Chapter 5
Numerical Integration

Commit your blunders on a small scale and make your profits on a large scale.
—Leo Hendrik Baekeland

5.1 Interpolatory Quadrature Rules

5.1.1 Introduction

In this chapter we study the approximate calculation of a definite integral

$$I[f] = \int_a^b f(x)\,dx, \tag{5.1.1}$$

where $f(x)$ is a given function and $[a, b]$ a finite interval. This problem is often called **numerical quadrature**, since it relates to the ancient problem of the quadrature of the circle, i.e., constructing a square with equal area to that of a circle. The computation of (5.1.1) is equivalent to solving the initial value problem

$$y'(x) = f(x), \quad y(a) = 0, \quad x \in [a, b] \tag{5.1.2}$$

for $y(b) = I[f]$; cf. Sec. 1.5.

As is well known, even many relatively simple integrals cannot be expressed in finite terms of elementary functions, and thus must be evaluated by numerical methods. (For a table of integrals that have closed analytical solutions, see [168].) Even when a closed form analytical solution exists it may be preferable to use a numerical quadrature formula.

Since $I[f]$ is a linear functional, numerical integration is a special case of the problem of approximating a linear functional studied in Sec. 3.3.4. The quadrature rules considered will be of the form

$$I[f] \approx \sum_{i=1}^{n} w_i f(x_i), \tag{5.1.3}$$

where $x_1 < x_2 < \cdots < x_n$ are distinct **nodes** and $w_1, w_2, \ldots, w_n$ the corresponding **weights**. Often (but not always) all nodes lie in $[a, b]$.

The weights w_i are usually determined so that the formula (5.1.3) is exact for polynomials of as high degree as possible. The accuracy therefore depends on how well the integrand $f(x)$ can be approximated by a polynomial in $[a, b]$. If the integrand has a singularity, for example, it becomes infinite at some point in or near the interval of integration, some modification is necessary. Another complication arises when the interval of integration is infinite. In both cases it may be advantageous to consider a weighted quadrature rule:

$$\int_a^b f(x)w(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i). \tag{5.1.4}$$

Here $w(x) \geq 0$ is a **weight function** (or density function) that incorporates the singularity so that $f(x)$ can be well approximated by a polynomial. The limits (a, b) of integration are now allowed to be infinite.

To ensure that the integral (5.1.4) is well defined when $f(x)$ is a polynomial, we assume in the following that the integrals

$$\mu_k = \int_a^b x^k w(x)\,dx, \quad k = 1, 2, \ldots, \tag{5.1.5}$$

are defined for all $k \geq 0$, and $\mu_0 > 0$. The quantity μ_k is called the kth **moment** with respect to the weight function $w(x)$. Note that for the formula (5.1.4) to be exact for $f(x) = 1$ it must hold that

$$\mu_0 = \int_a^b 1 \cdot w(x)\,dx = \sum_{i=1}^{n} w_i. \tag{5.1.6}$$

In the special case that $w(x) = 1$, we have $\mu_0 = b - a$.

Definition 5.1.1.

A quadrature rule (5.1.3) *has* **order of accuracy** *(or degree of exactness) equal to d if it is exact for all polynomials of degree $\leq d$, i.e., for all $p \in \mathcal{P}_{d+1}$.*

In a weighted **interpolatory** quadrature formula the integral is approximated by

$$\int_a^b p(x)w(x)\,dx,$$

where $p(x)$ is the unique polynomial of degree $n-1$ interpolating $f(x)$ at the distinct points $x_1, x_2, \ldots, x_n$. By Lagrange's interpolation formula (Theorem 4.1.1)

$$p(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x), \qquad \ell_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)},$$

where $\ell_i(x)$ are the elementary Lagrange polynomials associated with the nodes $x_1, x_2, \ldots, x_n$. It follows that for an interpolatory quadrature formula the weights are given by

$$w_i = \int_a^b \ell_i(x)w(x)\,dx. \tag{5.1.7}$$

In practice, the coefficients are often more easily computed using the method of undetermined coefficients rather than by integrating $\ell_i(x)$.

An expression for the truncation error is obtained by integrating the remainder (see Theorems 4.2.3 and 4.2.4):

$$R_n(f) = \int_a^b [x_1, \ldots, x_n, x] f \prod_{i=1}^{n} (x - x_i) w(x)\, dx$$
$$= \frac{1}{n!} \int_a^b f^{(n)}(\xi_x) \prod_{i=1}^{n} (x - x_i) w(x)\, dx, \quad \xi_x \in [a, b], \tag{5.1.8}$$

where the second expression holds if $f^{(n)}$ is continuous in $[a, b]$.

Theorem 5.1.2.

For any given set of nodes $x_1, x_2, \ldots, x_n$ an interpolatory quadrature formula with weights given by (5.1.7) *has order of exactness equal to at least $n - 1$. Conversely, if the formula has degree of exactness $n - 1$, then the formula must be interpolatory.*

Proof. For any $f \in \mathcal{P}_n$ we have $p(x) = f(x)$, and hence (5.1.7) has degree of exactness at least equal to $n - 1$. On the other hand, if the degree of exactness of (5.1.7) is $n - 1$, then putting $f = \ell_i(x)$ shows that the weights w_i satisfy (5.1.7); that is, the formula is interpolatory. □

In general the function values $f(x_i)$ cannot be evaluated exactly. Assume that the error in $f(x_i)$ is e_i, where $|e_i| \leq \epsilon$, $i = 1 : n$. Then, if $w_i \geq 0$, the related error in the quadrature formula satisfies

$$\Big| \sum_{i=1}^{n} w_i e_i \Big| \leq \epsilon \sum_{i=1}^{n} |w_i| \leq \epsilon \mu_0. \tag{5.1.9}$$

The last upper bound holds only if *all weights in the quadrature rules are positive*.

So far we have assumed that all the nodes x_i of the quadrature formula are fixed. A natural question is whether we can do better by a judicious choice of the nodes. This question is answered positively in the following theorem. Indeed, by a careful choice of nodes the order of accuracy of the quadrature rule can be substantially improved.

Theorem 5.1.3.

Let k be an integer such that $0 \leq k \leq n$. Consider the integral

$$I[f] = \int_a^b f(x) w(x)\, dx,$$

and an interpolatory quadrature rule

$$I_n(f) = \sum_{i=1}^{n} w_i f(x_i),$$

using n nodes. Let

$$\gamma(x) = \prod_{i=1}^{n}(x - x_i) \tag{5.1.10}$$

be the corresponding **node polynomial**. *Then the quadrature rule $I[f] \approx I_n(f)$ has degree of exactness equal to $d = n + k - 1$ if and only if, for all polynomials $p \in \mathcal{P}_k$, the node polynomial satisfies*

$$\int_a^b p(x)\gamma(x)w(x)\,dx = 0. \tag{5.1.11}$$

Proof. We first prove the *necessity* of the condition (5.1.11). For any $p \in \mathcal{P}_k$ the product $p(x)\gamma(x)$ is in $\mathcal{P}_{n+k}$. Then since $\gamma(x_i) = 0$, $i = 1 : n$,

$$\int_a^b p(x)\gamma(x)w(x)\,dx = \sum_{i=1}^{n} w_i f(x_i)\gamma(x_i) = 0,$$

and thus (5.1.11) holds.

To prove the *sufficiency*, let $p(x)$ be any polynomial of degree $n + k - 1$. Let $q(x)$ and $r(x)$ be the quotient and remainder, respectively, in the division

$$p(x) = q(x)\gamma(x) + r(x).$$

Then $q(x)$ and $r(x)$ are polynomials of degree $k - 1$ and $n - 1$, respectively. It holds that

$$\int_a^b p(x)w(x)\,dx = \int_a^b q(x)\gamma(x)w(x)\,dx + \int_a^b r(x)w(x)\,dx,$$

where the first integral on the right-hand side is zero because of the orthogonality property of $\gamma(x)$. For the second integral we have

$$\int_a^b r(x)w(x)\,dx = \sum_{i=1}^{n} w_i r(x_i),$$

since the weights were chosen such that the formula was interpolatory and therefore exact for all polynomials of degree $n - 1$. Further, since

$$p(x_i) = q(x_i)\gamma(x_i) + r(x_i) = r(x_i), \quad i = 1 : n,$$

it follows that

$$\int_a^b p(x)w(x)\,dx = \int_a^b r(x)w(x)\,dx = \sum_{i=1}^{n} w_i r(x_i) = \sum_{i=1}^{n} w_i p(x_i),$$

which shows that the quadrature rule is exact for $p(x)$. □

How to determine quadrature rules of optimal order will be the topic of Sec. 5.3.

5.1.2 Treating Singularities

When the integrand or some of its low-order derivative is infinite at some point in or near the interval of integration, standard quadrature rules will not work well. It is not uncommon that a single step taken close to such a singular point will give a larger error than all other steps combined. In some cases a singularity can be completely missed by the quadrature rule.

If the singular points are known, then the integral should first be broken up into several pieces so that all the singularities are located at one (or both) ends of the interval $[a, b]$. Many integrals can then be treated by weighted quadrature rules, i.e., the singularity is incorporated into the weight function. Romberg's method can be modified to treat integrals where the integrand has an algebraic endpoint singularity; see Sec. 5.2.2.

It is often profitable to investigate whether one can transform or modify the given problem analytically to make it more suitable for numerical integration. Some difficulties and possibilities in numerical integration are illustrated below in a series of simple examples.

Example 5.1.1.

In the integral

$$I = \int_0^1 \frac{1}{\sqrt{x}} e^x \, dx$$

the integrand is infinite at the origin. By the substitution $x = t^2$ we get

$$I = 2 \int_0^1 e^{t^2} \, dt,$$

which can be treated without difficulty.

Another possibility is to use integration by parts:

$$\begin{aligned} I &= \int_0^1 x^{-1/2} e^x \, dx = 2x^{1/2} e^x \big|_0^1 - 2 \int_0^1 x^{1/2} e^x \, dx \\ &= 2e - 2\frac{2}{3} x^{3/2} e^x \big|_0^1 + \frac{4}{3} \int_0^1 x^{3/2} e^x \, dx = \frac{2}{3} e + \frac{4}{3} \int_0^1 x^{3/2} e^x \, dx. \end{aligned}$$

The last integral has a mild singularity at the origin. If one wants high accuracy, then it is advisable to integrate by parts a few more times before the numerical treatment.

Example 5.1.2.

Sometimes a simple comparison problem can be used. In

$$I = \int_{0.1}^1 x^{-3} e^x \, dx$$

the integrand is infinite near the left endpoint. If we write

$$I = \int_{0.1}^1 x^{-3} \Big(1 + x + \frac{x^2}{2}\Big) \, dx + \int_{0.1}^1 x^{-3} \Big(e^x - 1 - x - \frac{x^2}{2}\Big) \, dx,$$

the first integral can be computed analytically. The second integrand can be treated numerically. The integrand and its derivatives are of moderate size. Note, however, the cancellation in the evaluation of the integrand.

For integrals over an infinite interval one can try some substitution which maps the interval $(0, \infty)$ to $(0, 1)$, for example, $t = e^{-x}$ of $t = 1/(1 + x)$. But in such cases one must be careful not to introduce an unpleasant singularity into the integrand instead.

Example 5.1.3.
More general integrals of the form

$$\int_0^{2h} x^{-1/2} f(x)\, dx$$

need a special treatment, due to the integrable singularity at $x = 0$. A formula which is exact for any second-degree polynomial $f(x)$ can be found using the method of undetermined coefficients. We set

$$\frac{1}{\sqrt{2h}} \int_0^{2h} x^{-1/2} f(x)\, dx \approx w_0 f(0) + w_1 f(h) + w_2 f(2h),$$

and equate the left- and right-hand sides for $f(x) = 1, x, x^2$. This gives

$$w_0 + w_1 + w_2 = 2, \qquad \frac{1}{2} w_1 + w_2 = \frac{2}{3}, \qquad \frac{1}{4} w_1 + w_2 = \frac{2}{5}.$$

This linear system is easily solved, giving $w_0 = 12/15$, $w_1 = 16/15$, $w_2 = 2/15$.

