

Introduction to Numerical Analysis

Doron Levy

Department of Mathematics

and

Center for Scientific Computation and Mathematical Modeling (CSCAMM)

University of Maryland

January 31, 2017

Contents

1	Introduction	1
2	Methods for Solving Nonlinear Problems	2
2.1	Preliminary Discussion	2
2.1.1	Are there any roots anywhere?	3
2.1.2	Examples of root-finding methods	5
2.2	Iterative Methods	6
2.3	The Bisection Method	8
2.4	Newton's Method	11
2.5	The Secant Method	15
3	Interpolation	19
3.1	What is Interpolation?	19
3.2	The Interpolation Problem	20
3.3	Newton's Form of the Interpolation Polynomial	22
3.4	The Interpolation Problem and the Vandermonde Determinant	23
3.5	The Lagrange Form of the Interpolation Polynomial	25
3.6	Divided Differences	28
3.7	The Error in Polynomial Interpolation	31
3.8	Interpolation at the Chebyshev Points	34
3.9	Hermite Interpolation	41
3.9.1	Divided differences with repetitions	43
3.9.2	The Lagrange form of the Hermite interpolant	44
3.10	Spline Interpolation	47
3.10.1	Cubic splines	50
3.10.2	What is natural about the natural spline?	54
4	Approximations	57
4.1	Background	57
4.2	The Minimax Approximation Problem	62
4.2.1	Existence of the minimax polynomial	63
4.2.2	Bounds on the minimax error	65
4.2.3	Characterization of the minimax polynomial	66
4.2.4	Uniqueness of the minimax polynomial	66
4.2.5	The near-minimax polynomial	67
4.2.6	Construction of the minimax polynomial	68
4.3	Least-squares Approximations	70
4.3.1	The least-squares approximation problem	70
4.3.2	Solving the least-squares problem: a direct method	70
4.3.3	Solving the least-squares problem with orthogonal polynomials	73
4.3.4	The weighted least-squares problem	74

4.3.5	Orthogonal polynomials	76
4.4	Examples of orthogonal polynomials	78
4.4.1	Another approach to the least-squares problem	80
4.4.2	Properties of orthogonal polynomials	86
5	Numerical Differentiation	89
5.1	Basic Concepts	89
5.2	Differentiation Via Interpolation	91
5.3	The Method of Undetermined Coefficients	95
5.4	Richardson's Extrapolation	97
6	Numerical Integration	100
6.1	Basic Concepts	100
6.2	Integration via Interpolation	104
6.3	Composite Integration Rules	107
6.4	Additional Integration Techniques	109
6.4.1	The method of undetermined coefficients	109
6.4.2	Change of an interval	110
6.5	Simpson's Integration	111
6.5.1	The quadrature error	111
6.5.2	Composite Simpson rule	114
6.6	Weighted Quadratures	115
6.7	Gaussian Quadrature	116
6.7.1	Maximizing the quadrature's accuracy	116
6.7.2	Convergence and error analysis	119
6.8	Romberg Integration	122
	Bibliography	124
	index	125

1 Introduction

2 Methods for Solving Nonlinear Problems

2.1 Preliminary Discussion

In this chapter we will learn methods for approximating solutions of nonlinear algebraic equations. We will limit our attention to the case of finding roots of a single equation of one variable. Thus, given a function, $f(x)$, we will be interested in finding points x^* , for which $f(x^*) = 0$. A classical example that we are all familiar with is the case in which $f(x)$ is a quadratic equation. If, $f(x) = ax^2 + bx + c$, it is well known that the roots of $f(x)$ are given by

$$x_{1,2}^* = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

These roots may be complex or repeat (if the discriminant vanishes). This is a simple case in which they can be computed using a closed analytic formula. There exist formulas for finding roots of polynomials of degree 3 and 4, but these are rather complex. In more general cases, when $f(x)$ is a polynomial of degree that is ≥ 5 , formulas for the roots no longer exist. Of course, there is no reason to limit ourselves to study polynomials, and in most cases, when $f(x)$ is an arbitrary function, there are no analytic tools for calculating the desired roots. Instead, we must use approximation methods. In fact, even in cases in which exact formulas are available (such as with polynomials of degree 3 or 4) an exact formula might be too complex to be used in practice, and approximation methods may quickly provide an accurate solution.

An equation $f(x) = 0$ may or may not have solutions. We are not going to focus on finding methods to decide whether an equation has a solution or not, but we will look for approximation methods assuming that solutions actually exist. We will also assume that we are looking only for real roots. There are extensions of some of the methods that we will describe to the case of complex roots but we will not deal with this case. Even with the simple example of the quadratic equation, it is clear that a nonlinear equation $f(x) = 0$ may have more than one root. We will not develop any general methods for calculating the number of the roots. This issue will have to be dealt with on a case by case basis. We will also not deal with general methods for finding *all* the solutions of a given equation. Rather, we will focus on approximating *one* of the solutions.

The methods that we will describe, all belong to the category of *iterative* methods. Such methods will typically start with an initial guess of the root (or of the neighborhood of the root) and will gradually attempt to approach the root. In some cases, the sequence of iterations will converge to a limit, in which case we will then ask if the limit point is actually a solution of the equation. If this is indeed the case, another question of interest is how fast does the method converge to the solution? To be more precise, this question can be formulated in the following way: how many iterations of the method are required to guarantee a certain accuracy in the approximation of the solution of the equation.

2.1.1 Are there any roots anywhere?

There really are not that many general tools to knowing up front whether the root-finding problem can be solved. For our purposes, the most important issue will be to obtain some information about whether a root exists or not, and if a root does exist, then it will be important to make an attempt to estimate an interval to which such a solution belongs. One of our first attempts in solving such a problem may be to try to plot the function. After all, if the goal is to solve $f(x) = 0$, and the function $f(x)$ can be plotted in a way that the intersection of $f(x)$ with the x -axis is visible, then we should have a rather good idea as of where to look for the root. There is absolutely nothing wrong with such a method, but it is not always easy to plot the function. There are many cases, in which it is rather easy to miss the root, and the situation always gets worse when moving to higher dimensions (i.e., more equations that should simultaneously be solved). Instead, something that is sometimes easier, is to verify that the function $f(x)$ is continuous (which hopefully it is) in which case all that we need is to find a point a in which $f(a) > 0$, and a point b , in which $f(b) < 0$. The continuity will then guarantee (due to the intermediate value theorem) that there exists a point c between a and b for which $f(c) = 0$, and the hunt for that point can then begin. How to find such points a and b ? Again, there really is no general recipe. A combination of intuition, common sense, graphics, thinking, and trial-and-error is typically helpful. We would now like to consider several examples:

Example 2.1

A standard way of attempting to determine if a continuous function has a root in an interval is to try to find a point in which it is positive, and a second point in which it is negative. The intermediate value theorem for continuous functions then guarantees the existence of at least one point for which the function vanishes. To demonstrate this method, consider $f(x) = \sin(x) - x + 0.5$. At $x = 0$, $f(0) = 0.5 > 0$, while at $x = 5$, clearly $f(x)$ must be negative. Hence the intermediate value theorem guarantees the existence of at least one point $x^* \in (0, 5)$ for which $f(x^*) = 0$.

Example 2.2

Consider the problem $e^{-x} = x$, for which we are being asked to determine if a solution exists. One possible way to approach this problem is to define a function $f(x) = e^{-x} - x$, rewrite the problem as $f(x) = 0$, and plot $f(x)$. This is not so bad, but already requires a graphic calculator or a calculus-like analysis of the function $f(x)$ in order to plot it. Instead, it is a reasonable idea to start with the original problem, and plot both functions e^{-x} and x . Clearly, these functions intersect each other, and the intersection is the desirable root. Now, we can return to $f(x)$ and use its continuity (as a difference between continuous functions) to check its sign at a couple of points. For example, at $x = 0$, we have that $f(0) = 1 > 0$, while at $x = 1$, $f(1) = 1/e - 1 < 0$. Hence, due to the intermediate value theorem, there must exist a point x^* in the interval $(0, 1)$ for which $f(x^*) = 0$. At that point x^* we have $e^{-x^*} = x^*$. Note that while the graphical

argument clearly indicates that there exists one and only one solution for the equation, the argument that is based on the intermediate value theorem provides the existence of at least one solution.

A tool that is related to the intermediate value theorem is Brouwer's fixed point theorem:

Theorem 2.3 (Brouwer's Fixed Point Theorem) *Assume that $g(x)$ is continuous on the closed interval $[a, b]$. Assume that the interval $[a, b]$ is mapped to itself by $g(x)$, i.e., for any $x \in [a, b]$, $g(x) \in [a, b]$. Then there exists a point $c \in [a, b]$ such that $g(c) = c$. The point c is a fixed point of $g(x)$.*

The theorem is demonstrated in Figure 2.1. Since the interval $[a, b]$ is mapped to itself, the continuity of $g(x)$ implies that it must intersect the line x in the interval $[a, b]$ at least once. Such intersection points are the desirable fixed points of the function $g(x)$, as guaranteed by Theorem 2.3.

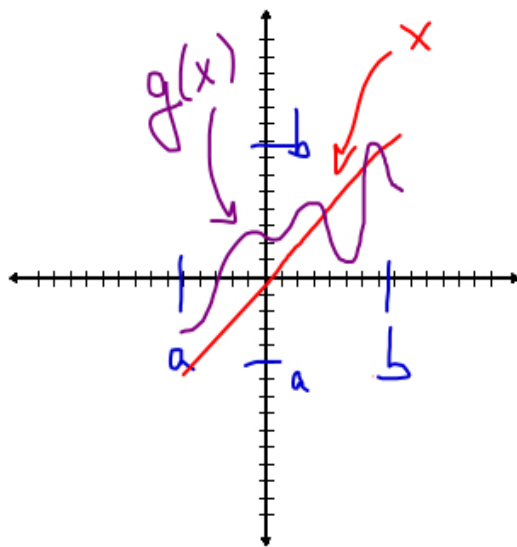


Figure 2.1: An illustration of the Brouwer fixed point theorem

Proof. Let $f(x) = x - g(x)$. Since $g(a) \in [a, b]$ and also $g(b) \in [a, b]$, we know that $f(a) = a - g(a) \leq 0$ while $f(b) = b - g(b) \geq 0$. Since $g(x)$ is continuous in $[a, b]$, so is $f(x)$, and hence according to the intermediate value theorem, there must exist a point $c \in [a, b]$ at which $f(c) = 0$. At this point $g(c) = c$. ■

How much does Theorem 2.3 add in terms of tools for proving that a root exists in a certain interval? In practice, the actual contribution is rather marginal, but there are cases where it adds something. Clearly if we are looking for roots of a function $f(x)$, we can always reformulate the problem as a fixed point problem for a function

$g(x)$ by defining $g(x) = f(x) + x$. Usually this is not the only way in which a root finding problem can be converted into a fixed point problem. In order to be able to use Theorem 2.3, the key point is always to look for a fixed point problem in which the interval of interest is mapped to itself.

Example 2.4

To demonstrate how the fixed point theorem can be used, consider the function $f(x) = e^x - x^2 - 3$ for $x \in [1, 2]$. Define $g(x) = \ln(x^2 + 3)$. Fixed points of $g(x)$ is a root of $f(x)$. Clearly, $g(1) = \ln 4 > \ln e = 1$ and $g(2) = \ln(7) < \ln(e^2) = 2$, and since $g(x)$ is continuous and monotone in $[1, 2]$, we have that $g([1, 2]) \subset [1, 2]$. Hence the conditions of Theorem 2.3 are satisfied and $f(x)$ must have a root in the interval $[1, 2]$.

2.1.2 Examples of root-finding methods

So far our focus has been on attempting to figure out if a given function has any roots, and if it does have roots, approximately where can they be. However, we have not went into any details in developing methods for approximating the values of such roots. Before we start with a detailed study of such methods, we would like to go over a couple of the methods that will be studied later on, emphasizing that they are all iterative methods. The methods that we will briefly describe are Newton's method and the secant method. A more detailed study of these methods will be conducted in the following sections.

1. **Newton's method.** Newton's method for finding a root of a differentiable function $f(x)$ is given by:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (2.1)$$

We note that for the formula (2.1) to be well-defined, we must require that $f'(x_n) \neq 0$ for any x_n . To provide us with a list of successive approximation, Newton's method (2.1) should be supplemented with one initial guess, say x_0 . The equation (2.1) will then provide the values of x_1, x_2, \dots

One way of obtaining Newton's method is the following: Given a point x_n we are looking for the next point x_{n+1} . A linear approximation of $f(x)$ at x_{n+1} is

$$f(x_{n+1}) \approx f(x_n) + (x_{n+1} - x_n)f'(x_n).$$

Since x_{n+1} should be an approximation to the root of $f(x)$, we set $f(x_{n+1}) = 0$, rearrange the terms and get (2.1).

2. **The secant method.** The secant method is obtained by replacing the derivative in Newton's method, $f'(x_n)$, by the following finite difference approximation:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \quad (2.2)$$

The secant method is thus:

$$x_{n+1} = x_n - f(x_n) \left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right]. \quad (2.3)$$

The secant method (2.3) should be supplemented by two initial values, say x_0 , and x_1 . Using these two values, (2.3) will provide the values of x_2, x_3, \dots

2.2 Iterative Methods

At this point we would like to explore more tools for studying iterative methods. We start by considering simple iterates, in which given an initial value x_0 , the iterates are given by the following recursion:

$$x_{n+1} = g(x_n), \quad n = 0, 1, \dots \quad (2.4)$$

If the sequence $\{x_n\}$ in (2.4) converges, and if the function $g(x)$ is continuous, the limit must be a fixed point of the function $g(x)$. This is obvious, since if $x_n \rightarrow x^*$ as $n \rightarrow \infty$, then the continuity of $g(x)$ implies that in the limit we have

$$x^* = g(x^*).$$

Since things seem to work well when the sequence $\{x_n\}$ converges, we are now interested in studying exactly how can the convergence of this sequence be guaranteed? Intuitively, we expect that a convergence of the sequence will occur if the function $g(x)$ is “shrinking” the distance between any two points in a given interval. Formally, such a concept is known as “contraction” and is given by the following definition:

Definition 2.5 Assume that $g(x)$ is a continuous function in $[a, b]$. Then $g(x)$ is a contraction on $[a, b]$ if there exists a constant L such that $0 < L < 1$ for which for any x and y in $[a, b]$:

$$|g(x) - g(y)| \leq L|x - y|. \quad (2.5)$$

The equation (2.5) is referred to as a *Lipschitz condition* and the constant L is the *Lipschitz constant*.

Indeed, if the function $g(x)$ is a contraction, i.e., if it satisfies the Lipschitz condition (2.5), we can expect the iterates (2.4) to converge as given by the Contraction Mapping Theorem.

Theorem 2.6 (Contraction Mapping) Assume that $g(x)$ is a continuous function on $[a, b]$. Assume that $g(x)$ satisfies the Lipschitz condition (2.5), and that $g([a, b]) \subset [a, b]$. Then $g(x)$ has a unique fixed point $c \in [a, b]$. Also, the sequence $\{x_n\}$ defined in (2.4) converges to c as $n \rightarrow \infty$ for any $x_0 \in [a, b]$.

Proof. We know that the function $g(x)$ must have at least one fixed point due to Theorem 2.3. To prove the uniqueness of the fixed point, we assume that there are two fixed points c_1 and c_2 . We will prove that these two points must be identical.

$$|c_1 - c_2| = |g(c_1) - g(c_2)| \leq L|c_1 - c_2|,$$

and since $0 < L < 1$, c_1 must be equal to c_2 .

Finally, we prove that the iterates in (2.4) converge to c for any $x_0 \in [a, b]$.

$$|x_{n+1} - c| = |g(x_n) - g(c)| \leq L|x_n - c| \leq \dots \leq L^{n+1}|x_0 - c|. \quad (2.6)$$

Since $0 < L < 1$, we have that as $n \rightarrow \infty$, $|x_{n+1} - c| \rightarrow 0$, and we have convergence of the iterates to the fixed point of $g(x)$ independently of the starting point x_0 . ■

Remarks.

1. In order to use the Contraction Mapping Theorem, we must verify that the function $g(x)$ satisfies the Lipschitz condition, but what does it mean? The Lipschitz condition provides information about the “slope” of the function. The quotation marks are being used here, because we never required that the function $g(x)$ is differentiable. Our only requirement had to do with the continuity of $g(x)$. The Lipschitz condition can be rewritten as:

$$\frac{|g(x) - g(y)|}{|x - y|} \leq L, \quad \forall x, y \in [a, b], \quad x \neq y,$$

with $0 < L < 1$. The term on the LHS is a discrete approximation to the slope of $g(x)$. In fact, if the function $g(x)$ is differentiable, according to the Mean Value Theorem, there exists a point ξ between x and y such that

$$g'(\xi) = \frac{g(x) - g(y)}{x - y}.$$

Hence, in practice, if the function $g(x)$ is differentiable in the interval (a, b) , and if there exists $L \in (0, 1)$, such that $|g'(x)| < L$ for any $x \in (a, b)$, then the assumptions on $g(x)$ satisfying the Lipschitz condition in Theorem 2.6 hold. Having $g(x)$ differentiable is more than the theorem requires but in many practical cases, we anyhow deal with differentiable g 's so it is straightforward to use the condition that involves the derivative.

2. Another typical thing that can happen is that the function $g(x)$ will be differentiable, and $|g'(x)|$ will be less than 1, but only in a neighborhood of the fixed point. In this case, we can still formulate a “local” version of the contraction mapping theorem. This theorem will guarantee convergence to a fixed point, c , of $g(x)$ if we start the iterations sufficiently close to that point c .

Starting “far” from c may or may not lead to a convergence to c . Also, since we consider only a neighborhood of the fixed point c , we can no longer guarantee the uniqueness of the fixed point, as away from there, we do not post any restriction on the slope of $g(x)$ and therefore anything can happen.

3. When the contraction mapping theorem holds, and convergence of the iterates to the unique fixed point follows, it is of interest to know how many iterations are required in order to approximate the fixed point with a given accuracy. If our goal is to approximate c within a distance ε , then this means that we are looking for n such that

$$|x_n - c| \leq \varepsilon.$$

We know from (2.6) that

$$|x_n - c| \leq L^n |x_0 - c|, \quad n \geq 1. \quad (2.7)$$

In order to get rid of c from the RHS of (2.7), we compute

$$|x_0 - c| = |x_c - x_1 + x_1 - c| \leq |x_0 - x_1| + |x_1 - c| \leq L|x_0 - c| + |x_1 - x_0|.$$

Hence

$$|x_0 - c| \leq \frac{|x_1 - x_0|}{1 - L}.$$

We thus have

$$|x_n - c| \leq \frac{L^n}{1 - L} |x_1 - x_0|,$$

and for $|x_n - c| < \varepsilon$ we require that

$$L^n \leq \frac{\varepsilon(1 - L)}{|x_1 - x_0|},$$

which implies that the number of iterations that will guarantee that the approximation error will be under ε must exceed

$$n \geq \frac{1}{\ln(L)} \cdot \ln \left[\frac{(1 - L)\varepsilon}{|x_1 - x_0|} \right]. \quad (2.8)$$

2.3 The Bisection Method

Before returning to Newton’s method, we would like to present and study a method for finding roots which is one of the most intuitive methods one can easily come up with. The method we will consider is known as the “bisection method” .

We are looking for a root of a function $f(x)$ which we assume is continuous on the interval $[a, b]$. We also assume that it has opposite signs at both edges of the interval, i.e., $f(a)f(b) < 0$. We then know that $f(x)$ has at least one zero in $[a, b]$. Of course $f(x)$ may have more than one zero in the interval. The bisection method is only going to converge to one of the zeros of $f(x)$. There will also be no indication as of how many zeros $f(x)$ has in the interval, and no hints regarding where can we actually hope to find more roots, if indeed there are additional roots.

The first step is to divide the interval into two equal subintervals,

$$c = \frac{a + b}{2}.$$

This generates two subintervals, $[a, c]$ and $[c, b]$, of equal lengths. We want to keep the subinterval that is guaranteed to contain a root. Of course, in the rare event where $f(c) = 0$ we are done. Otherwise, we check if $f(a)f(c) < 0$. If yes, we keep the left subinterval $[a, c]$. If $f(a)f(c) > 0$, we keep the right subinterval $[c, b]$. This procedure repeats until the stopping criterion is satisfied: we fix a small parameter $\varepsilon > 0$ and stop when $|f(c)| < \varepsilon$. To simplify the notation, we denote the successive intervals by $[a_0, b_0]$, $[a_1, b_1], \dots$. The first two iterations in the bisection method are shown in Figure 2.2. Note that in the case that is shown in the figure, the function $f(x)$ has multiple roots but the method converges to only one of them.

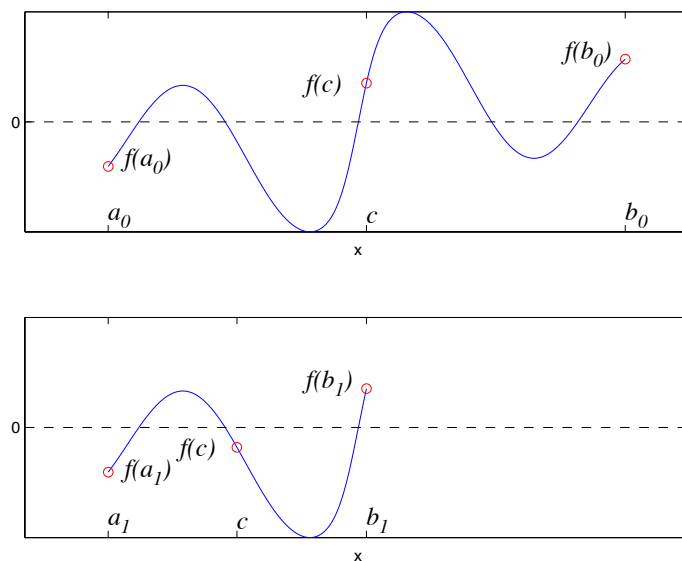


Figure 2.2: The first two iterations in a bisection root-finding method

We would now like to understand if the bisection method always converges to a root. We would also like to figure out how close we are to a root after iterating the algorithm

several times. We first note that

$$a_0 \leq a_1 \leq a_2 \leq \dots \leq b_0,$$

and

$$b_0 \geq b_1 \geq b_2 \geq \dots \geq a_0.$$

We also know that every iteration shrinks the length of the interval by a half, i.e.,

$$b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n), \quad n \geq 0,$$

which means that

$$b_n - a_n = 2^{-n}(b_0 - a_0).$$

The sequences $\{a_n\}_{n \geq 0}$ and $\{b_n\}_{n \geq 0}$ are monotone and bounded, and hence converge. Also

$$\lim_{n \rightarrow \infty} b_n - \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} 2^{-n}(b_0 - a_0) = 0,$$

so that both sequences converge to the same value. We denote that value by r , i.e.,

$$r = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n.$$

Since $f(a_n)f(b_n) < 0$, we know that $(f(r))^2 \leq 0$, which means that $f(r) = 0$, i.e., r is a root of $f(x)$.

We now assume that we stop in the interval $[a_n, b_n]$. This means that $r \in [a_n, b_n]$. Given such an interval, if we have to guess where is the root (which we know is in the interval), it is easy to see that the best estimate for the location of the root is the center of the interval, i.e.,

$$c_n = \frac{a_n + b_n}{2}.$$

In this case, we have

$$|r - c_n| \leq \frac{1}{2}(b_n - a_n) = 2^{-(n+1)}(b_0 - a_0).$$

We summarize this result with the following theorem.

Theorem 2.7 *If $[a_n, b_n]$ is the interval that is obtained in the n^{th} iteration of the bisection method, then the limits $\lim_{n \rightarrow \infty} a_n$ and $\lim_{n \rightarrow \infty} b_n$ exist, and*

$$\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = r,$$

where $f(r) = 0$. In addition, if

$$c_n = \frac{a_n + b_n}{2},$$

then

$$|r - c_n| \leq 2^{-(n+1)}(b_0 - a_0).$$

2.4 Newton's Method

Newton's method is a relatively simple, practical, and widely-used root finding method. It is easy to see that while in some cases the method rapidly converges to a root of the function, in some other cases it may fail to converge at all. This is one reason as of why it is so important not only to understand the construction of the method, but also to understand its limitations.

As always, we assume that $f(x)$ has at least one (real) root, and denote it by r . We start with an initial guess for the location of the root, say x_0 . We then let $l(x)$ be the tangent line to $f(x)$ at x_0 , i.e.,

$$l(x) - f(x_0) = f'(x_0)(x - x_0).$$

The intersection of $l(x)$ with the x -axis serves as the next estimate of the root. We denote this point by x_1 and write

$$0 - f(x_0) = f'(x_0)(x_1 - x_0),$$

which means that

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}. \quad (2.9)$$

In general, **the Newton method** (also known as the Newton-Raphson method) for finding a root is given by iterating (2.9) repeatedly, i.e.,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (2.10)$$

Two sample iterations of the method are shown in Figure 2.3. Starting from a point x_n , we find the next approximation of the root x_{n+1} , from which we find x_{n+2} and so on. In this case, we do converge to the root of $f(x)$.

It is easy to see that Newton's method does not always converge. We demonstrate such a case in Figure 2.4. Here we consider the function $f(x) = \tan^{-1}(x)$ and show what happens if we start with a point which is a fixed point of Newton's method, iterated twice. In this case, $x_0 \approx 1.3917$ is such a point.

In order to analyze the error in Newton's method we let the error in the n^{th} iteration be

$$e_n = x_n - r.$$

We assume that $f''(x)$ is continuous and that $f'(r) \neq 0$, i.e., that r is a simple root of $f(x)$. We will show that the method has a quadratic convergence rate, i.e.,

$$e_{n+1} \approx ce_n^2. \quad (2.11)$$

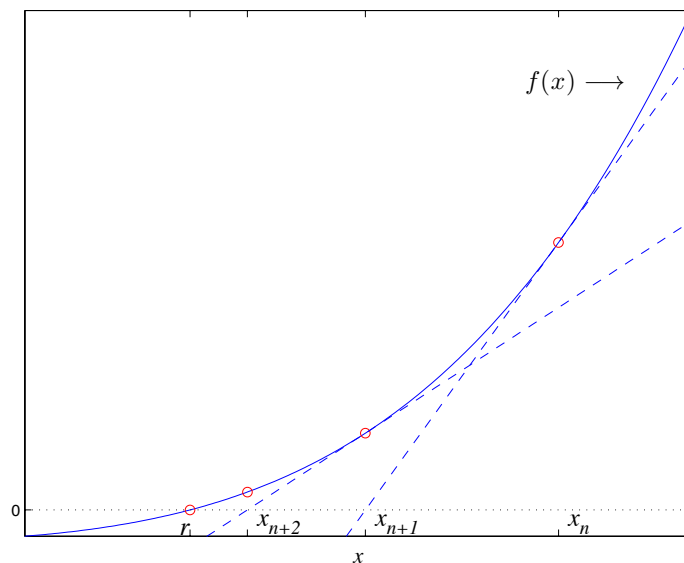


Figure 2.3: Two iterations in Newton's root-finding method. r is the root of $f(x)$ we approach by starting from x_n , computing x_{n+1} , then x_{n+2} , etc.

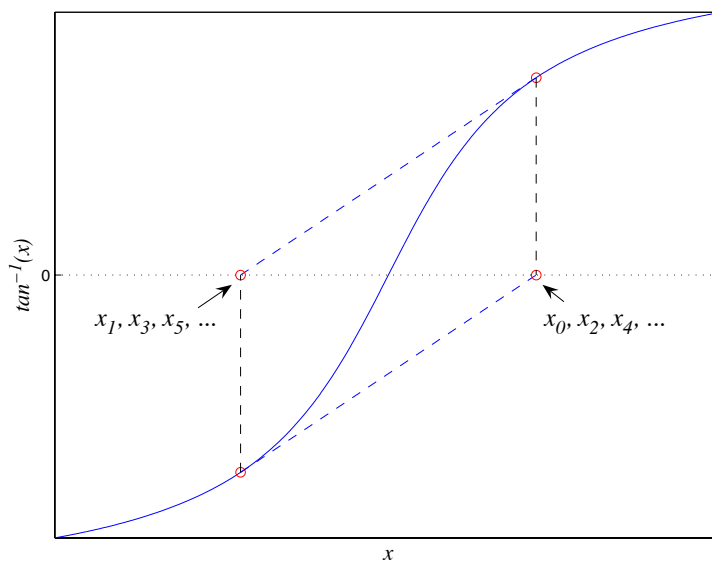


Figure 2.4: Newton's method does not always converge. In this case, the starting point is a fixed point of Newton's method iterated twice

A convergence rate estimate of the type (2.11) makes sense, of course, only if the method converges. Indeed, we will prove the convergence of the method for certain functions $f(x)$, but before we get to the convergence issue, let's derive the estimate (2.11). We rewrite e_{n+1} as

$$e_{n+1} = x_{n+1} - r = x_n - \frac{f(x_n)}{f'(x_n)} - r = e_n - \frac{f(x_n)}{f'(x_n)} = \frac{e_n f'(x_n) - f(x_n)}{f'(x_n)}.$$

Writing a Taylor expansion of $f(r)$ about $x = x_n$ we have

$$0 = f(r) = f(x_n - e_n) = f(x_n) - e_n f'(x_n) + \frac{1}{2} e_n^2 f''(\xi_n),$$

which means that

$$e_n f'(x_n) - f(x_n) = \frac{1}{2} f''(\xi_n) e_n^2.$$

Hence, the relation (2.11), $e_{n+1} \approx c e_n^2$, holds with

$$c = \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} \quad (2.12)$$

Since we assume that the method converges, in the limit as $n \rightarrow \infty$ we can replace (2.12) by

$$c = \frac{1}{2} \frac{f''(r)}{f'(r)}. \quad (2.13)$$

We now return to the issue of convergence and prove that for certain functions Newton's method converges regardless of the starting point.

Theorem 2.8 *Assume that $f(x)$ has two continuous derivatives, is monotonically increasing, convex, and has a zero. Then the zero is unique and Newton's method will converge to it from every starting point.*

Proof. The assumptions on the function $f(x)$ imply that $\forall x, f''(x) > 0$ and $f'(x) > 0$. By (2.12), the error at the $(n+1)^{\text{th}}$ iteration, e_{n+1} , is given by

$$e_{n+1} = \frac{1}{2} \frac{f''(\xi_n)}{f'(x_n)} e_n^2,$$

and hence it is positive, i.e., $e_{n+1} > 0$. This implies that $\forall n \geq 1, x_n > r$. Since $f'(x) > 0$, we have

$$f(x_n) > f(r) = 0.$$

Now, subtracting r from both sides of (2.10) we may write

$$e_{n+1} = e_n - \frac{f(x_n)}{f'(x_n)}, \quad (2.14)$$

which means that $e_{n+1} < e_n$ (and hence $x_{n+1} < x_n$). Hence, both $\{e_n\}_{n \geq 0}$ and $\{x_n\}_{n \geq 0}$ are decreasing and bounded from below. This means that both series converge, i.e., there exists e^* such that,

$$e^* = \lim_{n \rightarrow \infty} e_n,$$

and there exists x^* such that

$$x^* = \lim_{n \rightarrow \infty} x_n.$$

By (2.14) we have

$$e^* = e^* - \frac{f(x^*)}{f'(x^*)},$$

so that $f(x^*) = 0$, and hence $x^* = r$. ■

Theorem 2.8 guarantees global convergence to the unique root of a monotonically increasing, convex smooth function. If we relax some of the requirements on the function, Newton's method may still converge. The price that we will have to pay is that the convergence theorem will no longer be global. Convergence to a root will happen only if we start sufficiently close to it. Such a result is formulated in the following theorem.

Theorem 2.9 *Assume $f(x)$ is a continuous function with a continuous second derivative, that is defined on an interval $I = [r - \delta, r + \delta]$, with $\delta > 0$. Assume that $f(r) = 0$, and that $f''(r) \neq 0$. Assume that there exists a constant A such that*

$$\frac{|f''(x)|}{|f'(y)|} \leq A, \quad \forall x, y \in I.$$

If the initial guess x_0 is sufficiently close to the root r , i.e., if $|r - x_0| \leq \min\{\delta, 1/A\}$, then the sequence $\{x_n\}$ defined in (2.10) converges quadratically to the root r .

