

# Chapter 6

## Numerical integration

### Introduction

Integrals are ubiquitous in science and mathematics. In this chapter, we are concerned with the problem of calculating numerically integrals of the form

$$I = \int_{\Omega} u(\mathbf{x}) \, d\mathbf{x}, \quad (6.1)$$

Perhaps somewhat surprisingly, the numerical calculation of such integrals when  $n \gg 1$  is still a very active area of research today. In this chapter, we will focus for simplicity on the one-dimensional setting where  $\Omega = [a, b] \subset \mathbf{R}$ . We assume throughout this chapter that the function  $u$  is Riemann-integrable. This means that

$$I = \lim_{h \rightarrow 0} \sum_{i=0}^{n-1} u(t_i)(z_{i+1} - z_i),$$

where  $a = z_0 < \dots < z_n = b$  is a partition of the interval  $[a, b]$  such that the maximum spacing between successive  $x$  values is equal to  $h$ , and with  $t_i \in [x_i, x_{i+1}]$  for all  $i \in \{0, \dots, n-1\}$ .

All the numerical integration formulas that we present in this chapter are based on a deterministic approximation of the form

$$\hat{I} = \sum_{i=0}^m w_i u(x_i), \quad (6.2)$$

where  $x_0 < \dots < x_n$  are the *integration nodes* and  $w_0, \dots, w_n$  are the *integration weights*. In many cases, integration formulas contain a small parameter that can be changed to improve the accuracy of the approximation. In methods based on equidistant interpolation nodes, for example, this parameter encodes the distance between nodes and is typically denoted by  $h$ , and we often use the notation  $\hat{I}_h$  to emphasize the dependence of the approximation on  $h$ . The difference  $E_h = I - I_h$  is called the *integration error* or *discretization error*, and the *degree of precision* of an integration method is the smallest integer number  $d$  such that the integration error is zero for all the polynomials of degree less than or equal to  $d$ .

We observe that, without loss of generality, we can consider that the integration interval is equal to  $[-1, 1]$ . Indeed, using the change of variable

$$\begin{aligned}\zeta: [-1, 1] &\rightarrow [a, b]; \\ y &\mapsto \frac{b+a}{2} + \frac{(b-a)}{2}y,\end{aligned}\tag{6.3}$$

we have

$$\int_a^b u(x) dx = \int_{-1}^1 u(\zeta(y)) \zeta'(y) dy = \frac{b-a}{2} \int_{-1}^1 u \circ \zeta(y) dy,\tag{6.4}$$

and the right-hand side is the integral of  $u \circ \zeta$  over the interval  $[-1, 1]$ .

## 6.1 The Newton–Cotes method

Given a set of equidistant points  $-1 = x_0 < \dots < x_m = 1$ , a natural method for approximating the integral (6.1) of a function  $u: [-1, 1] \rightarrow \mathbf{R}$  is to first construct the interpolating polynomial  $\hat{u}$  through the nodes, and then calculate the integral of this polynomial. By construction, this method is exact for polynomials of degree up to  $m$ , and so the degree of precision is equal to *at least*  $m$ . Let  $\varphi_0, \dots, \varphi_m$  denote the Lagrange polynomials associated with the integration nodes. Then we have

$$I \approx \int_{-1}^1 \hat{u}(x) dx = \int_{-1}^1 \sum_{i=0}^n u(x_i) \varphi_i(x) dx = \sum_{i=0}^n u(x_i) \underbrace{\int_{-1}^1 \varphi_i(x) dx}_{w_i}.$$

The weights are independent of the function  $u$ , and so they can be calculated a priori. The class of integration methods obtained using this approach are known as *Newton–Cotes methods*. We present a few particular cases:

- $m = 1, d = 1$  (trapezoidal rule):

$$\int_{-1}^1 u(x) dx \approx u(-1) + u(1).\tag{6.5}$$

- $m = 2, d = 3$  (Simpson's rule):

$$\int_{-1}^1 u(x) dx \approx \frac{1}{3}u(-1) + \frac{4}{3}u(0) + \frac{1}{3}u(1).\tag{6.6}$$

- $m = 3, d = 3$  (Simpson's  $\frac{3}{8}$  rule):

$$\int_{-1}^1 u(x) dx \approx \frac{1}{4}u(-1) + \frac{3}{4}u(-1/3) + \frac{3}{4}u(1/3) + \frac{1}{4}u(1).$$

- $m = 4, d = 5$  (Bode's rule):

$$\int_{-1}^1 u(x) dx \approx \frac{7}{45}u(-1) + \frac{32}{45}u\left(-\frac{1}{2}\right) + \frac{12}{45}u(0) + \frac{32}{45}u\left(\frac{1}{2}\right) + \frac{7}{45}u(1).$$

In principle, this approach could be employed in order to construct integration rules of arbitrary high degree of precision. In practice, however, the weights become more and more imbalanced as the number of interpolation points increases, with some of them becoming negative. As a result, roundoff errors become increasingly detrimental to accuracy as the degree of precision increases. In addition, in cases where the interpolating polynomial does not converge to  $u$ , for example if  $u$  is Runge's function, the approximate integral may not even converge to the correct value in the limit as  $m \rightarrow \infty$  in exact arithmetic!