Example 5.1.4.
Consider the integral

$$I = \int_0^{\infty} (1 + x^2)^{-4/3}\, dx.$$

If one wants five decimal digits in the result, then $\int_R^{\infty}$ is not negligible until $R \approx 10^3$. But one can expand the integrand in powers of x^{-1} and integrate termwise:

$$\begin{aligned} \int_R^{\infty} (1 + x^2)^{-4/3}\, dx &= \int_R^{\infty} x^{-8/3} (1 + x^{-2})^{-4/3}\, dx \\ &= \int_R^{\infty} \Big(x^{-8/3} - \frac{4}{3} x^{-14/3} + \frac{14}{9} x^{-20/3} - \cdots \Big) \\ &= R^{-5/3} \Big(\frac{3}{5} - \frac{4}{11} R^{-2} + \frac{14}{51} R^{-4} - \cdots \Big). \end{aligned}$$

If this expansion is used, then one need only apply numerical integration to the interval $[0, 8]$.

With the substitution $t = 1/(1 + x)$ the integral becomes

$$I = \int_0^1 (t^2 + (1 - t)^2)^{-4/3} t^{2/3}\, dt.$$

The integrand now has an infinite derivative at the origin. This can be eliminated by making the substitution $t = u^3$ to get

$$I = \int_0^1 (u^6 + (1-u^3)^2)^{-4/3} 3u^4\, du,$$

which can be computed with, for example, a Newton–Cotes' method.

5.1.3 Some Classical Formulas

Interpolatory quadrature formulas, where the nodes are constrained to be equally spaced, are called **Newton–Cotes**[169] formulas. These are especially suited for integrating a tabulated function, a task that was more common before the computer age. The midpoint, trapezoidal, and Simpson's rules, to be described here, are all special cases of (unweighted) Newton–Cotes' formulas.

The **trapezoidal rule** (cf. Figure 1.1.5) is based on linear interpolation of $f(x)$ at $x_1 = a$ and $x_2 = b$; i.e., $f(x)$ is approximated by

$$p(x) = f(a) + (x-a)[a,b]f = f(a) + (x-a)\frac{f(b)-f(a)}{b-a}.$$

The integral of $p(x)$ equals the area of a trapezoid with base $(b-a)$ times the average height $\frac{1}{2}(f(a)+f(b))$. Hence

$$\int_a^b f(x)\, dx \approx \frac{(b-a)}{2}(f(a)+f(b)).$$

To increase the accuracy we subdivide the interval $[a,b]$ and assume that $f_i = f(x_i)$ is known on a grid of equidistant points

$$x_0 = a, \quad x_i = x_0 + ih, \quad x_n = b, \tag{5.1.12}$$

where $h = (b-a)/n$ is the **step length**. The trapezoidal approximation for the ith subinterval is

$$\int_{x_i}^{x_{i+1}} f(x)\, dx = T(h) + R_i, \quad T(h) = \frac{h}{2}(f_i + f_{i+1}). \tag{5.1.13}$$

Assuming that $f''(x)$ is continuous in $[a,b]$ and using the exact remainder in Newton's interpolation formula (see Theorem 4.2.1) we get

$$R_i = \int_{x_i}^{x_{i+1}} (f(x) - p_2(x))\, dx = \int_{x_i}^{x_{i+1}} (x-x_i)(x-x_{i+1})\,[x_i, x_{i+1}, x]f\, dx. \tag{5.1.14}$$

Since $[x_i, x_{i+1}, x]f$ is a continuous function of x and $(x-x_i)(x-x_{i+1})$ has constant (negative) sign for $x \in [x_i, x_{i+1}]$, the mean value theorem of integral calculus gives

$$R_i = [x_i, x_{i+1}, \xi_i]f \int_{x_i}^{x_{i+1}} (x-x_i)(x-x_{i+1})\, dx, \quad \xi_i \in [x_i, x_{i+1}].$$

[169] Roger Cotes (1682–1716) was a highly appreciated young colleague of Isaac Newton. He was entrusted with the preparation of the second edition of Newton's *Principia*. He worked out and published the coefficients for Newton's formulas for numerical integration for $n \le 11$.

Setting $x = x_i + ht$ and using the Theorem 4.2.3, we get

$$R_i = -\frac{1}{2}f''(\zeta_i)\int_0^1 h^2 t(t-1)h\,dt = -\frac{1}{12}h^3 f''(\zeta_i), \quad \zeta_i \in [x_i, x_{i+1}]. \tag{5.1.15}$$

For another proof of this result using the Peano kernel see Example 3.3.16.

Summing the contributions for each subinterval $[x_i, x_{i+1}]$, $i = 0:n$, gives

$$\int_a^b f(x)\,dx = T(h) + R_T, \quad T(h) = \frac{h}{2}(f_0 + f_n) + h\sum_{i=2}^{n-1} f_i, \tag{5.1.16}$$

which is the **composite trapezoidal rule**. The **global** truncation error is

$$R_T = -\frac{h^3}{12}\sum_{i=0}^{n-1} f''(\zeta_i) = -\frac{1}{12}(b-a)h^2 f''(\xi), \quad \xi \in [a, b]. \tag{5.1.17}$$

(The last equality follows since f'' was assumed to be continuous on the interval $[a, b]$.) This shows that by choosing h small enough we can make the truncation error arbitrarily small. In other words, we have **asymptotic convergence** when $h \to 0$.

In the **midpoint rule** $f(x)$ is approximated on $[x_i, x_{i+1}]$ by its value

$$f_{i+1/2} = f(x_{i+1/2}), \quad x_{i+1/2} = \frac{1}{2}(x_i + x_{i+1}),$$

at the midpoint of the interval. This leads to the approximation

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = M(h) + R_i, \quad M(h) = hf_{i+1/2}. \tag{5.1.18}$$

The midpoint rule approximation can be interpreted as the area of the trapezium defined by the tangent of f at the midpoint $x_{i+1/2}$.

The remainder term in Taylor's formula gives

$$f(x) - (f_{i+1/2} + (x - x_{i+1/2})f'_{i+1/2}) = \frac{1}{2}(x - x_{i+1/2})^2 f''(\zeta_x), \quad \zeta_x \in [x_i, x_{i+1/2}].$$

By symmetry the integral over $[x_i, x_{i+1}]$ of the linear term vanishes. We can use the mean value theorem to show that

$$R_i = \int_{x_i}^{x_{i+1}} \frac{1}{2}f''(\zeta_x)(x - x_{i+1/2})^2\,dx = \frac{1}{2}f''(\zeta_i)\int_{-1/2}^{1/2} h^3 t^2\,dt = \frac{h^3}{24}f''(\zeta_i).$$

Although it uses just *one function value*, the midpoint rule, like the trapezoidal rule, is exact when $f(x)$ is a linear function. Summing the contributions for each subinterval, we obtain the **composite midpoint rule**:

$$\int_a^b f(x)\,dx = M(h) + R_M, \quad M(h) = h\sum_{i=0}^{n-1} f_{i+1/2}. \tag{5.1.19}$$

(Compare the above approximation with the Riemann sum in the *definition* of a definite integral.) For the global error we have

$$R_M = \frac{(b-a)h^2}{24} f''(\zeta), \quad \zeta \in [a, b]. \tag{5.1.20}$$

The trapezoidal rule is called a **closed rule** because values of f at both endpoints are used. It is not uncommon that f has an integrable singularity at an endpoint. In that case an **open rule**, like the midpoint rule, can still be applied.

If $f''(x)$ has constant sign in each subinterval, then the error in the midpoint rule is approximately half as large as that for the trapezoidal rule and has the opposite sign. But the trapezoidal rule is more economical to use when a sequence of approximations for $h, h/2, h/4, \ldots$ is to be computed, since about half of the values needed for $h/2$ were already computed and used for h. Indeed, it is easy to verify the following useful relation between the trapezoidal and midpoint rules:

$$T\left(\frac{h}{2}\right) = \frac{1}{2}(T(h) + M(h)). \tag{5.1.21}$$

If the magnitude of the error in the function values does not exceed $\frac{1}{2}U$, then the magnitude of the propagated error in the approximation for the trapezoidal and midpoint rules is bounded by

$$R_A = \frac{1}{2}(b-a)U, \tag{5.1.22}$$

independent of h. By (5.1.9) this holds *for any quadrature formula of the form* (5.1.3), *provided that all weights* w_i *are positive*.

If the roundoff error is negligible and h sufficiently small, then it follows from (5.1.17) that the error in $T(h/2)$ is about one-quarter of that in $T(h)$. Hence the magnitude of the error in $T(h/2)$ can be estimated by $(1/3)|T(h/2) - T(h)|$, or more conservatively by $|T(h/2) - T(h)|$. (A more systematic use of Richardson extrapolation is made in Romberg's method; see Sec. 5.2.2.)

Example 5.1.5.

Use (5.1.21) to compute the sine integral $\mathrm{Si}(x) = \int_0^x \frac{\sin t}{t}\, dt$ for $x = 0.8$. Midpoint and trapezoidal sums (correct to eight decimals) are given below.

h	$M(h)$	$T(h)$
0.8	0.77883 668	0.75867 805
0.4	0.77376 698	0.76875 736
0.2	0.77251 272	0.77126 217
0.1		0.77188 744

The correct value, to ten decimals, is 0.77209 57855 (see [1, Table 5.2]). We verify that in this example the error is approximately proportional to h^2 for both $M(h)$ and $T(h)$, and we estimate the error in $T(0.1)$ to be $\frac{1}{3}6.26 \cdot 10^{-4} \leq 2.1 \cdot 10^{-4}$.

From the error analysis above we note that the error in the midpoint rule is roughly half the size of the error in the trapezoidal rule and of opposite sign. Hence it seems that the linear combination

$$S(h) = \frac{1}{3}(T(h) + 2M(h)) \tag{5.1.23}$$

should be a better approximation. This is indeed the case and (5.1.23) is equivalent to **Simpson's rule**.[170]

Another way to derive Simpson's rule is to approximate $f(x)$ by a piecewise polynomial of third degree. It is convenient to shift the origin to the midpoint of the interval and consider the integral over the interval $[x_i - h, x_i + h]$. From Taylor's formula we have

$$f(x) = f_i + (x - x_i)f_i' + \frac{(x - x_i)^2}{2}f_i'' + \frac{(x - x_i)^3}{3!}f_i''' + O(h^4),$$

where the remainder is zero for all polynomials of degree three or less. Integrating term by term, the integrals of the second and fourth term vanish by symmetry, giving

$$\int_{x_i-h}^{x_i+h} f(x)\,dx = 2hf_i + 0 + \frac{1}{3}h^3 f_i'' + 0 + O(h^5).$$

Using the difference approximation $h^2 f_i'' = (f_{i-1} - 2f_i + f_{i+1}) + O(h^4)$ (see (4.7.5)) we obtain

$$\begin{aligned}\int_{x_i-h}^{x_i+h} f(x)\,dx &= 2hf_i + \frac{1}{3}h(f_{i-1} - 2f_i + f_{i+1}) + O(h^5) \\ &= \frac{1}{3}h(f_{i-1} + 4f_i + f_{i+1}) + O(h^5),\end{aligned} \tag{5.1.24}$$

where the remainder term is zero for all third-degree polynomials. We now determine the error term for $f(x) = (x - x_i)^4$, which is

$$R_T = \frac{1}{3}h(h^4 + 0 + h^4) - \int_{x_i-h}^{x_i+h} x^4\,dx = \left(\frac{2}{3} - \frac{2}{5}\right)h^5 = \frac{4}{15}h^5.$$

It follows that an *asymptotic error estimate* for Simpson's rule is

$$R_T = h^5\frac{4}{15}\frac{f^{(4)}(x_i)}{4!} + O(h^6) = \frac{h^5}{90}f^{(4)}(x_i) + O(h^6). \tag{5.1.25}$$

A strict error estimate for Simpson's rule is more difficult to obtain. As for the midpoint formula, the midpoint x_i can be considered as a **double point** of interpolation; see Problem 5.1.3. The general error formula (5.1.8) then gives

$$R_i(f) = \frac{1}{4!}\int_{x_{i-1}}^{x_{i+1}} f^{(4)}(\xi_x)(x - x_{i-1})(x - x_i)^2(x - x_{i+1})\,dx,$$

[170] Thomas Simpson (1710–1761), English mathematician best remembered for his work on interpolation and numerical methods of integration. He taught mathematics privately in the London coffee houses and from 1737 began to write texts on mathematics.

where $(x - x_{i-1})(x - x_i)^2(x - x_{i+1})$ has constant negative sign on $[x_{i-1}, x_{i+1}]$. Using the mean value theorem gives the error

$$R_T(f) = \frac{1}{90} f^{(4)}(\xi) h^5, \quad \xi \in [x_i - h, x_i + h]. \tag{5.1.26}$$

The remainder can also be obtained from Peano's error representation. It can be shown (see [331, p. 152 ff]) that for Simpson's rule

$$Rf = \int_{\mathbf{R}} f^{(4)}(u) K(u)\, du,$$

where the kernel equals

$$K(u) = -\frac{1}{72}(h - u)^3(3u + h)^2, \quad 0 \le u \le h,$$

and $K(u) = K(|u|)$ for $u < 0$, $K(u) = 0$ for $|u| > h$. This again gives (5.1.26).