Proof. We assume that $x_n \in I$. Since $f(r) = 0$, a Taylor expansion of $f(x)$ at $x = x_n$, evaluated at $x = r$ is:

$$0 = f(r) = f(x_n) + (r - x_n)f'(x_n) + \frac{(r - x_n)^2}{2}f''(\xi_n), \quad (2.15)$$

where ξ_n is between r and x_n , and hence $\xi \in I$. Equation (2.15) implies that

$$r - x_n = \frac{-2f(x_n) - (r - x_n)^2 f''(\xi_n)}{2f'(x_n)}.$$

Since x_{n+1} are the Newton iterates and hence satisfy (2.10), we have

$$r - x_{n+1} = r - x_n + \frac{f(x_n)}{f'(x_n)} = -\frac{(r - x_n)^2 f''(\xi_n)}{2f'(x_n)}. \quad (2.16)$$

Hence

$$|r - x_{n+1}| \leq \frac{(r - x_n)^2}{2} A \leq \frac{|r - x_n|}{2} \leq \dots \leq 2^{-(n-1)} |r - x_0|$$

which implies that $x_n \rightarrow r$ as $n \rightarrow \infty$.

It remains to show that the convergence rate of $\{x_n\}$ to r is quadratic. Since ξ_n is between the root r and x_n , it also converges to r as $n \rightarrow \infty$. The derivatives f' and f'' are continuous and therefore we can take the limit of (2.16) as $n \rightarrow \infty$ and write

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - r|}{|x_n - r|^2} = \left| \frac{f''(r)}{2f'(r)} \right|,$$

which implies the quadratic convergence of $\{x_n\}$ to r . ■

2.5 The Secant Method

We recall that Newton's root finding method is given by equation (2.10), i.e.,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

We now assume that we do not know that the function $f(x)$ is differentiable at x_n , and thus can not use Newton's method as is. Instead, we can replace the derivative $f'(x_n)$ that appears in Newton's method by a difference approximation. A particular choice of such an approximation,

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}},$$

leads to **the secant method** which is given by

$$x_{n+1} = x_n - f(x_n) \left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right], \quad n \geq 1. \quad (2.17)$$

A geometric interpretation of the secant method is shown in Figure 2.5. Given two points, $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$, the line $l(x)$ that connects them satisfies

$$l(x) - f(x_n) = \frac{f(x_{n-1}) - f(x_n)}{x_{n-1} - x_n} (x - x_n).$$

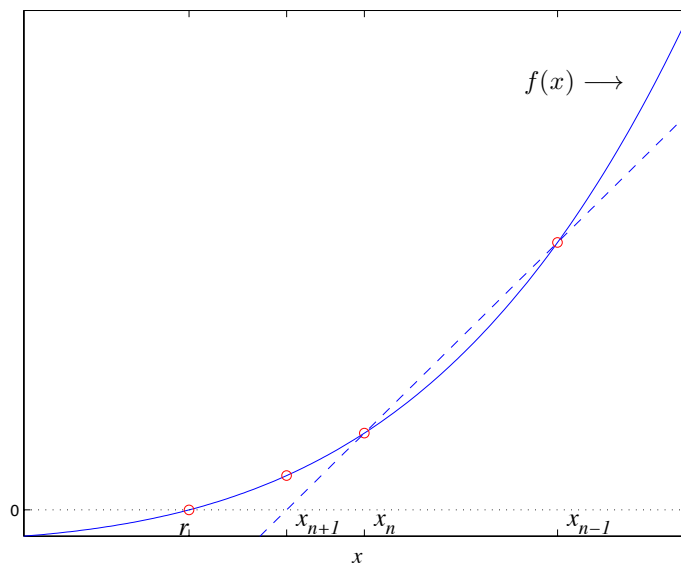


Figure 2.5: The Secant root-finding method. The points x_{n-1} and x_n are used to obtain x_{n+1} , which is the next approximation of the root r

The next approximation of the root, x_{n+1} , is defined as the intersection of $l(x)$ and the x -axis, i.e.,

$$0 - f(x_n) = \frac{f(x_{n-1}) - f(x_n)}{x_{n-1} - x_n}(x_{n+1} - x_n). \quad (2.18)$$

Rearranging the terms in (2.18) we end up with the secant method (2.17).

We note that the secant method (2.17) requires two initial points. While this is an extra requirement compared with, e.g., Newton's method, we note that in the secant method there is no need to evaluate any derivatives. In addition, if implemented properly, every stage requires only one new function evaluation.

We now proceed with an error analysis for the secant method. As usual, we denote the error at the n^{th} iteration by $e_n = x_n - r$. We claim that the rate of convergence of the secant method is **superlinear** (meaning, better than linear but less than quadratic). More precisely, we will show that it is given by

$$|e_{n+1}| \approx |e_n|^\alpha, \quad (2.19)$$

with

$$\alpha = \frac{1 + \sqrt{5}}{2}. \quad (2.20)$$

We start by rewriting e_{n+1} as

$$e_{n+1} = x_{n+1} - r = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})} - r = \frac{f(x_n)e_{n-1} - f(x_{n-1})e_n}{f(x_n) - f(x_{n-1})}.$$

Hence

$$e_{n+1} = e_n e_{n-1} \left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right] \left[\frac{\frac{f(x_n) - f(x_{n-1})}{e_n} - \frac{f(x_{n-1}) - f(x_{n-1})}{e_{n-1}}}{x_n - x_{n-1}} \right]. \quad (2.21)$$

A Taylor expansion of $f(x_n)$ about $x = r$ reads

$$f(x_n) = f(r + e_n) = f(r) + e_n f'(r) + \frac{1}{2} e_n^2 f''(r) + O(e_n^3),$$

and hence

$$\frac{f(x_n)}{e_n} = f'(r) + \frac{1}{2} e_n f''(r) + O(e_n^2).$$

We thus have

$$\begin{aligned} \frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}} &= \frac{1}{2} (e_n - e_{n-1}) f''(r) + O(e_{n-1}^2) + O(e_n^2) \\ &= \frac{1}{2} (x_n - x_{n-1}) f''(r) + O(e_{n-1}^2) + O(e_n^2). \end{aligned}$$

Therefore,

$$\frac{\frac{f(x_n) - f(x_{n-1})}{e_n} - \frac{f(x_{n-1}) - f(x_{n-1})}{e_{n-1}}}{x_n - x_{n-1}} \approx \frac{1}{2} f''(r),$$

and

$$\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \approx \frac{1}{f'(r)}.$$

The error expression (2.21) can be now simplified to

$$e_{n+1} \approx \frac{1}{2} \frac{f''(r)}{f'(r)} e_n e_{n-1} = c e_n e_{n-1}. \quad (2.22)$$

Equation (2.22) expresses the error at iteration $n + 1$ in terms of the errors at iterations n and $n - 1$. In order to turn this into a relation between the error at the $(n + 1)^{\text{th}}$ iteration and the error at the n^{th} iteration, we now assume that the order of convergence is α , i.e.,

$$|e_{n+1}| \sim A |e_n|^\alpha. \quad (2.23)$$

Since (2.23) also means that $|e_n| \sim A|e_{n-1}|^\alpha$, we have

$$A|e_n|^\alpha \sim C|e_n|A^{-\frac{1}{\alpha}}|e_n|^{\frac{1}{\alpha}}.$$

This implies that

$$A^{1+\frac{1}{\alpha}}C^{-1} \sim |e_n|^{1-\alpha+\frac{1}{\alpha}}. \quad (2.24)$$

The left-hand-side of (2.24) is non-zero while the right-hand-side of (2.24) tends to zero as $n \rightarrow \infty$ (assuming, of course, that the method converges). This is possible only if

$$1 - \alpha + \frac{1}{\alpha} = 0,$$

which, in turn, means that

$$\alpha = \frac{1 + \sqrt{5}}{2}.$$

The constant A in (2.23) is thus given by

$$A = C^{\frac{1}{1+\frac{1}{\alpha}}} = C^{\frac{1}{\alpha}} = C^{\alpha-1} = \left[\frac{f''(r)}{2f'(r)} \right]^{\alpha-1}.$$

We summarize this result with the theorem:

Theorem 2.10 *Assume that $f''(x)$ is continuous $\forall x$ in an interval I . Assume that $f(r) = 0$ and that $f'(r) \neq 0$. If x_0, x_1 are sufficiently close to the root r , then $x_n \rightarrow r$. In this case, the convergence is of order $\frac{1+\sqrt{5}}{2}$.*

3 Interpolation

3.1 What is Interpolation?

Imagine that there is an unknown function $f(x)$ for which someone supplies you with its (exact) values at $(n+1)$ distinct points $x_0 < x_1 < \dots < x_n$, i.e., $f(x_0), \dots, f(x_n)$ are given. The interpolation problem is to construct a function $Q(x)$ that passes through these points, i.e., to find a function $Q(x)$ such that the **interpolation requirements**

$$Q(x_j) = f(x_j), \quad 0 \leq j \leq n, \quad (3.1)$$

are satisfied (see Figure 3.1). One easy way of obtaining such a function, is to connect the given points with straight lines. While this is a legitimate solution of the interpolation problem, usually (though not always) we are interested in a different kind of a solution, e.g., a smoother function. We therefore always specify a certain class of functions from which we would like to find one that solves the **interpolation problem**. For example, we may look for a function $Q(x)$ that is a polynomial, $Q(x)$. Alternatively, the function $Q(x)$ can be a trigonometric function or a piecewise-smooth polynomial, and so on.

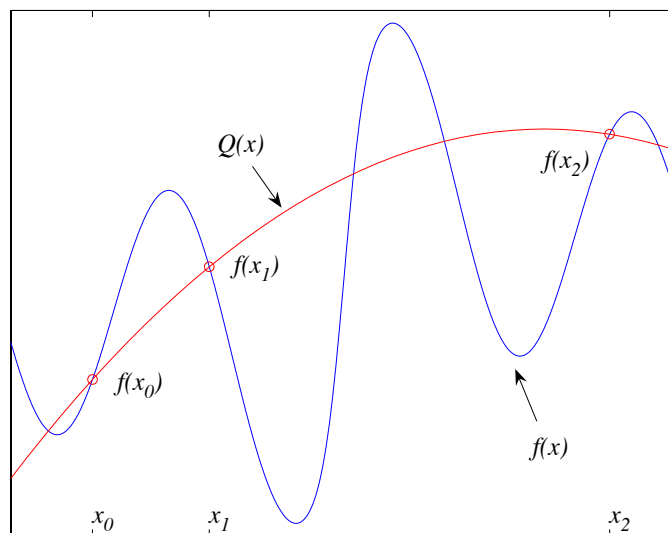


Figure 3.1: The function $f(x)$, the interpolation points x_0, x_1, x_2 , and the interpolating polynomial $Q(x)$

As a simple example let's consider values of a function that are prescribed at two points: $(x_0, f(x_0))$ and $(x_1, f(x_1))$. There are infinitely many functions that pass through these two points. However, if we limit ourselves to polynomials of degree less than or equal to one, there is only one such function that passes through these two points: the

line that connects them. A line, in general, is a polynomial of degree one, but if the two given values are equal, $f(x_0) = f(x_1)$, the line that connects them is the constant $Q_0(x) \equiv f(x_0)$, which is a polynomial of degree zero. This is why we say that there is a unique polynomial of degree ≤ 1 that connects these two points (and not “a polynomial of degree 1”).

The points x_0, \dots, x_n are called the **interpolation points**. The property of “passing through these points” is referred to as **interpolating the data**. The function that interpolates the data is **an interpolant** or **an interpolating polynomial** (or whatever function is being used).

There are cases where the interpolation problem has no solution, e.g., if we look for a linear polynomial that interpolates three points that do not lie on a straight line. When a solution exists, it can be unique (a linear polynomial and two points), or the problem can have more than one solution (a quadratic polynomial and two points). What we are going to study in this section is precisely how to distinguish between these cases. We are also going to present different approaches to constructing the interpolant.

Other than agreeing at the interpolation points, the interpolant $Q(x)$ and the underlying function $f(x)$ are generally different. The **interpolation error** is a measure on how different these two functions are. We will study ways of estimating the interpolation error. We will also discuss strategies on how to minimize this error.

It is important to note that it is possible to formulate the interpolation problem without referring to (or even assuming the existence of) any underlying function $f(x)$. For example, you may have a list of interpolation points x_0, \dots, x_n , and data that is experimentally collected at these points, y_0, y_1, \dots, y_n , which you would like to interpolate. The solution to this interpolation problem is identical to the one where the values are taken from an underlying function.

3.2 The Interpolation Problem

We begin our study with the problem of **polynomial interpolation**: Given $n + 1$ distinct points x_0, \dots, x_n , we seek a polynomial $Q_n(x)$ of the lowest degree such that the following interpolation conditions are satisfied:

$$Q_n(x_j) = f(x_j), \quad j = 0, \dots, n. \quad (3.2)$$

Note that we do not assume any ordering between the points x_0, \dots, x_n , as such an order will make no difference. If we do not limit the degree of the interpolation polynomial it is easy to see that there are infinitely many polynomials that interpolate the data. However, limiting the degree of $Q_n(x)$ to be $\deg(Q_n(x)) \leq n$, singles out precisely one interpolant that will do the job. For example, if $n = 1$, there are infinitely many polynomials that interpolate $(x_0, f(x_0))$ and $(x_1, f(x_1))$. However, there is only one polynomial $Q_n(x)$ with $\deg(Q_n(x)) \leq 1$ that does the job. This result is formally stated in the following theorem:

Theorem 3.1 *If $x_0, \dots, x_n \in \mathbb{R}$ are distinct, then for any $f(x_0), \dots, f(x_n)$ there exists a unique polynomial $Q_n(x)$ of degree $\leq n$ such that the interpolation conditions (3.2) are satisfied.*

Proof. We start with the *existence* part and prove the result by induction. For $n = 0$, $Q_0 = f(x_0)$. Suppose that Q_{n-1} is a polynomial of degree $\leq n - 1$, and suppose also that

$$Q_{n-1}(x_j) = f(x_j), \quad 0 \leq j \leq n - 1.$$

Let us now construct from $Q_{n-1}(x)$ a new polynomial, $Q_n(x)$, in the following way:

$$Q_n(x) = Q_{n-1}(x) + c(x - x_0) \cdot \dots \cdot (x - x_{n-1}). \quad (3.3)$$

The constant c in (3.3) is yet to be determined. Clearly, the construction of $Q_n(x)$ implies that $\deg(Q_n(x)) \leq n$. (Since we might end up with $c = 0$, $Q_n(x)$ could actually be of degree that is less than n .) In addition, the polynomial $Q_n(x)$ satisfies the interpolation requirements $Q_n(x_j) = f(x_j)$ for $0 \leq j \leq n - 1$. All that remains is to determine the constant c in such a way that the last interpolation condition, $Q_n(x_n) = f(x_n)$, is satisfied, i.e.,

$$Q_n(x_n) = Q_{n-1}(x_n) + c(x_n - x_0) \cdot \dots \cdot (x_n - x_{n-1}). \quad (3.4)$$

The condition (3.4) implies that c should be defined as

$$c = \frac{f(x_n) - Q_{n-1}(x_n)}{\prod_{j=0}^{n-1} (x_n - x_j)}, \quad (3.5)$$

and we are done with the proof of existence.

As for *uniqueness*, suppose that there are two polynomials $Q_n(x), P_n(x)$ of degree $\leq n$ that satisfy the interpolation conditions (3.2). Define a polynomial $H_n(x)$ as the difference

$$H_n(x) = Q_n(x) - P_n(x).$$

The degree of $H_n(x)$ is at most n which means that it can have at most n zeros (unless it is identically zero). However, since both $Q_n(x)$ and $P_n(x)$ satisfy all the interpolation requirements (3.2), we have

$$H_n(x_j) = (Q_n - P_n)(x_j) = 0, \quad 0 \leq j \leq n,$$

which means that $H_n(x)$ has $n + 1$ distinct zeros. This contradiction can be resolved only if $H_n(x)$ is the zero polynomial, i.e.,

$$P_n(x) \equiv Q_n(x),$$

and uniqueness is established. ■

3.3 Newton's Form of the Interpolation Polynomial

One good thing about the proof of Theorem 3.1 is that it is constructive. In other words, we can use the proof to write down a formula for the interpolation polynomial. We follow the procedure given by (3.4) for reconstructing the interpolation polynomial. We do it in the following way:

- Let

$$Q_0(x) = a_0,$$

where $a_0 = f(x_0)$.

- Let

$$Q_1(x) = a_0 + a_1(x - x_0).$$

Following (3.5) we have

$$a_1 = \frac{f(x_1) - Q_0(x_1)}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

We note that $Q_1(x)$ is nothing but the straight line connecting the two points $(x_0, f(x_0))$ and $(x_1, f(x_1))$.

In general, let

$$\begin{aligned} Q_n(x) &= a_0 + a_1(x - x_0) + \dots + a_n(x - x_0) \cdot \dots \cdot (x - x_{n-1}) \\ &= a_0 + \sum_{j=1}^n a_j \prod_{k=0}^{j-1} (x - x_k). \end{aligned} \tag{3.6}$$

The coefficients a_j in (3.6) are given by

$$\begin{aligned} a_0 &= f(x_0), \\ a_j &= \frac{f(x_j) - Q_{j-1}(x_j)}{\prod_{k=0}^{j-1} (x_j - x_k)}, \quad 1 \leq j \leq n. \end{aligned} \tag{3.7}$$

We refer to the interpolation polynomial when written in the form (3.6)–(3.7) as **the Newton form of the interpolation polynomial**. As we shall see below, there are various ways of writing the interpolation polynomial. The uniqueness of the interpolation polynomial as guaranteed by Theorem 3.1 implies that we will only be rewriting the same polynomial in different ways.

Example 3.2

The Newton form of the polynomial that interpolates $(x_0, f(x_0))$ and $(x_1, f(x_1))$ is

$$Q_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0).$$

Example 3.3

The Newton form of the polynomial that interpolates the three points $(x_0, f(x_0))$, $(x_1, f(x_1))$, and $(x_2, f(x_2))$ is

$$Q_2(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{f(x_2) - \left[f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0) \right]}{(x_2 - x_0)(x_2 - x_1)}(x - x_0)(x - x_1).$$

3.4 The Interpolation Problem and the Vandermonde Determinant

An alternative approach to the interpolation problem is to consider directly a polynomial of the form

$$Q_n(x) = \sum_{k=0}^n b_k x^k, \quad (3.8)$$

and require that the following interpolation conditions are satisfied

$$Q_n(x_j) = f(x_j), \quad 0 \leq j \leq n. \quad (3.9)$$

In view of Theorem 3.1 we already know that this problem has a unique solution, so we should be able to compute the coefficients of the polynomial directly from (3.8). Indeed, the interpolation conditions, (3.9), imply that the following equations should hold:

$$b_0 + b_1 x_j + \dots + b_n x_j^n = f(x_j), \quad j = 0, \dots, n. \quad (3.10)$$

In matrix form, (3.10) can be rewritten as

$$\begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}. \quad (3.11)$$

In order for the system (3.11) to have a *unique* solution, it has to be nonsingular. This means, e.g., that the determinant of its coefficients matrix must not vanish, i.e.

$$\begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} \neq 0. \quad (3.12)$$

The determinant (3.12), is known as **the Vandermonde determinant**. In Lemma 3.4 we will show that the Vandermonde determinant equals to the product of terms of the form $x_i - x_j$ for $i > j$. Since we assume that the points x_0, \dots, x_n are distinct, the determinant in (3.12) is indeed non zero. Hence, the system (3.11) has a solution that is also unique, which confirms what we already know according to Theorem 3.1.

Lemma 3.4

$$\begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} = \prod_{i>j} (x_i - x_j). \quad (3.13)$$

Proof. We will prove (3.13) by induction. First we verify that the result holds in the 2×2 case. Indeed,

$$\begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} = x_1 - x_0.$$

We now assume that the result holds for $n - 1$ and consider n . We note that the index n corresponds to a matrix of dimensions $(n + 1) \times (n + 1)$, hence our induction assumption is that (3.13) holds for any Vandermonde determinant of dimension $n \times n$. We subtract the first row from all other rows, and expand the determinant along the first column:

$$\begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} = \begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 0 & x_1 - x_0 & \dots & x_1^n - x_0^n \\ \vdots & \vdots & & \vdots \\ 0 & x_n - x_0 & \dots & x_n^n - x_0^n \end{vmatrix} = \begin{vmatrix} x_1 - x_0 & \dots & x_1^n - x_0^n \\ \vdots & & \vdots \\ x_n - x_0 & \dots & x_n^n - x_0^n \end{vmatrix}$$

For every row k we factor out a term $x_k - x_0$:

$$\begin{vmatrix} x_1 - x_0 & \dots & x_1^n - x_0^n \\ \vdots & & \vdots \\ x_n - x_0 & \dots & x_n^n - x_0^n \end{vmatrix} = \prod_{k=1}^n (x_k - x_0) \begin{vmatrix} 1 & x_1 + x_0 & \dots & \sum_{i=0}^{n-1} x_1^{n-1-i} x_0^i \\ 1 & x_2 + x_0 & \dots & \sum_{i=0}^{n-1} x_2^{n-1-i} x_0^i \\ \vdots & \vdots & & \vdots \\ 1 & x_n + x_0 & \dots & \sum_{i=0}^{n-1} x_n^{n-1-i} x_0^i \end{vmatrix}$$

Here, we used the expansion

$$x_1^n - x_0^n = (x_1 - x_0)(x_1^{n-1} + x_1^{n-2}x_0 + x_1^{n-3}x_0^2 + \dots + x_0^{n-1}),$$

for the first row, and similar expansions for all other rows. For every column l , starting from the second one, subtracting the sum of x_0^i times column i (summing only over “previous” columns, i.e., columns i with $i < l$), we end up with

$$\prod_{k=1}^n (x_k - x_0) \begin{vmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^{n-1} \end{vmatrix}. \quad (3.14)$$

Since now we have on the RHS of (3.14) a Vandermonde determinant of dimension $n \times n$, we can use the induction to conclude with the desired result. ■

3.5 The Lagrange Form of the Interpolation Polynomial

The form of the interpolation polynomial that we used in (3.8) assumed a linear combination of polynomials of degrees $0, \dots, n$, in which the coefficients were unknown. In this section we take a different approach and assume that the interpolation polynomial is given as a linear combination of $n + 1$ polynomials of degree n . This time, we set the coefficients as the interpolated values, $\{f(x_j)\}_{j=0}^n$, while the unknowns are the polynomials. We thus let

$$Q_n(x) = \sum_{j=0}^n f(x_j)l_j^n(x), \quad (3.15)$$

where $l_j^n(x)$ are $n+1$ polynomials of degree $\leq n$. We use two indices in these polynomials: the subscript j enumerates $l_j^n(x)$ from 0 to n and the superscript n is used to remind us that the degree of $l_j^n(x)$ is n . Note that in this particular case, the polynomials $l_j^n(x)$ are precisely of degree n (and not $\leq n$). However, $Q_n(x)$, given by (3.15) may have a lower degree. In either case, the degree of $Q_n(x)$ is n at the most. We now require that $Q_n(x)$ satisfies the interpolation conditions

$$Q_n(x_i) = f(x_i), \quad 0 \leq i \leq n. \quad (3.16)$$

By substituting x_i for x in (3.15) we have

$$Q_n(x_i) = \sum_{j=0}^n f(x_j)l_j^n(x_i), \quad 0 \leq i \leq n.$$

In view of (3.16) we may conclude that $l_j^n(x)$ must satisfy

$$l_j^n(x_i) = \delta_{ij}, \quad i, j = 0, \dots, n, \quad (3.17)$$

where δ_{ij} is the Krönecker delta, defined as

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Each polynomial $l_j^n(x)$ has $n + 1$ unknown coefficients. The conditions (3.17) provide exactly $n + 1$ equations that the polynomials $l_j^n(x)$ must satisfy and these equations can be solved in order to determine all $l_j^n(x)$'s. Fortunately there is a shortcut. An obvious way of constructing polynomials $l_j^n(x)$ of degree $\leq n$ that satisfy (3.17) is the following:

$$l_j^n(x) = \frac{(x - x_0) \cdot \dots \cdot (x - x_{j-1})(x - x_{j+1}) \cdot \dots \cdot (x - x_n)}{(x_j - x_0) \cdot \dots \cdot (x_j - x_{j-1})(x_j - x_{j+1}) \cdot \dots \cdot (x_j - x_n)}, \quad 0 \leq j \leq n. \quad (3.18)$$

The uniqueness of the interpolating polynomial of degree $\leq n$ given $n + 1$ distinct interpolation points implies that the polynomials $l_j^n(x)$ given by (3.17) are the only polynomials of degree $\leq n$ that satisfy (3.17).

Note that the denominator in (3.18) does not vanish since we assume that all interpolation points are distinct. **The Lagrange form of the interpolation polynomial** is the polynomial $Q_n(x)$ given by (3.15), where the polynomials $l_j^n(x)$ of degree $\leq n$ are given by (3.18). A compact form of rewriting (3.18) using the product notation is

$$l_j^n(x) = \frac{\prod_{\substack{i=0 \\ i \neq j}}^n (x - x_i)}{\prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i)}, \quad j = 0, \dots, n. \quad (3.19)$$

Example 3.5

We are interested in finding the Lagrange form of the interpolation polynomial that interpolates two points: $(x_0, f(x_0))$ and $(x_1, f(x_1))$. We know that the unique interpolation polynomial through these two points is the line that connects the two points. Such a line can be written in many different forms. In order to obtain the Lagrange form we let

$$l_0^1(x) = \frac{x - x_1}{x_0 - x_1}, \quad l_1^1(x) = \frac{x - x_0}{x_1 - x_0}.$$

The desired polynomial is therefore given by the familiar formula

$$Q_1(x) = f(x_0)l_0^1(x) + f(x_1)l_1^1(x) = f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0}.$$

Example 3.6

This time we are looking for the Lagrange form of the interpolation polynomial, $Q_2(x)$, that interpolates three points: $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$. Unfortunately, the Lagrange form of the interpolation polynomial does not let us use the interpolation polynomial through the first two points, $Q_1(x)$, as a building block for $Q_2(x)$. This means that we have to compute all the polynomials $l_j^2(x)$ from scratch. We start with

$$\begin{aligned} l_0^2(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \\ l_1^2(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \\ l_2^2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \end{aligned}$$

The interpolation polynomial is therefore given by

$$\begin{aligned} Q_2(x) &= f(x_0)l_0^2(x) + f(x_1)l_1^2(x) + f(x_2)l_2^2(x) \\ &= f(x_0)\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}. \end{aligned}$$

It is easy to verify that indeed $Q_2(x_j) = f(x_j)$ for $j = 0, 1, 2$, as desired.

Remarks.

1. One instance where the Lagrange form of the interpolation polynomial may seem to be advantageous when compared with the Newton form is when there is a need to solve several interpolation problems, all given at the same interpolation points x_0, \dots, x_n but with different values $f(x_0), \dots, f(x_n)$. In this case, the polynomials $l_j^n(x)$ are identical for all problems since they depend only on the points but not on the values of the function at these points. Therefore, they have to be constructed only once.
2. An alternative form for $l_j^n(x)$ can be obtained in the following way. Define the polynomials $w_n(x)$ of degree $n + 1$ by

$$w_n(x) = \prod_{i=0}^n (x - x_i).$$

Then its derivative is

$$w_n'(x) = \sum_{j=0}^n \prod_{\substack{i=0 \\ i \neq j}}^n (x - x_i). \quad (3.20)$$

When $w_n'(x)$ is evaluated at an interpolation point, x_j , there is only one term in the sum in (3.20) that does not vanish:

$$w_n'(x_j) = \prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i).$$

Hence, in view of (3.19), $l_j^n(x)$ can be rewritten as

$$l_j^n(x) = \frac{w_n(x)}{(x - x_j)w_n'(x_j)}, \quad 0 \leq j \leq n. \quad (3.21)$$

3. For future reference we note that the coefficient of x^n in the interpolation polynomial $Q_n(x)$ is

$$\sum_{j=0}^n \frac{f(x_j)}{\prod_{\substack{k=0 \\ k \neq j}}^n (x_j - x_k)}. \quad (3.22)$$

For example, the coefficient of x in $Q_1(x)$ in Example 3.5 is

$$\frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}.$$

3.6 Divided Differences

We recall that Newton's form of the interpolation polynomial is given by (see (3.6)–(3.7))

$$Q_n(x) = a_0 + a_1(x - x_0) + \dots + a_n(x - x_0) \cdots (x - x_{n-1}),$$

with $a_0 = f(x_0)$ and

$$a_j = \frac{f(x_j) - Q_{j-1}(x_j)}{\prod_{k=0}^{j-1} (x_j - x_k)}, \quad 1 \leq j \leq n.$$

From now on, we will refer to the coefficient, a_j , as **the j^{th} -order divided difference**.

The j^{th} -order divided difference, a_j , is based on the points x_0, \dots, x_j and on the values of the function at these points $f(x_0), \dots, f(x_j)$. To emphasize this dependence, we use the following notation:

$$a_j = f[x_0, \dots, x_j], \quad 1 \leq j \leq n. \quad (3.23)$$

We also denote the zeroth-order divided difference as

$$a_0 = f[x_0],$$

where

$$f[x_0] = f(x_0).$$

Using the divided differences notation (3.23), the Newton form of the interpolation polynomial becomes

$$Q_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_n] \prod_{k=0}^{n-1} (x - x_k). \quad (3.24)$$

There is a simple recursive way of computing the j^{th} -order divided difference from divided differences of lower order, as shown by the following lemma:

Lemma 3.7 *The divided differences satisfy:*

$$f[x_0, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}. \quad (3.25)$$

Proof. For any k , we denote by $Q_k(x)$, a polynomial of degree $\leq k$, that interpolates $f(x)$ at x_0, \dots, x_k , i.e.,

$$Q_k(x_j) = f(x_j), \quad 0 \leq j \leq k.$$

We now consider the unique polynomial $P(x)$ of degree $\leq n - 1$ that interpolates $f(x)$ at x_1, \dots, x_n , and claim that

$$Q_n(x) = P(x) + \frac{x - x_n}{x_n - x_0} [P(x) - Q_{n-1}(x)]. \quad (3.26)$$

In order to verify this equality, we note that for $i = 1, \dots, n - 1$, $P(x_i) = Q_{n-1}(x_i)$ so that

$$RHS(x_i) = P(x_i) = f(x_i).$$

At x_n , $RHS(x_n) = P(x_n) = f(x_n)$. Finally, at x_0 ,

$$RHS(x_0) = P(x_0) + \frac{x_0 - x_n}{x_n - x_0} [P(x_0) - Q_{n-1}(x_0)] = Q_{n-1}(x_0) = f(x_0).$$

Hence, the RHS of (3.26) interpolates $f(x)$ at the $n + 1$ points x_0, \dots, x_n , which is also true for $Q_n(x)$ due to its definition. Since the RHS and the LHS in equation (3.26) are both polynomials of degree $\leq n$, the uniqueness of the interpolating polynomial (in this case through $n + 1$ points) implies the equality in (3.26).

Once we established the equality in (3.26) we can compare the coefficients of the monomials on both sides of the equation. The coefficient of x^n on the left-hand-side of (3.26) is $f[x_0, \dots, x_n]$. The coefficient of x^{n-1} in $P(x)$ is $f[x_1, \dots, x_n]$ and the coefficient of x^{n-1} in $Q_{n-1}(x)$ is $f[x_0, \dots, x_{n-1}]$. Hence, the coefficient of x^n on the right-hand-side of (3.26) is

$$\frac{1}{x_n - x_0} (f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]),$$

which means that

$$f[x_0, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}. \quad \blacksquare$$

Remark. In some books, instead of defining the divided difference in such a way that they satisfy (3.25), the divided differences are defined by the formula

$$f[x_0, \dots, x_n] = f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}].$$

If this is the case, all our results on divided differences should be adjusted accordingly as to account for the missing factor in the denominator.

Example 3.8

The second-order divided difference is

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}.$$

Hence, the unique polynomial that interpolates $(x_0, f(x_0))$, $(x_1, f(x_1))$, and $(x_2, f(x_2))$ is

$$\begin{aligned} Q_2(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}(x - x_0)(x - x_1). \end{aligned}$$

For example, if we want to find the polynomial of degree ≤ 2 that interpolates $(-1, 9)$, $(0, 5)$, and $(1, 3)$, we have

$$f(-1) = 9,$$

$$f[-1, 0] = \frac{5 - 9}{0 - (-1)} = -4, \quad f[0, 1] = \frac{3 - 5}{1 - 0} = -2,$$

$$f[-1, 0, 1] = \frac{f[0, 1] - f[-1, 0]}{1 - (-1)} = \frac{-2 + 4}{2} = 1.$$

so that

$$Q_2(x) = 9 - 4(x + 1) + (x + 1)x = 5 - 3x + x^2.$$

The relations between the divided differences are schematically portrayed in Table 3.1 (up to third-order). We note that the divided differences that are being used as the coefficients in the interpolation polynomial are those that are located in the top of every column. The recursive structure of the divided differences implies that it is required to compute all the low order coefficients in the table in order to get the high-order ones.