Note that, although it is based on a quadratic polynomial interpolation, Simpson's rule (6.6) has a degree of precision equal to 3. This is because any integration rule with nodes and weights symmetric around  $x = 0$  is exact for odd functions, in particular  $x^3$ . Likewise, the degree of precision of Bode's rule is equal to 5.

## 6.2 Composite methods with equidistant nodes

A natural alternative to the approach presented in Section 6.1 is to construct an integration rule using piecewise polynomial interpolation, which we studied in Section 5.1.7. After partitioning the integration interval in a number of subintervals, the integral can be approximated by using one of the rules presented in (6.1) within each subinterval.

**Composite trapezoidal rule.** Let us illustrate the composite approach with an example. To this end, we introduce a partition  $a = x_0 < \dots < x_m = b$  of the interval  $[a, b]$  and assume that the nodes are equidistant with  $x_{i+1} - x_i = h$ . Using (6.4), we first generalize (6.5) to an interval  $[x_i, x_{i+1}]$  as follows:

$$\int_{x_i}^{x_{i+1}} u(x) dx = \int_{-1}^1 u \circ \zeta(y) dy \approx u \circ \zeta(-1) + u \circ \zeta(1) = \frac{h}{2}(u(x_i) + u(x_{i+1})),$$

where  $\approx$  in this equation indicates approximation using the trapezoidal rule. Applying this approximation to each subinterval of the partition, we obtain the composite trapezoidal rule:

$$\begin{aligned} \int_a^b u(x) dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} u(x) dx \approx \frac{h}{2} \sum_{i=0}^{n-1} (u(x_i) + u(x_{i+1})) \\ &= \frac{h}{2} (u(x_0) + 2u(x_1) + 2u(x_2) + \dots + 2u(x_{n-2}) + 2u(x_{n-1}) + u(x_n)). \end{aligned} \quad (6.7)$$

Like the trapezoidal rule (6.5), the composite trapezoidal rule (6.7) has a degree of precision equal to 1. However, the accuracy of the method depends on the parameter  $h$ , which represents the width of each subinterval: for very small  $h$ , equation (6.7) is expected to provide a good approximation of the integral. An error estimate can be obtained directly from the formula in Theorem 5.2 for interpolation error, provided that we assume that  $u \in C^2([a, b])$ . Denoting by  $\widehat{I}_h$  the approximate integral calculated using (6.7), and by  $\widehat{u}_h$  the piecewise linear interpolation of  $u$ , we have

$$\int_{x_i}^{x_{i+1}} u(x) - \widehat{u}_h(x) dx = \frac{1}{2} \int_{x_i}^{x_{i+1}} u''(\xi(x))(x - x_i)(x - x_{i+1}) dx.$$

Since  $(x - x_i)(x - x_{i+1})$  is nonpositive over the interval  $[x_i, x_{i+1}]$ , we deduce that

$$\left| \int_{x_i}^{x_{i+1}} u(x) - \widehat{u}(x) \, dx \right| \leq \left( \sup_{\xi \in [a, b]} |u''(\xi)| \right) \int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i+1}) \, dx = C_2 \frac{h^3}{12},$$

where we introduced

$$C_2 = \sup_{\xi \in [a, b]} |u''(\xi)|.$$

Summing the contributions of all the intervals, we obtain

$$|I - \widehat{I}| \leq \sum_{i=0}^{n-1} \left| \int_{x_i}^{x_{i+1}} u(x) - \widehat{u}(x) \, dx \right| \leq n \times C_2 \frac{h^3}{12} = \frac{b-a}{12} C_2 h^2. \quad (6.8)$$

The integration error therefore scales as  $\mathcal{O}(h^2)$ . (Strictly speaking, we have shown only that the integration error admits an upper bound that scales at  $\mathcal{O}(h^2)$ , but it turns out that the dependence on  $h$  of this bound is optimal).

**Composite Simpson rule.** The composite Simpson rule is derived in [Exercise 6.2](#). Given an odd number  $n + 1$  of equidistant points  $a = x_0 < x_1 < \dots < x_n = b$ , it is given by

$$\widehat{I}_h = \frac{h}{3} \left( u(x_0) + 4u(x_1) + 2u(x_2) + 4u(x_3) + 2u(x_4) + \dots + 2u(x_{n-2}) + 4u(x_{n-1}) + u(x_n) \right). \quad (6.9)$$