In the **composite Simpson's rule** one divides the interval $[a, b]$ into an *even* number $n = 2m$ steps of length h and uses the formula (5.1.24) on each of m double steps, giving

$$\int_a^b f(x)\, dx = \frac{h}{3}(f_0 + 4S_1 + 2S_2 + f_n) + R_T, \tag{5.1.27}$$

where

$$S_1 = f_1 + f_3 + \cdots + f_{n-1}, \qquad S_2 = f_2 + f_4 + \cdots + f_{n-2}$$

are sums over odd and even indices, respectively. The remainder is

$$R_T(f) = \sum_{i=0}^{m-1} \frac{h^5}{90} f^{(4)}(\xi_i) = \frac{(b-a)}{180} h^4 f^{(4)}(\xi), \quad \xi \in [a, b]. \tag{5.1.28}$$

This shows that we have gained *two orders of accuracy* compared to the trapezoidal rule, without using more function evaluations. This is why Simpson's rule is such a popular general-purpose quadrature rule.

5.1.4 Superconvergence of the Trapezoidal Rule

In general the composite trapezoidal rule integrates exactly polynomials of degree 1 only. It does much better with trigonometric polynomials.

Theorem 5.1.4.
Consider the integral $\int_0^{2\pi} t_m(x)\, dx$, where

$$\begin{aligned} t_m(x) = a_0 &+ a_1 \cos x + a_2 \cos 2x + \cdots + a_m \cos mx \\ &+ b_1 \sin x + b_2 \sin 2x + \cdots + b_m \sin mx \end{aligned}$$

is any trigonometric polynomial of degree m. Then the composite trapezoidal rule with step length $h = 2\pi/n$, $n \ge m + 1$, integrates this exactly.

Proof. See Problem 5.1.16. □

Suppose that the function f is infinitely differentiable for $x \in \mathbf{R}$, and that f *has* $[a, b]$ *as an interval of periodicity*, i.e., $f(x + (b - a)) = f(x)$ for all $x \in \mathbf{R}$. Then

$$f^{(k)}(b) = f^{(k)}(a), \quad k = 0, 1, 2, \ldots,$$

hence *every term in the Euler–Maclaurin expansion is zero* for the integral over *the whole period* $[a, b]$. One could be led to believe that the trapezoidal rule gives the exact value of the integral, but this is usually not the case. For most periodic functions f, $\lim_{r\to\infty} R_{2r+2} f \neq 0$; *the expansion converges, of course, though not necessarily to the correct result.*

On the other hand, the convergence as $h \to 0$ for a fixed (though arbitrary) r is a different story; the error bound (5.2.10) shows that

$$|R_{2r+2}(a, h, b) f| = O(h^{2r+2}).$$

Since r is arbitrary, this means that *for this class of functions, the trapezoidal error tends to zero faster than any power of h, as* $h \to 0$. We may call this **superconvergence**. The application of the trapezoidal rule to an integral over $[0, \infty)$ of a function $f \in C^\infty(0, \infty)$ often yields similar features, sometimes even more striking.

Suppose that the periodic function $f(z)$, $z = x + iy$, is analytic in a strip, $|y| < c$, around the real axis. It can then be shown that the error of the trapezoidal rule is

$$O(e^{-\eta/h}), \quad h \downarrow 0,$$

where η is related to the width of the strip. A similar result (3.2.19) was obtained in Sec. 3.2.2, for an annulus instead of a strip. There the trapezoidal rule was used in the calculation of the integral of a periodic analytic function over a full period $[0, 2\pi]$ that defined its Taylor coefficients. The error was shown to tend to zero faster than any power of the step length $\Delta\theta$.

As a rule, this discussion does *not* apply to periodic functions which are defined by periodic continuation of a function originally defined on $[a, b]$ (such as the Bernoulli functions). They usually become nonanalytic at a and b, and at all points $a + (b - a)n$, $n = 0, \pm 1, \pm 2, \ldots$.

The **Poisson summation formula** is even better than the Euler–Maclaurin formula for the quantitative study of the trapezoidal truncation error on an infinite interval. For convenient reference we now formulate the following surprising result.

Theorem 5.1.5.

Suppose that the trapezoidal rule (or, equivalently, the rectangle rule) is applied with constant *step size h to* $\int_{-\infty}^{\infty} f(x)\,dx$. *The Fourier transform of f reads*

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega t}\,dt.$$

Then the integration error decreases like $2\hat{f}(2\pi/h)$ *as* $h \downarrow 0$.

Example 5.1.6.

For the normal probability density, we have

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{t}{\sigma})^2}, \qquad \hat{f}(\omega) = e^{-\frac{1}{2}(\omega\sigma)^2}.$$

The integration error is thus approximately $2\exp(-2(\pi\sigma/h)^2)$. Roughly speaking, the number of correct digits is doubled if h is divided by $\sqrt{2}$; for example, the error is approximately $5.4 \cdot 10^{-9}$ for $h = \sigma$, and $1.4 \cdot 10^{-17}$ for $h = \sigma/\sqrt{2}$.

The application of the trapezoidal rule to an integral over $[0, \infty)$ of a function $f \in C^\infty(0, \infty)$ often yields similar features, sometimes even more striking. Suppose that, for $k = 1, 2, 3, \ldots$,

$$f^{(2k-1)}(0) = 0 \text{ and } f^{(2k-1)}(x) \to 0, \quad x \to \infty,$$

and

$$\int_0^\infty |f^{(2k)}(x)|\, dx < \infty.$$

(Note that for any function $g \in C^\infty(-\infty, \infty)$ the function $f(x) = g(x) + g(-x)$ satisfies such conditions at the origin.) Then all terms of the Euler–Maclaurin expansion are zero, and one can be misled to believe that the trapezoidal sum gives $\int_0^\infty f(x)\, dx$ exactly for any step size h! The explanation is that the remainder $R_{2r+2}(a, h, \infty)$ will typically not tend to zero, as $r \to \infty$ for fixed h. On the other hand, if we consider the behavior of the truncation error as $h \to 0$ for given r, we find that it is $o(h^{2r})$ for any r, just like the case of a periodic function.

For a finite subinterval of $[0, \infty)$, however, the remainder is still typically $O(h^2)$, and for each step the remainder is typically $O(h^3)$. So, there is an *enormous cancellation of the local truncation errors*, when a C^∞-function with vanishing odd-order derivatives at the origin is integrated by the trapezoidal rule over $[0, \infty)$.

Example 5.1.7.

For integrals of the form $\int_{-\infty}^\infty f(x)\, dx$, the trapezoidal rule (or the midpoint rule) often gives good accuracy if one integrates over the interval $[-R_1, R_2]$, assuming that $f(x)$ and its lower derivatives are small for $x \le -R_1$ and $x \ge R_2$.

The correct value to six decimal digits of the integral $\int_{-\infty}^\infty e^{-x^2}\, dx$ is $\pi^{1/2} = 1.772454$. For $x \pm 4$, the integrand is less than $0.5 \cdot 10^{-6}$. Using the trapezoidal rule with $h = 1/2$ for the integral over $[-4, 4]$ we get the estimate 1.772453, an amazingly good result. (The function values have been taken from a six-place table.) The truncation error in the value of the integral is here less than 1/10,000 of the truncation error in the largest term of the trapezoidal sum—a superb example of "cancellation of truncation error." The error committed when we replace ∞ by 4 can be estimated in the following way:

$$|R| = 2\int_4^\infty e^{-x^2}\, dx = \int_{16}^\infty e^{-t}\frac{1}{\sqrt{t}}\, dt < \int_{16}^\infty e^{-t}\frac{1}{\sqrt{16}}\, dt = \frac{1}{4}e^{-16} < 10^{-7}.$$

5.1.5 Higher-Order Newton–Cotes' Formulas

The classical Newton–Cotes' quadrature rules are interpolatory rules obtained for $w(x) = 1$ and equidistant points in $[0, 1]$. There are two classes: **closed formulas**, where the endpoints of the interval belong to the nodes, and **open formulas**, where all nodes lie strictly in the interior of the interval (cf. the trapezoidal and midpoint rules).

The closed Newton–Cotes' formulas are usually written as

$$\int_0^{nh} f(x)\,dx = h\sum_{j=0}^{n} w_j f(jh) + R_n(f). \tag{5.1.29}$$

The weights satisfy $w_j = w_{n-j}$ and can, in principle, be determined from (5.1.7). Further, by (5.1.6) it holds that

$$\sum_{j=0}^{n} hw_j = nh. \tag{5.1.30}$$

(Note that here we sum over $n+1$ points in contrast to our previous notation.)

It can be shown that the closed Newton–Cotes' formula has order of accuracy $d = n$ for n odd and $d = n+1$ for n even. The extra accuracy for n even is, as in Simpson's rule, due to symmetry. For $n \le 7$ all coefficients w_i are positive, but for $n = 8$ and $n \ge 10$ negative coefficients appear. Such formulas may still be useful, but since $\sum_{j=0}^{n} h|w_j| > nh$, they are less robust with respect to errors in the function values f_i.

The closed Newton–Cotes' rules for $n = 1$ and $n = 2$ are equivalent to the trapezoidal rule and Simpson's rule, respectively. The formula for $n = 3$ is called the 3/8th rule, for $n = 4$ Milne's rule, and for $n = 6$ Weddle's rule. The weights $w_i = Ac_i$ and error coefficient $c_{n,d}$ of Newton–Cotes' closed formulas are given for $n \le 6$ in Table 5.1.1.

Table 5.1.1. *The coefficients $w_i = Ac_i$ in the n-point closed Newton–Cotes' formulas.*

n	d	A	c_0	c_1	c_2	c_3	c_4	c_5	c_6	$c_{n,d}$
1	1	1/2	1	1						−1/12
2	3	1/3	1	4	1					−1/90
3	3	3/8	1	3	3	1				−3/80
4	5	2/45	7	32	12	32	7			−8/945
5	5	5/288	19	75	50	50	75	19		−275/12,096
6	7	1/140	41	236	27	272	27	236	41	−9/1400

The open Newton–Cotes' formulas are usually written as

$$\int_0^{nh} f(x)\,dx = h\sum_{i=1}^{n-1} w_i f(ih) + R_{n-1,n}(h)$$

and use $n-1$ nodes. The weights satisfy $w_{-j} = w_{n-j}$. The simplest open Newton–Cotes' formula for $n = 2$ is the midpoint rule with step size $2h$. The open formulas have order of accuracy $d = n-1$ for n even and $d = n-2$ for n odd. For the open formulas negative coefficients occur already for $n = 4$ and $n = 6$.

The weights and error coefficients of open formulas for $n \le 5$ are given in Table 5.1.2. We recognize the midpoint rule for $n = 2$. Note that the sign of the error coefficients in the open rules are opposite the sign in the closed rules.

Table 5.1.2. *The coefficients $w_i = Ac_i$ in the n-point open Newton–Cotes' formulas.*

n	d	A	c_1	c_2	c_3	c_4	c_5		$c_{n,d}$
2	1	2	1						1/24
3	1	3/2	1	1					1/4
4	3	4/3	2	−1	2				14/45
5	3	5/24	11	1	1	11			95/144
6	5	3/10	11	−14	26	−14	11		41/140
7	5	7/1440	611	−453	562	562	−453	611	5257/8640

The Peano kernels for both the open and the closed formulas can be shown to have constant sign (Steffensen [323]). Thus the local truncation error can be written as

$$R_n(h) = c_{n,d} h^{d+1} f^{(d)}(\zeta), \quad \zeta \in [0, nh]. \tag{5.1.31}$$

It is easily shown that the Peano kernels for the corresponding composite formulas also have constant sign.

Higher-order Newton–Cotes' formulas can be found in [1, pp. 886–887]. We now show how they can be derived using the operator methods developed in Sec. 3.3.4. Let m, n be given integers and let h be a positive step size. In order to utilize the symmetry of the problem more easily, we move the origin to the midpoint of the interval of integration. If we set

$$x_j = jh, \qquad f_j = f(jh), \quad j = -n/2 : 1 : n/2,$$

the Newton–Cotes' formula now reads

$$\int_{-mh/2}^{mh/2} f(x)\,dx = h \sum_{j=-n/2}^{n/2} w_j f_j + R_{m,n}(h), \quad w_{-j} = w_j. \tag{5.1.32}$$

Note that j, $n/2$, and $m/2$ are not necessarily integers. For a Newton–Cotes' formula, $n/2 - j$ and $m/2 - j$ are evidently integers and hence $(m - n)/2$ is an integer too. There may, however, be other formulas, perhaps almost as good, where this is not the case. The coefficients $w_j = w_{j;m,n}$ are to be determined so that the remainder $R_{m,n}$ vanishes if $f \in \mathcal{P}_q$, with q as large as possible for given m, n.