One important property of any divided difference is that it is a symmetric function of its arguments. This means that if we assume that y_0, \dots, y_n is any permutation of x_0, \dots, x_n , then

$$f[y_0, \dots, y_n] = f[x_0, \dots, x_n].$$

This property can be clearly explained by recalling that $f[x_0, \dots, x_n]$ plays the role of the coefficient of x^n in the polynomial that interpolates $f(x)$ at x_0, \dots, x_n . At the same time, $f[y_0, \dots, y_n]$ is the coefficient of x^n at the polynomial that interpolates $f(x)$ at the same points. Since the interpolation polynomial is unique for any given data set, the order of the points does not matter, and hence these two coefficients must be identical.

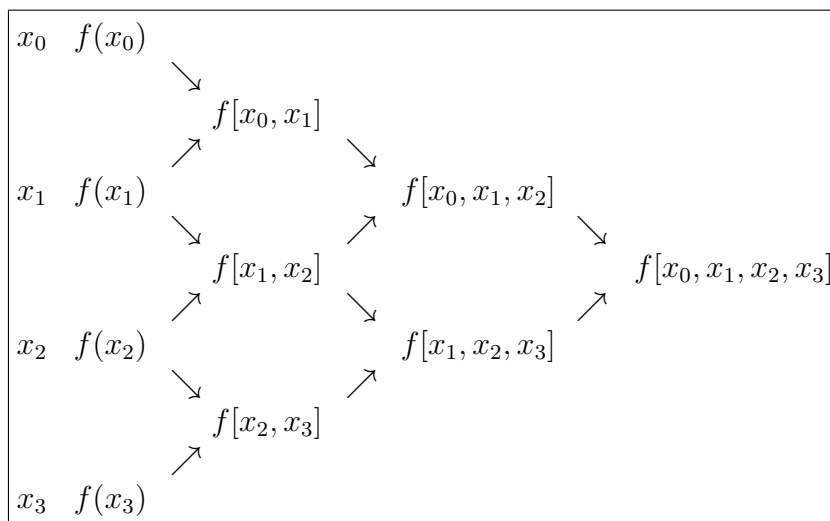


Table 3.1: Divided Differences

3.7 The Error in Polynomial Interpolation

Our goal in this section is to provide estimates on the “error” we make when interpolating data that is taken from sampling an underlying function $f(x)$. While the interpolant and the function agree with each other at the interpolation points, there is, in general, no reason to expect them to be close to each other elsewhere. Nevertheless, we can estimate the difference between them, a difference which we refer to as the **interpolation error**. We let Π_n denote the space of polynomials of degree $\leq n$, and let $C^{n+1}[a, b]$ denote the space of functions that have $n + 1$ continuous derivatives on the interval $[a, b]$.

Theorem 3.9 *Let $f(x) \in C^{n+1}[a, b]$. Let $Q_n(x) \in \Pi_n$ such that it interpolates $f(x)$ at the $n + 1$ distinct points $x_0, \dots, x_n \in [a, b]$. Then $\forall x \in [a, b]$, $\exists \xi_n \in (a, b)$ such that*

$$f(x) - Q_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_n) \prod_{j=0}^n (x - x_j). \quad (3.27)$$

Proof. Fix a point $x \in [a, b]$. If x is one of the interpolation points x_0, \dots, x_n , then the left-hand-side and the right-hand-side of (3.27) are both zero, and the result holds trivially. We therefore assume that $x \neq x_j$ $0 \leq j \leq n$, and let

$$w(x) = \prod_{j=0}^n (x - x_j).$$

We now let

$$F(y) = f(y) - Q_n(y) - \lambda w(y),$$

where λ is chosen as to guarantee that $F(x) = 0$, i.e.,

$$\lambda = \frac{f(x) - Q_n(x)}{w(x)}.$$

Since the interpolation points x_0, \dots, x_n and x are distinct, $w(x)$ does not vanish and λ is well defined. We now note that since $f \in C^{n+1}[a, b]$ and since Q_n and w are polynomials, then also $F \in C^{n+1}[a, b]$. In addition, F vanishes at $n + 2$ points: x_0, \dots, x_n and x . According to Rolle's theorem, F' has at least $n + 1$ distinct zeros in (a, b) , F'' has at least n distinct zeros in (a, b) , and similarly, $F^{(n+1)}$ has at least one zero in (a, b) , which we denote by ξ_n . We have

$$\begin{aligned} 0 &= F^{(n+1)}(\xi_n) = f^{(n+1)}(\xi_n) - Q_n^{(n+1)}(\xi_n) - \lambda(x)w^{(n+1)}(\xi_n) \\ &= f^{(n+1)}(\xi_n) - \frac{f(x) - Q_n(x)}{w(x)}(n + 1)! \end{aligned} \quad (3.28)$$

Here, we used the fact that the leading term of $w(x)$ is x^{n+1} , which guarantees that its $(n + 1)^{\text{th}}$ derivative is

$$w^{(n+1)}(x) = (n + 1)! \quad (3.29)$$

Reordering the terms in (3.28) we conclude with

$$f(x) - Q_n(x) = \frac{1}{(n + 1)!} f^{(n+1)}(\xi_n) w(x). \quad \blacksquare$$

In addition to the interpretation of the divided difference of order n as the coefficient of x^n in some interpolation polynomial, it can also be characterized in another important way. Consider, e.g., the first-order divided difference

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

Since the order of the points does not change the value of the divided difference, we can assume, without any loss of generality, that $x_0 < x_1$. If we assume, in addition, that $f(x)$ is continuously differentiable in the interval $[x_0, x_1]$, then this divided difference equals to the derivative of $f(x)$ at an intermediate point, i.e.,

$$f[x_0, x_1] = f'(\xi), \quad \xi \in (x_0, x_1).$$

In other words, the first-order divided difference can be viewed as an approximation of the first derivative of $f(x)$ in the interval. It is important to note that while this interpretation is based on additional smoothness requirements from $f(x)$ (i.e. its being differentiable), the divided differences are well defined also for non-differentiable functions.

This notion can be extended to divided differences of higher order as stated by the following lemma.

Lemma 3.10 *Let x, x_0, \dots, x_{n-1} be $n + 1$ distinct points. Let $a = \min(x, x_0, \dots, x_{n-1})$ and $b = \max(x, x_0, \dots, x_{n-1})$. Assume that $f(y)$ has a continuous derivative of order n in the interval (a, b) . Then*

$$f[x_0, \dots, x_{n-1}, x] = \frac{f^{(n)}(\xi)}{n!}, \quad (3.30)$$

where $\xi \in (a, b)$.

Proof. Let $Q_n(y)$ interpolate $f(y)$ at x_0, \dots, x_{n-1}, x . Then according to the construction of the Newton form of the interpolation polynomial (3.24), we know that

$$Q_n(y) = Q_{n-1}(y) + f[x_0, \dots, x_{n-1}, x] \prod_{j=0}^{n-1} (y - x_j).$$

Since $Q_n(y)$ interpolated $f(y)$ at x , we have

$$f(x) = Q_{n-1}(x) + f[x_0, \dots, x_{n-1}, x] \prod_{j=0}^{n-1} (x - x_j).$$

By Theorem 3.9 we know that the interpolation error is given by

$$f(x) - Q_{n-1}(x) = \frac{1}{n!} f^{(n)}(\xi_{n-1}) \prod_{j=0}^{n-1} (x - x_j),$$

which implies the result (3.30). ■

Remark. In equation (3.30), we could as well think of the interpolation point x as any other interpolation point, and name it, e.g., x_n . In this case, the equation (3.30) takes the somewhat more natural form of

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

In other words, the n^{th} -order divided difference is an n^{th} -derivative of the function $f(x)$ at an intermediate point, assuming that the function has n continuous derivatives. Similarly to the first-order divided difference, we would like to emphasize that the n^{th} -order divided difference is also well defined in cases where the function is not as smooth as required in the theorem, though if this is the case, we can no longer consider this divided difference to represent a n^{th} -order derivative of the function.

3.8 Interpolation at the Chebyshev Points

In the entire discussion up to now, we assumed that the interpolation points are given. There may be cases where one may have the flexibility of choosing the interpolation points. If this is the case, it would be reasonable to use this degree of freedom to minimize the interpolation error.

We recall that if we are interpolating values of a function $f(x)$ that has n continuous derivatives, the interpolation error is of the form

$$f(x) - Q_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_n) \prod_{j=0}^n (x - x_j). \quad (3.31)$$

Here, $Q_n(x)$ is the interpolating polynomial and ξ_n is an intermediate point in the interval of interest (see (3.27)).

It is important to note that the interpolation points influence two terms on the right-hand-side of (3.31). The obvious one is the product

$$\prod_{j=0}^n (x - x_j). \quad (3.32)$$

The second term that depends on the interpolation points is $f^{(n+1)}(\xi_n)$ since the value of the intermediate point ξ_n depends on $\{x_j\}$. Due to the implicit dependence of ξ_n on the interpolation points, minimizing the interpolation error is not an easy task. We will return to this “full” problem later on in the context of the minimax approximation problem. For the time being, we are going to focus on a simpler problem, namely, how to choose the interpolation points x_0, \dots, x_n such that the product (3.32) is minimized. The solution of this problem is the topic of this section. Once again, we would like to emphasize that a solution of this problem does not (in general) provide an optimal choice of interpolation points that minimizes the interpolation error. All that it guarantees is that the product part of the interpolation error is minimal.

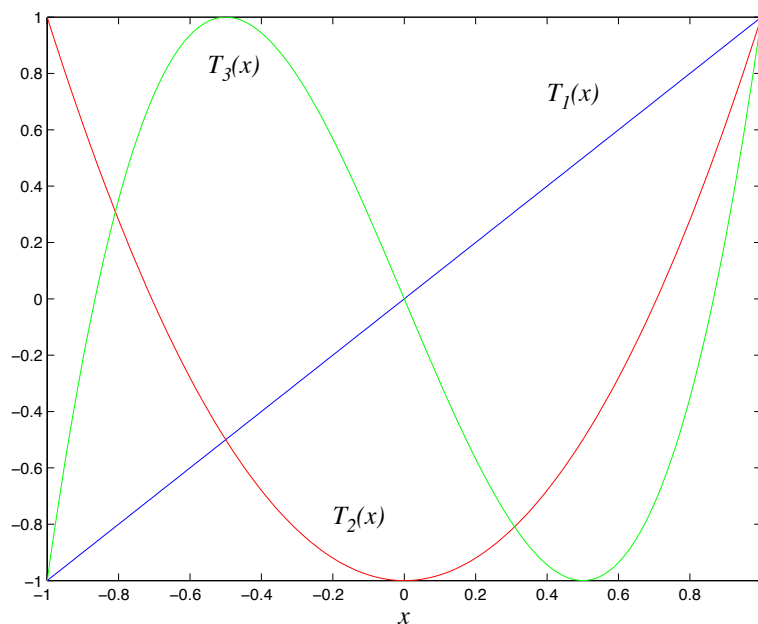
The tool that we are going to use is the Chebyshev polynomials. The solution of the problem will be to choose the interpolation points as the Chebyshev points. We will first introduce the Chebyshev polynomials and the Chebyshev points and then explain why interpolating at these points minimizes (3.32).

The **Chebyshev polynomials** can be defined using the following recursion relation:

$$\begin{cases} T_0(x) = 1, \\ T_1(x) = x, \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1. \end{cases} \quad (3.33)$$

For example, $T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1$, and $T_3(x) = 4x^3 - 3x$. The polynomials $T_1(x)$, $T_2(x)$ and $T_3(x)$ are plotted in Figure 3.2.

Instead of writing the recursion formula, (3.33), it is possible to write an explicit formula for the Chebyshev polynomials:

Figure 3.2: The Chebyshev polynomials $T_1(x)$, $T_2(x)$ and $T_3(x)$

Lemma 3.11 For $x \in [-1, 1]$,

$$T_n(x) = \cos(n \cos^{-1} x), \quad n \geq 0. \quad (3.34)$$

Proof. Standard trigonometric identities imply that

$$\begin{aligned} \cos(n+1)\theta &= \cos\theta \cos n\theta - \sin\theta \sin n\theta, \\ \cos(n-1)\theta &= \cos\theta \cos n\theta + \sin\theta \sin n\theta. \end{aligned}$$

Hence

$$\cos(n+1)\theta = 2 \cos\theta \cos n\theta - \cos(n-1)\theta. \quad (3.35)$$

We now let $\theta = \cos^{-1} x$, i.e., $x = \cos\theta$, and define

$$t_n(x) = \cos(n \cos^{-1} x) = \cos(n\theta).$$

Then by (3.35)

$$\begin{cases} t_0(x) = 1, \\ t_1(x) = x, \\ t_{n+1}(x) = 2xt_n(x) - t_{n-1}(x), \quad n \geq 1. \end{cases}$$

Hence $t_n(x) = T_n(x)$. ■

What is so special about the Chebyshev polynomials, and what is the connection between these polynomials and minimizing the interpolation error? We are about to answer these questions, but before doing so, there is one more issue that we must clarify.

We define a **monic polynomial** as a polynomial for which the coefficient of the leading term is one, i.e., a polynomial of degree n is monic, if it is of the form

$$p_n(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0.$$

Note that Chebyshev polynomials are not monic: the definition (3.33) implies that the Chebyshev polynomial of degree n is of the form

$$T_n(x) = 2^{n-1}x^n + \dots$$

This means that $T_n(x)$ divided by 2^{n-1} is monic, i.e.,

$$2^{1-n}T_n(x) = x^n + \dots$$

A general result about monic polynomials is given by the following theorem

Theorem 3.12 *If $p_n(x)$ is a monic polynomial of degree n , then*

$$\max_{-1 \leq x \leq 1} |p_n(x)| \geq 2^{1-n}. \quad (3.36)$$

Proof. We prove (3.36) by contradiction. Suppose that

$$|p_n(x)| < 2^{1-n}, \quad |x| \leq 1.$$

Let

$$q_n(x) = 2^{1-n}T_n(x),$$

and let x_j be the following $n + 1$ points

$$x_j = \cos\left(\frac{j\pi}{n}\right), \quad 0 \leq j \leq n.$$

Since

$$T_n\left(\cos\frac{j\pi}{n}\right) = (-1)^j,$$

we have

$$(-1)^j q_n(x_j) = 2^{1-n}.$$

Hence

$$(-1)^j p_n(x_j) \leq |p_n(x_j)| < 2^{1-n} = (-1)^j q_n(x_j).$$

This means that

$$(-1)^j (q_n(x_j) - p_n(x_j)) > 0, \quad 0 \leq j \leq n.$$

Hence, the polynomial $(q_n - p_n)(x)$ oscillates $(n + 1)$ times in the interval $[-1, 1]$, which means that $(q_n - p_n)(x)$ has at least n distinct roots in the interval. However, $p_n(x)$ and $q_n(x)$ are both monic polynomials which means that their difference is a polynomial of degree $n - 1$ at most. Such a polynomial cannot have more than $n - 1$ distinct roots, which leads to a contradiction. Note that $p_n - q_n$ cannot be the zero polynomial because that will imply that $p_n(x)$ and $q_n(x)$ are identical which again is not possible due to the assumptions on their maximum values. ■

We are now ready to use Theorem 3.12 to figure out how to reduce the interpolation error. We know by Theorem 3.9 that if the interpolation points $x_0, \dots, x_n \in [-1, 1]$, then there exists $\xi_n \in (-1, 1)$ such that the distance between the function whose values we interpolate, $f(x)$, and the interpolation polynomial, $Q_n(x)$, is

$$\max_{|x| \leq 1} |f(x) - Q_n(x)| \leq \frac{1}{(n+1)!} \max_{|x| \leq 1} |f^{(n+1)}(\xi_n)| \max_{|x| \leq 1} \left| \prod_{j=0}^n (x - x_j) \right|.$$

We are interested in minimizing

$$\max_{|x| \leq 1} \left| \prod_{j=0}^n (x - x_j) \right|.$$

We note that $\prod_{j=0}^n (x - x_j)$ is a monic polynomial of degree $n + 1$ and hence by Theorem 3.12

$$\max_{|x| \leq 1} \left| \prod_{j=0}^n (x - x_j) \right| \geq 2^{-n}.$$

The minimal value of 2^{-n} can be actually obtained if we set

$$2^{-n} T_{n+1}(x) = \prod_{j=0}^n (x - x_j),$$

which is equivalent to choosing x_j as the roots of the Chebyshev polynomial $T_{n+1}(x)$. Here, we have used the obvious fact that $|T_n(x)| \leq 1$.

What are the roots of the Chebyshev polynomial $T_{n+1}(x)$? By Lemma 3.11

$$T_{n+1}(x) = \cos((n+1) \cos^{-1} x).$$

The roots of $T_{n+1}(x)$, x_0, \dots, x_n , are therefore obtained if

$$(n+1) \cos^{-1}(x_j) = \left(j + \frac{1}{2}\right) \pi, \quad 0 \leq j \leq n,$$

i.e., the $(n+1)$ roots of $T_{n+1}(x)$ are

$$x_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \quad 0 \leq j \leq n. \quad (3.37)$$

The roots of the Chebyshev polynomials are sometimes referred to as **the Chebyshev points**. The formula (3.37) for the roots of the Chebyshev polynomial has the following geometrical interpretation. In order to find the roots of $T_n(x)$, define $\alpha = \pi/n$. Divide the upper half of the unit circle into $n+1$ parts such that the two side angles are $\alpha/2$ and the other angles are α . The Chebyshev points are then obtained by projecting these points on the x -axis. This procedure is demonstrated in Figure 3.3 for $T_4(x)$.

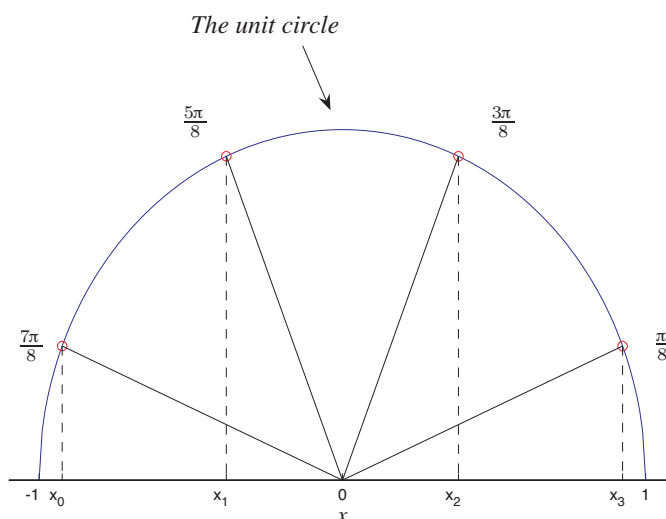


Figure 3.3: The roots of the Chebyshev polynomial $T_4(x)$, x_0, \dots, x_3 . Note that they become dense next to the boundary of the interval

The following theorem summarizes the discussion on interpolation at the Chebyshev points. It also provides an estimate of the error for this case.

Theorem 3.13 Assume that $Q_n(x)$ interpolates $f(x)$ at x_0, \dots, x_n . Assume also that these $(n + 1)$ interpolation points are the $(n + 1)$ roots of the Chebyshev polynomial of degree $n + 1$, $T_{n+1}(x)$, i.e.,

$$x_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \quad 0 \leq j \leq n.$$

Then $\forall |x| \leq 1$,

$$|f(x) - Q_n(x)| \leq \frac{1}{2^n(n+1)!} \max_{|\xi| \leq 1} |f^{(n+1)}(\xi)|. \quad (3.38)$$

Example 3.14

Problem: Let $f(x) = \sin(\pi x)$ in the interval $[-1, 1]$. Find $Q_2(x)$ which interpolates $f(x)$ in the Chebyshev points. Estimate the error.

Solution: Since we are asked to find an interpolation polynomial of degree ≤ 2 , we need 3 interpolation points. We are also asked to interpolate at the Chebyshev points, and hence we first need to compute the 3 roots of the Chebyshev polynomial of degree 3,

$$T_3(x) = 4x^3 - 3x.$$

The roots of $T_3(x)$ can be easily found from $x(4x^2 - 3) = 0$, i.e.,

$$x_0 = -\frac{\sqrt{3}}{2}, \quad x_1 = 0, \quad x_2 = \frac{\sqrt{3}}{2}.$$

The corresponding values of $f(x)$ at these interpolation points are

$$\begin{aligned} f(x_0) &= \sin\left(-\frac{\sqrt{3}}{2}\pi\right) \approx -0.4086, \\ f(x_1) &= 0, \\ f(x_2) &= \sin\left(\frac{\sqrt{3}}{2}\pi\right) \approx 0.4086. \end{aligned}$$

The first-order divided differences are

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \approx 0.4718, \\ f[x_1, x_2] &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} \approx 0.4718, \end{aligned}$$

and the second-order divided difference is

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = 0.$$

The interpolation polynomial is

$$Q_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \approx 0.4718x.$$

The original function $f(x)$ and the interpolant at the Chebyshev points, $Q_2(x)$, are plotted in Figure 3.4.

As of the error estimate, $\forall |x| \leq 1$,

$$|\sin \pi x - Q_2(x)| \leq \frac{1}{2^2 3!} \max_{|\xi| \leq 1} |(\sin \pi t)^{(3)}| \leq \frac{\pi^3}{2^2 3!} \leq 1.292$$

A brief examination of Figure 3.4 reveals that while this error estimate is correct, it is far from being sharp.

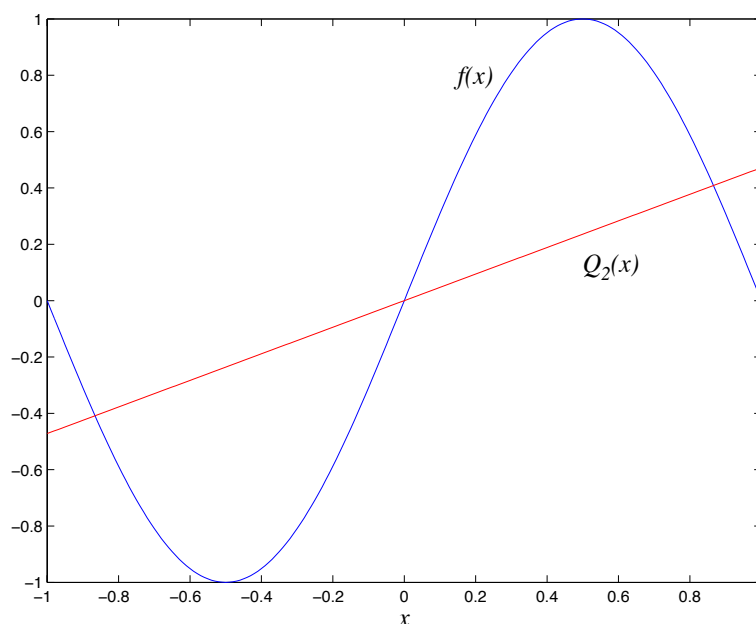


Figure 3.4: The function $f(x) = \sin(\pi(x))$ and the interpolation polynomial $Q_2(x)$ that interpolates $f(x)$ at the Chebyshev points. See Example 3.14.

Remark. In the more general case where the interpolation interval for the function $f(x)$ is $x \in [a, b]$, it is still possible to use the previous results by following the following steps: Start by converting the interpolation interval to $y \in [-1, 1]$:

$$x = \frac{(b - a)y + (a + b)}{2}.$$

This converts the interpolation problem for $f(x)$ on $[a, b]$ into an interpolation problem for $f(x) = g(x(y))$ in $y \in [-1, 1]$. The Chebyshev points in the interval $y \in [-1, 1]$ are

the roots of the Chebyshev polynomial $T_{n+1}(x)$, i.e.,

$$y_j = \cos\left(\frac{2j+1}{2n+2}\pi\right), \quad 0 \leq j \leq n.$$

The corresponding $n+1$ interpolation points in the interval $[a, b]$ are

$$x_j = \frac{(b-a)y_j + (a+b)}{2}, \quad 0 \leq j \leq n.$$

In this case, the product term in the interpolation error is

$$\max_{x \in [a, b]} \left| \prod_{j=0}^n (x - x_j) \right| = \left| \frac{b-a}{2} \right|^{n+1} \max_{|y| \leq 1} \left| \prod_{j=0}^n (x - y_j) \right|,$$

and the interpolation error is given by

$$|f(x) - Q_n(x)| \leq \frac{1}{2^n(n+1)!} \left| \frac{b-a}{2} \right|^{n+1} \max_{\xi \in [a, b]} |f^{(n+1)}(\xi)|. \quad (3.39)$$

3.9 Hermite Interpolation

We now turn to a slightly different interpolation problem in which we assume that in addition to interpolating the values of the function at certain points, we are also interested in interpolating its derivatives. Interpolation that involves the derivatives is called **Hermite interpolation**. Such an interpolation problem is demonstrated in the following example:

Example 3.15

Problem: Find a polynomials $p(x)$ such that $p(1) = -1$, $p'(1) = -1$, and $p(0) = 1$.

Solution: Since three conditions have to be satisfied, we can use these conditions to determine three degrees of freedom, which means that it is reasonable to expect that these conditions uniquely determine a polynomial of degree ≤ 2 . We therefore let

$$p(x) = a_0 + a_1x + a_2x^2.$$

The conditions of the problem then imply that

$$\begin{cases} a_0 + a_1 + a_2 = -1, \\ a_1 + 2a_2 = -1, \\ a_0 = 1. \end{cases}$$

Hence, there is indeed a unique polynomial that satisfies the interpolation conditions and it is

$$p(x) = x^2 - 3x + 1.$$

In general, we may have to interpolate high-order derivatives and not only first-order derivatives. Also, we assume that for any point x_j in which we have to satisfy an interpolation condition of the form

$$p^{(l)}(x_j) = f(x_j),$$

(with $p^{(l)}$ being the l^{th} -order derivative of $p(x)$), we are also given all the values of the lower-order derivatives up to l as part of the interpolation requirements, i.e.,

$$p^{(i)}(x_j) = f^{(i)}(x_j), \quad 0 \leq i \leq l.$$

If this is not the case, it may not be possible to find a unique interpolant as demonstrated in the following example.

Example 3.16

Problem: Find $p(x)$ such that $p'(0) = 1$ and $p'(1) = -1$.

Solution: Since we are asked to interpolate two conditions, we may expect them to uniquely determine a linear function, say

$$p(x) = a_0 + a_1x.$$

However, both conditions specify the derivative of $p(x)$ at two distinct points to be of different values, which amounts to a contradicting information on the value of a_1 . Hence, a linear polynomial cannot interpolate the data and we must consider higher-order polynomials. Unfortunately, a polynomial of order ≥ 2 will no longer be unique because not enough information is given. Note that even if the prescribed values of the derivatives were identical, we will not have problems with the coefficient of the linear term a_1 , but we will still not have enough information to determine the constant a_0 .

A simple case that you are probably already familiar with is the **Taylor series**. When viewed from the point of view that we advocate in this section, one can consider the Taylor series as an interpolation problem in which one has to interpolate the value of the function and its first n derivatives at a given point, say x_0 , i.e., the interpolation conditions are:

$$p^{(j)}(x_0) = f^{(j)}(x_0), \quad 0 \leq j \leq n.$$

The unique solution of this problem in terms of a polynomial of degree $\leq n$ is

$$p(x) = f(x_0) + f'(x_0)(x - x_0) + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n = \sum_{j=0}^n \frac{f^{(j)}(x_0)}{j!}(x - x_0)^j,$$

which is the Taylor series of $f(x)$ expanded about $x = x_0$.

3.9.1 Divided differences with repetitions

We are now ready to consider the Hermite interpolation problem. The first form we study is the Newton form of the Hermite interpolation polynomial. We start by extending the definition of divided differences in such a way that they can handle derivatives. We already know that the first derivative is connected with the first-order divided difference by

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{x \rightarrow x_0} f[x, x_0].$$

Hence, it is natural to extend the notion of divided differences by the following definition.

Definition 3.17 The first-order **divided difference with repetitions** is defined as

$$f[x_0, x_0] = f'(x_0). \quad (3.40)$$

In a similar way, we can extend the notion of divided differences to high-order derivatives:

Definition 3.18 Let $x_0 \leq x_1 \leq \dots \leq x_n$. Then the divided differences satisfy

$$f[x_0, \dots, x_n] = \begin{cases} \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}, & x_n \neq x_0, \\ \frac{f^{(n)}(x_0)}{n!}, & x_n = x_0. \end{cases} \quad (3.41)$$

We now consider the following Hermite interpolation problem: The interpolation points are x_0, \dots, x_l (which we assume are ordered from small to large). At each interpolation point x_j , we have to satisfy the interpolation conditions:

$$p^{(i)}(x_j) = f^{(i)}(x_j), \quad 0 \leq i \leq m_j.$$

Here, m_j denotes the number of derivatives that we have to interpolate for each point x_j (with the standard notation that zero derivatives refers to the value of the function only). In general, the number of derivatives that we have to interpolate may change from point to point. The extended notion of divided differences allows us to write the solution to this problem in the following way:

We let n denote the total number of points including their multiplicities (that correspond to the number of derivatives we have to interpolate at each point), i.e.,

$$n = m_0 + m_1 + \dots + m_l.$$

We then list all the points including their multiplicities (that correspond to the number of derivatives we have to interpolate). To simplify the notations we identify these points with a new ordered list of points y_i :

$$\{y_0, \dots, y_{n-1}\} = \underbrace{\{x_0, \dots, x_0\}}_{m_0}, \underbrace{\{x_1, \dots, x_1\}}_{m_1}, \dots, \underbrace{\{x_l, \dots, x_l\}}_{m_l}.$$

The interpolation polynomial $p_{n-1}(x)$ is given by

$$p_{n-1}(x) = f[y_0] + \sum_{j=1}^{n-1} f[y_0, \dots, y_j] \prod_{k=0}^{j-1} (x - y_k). \quad (3.42)$$

Whenever a point repeats in $f[y_0, \dots, y_j]$, we interpret this divided difference in terms of the extended definition (3.41). In practice, there is no need to shift the notations to y 's and we work directly with the original points. We demonstrate this interpolation procedure in the following example.

Example 3.19

Problem: Find an interpolation polynomial $p(x)$ that satisfies

$$\begin{cases} p(x_0) = f(x_0), \\ p(x_1) = f(x_1), \\ p'(x_1) = f'(x_1). \end{cases}$$

Solution: The interpolation polynomial $p(x)$ is

$$p(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_1](x - x_0)(x - x_1).$$

The divided differences:

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

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

Hence

$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \frac{(x_1 - x_0)f'(x_1) - [f(x_1) - f(x_0)]}{(x_1 - x_0)^2}(x - x_0)(x - x_1).$$

3.9.2 The Lagrange form of the Hermite interpolant

In this section we are interested in writing the Lagrange form of the Hermite interpolant in the special case in which the nodes are x_0, \dots, x_n and the interpolation conditions are

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i), \quad 0 \leq i \leq n. \quad (3.43)$$

We look for an interpolant of the form

$$p(x) = \sum_{i=0}^n f(x_i)A_i(x) + \sum_{i=0}^n f'(x_i)B_i(x). \quad (3.44)$$

In order to satisfy the interpolation conditions (3.43), the polynomials $p(x)$ in (3.44) must satisfy the $2n + 2$ conditions:

$$\begin{cases} A_i(x_j) = \delta_{ij}, & B_i(x_j) = 0, \\ A'_i(x_j) = 0, & B'_i(x_j) = \delta_{ij}, \end{cases} \quad i, j = 0, \dots, n. \quad (3.45)$$

We thus expect to have a unique polynomial $p(x)$ that satisfies the constraints (3.45) assuming that we limit its degree to be $\leq 2n + 1$.