This approximation is obtained by integrating the piecewise quadratic interpolant over a partition of the integration interval into  $n/2$  subintervals of equal width. Obtaining an optimal error estimate, in terms of the dependence on  $h$ , for this integration formula is slightly more involved. For a given subinterval  $[x_{2i}, x_{2i+2}]$ , let us denote by  $\widehat{u}_2(x)$  the quadratic interpolating polynomial at  $x_{2i}, x_{2i+1}, x_{2i+2}$ , and by  $\widehat{u}_3(x)$  a cubic interpolating polynomial relative to the nodes  $x_{2i}, x_{2i+1}, x_{2i+2}, x_\alpha$ , for some  $\alpha \in [x_{2i}, x_{2i+1}]$ . We have

$$\int_{x_{2i}}^{x_{2i+2}} u(x) - \widehat{u}_2(x) \, dx = \int_{x_{2i}}^{x_{2i+2}} u(x) - \widehat{u}_3(x) \, dx + \int_{x_{2i}}^{x_{2i+2}} \widehat{u}_3(x) - \widehat{u}_2(x) \, dx. \quad (6.10)$$

The second term is zero, because the integrand is a cubic polynomial with zeros at  $x_{2i}, x_{2i+1}$  and  $x_{2i+2}$ , and because

$$\int_{x_{2i}}^{x_{2i+2}} (x - x_{2i})(x - x_{2i+1})(x - x_{2i+2}) \, dx = 0.$$

(Notice that the integrand is even around  $x_{2i+1}$ .) The cancellation of the second term in (6.10) also follows from the fact that the degree of precision of the Simpson rule (6.6) is equal to 3, and so

$$\int_{x_{2i}}^{x_{2i+2}} \widehat{u}_3(x) - \widehat{u}_2(x) \, dx = \frac{1}{3}(\widehat{u}_3 - \widehat{u}_2)(x_{2i}) + \frac{4}{3}(\widehat{u}_3 - \widehat{u}_2)(x_{2i+1}) + \frac{1}{3}(\widehat{u}_3 - \widehat{u}_2)(x_{2i+2}) = 0.$$

Using [Theorem 5.2](#), we rewrite first term in [\(6.10\)](#) from above as follows:

$$\left| \int_{x_{2i}}^{x_{2i+2}} u(x) - \widehat{u}_3(x) \, dx \right| \leq \int_{x_{2i}}^{x_{2i+2}} \frac{u^{(4)}(\xi(x))}{24} (x - x_{2i})(x - x_{2i+1})(x - x_{2i+2})(x - x_{\alpha}) \, dx.$$

Since this formula is valid for all  $\alpha \in [2i, 2i + 2]$ , we are allowed to take  $\alpha = x_{2i+1}$ . Given that

$$\int_{x_{2i}}^{x_{2i+2}} (x - x_{2i})(x - x_{2i+1})^2(x - x_{2i+2}) \, dx = \frac{4}{15}h^5,$$

with an integrand everywhere nonpositive in the interval  $[x_{2i}, x_{2i+2}]$ , we conclude that

$$\left| \int_{x_{2i}}^{x_{2i+2}} u(x) - \widehat{u}_3(x) \, dx \right| \leq \frac{C_4}{90}h^5, \quad C_4 = \sup_{\xi \in [a,b]} |u^{(4)}(\xi)|.$$

Summing the contributions of all the subintervals, we finally obtain

$$|I - \widehat{I}_h| \leq \frac{n}{2} \times \frac{C_4 h^5}{90} = (b - a) \frac{C_4 h^4}{180}. \quad (6.11)$$

**Estimating the error.** In practice, it is useful to be able to estimate the integration error so that, if the error is deemed too large, a better approximation of the integral can then be calculated by using a smaller value for the step size  $h$ . Calculating the exact error  $I - \widehat{I}_h$  is impossible in general, because this would require to know the exact value of the integral, but it is possible to calculate a rough approximation of the error based on two numerical approximations of the integral, as we illustrate formally hereafter for the composite Simpson rule.

Suppose that  $\widehat{I}_{2h}$  and  $\widehat{I}_h$  are two approximations of the integral, calculated using the composite Simpson rule with step size  $2h$  and  $h$ , respectively. If we assume that the error scales as  $\mathcal{O}(h^4)$  as [\(6.11\)](#) suggests, then it holds approximately that

$$I - \widehat{I}_h \approx \frac{1}{24}(I - \widehat{I}_{2h}). \quad (6.12)$$

This implies that

$$I - \widehat{I}_{2h} = (I - \widehat{I}_h) + (I_h - \widehat{I}_{2h}) \approx \frac{1}{16}(I - \widehat{I}_{2h}) + (I_h - \widehat{I}_{2h}).$$

Rearranging this equation gives an approximation of the error for  $\widehat{I}_{2h}$ :

$$I - \widehat{I}_{2h} \approx \frac{16}{15}(\widehat{I}_h - \widehat{I}_{2h}).$$

Using [\(6.12\)](#), we can then derive an error estimate for  $\widehat{I}_h$ :

$$|I - \widehat{I}_h| \approx \frac{1}{15}|\widehat{I}_h - \widehat{I}_{2h}|. \quad (6.13)$$

The right-hand side can be calculated numerically, because it does not depend on the exact value of the integral. In practice, the two sides of [\(6.13\)](#) are often very close for small  $h$ . In the

code example below, we approximate the integral

$$I = \int_0^{\frac{\pi}{2}} \cos(x) dx = 1 \quad (6.14)$$

for different step sizes and compare the exact error with the approximate error obtained using (6.13). The results obtained are summarized in Table 6.1, which shows a good match between the two quantities.