The left-hand side of (5.1.32), divided by h, reads in operator form

$$(e^{mhD/2} - e^{-mhD/2})(hD)^{-1} f(x_0),$$

which is an even function of hD. By (3.3.42), hD is an odd function of δ. It follows that the left-hand side is an even function of δ; hence we can, for every m, write

$$(e^{hDm/2} - e^{-hDm/2})(hD)^{-1} \;\mapsto A_m(\delta^2) = a_{1m} + a_{2m}\delta^2 + \cdots + a_{k+1,m}\delta^{2k} \cdots. \tag{5.1.33}$$

We truncate after (say) δ^{2k}; the first neglected term is then $a_{k+2,m}\delta^{2k+2}$. We saw in Sec. 3.3.4 how to bring a truncated δ^2-expansion to $B(E)$-form,

$$b_1 + b_2(E + E^{-1}) + b_3(E^2 + E^{-2}) + \cdots + b_k(E^k + E^{-k}),$$

by matrix multiplication with a matrix M of the form given in (3.3.49). By comparison with (5.1.32), we conclude that $n/2 = k$, that the indices j are integers, and that $w_j = b_{j+1}$ (if $j \geq 0$). *If m is even, this becomes a Newton–Cotes' formula.* If m is odd, it may still be a useful formula, but it does not belong to the Newton–Cotes' family, because $(m - n)/2 = m/2 - k$ is no integer.

If $n = m$, a formula is of the closed type. Its remainder term is the first neglected term of the operator series, truncated after δ^{2k}, $2k = n = m$ (and multiplied by h). Hence the remainder of (5.1.32) can be estimated by $a_{2+m/2}\delta^{m+2} f_0$ or (better)

$$R_{m,m} \sim (a_{m/2+2}/m)H(hD)^{m+2} f_0,$$

where we call $H = mh$ the "bigstep."

If the integral is computed over $[a, b]$ by means of a sequence of bigsteps, each of length H, an estimate of the *global* error has the same form, except that H is replaced by $b - a$ and f_0 is replaced by $\max_{x\in[a,b]} |f(x)|$. The exponent of hD in an error estimate that contains H or $b - a$ is known as the *global order of accuracy* of the method.

If $n < m$, a formula of the open type is obtained. Among the open formulas we shall only consider the case that $n = m - 2$, which is the open Newton–Cotes' formula. The operator expansion is truncated after δ^{m-2}, and we obtain

$$R_{m-2,m} \sim (a_{m/2+1}/m)H(hD)^m f_0.$$

Formulas with $n > m$ are rarely mentioned in the literature (except for $m = 1$). We do not understand why; it is rather common that an integrand has a smooth continuation outside the interval of integration.

We next consider the effect of a linear transformation of the independent variable. Suppose that a formula

$$\sum_{j=1}^{N} a_j f(t_j) - \int_0^1 f(t)\,dt \approx c_N f^{(N)}$$

has been derived for the standard interval [0, 1]. Setting $x = x_k + th$, $dx = hdt$ we find that the corresponding formula and error constant for the interval $[x_k, x_k + h]$ reads

$$\sum_{j=1}^{N} a_j g(x_k + ht_j) - \int_{x_k}^{x_k+h} g(x)\,dx \approx c_N h^{N+1} g^{(N)}(x_k). \tag{5.1.34}$$

This error estimate is valid asymptotically as $h \to 0$. The **local order of accuracy**, i.e., over one step of length h, is $N + 1$; the **global order of accuracy**, i.e., over $(b - a)/h$ steps of length h, becomes N.

For example, the trapezoidal rule is exact for polynomials of degree 1 and hence $N = 2$. For the interval [0, 1], $L(t^2) = \frac{1}{3}$, $\tilde{L}(t^2) = \frac{1}{2}$, and thus $c_2 = \frac{1}{2}(\frac{1}{2} - \frac{1}{3}) = 1/12$. On

an interval of length h the asymptotic error becomes $h^3 g''/12$. The local order of accuracy is $N + 1 = 3$; the global order of accuracy is $N = 2$.

If the "standard interval" is $[-1, 1]$ instead, the transformation becomes $x = \frac{1}{2}ht$, and h is to be replaced by $\frac{1}{2}h$ everywhere in (5.1.34). Be careful about the exact meaning of a remainder term for a formula of this type provided by a table.

We shall illustrate the use of the Cauchy–FFT method for computing the coefficients a_{im} in the expansion (5.1.33). In this way extensive algebraic calculations are avoided.[171] It can be shown that the exact coefficients are rational numbers, though it is sometimes hard to estimate in advance the order of magnitude of the denominators. The algorithm must be used with judgment. Figure 5.1.1 was obtained for $N = 32$, $r = 2$; the absolute errors of the coefficients (see Lemma 3.1.2 about the error estimation) are then less than 10^{-13}. The smoothness of the curves for $j \geq 14$ indicates that the relative accuracy of the values of $a_{m,j}$ are still good there; in fact, other computations show that it is still good when the coefficients are as small as 10^{-20}.

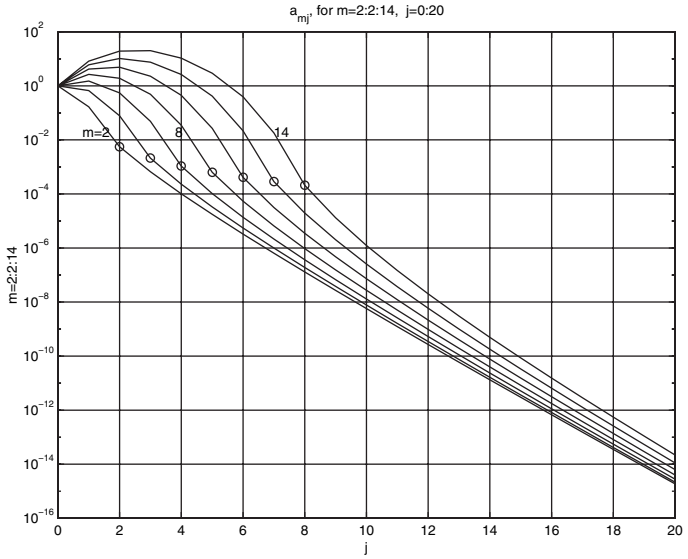

Figure 5.1.1. *The coefficients $|a_{m,j}|$ of the δ^2-expansion for $m = 2 : 2 : 14$, $j = 0 : 20$. The circles are the coefficients for the closed Cotes' formulas, i.e., $j = 1+m/2$.*

The coefficients are first obtained in floating-point representation. The transformation to rational form is obtained by a continued fraction algorithm, described in Example 3.5.3. For the case $m = 8$ the result reads

$$A_8(\delta^2) = 8 + \frac{64}{3}\delta^2 + \frac{688}{45}\delta^4 + \frac{736}{189}\delta^6 + \frac{3956}{14,175}\delta^8 - \frac{2368}{467,775}\delta^{10} + \ldots. \qquad (5.1.35)$$

[171] These could, however, be carried out using a system such as Maple.

The closed integration formula becomes

$$\int_{-x_4}^{x_4} f(x)dx = \frac{4h}{14,175}\Big(-4540f_0 + 10,496(f_1 + f_{-1}) - 928(f_2 + f_{-2}) + 5888(f_3 + f_{-3}) + 989(f_4 + f_{-4})\Big) + R, \tag{5.1.36}$$

$$R \sim \frac{296}{467,775} H h^{10} f^{(10)}(x_0). \tag{5.1.37}$$

It goes without saying that this is not how Newton and Cotes found their methods. Our method may seem complicated, but the MATLAB programs for this are rather short, and to a large extent useful for other purposes. The computation of about 150 Cotes coefficients and 25 remainders ($m = 2 : 14$) took less than two seconds on a PC. This includes the calculation of several alternatives for rational approximations to the floating-point results. For a small number of the 150 coefficients the judicious choice among the alternatives took, however, much more than two (human) seconds; this detail is both science and art.

It was mentioned that *if m is odd*, (5.1.33) does not provide formulas of the Newton–Cotes' family, since $(m - n)/2$ is no integer, nor are the indices j in (5.1.32) integers. Therefore, the operator associated with the right-hand side of (5.1.32) is of the form

$$c_1(E^{1/2} + E^{-1/2}) + c_2(E^{3/2} + E^{-3/2}) + c_3(E^{5/2} + E^{-5/2}) + \cdots.$$

If it is divided algebraically by $\mu = 1/2(E^{1/2} + E^{-1/2})$, however, it becomes the $B(E)$-form (say)

$$b_1' + b_2'(E + E^{-1}) + b_3'(E^2 + E^{-2}) + \cdots + b_k(E^k + E^{-k}).$$

If m is odd, we therefore expand

$$(e^{hDm/2} - e^{-hDm/2})(hD)^{-1}/\mu, \quad \mu = \sqrt{1 + \delta^2/4},$$

into a δ^2-series, with coefficients a_j'. Again, this can be done numerically by the Cauchy–FFT method. For each m, two truncated δ^2-series (one for the closed and one for the open case) are then transformed into $B(E)$-expressions numerically by means of the matrix M, as described above. The expressions are then multiplied *algebraically* by $\mu = (1/2)(E^{1/2} + E^{-1/2})$. We then have the coefficients of a Newton–Cotes' formula with m odd.

The asymptotic error is

$$a_{m/2+1}' H(hD)^{m+1} \quad \text{and} \quad a_{m/2-1}' H(hD)^{m-1}$$

for the closed type and open type, respectively ($2k = m - 1$). The global orders of accuracy for Newton–Cotes' methods with odd m are thus the same as for the methods where m is one less.

5.1.6 Fejér and Clenshaw–Curtis Rules

Equally spaced interpolation points as used in the Newton–Cotes' formulas are useful for low degrees but can diverge as fast as 2^n as $n \to \infty$. Quadrature rules which use a set of points which cluster near the endpoints of the interval have better properties for large n.

Fejér [115] suggested using the zeros of a Chebyshev polynomial of first or second kind as interpolation points for quadrature rules of the form

$$\int_{-1}^{1} f(x)\,dx = \sum_{k=0}^{n} w_k f(x_k). \tag{5.1.38}$$

Fejér's first rule uses the zeros of the Chebyshev polynomial $T_n(x) = \cos(n \arccos x)$ of the first kind in $(-1, 1)$, which are

$$x_k = \cos\theta_k, \quad \theta_k = \frac{(2k-1)\pi}{2n}, \quad k = 1:n. \tag{5.1.39}$$

The following explicit formula for the weights is known (see [91]):

$$w_k^{f1} = \frac{2}{n}\left(1 - 2\sum_{j=1}^{\lfloor n/2 \rfloor} \frac{\cos(2j\theta_k)}{4j^2-1}\right), \quad k = 1:n. \tag{5.1.40}$$

Fejér's second rule uses the zeros of the Chebyshev polynomial $U_{n-1}(x)$ of the second kind, which are the extreme points of $T_n(x)$ in $(-1, 1)$ (see Sec. 3.2.3):

$$x_k = \cos\theta_k, \quad \theta_k = \frac{k\pi}{n}, \quad k = 1:n-1. \tag{5.1.41}$$

An explicit formula for the weights is (see [91])

$$w_k^{f2} = \frac{4\sin\theta_k}{n} \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{\sin(2j-1)\theta_k}{2j-1}, \quad k = 1:n-1. \tag{5.1.42}$$

Both Fejér's rules are open quadrature rules, i.e., they do not use the endpoints of the interval $[-1, 1]$. Fejér's second rule is the more practical, because going from $n+1$ to $2n+1$ points, only n new function values need to be evaluated; cf. the trapezoidal rule.

