.30) is very similar to the error term in the Taylor expansion (9.8). 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 9.8 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

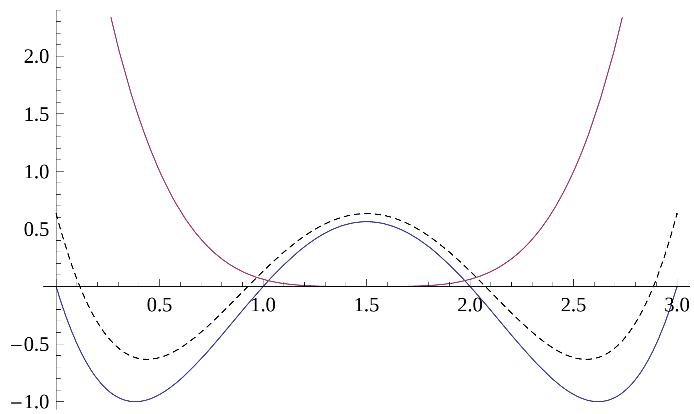



**Figure 9.8**. The solid, nonnegative graph is the polynomial factor $(x-3/2)^4$ in the error term for Taylor expansion of degree 3 about the $a = 3/2$, while the other solid graph is the polynomial part $x(x-1)(x-2)(x-3)$ of the error term for interpolation at 0, 1, 2, 3. The dashed graph is the smallest possible polynomial part of an error terms for interpolation at 4 points in $[0,3]$.

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 9.9 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.

## 9.6 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

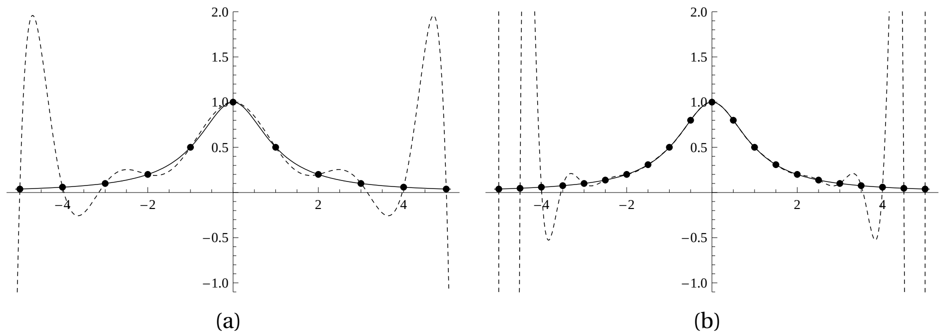



**Figure 9.9**. Interpolation of the function $f(x) = 1/(1+x^2)$ on the interval $[-5,5]$ with polynomials of degree 10 in (a), and degree 20 in (b). The points are uniformly distributed in the interval in each case.

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.