Table 6.1: Comparison between the exact integration error and the approximate integration error calculated using (6.13).

$h$	Exact error $ I - \widehat{I}_h $	Approximate error $\frac{1}{15} \widehat{I}_h - \widehat{I}_{2h} $
$2^{-4}$	$5.166847063531321 \times 10^{-7}$	$5.185892840930961 \times 10^{-7}$
$2^{-5}$	$3.226500089326123 \times 10^{-8}$	$3.229464703065806 \times 10^{-8}$
$2^{-6}$	$2.0161285974040766 \times 10^{-9}$	$2.016591486390477 \times 10^{-9}$
$2^{-7}$	$1.2600120946615334 \times 10^{-10}$	$1.260084925291949 \times 10^{-10}$

```
# Composite Simpson's rule
function composite_simpson(u, a, b, n)
    # Integration nodes
    x = LinRange(a, b, n + 1)
    # Evaluation of u at the nodes
    ux = u.(x)
    # Step size
    h = x[2] - x[1]
    # Approximation of the integral
    return (h/3) * sum([ux[1]; ux[end]; 4ux[2:2:end-1]; 2ux[3:2:end-2]])
end
# Function to integrate
u(x) = cos(x)
# Integration bounds
a, b = 0, pi/2
# Exact integral
I = 1.0
# Number of subintervals
ns = [8; 16; 32; 64; 128]
# Approximate integrals
Î = composite_simpson.(u, a, b, ns)
# Calculate exact and approximate errors
for i in 2:length(ns)
    println("Exact error: $(I - Î[i]), ",
           "Approx error: $((Î[i] - Î[i-1])/15)")
end
```

### 6.3 Richardson extrapolation and Romberg's method

In the previous section, we showed how the integration error could be approximated based on two approximations of the integral with different step sizes. The aim of this section is to show that, by cleverly combining two approximations  $\widehat{I}_h$  and  $\widehat{I}_{2h}$  of an integral, an approximation even better than  $\widehat{I}_h$  can be constructed.

This approach is based on *Richardson's extrapolation*, which is a general method for accelerating the convergence of sequences, with applications beyond numerical integration. The idea is the following: assume that  $J(h)$  is an approximation with step size  $h$  of some unknown quantity  $J_* = \lim_{h \rightarrow 0} J(h)$ , and that we have access to evaluations of  $J$  at  $h, h/2, h/4, h/8, \dots$ . Assuming that  $J$  extends to a smooth function over  $[0, H]$ , we have by Taylor expansion that

$$J(\eta) = J(0) + J'(0)\eta + J''(0)\frac{\eta^2}{2} + J^{(3)}(0)\frac{\eta^3}{3!} + \dots + J^{(k)}(0)\frac{\eta^k}{k!} + \mathcal{O}(\eta^{k+1}).$$

**Elimination of the linear error term.** Let us assume that  $J'(0) \neq 0$ , so that the leading order term after the constant  $J(0)$  scales as  $\eta$ . Then we have

$$\begin{aligned} J(h) &= J(0) + J'(0)h + \mathcal{O}(h^2) \\ J(h/2) &= J(0) + J'(0)\frac{h}{2} + \mathcal{O}(h^2). \end{aligned}$$

We now ask the following question: can we combine linearly  $J(h)$  and  $J(h/2)$  in order to approximate  $J(0)$  with an error scaling as  $\mathcal{O}(h^2)$ ? Using the ansatz  $J_1(h/2) = \alpha J(h) + \beta J(h/2)$ , we calculate

$$J_1(h/2) = (\alpha + \beta)J(0) + J'(0)h \left( \alpha + \frac{1 - \alpha}{2} \right) + \mathcal{O}(h^2). \quad (6.15)$$

Since we want this expression to approximate  $J(0)$  for small  $h$ , we need to impose that  $\alpha + \beta = 1$ . Then, in order for the term multiplying  $h$  to cancel out, we require that

$$\alpha + \frac{1 - \alpha}{2} = 0 \quad \Leftrightarrow \quad \alpha = -1.$$

This yields the formula

$$J_1(h/2) = 2J(h/2) - J(h). \quad (6.16)$$

Notice that, in the case where  $J$  is a linear function,  $J_1(h/2)$  is exactly equal to  $J(0)$ . This reveals a geometric interpretation of (6.16): the approximation  $J_1(h/2)$  is simply the  $y$  intercept of the straight line passing through the points  $(h/2, J(h/2))$  and  $(h, J(h))$ .