The quadrature rule of Clenshaw and Curtis [71] is a closed version of Fejér's second rule; i.e., the nodes are the $n+1$ extreme points of $T_n(x)$, in $[-1, 1]$, including the endpoints $x_0 = 1$, $x_n = -1$. An explicit formula for the Clenshaw–Curtis weights is

$$w_k^{cc} = \frac{c_k}{n}\left(1 - \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{b_j}{4j^2-1}\cos(2j\theta_k)\right), \quad k = 0:n, \tag{5.1.43}$$

where

$$b_j = \begin{cases} 1 & \text{if } j = n/2, \\ 2 & \text{if } j < n/2, \end{cases} \qquad c_k = \begin{cases} 1 & \text{if } k = 0, n, \\ 2 & \text{otherwise.} \end{cases} \tag{5.1.44}$$

In particular the weights at the two boundary points are

$$w_0^{cc} = w_n^{cc} = \frac{1}{n^2 - 1 + \operatorname{mod}(n, 2)}. \tag{5.1.45}$$

For both the Fejér and Clenshaw–Curtis rules the weights can be shown to be positive; see Imhof [203]. Therefore the convergence of $I_n(f)$ as $n \to \infty$ for all $f \in C[-1, 1]$ is

assured for these rules by the following theorem, which is a consequence of Weierstrass' theorem.

Theorem 5.1.6.

Let x_{nj} and a_{nj}, where $j = 1 : n$, $n = 1, 2, 3, \ldots$, be triangular arrays of nodes and weights, respectively, and suppose that $a_{nj} > 0$ for all n, $j \geq 1$. Consider the sequence of quadrature rules

$$I_n f = \sum_{j=1}^{n} a_{nj} f(x_{nj})$$

for the integral $If = \int_a^b f(x)w(x)\,dx$, where $[a, b]$ is a closed, bounded interval, and $w(x)$ is an integrable weight function. Suppose that $I_n p = Ip$ for all $p \in \mathcal{P}_{k_n}$, where $\{k_n\}_{n=1}^{\infty}$ is a strictly increasing sequence. Then

$$I_n f \to If \quad \forall f \in C[a, b].$$

Note that this theorem is not applicable to Cotes' formulas, where some weights are negative.

Convergence will be fast for the Fejér and Clenshaw–Curtis rules provided the integrand is k times continuously differentiable—a property that the user can often check. However, if the integrand is discontinuous, the interval of integration should be partitioned at the discontinuities and the subintervals treated separately.

Despite its advantages these quadrature rules did not receive much use early on, because computing the weights using the explicit formulas given above is costly ($O(n^2)$ flops) and not numerically stable for large values of n. However, it is not necessary to compute the weights explicitly. Gentleman [151, 150] showed how the Clenshaw–Curtis rule can be implemented by means of a discrete cosine transform (DCT; see Sec. 4.7.4). We recall that the FFT is not only fast, but also very resistant to roundoff errors.

The interpolation polynomial $L_n(x)$ can be represented in terms of Chebyshev polynomials

$$L_n(x) = \sum_{k=0}^{n}{}'' c_k T_k(x), \quad c_k = \frac{2}{n} \sum_{j=0}^{n}{}'' f(x_j) \cos\left(\frac{kj\pi}{n}\right),$$

where $x_j = \cos(j\pi/n)$. This is the real part of an FFT. (The double prime on the sum means that the first and last terms are to be halved.) Then we have

$$I_n(f) = \int_{-1}^{1} L_n(x)\,dx = \sum_{k=0}^{n}{}'' c_k \mu_k, \quad \mu_k = \int_{-1}^{1} T_k(x)\,dx,$$

where μ_k are the moments of the Chebyshev polynomials. It can be shown (Problem 5.1.7) that

$$\mu_k = \int_{-1}^{1} T_k(x)\,dx = \begin{cases} 0 & \text{if } k \text{ odd}, \\ 2/(1-k^2) & \text{if } k \text{ even}. \end{cases}$$

The following MATLAB program, due to Trefethen [352], is a compact implementation of this version of the Clenshaw–Curtis quadrature.

ALGORITHM 5.1. *Clenshaw–Curtis Quadrature.*

```
function I = clenshaw_curtis(f,n);
% Computes the integral I of f over [-1,1] by the
% Clenshaw-Curtis quadrature rule with n+1 nodes.
x =  cos(pi*(0:n)'/n);
%Chebyshev extreme points
fx = feval(f,x)/(2*n);
%Fast Fourier transform
g = real(fft(fx([1:n+1 n:-1:2])));
%Chebyshev coefficients
a = [g(1); g(2:n)+g(2*n:-1:n+2); g(n+1)];
w = 0*a'; w(1:2:end) = 2./(1-(0:2:n).^2);
I = w*a;
```

A fast and accurate algorithm for computing the weights in the Fejér and Clenshaw–Curtis rules in $O(n \log n)$ flops has been given by Waldvogel [361]. The weights are obtained as the inverse FFT of certain vectors given by explicit rational expressions. On an average laptop this takes just about five seconds for $n = 2^{20} + 1 = 1{,}048{,}577$ nodes!

For Fejér's second rule the weights are the inverse discrete FFT of the vector v with components v_k given by the expressions

$$\begin{aligned} v_k &= \frac{2}{1-4k^2}, \quad k = 0 : \lfloor n/2 \rfloor - 1, \\ v_{\lfloor n/2 \rfloor} &= \frac{n-3}{2\lfloor n/2 \rfloor - 1} - 1, \\ v_{n-k} &= v_k, \quad k = 1 : \lfloor (n-1)/2 \rfloor. \end{aligned} \tag{5.1.46}$$

(Note that this will give zero weights for $k = 0, n$ corresponding to the endpoint nodes $x_0 = -1$ and $x_n = 1$.)

For the Clenshaw–Curtis rule the weights are the inverse FFT of the vector $v + g$, where

$$\begin{aligned} g_k &= -w_0^{cc}, \quad k = 0 : \lfloor n/2 \rfloor - 1, \\ g_{\lfloor n/2 \rfloor} &= w_0 \left[(2 - \bmod(n, 2))\, n - 1 \right], \\ g_{n-k} &= g_k, \quad k = 1 : \lfloor (n-1)/2 \rfloor, \end{aligned} \tag{5.1.47}$$

and w_0^{cc} is given by (5.1.45). For the weights Fejér's first rule and MATLAB files implementing the algorithm, we refer to [361].

Since the complexity of the inverse FFT is $O(n \log n)$, this approach allows fast and accurate calculation of the weights for rules of high order, in particular when n is a power of 2. For example, using the MATLAB routine `IFFT` the weights for $n = 1024$ only takes a few milliseconds on a PC.

Review Questions

5.1.1 Name three classical quadrature methods and give their order of accuracy.

5.1.2 What is meant by a composite quadrature rule? What is the difference between local and global error?

5.1.3 What is the advantage of including a weight function $w(x) > 0$ in some quadrature rules?

5.1.4 Describe some possibilities for treating integrals where the integrand has a **singularity** or is "almost singular."

5.1.5 For some classes of functions the composite trapezoidal rule exhibits so-called *superconvergence*. What is meant by this term? Give an example of a class of functions for which this is true.

5.1.6 Give an account of the theoretical background of the classical Newton–Cotes' rules.

Problems and Computer Exercises

5.1.1 (a) Derive the closed Newton–Cotes' rule for $m = 3$,

$$I = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) + R_T, \quad h = \frac{(b-a)}{3},$$

also known as Simpson's (3/8)-rule.

(b) Derive the open Newton–Cotes' rule for $m = 4$,

$$I = \frac{4h}{3}(2f_1 - f_2 + 2f_3) + R_T, \qquad h = \frac{(b-a)}{4}.$$

(c) Find asymptotic error estimates for the formulas in (a) and (b) by applying them to suitable polynomials.

5.1.2 (a) Show that Simpson's rule is the unique quadrature formula of the form

$$\int_{-h}^{h} f(x)\,dx \approx h(a_{-1}f(-h) + a_0 f(0) + a_1 f(h))$$

that is exact whenever $f \in \mathcal{P}_4$. Try to find several derivations of Simpson's rule, with or without the use of difference operators.

(b) Find the Peano kernel $K_2(u)$ such that $Rf = \int_{\mathbf{R}} f''(u)K_2(u)\,du$, and find the best constants c, p such that

$$|Rf| \le ch^p \max|f''(u)| \quad \forall f \in C^2[-h, h].$$

If you are going to deal with functions that are not in C^3, would you still prefer Simpson's rule to the trapezoidal rule?

5.1.3 The quadrature formula

$$\int_{x_{i-1}}^{x_{i+1}} f(x)\,dx \approx h\big(af(x_{i-1}) + bf(x_i) + cf(x_{i+1})\big) + h^2 df'(x_i)$$

can be interpreted as a Hermite interpolatory formula with a *double point* at x_i. Show that $d = 0$ and that this formula is identical to Simpson's rule. Then show that the error can be written as

$$R(f) = \frac{1}{4!}\int_{x_{i-1}}^{x_{i+1}} f^{(4)}(\xi_x)(x - x_{i-1})(x - x_i)^2(x - x_{i+1})\,dx,$$

where $f^{(4)}(\xi_x)$ is a continuous function of x. Deduce the error formula for Simpson's rule. Setting $x = x_i + ht$, we get

$$R(f) = \frac{h^4}{24} f^{(4)}(\xi_i)\int_{-1}^{1} (t+1)t^2(t-1)h\,dt = \frac{h^5}{90} f^{(4)}(\xi_i).$$

5.1.4 A second kind of Newton–Cotes' open quadrature rule uses the midpoints of the equidistant grid $x_i = ih$, $i = 1 : n$, i.e.,

$$\int_{x_0}^{x_n} f(x)\,dx = \sum_{i=1}^{n} w_i f_{i-1/2}, \quad x_{i-1/2} = \frac{1}{2}(x_{i-1} + x_i).$$

(a) For $n = 1$ we get the midpoint rule. Determine the weights in this formula for $n = 3$ and $n = 5$. (Use symmetry!)

(b) What is the order of accuracy of these two rules?

5.1.5 (a) Simpson's rule with end corrections is a quadrature formula of the form

$$\int_{-h}^{h} f(x)\,dx \approx h\big(\alpha f(-h) + \beta f(0) + \alpha f(h)\big) + h^2\gamma(f'(-h) - f'(h)),$$

which is exact for polynomials of degree five. Determine the weights α, β, γ by using the test functions $f(x) = 1, x^2, x^4$. Use $f(x) = x^6$ to determine the error term.

(b) Show that in the corresponding composite formula for the interval $[a, b]$ with $b - a = 2nh$, only the endpoint derivatives $f'(a)$ and $f'(b)$ are needed.

5.1.6 (Lyness) Consider the integral

$$I(f, g) = \int_0^{nh} f(x)g'(x)\,dx. \tag{5.1.48}$$

An approximation related to the trapezoidal rule is

$$S_m = \frac{1}{2}\sum_{j=0}^{n-1}\big[f(jh) + f((j+1)h)\big]\big[g((j+1)h) - (g(jh)\big],$$

which requires $2(m+1)$ function evaluations. Similarly, an analogue to the midpoint rule is

$$R_m = \frac{1}{2}\sum_{j=0}^{n-1}{}'' f(jh)\big[g((j+1)h) - (g((j-1)h)\big],$$

where the double prime on the summation indicates that the extreme values $j = 0$ and $j = m$ are assigned a weighting factor $\frac{1}{2}$. This rule requires $2(m+2)$ function evaluations, two of which lie outside the interval of integration. Show that the difference $S_m - R_m$ is of order $O(h^2)$.

5.1.7 Show the relations

$$\int_{-1}^{x} T_n(t)\,dt = \begin{cases} \dfrac{T_{n+1}(x)}{2(n+1)} - \dfrac{T_{n-1}(x)}{2(n-1)} + \dfrac{(-1)^{n+1}}{n^2-1} & \text{if } n \geq 2, \\ \dfrac{(T_2(x)-1)}{4} & \text{if } n = 1, \\ T_1(x) + 1 & \text{if } n = 0. \end{cases}$$

Then deduce that

$$\int_{-1}^{1} T_n(x)\,dx = \begin{cases} 0 & \text{if } n \text{ odd}, \\ 2/(1-n^2) & \text{if } n \text{ even}. \end{cases}$$

Hint: Make a change of variable in the integral and use the trigonometric identity $2\cos n\phi \sin\phi = \sin(n+1)\phi - \sin(n-1)\phi$.

5.1.8 Compute the integral $\int_0^\infty (1+x^2)^{-4/3}\,dx$ with five correct decimals. Expand the integrand in powers of x^{-1} and integrate termwise over the interval $[R, \infty]$ for a suitable value of R. Then use a Newton–Cotes' rule on the remaining interval $[0, R]$.

5.1.9 Write a program for the derivation of a formula for integrals of the form $I = \int_0^1 x^{-1/2} f(x)\,dx$ that is exact for $f \in \mathcal{P}_n$ and uses the values $f(x_i)$, $i = 1:n$, by means of the power basis.