It is convenient to start the construction with the functions we have used in the Lagrange form of the standard interpolation problem (Section 3.5). We already know that

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j},$$

satisfy $l_i(x_j) = \delta_{ij}$. In addition, for $i \neq j$,

$$l_i^2(x_j) = 0, \quad (l_i^2(x_j))' = 0.$$

The degree of $l_i(x)$ is n , which means that the degree of $l_i^2(x)$ is $2n$. We will thus assume that the unknown polynomials $A_i(x)$ and $B_i(x)$ in (3.45) can be written as

$$\begin{cases} A_i(x) = r_i(x)l_i^2(x), \\ B_i(x) = s_i(x)l_i^2(x). \end{cases}$$

The functions $r_i(x)$ and $s_i(x)$ are both assumed to be linear, which implies that $\deg(A_i) = \deg(B_i) = 2n + 1$, as desired. Now, according to (3.45)

$$\delta_{ij} = A_i(x_j) = r_i(x_j)l_i^2(x_j) = r_i(x_j)\delta_{ij}.$$

Hence

$$r_i(x_i) = 1. \quad (3.46)$$

Also,

$$0 = A'_i(x_j) = r'_i(x_j)[l_i(x_j)]^2 + 2r_i(x_j)l_i(x_j)l'_i(x_j) = r'_i(x_j)\delta_{ij} + 2r_i(x_j)\delta_{ij}l'_i(x_j),$$

and thus

$$r'_i(x_i) + 2l'_i(x_i) = 0. \quad (3.47)$$

Assuming that $r_i(x)$ is linear, $r_i(x) = ax + b$, equations (3.46),(3.47), imply that

$$a = -2l'_i(x_i), \quad b = 1 + 2l'_i(x_i)x_i.$$

Therefore

$$A_i(x) = [1 + 2l'_i(x_i)(x_i - x)]l_i^2(x).$$

As of $B_i(x)$ in (3.44), the conditions (3.45) imply that

$$0 = B_i(x_j) = s_i(x_j)l_i^2(x_j) \implies s_i(x_i) = 0, \quad (3.48)$$

and

$$\delta_{ij} = B'_i(x_j) = s'_i(x_j)l_i^2(x_j) + 2s_i(x_j)(l_i^2(x_j))' \implies s'_i(x_i) = 1. \quad (3.49)$$

Combining (3.48) and (3.49), we obtain

$$s_i(x) = x - x_i,$$

so that

$$B_i(x) = (x - x_i)l_i^2(x).$$

To summarize, the Lagrange form of the Hermite interpolation polynomial is given by

$$p(x) = \sum_{i=0}^n f(x_i)[1 + 2l'_i(x_i)(x_i - x)]l_i^2(x) + \sum_{i=0}^n f'(x_i)(x - x_i)l_i^2(x). \quad (3.50)$$

The error in the Hermite interpolation (3.50) is given by the following theorem.

Theorem 3.20 *Let x_0, \dots, x_n be distinct nodes in $[a, b]$ and $f \in C^{2n+2}[a, b]$. If $p \in \Pi_{2n+1}$, such that $\forall 0 \leq i \leq n$,*

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i),$$

then $\forall x \in [a, b]$, there exists $\xi \in (a, b)$ such that

$$f(x) - p(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{i=0}^n (x - x_i)^2. \quad (3.51)$$

Proof. The proof follows the same techniques we used in proving Theorem 3.9. If x is one of the interpolation points, the result trivially holds. We thus fix x as a non-interpolation point and define

$$w(y) = \prod_{i=0}^n (y - x_i)^2.$$

We also have

$$\phi(y) = f(y) - p(y) - \lambda w(y),$$

and select λ such that $\phi(x) = 0$, i.e.,

$$\lambda = \frac{f(x) - p(x)}{w(x)}.$$

ϕ has (at least) $n + 2$ zeros in $[a, b]$: (x, x_0, \dots, x_n) . By Rolle's theorem, we know that ϕ' has (at least) $n + 1$ zeros that are different than (x, x_0, \dots, x_n) . Also, ϕ' vanishes at x_0, \dots, x_n , which means that ϕ' has at least $2n + 2$ zeros in $[a, b]$.

Similarly, Rolle's theorem implies that ϕ'' has at least $2n + 1$ zeros in (a, b) , and by induction, $\phi^{(2n+2)}$ has at least one zero in (a, b) , say ξ .

Hence

$$0 = \phi^{(2n+2)}(\xi) = f^{(2n+2)}(\xi) - p^{(2n+2)}(\xi) - \lambda w^{(2n+2)}(\xi).$$

Since the leading term in $w(y)$ is x^{2n+2} , $w^{(2n+2)}(\xi) = (2n + 2)!$. Also, since $p(x) \in \Pi_{2n+1}$, $p^{(2n+2)}(\xi) = 0$. We recall that x was an arbitrary (non-interpolation) point and hence we have

$$f(x) - p(x) = \frac{f^{(2n+2)}(\xi)}{(2n + 2)!} \prod_{i=0}^n (x - x_i)^2. \quad \blacksquare$$

Example 3.21

Assume that we would like to find the Hermite interpolation polynomial that satisfies:

$$p(x_0) = y_0, \quad p'(x_0) = d_0, \quad p(x_1) = y_1, \quad p'(x_1) = d_1.$$

In this case $n = 1$, and

$$l_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad l'_0(x) = \frac{1}{x_0 - x_1}, \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}, \quad l'_1(x) = \frac{1}{x_1 - x_0}.$$

According to (3.50), the desired polynomial is given by (check!)

$$\begin{aligned} p(x) = & y_0 \left[1 + \frac{2}{x_0 - x_1}(x_0 - x) \right] \left(\frac{x - x_1}{x_0 - x_1} \right)^2 + y_1 \left[1 + \frac{2}{x_1 - x_0}(x_1 - x) \right] \left(\frac{x - x_0}{x_1 - x_0} \right)^2 \\ & + d_0(x - x_0) \left(\frac{x - x_1}{x_0 - x_1} \right)^2 + d_1(x - x_1) \left(\frac{x - x_0}{x_1 - x_0} \right)^2. \end{aligned}$$

3.10 Spline Interpolation

So far, the only type of interpolation we were dealing with was polynomial interpolation. In this section we discuss a different type of interpolation: piecewise-polynomial interpolation. A simple example of such interpolants will be the function we get by connecting data with straight lines (see Figure 3.5). Of course, we would like to generate functions

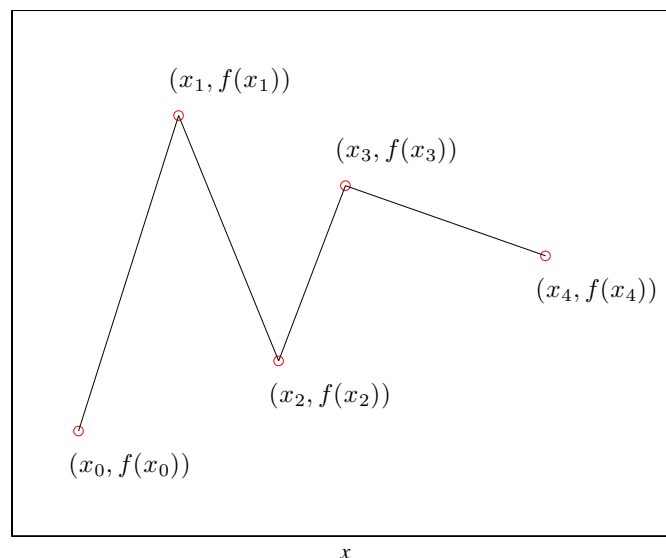


Figure 3.5: A piecewise-linear spline. In every subinterval the function is linear. Overall it is continuous where the regularity is lost at the knots

that are somewhat smoother than piecewise-linear functions, and still interpolate the data. The functions we will discuss in this section are **splines**.

You may still wonder why are we interested in such functions at all? It is easy to motivate this discussion by looking at Figure 3.6. In this figure we demonstrate what a high-order interpolant looks like. Even though the data that we interpolate has only one extrema in the domain, we have no control over the oscillatory nature of the high-order interpolating polynomial. In general, high-order polynomials are oscillatory, which rules them as non-practical for many applications. That is why we focus our attention in this section on splines.

Splines, should be thought of as polynomials on subintervals that are connected in a “smooth way”. We will be more rigorous when we define precisely what we mean by smooth. First, we pick $n + 1$ points which we refer to as the **knots**: $t_0 < t_1 < \dots < t_n$.

A **spline of degree k** having knots t_0, \dots, t_n is a function $s(x)$ that satisfies the following two properties:

1. On $[t_{i-1}, t_i)$ $s(x)$ is a polynomial of degree $\leq k$, i.e., $s(x)$ is a polynomial on every subinterval that is defined by the knots.
2. Smoothness: $s(x)$ has a continuous $(k - 1)^{\text{th}}$ derivative on the interval $[t_0, t_n]$.

A spline of degree 0 is a piecewise-constant function (see Figure 3.7). A spline of

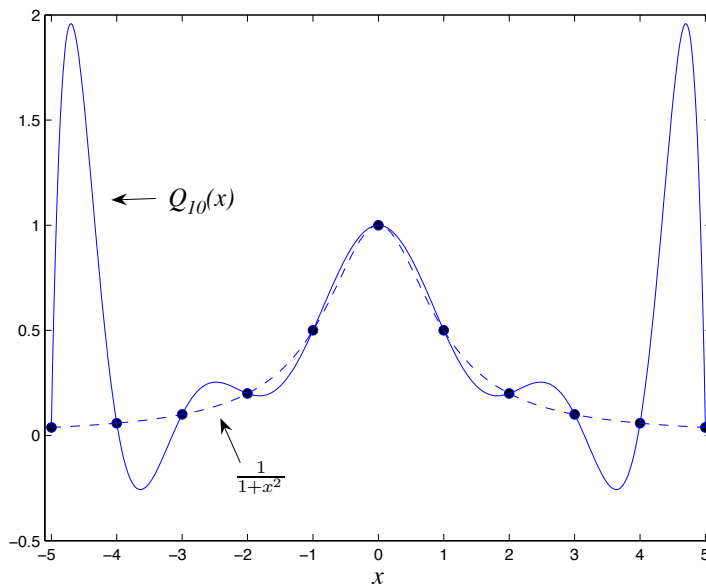


Figure 3.6: An interpolant “goes bad”. In this example we interpolate 11 equally spaced samples of $f(x) = \frac{1}{1+x^2}$ with a polynomial of degree 10, $Q_{10}(x)$

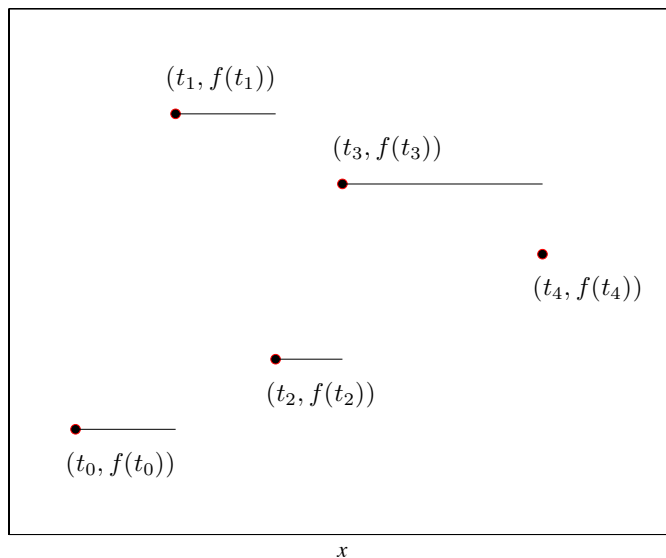


Figure 3.7: A zeroth-order (piecewise-constant) spline. The knots are at the interpolation points. Since the spline is of degree zero, the function is not even continuous

degree 1 is a piecewise-linear function that can be explicitly written as

$$s(x) = \begin{cases} s_0(x) = a_0x + b_0, & x \in [t_0, t_1), \\ s_1(x) = a_1x + b_1, & x \in [t_1, t_2), \\ \vdots & \vdots \\ s_{n-1}(x) = a_{n-1}x + b_{n-1}, & x \in [t_{n-1}, t_n], \end{cases}$$

(see Figure 3.5 where the knots $\{t_i\}$ and the interpolation points $\{x_i\}$ are assumed to be identical). It is now obvious why the points t_0, \dots, t_n are called knots: these are the points that connect the different polynomials with each other. To qualify as an interpolating function, $s(x)$ will have to satisfy interpolation conditions that we will discuss below. We would like to comment already at this point that knots should not be confused with the interpolation points. Sometimes it is convenient to choose the knots to coincide with the interpolation points but this is only optional, and other choices can be made.

3.10.1 Cubic splines

A special case (which is the most common spline function that is used in practice) is the cubic spline. A cubic spline is a spline for which the function is a polynomial of degree ≤ 3 on every subinterval, and a function with two continuous derivatives overall (see Figure 3.8).

Let's denote such a function by $s(x)$, i.e.,

$$s(x) = \begin{cases} s_0(x), & x \in [t_0, t_1), \\ s_1(x), & x \in [t_1, t_2), \\ \vdots & \vdots \\ s_{n-1}(x), & x \in [t_{n-1}, t_n], \end{cases}$$

where $\forall i$, the degree of $s_i(x)$ is ≤ 3 .

We now assume that some data (that $s(x)$ should interpolate) is given at the knots, i.e.,

$$s(t_i) = y_i, \quad 0 \leq i \leq n. \quad (3.52)$$

The interpolation conditions (3.52) in addition to requiring that $s(x)$ is continuous, imply that

$$s_{i-1}(t_i) = y_i = s_i(t_i), \quad 1 \leq i \leq n-1. \quad (3.53)$$

We also require the continuity of the first and the second derivatives, i.e.,

$$\begin{aligned} s'_i(t_{i+1}) &= s'_{i+1}(t_{i+1}), & 0 \leq i \leq n-2, \\ s''_i(t_{i+1}) &= s''_{i+1}(t_{i+1}), & 0 \leq i \leq n-2. \end{aligned} \quad (3.54)$$

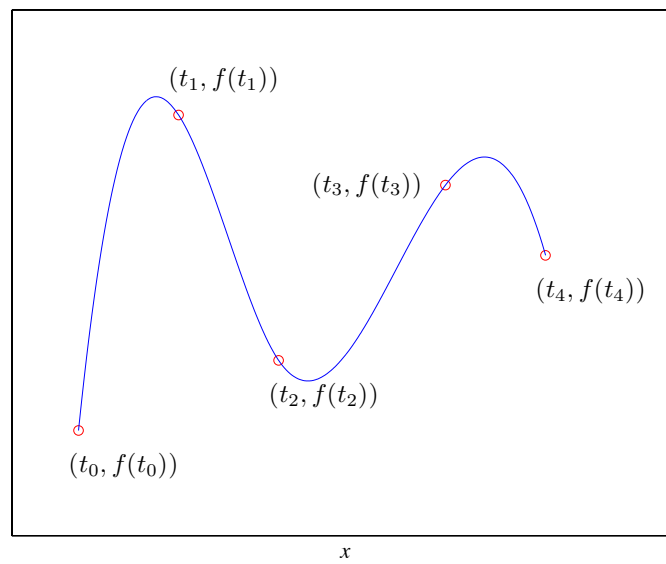


Figure 3.8: A cubic spline. In every subinterval $[t_{i-1}, t_i]$, the function is a polynomial of degree ≤ 2 . The polynomials on the different subintervals are connected to each other in such a way that the spline has a second-order continuous derivative. In this example we use the not-a-knot condition.

Before actually computing the spline, let's check if we have enough equations to determine a unique solution for the problem. There are n subintervals, and in each subinterval we have to determine a polynomial of degree ≤ 3 . Each such polynomial has 4 coefficients, which leaves us with $4n$ coefficients to determine. The interpolation and continuity conditions (3.53) for $s_i(t_i)$ and $s_i(t_{i+1})$ amount to $2n$ equations. The continuity of the first and the second derivatives (3.54) add $2(n-1) = 2n-2$ equations. Altogether we have $4n-2$ equations but $4n$ unknowns which leaves us with 2 degrees of freedom. These indeed are two degrees of freedom that can be determined in various ways as we shall see below.

We are now ready to compute the spline. We will use the following notation:

$$h_i = t_{i+1} - t_i.$$

We also set

$$z_i = s''(t_i).$$

Since the second derivative of a cubic function is linear, we observe that $s_i''(x)$ is the line connecting (t_i, z_i) and (t_{i+1}, z_{i+1}) , i.e.,

$$s_i''(x) = \frac{x - t_i}{h_i} z_{i+1} - \frac{x - t_{i+1}}{h_i} z_i. \quad (3.55)$$

Integrating (3.55) once, we have

$$s_i'(x) = \frac{1}{2}(x - t_i)^2 \frac{z_{i+1}}{h_i} - \frac{1}{2}(x - t_{i+1})^2 \frac{z_i}{h_i} + \tilde{c}.$$

Integrating again

$$s_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + C(x - t_i) + D(t_{i+1} - x).$$

The interpolation condition, $s(t_i) = y_i$, implies that

$$y_i = \frac{z_i}{6h_i} h_i^3 + D h_i,$$

i.e.,

$$D = \frac{y_i}{h_i} - \frac{z_i h_i}{6}.$$

Similarly, $s_i(t_{i+1}) = y_{i+1}$, implies that

$$y_{i+1} = \frac{z_{i+1}}{6h_i} h_i^3 + C h_i,$$

i.e.,

$$C = \frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i.$$

This means that we can rewrite $s_i(x)$ as

$$s_i(x) = \frac{z_{i+1}}{6h_i}(x-t_i)^3 + \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \left(\frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i\right)(x-t_i) + \left(\frac{y_i}{h_i} - \frac{z_i}{6}h_i\right)(t_{i+1}-x).$$

All that remains to determine is the second derivatives of $s(x)$, z_0, \dots, z_n . We can set z_1, \dots, z_{n-1} using the continuity conditions on $s'(x)$, i.e., $s'_i(t_i) = s'_{i-1}(t_i)$. We first compute $s'_i(x)$ and $s'_{i-1}(x)$:

$$\begin{aligned} s'_i(x) &= \frac{z_{i+1}}{2h_i}(x-t_i)^2 - \frac{z_i}{2h_i}(t_{i+1}-x)^2 + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i - \frac{y_i}{h_i} + \frac{z_i h_i}{6}. \\ s'_{i-1}(x) &= \frac{z_i}{2h_{i-1}}(x-t_{i-1})^2 - \frac{z_{i-1}}{2h_{i-1}}(t_i-x)^2 + \frac{y_i}{h_{i-1}} - \frac{z_i}{6}h_{i-1} - \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1}h_{i-1}}{6}. \end{aligned}$$

So that

$$\begin{aligned} s'_i(t_i) &= -\frac{z_i}{2h_i}h_i^2 + \frac{y_{i+1}}{h_i} - \frac{z_{i+1}}{6}h_i - \frac{y_i}{h_i} + \frac{z_i h_i}{6} \\ &= -\frac{h_i}{3}z_i - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i}, \\ s'_{i-1}(t_i) &= \frac{z_i}{2h_{i-1}}h_{i-1}^2 + \frac{y_i}{h_{i-1}} - \frac{z_i}{6}h_{i-1} - \frac{y_{i-1}}{h_{i-1}} + \frac{z_{i-1}h_{i-1}}{6} \\ &= \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i - \frac{y_{i-1}}{h_{i-1}} + \frac{y_i}{h_{i-1}}. \end{aligned}$$

Hence, for $1 \leq i \leq n-1$, we obtain the system of equations

$$\frac{h_{i-1}}{6}z_{i-1} + \frac{h_i + h_{i-1}}{3}z_i + \frac{h_i}{6}z_{i+1} = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{h_{i-1}}(y_i - y_{i-1}). \quad (3.56)$$

These are $n-1$ equations for the $n+1$ unknowns, z_0, \dots, z_n , which means that we have 2 degrees of freedom. Without any additional information about the problem, the only way to proceed is by making an arbitrary choice. There are several standard ways to proceed. One option is to set the end values to zero, i.e.,

$$z_0 = z_n = 0. \quad (3.57)$$

This choice of the second derivative at the end points leads to the so-called, **natural cubic spline**. We will explain later in what sense this spline is “natural”. In this case, we end up with the following linear system of equations

$$\begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & & & & \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & \\ & & \ddots & \ddots & & \\ & & & \frac{h_{n-3}}{6} & \frac{h_{n-3}+h_{n-2}}{3} & \frac{h_{n-2}}{6} \\ & & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{pmatrix} = \begin{pmatrix} \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \vdots \\ \frac{y_{n-1}-y_{n-2}}{h_{n-2}} - \frac{y_{n-2}-y_{n-3}}{h_{n-3}} \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \end{pmatrix}$$

The coefficients matrix is symmetric, tridiagonal, and diagonally dominant (i.e., $|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|, \forall i$), which means that it can always be (efficiently) inverted.

In the special case where the points are equally spaced, i.e., $h_i = h, \forall i$, the system becomes

$$\begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ \vdots \\ y_{n-1} - 2y_{n-2} + y_{n-3} \\ y_n - 2y_{n-1} + y_{n-2} \end{pmatrix} \quad (3.58)$$

In addition to the natural spline (3.57), there are other standard options:

1. If the values of the derivatives at the endpoints are known, one can specify them

$$s'(t_0) = y'_0, \quad s'(t_n) = y'_n.$$

2. The **not-a-knot** condition. Here, we require the third-derivative $s^{(3)}(x)$ to be continuous at the points t_1, t_{n-1} . In this case we end up with a cubic spline with knots $t_0, t_2, t_3, \dots, t_{n-2}, t_n$. The points t_1 and t_{n-1} no longer function as knots. The interpolation requirements are still satisfied at $t_0, t_1, \dots, t_{n-1}, t_n$. Figure 3.9 shows two different cubic splines that interpolate the same initial data. The spline that is plotted with a solid line is the not-a-knot spline. The spline that is plotted with a dashed line is obtained by setting the derivatives at both end-points to zero.

3.10.2 What is natural about the natural spline?

The following theorem states that the natural spline cannot have a larger L^2 -norm of the second-derivative than the function it interpolates (assuming that that function has a continuous second-derivative). In fact, we are minimizing the L^2 -norm of the second-derivative not only with respect to the “original” function which we are interpolating, but with respect to any function that interpolates the data (and has a continuous second-derivative). In that sense, we refer to the natural spline as “natural”.

Theorem 3.22 *Assume that $f''(x)$ is continuous in $[a, b]$, and let $a = t_0 < t_1 < \dots < t_n = b$. If $s(x)$ is the natural cubic spline interpolating $f(x)$ at the knots $\{t_i\}$ then*

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (f''(x))^2 dx.$$

Proof. Define $g(x) = f(x) - s(x)$. Then since $s(x)$ interpolates $f(x)$ at the knots $\{t_i\}$ their difference vanishes at these points, i.e.,

$$g(t_i) = 0, \quad 0 \leq i \leq n.$$

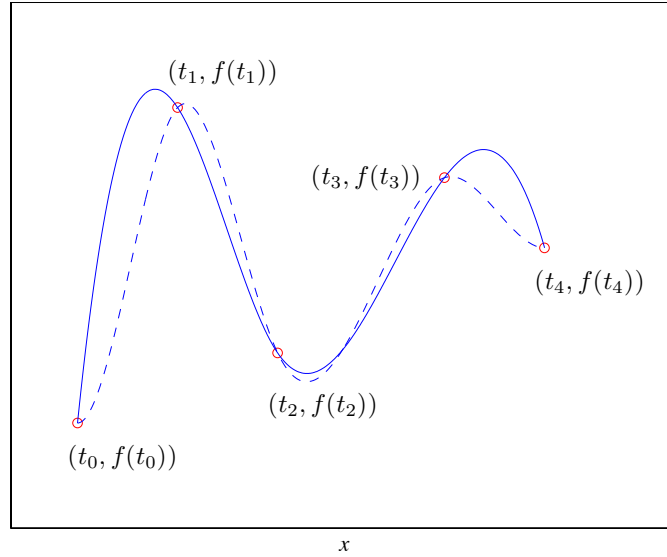


Figure 3.9: Two cubic splines that interpolate the same data. *Solid line*: a not-a-knot spline; *Dashed line*: the derivative is set to zero at both end-points

Now

$$\int_a^b (f'')^2 dx = \int_a^b (s'')^2 dx + \int_a^b (g'')^2 dx + 2 \int_a^b s'' g'' dx. \quad (3.59)$$

We will show that the last term on the right-hand-side of (3.59) is zero, which will conclude the proof as the other two terms on the right-hand-side of (3.59) are non-negative. Splitting that term into a sum of integrals on the subintervals and integrating by parts on every subinterval, we have

$$\int_a^b s'' g'' dx = \sum_{i=1}^n \int_{t_{i-1}}^{t_i} s'' g'' dx = \sum_{i=1}^n \left[(s'' g') \Big|_{t_{i-1}}^{t_i} - \int_{t_{i-1}}^{t_i} s''' g' dx \right].$$

Since we are dealing with the “natural” choice $s''(t_0) = s''(t_n) = 0$, and since $s'''(x)$ is constant on $[t_{i-1}, t_i]$ (say c_i), we end up with

$$\int_a^b s'' g'' dx = - \sum_{i=1}^n \int_{t_{i-1}}^{t_i} s''' g' dx = - \sum_{i=1}^n c_i \int_{t_{i-1}}^{t_i} g' dx = - \sum_{i=1}^n c_i (g(t_i) - g(t_{i-1})) = 0. \quad \blacksquare$$

We note that $f''(x)$ can be viewed as a linear approximation of the curvature

$$\frac{|f''(x)|}{(1 + (f'(x))^2)^{\frac{3}{2}}}.$$

From that point of view, minimizing $\int_a^b (f''(x))^2 dx$, can be viewed as finding the curve with a minimal $|f''(x)|$ over an interval.

4 Approximations

4.1 Background

In this chapter we are interested in approximation problems. Generally speaking, starting from a function $f(x)$ we would like to find a different function $g(x)$ that belongs to a given class of functions and is “close” to $f(x)$ in some sense. As far as the class of functions that $g(x)$ belongs to, we will typically assume that $g(x)$ is a polynomial of a given degree (though it can be a trigonometric function, or any other function). A typical approximation problem, will therefore be: find the “closest” polynomial of degree $\leq n$ to $f(x)$.

What do we mean by “close”? There are different ways of measuring the “distance” between two functions. We will focus on two such measurements (among many): the L^∞ -norm and the L^2 -norm. We chose to focus on these two examples because of the different mathematical techniques that are required to solve the corresponding approximation problems.

We start with several definitions. We recall that a **norm** on a vector space V over \mathbb{R} is a function $\|\cdot\| : V \rightarrow \mathbb{R}$ with the following properties:

1. $\|\lambda f\| = |\lambda| \|f\|$, $\forall \lambda \in \mathbb{R}$ and $\forall f \in V$.
2. $\|f\| \geq 0$, $\forall f \in V$. Also $\|f\| = 0$ iff f is the zero element of V .
3. The triangle inequality: $\|f + g\| \leq \|f\| + \|g\|$, $\forall f, g \in V$.

We assume that the function $f(x) \in C^0[a, b]$ (continuous on $[a, b]$). A continuous function on a closed interval obtains a maximum in the interval. We can therefore define **the L^∞ norm** (also known as **the maximum norm**) of such a function by

$$\|f\|_\infty = \max_{a \leq x \leq b} |f(x)|. \quad (4.1)$$

The L^∞ -distance between two functions $f(x), g(x) \in C^0[a, b]$ is thus given by

$$\|f - g\|_\infty = \max_{a \leq x \leq b} |f(x) - g(x)|. \quad (4.2)$$

We note that the definition of the L^∞ -norm can be extended to functions that are less regular than continuous functions. This generalization requires some subtleties that we would like to avoid in the following discussion, hence, we will limit ourselves to continuous functions.

We proceed by defining **the L^2 -norm** of a continuous function $f(x)$ as

$$\|f\|_2 = \sqrt{\int_a^b |f(x)|^2 dx}. \quad (4.3)$$

The L^2 function space is the collection of functions $f(x)$ for which $\|f\|_2 < \infty$. Of course, we do not have to assume that $f(x)$ is continuous for the definition (4.3) to make sense. However, if we allow $f(x)$ to be discontinuous, we then have to be more rigorous in terms of the definition of the interval so that we end up with a norm (the problem is, e.g., in defining what is the “zero” element in the space). We therefore limit ourselves also in this case to continuous functions only. The L^2 -distance between two functions $f(x)$ and $g(x)$ is

$$\|f - g\|_2 = \sqrt{\int_a^b |f(x) - g(x)|^2 dx}. \quad (4.4)$$

At this point, a natural question is how important is the choice of norm in terms of the solution of the approximation problem. It is easy to see that the value of the norm of a function may vary substantially based on the function as well as the choice of the norm. For example, assume that $\|f\|_\infty < \infty$. Then, clearly

$$\|f\|_2 = \sqrt{\int_a^b |f|^2 dx} \leq (b - a)\|f\|_\infty.$$

On the other hand, it is easy to construct a function with an arbitrary small $\|f\|_2$ and an arbitrarily large $\|f\|_\infty$. Hence, the choice of norm may have a significant impact on the solution of the approximation problem.

As you have probably already anticipated, there is a strong connection between some approximation problems and interpolation problems. For example, one possible method of constructing an approximation to a given function is by sampling it at certain points and then interpolating the sampled data. Is that the best we can do? Sometimes the answer is positive, but the problem still remains difficult because we have to determine the best sampling points. We will address these issues in the following sections.

The following theorem, the Weierstrass approximation theorem, plays a central role in any discussion of approximations of functions. Loosely speaking, this theorem states that any continuous function can be approached as close as we want to with polynomials, assuming that the polynomials can be of any degree. We formulate this theorem in the L^∞ norm and note that a similar theorem holds also in the L^2 sense. We let Π_n denote the space of polynomials of degree $\leq n$.

Theorem 4.1 (Weierstrass Approximation Theorem) *Let $f(x)$ be a continuous function on $[a, b]$. Then there exists a sequence of polynomials $P_n(x)$ that converges uniformly to $f(x)$ on $[a, b]$, i.e., $\forall \varepsilon > 0$, there exists an $N \in \mathbb{N}$ and polynomials $P_n(x) \in \Pi_n$, such that $\forall x \in [a, b]$*

$$|f(x) - P_n(x)| < \varepsilon, \quad \forall n \geq N.$$

We will provide a constructive proof of the Weierstrass approximation theorem: first, we will define a family of polynomials, known as **the Bernstein polynomials**, and then we will show that they uniformly converge to $f(x)$.

We start with the definition. Given a continuous function $f(x)$ in $[0, 1]$, we define the Bernstein polynomials as

$$(B_n f)(x) = \sum_{j=0}^n f\left(\frac{j}{n}\right) \binom{n}{j} x^j (1-x)^{n-j}, \quad 0 \leq x \leq 1.$$

We emphasize that the Bernstein polynomials depend on the function $f(x)$.

Example 4.2

Three Bernstein polynomials $B_6(x)$, $B_{10}(x)$, and $B_{20}(x)$ for the function

$$f(x) = \frac{1}{1 + 10(x - 0.5)^2}$$

on the interval $[0, 1]$ are shown in Figure 4.1. Note the gradual convergence of $B_n(x)$ to $f(x)$.

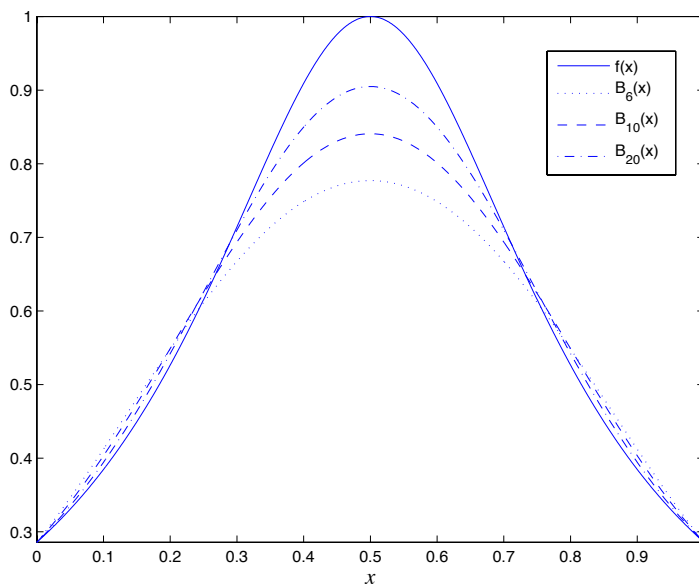


Figure 4.1: The Bernstein polynomials $B_6(x)$, $B_{10}(x)$, and $B_{20}(x)$ for the function $f(x) = \frac{1}{1+10(x-0.5)^2}$ on the interval $[0, 1]$

We now state and prove several properties of $B_n(x)$ that will be used when we prove Theorem 4.1.

Lemma 4.3 *The following relations hold:*

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

Proof.

$$(B_n 1)(x) = \sum_{j=0}^n \binom{n}{j} x^j (1-x)^{n-j} = (x + (1-x))^n = 1.$$

$$\begin{aligned} (B_n x)(x) &= \sum_{j=0}^n \frac{j}{n} \binom{n}{j} x^j (1-x)^{n-j} = x \sum_{j=1}^n \binom{n-1}{j-1} x^{j-1} (1-x)^{n-j} \\ &= x \sum_{j=0}^{n-1} \binom{n-1}{j} x^j (1-x)^{n-1-j} = x[x + (1-x)]^{n-1} = x. \end{aligned}$$

Finally,

$$\begin{aligned} \left(\frac{j}{n}\right)^2 \binom{n}{j} &= \frac{j}{n} \frac{(n-1)!}{(n-j)!(j-1)!} = \frac{n-1}{n} \frac{j-1}{n} \frac{(n-1)!}{(n-j)!(j-1)!} + \frac{1}{n} \frac{(n-1)!}{(n-j)!(j-1)!} \\ &= \frac{n-1}{n} \binom{n-2}{j-2} + \frac{1}{n} \binom{n-1}{j-1}. \end{aligned}$$

Hence

$$\begin{aligned} (B_n x^2)(x) &= \sum_{j=0}^n \left(\frac{j}{n}\right)^2 \binom{n}{j} x^j (1-x)^{n-j} \\ &= \frac{n-1}{n} x^2 \sum_{j=2}^n \binom{n-2}{j-2} x^{j-2} (1-x)^{n-j} + \frac{1}{n} x \sum_{j=1}^n \binom{n-1}{j-1} x^{j-1} (1-x)^{n-j} \\ &= \frac{n-1}{n} x^2 (x + (1-x))^{n-2} + \frac{1}{n} x (x + (1-x))^{n-1} = \frac{n-1}{n} x^2 + \frac{x}{n}. \quad \blacksquare \end{aligned}$$

In the following lemma we state several additional properties of the Bernstein polynomials. The proof is left as an exercise.