**Elimination of the quadratic error term.** If we had tracked the coefficient of  $h^2$  in the previous paragraph, we would have obtained instead of (6.15) the following equation:

$$J_1(h/2) = J(0) - J^{(3)}(0)\frac{h^2}{4} + \mathcal{O}(h^3).$$

Provided that we have access also to  $J(h/4)$ , we can also calculate

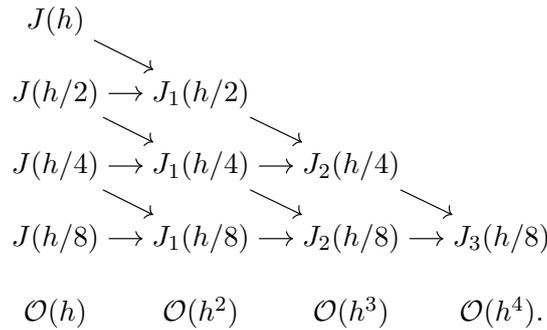
$$J_1(h/4) = \frac{2J(h/4) - J(h/2)}{2} = J(0) - J^{(3)}(0)\frac{h^2}{16} + \mathcal{O}(h^3).$$

At this point, it is natural to wonder whether we can combine  $J_1(h/2)$  and  $J_1(h/4)$  in order to produce an even better approximation of  $J(0)$ . Applying the same reasoning as in the previous section leads us to introduce

$$J_2(h/4) = \frac{4J_1(h/2) - J_2(h/4)}{4 - 1} = J(0) + \mathcal{O}(h^3).$$

This is an exact approximation of  $J(0)$  if  $J$  is a quadratic polynomial, implying that  $J_2(h/4)$  is simply the  $y$  intercept of the quadratic polynomial interpolation through the points  $(h/4, J(h/4))$ ,  $(h/2, J(h/2))$  and  $(h, J(h))$ .

**Elimination of higher order terms.** The procedure above can be repeated in order to eliminate terms of higher and higher orders. The following schematic illustrates, for example, the calculation of an approximation  $J_3(h/8) = J(0) + \mathcal{O}(h^4)$ .



The linear combination in order to calculate  $J_i(h/2^i)$  is always of the form

$$J_i(h/2^i) = \frac{2^i J_{i-1}(h/2^i) - J_{i-1}(h/2^{i-1})}{2^i - 1}, \quad J_0 = J.$$

In practice we calculate the values taken by  $J, J_1, J_2, \dots$  at specific values of  $h$ , but these are in fact functions of  $h$ . In [Figure 6.1](#), we plot these functions when  $J(h) = 1 + \sin(h)$ . It appears clearly from the figure that, for sufficiently small  $h$ ,  $J_3(h)$  provides the most precise approximation of  $J(0) = 1$ . Constructing the functions in Julia can be achieved in just a few lines of code.

```

J(h) = 1 + sin(h)
J_1(h) = 2J(h) - J(2h)
J_2(h) = (4J_1(h) - J_1(2h))/3
J_3(h) = (8J_2(h) - J_2(2h))/7

```

**Generalization.** Sometimes, it is known a priori that the Taylor development of the function  $J$  around zero contains only even powers of  $h$ . In this case, the Richardson extrapolation procedure can be slightly modified to produce approximations with errors scaling as  $\mathcal{O}(h^4)$ , then  $\mathcal{O}(h^6)$ ,

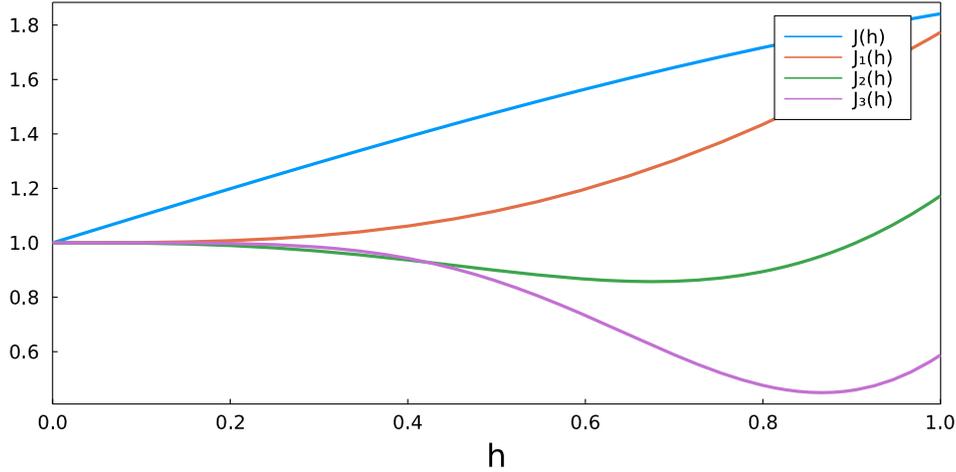
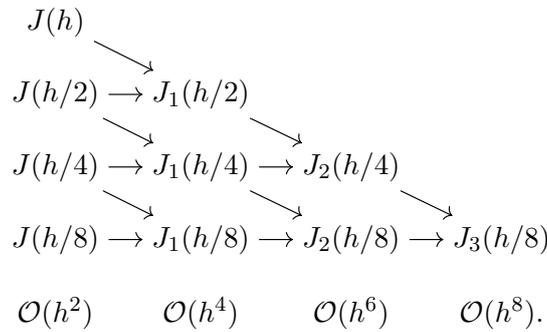