(a) Compute the coefficients b_i for $n = 6:8$ with equidistant points, $x_i = (i-1)/(n-1)$, $i = 1:n$. Apply the formulas to the integrals

$$\int_0^1 x^{-1/2}e^{-x}\,dx, \qquad \int_0^1 \frac{dx}{\sin\sqrt{x}}, \qquad \int_0^1 (1-t^3)^{-1/2}\,dt.$$

In the first of the integrals compare with the result obtained by series expansion in Problem 3.1.1. A substitution is needed for bringing the last integral to the right form.

(b) Do the same for the case where the step size $x_{i+1} - x_i$ grows proportionally to i; $x_1 = 0$; $x_n = 1$. Is the accuracy significantly different compared to (a), for the same number of points?

(c) Make some very small random perturbations of the x_i, $i = 1:n$ in (a), (say) of the order of 10^{-13}. Of which order of magnitude are the changes in the coefficients b_i, and the changes in the results for the first of the integrals?

5.1.10 Propose a suitable plan (using a computer) for computing the following integrals, for $s = 0.5, 0.6, 0.7, \ldots, 3.0$.

(a) $\int_0^\infty (x^3 + sx)^{-1/2}\,dx$; (b) $\int_0^\infty (x^2+1)^{-1/2} e^{-sx}\,dx$, error $< 10^{-6}$;

(c) $\int_\pi^\infty (s+x)^{-1/3} \sin x\,dx$.

5.1.11 It is not true that any degree of accuracy can be obtained by using a Newton–Cotes' formula of sufficiently high order. To show this, compute approximations to the integral

$$\int_{-4}^{4} \frac{dx}{1+x^2} = 2\tan^{-1} 4 \approx 2.6516353\ldots$$

using the closed Newton–Cotes' formula with $n = 2, 4, 6, 8$. Which formula gives the smallest error?

5.1.12 For expressing integrals appearing in the solution of certain integral equations, the following modification of the midpoint rule is often used:

$$\int_{x_0}^{x_n} K(x_j, x) y(x)\,dx = \sum_{i=0}^{n-1} m_{ij} y_{i+1/2},$$

where $y_{i+1/2} = y(\frac{1}{2}(x_i + x_{i+1}))$ and m_{ij} is the moment integral

$$m_{ij} = \int_{x_i}^{x_{i+1}} K(x_j, x)\,dx.$$

Derive an error estimate for this formula.

5.1.13 (a) Suppose that you have found a truncated δ^2-expansion, (say) $A(\delta^2) \equiv a_1 + a_2\delta^2 + \cdots + a_{k+1}\delta^{2k}$. Then an equivalent symmetric expression of the form $B(E) \equiv b_1 + b_2(E + E^{-1}) + \cdots + b_{k+1}(E^k + E^{-k})$ can be obtained as $b = M_{k+1}a$, where a, b are column vectors for the coefficients, and M_{k+1} is the $(k+1) \times (k+1)$ submatrix of the matrix M given in (3.3.49).

Use this for deriving (5.1.36) from (5.1.35). How do you obtain the remainder term? If you obtain the coefficients as decimal fractions, multiply them by $14,175/4$ in order to check that they agree with (5.1.36).

(b) Use Cauchy–FFT for deriving (5.1.35), and the open formula and the remainder for the same interval.

(c) Set $z_n = \nabla^{-1} y_n - \Delta^{-1} y_0$. We have, in the literature, seen the interpretation that $z_n = \sum_{j=0}^{n} y_j$ if $n \geq 0$. It seems to require some extra conditions to be true. Investigate if the conditions $z_{-1} = y_{-1} = 0$ are necessary and sufficient. Can you suggest better conditions? (The equations $\Delta\Delta^{-1} = \nabla\nabla^{-1} = 1$ mentioned earlier are assumed to be true.)

5.1.14 (a) Write a program for the derivation of quadrature formulas and error estimates using the Cauchy–FFT method in Sec. 5.1.5 for $m = n - 1, n, n + 1$. Test the formulas and the error estimates for some m, n on some simple (though not too simple) examples. Some of these formulas are listed in the Handbook [1, Sec. 25.4].

In particular, check the closed Newton–Cotes' 9-point formula ($n = 8$).

(b) Sketch a program for the case that $h = 1/(2n + 1)$, with the computation of f at $2m$ symmetrical points.

(c) [1, Sec. 25.4] gives several Newton–Cotes' formulas of closed and open types, with remainders. Try to reproduce and extend their tables with techniques related to Sec. 5.3.1.

5.1.15 Compute the integral

$$\frac{1}{2\pi}\int_0^{2\pi} e^{\frac{1}{\sqrt{2}}\sin x}\,dx$$

by the trapezoidal rule, using $h = \pi/2^k$ $k = 0, 1, 2, \ldots$, until the error is on the level of the roundoff errors. Observe how the number of correct digits vary with h. Notice that Romberg is of no use in this problem.

Hint: First estimate how well the function $g(x) = e^{x/\sqrt{2}}$ can be approximated by a polynomial in $\mathcal{P}_8$ for $x \in [-1, 1]$. The estimate found by the truncated Maclaurin expansion is not quite good enough. Theorem 3.1.5 provides a sharper estimate with an appropriate choice of R; remember Scylla and Charybdis.

5.1.16 (a) Show that the trapezoidal rule, with $h = 2\pi/(n+1)$, is exact for all trigonometric polynomials of period 2π, i.e., for functions of the type

$$\sum_{k=-n}^{n} c_k e^{ikt}, \quad i^2 = -1,$$

when it is used for integration over a whole period.

(b) Show that if $f(x)$ can be approximated by a trigonometric polynomial of degree n so that the magnitude of the error is less than ϵ, in the interval $(0, 2\pi)$, then the error with the use of the trapezoidal rule with $h = 2\pi/(n + 1)$ on the integral

$$\frac{1}{2\pi}\int_0^{2\pi} f(x)\,dx$$

is less than 2ϵ.

(c) Use the above to explain the sensationally good result in Problem 5.1.15 above, when $h = \pi/4$.

5.2 Integration by Extrapolation

5.2.1 The Euler–Maclaurin Formula

Newton–Cotes' rules have the drawback that they do not provide a convenient way of estimating the error. Also, for high-order rules negative weights appear. In this section we will derive formulas of high order, based on the Euler–Maclaurin formula (see Sec. 3.4.5), which do not share these drawbacks.

Let $x_i = a + ih$, $x_n = b$, and let $T(a : h : b)f$ denote the trapezoidal sum

$$T(a : h : b)f = \sum_{i=1}^{n} \frac{h}{2}\big(f(x_{i-1}) + f(x_i)\big). \tag{5.2.1}$$

According to Theorem 3.4.10, if $f \in C^{2r+2}[a, b]$, then

$$T(a : h : b)f - \int_a^b f(x)\,dx = \frac{h^2}{12}\big(f'(b) - f'(a)\big) - \frac{h^4}{720}\big(f'''(b) - f'''(a)\big) + \cdots + \frac{B_{2r}h^{2r}}{(2r)!}\big(f^{(2r-1)}(b) - f^{(2r-1)}(a)\big) + R_{2r+2}(a, h, b)f.$$

By (3.4.37) the remainder $R_{2r+2}(a, h, b)f$ is $O(h^{2r+2})$ and represented by an integral with a kernel of constant sign in $[a, b]$. The estimation of the remainder is very simple in certain important particular cases. Note that although the expansion contains derivatives at the boundary points only, the remainder requires that $|f^{(2r+2)}|$ is integrable on the whole interval $[a, b]$.

We recall the following simple and useful relation between the trapezoidal sum and the midpoint sum (cf. (5.1.21)):

$$M(a : h : b)f = \sum_{i=1}^{n} hf(x_{i-1/2}) = 2T\left(a : \frac{1}{2}h : b\right)f - T(a : h : b)f. \tag{5.2.2}$$

From this one easily derives the expansion

$$M(a : h : b)f = \int_a^b f(x)\,dx - \frac{h^2}{24}\big(f'(b) - f'(a)\big) + \frac{7h^4}{5760}\big(f'''(b) - f'''(a)\big) + \cdots + \Big(\frac{1}{2^{2r-1}} - 1\Big)\frac{B_{2r}h^{2r}}{(2r)!}\big(f^{(2r-1)}(b) - f^{(2r-1)}(a)\big) + \cdots,$$

which has the same relation to the midpoint sum as the Euler–Maclaurin formula has to the trapezoidal sum.

The Euler–Maclaurin formula can be used for highly accurate numerical integration when the values of derivatives of f are known at $x = a$ and $x = b$. It is also possible to use difference approximations to estimate the derivatives needed. A variant with uncentered differences is **Gregory's**[172] **quadrature formula**:

$$\int_a^b f(x)\,dx = h\frac{E^n - 1}{hD}f_0 = h\left(\frac{f_n}{-\ln(1 - \nabla)} - \frac{f_0}{\ln(1 + \Delta)}\right) = T(a; h; b) + h\sum_{j=1}^{\infty} a_{j+1}(\nabla^j f_n + (-\Delta)^j f_0),$$

[172] James Gregory (1638–1675), a Scotch mathematician, discovered this formula long before the Euler–Maclaurin formula. It seems to have been used primarily for numerical quadrature. It can be used also for summation, but the variants with central differences are typically more efficient.

where $T(a : h : b)$ is the trapezoidal sum. The operator expansion must be truncated at $\nabla^k f_n$ and $\Delta^l f_0$, where $k \le n$, $l \le n$. (Explain why the coefficients a_{j+1}, $j \ge 1$, occur in the implicit Adams formula too; see Problem 3.3.10 (a).)

5.2.2 Romberg's Method

The Euler–Maclaurin formula is the theoretical basis for the application of repeated Richardson extrapolation (see Sec. 3.4.6) to the results of the trapezoidal rule. This method is known as **Romberg's method**.[173] It is one of the most widely used methods, because it allows a simple strategy for the automatic determination of a suitable step size and order. Romberg's method was made widely known through Stiefel [329]. A thorough analysis of the method was carried out by Bauer, Rutishauser, and Stiefel in [20], which we shall refer to for proof details.

Let $f \in C^{2m+2}[a, b]$ be a real function to be integrated over $[a, b]$ and denote the trapezoidal sum by $T(h) = T(a : h : b) f$. By the Euler–Maclaurin formula it follows that

$$T(h) - \int_a^b f(x)\,dx = c_2 h^2 + c_4 h^4 + \cdots + c_m h^{2m} + \tau_{m+1}(h) h^{2m+2}, \tag{5.2.3}$$

where $c_k = 0$ if $f \in \mathcal{P}_k$. This suggests the use of repeated Richardson extrapolation applied to the trapezoidal sums computed with step lengths

$$h_1 = \frac{b-a}{n_1}, \quad h_2 = \frac{h_1}{n_1}, \quad \ldots, \quad h_q = \frac{h_1}{n_q}, \tag{5.2.4}$$

where $n_1, n_2, \ldots, n_q$ are strictly increasing positive integers. If we set $T_{m,1} = T(a, h_m, b) f$, $m = 1 : q$, then using Neville's interpolation scheme the extrapolated values can be computed from the recursion:

$$T_{m,k+1} = T_{m,k} + \frac{T_{m,k} - T_{m-1,k}}{(h_{m-k}/h_m)^2 - 1}, \quad 1 \le k < m. \tag{5.2.5}$$

Romberg used step sizes in a geometric progression, $h_m/h_{m-1} = q = 2$. In this case the denominators in (5.2.5) become $2^{2k} - 1$. This choice has the advantage that successive trapezoidal sums can be computed using the relation

$$T\left(\frac{h}{2}\right) = \frac{1}{2}(T(h) + M(h)), \quad M(h) = \sum_{i=1}^{n} h f(x_{i-1/2}), \tag{5.2.6}$$

where $M(h)$ is the midpoint sum. This makes it possible to reuse the function values that have been computed earlier.

We remark that, usually, a composite form of Romberg's method is used, the method is applied to a sequence interval $[a + iH, a + (i+1)H]$ for some bigstep H. The applications

[173] Werner Romberg (1909–2003) was a German mathematician. For political reasons he fled Germany in 1937, first to Ukraine and then to Norway, where in 1938 he joined the University of Oslo. He spent the war years in Sweden and then returned to Norway. In 1949 he joined the Norwegian Institute of Technology in Trondheim. He was called back to Germany in 1968 to take up a position at the University of Heidelberg.

of repeated Richardson extrapolation and the Neville algorithms to differential equations belong to the most important.