Lemma 4.4 *For all functions $f(x), g(x)$ that are continuous in $[0, 1]$, and $\forall \alpha \in \mathbb{R}$*

1. *Linearity.*

$$(B_n(\alpha f + g))(x) = \alpha(B_n f)(x) + (B_n g)(x).$$

2. *Monotonicity.* If $f(x) \leq g(x) \forall x \in [0, 1]$, then

$$(B_n f)(x) \leq (B_n g)(x).$$

Also, if $|f(x)| \leq g(x) \forall x \in [0, 1]$ then

$$|(B_n f)(x)| \leq (B_n g)(x).$$

3. *Positivity.* If $f(x) \geq 0$ then

$$(B_n f)(x) \geq 0.$$

We are now ready to prove the Weierstrass approximation theorem, Theorem 4.1.

Proof. We will prove the theorem in the interval $[0, 1]$. The extension to $[a, b]$ is left as an exercise. Since $f(x)$ is continuous on a closed interval, it is uniformly continuous. Hence $\forall x, y \in [0, 1]$, such that $|x - y| \leq \delta$,

$$|f(x) - f(y)| \leq \varepsilon. \tag{4.5}$$

In addition, since $f(x)$ is continuous on a closed interval, it is also bounded. Let

$$M = \max_{x \in [0, 1]} |f(x)|.$$

Fix any point $a \in [0, 1]$. If $|x - a| \leq \delta$ then (4.5) holds. If $|x - a| > \delta$ then

$$|f(x) - f(a)| \leq 2M \leq 2M \left(\frac{|x - a|}{\delta} \right)^2.$$

(at first sight this seems to be a strange way of upper bounding a function. We will use it later on to our advantage). Combining the estimates for both cases we have

$$|f(x) - f(a)| \leq \varepsilon + \frac{2M}{\delta^2} (x - a)^2.$$

We would now like to estimate the difference between $B_n f$ and f . The linearity of B_n and the property $(B_n 1)(x) = 1$ imply that

$$B_n(f - f(a))(x) = (B_n f)(x) - f(a).$$

Hence using the monotonicity of B_n and the mapping properties of x and x^2 , we have,

$$\begin{aligned} |B_n f(x) - f(a)| &\leq B_n \left(\varepsilon + \frac{2M}{\delta^2} (x - a)^2 \right) = \varepsilon + \frac{2M}{\delta^2} \left(\frac{n-1}{n} x^2 + \frac{x}{n} - 2ax + a^2 \right) \\ &= \varepsilon + \frac{2M}{\delta^2} (x - a)^2 + \frac{2M}{\delta^2} \frac{x - x^2}{n}. \end{aligned}$$

Evaluating at $x = a$ we have (observing that $\max_{a \in [0,1]}(a - a^2) = \frac{1}{4}$)

$$|B_n f(a) - f(a)| \leq \varepsilon + \frac{2M}{\delta^2} \frac{a - a^2}{n} \leq \varepsilon + \frac{M}{2\delta^2 n}. \quad (4.6)$$

The point a was arbitrary so the result (4.6) holds for any point $a \in [0, 1]$. Choosing $N \geq \frac{M}{2\delta^2 \varepsilon}$ we have $\forall n \geq N$,

$$\|B_n f - f\|_\infty \leq \varepsilon + \frac{M}{2\delta^2 N} \leq 2\varepsilon. \quad \blacksquare$$

- Is interpolation a good way of approximating functions in the ∞ -norm? Not necessarily. Discuss Runge's example...

4.2 The Minimax Approximation Problem

We assume that the function $f(x)$ is continuous on $[a, b]$, and assume that $P_n(x)$ is a polynomial of degree $\leq n$. We recall that the L^∞ -distance between $f(x)$ and $P_n(x)$ on the interval $[a, b]$ is given by

$$\|f - P_n\|_\infty = \max_{a \leq x \leq b} |f(x) - P_n(x)|. \quad (4.7)$$

Clearly, we can construct polynomials that will have an arbitrary large distance from $f(x)$. The question we would like to address is how close can we get to $f(x)$ (in the L^∞ sense) with polynomials of a given degree. We define $d_n(f)$ as the infimum of (4.7) over all polynomials of degree $\leq n$, i.e.,

$$d_n(f) = \inf_{P_n \in \Pi_n} \|f - P_n\|_\infty \quad (4.8)$$

The goal is to find a polynomial $P_n^*(x)$ for which the infimum (4.8) is actually obtained, i.e.,

$$d_n(f) = \|f - P_n^*(x)\|_\infty. \quad (4.9)$$

We will refer to a polynomial $P_n^*(x)$ that satisfies (4.9) as a **polynomial of best approximation** or **the minimax polynomial**. The minimal distance in (4.9) will be referred to as **the minimax error**.

The theory we will explore in the following sections will show that the minimax polynomial always exists and is unique. We will also provide a characterization of the minimax polynomial that will allow us to identify it if we actually see it. The general construction of the minimax polynomial will not be addressed in this text as it is relatively technically involved. We will limit ourselves to simple examples.

Example 4.5

We let $f(x)$ be a monotonically increasing and continuous function on the interval $[a, b]$ and are interested in finding the minimax polynomial of degree zero to $f(x)$ in that interval. We denote this minimax polynomial by

$$P_0^*(x) \equiv c.$$

Clearly, the smallest distance between $f(x)$ and P_0^* in the L^∞ -norm will be obtained if

$$c = \frac{f(a) + f(b)}{2}.$$

The maximal distance between $f(x)$ and P_0^* will be attained at both edges and will be equal to

$$\pm \frac{f(b) - f(a)}{2}.$$

4.2.1 Existence of the minimax polynomial

The existence of the minimax polynomial is provided by the following theorem.

Theorem 4.6 (Existence) *Let $f \in C^0[a, b]$. Then for any $n \in \mathbb{N}$ there exists $P_n^*(x) \in \Pi_n$, that minimizes $\|f(x) - P_n(x)\|_\infty$ among all polynomials $P(x) \in \Pi_n$.*

Proof. We follow the proof as given in [7]. Let $\eta = (\eta_0, \dots, \eta_n)$ be an arbitrary point in \mathbb{R}^{n+1} and let

$$P_n(x) = \sum_{i=0}^n \eta_i x^i \in \Pi_n.$$

We also let

$$\phi(\eta) = \phi(\eta_0, \dots, \eta_n) = \|f - P_n\|_\infty.$$

Our goal is to show that ϕ obtains a minimum in \mathbb{R}^{n+1} , i.e., that there exists a point $\eta^* = (\eta_0^*, \dots, \eta_n^*)$ such that

$$\phi(\eta^*) = \min_{\eta \in \mathbb{R}^{n+1}} \phi(\eta).$$

Step 1. We first show that $\phi(\eta)$ is a continuous function on \mathbb{R}^{n+1} . For an arbitrary $\delta = (\delta_0, \dots, \delta_n) \in \mathbb{R}^{n+1}$, define

$$q_n(x) = \sum_{i=0}^n \delta_i x^i.$$

Then

$$\phi(\eta + \delta) = \|f - (P_n + q_n)\|_\infty \leq \|f - P_n\|_\infty + \|q_n\|_\infty = \phi(\eta) + \|q_n\|_\infty.$$

Hence

$$\phi(\eta + \delta) - \phi(\eta) \leq \|q_n\|_\infty \leq \max_{x \in [a, b]} (|\delta_0| + |\delta_1||x| + \dots + |\delta_n||x|^n).$$

For any $\varepsilon > 0$, let $\tilde{\delta} = \varepsilon / (1 + c + \dots + c^n)$, where $c = \max(|a|, |b|)$. Then for any $\delta = (\delta_0, \dots, \delta_n)$ such that $\max |\delta_i| \leq \tilde{\delta}$, $0 \leq i \leq n$,

$$\phi(\eta + \delta) - \phi(\eta) \leq \varepsilon. \tag{4.10}$$

Similarly

$$\phi(\eta) = \|f - P_n\|_\infty = \|f - (P_n + q_n) + q_n\|_\infty \leq \|f - (P_n + q_n)\|_\infty + \|q_n\|_\infty = \phi(\eta + \delta) + \|q_n\|_\infty,$$

which implies that under the same conditions as in (4.10) we also get

$$\phi(\eta) - \phi(\eta + \delta) \leq \varepsilon,$$

Altogether,

$$|\phi(\eta + \delta) - \phi(\eta)| \leq \varepsilon,$$

which means that ϕ is continuous at η . Since η was an arbitrary point in \mathbb{R}^{n+1} , ϕ is continuous in the entire \mathbb{R}^{n+1} .

Step 2. We now construct a compact set in \mathbb{R}^{n+1} on which ϕ obtains a minimum. We let

$$S = \{\eta \in \mathbb{R}^{n+1} \mid \phi(\eta) \leq \|f\|_\infty\}.$$

We have

$$\phi(0) = \|f\|_\infty,$$

hence, $0 \in S$, and the set S is nonempty. We also note that the set S is bounded and closed (check!). Since ϕ is continuous on the entire \mathbb{R}^{n+1} , it is also continuous on S , and hence it must obtain a minimum on S , say at $\eta^* \in \mathbb{R}^{n+1}$, i.e.,

$$\min_{\eta \in S} \phi(\eta) = \phi(\eta^*).$$

Step 3. Since $0 \in S$, we know that

$$\min_{\eta \in S} \phi(\eta) \leq \phi(0) = \|f\|_\infty.$$

Hence, if $\eta \in \mathbb{R}^{n+1}$ but $\eta \notin S$ then

$$\phi(\eta) > \|f\|_\infty \geq \min_{\eta \in S} \phi(\eta).$$

This means that the minimum of ϕ over S is the same as the minimum over the entire \mathbb{R}^{n+1} . Therefore

$$P_n^*(x) = \sum_{i=0}^n \eta_i^* x^i, \quad (4.11)$$

is the best approximation of $f(x)$ in the L^∞ norm on $[a, b]$, i.e., it is the minimax polynomial, and hence the minimax polynomial exists. ■

We note that the proof of Theorem 4.6 is not a constructive proof. The proof does not tell us what the point η^* is, and hence, we do not know the coefficients of the minimax polynomial as written in (4.11). We will discuss the characterization of the minimax polynomial and some simple cases of its construction in the following sections.

4.2.2 Bounds on the minimax error

It is trivial to obtain an upper bound on the minimax error, since by the definition of $d_n(f)$ in (4.8) we have

$$d_n(f) \leq \|f - P_n\|_\infty, \quad \forall P_n(x) \in \Pi_n.$$

A lower bound is provided by the following theorem.

Theorem 4.7 (de la Vallée-Poussin) *Let $a \leq x_0 < x_1 < \dots < x_{n+1} \leq b$. Let $P_n(x)$ be a polynomial of degree $\leq n$. Suppose that*

$$f(x_j) - P_n(x_j) = (-1)^j e_j, \quad j = 0, \dots, n+1,$$

where all $e_j \neq 0$ and are of an identical sign. Then

$$\min_j |e_j| \leq d_n(f).$$

Proof. By contradiction. Assume for some $Q_n(x)$ that

$$\|f - Q_n\|_\infty < \min_j |e_j|.$$

Then the polynomial

$$(Q_n - P_n)(x) = (f - P_n)(x) - (f - Q_n)(x),$$

is a polynomial of degree $\leq n$ that has the same sign at x_j as does $f(x) - P_n(x)$. This implies that $(Q_n - P_n)(x)$ changes sign at least $n + 2$ times, and hence it has at least $n + 1$ zeros. Being a polynomial of degree $\leq n$ this is possible only if it is identically zero, i.e., if $P_n(x) \equiv Q_n(x)$, which contradicts the assumptions on $Q_n(x)$ and $P_n(x)$. ■

4.2.3 Characterization of the minimax polynomial

The following theorem provides a characterization of the minimax polynomial in terms of its oscillations property.

Theorem 4.8 (The oscillating theorem) *Suppose that $f(x)$ is continuous in $[a, b]$. The polynomial $P_n^*(x) \in \Pi_n$ is the minimax polynomial of degree n to $f(x)$ in $[a, b]$ if and only if $f(x) - P_n^*(x)$ assumes the values $\pm \|f - P_n^*\|_\infty$ with an alternating change of sign at least $n + 2$ times in $[a, b]$.*

Proof. We prove here only the *sufficiency* part of the theorem. For the *necessary* part of the theorem we refer to [7].

Without loss of generality, suppose that

$$(f - P_n^*)(x_i) = (-1)^i \|f - P_n^*\|_\infty, \quad 0 \leq i \leq n + 1.$$

Let

$$D^* = \|f - P_n^*\|_\infty,$$

and let

$$d_n(f) = \min_{P_n \in \Pi_n} \|f - P_n\|_\infty.$$

We replace the infimum in the original definition of $d_n(f)$ by a minimum because we already know that a minimum exists. de la Vallée-Poussin's theorem (Theorem 4.7) implies that $D^* \leq d_n$. On the other hand, the definition of d_n implies that $d_n \leq D^*$. Hence $D^* = d_n$ and $P_n^*(x)$ is the minimax polynomial. ■

Remark. In view of these theorems it is obvious why the Taylor expansion is a poor uniform approximation. The sum is non oscillatory.

4.2.4 Uniqueness of the minimax polynomial

Theorem 4.9 (Uniqueness) *Let $f(x)$ be continuous on $[a, b]$. Then its minimax polynomial $P_n^*(x) \in \Pi_n$ is unique.*

Proof. Let

$$d_n(f) = \min_{P_n \in \Pi_n} \|f - P_n\|_\infty.$$

Assume that $Q_n(x)$ is also a minimax polynomial. Then

$$\|f - P_n^*\|_\infty = \|f - Q_n\|_\infty = d_n(f).$$

The triangle inequality implies that

$$\|f - \frac{1}{2}(P_n^* + Q_n)\|_\infty \leq \frac{1}{2}\|f - P_n^*\|_\infty + \frac{1}{2}\|f - Q_n\|_\infty = d_n(f).$$

Hence, $\frac{1}{2}(P_n^* + Q_n) \in \Pi_n$ is also a minimax polynomial. The oscillating theorem (Theorem 4.8) implies that there exist $x_0, \dots, x_{n+1} \in [a, b]$ such that

$$|f(x_i) - \frac{1}{2}(P_n^*(x_i) + Q_n(x_i))| = d_n(f), \quad 0 \leq i \leq n+1. \quad (4.12)$$

Equation (4.12) can be rewritten as

$$|f(x_i) - P_n^*(x_i) + f(x_i) - Q_n(x_i)| = 2d_n(f), \quad 0 \leq i \leq n+1. \quad (4.13)$$

Since $P_n^*(x)$ and $Q_n(x)$ are both minimax polynomials, we have

$$|f(x_i) - P_n^*(x_i)| \leq \|f - P_n^*\|_\infty = d_n(f), \quad 0 \leq i \leq n+1. \quad (4.14)$$

and

$$|f(x_i) - Q_n(x_i)| \leq \|f - Q_n\|_\infty = d_n(f), \quad 0 \leq i \leq n+1. \quad (4.15)$$

For any i , equations (4.13)–(4.15) mean that the absolute value of two numbers that are $\leq d_n(f)$ add up to $2d_n(f)$. This is possible only if they are equal to each other, i.e.,

$$f(x_i) - P_n^*(x_i) = f(x_i) - Q_n(x_i), \quad 0 \leq i \leq n+1,$$

i.e.,

$$(P_n^* - Q_n)(x_i) = 0, \quad 0 \leq i \leq n+1.$$

Hence, the polynomial $(P_n^* - Q_n)(x) \in \Pi_n$ has $n+2$ distinct roots which is possible for a polynomial of degree $\leq n$ only if it is identically zero. Hence

$$Q_n(x) \equiv P_n^*(x),$$

and the uniqueness of the minimax polynomial is established. ■

4.2.5 The near-minimax polynomial

We now connect between the minimax approximation problem and polynomial interpolation. In order for $f(x) - P_n(x)$ to change its sign $n+2$ times, there should be $n+1$ points on which $f(x)$ and $P_n(x)$ agree with each other. In other words, we can think of $P_n(x)$ as a function that interpolates $f(x)$ at (least in) $n+1$ points, say x_0, \dots, x_n . What can we say about these points?

We recall that the interpolation error is given by (3.27),

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^n (x - x_i).$$

If $P_n(x)$ is indeed the minimax polynomial, we know that the maximum of

$$f^{(n+1)}(\xi) \prod_{i=0}^n (x - x_i), \tag{4.16}$$

will oscillate with equal values. Due to the dependency of $f^{(n+1)}(\xi)$ on the intermediate point ξ , we know that minimizing the error term (4.16) is a difficult task. We recall that interpolation at the Chebyshev points minimizes the multiplicative part of the error term, i.e.,

$$\prod_{i=0}^n (x - x_i).$$

Hence, choosing x_0, \dots, x_n to be the Chebyshev points will not result with the minimax polynomial, but nevertheless, this relation motivates us to refer to the interpolant at the Chebyshev points as the **near-minimax** polynomial. We note that the term “near-minimax” does not mean that the near-minimax polynomial is actually close to the minimax polynomial.

4.2.6 Construction of the minimax polynomial

The characterization of the minimax polynomial in terms of the number of points in which the maximum distance should be obtained with oscillating signs allows us to construct the minimax polynomial in simple cases by a direct computation.

We are not going to deal with the construction of the minimax polynomial in the general case. The algorithm for doing so is known as the Remez algorithm, and we refer the interested reader to [2] and the references therein.

A simple case where we can demonstrate a direct construction of the polynomial is when the function is convex, as done in the following example.

Example 4.10

Problem: Let $f(x) = e^x$, $x \in [1, 3]$. Find the minimax polynomial of degree ≤ 1 , $P_1^*(x)$.

Solution: Based on the characterization of the minimax polynomial, we will be looking for a linear function $P_1^*(x)$ such that its maximal distance between $P_1^*(x)$ and $f(x)$ is obtained 3 times with alternative signs. Clearly, in the case of the present problem, since the function is convex, the maximal distance will be obtained at both edges and at one interior point. We will use this observation in the construction that follows. The construction itself is graphically shown in Figure 4.2.

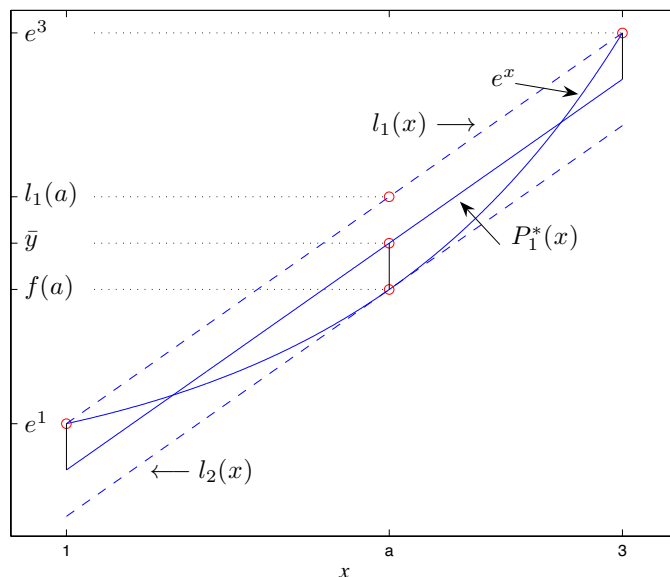


Figure 4.2: A construction of the linear minimax polynomial for the convex function e^x on $[1, 3]$

We let $l_1(x)$ denote the line that connects the endpoints $(1, e)$ and $(3, e^3)$, i.e.,

$$l_1(x) = e + m(x - 1).$$

Here, the slope m is given by

$$m = \frac{e^3 - e}{2}. \quad (4.17)$$

Let $l_2(x)$ denote the tangent to $f(x)$ at a point a that is identified such that the slope is m . Since $f'(x) = e^x$, we have $e^a = m$, i.e.,

$$a = \log m.$$

Now

$$f(a) = e^{\log m} = m,$$

and

$$l_1(a) = e + m(\log m - 1).$$

Hence, the average between $f(a)$ and $l_1(a)$ which we denote by \bar{y} is given by

$$\bar{y} = \frac{f(a) + l_1(a)}{2} = \frac{m + e + m \log m - m}{2} = \frac{e + m \log m}{2}.$$

The minimax polynomial $P_1^*(x)$ is the line of slope m that passes through (a, \bar{y}) ,

$$P_1^*(x) - \frac{e + m \log m}{2} = m(x - \log m),$$

i.e.,

$$P_1^*(x) = mx + \frac{e - m \log m}{2},$$

where the slope m is given by (4.17). We note that the maximal difference between $P_1^*(x)$ and $f(x)$ is obtained at $x = 1, a, 3$.

4.3 Least-squares Approximations

4.3.1 The least-squares approximation problem

We recall that the L^2 -norm of a function $f(x)$ is defined as

$$\|f\|_2 = \sqrt{\int_a^b |f(x)|^2 dx}.$$

As before, we let Π_n denote the space of all polynomials of degree $\leq n$. The **least-squares approximation problem** is to find the polynomial that is closest to $f(x)$ in the L^2 -norm among all polynomials of degree $\leq n$, i.e., to find $Q_n^*(x) \in \Pi_n$ such that

$$\|f(x) - Q_n^*(x)\|_2 = \min_{Q_n(x) \in \Pi_n} \|f(x) - Q_n(x)\|_2 = \min_{Q_n(x) \in \Pi_n} \sqrt{\int_a^b |f(x) - Q_n(x)|^2}.$$

4.3.2 Solving the least-squares problem: a direct method

Our goal is to find a polynomial in Π_n that minimizes the distance $\|f(x) - Q_n(x)\|_2$ among all polynomials $Q_n \in \Pi_n$. We thus consider

$$Q_n(x) = \sum_{j=0}^n c_j x^j.$$

For convenience, instead of minimizing the L_2 norm of the difference, we will minimize its square, i.e., we will find a polynomial $Q_n(x) \in \Pi_n$ that minimizes:

$$\int_a^b (f(x) - Q_n(x))^2.$$

We thus let ϕ denote the square of the L_2 -distance between $f(x)$ and $Q_n(x)$, i.e.,

$$\begin{aligned}\phi(c_0, \dots, c_n) &= \int_a^b (f(x) - Q_n(x))^2 dx = \int_a^b \left(f(x) - \sum_{j=0}^n c_j x^j \right)^2 dx = \\ &= \int_a^b (f(x))^2 dx - 2 \sum_{j=0}^n c_j \int_a^b x^j f(x) dx + \sum_{i=0}^n \sum_{j=0}^n c_i c_j \int_a^b x^{i+j} dx.\end{aligned}$$

ϕ is a function of the $n + 1$ coefficients in the polynomial $Q_n(x)$. We want to find a point $c^* = (c_0^*, \dots, c_n^*) \in \mathbb{R}^{n+1}$ for which ϕ obtains a minimum. At this point, the partial derivative with respect to all variables should vanish:

$$\left. \frac{\partial \phi}{\partial c_k} \right|_{c=c^*} = 0. \quad (4.18)$$

The condition (4.18) implies that

$$\begin{aligned}0 &= -2 \int_a^b x^k f(x) dx + \sum_{i=0}^n c_i^* \int_a^b x^{i+k} dx + \sum_{j=0}^n c_j^* \int_a^b x^{j+k} dx \\ &= 2 \left[\sum_{j=0}^n c_j^* \int_a^b x^{j+k} dx - \int_a^b x^k f(x) dx \right].\end{aligned} \quad (4.19)$$

Since the function $f(x)$ is known, equation (4.19) is a linear system for the $n + 1$ unknowns (c_0^*, \dots, c_n^*) :

$$\sum_{j=0}^n c_j^* \int_a^b x^{j+k} dx = \int_a^b x^k f(x) dx, \quad k = 0, \dots, n. \quad (4.20)$$

In matrix form, the linear system (4.20) can be written as

$$H_{n+1} c^* = g, \quad (4.21)$$

The coefficients matrix, $H_{n+1}(a, b)$ is the $(n + 1) \times (n + 1)$ matrix:

$$(H_{n+1}(a, b))_{j,k} = \int_a^b x^{j+k} dx, \quad 0 \leq j, k \leq n. \quad (4.22)$$

c^* is the vector of unknowns, $c^* = (c_0^*, \dots, c_n^*)$, and the components of $g = (g_0, \dots, g_n)$, are the RHS of (4.20):

$$g_k = \int_a^b x^k f(x) dx, \quad k = 0, \dots, n.$$

For example, in the case where $[a, b] = [0, 1]$,

$$H_n(0, 1) = \begin{pmatrix} 1/1 & 1/2 & \dots & 1/n \\ 1/2 & 1/3 & \dots & 1/(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/(n+1) & \dots & 1/(2n-1) \end{pmatrix} \quad (4.23)$$

The matrix (4.23) is known as the **Hilbert matrix**. It can be shown that the determinant of H_n is given by

$$\det(H_n) = \frac{(1!2! \cdots (n-1)!)^4}{1!2! \cdots (2n-1)!}.$$

Hence, $\det(H_n) \neq 0$ and H_n is invertible.

We thus know that the solution of the least-squares problem is the polynomial

$$Q_n^*(x) = \sum_{j=0}^n c_j^* x^j,$$

where the coefficients c_j^* , $j = 0, \dots, n$, are the solution of the linear system (4.20). Since the coefficients matrix, H_{n+1} is invertible, the system (4.20) has a unique solution, which proves that not only the least-squares problem has a solution, but it is also unique.

Is solving the system (4.21), i.e., inverting the Hilbert matrix, a good way of solving the least-squares problem? Unfortunately not. There are numerical instabilities that are associated with inverting H . We demonstrate this with the following example.

Example 4.11

The Hilbert matrix H_5 is

$$H_5 = \begin{pmatrix} 1/1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{pmatrix}$$

The inverse of H_5 is

$$H_5^{-1} = \begin{pmatrix} 25 & -300 & 1050 & -1400 & 630 \\ -300 & 4800 & -18900 & 26880 & -12600 \\ 1050 & -18900 & 79380 & -117600 & 56700 \\ -1400 & 26880 & -117600 & 179200 & -88200 \\ 630 & -12600 & 56700 & -88200 & 44100 \end{pmatrix}$$

The condition number of H_5 is $4.77 \cdot 10^5$, which indicates that the Hilbert matrix is ill-conditioned. In fact, the condition number of H_n increases with the dimension n so inverting it becomes more difficult with an increasing dimension.

4.3.3 Solving the least-squares problem with orthogonal polynomials

In Section 4.3.2 we showed that the least squares problem can be solved by considering a polynomial of the form:

$$Q_n^*(x) = \sum_{j=0}^n c_j^* x^j, \quad (4.24)$$

Unfortunately computing the coefficients c_0^*, \dots, c_n^* require inverting the ill-posed Hilbert matrix (4.22).

In this section we provide an alternative approach for solving the least-squares problem. Now that we know that the least-squares problem has a solution and it is unique, our goal is to find a better computational approach for computing it. We are still looking for a polynomial of degree $\leq n$, but instead of insisting that it is written in the form (4.24), i.e., as a linear combination of the functions $1, x, x^2, \dots, x^n$, we will write the polynomial $Q_n(x)$ as a linear combination of more general polynomials:

$$Q_n(x) = \sum_{j=0}^n c_j P_j(x). \quad (4.25)$$

Here, every $P_k(x)$ is a polynomial of $\deg(P_k(x)) = k$. These polynomials are yet to be determined, but for the time being we assume they are given, known polynomials. Clearly, since $Q_n(x)$ is written as a linear combination of polynomials of degree $k = 0, \dots, n$, $Q_n(x)$ is a polynomial of degree $\leq n$.

We repeat the calculations of Section 4.3.2, only that this time, the unknowns (c_0, \dots, c_n) are the coefficients of the linear combination (4.25). Our goal is to minimize

$$\phi(c_0, \dots, c_n) = \int_a^b (f(x) - Q_n(x))^2 dx.$$

At the minimum, $c^* = (c_0^*, \dots, c_n^*)$, we have

$$0 = \left. \frac{\partial \phi}{\partial c_k} \right|_{c=c^*} = -2 \int_a^b P_k(x) f(x) dx + 2 \sum_{j=0}^n c_j^* \int_a^b P_j(x) P_k(x) dx,$$

i.e.,

$$\sum_{j=0}^n c_j^* \int_a^b P_j(x) P_k(x) dx = \int_a^b P_k(x) f(x) dx, \quad k = 0, \dots, n. \quad (4.26)$$

Note the similarity between equation (4.26) and (4.20). In (4.20), we used the basis functions $\{x^k\}_{k=0}^n$ (a basis of Π_n), while here we work with the polynomials $\{P_k(x)\}_{k=0}^n$ instead.

The key idea is to choose the polynomials $\{P_k(x)\}_{k=0}^n$ such that the system (4.26) can be easily solved. We would like to choose polynomials that satisfy:

$$\int_a^b P_j(x)P_k(x)dx = \delta_{jk} = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases} \quad (4.27)$$

Polynomials that satisfy (4.27) are called **orthonormal polynomials**. If, indeed, the polynomials $\{P_k(x)\}_{k=0}^n$ are orthonormal, then (4.26) implies that

$$c_j^* = \int_a^b P_j(x)f(x)dx, \quad j = 0, \dots, n. \quad (4.28)$$

The solution of the least-squares problem is a polynomial

$$Q_n^*(x) = \sum_{j=0}^n c_j^* P_j(x), \quad (4.29)$$

with coefficients c_j^* , $j = 0, \dots, n$, that are given by (4.28).

Remark. Polynomials that satisfy

$$\int_a^b P_j(x)P_k(x)dx = \begin{cases} \int_a^b (P_j(x))^2 dx, & j = k, \\ 0, & \text{otherwise,} \end{cases}$$

with $\int_a^b (P_j(x))^2 dx$ that is not necessarily 1 are called **orthogonal polynomials**. In this case, the solution of the least-squares problem is given by the polynomial $Q_n^*(x)$ in (4.29) with the coefficients

$$c_j^* = \frac{\int_a^b P_j(x)f(x)dx}{\int_a^b (P_j(x))^2 dx}, \quad j = 0, \dots, n. \quad (4.30)$$

At this point we know how to solve the least-squares problem if the orthogonal (or orthonormal) polynomials are given. We still have to discuss how such polynomials can be found. Before doing so, we will briefly comment on a more general framework for the least-squares problem, the weighted least-squares problem.

4.3.4 The weighted least-squares problem

A more general least-squares problem is the **weighted least squares approximation problem**. We consider a **weight function**, $w(x)$, to be a non-negative continuous function on (a, b) , with a positive mass, i.e.,

$$\int_a^b w(x)dx > 0.$$

Note that $w(x)$ may be singular at the edges of the interval since we do not require it to be continuous on the closed interval $[a, b]$. For any weight $w(x)$, we define the corresponding weighted L^2 -norm of a function $f(x)$ as

$$\|f\|_{2,w} = \sqrt{\int_a^b (f(x))^2 w(x) dx}.$$

The weighted least-squares problem is to find the closest polynomial $Q_n^*(x) \in \Pi_n$ to $f(x)$, this time in the weighted L^2 -norm sense, i.e., we look for a polynomial $Q_n^*(x)$ of degree $\leq n$ such that

$$\|f - Q_n^*\|_{2,w} = \min_{Q_n \in \Pi_n} \|f - Q_n\|_{2,w} = \min_{Q_n \in \Pi_n} \sqrt{\int_a^b (f(x) - Q_n(x))^2 w(x) dx}. \quad (4.31)$$

In order to solve the weighted least-squares problem (4.31) we follow the approach of Section 4.3.3, and consider polynomials $\{P_k\}_{k=0}^n$ such that $\deg(P_k(x)) = k$. We then consider a polynomial $Q_n(x)$ that is written as their linear combination:

$$Q_n(x) = \sum_{j=0}^n c_j P_j(x).$$

By repeating the calculations of Section 4.3.3, we obtain for the coefficients of the minimizer Q_n^* ,

$$\sum_{j=0}^n c_j^* \int_a^b w(x) P_j(x) P_k(x) dx = \int_a^b w(x) P_k(x) f(x) dx, \quad k = 0, \dots, n, \quad (4.32)$$

(compare with (4.26)). The system (4.32) can be easily solved if we choose $\{P_k(x)\}$ to be orthonormal with respect to the weight $w(x)$, i.e.,

$$\int_a^b P_j(x) P_k(x) w(x) dx = \delta_{jk}.$$

Hence, the solution of the weighted least-squares problem is given by

$$Q_n^*(x) = \sum_{j=0}^n c_j^* P_j(x), \quad (4.33)$$

with coefficients that are given by

$$c_j^* = \int_a^b P_j(x) f(x) w(x) dx, \quad j = 0, \dots, n. \quad (4.34)$$

Remark. In the case where $\{P_k(x)\}$ are orthogonal but not necessarily normalized, the solution of the weighted least-squares problem is given by

$$Q_n^*(x) = \sum_{j=0}^n c_j^* P_j(x)$$

with

$$c_j^* = \frac{\int_a^b P_j(x) f(x) w(x) dx}{\int_a^b (P_j(x))^2 w(x) dx}, \quad j = 0, \dots, n.$$

4.3.5 Orthogonal polynomials

At this point we already know that orthogonal polynomials play a central role in the solution of least-squares problems. In this section we will focus on the construction of orthogonal polynomials. The properties of orthogonal polynomials will be studied in Section 4.4.2.