Figure 6.1: Illustration of the functions  $J_1$ ,  $J_2$  and  $J_3$  constructed by Richardson interpolation.

then  $\mathcal{O}(h^8)$ , etc. This procedure is illustrated below:



This time, the linear combinations required for populating this table are given by:

$$J_i(h/2^i) = \frac{2^{2i} J_{i-1}(h/2^i) - J_{i-1}(h/2^{i-1})}{2^{2i} - 1}. \tag{6.17}$$

**Application to integration: Romberg’s method** Romberg’s integration method consists of applying Richardson’s extrapolation to the function

$$J(h) = \widehat{I}_h = u(x_0) + 2u(x_1) + 2u(x_2) + \cdots + 2u(x_{n-1}) + 2u(x_n), \quad h \in \left\{ \frac{b-a}{n}; n \in \mathbf{N} \right\}.$$

where  $a = x_0 < x_1 < \cdots < x_n = b$  are equidistant nodes. The right-hand side of this equation is simply the composite trapezoidal rule with step size  $h$ . It is possible to show, see [9], that  $J(h)$  may be expanded as follows:

$$\forall k \in \mathbf{N}, \quad J(h) = I + \alpha_1 h^2 + \alpha_2 h^4 + \cdots + \alpha_k h^{2k} + \mathcal{O}(h^{2k+2}).$$

Richardson’s extrapolation (6.17) can therefore be employed in order to compute approximations of the integral increasing accuracy. The convergence of Romberg’s method for calculating the integral (6.14) is illustrated in Figure 6.2.

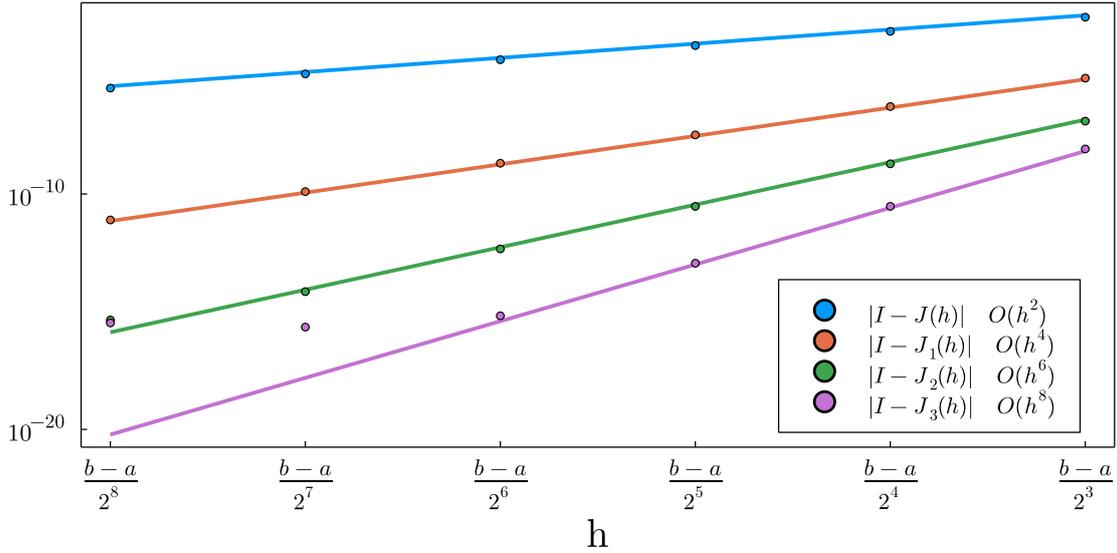


Figure 6.2: Convergence of Romberg's method. The straight lines correspond to the monomial functions  $f(h) = C_i h^i$ , with  $i = 2, 4, 6, 8$  and for appropriate constants  $C_i$ . We observe a good agreement between the observed and theoretical convergence rates.

## 6.4 Methods with non-equidistant nodes

The Newton–Cotes method relies on equidistant integration nodes, and the only degrees of freedom are the integration weights. If the nodes are not fixed, then additional degrees of freedom are available, and these can be leveraged in order to construct a better integration formula. The total number of degrees of freedom for a general integration rule of the form (6.2) is  $2n + 2$ , which enable to construct an integration rule with degree of precision equal to  $2n + 1$ .