Rational extrapolation can also be used. This gives rise to a recursion of a form similar to (5.2.5):

$$T_{m,k+1} = T_{m,k} + \frac{T_{m,k} - T_{m-1,k}}{\left(\frac{h_{m-k}}{h_m}\right)^2 \left[1 - \frac{T_{m,k} - T_{m-1,k}}{T_{m,k} - T_{m-1,k-1}}\right] - 1}, \quad 1 \le k \le m; \tag{5.2.7}$$

see Sec. 4.3.3.

For practical numerical calculations the values of the coefficients c_k in (5.2.3) are not needed, but they are used, for example, in the derivation of an error bound; see Theorem 5.2.1. It is also important to remember that the coefficients depend on derivatives of increasing order; the success of repeated Richardson extrapolations is thus related to the behavior in $[a, b]$ of the higher derivatives of the integrand.

Theorem 5.2.1 (*Error Bound for Romberg's Method*).

The items $T_{m,k}$ in Romberg's method are estimates of the integral $\int_a^b f(x)\,dx$ that can be expressed as a linear functional,

$$T_{m,k} = (b-a)\sum_{j=0}^{n} \alpha_{m,j}^{(k)} f(a + jh), \tag{5.2.8}$$

where $n = 2^{m-1}$, $h = (b-a)/n$, and

$$\sum_{j=0}^{n} \alpha_{m,j}^{(k)} = 1, \quad \alpha_{m,j}^{(k)} > 0. \tag{5.2.9}$$

The remainder functional for $T_{m,k}$ is zero for $f \in \mathcal{P}_{2k}$, and its Peano kernel is positive in the interval (a, b). The truncation error of $T_{m,k}$ reads

$$\begin{aligned} T_{m,k} - \int_a^b f(x)dx &= r_k h^{2k}(b-a) f^{(2k)}\left(\frac{1}{2}(a+b)\right) + O(h^{2k+2}(b-a) f^{(2k+2)}) \\ &= r_k h^{2k}(b-a) f^{(2k)}(\xi), \quad \xi \in (a, b), \end{aligned} \tag{5.2.10}$$

where

$$r_k = 2^{k(k-1)}|B_{2k}|/(2k)!, \quad h = 2^{1-m}(b-a).$$

Proof. Sketch: Equation (5.2.8) follows directly from the construction of the Romberg scheme. (It is for theoretical use only; the recursion formulas are better for practical use.) The first formula in (5.2.9) holds because $T_{m,k}$ is exact if $f = 1$. The second formula is easily proved for low values of k. The general proof is more complicated; see [20, Theorem 4].

The Peano kernel for $m = k = 1$ (the trapezoidal rule) was constructed in Example 3.3.7. For $m = k = 2$ (Simpson's rule), see Sec. 5.1.3. The general case is more complicated. Recall that, by Corollary 3.3.9 of Peano's remainder theorem, a remainder formula with a mean value $\xi \in (a, b)$ exists if and only if the Peano kernel does not change sign.

Bauer, Rutishauser, and Stiefel [20, pp. 207–210] constructed a recursion formula for the kernels, and succeeded in proving that they are all positive, by an ingenious use of the recursion. The expression for r_k is also derived there, although with a different notation. □

From (5.2.9) it follows that if the magnitude of the irregular error in $f(a + jh)$ is at most ϵ, then the magnitude of the inherited irregular error in $T_{m,k}$ is at most $\epsilon(b-a)$. There is another way of finding r_k. Note that for each value of k, the error of $T_{k,k}$ for $f(x) = x^{2k}$ can be determined numerically. Then r_k can be obtained from (5.2.10). $T_{m,k}$ is the same formula as $T_{k,k}$, although with a different h.

According to the discussion of repeated Richardson extrapolation in Sec. 3.4.6, one continues the process until two values *in the same row* agree to the desired accuracy. If no other error estimate is available, $\min_k |T_{m,k} - T_{m,k-1}|$ is usually chosen as an estimate of the truncation error, even though it is usually a strong overestimate. A feature of the Romberg algorithm is that it also contains exits with lower accuracy at a lower cost.

Example 5.2.1 (*A Numerical Illustration to Romberg's Method*).

Use Romberg's method to compute the integral (cf. Example 5.1.5)

$$\int_0^{0.8} \frac{\sin x}{x}\, dx.$$

The midpoint and trapezoidal sums are with ten correct decimals equal to

h	$M(h)f$	$T(h)f$
0.8	0.77883 66846	0.75867 80454
0.4	0.77376 69771	0.76875 73650
0.2	0.77251 27161	0.77126 21711
0.1		0.77188 74436

.

It can be verified that in this example the error is approximately proportional to h^2 for both $M(h)$ and $T(h)$. We estimate the error in $T(0.1)$ to be $\frac{1}{3}6.26 \cdot 10^{-4} \leq 2.1 \cdot 10^{-4}$.

The trapezoidal sums are then copied to the first column of the Romberg scheme. Repeated Richardson extrapolation is performed giving the following table.

m	T_{m1}	T_{m2}	T_{m3}	T_{m4}
1	0.75867 80454			
2	0.76875 73650	0.77211 71382		
3	0.77126 21711	0.77209 71065	0.77209 57710	
4	0.77188 74437	0.77209 58678	0.77209 57853	0.77209 57855
5	0.77204 37039	0.77209 57906	0.77209 57855	0.77209 57855

.

We find that $|T_{44} - T_{43}| = 2 \cdot 10^{-10}$, and the irregular errors are less than 10^{-10}. Indeed, all ten digits in T_{44} are correct, and $I = 0.77209\,57854\,82\ldots$. Note that the rate of convergence in successive columns is $h^2, h^4, h^6, h8, \ldots$.

The following MATLAB program implements Romberg's method. In each major step a new row in the Romberg table is computed.

ALGORITHM 5.2. *Romberg's Method.*

```
function [I, md, T] = romberg(f,a,b,tol,q);
% Romberg's method for computing the integral of f over [a,b]
% using at most q extrapolations. Stop when two adjacent values
% in the same column differ by less than tol or when q
% extrapolations have been performed. Output is an estimate
% I of the integral with error bound md and the active part
% of the Romberg table.
%
T = zeros(q+2,q+1);
h = b - a;  m = 1; P = 1;
T(1,1) = h*(feval(f,a) + feval(f,b))/2;
for m = 2:q+1
  h = h/2; m = 2*m;
  M = 0;              % Compute midpoint sum
  for k = 1:2:m
    M = M + feval(f, a+k*h);
  end
  T(m,1) = T(m-1,1)/2 + h*M;
  kmax = min(m-1,q);
  for k = 1:kmax      % Repeated Richardson extrapolation
    T(m,k+1) = T(m,k) + (T(m,k) - T(m-1,k))/(2^(2*k) - 1);
  end
  [md, kb] = min(abs(T(m,1:kmax) - T(m-1,1:kmax)));
  I = T(m,kb);
  if md <= tol           % Check accuracy
    T = T(1:m,1:kmax+1); % Active part of T
  return
  end
end
```

In the above algorithm the value $T_{m,k}$ is accepted when $|T_{m,k} - T_{m-1,k}| \leq tol$, where tol is the permissible error. Thus one extrapolates until two values *in the same column* agree to the desired accuracy. In most situations, this gives, if h is sufficiently small, with a large margin a bound for the truncation error in the lower of the two values. Often instead the subdiagonal error criterion $|T_{m,m-1} - T_{m,m}| < \delta$ is used, and T_{mm} taken as the numerical result.

If the use of the basic asymptotic expansion is doubtful, then the uppermost diagonal of the extrapolation scheme should be ignored. Such a case can be detected by inspection of the difference quotients in a column. If for some k, where $T_{k+2,k}$ has been computed and the modulus of the relative irregular error of $T_{k+2,k} - T_{k+1,k}$ is less than (say) 20%, *and*, most important, the difference quotient

$$(T_{k+1,k} - T_{k,k})/(T_{k+2,k} - T_{k+1,k})$$

is very different from its theoretical value q^{p_k}, then the uppermost diagonal is to be ignored (except for its first element).

Sometimes several of the uppermost diagonals are to be ignored. For the integration of a class of periodic functions the trapezoidal rule is superconvergent; see Sec. 5.1.4. In this case all the difference quotients in the first column are much larger than $q^{p_1} = q^2$. According to the rule just formulated, every element of the Romberg scheme outside the first column should be ignored. This is correct; *in superconvergent cases Romberg's method is of no use*; it destroys the excellent results that the trapezoidal rule has produced.

Example 5.2.2.

The remainder for $T_{k,k}$ in Romberg's method reads

$$T_{k,k} - \int_a^b f(x)\,dx = r_k h^{2^k}(b-a) f^{2^k}(\xi).$$

For $k = 1$, T_{11} is the trapezoidal rule with remainder $r_1 h^2 (b-a) f^{(2)}(\xi)$. By working algebraically in the Romberg scheme, we see that T_{22} is the same as Simpson's rule. It can also be shown that T_{33} is the same as Milne's formula, i.e., the five-point closed Newton–Cotes' formula. It follows that for $k = \{1, 2, 3\}$ both methods give, with $k' = \{2, 3, 5\}$ function values, exact results for $f \in \mathcal{P}_{k'}$.

This equivalence can also be proved by the following argument. By Corollary 3.3.8, there is only one linear combination of the values of the function f at $n+1$ given points that can yield $\int_a^b f(x)\,dx$ exactly for all polynomials $f \in \mathcal{P}_{n+1}$. It follows that the methods of Cotes and Romberg for T_{kk} are identical for $k = 1, 2, 3$.

For $k > 3$ the methods are not identical. For $k = 4$ (9 function values), Cotes is exact in $\mathcal{P}_{10}$, while T_{44} is exact in $\mathcal{P}_8$. For $k = 5$ (17 function values), Cotes is exact in $\mathcal{P}_{18}$, while T_{55} is exact in $\mathcal{P}_{10}$. This sounds like an advantage for Cotes, but one has to be sceptical about formulas that use equidistant points in polynomial approximation of very high degree; see the discussion of Runge's phenomena in Chapter 4.

Note that the remainder of T_{44} is

$$r_4 h^8 (b-a) f^{(8)}(\xi) \approx r_4 (b-a) \Delta^8 f(a), \quad r_4 = 16/4725,$$

where $\Delta^8 f(a)$ uses the same function values as T_{44} and C_8. So we can use $r_4(b-a)\Delta^8 f(a)$ as an asymptotically correct error estimate for T_{44}.

We have assumed so far that the integrand is a real function $f \in C^{2m+2}[a, b]$. For example, if the integrand $f(x)$ has an algebraic endpoint singularity,

$$f(x) = x^\beta h(x), \quad -1 < \beta \le 0,$$

where $h(x) \in C^{p+1}[a, b]$, this assumption is not valid. In this case an asymptotic error expansion of the form

$$T(h) - I = \sum_{q=1}^{n} a_q h^{q+\beta} + \sum_{q=2}^{q} b_q h^q + O(h^{q+1}) \tag{5.2.11}$$

can be shown to hold for a trapezoidal sum. Similar but more complicated expansions can be obtained for other classes of singularities. If $p = -1/2$, then $T(h)$ has an error expansion in $h^{1/2}$:

$$T(h) - I = a_1 h^{3/2} + b_2 h^2 + a_2 h^{5/2} + b_3 h^3 + a_3 h^{5/2} + \cdots.$$

Richardson extrapolation can then be used with the denominators

$$2^{p_j} - 1, \quad p_j = 1.5, 2, 2.5, 3, \ldots.$$

Clearly the convergence acceleration will be much less effective than in the standard Romberg case.

In Richardson extrapolation schemes the exponents in the asymptotic error expansions have to be known *explicitly*. In cases when *the exponents are unknown* a nonlinear extrapolation scheme like the ϵ algorithm should be used. In this a two-dimensional array of numbers $\epsilon_k^{(p)}$, initialized with the trapezoidal approximations $T_m = T(h_m)$, $h_m = (b-a)/2^m$, is computed by the recurrence relation

$$\begin{aligned}
&\epsilon_{-1}^{(m)} = 0, \quad m = 1 : n-1, \ldots, \\
&\epsilon_0^{(m)} = T_m, \quad m = 0 : n, \\
&\epsilon_{k+1}^{(m)} = \epsilon_{k-1}^{(m+1)} + \frac{1}{\epsilon_k^{(m+1)} - \epsilon_k^{(m)}}, \quad k = 0 : n-2, \quad m = 0 : n-k-1.
\end{aligned}$$

Example 5.2.3.

Accelerating the sequence of trapezoidal sums using the epsilon algorithm may work when Romberg's method fails. In the integral

$$\int_0^1 \sqrt{x}\, dx = 2/3,$$

the integrand has a singularity at the left endpoint.