We start by defining the **weighted inner product** between two functions $f(x)$ and $g(x)$ (with respect to the weight $w(x)$):

$$\langle f, g \rangle_w = \int_a^b f(x) g(x) w(x) dx.$$

To simplify the notations, even in the weighted case, we will typically write $\langle f, g \rangle$ instead of $\langle f, g \rangle_w$. Some properties of the weighted inner product include

1. $\langle \alpha f, g \rangle = \langle f, \alpha g \rangle = \alpha \langle f, g \rangle, \quad \forall \alpha \in \mathbb{R}.$
2. $\langle f_1 + f_2, g \rangle = \langle f_1, g \rangle + \langle f_2, g \rangle.$
3. $\langle f, g \rangle = \langle g, f \rangle$
4. $\langle f, f \rangle \geq 0$ and $\langle f, f \rangle = 0$ iff $f \equiv 0$. Here we must assume that $f(x)$ is continuous in the interval $[a, b]$. If it is not continuous, we can have $\langle f, f \rangle = 0$ and $f(x)$ can still be non-zero (e.g., in one point).

The weighted L_2 -norm can be obtained from the weighted inner product by

$$\|f\|_{2,w} = \sqrt{\langle f, f \rangle_w}.$$

Given a weight $w(x)$, we are interested in constructing orthogonal (or orthonormal) polynomials. This can be done using **the Gram-Schmidt orthogonalization process**, which we now describe in detail.

In the general context of linear algebra, the Gram-Schmidt process is being used to convert one set of linearly independent vectors to an orthogonal set of vectors that spans the same subspace as the original set. In our context, we should think about the process as converting one set of polynomials that span the space of polynomials of degree $\leq n$ to an orthogonal set of polynomials that spans the same space Π_n . Accordingly, we set the initial set of polynomials as $\{1, x, x^2, \dots, x^n\}$, which we would like to convert to orthogonal polynomials (of an increasing degree) with respect to the weight $w(x)$.

We will first demonstrate the process with the weight $w(x) \equiv 1$. We will generate a set of orthogonal polynomials $\{P_0(x), \dots, P_n(x)\}$ from $\{1, x, \dots, x^n\}$. The degree of the polynomials P_i is i .

We start by setting

$$P_0(x) = 1.$$

We then let

$$P_1(x) = x - c_1^0 P_0(x) = x - c_1^0.$$

The orthogonality condition $\int_a^b P_1 P_0 dx = 0$, implies that

$$\int_a^b 1 \cdot (x - c_1^0) dx = 0,$$

from which $c = (a+b)/2$, and thus

$$P_1(x) = x - \frac{a+b}{2}.$$

The computation continues in a similar fashion. We set

$$P_2(x) = x^2 - c_2^0 P_0(x) - c_2^1 P_1(x).$$

The two unknown coefficients, c_2^0 and c_2^1 , are computed from the orthogonality conditions. This time, $P_2(x)$ should be orthogonal to $P_0(x)$ and to $P_1(x)$, i.e.,

$$\int_a^b P_2(x) P_0(x) dx = 0, \quad \text{and} \quad \int_a^b P_2(x) P_1(x) dx = 0,$$

and so on. If, in addition to the orthogonality condition, we would like the polynomials to be orthonormal, all that remains is to normalize:

$$\hat{P}_n(x) = \frac{P_n(x)}{\|P_n(x)\|} = \frac{P_n(x)}{\sqrt{\int_a^b (P_n(x))^2 dx}}, \quad \forall n.$$

The orthogonalization process is identical to the process that we described even when the weight $w(x)$ is not uniformly one. In this case, every integral will contain the weight.

4.4 Examples of orthogonal polynomials

This section includes several examples of orthogonal polynomials and a very brief summary of some of their properties.

1. **Legendre polynomials.** We start with the Legendre polynomials. This is a family of polynomials that are orthogonal with respect to the weight

$$w(x) \equiv 1,$$

on the interval $[-1, 1]$.

In addition to deriving the Legendre polynomials through the Gram-Schmidt orthogonalization process, it can be shown that the Legendre polynomials can be obtained from the recurrence relation

$$(n+1)P_{n+1}(x) - (2n+1)xP_n(x) + nP_{n-1}(x) = 0, \quad n \geq 1, \quad (4.35)$$

starting from the first two polynomials:

$$P_0(x) = 1, \quad P_1(x) = x.$$

Instead of calculating these polynomials one by one from the recurrence relation, they can be obtained directly from Rodrigues' formula

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n], \quad n \geq 0. \quad (4.36)$$

The Legendre polynomials satisfy the orthogonality condition

$$\langle P_n, P_m \rangle = \frac{2}{2n+1} \delta_{nm}. \quad (4.37)$$

2. **Chebyshev polynomials.** Our second example is of the Chebyshev polynomials. These polynomials are orthogonal with respect to the weight

$$w(x) = \frac{1}{\sqrt{1-x^2}},$$

on the interval $[-1, 1]$. They satisfy the recurrence relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1, \quad (4.38)$$

together with $T_0(x) = 1$ and $T_1(x) = x$ (see (3.33)). They and are explicitly given by

$$T_n(x) = \cos(n \cos^{-1} x), \quad n \geq 0. \quad (4.39)$$

(see (3.34)). The orthogonality relation that they satisfy is

$$\langle T_n, T_m \rangle = \begin{cases} 0, & n \neq m, \\ \pi, & n = m = 0, \\ \frac{\pi}{2}, & n = m \neq 0. \end{cases} \quad (4.40)$$

3. **Laguerre polynomials.** We proceed with the Laguerre polynomials. Here the interval is given by $[0, \infty)$ with the weight function

$$w(x) = e^{-x}.$$

The Laguerre polynomials are given by

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x}), \quad n \geq 0. \quad (4.41)$$

The normalization condition is

$$\|L_n\| = 1. \quad (4.42)$$

A more general form of the Laguerre polynomials is obtained when the weight is taken as

$$e^{-x} x^\alpha,$$

for an arbitrary real $\alpha > -1$, on the interval $[0, \infty)$.

4. **Hermite polynomials.** The Hermite polynomials are orthogonal with respect to the weight

$$w(x) = e^{-x^2},$$

on the interval $(-\infty, \infty)$. The can be explicitly written as

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n e^{-x^2}}{dx^n}, \quad n \geq 0. \quad (4.43)$$

Another way of expressing them is by

$$H_n(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{(-1)^k n!}{k!(n-2k)!} (2x)^{n-2k}, \quad (4.44)$$

where $\lfloor x \rfloor$ denotes the largest integer that is $\leq x$. The Hermite polynomials satisfy the recurrence relation

$$H_{n+1}(x) - 2xH_n(x) + 2nH_{n-1}(x) = 0, \quad n \geq 1, \quad (4.45)$$

together with

$$H_0(x) = 1, \quad H_1(x) = 2x.$$

They satisfy the orthogonality relation

$$\int_{-\infty}^{\infty} e^{-x^2} H_n(x) H_m(x) dx = 2^n n! \sqrt{\pi} \delta_{nm}. \quad (4.46)$$

4.4.1 Another approach to the least-squares problem

In this section we present yet another way of deriving the solution of the least-squares problem. Along the way, we will be able to derive some new results. We recall that our goal is to minimize $\|f(x) - Q_n(x)\|_{2,w}$, $\forall Q_n \in \Pi_n$, i.e., to minimize the integral

$$\int_a^b w(x) (f(x) - Q_n(x))^2 dx \quad (4.47)$$

among all the polynomials $Q_n(x)$ of degree $\leq n$. The minimizer of (4.47) is denoted by $Q_n^*(x)$.

Assume that $\{P_k(x)\}_{k \geq 0}$ is an orthogonal family of polynomials with respect to $w(x)$, and let

$$Q_n(x) = \sum_{j=0}^n c_j P_j(x).$$

Then

$$\|f - Q_n\|_{2,w}^2 = \int_a^b w(x) \left(f(x) - \sum_{j=0}^n c_j P_j(x) \right)^2 dx.$$

Hence

$$\begin{aligned} 0 &\leq \left\langle f - \sum_{j=0}^n c_j P_j, f - \sum_{j=0}^n c_j P_j \right\rangle_w = \langle f, f \rangle_w - 2 \sum_{j=0}^n c_j \langle f, P_j \rangle_w + \sum_{i=0}^n \sum_{j=0}^n c_i c_j \langle P_i, P_j \rangle_w \\ &= \|f\|_{2,w}^2 - 2 \sum_{j=0}^n c_j \langle f, P_j \rangle_w + \sum_{j=0}^n c_j^2 \|P_j\|_{2,w}^2 \\ &= \|f\|_{2,w}^2 - \sum_{j=0}^n \frac{\langle f, P_j \rangle_w^2}{\|P_j\|_{2,w}^2} + \sum_{j=0}^n \left(\frac{\langle f, P_j \rangle_w}{\|P_j\|_{2,w}} - c_j \|P_j\|_{2,w} \right)^2. \end{aligned}$$

The last expression is minimal iff

$$\frac{\langle f, P_j \rangle_w}{\|P_j\|_{2,w}} - c_j \|P_j\|_{2,w} = 0, \quad \forall 0 \leq j \leq n,$$

i.e., if

$$c_j = \frac{\langle f, P_j \rangle_w}{\|P_j\|_{2,w}^2}.$$

Hence, there exists a unique least-squares approximation which is given by

$$Q_n^*(x) = \sum_{j=0}^n \frac{\langle f, P_j \rangle_w}{\|P_j\|_{2,w}^2} P_j(x). \quad (4.48)$$

If the polynomials $\{P_j(x)\}$ are also normalized so that $\|P_j\|_{2,w} = 1$, then the minimizer $Q_n^*(x)$ in (4.48) becomes

$$Q_n^*(x) = \sum_{j=0}^n \langle f, P_j \rangle_w P_j(x).$$

Remarks.

1. We can write

$$\begin{aligned} \|f - Q_n^*\|_{2,w}^2 &= \int_a^b w(x) \left(f(x) - \sum_{j=0}^n c_j P_j(x) \right)^2 dx = \\ &= \|f\|_{2,w}^2 - 2 \sum_{j=0}^n \langle f, P_j \rangle_w c_j + \sum_{j=0}^n \|P_j\|_{2,w}^2 c_j^2. \end{aligned}$$

Since $\|P_j\|_{2,w} = 1$, $c_j = \langle f, P_j \rangle_w$, so that

$$\|f - Q_n^*\|_{2,w}^2 = \|f\|_{2,w}^2 - \sum_{j=0}^n \langle f, P_j \rangle_w^2.$$

Hence

$$\sum_{j=0}^n \langle f, P_j \rangle_w^2 = \|f\|^2 - \|f - Q_n^*\|^2 \leq \|f\|^2,$$

i.e.,

$$\sum_{j=0}^n \langle f, P_j \rangle_w^2 \leq \|f\|_{2,w}^2. \quad (4.49)$$

The inequality (4.49) is called **Bessel's inequality**.

2. Assuming that $[a, b]$ is finite, we have

$$\lim_{n \rightarrow \infty} \|f - Q_n^*\|_{2,w} = 0.$$

Hence

$$\|f\|_{2,w}^2 = \sum_{j=0}^{\infty} \langle f, P_j \rangle_w^2, \quad (4.50)$$

which is known as **Parseval's equality**.

Remark. An informal, quick way to find out the formula for the coefficients of the least-squares solution is the following. We are looking for a least-squares approximation of $f(x)$ of the form:

$$Q_n(x) = \sum_j^n c_j P_j(x). \quad (4.51)$$

We assume that $\{P_j(x)\}$ are orthogonal polynomials of degree j , $j = 0, \dots, n$. We let

$$f(x) \approx Q_n(x),$$

and multiply both sides by P_k using the weighted inner product $\langle \cdot \rangle_w$. This leads to

$$\langle f(x), P_k(x) \rangle_w \approx \langle Q_n(x), P_k(x) \rangle_w = \sum_j c_j \langle P_j(x), P_k(x) \rangle_w$$

which due to the orthogonality is equal to $c_k \|P_k\|^2$. Hence, the coefficients c_k are given by

$$c_k = \frac{\langle f(x), P_k(x) \rangle_w}{\|P_k(x)\|_w^2} = \frac{\int_a^b f(x) P_k(x) w(x) dx}{\int_a^b (P_k(x))^2 w(x) dx}.$$

Example 4.12

Problem: Let $f(x) = \cos x$ on $[-1, 1]$. Find the polynomial in Π_2 , that minimizes

$$\int_{-1}^1 [f(x) - Q_2(x)]^2 dx.$$

Solution: The weight $w(x) \equiv 1$ on $[-1, 1]$ implies that the orthogonal polynomials we need to use are the Legendre polynomials. We are seeking for polynomials of degree ≤ 2 so we write the first three Legendre polynomials

$$P_0(x) \equiv 1, \quad P_1(x) = x, \quad P_2(x) = \frac{1}{2}(3x^2 - 1).$$

The normalization factor satisfies, in general,

$$\int_{-1}^1 P_n^2(x) dx = \frac{2}{2n+1}.$$

Hence

$$\int_{-1}^1 P_0^2(x) dx = 2, \quad \int_{-1}^1 P_1(x) dx = \frac{2}{3}, \quad \int_{-1}^1 P_2^2(x) dx = \frac{2}{5}.$$

We can then replace the Legendre polynomials by their normalized counterparts:

$$P_0(x) \equiv \frac{1}{\sqrt{2}}, \quad P_1(x) = \sqrt{\frac{3}{2}}x, \quad P_2(x) = \frac{\sqrt{5}}{2\sqrt{2}}(3x^2 - 1).$$

We now have

$$\langle f, P_0 \rangle = \int_{-1}^1 \cos x \frac{1}{\sqrt{2}} dx = \frac{1}{\sqrt{2}} \sin x \Big|_{-1}^1 = \sqrt{2} \sin 1.$$

Hence

$$Q_0^*(x) \equiv \sin 1.$$

We also have

$$\langle f, P_1 \rangle = \int_{-1}^1 \cos x \sqrt{\frac{3}{2}} x dx = 0.$$

which means that $Q_1^*(x) = Q_0^*(x)$. Finally,

$$\langle f, P_2 \rangle = \int_{-1}^1 \cos x \sqrt{\frac{5}{2}} \frac{3x^2 - 1}{2} dx = \frac{1}{2} \sqrt{\frac{5}{2}} (12 \cos 1 - 8 \sin 1),$$

and hence the desired polynomial, $Q_2^*(x)$, is given by

$$Q_2^*(x) = \sin 1 + \left(\frac{15}{2} \cos 1 - 5 \sin 1 \right) (3x^2 - 1).$$

In Figure 4.3 we plot the original function $f(x) = \cos x$ (solid line) and its approximation $Q_2^*(x)$ (dashed line). We zoom on the interval $x \in [0, 1]$.

Remark. If the weight is $w(x) \equiv 1$ but the interval is $[a, b]$, we can still use the Legendre polynomials if we make the following change of variables. Define

$$x = \frac{b + a + (b - a)t}{2}.$$

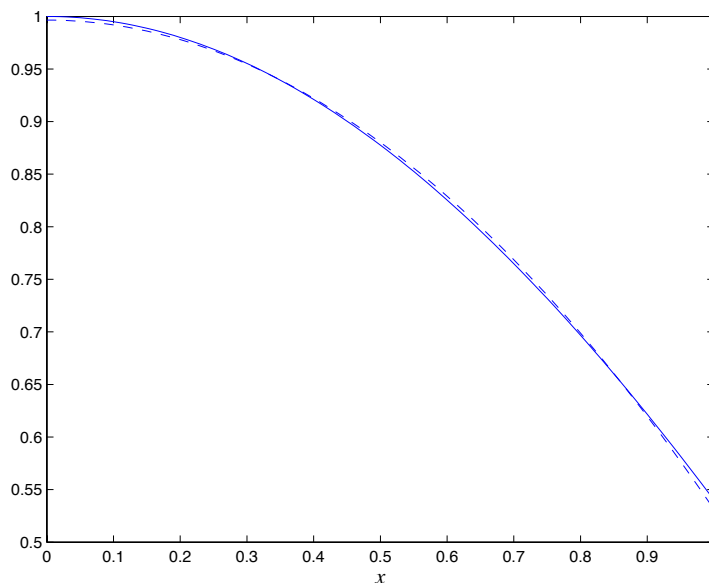


Figure 4.3: A second-order L^2 -approximation of $f(x) = \cos x$. *Solid line: $f(x)$; Dashed line: its approximation $Q_2^*(x)$*

Then the interval $-1 \leq t \leq 1$ is mapped to $a \leq x \leq b$. Now, define

$$F(t) = f\left(\frac{b+a+(b-a)t}{2}\right) = f(x).$$

Hence

$$\int_a^b [f(x) - Q_n(x)]^2 dx = \frac{b-a}{2} \int_{-1}^1 [F(t) - q_n(t)]^2 dt.$$

Example 4.13

Problem: Let $f(x) = \cos x$ on $[0, \pi]$. Find the polynomial in Π_1 that minimizes

$$\int_0^\pi [f(x) - Q_1(x)]^2 dx.$$

Solution:

$$\int_0^\pi (f(x) - Q_1^*(x))^2 dx = \frac{\pi}{2} \int_{-1}^1 [F(t) - q_n(t)]^2 dt.$$

Letting

$$x = \frac{\pi + \pi t}{2} = \frac{\pi}{2}(1+t),$$

we have

$$F(t) = \cos\left(\frac{\pi}{2}(1+t)\right) = -\sin\frac{\pi t}{2}.$$

We already know that the first two normalized Legendre polynomials are

$$P_0(t) = \frac{1}{\sqrt{2}}, \quad P_1(t) = \sqrt{\frac{3}{2}}t.$$

Hence

$$\langle F, P_0 \rangle = -\int_{-1}^1 \frac{1}{\sqrt{2}} \sin\frac{\pi t}{2} dt = 0,$$

which means that $Q_0^*(t) = 0$. Also

$$\langle F, P_1 \rangle = -\int_{-1}^1 \sin\frac{\pi t}{2} \sqrt{\frac{3}{2}}t dt = -\sqrt{\frac{3}{2}} \left[\frac{\sin\frac{\pi t}{2}}{\left(\frac{\pi}{2}\right)^2} - \frac{t \cos\frac{\pi t}{2}}{\frac{\pi}{2}} \right]_{-1}^1 = -\sqrt{\frac{3}{2}} \cdot \frac{8}{\pi^2}.$$

Hence

$$q_1^*(t) = -\frac{3}{2} \cdot \frac{8}{\pi^2} t = -\frac{12}{\pi^2} t \quad \implies \quad Q_1^*(x) = -\frac{12}{\pi^2} \left(\frac{2}{\pi} x - 1 \right).$$

In Figure 4.4 we plot the original function $f(x) = \cos x$ (solid line) and its approximation $Q_1^*(x)$ (dashed line).

Example 4.14

Problem: Let $f(x) = \cos x$ in $[0, \infty)$. Find the polynomial in Π_1 that minimizes

$$\int_0^\infty e^{-x} [f(x) - Q_1(x)]^2 dx.$$

Solution: The family of orthogonal polynomials that correspond to this weight on $[0, \infty)$ are Laguerre polynomials. Since we are looking for the minimizer of the weighted L_2 norm among polynomials of degree ≤ 1 , we will need to use the first two Laguerre polynomials:

$$L_0(x) = 1, \quad L_1(x) = 1 - x.$$

We thus have

$$\langle f, L_0 \rangle_w = \int_0^\infty e^{-x} \cos x dx = \frac{e^{-x}(-\cos x + \sin x)}{2} \Big|_0^\infty = \frac{1}{2}.$$

Also

$$\langle f, L_1 \rangle_w = \int_0^\infty e^{-x} \cos x (1-x) dx = \frac{1}{2} - \left[\frac{x e^{-x}(-\cos x + \sin x)}{2} - \frac{e^{-x}(-2 \sin x)}{4} \right]_0^\infty = \frac{1}{2}.$$

This means that

$$Q_1^*(x) = \langle f, L_0 \rangle_w L_0(x) + \langle f, L_1 \rangle_w L_1(x) = \frac{1}{2} + \frac{1}{2}(1-x) = 1 - \frac{x}{2}.$$

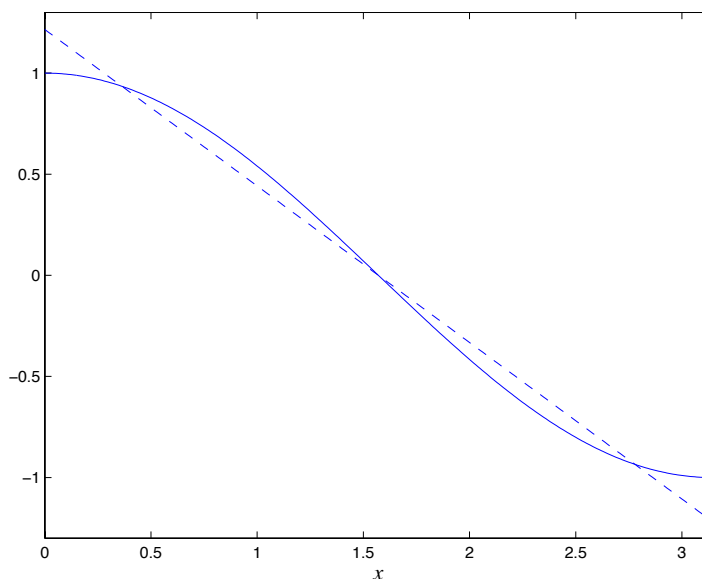


Figure 4.4: A first-order L^2 -approximation of $f(x) = \cos x$ on the interval $[0, \pi]$. *Solid line: $f(x)$, Dashed line: its approximation $Q_1^*(x)$*

4.4.2 Properties of orthogonal polynomials

We start with a theorem that deals with some of the properties of the roots of orthogonal polynomials. This theorem will become handy when we discuss Gaussian quadratures in Section 6.7. We let $\{P_n(x)\}_{n \geq 0}$ be orthogonal polynomials in $[a, b]$ with respect to the weight $w(x)$.

Theorem 4.15 *The roots x_j , $j = 1, \dots, n$ of $P_n(x)$ are all real, simple, and are in (a, b) .*

Proof. Let x_1, \dots, x_r be the roots of $P_n(x)$ in (a, b) . Let

$$Q_r(x) = (x - x_1) \cdot \dots \cdot (x - x_r).$$

Then $Q_r(x)$ and $P_n(x)$ change their signs together in (a, b) . Also

$$\deg(Q_r(x)) = r \leq n.$$

Hence $(P_n Q_r)(x)$ is a polynomial with one sign in (a, b) . This implies that

$$\int_a^b P_n(x) Q_r(x) w(x) dx \neq 0,$$

and hence $r = n$ since $P_n(x)$ is orthogonal to polynomials of degree less than n .

Without loss of generality we now assume that x_1 is a multiple root, i.e.,

$$P_n(x) = (x - x_1)^2 P_{n-2}(x).$$

Hence

$$P_n(x)P_{n-2}(x) = \left(\frac{P_n(x)}{x - x_1} \right)^2 \geq 0,$$

which implies that

$$\int_a^b P_n(x)P_{n-2}(x)dx > 0.$$

This is not possible since P_n is orthogonal to P_{n-2} . Hence roots can not repeat. ■

Another important property of orthogonal polynomials is that they can all be written in terms of recursion relations. We have already seen specific examples of such relations for the Legendre, Chebyshev, and Hermite polynomials (see (4.35), (4.38), and (4.45)). The following theorem states such relations always hold.

Theorem 4.16 (Triple Recursion Relation) *Any three consecutive orthonormal polynomials are related by a recursion formula of the form*

$$P_{n+1}(x) = (A_n x + B_n)P_n(x) - C_n P_{n-1}(x).$$

If a_k and b_k are the coefficients of the terms of degree k and degree $k - 1$ in $P_k(x)$, then

$$A_n = \frac{a_{n+1}}{a_n}, \quad B_n = \frac{a_{n+1}}{a_n} \left(\frac{b_{n+1}}{a_{n+1}} - \frac{b_n}{a_n} \right), \quad C_n = \frac{a_{n+1}a_{n-1}}{a_n^2}.$$

Proof. For

$$A_n = \frac{a_{n+1}}{a_n},$$

let

$$\begin{aligned} Q_n(x) &= P_{n+1}(x) - A_n x P_n(x) \\ &= (a_{n+1}x^{n+1} + b_{n+1}x^n + \dots) - \frac{a_{n+1}}{a_n}x(a_n x^n + b_n x^{n-1} + \dots) \\ &= \left(b_{n+1} - \frac{a_{n+1}b_n}{a_n} \right) x^n + \dots \end{aligned}$$

Hence $\deg(Q(x)) \leq n$, which means that there exists $\alpha_0, \dots, \alpha_n$ such that

$$Q(x) = \sum_{i=0}^n \alpha_i P_i(x).$$

For $0 \leq i \leq n-2$,

$$\alpha_i = \frac{\langle Q, P_i \rangle}{\langle P_i, P_i \rangle} = \langle Q, P_i \rangle = \langle P_{n+1} - A_n x P_n, P_i \rangle = -A_n \langle x P_n, P_i \rangle = 0.$$

Hence

$$Q_n(x) = \alpha_n P_n(x) + \alpha_{n-1} P_{n-1}(x).$$

Set $\alpha_n = B_n$ and $\alpha_{n-1} = -C_n$. Then, since

$$x P_{n-1} = \frac{a_{n-1}}{a_n} P_n + q_{n-1},$$

we have

$$C_n = A_n \langle x P_n, P_{n-1} \rangle = A_n \langle P_n, x P_{n-1} \rangle = A_n \left\langle P_n, \frac{a_{n-1}}{a_n} P_n + q_{n-1} \right\rangle = A_n \frac{a_{n-1}}{a_n}.$$

Finally

$$P_{n+1} = (A_n x + B_n) P_n - C_n P_{n-1},$$

can be explicitly written as

$$a_{n+1} x^{n+1} + b_{n+1} x^n + \dots = (A_n x + B_n)(a_n x^n + b_n x^{n-1} + \dots) - C_n (a_{n-1} x^{n-1} + b_{n-1} x^{n-2} + \dots).$$

The coefficient of x^n is

$$b_{n+1} = A_n b_n + B_n a_n,$$

which means that

$$B_n = (b_{n+1} - A_n b_n) \frac{1}{a_n}. \quad \blacksquare$$

5 Numerical Differentiation

5.1 Basic Concepts

This chapter deals with numerical approximations of derivatives. The first question that comes up to mind is: why do we need to approximate derivatives at all? After all, we do know how to analytically differentiate every function. Nevertheless, there are several reasons as of why we still need to approximate derivatives:

- Even if there exists an underlying function that we need to differentiate, we might know its values only at a sampled data set without knowing the function itself.
- There are some cases where it may not be obvious that an underlying function exists and all that we have is a discrete data set. We may still be interested in studying changes in the data, which are related, of course, to derivatives.
- There are times in which exact formulas are available but they are very complicated to the point that an exact computation of the derivative requires a lot of function evaluations. It might be significantly simpler to approximate the derivative instead of computing its exact value.
- When approximating solutions to ordinary (or partial) differential equations, we typically represent the solution as a discrete approximation that is defined on a grid. Since we then have to evaluate derivatives at the grid points, we need to be able to come up with methods for approximating the derivatives at these points, and again, this will typically be done using only values that are defined on a lattice. The underlying function itself (which in this case is the solution of the equation) is unknown.

A simple approximation of the first derivative is

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \quad (5.1)$$

where we assume that $h > 0$. What do we mean when we say that the expression on the right-hand-side of (5.1) is an approximation of the derivative? For linear functions (5.1) is actually an exact expression for the derivative. For almost all other functions, (5.1) is not the exact derivative.

Let's compute the approximation error. We write a Taylor expansion of $f(x+h)$ about x , i.e.,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(\xi), \quad \xi \in (x, x+h). \quad (5.2)$$

For such an expansion to be valid, we assume that $f(x)$ has two continuous derivatives. The Taylor expansion (5.2) means that we can now replace the approximation (5.1) with

an exact formula of the form

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(\xi), \quad \xi \in (x, x+h). \quad (5.3)$$

Since this approximation of the derivative at x is based on the values of the function at x and $x+h$, the approximation (5.1) is called a **forward differencing** or **one-sided differencing**. The approximation of the derivative at x that is based on the values of the function at $x-h$ and x , i.e.,

$$f'(x) \approx \frac{f(x) - f(x-h)}{h},$$

is called a **backward differencing** (which is obviously also a one-sided differencing formula).

The second term on the right-hand-side of (5.3) is the **error term**. Since the approximation (5.1) can be thought of as being obtained by truncating this term from the exact formula (5.3), this error is called the **truncation error**. The small parameter h denotes the distance between the two points x and $x+h$. As this distance tends to zero, i.e., $h \rightarrow 0$, the two points approach each other and we expect the approximation (5.1) to improve. This is indeed the case if the truncation error goes to zero, which in turn is the case if $f''(\xi)$ is well defined in the interval $(x, x+h)$. The “speed” in which the error goes to zero as $h \rightarrow 0$ is called the **rate of convergence**. When the truncation error is of the order of $O(h)$, we say that the method is a **first order method**. We refer to a methods as a **p^{th} -order method** if the truncation error is of the order of $O(h^p)$.

It is possible to write more accurate formulas than (5.3) for the first derivative. For example, a more accurate approximation for the first derivative that is based on the values of the function at the points $f(x-h)$ and $f(x+h)$ is the **centered differencing** formula

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \quad (5.4)$$

Let's verify that this is indeed a more accurate formula than (5.1). Taylor expansions of the terms on the right-hand-side of (5.4) are

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(\xi_1), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\xi_2). \end{aligned}$$

Here $\xi_1 \in (x, x+h)$ and $\xi_2 \in (x-h, x)$. Hence

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{12}[f'''(\xi_1) + f'''(\xi_2)],$$

which means that the truncation error in the approximation (5.4) is

$$-\frac{h^2}{12}[f'''(\xi_1) + f'''(\xi_2)].$$

If the third-order derivative $f'''(x)$ is a continuous function in the interval $[x - h, x + h]$, then the intermediate value theorem implies that there exists a point $\xi \in (x - h, x + h)$ such that

$$f'''(\xi) = \frac{1}{2}[f'''(\xi_1) + f'''(\xi_2)].$$

Hence

$$f'(x) = \frac{f(x + h) - f(x - h)}{2h} - \frac{h^2}{6}f'''(\xi), \quad (5.5)$$

which means that the expression (5.4) is a second-order approximation of the first derivative.

In a similar way we can approximate the values of higher-order derivatives. For example, it is easy to verify that the following is a second-order approximation of the second derivative

$$f''(x) \approx \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}. \quad (5.6)$$

To verify the consistency and the order of approximation of (5.6) we expand

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2}f''(x) \pm \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\xi_{\pm}).$$

Here, $\xi_- \in (x - h, x)$ and $\xi_+ \in (x, x + h)$. Hence

$$\frac{f(x + h) - 2f(x) + f(x - h)}{h^2} = f''(x) + \frac{h^2}{24}(f^{(4)}(\xi_-) + f^{(4)}(\xi_+)) = f''(x) + \frac{h^2}{12}f^{(4)}(\xi),$$

where we assume that $\xi \in (x - h, x + h)$ and that $f(x)$ has four continuous derivatives in the interval. Hence, the approximation (5.6) is indeed a second-order approximation of the derivative, with a truncation error that is given by

$$-\frac{h^2}{12}f^{(4)}(\xi), \quad \xi \in (x - h, x + h).$$

5.2 Differentiation Via Interpolation

In this section we demonstrate how to generate differentiation formulas by differentiating an interpolant. The idea is straightforward: The first step is to construct an interpolating polynomial from the data. An approximation of the derivative at any point can then be obtained by directly differentiating the interpolant.