A necessary condition for an integration rule of the form (6.2) to have a degree of precision equal to  $2n + 1$  is that it integrates exactly all the monomials of degree 0 to  $2n + 1$ . This condition is also sufficient because, assuming that it is satisfied, we have by linearity of the functionals  $I$  and  $\widehat{I}$  that

$$\begin{aligned} \widehat{I}(\alpha_0 + \alpha_1 x + \cdots + \alpha_{2n+1} x^{2n+1}) &= \alpha_0 \widehat{I}(1) + \alpha_1 \widehat{I}(x) + \cdots + \alpha_{2n+1} \widehat{I}(x^{2n+1}) \\ &= \alpha_0 I(1) + \alpha_1 I(x) + \cdots + \alpha_{2n+1} I(x^{2n+1}) \\ &= I(\alpha_0 + \alpha_1 x + \cdots + \alpha_{2n+1} x^{2n+1}), \end{aligned}$$

Here  $I(u)$  and  $\widehat{I}(u)$  denote respectively the exact integral of  $u$  and its approximate integral using (6.2). Finding the nodes and weights of the integration rule, we can therefore solve the nonlinear system of  $2n + 2$  equations with  $2n + 2$  unknowns:

$$\sum_{i=0}^n w_i x_i^d = \int_{-1}^1 x^d dx, \quad d = 0, \dots, 2n + 1. \quad (6.18)$$

The quadrature rule obtained by solving this system of equations is called the *Gauss–Legendre quadrature*.

*Example 6.1.* Let us derive the Gauss–Legendre quadrature with  $n + 1 = 2$  nodes. The system of equations that we need to solve in this case is the following:

$$w_0 + w_1 = 2, \quad w_0x_0 + w_1x_1 = 0, \quad w_0x_0^2 + w_1x_1^2 = \frac{2}{3}, \quad w_0x_0^3 + w_1x_1^3 = 0.$$

The solution to these equations is given by

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

**Connection with orthogonal polynomials.**  The Gauss–Legendre quadrature rules can be obtained more constructively from orthogonal polynomials, an approach we presented in [Section 5.2.4](#). In particular, the integration nodes are related the roots of a Legendre polynomial, which opens the door to another computational approach for computing them. It is possible to prove, and this should be clear after reading [Section 5.2.4](#), that the integration weights are all positive, and so Gauss–Legendre quadrature rules are less susceptible to roundoff errors than the Newton–Cotes methods.

In this section, we prove, starting from the system of nonlinear equations (6.18), that the roots of the integration formula are necessarily the roots of a Legendre polynomial. This will imply, as a corollary, that the set of integration nodes satisfying (6.18) is unique, because the orthogonal polynomials are unique. To this end, let us denote by

$$p(x) = (x - x_0) \dots (x - x_n) =: \alpha_0 + \alpha_1x + \dots + \alpha_{n+1}x^{n+1}$$

the polynomial whose roots coincide with the unknown integration nodes. Multiplying the first equation of (6.18) ( $d = 0$ ) by  $\alpha_0$ , the second ( $d = 1$ ) by  $\alpha_1$ , and so forth until the equation corresponding to  $d = n + 1$ , we obtain after summing these equations

$$\sum_{i=0}^n w_i \sum_{d=0}^{n+1} \alpha_d x_i^d = \int_{-1}^1 \sum_{d=0}^{n+1} \alpha_d x^d dx \quad \Leftrightarrow \quad \sum_{i=0}^n w_i p(x_i) = \int_{-1}^1 p(x) dx.$$

Since the left-hand side of this equation is equal to 0 by definition of  $p$ , we deduce that  $p$  is orthogonal to the constant polynomial for the inner product

$$\langle f, g \rangle = \int_{-1}^1 f(x) g(x) dx. \quad (6.19)$$

Now if we multiply the *second* equation of (6.18) ( $d = 1$ ) by  $\alpha_0$ , the third ( $d = 2$ ) by  $\alpha_1$ , and so forth until until the equation corresponding to  $d = n + 2$ , we obtain after summation of these equations that

$$\sum_{i=0}^n w_i x_i \sum_{d=0}^{n+1} \alpha_d x_i^d = \int_{-1}^1 \sum_{d=0}^{n+1} \alpha_d x^{d+1} dx \quad \Leftrightarrow \quad \sum_{i=0}^n w_i x_i p(x_i) = \int_{-1}^1 p(x) x dx.$$

Since the left-hand side of this equation is again 0 because the nodes  $x_i$  are the roots of  $p$ , we deduce that  $p$  is orthogonal to the linear polynomial  $x \mapsto x$  for the inner product (6.19). This reasoning can be repeated in order to deduce that  $p$  is in fact orthogonal to all the monomials of degree 0 to  $n$ , implying that  $p$  is a multiple of the Legendre polynomial of degree  $n + 1$ .

**Generalization to higher dimensions.** Gauss–Legendre integration is ubiquitous in numerical methods for partial differential equations, in particular the *finite element method*. Its generalization to higher dimensions is immediate: for a function  $u: [-1, 1] \times [-1, 1] \rightarrow \mathbf{R}$ , we have

$$\int_0^1 \int_0^1 u(x, y) \, dy dx \approx \sum_{i=0}^n \sum_{j=0}^n w_i w_j u(x_i, y_j).$$

The degree of precision of this integration rule is the same as that of the corresponding one-dimensional rule.

## 6.5 Exercises

⚙ **Exercise 6.1.** Derive the Simpson’s integration rule (6.6).

⚙ **Exercise 6.2.** Derive the composite Simpson integration rule (6.9).

⚙ **Exercise 6.3.** Consider the integration rule

$$\int_0^1 u(x) \, dx \approx w_1 u(0) + w_2 u(1) + w_3 u'(0).$$

Find  $w_1$ ,  $w_2$  and  $w_3$  so that this integration rule has the highest possible degree of precision.

⚙ **Exercise 6.4.** Consider the integration rule

$$\int_{-1}^1 u(x) \, dx \approx w_1 u(x_1) + w_2 u'(x_1).$$

Find  $w_1$ ,  $w_2$  and  $x_1$  so that this integration rule has the highest possible degree of precision.

⚙ **Exercise 6.5.** What is the degree of precision of the following quadrature rule?

$$\int_{-1}^1 u(x) \, dx \approx \frac{2}{3} \left( 2u\left(-\frac{1}{2}\right) - u(0) + 2u\left(\frac{1}{2}\right) \right).$$

⚙ **Exercise 6.6.** The Gauss–Hermite quadrature rule with  $n + 1$  nodes is an approximation of the form

$$\int_{-\infty}^{\infty} u(x) e^{-\frac{x^2}{2}} \, dx \approx \sum_{i=0}^n w_i u(x_i),$$

such that the rule is exact for all polynomials of degree less than or equal to  $2n + 1$ . Find the Gauss–Hermite rule with two nodes.

⚙ **Exercise 6.7.** Use Romberg’s method to construct an integration rule with an error term scaling as  $\mathcal{O}(h^4)$ . Is there a link between the method you obtained and another integration rule seen in class?

⚙️ **Exercise 6.8** (Improving the error bound for the composite trapezoidal rule). *The notation used in this exercise is the same as in Section 6.2. In particular,  $\widehat{I}_h$  denotes the approximate integral obtained by using the composite trapezoidal rule (6.7), and  $\widehat{u}_h$  is the corresponding piecewise linear interpolant.*

A version of the mean value theorem states that, if  $g: [a, b] \rightarrow \mathbf{R}$  is a non-negative integrable function and  $f: [a, b] \rightarrow \mathbf{R}$  is continuous, then there exists  $\xi \in (a, b)$  such that

$$\int_a^b f(x)g(x) \, dx = f(\xi) \int_a^b g(x) \, dx. \quad (6.20)$$

- Using (6.20), show that, for all  $i \in \{0, \dots, n-1\}$ , there exists  $\xi_i \in (x_i, x_{i+1})$  such that

$$\int_{x_i}^{x_{i+1}} u(x) - \widehat{u}_h(x) \, dx = -u''(\xi_i) \frac{h^3}{12}.$$

- Prove, by using the intermediate value theorem, that if  $f: [a, b] \rightarrow \mathbf{R}$  is a continuous function, then for any set  $\xi_0, \dots, \xi_{n-1}$  of points within the interval  $(a, b)$ , there exists  $c \in (a, b)$  such that

$$\frac{1}{n} \sum_{i=0}^{n-1} f(\xi_i) = f(c).$$

- Combining the previous items, conclude that there exists  $\xi \in (a, b)$  such that

$$I - \widehat{I}_h = -u''(\xi)(a-b) \frac{h^2}{12},$$

which is a more precise expression of the error than that obtained in (6.8).

*Remark 6.1.* One may convince oneself of (6.20) by rewriting this equation as

$$\frac{\int_a^b f(x)g(x) \, dx}{\int_a^b g(x) \, dx} = f(c).$$

The left-hand side is the average of  $f(x)$  with respect to the probability measure with density given by

$$x \mapsto \frac{g(x)}{\int_a^b g(x) \, dx}.$$

## 6.6 Discussion and bibliography

In this chapter, we covered mainly *deterministic* integration formulas. The presentation of part of the material follows that in [6], and some exercises come from [9, Chapter 9]. Much of the research around the calculation of high-dimensional integrals today is concerned with *probabilistic* integration methods using Monte Carlo approaches. These methods are based on the

connection between integrals and expectations. For example, the integral

$$I = \int_0^1 x^2 dx$$

may be expressed as the expectation  $\mathbf{E}[X^2]$ , where  $\mathbf{E}$  is the expectation operator and  $X \sim \mathcal{U}(0, 1)$  is a uniformly distributed random variable over the interval  $[0, 1]$ . Therefore, in practice,  $I$  may be approximated by generating a large number of samples  $X_1, X_2, \dots$  from the distribution  $\mathcal{U}(0, 1)$  and averaging  $f(X_i)$  over all these samples.

```
n = 1000
f(x) = x^2
X = rand(n)
I_hat = (1/n) * sum(f.(X))
```

The main advantage of this approach is that it generalizes very easily to high-dimensional and infinite-dimensional settings.