We follow this procedure and assume that $f(x_0), \dots, f(x_n)$ are given. The Lagrange form of the interpolation polynomial through these points is

$$Q_n(x) = \sum_{j=0}^n f(x_j)l_j(x).$$

Here we simplify the notation and replace $l_i^n(x)$ which is the notation we used in Section 3.5 by $l_i(x)$. According to the error analysis of Section 3.7 we know that the interpolation error is

$$f(x) - Q_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_n) \prod_{j=0}^n (x - x_j),$$

where $\xi_n \in (\min(x, x_0, \dots, x_n), \max(x, x_0, \dots, x_n))$. Since here we are assuming that the points x_0, \dots, x_n are fixed, we would like to emphasize the dependence of ξ_n on x and hence replace the ξ_n notation by ξ_x . We that have:

$$f(x) = \sum_{j=0}^n f(x_j)l_j(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x)w(x), \quad (5.7)$$

where

$$w(x) = \prod_{i=0}^n (x - x_i).$$

Differentiating the interpolant (5.7):

$$f'(x) = \sum_{j=0}^n f(x_j)l'_j(x) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_x)w'(x) + \frac{1}{(n+1)!} w(x) \frac{d}{dx} f^{(n+1)}(\xi_x). \quad (5.8)$$

We now assume that x is one of the interpolation points, i.e., $x \in \{x_0, \dots, x_n\}$, say x_k , so that

$$f'(x_k) = \sum_{j=0}^n f(x_j)l'_j(x_k) + \frac{1}{(n+1)!} f^{(n+1)}(\xi_{x_k})w'(x_k). \quad (5.9)$$

Now,

$$w'(x) = \sum_{i=0}^n \prod_{\substack{j=0 \\ j \neq i}}^n (x - x_j) = \sum_{i=0}^n [(x - x_0) \cdot \dots \cdot (x - x_{i-1})(x - x_{i+1}) \cdot \dots \cdot (x - x_n)].$$

Hence, when $w'(x)$ is evaluated at an interpolation point x_k , there is only one term in $w'(x)$ that does not vanish, i.e.,

$$w'(x_k) = \prod_{\substack{j=0 \\ j \neq k}}^n (x_k - x_j).$$

The numerical differentiation formula, (5.9), then becomes

$$f'(x_k) = \sum_{j=0}^n f(x_j)l'_j(x_k) + \frac{1}{(n+1)!}f^{(n+1)}(\xi_{x_k}) \prod_{\substack{j=0 \\ j \neq k}}^n (x_k - x_j). \quad (5.10)$$

We refer to the formula (5.10) as a **differentiation by interpolation** algorithm.

Example 5.1

We demonstrate how to use the differentiation by integration formula (5.10) in the case where $n = 1$ and $k = 0$. This means that we use two interpolation points $(x_0, f(x_0))$ and $(x_1, f(x_1))$, and want to approximate $f'(x_0)$. The Lagrange interpolation polynomial in this case is

$$Q_1(x) = f(x_0)l_0(x) + f(x_1)l_1(x),$$

where

$$l_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

Hence

$$l'_0(x) = \frac{1}{x_0 - x_1}, \quad l'_1(x) = \frac{1}{x_1 - x_0}.$$

We thus have

$$Q'_1(x_0) = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0} + \frac{1}{2}f''(\xi)(x_0 - x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} - \frac{1}{2}f''(\xi)(x_1 - x_0). \quad (5.11)$$

Here, we simplify the notation and assume that $\xi \in (x_0, x_1)$. This result is not surprising. After all, we ended up with an approximation of the derivative that is nothing but the slope of the line that connects $(x_0, f(x_0))$ and $(x_1, f(x_1))$. If we now let $x_1 = x_0 + h$, then

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2}f''(\xi), \quad \xi \in (x_0, x_0 + h).$$

which is the (first-order) forward differencing approximation of $f'(x_0)$, (5.3).

Note that nowhere in the derivation we assumed that $x_0 < x_1$. For example, if we set $x_1 = x_0 - h$ in (5.11), we get the first-order backward differencing approximation:

$$f'(x_0) = \frac{f(x_0 - h) - f(x_0)}{h} + \frac{1}{2}f''(\xi)(-h) = \frac{f(x_0) - f(x_0 - h)}{h} + \frac{h}{2}f''(\xi),$$

with $\xi \in (x_0 - h, x_0)$.

Example 5.2

We repeat Example 5.1 with $n = 2$ and $k = 0$. This time we have a quadratic interpolant which we write in the Lagrange form:

$$Q_2(x) = f(x_0)l_0(x) + f(x_1)l_1(x) + f(x_2)l_2(x),$$

with

$$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \quad l_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

Hence

$$l'_0(x) = \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}, \quad l'_1(x) = \frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)}, \quad l'_2(x) = \frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}.$$

Evaluating $l'_j(x)$ for $j = 1, 2, 3$ at x_0 we have

$$l'_0(x_0) = \frac{2x_0 - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}, \quad l'_1(x_0) = \frac{x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)}, \quad l'_2(x_0) = \frac{x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}$$

Hence

$$\begin{aligned} Q'_2(x_0) &= f(x_0) \frac{2x_0 - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \\ &\quad + f(x_2) \frac{x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} + \frac{1}{6} f^{(3)}(\xi)(x_0 - x_1)(x_0 - x_2). \end{aligned} \quad (5.12)$$

Here, we assume $\xi \in (x_0, x_2)$. For $x_0 = x$, $x_1 = x + h$, and $x_2 = x + 2h$, equation (5.12) becomes

$$\begin{aligned} f'(x) \approx Q'_2(x) &= -f(x) \frac{3}{2h} + f(x+h) \frac{2}{h} + f(x+2h) \left(-\frac{1}{2h} \right) + \frac{f'''(\xi)}{3} h^2 \\ &= \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \frac{f'''(\xi)}{3} h^2, \end{aligned}$$

which is a one-sided, second-order approximation of the first derivative.

Example 5.3

If we repeat Example 5.2 with $n = 2$ while approximating the derivative at x_1 , we get:

$$\begin{aligned} Q'_2(x_1) &= f(x_0) \frac{2x_1 - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{2x_1 - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \\ &\quad + f(x_2) \frac{2x_1 - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} + \frac{1}{6} f^{(3)}(\xi)(x_1 - x_0)(x_1 - x_2), \end{aligned} \quad (5.13)$$

with $\xi \in (x_0, x_2)$. We then set $x_0 = x - h$, $x_1 = x$, and $x_2 = x + h$ in (5.13), which leads to

$$\begin{aligned} f'(x) \approx Q'_2(x) &= f(x-h) \frac{-h}{-h(-2h)} + f(x) \cdot 0 + f(x+h) \frac{h}{2h \cdot h} + \frac{1}{6} f'''(\xi) h(-h) = \\ &= \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6} f'''(\xi) h^2, \end{aligned}$$

which is precisely the second-ordered centered approximation of the first-derivative (5.5).

In fact, instead of repeating the previous example and approximating the derivative at x_1 , the same result can be obtained by noticing that equation (5.12) assume no particular order on the interpolation points. Hence, we can substitute $x_0 = x$, $x_1 = x-h$, $x_2 = x+h$ directly in equation (5.12) and directly obtain the centered approximation:

$$f'(x) \approx Q'_2(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{1}{6} f'''(\xi) h^2.$$

5.3 The Method of Undetermined Coefficients

In this section we present the **method of undetermined coefficients**, which is a very practical way for generating approximations of derivatives (as well as other quantities as we shall see, e.g., when we discuss integration).

Assume, for example, that we are interested in finding an approximation of the second derivative $f''(x)$ that is based on the values of the function at three equally spaced points, $f(x-h)$, $f(x)$, $f(x+h)$, i.e.,

$$f''(x) \approx Af(x+h) + Bf(x) + Cf(x-h). \quad (5.14)$$

The coefficients A , B , and C are to be determined in such a way that this linear combination is indeed an approximation of the second derivative. The Taylor expansions of the terms $f(x \pm h)$ are

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2} f''(x) \pm \frac{h^3}{6} f'''(x) + \frac{h^4}{24} f^{(4)}(\xi_{\pm}), \quad (5.15)$$

where (assuming that $h > 0$)

$$x - h \leq \xi_- \leq x \leq \xi_+ \leq x + h.$$

Using the expansions in (5.15) we can rewrite (5.14) as

$$\begin{aligned} f''(x) &\approx Af(x+h) + Bf(x) + Cf(x-h) \\ &= (A+B+C)f(x) + h(A-C)f'(x) + \frac{h^2}{2}(A+C)f''(x) \\ &\quad + \frac{h^3}{6}(A-C)f^{(3)}(x) + \frac{h^4}{24}[Af^{(4)}(\xi_+) + Cf^{(4)}(\xi_-)]. \end{aligned} \quad (5.16)$$

Equating the coefficients of $f(x)$, $f'(x)$, and $f''(x)$ on both sides of (5.16) we obtain the linear system

$$\begin{cases} A + B + C = 0, \\ A - C = 0, \\ A + C = \frac{2}{h^2}. \end{cases} \quad (5.17)$$

The system (5.17) has the unique solution:

$$A = C = \frac{1}{h^2}, \quad B = -\frac{2}{h^2}.$$

In this particular case, since A and C are equal to each other, the coefficient of $f^{(3)}(x)$ on the right-hand-side of (5.16) also vanishes and we end up with

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{h^2}{24}[f^{(4)}(\xi_+) + f^{(4)}(\xi_-)].$$

We note that the last two terms can be combined into one using an intermediate values theorem (assuming that $f(x)$ has four continuous derivatives), i.e.,

$$\frac{h^2}{24}[f^{(4)}(\xi_+) + f^{(4)}(\xi_-)] = \frac{h^2}{12}f^{(4)}(\xi), \quad \xi \in (x-h, x+h).$$

Hence we obtain the familiar second-order approximation of the second derivative:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{h^2}{12}f^{(4)}(\xi).$$

In terms of an algorithm, the method of undetermined coefficients follows what was just demonstrated in the example:

1. Assume that the derivative can be written as a linear combination of the values of the function at certain points.
2. Write the Taylor expansions of the function at the approximation points.
3. Equate the coefficients of the function and its derivatives on both sides.

The only question that remains open is how many terms should we use in the Taylor expansion. This question has, unfortunately, no simple answer. In the example, we have already seen that even though we used data that is taken from three points, we could satisfy four equations. In other words, the coefficient of the third-derivative vanished as well. If we were to stop the Taylor expansions at the third derivative instead of at the fourth derivative, we would have missed on this cancellation, and would have mistakenly concluded that the approximation method is only first-order accurate. The number of terms in the Taylor expansion should be sufficient to rule out additional cancellations. In other words, one should truncate the Taylor series after the leading term in the error has been identified.

5.4 Richardson's Extrapolation

Richardson's extrapolation can be viewed as a general procedure for improving the accuracy of approximations when the structure of the error is known. While we study it here in the context of numerical differentiation, it is by no means limited only to differentiation and we will get back to it later on when we study methods for numerical integration.

We start with an example in which we show how to turn a second-order approximation of the first derivative into a fourth order approximation of the same quantity. We already know that we can write a second-order approximation of $f'(x)$ given its values in $f(x \pm h)$. In order to improve this approximation we will need some more insight on the internal structure of the error. We therefore start with the Taylor expansions of $f(x \pm h)$ about the point x , i.e.,

$$\begin{aligned} f(x+h) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(x)}{k!} h^k, \\ f(x-h) &= \sum_{k=0}^{\infty} \frac{(-1)^k f^{(k)}(x)}{k!} h^k. \end{aligned}$$

Hence

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \left[\frac{h^2}{3!} f^{(3)}(x) + \frac{h^4}{5!} f^{(5)}(x) + \dots \right]. \quad (5.18)$$

We rewrite (5.18) as

$$L = D(h) + e_2 h^2 + e_4 h^4 + \dots, \quad (5.19)$$

where L denotes the quantity that we are interested in approximating, i.e.,

$$L = f'(x),$$

and $D(h)$ is the approximation, which in this case is

$$D(h) = \frac{f(x+h) - f(x-h)}{2h}.$$

The error is

$$E = e_2 h^2 + e_4 h^4 + \dots$$

where e_i denotes the coefficient of h^i in (5.18). The important property of the coefficients e_i 's is that they do not depend on h . We note that the formula

$$L \approx D(h),$$

is a second-order approximation of the first-derivative which is based on the values of $f(x)$ at $x \pm h$. We assume here that in general $e_i \neq 0$. In order to improve the approximation of L our strategy will be to eliminate the term e_2h^2 from the error. How can this be done? one possibility is to write another approximation that is based on the values of the function at different points. For example, we can write

$$L = D(2h) + e_2(2h)^2 + e_4(2h)^4 + \dots \quad (5.20)$$

This, of course, is still a second-order approximation of the derivative. However, the idea is to combine (5.19) with (5.20) such that the h^2 term in the error vanishes. Indeed, subtracting the following equations from each other

$$\begin{aligned} 4L &= 4D(h) + 4e_2h^2 + 4e_4h^4 + \dots, \\ L &= D(2h) + 4e_2h^2 + 16e_4h^4 + \dots, \end{aligned}$$

we have

$$L = \frac{4D(h) - D(2h)}{3} - 4e_4h^4 + \dots$$

In other words, a fourth-order approximation of the derivative is

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + O(h^4). \quad (5.21)$$

Note that (5.21) improves the accuracy of the approximation (5.18) by using more points.

This process can be repeated over and over as long as the structure of the error is known. For example, we can write (5.21) as

$$L = S(h) + a_4h^4 + a_6h^6 + \dots \quad (5.22)$$

where

$$S(h) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}.$$

Equation (5.22) can be turned into a sixth-order approximation of the derivative by eliminating the term a_4h^4 . We carry out such a procedure by writing

$$L = S(2h) + a_4(2h)^4 + a_6(2h)^6 + \dots \quad (5.23)$$

Combining (5.23) with (5.22) we end up with a sixth-order approximation of the derivative:

$$L = \frac{16S(h) - S(2h)}{15} + O(h^6).$$

Remarks.

1. In (5.20), instead of using $D(2h)$, it is possible to use other approximations, e.g., $D(h/2)$. If this is what is done, instead of (5.21) we would get a fourth-order approximation of the derivative that is based on the values of f at $x - h, x - h/2, x + h/2, x + h$.
2. Once again we would like to emphasize that Richardson's extrapolation is a general procedure for improving the accuracy of numerical approximations that can be used when the structure of the error is known. It is not specific for numerical differentiation.

6 Numerical Integration

6.1 Basic Concepts

In this chapter we are going to explore various ways for approximating the integral of a function over a given domain. There are various reasons as of why such approximations can be useful. First, not every function can be analytically integrated. Second, even if a closed integration formula exists, it might still not be the most efficient way of calculating the integral. In addition, it can happen that we need to integrate an unknown function, in which only some samples of the function are known.

In order to gain some insight on numerical integration, it is natural to review Riemann integration, a framework that can be viewed as an approach for approximating integrals. We assume that $f(x)$ is a bounded function defined on $[a, b]$ and that $\{x_0, \dots, x_n\}$ is a partition (P) of $[a, b]$. For each i we let

$$M_i(f) = \sup_{x \in [x_{i-1}, x_i]} f(x),$$

and

$$m_i(f) = \inf_{x \in [x_{i-1}, x_i]} f(x),$$

Letting $\Delta x_i = x_i - x_{i-1}$, **the upper (Darboux) sum** of $f(x)$ with respect to the partition P is defined as

$$U(f, P) = \sum_{i=1}^n M_i \Delta x_i, \tag{6.1}$$

while **the lower (Darboux) sum** of $f(x)$ with respect to the partition P is defined as

$$L(f, P) = \sum_{i=1}^n m_i \Delta x_i. \tag{6.2}$$

The upper integral of $f(x)$ on $[a, b]$ is defined as

$$U(f) = \inf(U(f, P)),$$

and **the lower integral** of $f(x)$ is defined as

$$L(f) = \sup(L(f, P)),$$

where both the infimum and the supremum are taken over all possible partitions, P , of the interval $[a, b]$. If the upper and lower integral of $f(x)$ are equal to each other, their

common value is denoted by $\int_a^b f(x)dx$ and is referred to as **the Riemann integral** of $f(x)$.

For the purpose of the present discussion we can think of the upper and lower Darboux sums (6.1), (6.2), as two approximations of the integral (assuming that the function is indeed integrable). Of course, these sums are not defined in the most convenient way for an approximation algorithm. This is because we need to find the extrema of the function in every subinterval. Finding the extrema of the function, may be a complicated task on its own, which we would like to avoid.

A simpler approach for approximating the value of $\int_a^b f(x)dx$ would be to compute the product of the value of the function at one of the end-points of the interval by the length of the interval. In case we choose the end-point where the function is evaluated to be $x = a$, we obtain

$$\int_a^b f(x)dx \approx f(a)(b - a). \quad (6.3)$$

This approximation (6.3) is called **the rectangular method** (see Figure 6.1). Numerical integration formulas are also referred to as **integration rules** or **quadratures**, and hence we can refer to (6.3) as the rectangular rule or the rectangular quadrature. The points x_0, \dots, x_n that are used in the quadrature formula are called **quadrature points**.

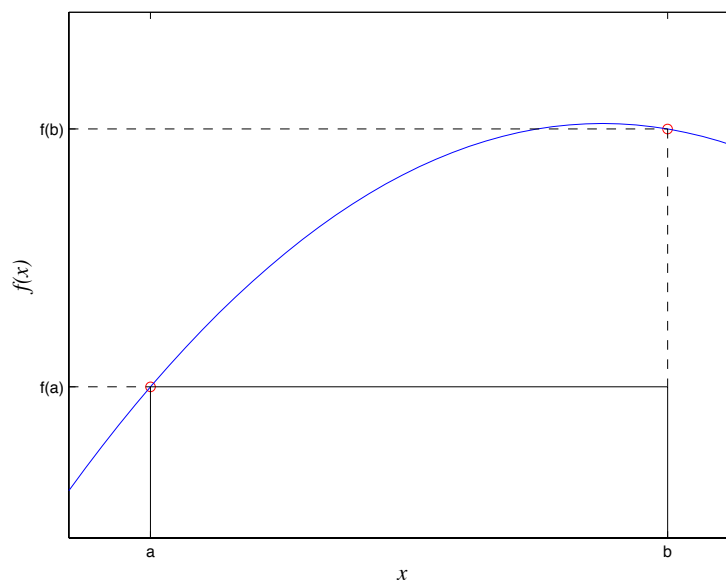


Figure 6.1: A rectangular quadrature

A variation on the rectangular rule is **the midpoint rule**. Similarly to the rectangular rule, we approximate the value of the integral $\int_a^b f(x)dx$ by multiplying the length

of the interval by the value of the function at one point. Only this time, we replace the value of the function at an endpoint, by the value of the function at the center point $\frac{1}{2}(a+b)$, i.e.,

$$\int_a^b f(x)dx \approx (b-a)f\left(\frac{a+b}{2}\right). \quad (6.4)$$

(see also Fig 6.2). As we shall see below, the midpoint quadrature (6.4) is a more accurate quadrature than the rectangular rule (6.3).

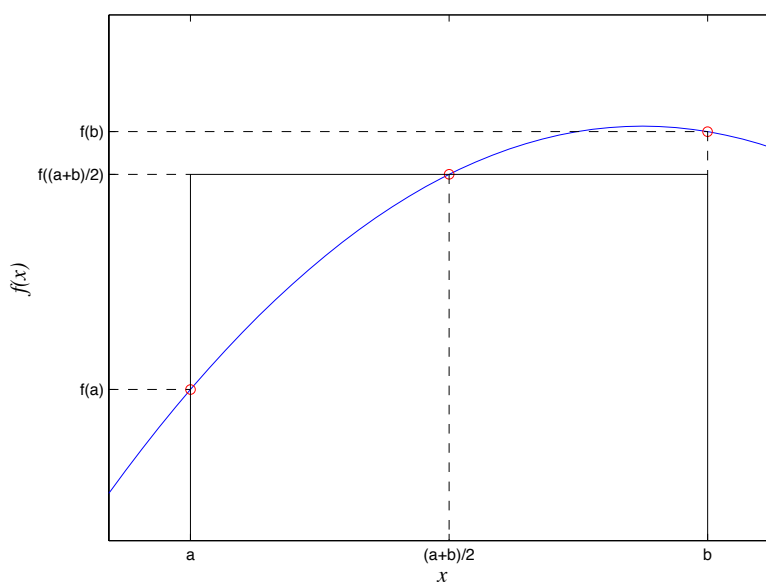


Figure 6.2: A midpoint quadrature

In order to compute the quadrature error for the midpoint rule (6.4), we consider the primitive function $F(x)$,

$$F(x) = \int_a^x f(s)ds,$$

and expand

$$\begin{aligned} \int_a^{a+h} f(s)ds &= F(a+h) = F(a) + hF'(a) + \frac{h^2}{2}F''(a) + \frac{h^3}{6}F'''(a) + O(h^4) \\ &= hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{6}f''(a) + O(h^4) \end{aligned} \quad (6.5)$$

If we let $b = a + h$, we have (expanding $f(a + h/2)$) for the quadrature error, E ,

$$E = \int_a^{a+h} f(s)ds - hf\left(a + \frac{h}{2}\right) = hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{6}f''(a) + O(h^4) \\ - h\left[f(a) + \frac{h}{2}f'(a) + \frac{h^2}{8}f''(a) + O(h^3)\right],$$

which means that the error term is of order $O(h^3)$. Having an error of order h^3 does not mean that this is a third-order method. In our case, the parameter h equals to $b - a$. It is not a parameter that should be varied as a small value that goes to zero. It is fixed. The error of the midpoint method is of the order of $O((b - a)^3)$. Unfortunately, these calculations cannot directly provide us with an accurate estimate of the error. This is the case since when truncating two Taylor approximations, we are left with an error terms that are evaluated at two (generally different) intermediate points. Hence we cannot directly combine the error term $\frac{1}{6}h^3f''(\xi_1)$ with $-\frac{1}{8}h^3f''(\xi_2)$. This can still be done, but we have to use a better approach.

The main difficulty in evaluating the difference between the exact value, $\int_a^b f(x)dx$, and its midpoint rule approximation, $(b - a)f\left(\frac{a+b}{2}\right)$, is due to having an integral in one term and no integral in the second term. The approach will be to replace the midpoint approximation with an integral expression. Indeed, if we denote the midpoint by c , i.e.,

$$c = \frac{a + b}{2},$$

then the tangent line to $f(x)$ at $x = c$ is given by

$$P_1(x) = f(c) + f'(c)(x - c).$$

Clearly,

$$\int_a^b P_1(x)dx = (b - a)f(c),$$

and hence

$$\int_a^b f(x)dx - (b - a)f\left(\frac{a + b}{2}\right) = \int_a^b (f(x) - P_1(x))dx.$$

To estimate the difference between $f(x)$ and $P_1(x)$ we can expand $f(x)$ around $x = c$. Assuming that $x \in [a, b]$, we have

$$f(x) = f(c + (x - c)) = f(c) + f'(c)(x - c) + \frac{1}{2}f''(\xi)(x - c)^2, \quad \xi \in (a, b).$$

Hence

$$\int_a^b (f(x) - P_1(x))dx = \int_a^b \frac{1}{2}f''(\xi_x)(x - c)^2dx.$$

In view of the midvalue theorem for integrals, the last integral can be replaced by

$$\frac{1}{2}f''(\xi) \int_a^b (x-c)^2 dx = \frac{1}{24}(b-a)^3 f''(\xi), \quad a < \xi < b. \quad (6.6)$$

Remark. Throughout this section we assumed that all functions we are interested in integrating are actually integrable in the domain of interest. We also assumed that they are bounded and that they are defined at every point, so that whenever we need to evaluate a function at a point, we can do it. We will go on and use these assumptions throughout the chapter.

6.2 Integration via Interpolation

One direct way of obtaining quadratures from given samples of a function is by integrating an interpolant. As always, our goal is to evaluate $I = \int_a^b f(x)dx$. We assume that the values of the function $f(x)$ are given at $n+1$ points: $x_0, \dots, x_n \in [a, b]$. Note that we do not require the first point x_0 to be equal to a , and the same holds for the right side of the interval. Given the values $f(x_0), \dots, f(x_n)$, we can write the interpolating polynomial of degree $\leq n$, which in the Lagrange form is

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

with

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}, \quad 0 \leq i \leq n.$$

The integral of $f(x)$ can then be approximated by the integral of $P_n(x)$, i.e.,

$$\int_a^b f(x)dx \approx \int_a^b P_n(x)dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x)dx = \sum_{i=0}^n A_i f(x_i). \quad (6.7)$$

The quadrature coefficients A_i in (6.7) are given by

$$A_i = \int_a^b l_i(x)dx. \quad (6.8)$$

Note that if we want to integrate several different functions, and use their values at the same points (x_0, \dots, x_n) , the quadrature coefficients (6.8) should be computed only once, since they do not depend on the function that is being integrated. If we change the interpolation/integration points, then we must recompute the quadrature coefficients.

For equally spaced points, x_0, \dots, x_n , a numerical integration formula of the form

$$\int_a^b f(x)dx \approx \sum_{i=0}^n A_i f(x_i), \quad (6.9)$$

is called a **Newton-Cotes formula**.

Example 6.1

We let $n = 1$ and consider two interpolation points which we set as

$$x_0 = a, \quad x_1 = b.$$

In this case

$$l_0(x) = \frac{b-x}{b-a}, \quad l_1(x) = \frac{x-a}{b-a}.$$

Hence

$$A_0 = \int_a^b l_0(x) dx = \int_a^b \frac{b-x}{b-a} dx = \frac{b-a}{2}.$$

Similarly,

$$A_1 = \int_a^b l_1(x) dx = \int_a^b \frac{x-a}{b-a} dx = \frac{b-a}{2} = A_0.$$

The resulting quadrature is the so-called **trapezoidal rule**,

$$\int_a^b dx \approx \frac{b-a}{2} [f(a) + f(b)], \quad (6.10)$$

(see Figure 6.3).

We can now use the interpolation error to compute the error in the quadrature (6.10). The interpolation error is

$$f(x) - P_1(x) = \frac{1}{2} f''(\xi_x)(x-a)(x-b), \quad \xi_x \in (a, b).$$

We recall that according to the midvalue theorem for integrals, if $u(x)$ and $v(x)$ are continuous on $[a, b]$ and if $v \geq 0$, then there exists $\xi \in (a, b)$ such that

$$\int_a^b u(x)v(x)dx = u(\xi) \int_a^b v(x)dx.$$

Hence, the interpolation error is given by

$$E = \int_a^b \frac{1}{2} f''(\xi_x)(x-a)(x-b) dx = \frac{f''(\xi)}{2} \int_a^b (x-a)(x-b) dx = -\frac{f''(\xi)}{12} (b-a)^3, \quad (6.11)$$

with $\xi \in (a, b)$.

Remarks.

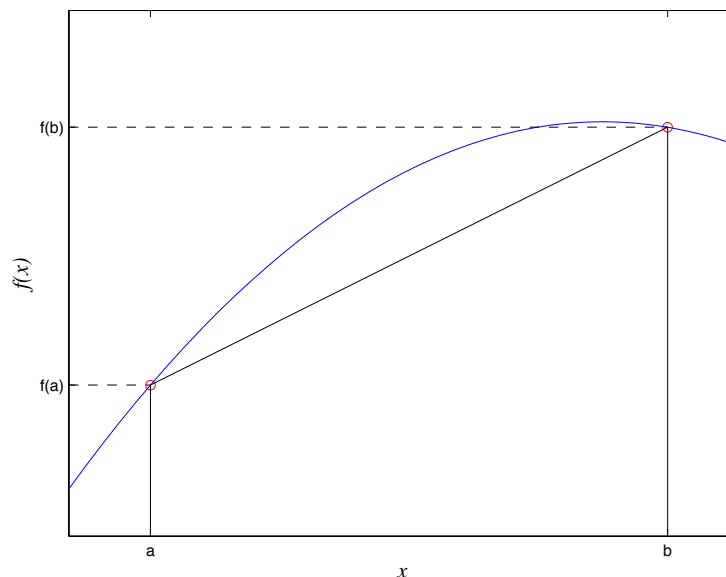


Figure 6.3: A trapezoidal quadrature

1. We note that the quadratures (6.7), (6.8), are exact for polynomials of degree $\leq n$. For if $p(x)$ is a polynomial of degree $\leq n$, it can be written as

$$p(x) = \sum_{i=0}^n p(x_i) l_i(x).$$

(Two polynomials of degree $\leq n$ that agree with each other at $n + 1$ points must be identical). Hence

$$\int_a^b p(x) dx = \sum_{i=0}^n p(x_i) \int_a^b l_i(x) dx = \sum_{i=0}^n A_i p(x_i).$$

2. As of the opposite direction. Assume that the quadrature

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i),$$

is exact for all polynomials of degree $\leq n$. We know that

$$\deg(l_j(x)) = n,$$

and hence

$$\int_a^b l_j(x) dx = \sum_{i=0}^n A_i l_j(x_i) = \sum_{i=0}^n A_i \delta_{ij} = A_j.$$

This means that the quadrature coefficients must be given by

$$A_j = \int_a^b l_j(x) dx.$$

6.3 Composite Integration Rules

In a composite quadrature, we divide the interval into subintervals and apply an integration rule to each subinterval. We demonstrate this idea with a couple of examples.

Example 6.2

Consider the points

$$a = x_0 < x_1 < \cdots < x_n = b.$$

The **composite trapezoidal rule** is obtained by applying the trapezoidal rule in each subinterval $[x_{i-1}, x_i]$, $i = 1, \dots, n$, i.e.,

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx \approx \frac{1}{2} \sum_{i=1}^n (x_i - x_{i-1}) [f(x_{i-1}) + f(x_i)], \quad (6.12)$$

(see Figure 6.4).

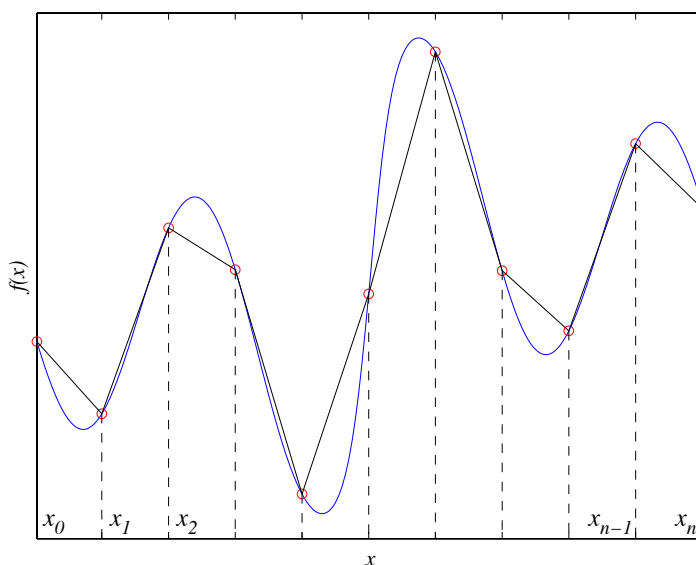


Figure 6.4: A composite trapezoidal rule

A particular case is when these points are uniformly spaced, i.e., when all intervals have an equal length. For example, if

$$x_i = a + ih,$$

where

$$h = \frac{b-a}{n},$$

then

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(a+ih) + f(b) \right] = h \sum_{i=0}^{n-1} f(a+ih). \quad (6.13)$$

The notation of a sum with two primes, \sum'' , means that we sum over all the terms with the exception of the first and last terms that are being divided by 2.

We can also compute the error term as a function of the distance between neighboring points, h . We know from (6.11) that in every subinterval the quadrature error is

$$-\frac{h^3}{12} f''(\xi_x).$$

Hence, the overall error is obtained by summing over n such terms:

$$\sum_{i=1}^n -\frac{h^3}{12} f''(\xi_i) = -\frac{h^3 n}{12} \left[\frac{1}{n} \sum_{i=1}^n f''(\xi_i) \right].$$

Here, we use the notation ξ_i to denote an intermediate point that belongs to the i^{th} interval. Let

$$M = \frac{1}{n} \sum_{i=1}^n f''(\xi_i).$$

Clearly

$$\min_{x \in [a,b]} f''(x) \leq M \leq \max_{x \in [a,b]} f''(x)$$

If we assume that $f''(x)$ is continuous in $[a, b]$ (which we anyhow do in order for the interpolation error formula to be valid) then there exists a point $\xi \in [a, b]$ such that

$$f''(\xi) = M.$$

Hence (recalling that $(b-a)/n = h$, we have

$$E = -\frac{(b-a)h^2}{12} f''(\xi), \quad \xi \in [a, b]. \quad (6.14)$$

This means that the composite trapezoidal rule is second-order accurate.

Example 6.3

In the interval $[a, b]$ we assume n subintervals and let

$$h = \frac{b - a}{n}.$$

The quadrature points are

$$x_j = a + \left(j - \frac{1}{2}\right) h, \quad j = 1, 2, \dots, n.$$

The composite midpoint rule is given by applying the midpoint rule (6.4) in every subinterval, i.e.,

$$\int_a^b f(x) dx \approx h \sum_{j=1}^n f(x_j). \quad (6.15)$$

Equation (6.15) is known as **the composite midpoint rule**.

In order to obtain the quadrature error in the approximation (6.15) we recall that in each subinterval the error is given according to (6.6), i.e.,

$$E_j = \frac{h^3}{24} f''(\xi_j), \quad \xi_j \in \left(x_j - \frac{h}{2}, x_j + \frac{h}{2}\right).$$

Hence

$$E = \sum_{j=1}^n E_j = \frac{h^3}{24} \sum_{j=1}^n f''(\xi_j) = \frac{h^3}{24} n \left[\frac{1}{n} \sum_{j=1}^n f''(\xi_j) \right] = \frac{h^2(b-a)}{24} f''(\xi), \quad (6.16)$$

where $\xi \in (a, b)$. This means that the composite midpoint rule is also second-order accurate (just like the composite trapezoidal rule).

6.4 Additional Integration Techniques

6.4.1 The method of undetermined coefficients

The methods of undetermined coefficients for deriving quadratures is the following:

1. Select the quadrature points.
2. Write a quadrature as a linear combination of the values of the function at the chosen quadrature points.
3. Determine the coefficients of the linear combination by requiring that the quadrature is *exact* for as many polynomials as possible from the the ordered set $\{1, x, x^2, \dots\}$.

We demonstrate this technique with the following example.

Example 6.4

Problem: Find a quadrature of the form

$$\int_0^1 f(x)dx \approx A_0f(0) + A_1f\left(\frac{1}{2}\right) + A_2f(1),$$

that is exact for all polynomials of degree ≤ 2 .

Solution: Since the quadrature has to be exact for all polynomials of degree ≤ 2 , it has to be exact for the polynomials 1, x , and x^2 . Hence we obtain the system of linear equations

$$\begin{aligned} 1 &= \int_0^1 1dx = A_0 + A_1 + A_2, \\ \frac{1}{2} &= \int_0^1 xdx = \frac{1}{2}A_1 + A_2, \\ \frac{1}{3} &= \int_0^1 x^2dx = \frac{1}{4}A_1 + A_2. \end{aligned}$$

Therefore, $A_0 = A_2 = \frac{1}{6}$ and $A_1 = \frac{2}{3}$, and the desired quadrature is

$$\int_0^1 f(x)dx \approx \frac{f(0) + 4f\left(\frac{1}{2}\right) + f(1)}{6}. \quad (6.17)$$

Since the resulting formula (6.17) is linear, its being exact for 1, x , and x^2 , implies that it is exact for any polynomial of degree ≤ 2 . In fact, we will show in Section 6.5.1 that this approximation is actually exact for polynomials of degree ≤ 3 .

6.4.2 Change of an interval

Suppose that we have a quadrature formula on the interval $[c, d]$ of the form

$$\int_c^d f(t)dt \approx \sum_{i=0}^n A_i f(t_i). \quad (6.18)$$

We would like to use (6.18) to find a quadrature on the interval $[a, b]$, that approximates for

$$\int_a^b f(x)dx.$$

The mapping between the intervals $[c, d] \rightarrow [a, b]$ can be written as a linear transformation of the form

$$\lambda(t) = \frac{b-a}{d-c}t + \frac{ad-bc}{d-c}.$$

Hence

$$\int_a^b f(x)dx = \frac{b-a}{d-c} \int_c^d f(\lambda(t))dt \approx \frac{b-a}{d-c} \sum_{i=0}^n A_i f(\lambda(t_i)).$$

This means that

$$\int_a^b f(x)dx \approx \frac{b-a}{d-c} \sum_{i=0}^n A_i f\left(\frac{b-a}{d-c}t_i + \frac{ad-bc}{d-c}\right). \quad (6.19)$$

We note that if the quadrature (6.18) was exact for polynomials of degree m , so is (6.19).

Example 6.5

We want to write the result of the previous example

$$\int_0^1 f(x)dx \approx \frac{f(0) + 4f\left(\frac{1}{2}\right) + f(1)}{6},$$

as a quadrature on the interval $[a, b]$. According to (6.19)

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]. \quad (6.20)$$

The approximation (6.20) is known as the **Simpson quadrature**.

6.5 Simpson's Integration

In the last example we obtained Simpson's quadrature (6.20). An alternative derivation is the following: start with a polynomial $Q_2(x)$ that interpolates $f(x)$ at the points a , $(a+b)/2$, and b . Then approximate

$$\begin{aligned} \int_a^b f(x)dx &\approx \int_a^b \left[\frac{(x-c)(x-b)}{(a-c)(a-b)} f(a) + \frac{(x-a)(x-b)}{(c-a)(c-b)} f(c) + \frac{(x-a)(x-c)}{(b-a)(b-c)} f(b) \right] dx \\ &= \dots = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right], \end{aligned}$$

which is Simpson's rule (6.20). Figure 6.5 demonstrates this process of deriving Simpson's quadrature for the specific choice of approximating $\int_1^3 \sin x dx$.

6.5.1 The quadrature error

It turns out that Simpson's quadrature is exact for polynomials of degree ≤ 3 and not only for polynomials of degree ≤ 2 , as expected by the way it was constructed. We will obtain this result by studying the error term.

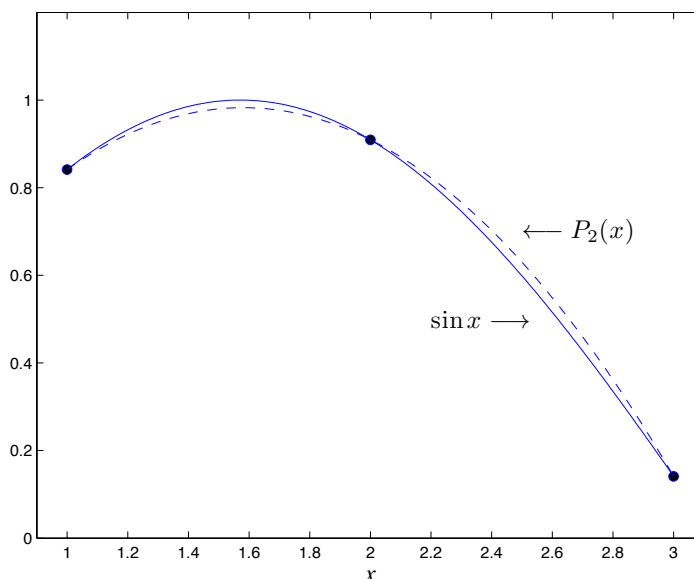


Figure 6.5: An example of Simpson's quadrature. The approximation of $\int_1^3 \sin x dx$ is obtained by integrating the quadratic interpolant $Q_2(x)$ over $[1, 3]$

In order to derive the error term for Simpson's method, we discuss an error analysis technique that is valid for quadratures that are obtained through integration. In all such cases, the quadrature error is the difference between the integral of the function and the integral of its interpolant, i.e.,

$$E = \int_a^b (f(t) - p_n(t)) dt = \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \omega(t) dt, \quad (6.21)$$

where

$$\omega(t) = \prod_{i=0}^n (t - t_i).$$

It $\omega(t)$ is always non-negative or non-positive between a and b , then according to the midvalue theorem for integrals, the error in (6.21) becomes

$$E = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b \omega(t) dt, \quad \xi \in (a, b).$$

Examples for such cases are the trapezoidal rule for which

$$E = \frac{f''(\xi)}{12} (b-a)^3,$$

and the rectangle rule for which

$$E = \frac{f'(\xi)}{1} \int_a^b (t-a) dt = \frac{f'(\xi)}{2} (b-a)^2.$$

Another case which is rather easy to analyze is the case in which

$$\int_a^b \omega(t) dt = 0. \quad (6.22)$$

Examples for the case in (6.22) include the midpoint rule for which

$$\int_a^b \omega(t) dt = \int_a^b \left(t - \frac{a+b}{2} \right) dt = 0,$$

and Simpson's rule for which

$$\int_a^b \omega(t) dt = \int_a^b (t-a) \left(t - \frac{a+b}{2} \right) (t-b) dt = 0.$$

In this case, we can add another interpolation point without changing the integral of the interpolant. This is the case since we replace $f(x)$ by $p_n(x)$ and integrate

$$\int_a^b f(t) dt \approx \int_a^b p_n(x) dx.$$

Adding an arbitrary interpolation point, x_{n+1} , to $p_n(x)$ turns it into an interpolating polynomial of a higher order, $p_{n+1}(x)$, that is given by

$$p_{n+1}(x) = p_n(x) + f[x_0, \dots, x_{n+1}] \omega(x). \quad (6.23)$$

Since $\int_a^b \omega(x) dx = 0$, when integrating (6.23) in order to obtain a quadrature, we observe that

$$\int_a^b f(t) dt \approx \int_a^b p_{n+1}(x) dx = \int_a^b p_n(x) dx,$$

so the original quadrature does not change by adding an arbitrary interpolation point.

We now have all the required tools in order to derive a quadrature for Simpson's method. Since in this case $\int_a^b \omega(t) dt = 0$, we add to $a, \frac{a+b}{2}, b$ an arbitrary interpolation point which we choose as $\frac{a+b}{2}$ again. The function $\omega(t)$ becomes

$$\omega(t) = (t-a) \left(t - \frac{a+b}{2} \right)^2 (t-b).$$

Hence, for $t \in [a, b]$, our new $\omega(t)$ satisfies $\omega(t) \leq 0$. By the midvalue theorem for integrals the error in Simpson's method can be written as

$$E = \frac{f^{(4)}(\xi)}{24} \int_a^b (t-a) \left(t - \frac{a+b}{2} \right)^2 (t-b) dt = -\frac{1}{90} \left(\frac{b-a}{2} \right)^5 f^{(4)}(\xi), \quad (6.24)$$

for $\xi \in (a, b)$. Since the fourth derivative of any polynomial of degree ≤ 3 is identically zero, the quadrature error formula (6.24) implies that Simpson's quadrature is exact for polynomials of degree ≤ 3 .

6.5.2 Composite Simpson rule

To derive a composite version of Simpson's quadrature, we divide the interval $[a, b]$ into an even number of subintervals, n , and let

$$x_i = a + ih, \quad 0 \leq i \leq n,$$

where

$$h = \frac{b-a}{n}.$$

Hence, if we replace the integral in every subintervals by Simpson's rule (6.20), we obtain

$$\begin{aligned} \int_a^b f(x)dx &= \int_{x_0}^{x_2} f(x)dx + \dots + \int_{x_{n-2}}^{x_n} f(x)dx = \sum_{i=1}^{n/2} \int_{x_{2i-2}}^{x_{2i}} f(x)dx \\ &\approx \frac{h}{3} \sum_{i=1}^{n/2} [f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})]. \end{aligned}$$

The **composite Simpson quadrature** is thus given by

$$\int_a^b f(x)dx \approx \frac{h}{3} \left[f(x_0) + 2 \sum_{i=0}^{n/2} f(x_{2i-2}) + 4 \sum_{i=1}^{n/2} f(x_{2i-1}) + f(x_n) \right]. \quad (6.25)$$

Summing the error terms (that are given by (6.24)) over all sub-intervals, the quadrature error takes the form

$$E = -\frac{h^5}{90} \sum_{i=1}^{n/2} f^{(4)}(\xi_i) = -\frac{h^5}{90} \cdot \frac{n}{2} \cdot \frac{2}{n} \sum_{i=1}^{n/2} f^{(4)}(\xi_i).$$

Since

$$\min_{x \in [a,b]} f^{(4)}(x) \leq \frac{2}{n} \sum_{i=1}^{n/2} f^{(4)}(\xi_i) \leq \max_{x \in [a,b]} f^{(4)}(x),$$

we can conclude that

$$E = -\frac{h^5}{90} \frac{n}{2} f^{(4)}(\xi) = -\frac{h^4}{180} f^{(4)}(\xi), \quad \xi \in [a, b], \quad (6.26)$$

i.e., the composite Simpson quadrature is fourth-order accurate.

6.6 Weighted Quadratures

We recall that a weight function is a continuous, non-negative function with a positive mass. We assume that such a weight function $w(x)$ is given and would like to write a quadrature of the form

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n A_i f(x_i). \quad (6.27)$$

Such quadratures are called **general (weighted) quadratures**.

Previously, for the case $w(x) \equiv 1$, we wrote a quadrature of the form

$$\int_a^b f(x)dx \approx \sum_{i=0}^n A_i f(x_i),$$

where

$$A_i = \int_a^b l_i(x)dx.$$

Repeating the derivation we carried out in Section 6.2, we construct an interpolant $Q_n(x)$ of degree $\leq n$ that passes through the points x_0, \dots, x_n . Its Lagrange form is

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

with the usual

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}, \quad 0 \leq i \leq n.$$

Hence

$$\int_a^b f(x)w(x)dx \approx \int_a^b Q_n(x)w(x)dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x)w(x)dx = \sum_{i=0}^n A_i f(x_i),$$

where the coefficients A_i are given by

$$A_i = \int_a^b l_i(x)w(x)dx. \quad (6.28)$$

To summarize, the general quadrature is

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n A_i f(x_i), \quad (6.29)$$

with quadrature coefficients, A_i , that are given by (6.28).

6.7 Gaussian Quadrature

6.7.1 Maximizing the quadrature's accuracy

So far, all the quadratures we encountered were of the form

$$\int_a^b f(x)dx \approx \sum_{i=0}^n A_i f(x_i). \quad (6.30)$$

An approximation of the form (6.30) was shown to be exact for polynomials of degree $\leq n$ for an appropriate choice of the quadrature coefficients A_i . In all cases, the quadrature points x_0, \dots, x_n were given up front. In other words, given a set of nodes x_0, \dots, x_n , the coefficients $\{A_i\}_{i=0}^n$ were determined such that the approximation was exact in Π_n .

We are now interested in investigating the possibility of writing more accurate quadratures without increasing the total number of quadrature points. This will be possible if we allow for the freedom of choosing the quadrature points. The quadrature problem becomes now a problem of choosing the quadrature points in addition to determining the corresponding coefficients in a way that the quadrature is exact for polynomials of a maximal degree. Quadratures that are obtained that way are called **Gaussian quadratures**.

Example 6.6

The quadrature formula

$$\int_{-1}^1 f(x)dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right),$$

is exact for polynomials of degree ≤ 3 (!) We will revisit this problem and prove this result in Example 6.9 below.

An equivalent problem can be stated for the more general weighted quadrature case. Here,

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n A_i f(x_i), \quad (6.31)$$

where $w(x) \geq 0$ is a weight function. Equation (6.31) is exact for $f \in \Pi_n$ if and only if

$$A_i = \int_a^b w(x) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} dx. \quad (6.32)$$

In both cases (6.30) and (6.31), the number of quadrature nodes, x_0, \dots, x_n , is $n+1$, and so is the number of quadrature coefficients, A_i . Hence, if we have the flexibility of determining the location of the points in addition to determining the coefficients,

we have altogether $2n + 2$ degrees of freedom, and hence we can expect to be able to derive quadratures that are exact for polynomials in Π_{2n+1} . This is indeed the case as we shall see below. We will show that the general solution of this integration problem is connected with the roots of orthogonal polynomials. We start with the following theorem.

Theorem 6.7 *Let $q(x)$ be a nonzero polynomial of degree $n + 1$ that is w -orthogonal to Π_n , i.e., $\forall p(x) \in \Pi_n$,*

$$\int_a^b p(x)q(x)w(x)dx = 0.$$

If x_0, \dots, x_n are the zeros of $q(x)$ then (6.31), with A_i given by (6.32), is exact $\forall f \in \Pi_{2n+1}$.

Proof. For $f(x) \in \Pi_{2n+1}$, write $f(x) = q(x)p(x) + r(x)$. We note that $p(x), r(x) \in \Pi_n$. Since x_0, \dots, x_n are the zeros of $q(x)$ then

$$f(x_i) = r(x_i).$$

Hence,

$$\begin{aligned} \int_a^b f(x)w(x)dx &= \int_a^b [q(x)p(x) + r(x)]w(x)dx = \int_a^b r(x)w(x)dx \\ &= \sum_{i=0}^n A_i r(x_i) = \sum_{i=0}^n A_i f(x_i). \end{aligned} \quad (6.33)$$

The second equality in (6.33) holds since $q(x)$ is w -orthogonal to Π_n . The third equality (6.33) holds since (6.31), with A_i given by (6.32), is exact for polynomials in Π_n . ■

According to Theorem 6.7 we already know that the quadrature points that will provide the most accurate quadrature rule are the $n+1$ roots of an orthogonal polynomial of degree $n + 1$ (where the orthogonality is with respect to the weight function $w(x)$). We recall that the roots of $q(x)$ are real, simple and lie in (a, b) , something we know from our previous discussion on orthogonal polynomials (see Theorem 4.15). In other words, we need $n + 1$ quadrature points in the interval, and an orthogonal polynomial of degree $n + 1$ does have $n + 1$ distinct roots in the interval. We now restate the result regarding the roots of orthogonal functions with an alternative proof.

Theorem 6.8 *Let $w(x)$ be a weight function. Assume that $f(x)$ is continuous in $[a, b]$ that is not the zero function, and that $f(x)$ is w -orthogonal to Π_n . Then $f(x)$ changes sign at least $n + 1$ times on (a, b) .*

Proof. Since $1 \in \Pi_n$,

$$\int_a^b f(x)w(x)dx = 0.$$

Hence, $f(x)$ changes sign at least once. Now suppose that $f(x)$ changes sign only r times, where $r \leq n$. Choose $\{t_i\}_{i \geq 0}$ such that

$$a = t_0 < t_1 < \cdots < t_r = b,$$

and $f(x)$ is of one sign on $(t_0, t_1), (t_1, t_2), \dots, (t_{r-1}, t_r)$. The polynomial

$$p(x) = \prod_{i=0}^{r-1} (x - t_i),$$

has the same sign property. Hence

$$\int_a^b f(x)p(x)w(x)dx \neq 0,$$

which leads to a contradiction since $p(x) \in \Pi_n$. ■

Example 6.9

We are looking for a quadrature of the form

$$\int_{-1}^1 f(x)dx \approx A_0 f(x_0) + A_1 f(x_1).$$

A straightforward computation will amount to making this quadrature exact for the polynomials of degree ≤ 3 . The linearity of the quadrature means that it is sufficient to make the quadrature exact for $1, x, x^2$, and x^3 . Hence we write the system of equations

$$\int_{-1}^1 f(x)dx = \int_{-1}^1 x^i dx = A_0 x_0^i + A_1 x_1^i, \quad i = 0, 1, 2, 3.$$

From this we can write

$$\begin{cases} A_0 + A_1 = 2, \\ A_0 x_0 + A_1 x_1 = 0, \\ A_0 x_0^2 + A_1 x_1^2 = \frac{2}{3}, \\ A_0 x_0^3 + A_1 x_1^3 = 0. \end{cases}$$

Solving for A_1, A_2, x_0 , and x_1 we get

$$A_1 = A_2 = 1, \quad x_0 = -x_1 = \frac{1}{\sqrt{3}},$$

so that the desired quadrature is

$$\int_{-1}^1 f(x)dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right). \quad (6.34)$$

Example 6.10

We repeat the previous problem using orthogonal polynomials. Since $n = 1$, we expect to find a quadrature that is exact for polynomials of degree $2n + 1 = 3$. The polynomial of degree $n + 1 = 2$ which is orthogonal to $\Pi_n = \Pi_1$ with weight $w(x) \equiv 1$ is the Legendre polynomial of degree 2, i.e.,

$$P_2(x) = \frac{1}{2}(3x^2 - 1).$$

The integration points will then be the zeros of $P_2(x)$, i.e.,

$$x_0 = -\frac{1}{\sqrt{3}}, \quad x_1 = \frac{1}{\sqrt{3}}.$$

All that remains is to determine the coefficients A_1, A_2 . This is done in the usual way, assuming that the quadrature

$$\int_{-1}^1 f(x)dx \approx A_0f(x_0) + A_1f(x_1),$$

is exact for polynomials of degree ≤ 1 . The simplest will be to use 1 and x , i.e.,

$$2 = \int_{-1}^1 1dx = A_0 + A_1,$$

and

$$0 = \int_{-1}^1 xdx = -A_0\frac{1}{\sqrt{3}} + A_1\frac{1}{\sqrt{3}}.$$

Hence $A_0 = A_1 = 1$, and the quadrature is the same as (6.34) (as should be).

6.7.2 Convergence and error analysis

Lemma 6.11 *In a Gaussian quadrature formula, the coefficients are positive and their sum is $\int_a^b w(x)dx$.*

Proof. Fix n . Let $q(x) \in \Pi_{n+1}$ be w -orthogonal to Π_n . Also assume that $q(x_i) = 0$ for $i = 0, \dots, n$, and take $\{x_i\}_{i=0}^n$ to be the quadrature points, i.e.,

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n A_i f(x_i). \quad (6.35)$$

Fix $0 \leq j \leq n$. Let $p(x) \in \Pi_n$ be defined as

$$p(x) = \frac{q(x)}{x - x_j}.$$

Since x_j is a root of $q(x)$, $p(x)$ is indeed a polynomial of degree $\leq n$. The degree of $p^2(x) \leq 2n$ which means that the Gaussian quadrature (6.35) is exact for it. Hence

$$0 < \int_a^b p^2(x)w(x)dx = \sum_{i=0}^n A_i p^2(x_i) = \sum_{i=0}^n A_i \frac{q^2(x_i)}{(x_i - x_j)^2} = A_j p^2(x_j),$$

which means that $\forall j, A_j > 0$. In addition, since the Gaussian quadrature is exact for $f(x) \equiv 1$, we have

$$\int_a^b w(x)dx = \sum_{i=0}^n A_i. \quad \blacksquare$$

In order to estimate the error in the Gaussian quadrature we would first like to present an alternative way of deriving the Gaussian quadrature. Our starting point is the Lagrange form of the Hermite polynomial that interpolates $f(x)$ and $f'(x)$ at x_0, \dots, x_n . It is given by (3.44), i.e.,

$$p(x) = \sum_{i=0}^n f(x_i)a_i(x) + \sum_{i=0}^n f'(x_i)b_i(x),$$

with

$$a_i(x) = (l_i(x))^2[1 + 2l_i'(x_i)(x - x_i)], \quad b_i(x) = (x - x_i)l_i^2(x), \quad 0 \leq i \leq n,$$

and

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}.$$

We now assume that $w(x)$ is a weight function in $[a, b]$ and approximate

$$\int_a^b w(x)f(x)dx \approx \int_a^b w(x)p_{2n+1}(x)dx = \sum_{i=0}^n A_i f(x_i) + \sum_{i=0}^n B_i f'(x_i), \quad (6.36)$$

where

$$A_i = \int_a^b w(x)a_i(x)dx, \quad (6.37)$$

and

$$B_i = \int_a^b w(x)b_i(x)dx. \quad (6.38)$$

In some sense, it seems to be rather strange to deal with the Hermite interpolant when we do not explicitly know the values of $f'(x)$ at the interpolation points. However, we can eliminate the derivatives from the quadrature (6.36) by setting $B_i = 0$ in (6.38). Indeed (assuming $n \neq 0$):

$$B_i = \int_a^b w(x)(x - x_i)l_i^2(x)dx = \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j) \int_a^b w(x) \prod_{j=0}^n (x - x_j)l_i(x)dx.$$

Hence, $B_i = 0$, if the product $\prod_{j=0}^n (x - x_j)$ is orthogonal to $l_i(x)$. Since $l_i(x)$ is a polynomial in Π_n , all that we need is to set the points x_0, \dots, x_n as the roots of a polynomial of degree $n+1$ that is w -orthogonal to Π_n . This is precisely what we defined as a Gaussian quadrature.

We are now ready to formally establish the fact that the Gaussian quadrature is exact for polynomials of degree $\leq 2n+1$.

Theorem 6.12 *Let $f \in C^{2n+2}[a, b]$ and let $w(x)$ be a weight function. Consider the Gaussian quadrature*

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n A_i f(x_i).$$

Then there exists $\zeta \in (a, b)$ such that

$$\int_a^b f(x)w(x)dx - \sum_{i=0}^n A_i f(x_i) = \frac{f^{(2n+2)}(\zeta)}{(2n+2)!} \int_a^b \prod_{j=0}^n (x - x_j)^2 w(x)dx.$$

Proof. We use the characterization of the Gaussian quadrature as the integral of a Hermite interpolant. We recall that the error formula for the Hermite interpolation is given by (3.51),

$$f(x) - p_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{j=0}^n (x - x_j)^2, \quad \xi \in (a, b).$$

Hence according to (6.36) we have

$$\begin{aligned} \int_a^b f(x)w(x)dx - \sum_{i=0}^n A_i f(x_i) &= \int_a^b f(x)w(x)dx - \int_a^b p_{2n+1}w(x)dx \\ &= \int_a^b w(x) \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \prod_{j=0}^n (x - x_j)^2 dx. \end{aligned}$$

The integral mean value theorem then implies that there exists $\zeta \in (a, b)$ such that

$$\int_a^b f(x)w(x)dx - \sum_{i=0}^n A_i f(x_i) = \frac{f^{(2n+2)}(\zeta)}{(2n+2)!} \int_a^b \prod_{j=0}^n (x - x_j)^2(x)w(x)dx. \quad \blacksquare$$

We conclude this section with a convergence theorem that states that for continuous functions, the Gaussian quadrature converges to the exact value of the integral as the number of quadrature points tends to infinity. This theorem is not of a great practical value because it does not provide an estimate on the rate of convergence. A proof of the theorem that is based on the Weierstrass approximation theorem can be found in, e.g., in [7].

Theorem 6.13 *We let $w(x)$ be a weight function and assuming that $f(x)$ is a continuous function on $[a, b]$. For each $n \in \mathbb{N}$ we let $\{x_{n_i}\}_{i=0}^n$ be the $n + 1$ roots of the polynomial of degree $n + 1$ that is w -orthogonal to Π_n , and consider the corresponding Gaussian quadrature:*

$$\int_a^b f(x)w(x)dx \approx \sum_{i=0}^n A_{n_i}f(x_{n_i}). \quad (6.39)$$

Then the right-hand-side of (6.39) converges to the left-hand-side as $n \rightarrow \infty$.

6.8 Romberg Integration

We have introduced Richardson's extrapolation in Section 5.4 in the context of numerical differentiation. We can use a similar principle with numerical integration.

We will demonstrate this principle with a particular example. Let I denote the exact integral that we would like to approximate, i.e.,

$$I = \int_a^b f(x)dx.$$

Let's assume that this integral is approximated with a composite trapezoidal rule on a uniform grid with mesh spacing h (6.13),

$$T(h) = h \sum_{i=0}^n f(a + ih).$$

We know that the composite trapezoidal rule is second-order accurate (see (6.14)). A more detailed study of the quadrature error reveals that the difference between I and $T(h)$ can be written as

$$I = T(h) + c_1h^2 + c_2h^4 + \dots + c_kh^k + O(h^{2k+2}).$$

The exact values of the coefficients, c_k , are of no interest to us as long as they do not depend on h (which is indeed the case). We can now write a similar quadrature that is based on half the number of points, i.e., $T(2h)$. Hence

$$I = T(2h) + c_1(2h)^2 + c_2(2h)^4 + \dots$$

This enables us to eliminate the h^2 error term:

$$I = \frac{4T(h) - T(2h)}{3} + \hat{c}_2 h^4 + \dots$$

Therefore

$$\begin{aligned} \frac{4T(h) - T(2h)}{3} &= \frac{1}{3} \left[4h \left(\frac{1}{2}f_0 + f_1 + \dots + f_{n-1} + \frac{1}{2}f_n \right) \right. \\ &\quad \left. - 2h \left(\frac{1}{2}f_0 + f_2 + \dots + f_{n-2} + \frac{1}{2}f_n \right) \right] \\ &= \frac{h}{3} (f_0 + 4f_1 + 2f_2 + \dots + 2f_{n-2} + 4f_{n-1} + f_n) = S(n). \end{aligned}$$

Here, $S(n)$ denotes the composite Simpson's rule with n subintervals. The procedure of increasing the accuracy of the quadrature by eliminating the leading error term is known as **Romberg integration**. In some places, Romberg integration is used to describe the specific case of turning the composite trapezoidal rule into Simpson's rule (and so on). The quadrature that is obtained from Simpson's rule by eliminating the leading error term is known as the **super Simpson rule**.

References

- [1] Atkinson K., *An introduction to numerical analysis*, Second edition, John Wiley & Sons, New York, NY, 1989
- [2] Cheney E.W., *Introduction to approximation theory*, Second edition, Chelsea publishing company, New York, NY, 1982
- [3] Dahlquist G., Björck A., *Numerical methods*, Prentice-Hall, Englewood cliffs, NJ, 1974
- [4] Davis P.J., *Interpolation and approximation*, Second edition, Dover, New York, NY, 1975
- [5] Isaacson E., Keller H.B., *Analysis of numerical methods*, Second edition, Dover, Mineola, NY, 1994
- [6] Stoer J., Burlisch R., *Introduction to numerical analysis*, Second edition, Springer-Verlag, New York, NY, 1993
- [7] Süli E., Mayers D., *An introduction to numerical analysis*, Cambridge university press, Cambridge, UK, 2003.

Index

L^2 -norm	58, 70	Gram-Schmidt	76
weighted	75, 76	Hermite polynomials	79
L^∞ -norm	57	Hilbert matrix	72
approximation		inner product	76
best approximation	63	weighted	76
existence	63	integration	
least-squares	70	Gaussian	115
Hilbert matrix	72	orthogonal polynomials	116
orthogonal polynomials	73	midpoint rule	100
solution	70	composite	108
minimax	62	Newton-Cotes	103
oscillating theorem	66	quadratures	100
remez	68	weighted	114
uniqueness	66	rectangular rule	100
near minimax	68	Riemann	99
Weierstrass	58	Romberg	121, 122
Bernstein polynomials	59	Simpson's rule	110
Bessel's inequality	81	composite	113
Chebyshev		error	110, 113
near minimax interpolation	68	super Simpson	122
points	38, 68	trapezoidal rule	104
polynomials	34, 78	composite	106
Chebyshev uniqueness theorem	66	undetermined coefficients	108
de la Vallée-Poussin	65	interpolation	
differentiation	88	Chebyshev points	34, 38
accuracy	89	divided differences	28, 32
backward differencing	89	with repetitions	43
centered differencing	89	error	31, 34
forward differencing	89	existence	21
one-sided differencing	89	Hermite	41
Richardson's extrapolation	96	Lagrange form	44
truncation error	89	Newton's form	43
undetermined coefficients	94	interpolation error	20
via interpolation	90, 92	interpolation points	20
divided differences	28, 32	Lagrange form	26
with repetitions	43	near minimax	68
		Newton's form	22, 28
		divided differences	28

-
- polynomial interpolation 20
 - splines 48
 - cubic 50
 - degree 48
 - knots 48
 - natural 53, 54
 - not-a-knot 54
 - uniqueness 21, 23
 - Vandermonde determinant 23
 - weighted least squares 74
 - Lagrange form *see* interpolation
 - Laguerre polynomials 79, 84
 - least-squares *see* approximation
 - weighted *see* approximation
 - Legendre polynomials 77, 82, 83
 - maximum norm 57
 - minimax error 62, 65
 - monic polynomial 36
 - Newton's form *see* interpolation
 - norm 57
 - orthogonal polynomials 73, 74, 76, 77, 116
 - Bessel's inequality 81
 - Chebyshev 78
 - Gram-Schmidt 76
 - Hermite 79
 - Laguerre 79, 84
 - Legendre 77, 82, 83
 - Parseval's equality 81
 - roots of 85
 - triple recursion relation 86
 - oscillating theorem 66
 - Parseval's equality 81
 - quadratures *see* integration
 - Remez algorithm 68
 - Richardson's extrapolation 96, 121
 - Riemann sums 99
 - Romberg integration 121, 122
 - root finding
 - are there any roots? 3
 - examples of methods 5
 - iterative methods 6
 - Newton's method 11
 - preliminary discussion 2
 - the bisection method 8
 - the secant method 15
 - splines *see* interpolation
 - Taylor series 42
 - triangle inequality 57
 - Vandermonde determinant 23
 - Weierstrass approximation theorem 58
-