Using the trapezoidal rule with $2^k + 1$ points, $k = 0 : 9$, the error is divided roughly by $2\sqrt{2} \approx 2.828$ when the step size is halved. For $k = 9$ we get he approximation $I \approx 0.66664\,88815$ with an error $0.18 \cdot 10^{-4}$.

Applying the ϵ algorithm to these trapezoidal sums, we obtained the accelerated values displayed in the table below. (Recall that the quantities in odd-numbered columns are only intermediate quantities.) The magnitude of the error in $\epsilon_8^{(1)}$ is close to full IEEE double precision. Note that we did not use any a priori knowledge of the error expansion.

k	$\epsilon_{2k}^{(9-2k)}$	Error
0	0.66664 88815 4995	$-0.1779 \cdot 10^{-4}$
1	0.66666 67335 1817	$0.6685 \cdot 10^{-7}$
2	0.66666 66666 6037	$-0.6292 \cdot 10^{-11}$
3	0.66666 66666 6669	$0.268 \cdot 10^{-13}$
4	0.66666 66666 6666	$-0.044 \cdot 10^{-13}$

An application of the epsilon algorithm to computing the integral of an oscillating integrand to high precision is given in Example 5.2.5.

5.2.3 Oscillating Integrands

Highly oscillating integrals of the form

$$I[f] = \int_a^b f(x)e^{i\omega g(x)}\,dx, \tag{5.2.12}$$

where $f(x)$ is a slowly varying function and $e^{i\omega g(x)}$ is oscillating, frequently occur in applications from electromagnetics, chemistry, fluid mechanics, etc. Such integrals are allegedly difficult to compute. When a standard numerical quadrature rule is used to compute (5.2.12), using a step size h such that $\omega h \ll 1$ is required. For large values of ω this means an exceedingly small step size and a large number of function evaluations.

Some previously mentioned techniques such as using a simple comparison problem, or a special integration formula, can be effective also for an oscillating integrand. Consider the case of a Fourier integral, where $g(x) = x$, in (5.2.12). The trapezoidal rule gives the approximation

$$I[f] \approx \frac{1}{2}h\big(f_0e^{i\omega a} + f_Ne^{i\omega b}\big) + h\sum_{j=1}^{N-1} f_je^{i\omega x_j}, \tag{5.2.13}$$

where $h = (b-a)/N$, $x_j = a + jh$, $f_j = f(x_j)$. This formula cannot be used unless $\omega h \ll 1$, since its validity is based on the assumption that the whole integrand varies linearly over an interval of length h.

A better method is obtained by approximating just $f(x)$ by a piecewise linear function,

$$p_j(x) = f_j + \frac{x - x_j}{h}(f_{j+1} - f_j), \quad x \in [x_j, x_{j+1}], \quad j = 0 : N-1.$$

The integral over $[x_j, x_{j+1}]$ can then be approximated by

$$\int_{x_j}^{xj+1} p_j(x)e^{i\omega x}\,dx = he^{i\omega x_j}\left(f_j\int_0^1 e^{i\omega ht}\,dt + (f_{j+1} - f_j)\int_0^1 te^{i\omega ht}\,dt\right),$$

where we have made the change of variables $x - x_j = th$. Let $\theta = h\omega$ be the **characteristic frequency**. Then

$$\int_0^1 e^{i\theta t}\,dt = \frac{1}{i\theta}(e^{i\theta} - 1) = \alpha,$$

and using integration by parts

$$\int_0^1 te^{i\theta t}\,dt = \frac{1}{i\theta}te^{i\theta t}\Big|_0^1 - \frac{1}{i\theta}\int_0^1 e^{i\theta t}\,dt = \frac{1}{i\theta}e^{i\theta} + \frac{1}{\theta^2}(e^{i\theta} - 1) = \beta.$$

Here α and β depend on θ but not on j. Summing the contributions from all intervals we obtain

$$\begin{aligned} &h(\alpha - \beta)\sum_{j=0}^{N-1} f_je^{i\omega x_j} + h\beta\sum_{j=0}^{N-1} f_{j+1}e^{i\omega x_j} \\ &= h(\alpha - \beta)\sum_{j=0}^{N-1} f_je^{i\omega x_j} + h\beta e^{-i\theta}\sum_{j=1}^{N} f_je^{i\omega x_j}. \end{aligned}$$

The resulting quadrature formula has the same form as (5.2.13),

$$I[f] \approx hw(\theta) \sum_{j=0}^{N-1} f_j e^{i\omega x_j} + hw_N(\theta)\big(f_N e^{i\omega x_N} - f_0 e^{i\omega x_0}\big), \tag{5.2.14}$$

with the weights $w_0(\theta) = \alpha - \beta$, $w_N(\theta) = \beta e^{-i\theta}$, and $w(\theta) = w_0 + w_N$. Then

$$w_0(\theta) = w_N(-\theta) = \frac{1 - i\theta - e^{-i\theta}}{\theta^2}, \qquad w(\theta) = \frac{(\sin \frac{1}{2}\theta)^2}{(\frac{1}{2}\theta)^2}. \tag{5.2.15}$$

Note that the same trigonometric sum is involved, now multiplied with the real factor $w(\theta)$. The sum in (5.2.14) can be computed using the FFT; see Sec. 4.7.3.

The weights tend to the trapezoidal weights when $\omega h \to 0$ (check this!). For small values of $|\theta|$ there will be cancellation in these expressions for the coefficients and the Taylor expansions should be used instead; see Problem 5.2.11.

A similar approach for computing trigonometric integrals of one of the forms

$$\int_a^b f(x)\cos(\omega x)\,dx, \qquad \int_a^b f(x)\sin(\omega x)\,dx \tag{5.2.16}$$

was advanced by Filon [116] already in 1928. In this the interval $[a, b]$ is divided into an even number of $2N$ subintervals of equal length $h = (b - a)/(2N)$. The function $f(x)$ is approximated over each double interval $[x_{2i}, x_{2(i+1)}]$ by the quadratic polynomial $p_i(x)$ interpolating $f(x)$ at x_{2i}, x_{2i+1}, and $x_{2(i+1)}$. Filon's formula is thus related to Simpson's rule. (The formula (5.2.14) is often called the Filon-trapezoidal rule.) For $\omega = 0$, Filon's formula reduces to the composite Simpson's formula, but it is not exact for cubic functions $f(x)$ when $\omega \neq 0$.

The integrals

$$\int_{x_{2i}}^{x_{2(i+1)}} p_i(x)\cos(\omega x)\,dx, \qquad \int_{x_{2i}}^{x_{2(i+1)}} p_i(x)\sin(\omega x)\,dx$$

can be computed analytically using integration by parts. This leads to Filon's integration formula; see the Handbook [1, Sec. 25.4.47].

Similar formulas can be developed by using different polynomial approximations of $f(x)$. Einarsson [103] uses a cubic spline approximation of $f(x)$ and assumes that the first and second derivatives of f at the boundary are available. The resulting quadrature formula has an error which usually is about four times smaller than that for Filon's rule.

Using the Euler–Maclaurin formula on the function it can be shown (see Einarsson [104, 105]) that the expansion of the error for the Filon-trapezoidal rule, the Filon–Simpson method, and the cubic spline method contain only even powers of h. Thus the accuracy can be improved by repeated Richardson extrapolation. For example, if the Filon-trapezoidal rule is used with a sequence of step sizes $h, h/2, h/4, \ldots$, then one can proceed as in Romberg's method. Note that the result after one extrapolation is not exactly equal to the Filon–Simpson rule, but gives a marginally better result when $\omega h = O(1)$.

Example 5.2.4 (*Einarsson* [104]).

Using the standard trapezoidal rule to compute the Fourier integral

$$I = \int_0^\infty e^{-x} \cos \omega x \, dx = \frac{1}{1+\omega^2}$$

gives the result

$$I_T = h\left(\frac{1}{2} + \Re \sum_{j=1}^{\infty} e^{-jh} e^{ih\omega j}\right) = \frac{h}{2} \frac{\sinh h}{\cosh h - \cosh h\omega},$$

where h is the step length. Assuming that $h\omega$ is sufficiently small we can expand the right-hand side in powers of h, obtaining

$$I_T = I\left(1 + \frac{h^2}{12}(1+\omega^2) + \frac{h^4}{720}(1+\omega^2)(3\omega^2 - 1) + O(h^6)\right).$$

For the Filon-trapezoidal rule the corresponding result is

$$I_{FT} = \left(\frac{\sin \frac{1}{2}\omega h}{\frac{1}{2}\omega h}\right)^2 I_T = I\left(1 + \frac{h^2}{12} - \frac{h^4}{720}(3\omega^2 + 1) + O(h^6)\right).$$

For small values of ω the two formulas are seen to be equivalent. However, for larger values of ω, the error in the standard trapezoidal rule increases rapidly.

The expansions only have even powers of h. After one step of extrapolation the Filon-trapezoidal rule gives a relative error equal to $h^4(3\omega^2 + 1)/180$, which can be shown to be slightly better than for the Filon–Simpson rule.

More general Filon-type methods can be developed as follows. Suppose we wish to approximate the integral

$$I[f] = \int_0^h f(x) e^{i\omega x} \, dx = h \int_0^1 f(ht) e^{ih\omega t} \, dt, \tag{5.2.17}$$

where f is itself sufficiently smooth. We choose distinct nodes $0 \le c_1 < c_2 < \cdots < c_\nu \le 1$ and consider the quadrature formula interpolatory weights $b_1, b_2, \ldots, b_\nu$. Let s be the largest integer j so that

$$\int_0^1 t^{j-1} \gamma(t) \, dt = 0, \quad \gamma(t) = \prod_{i=1}^{\nu} (t - c_i). \tag{5.2.18}$$

Then by Theorem 5.1.3, $s \le \nu$, and the order of the corresponding quadrature formula is $p = \nu + s$. A Filon-type quadrature rule is now obtained by interpolating f by the polynomial

$$p(x) = \sum_{k=1}^{\nu} \ell_k(x/h) f(c_k h),$$

where ℓ_k is the kth cardinal polynomial of Lagrange interpolation. Replacing f by p in (5.2.17), we obtain

$$Q_h[f] = \sum_{k=1}^{\nu} \beta_k(\theta) f(c_k h), \quad \beta_k(\theta) = \int_0^1 \ell_k(t) e^{ih\omega t}\, dt. \tag{5.2.19}$$

The coefficients $\beta_k(\theta)$ can be computed also from the moments

$$\mu_k(\theta) = \int_0^1 t^k e^{i\theta t}\, dt, \quad k = 0 : \nu - 1,$$

by solving the Vandermonde system

$$\sum_{j=1}^{\nu} \beta_j(\theta) c_j^k = \mu_k(\theta), \quad k = 0 : \nu - 1.$$

The derivation of the Filon-type quadrature rule is analogous to considering $e^{i\theta t}$ as a complex-valued weight function. However, any attempt to choose the nodes c_j so that the order of the integration rule is increased over ν is likely to lead to complex nodes and useless formulas.

The general behavior of Filon-type quadrature rules is that for $0 < \theta \ll 1$ they show similar accuracy to the corresponding standard interpolatory rule. For $\theta = O(1)$ they are also very effective, although having order $\nu \leq p$. The common wisdom is that if used in the region where θ is large they can give large errors. However, Einarsson [104] observed that the cubic spline method gives surprisingly good results also for large values of θ, seemingly in contradiction to the condition in the sampling theorem that at least two nodes per full period are needed.

Iserles [205] shows that once appropriate Filon-type methods are used the problem of highly oscillatory quadrature becomes relatively simple. Indeed, *the precision of the calculation actually increases as the oscillation grows*. This is quantified in the following theorem.

Theorem 5.2.2 (*Iserles* [205, Theorem 2]).

Let $\theta = h\omega$ be the characteristic frequency. Then the error $E_h[f]$ in the Filon-type quadrature formula (5.2.19) *is*

$$E_h[f] \sim \mathcal{O}(h^{\nu+1}\theta^{-p}), \tag{5.2.20}$$

where $p = 2$ if $c_1 = 0$ and $c_\nu = 0$; $p = 1$ otherwise.

To get the best error decay the quadrature formula should include the points $c_1 = 0$ and $c_\nu = 1$. This is the case both for the Filon-trapezoidal method and the Filon–Simpson rule. Figure 5.2.1 shows the absolute value of the integral

$$I = \int_0^h e^x e^{i\omega x}\, dx = (e^{(1+i\omega)h} - 1)/(1 + i\omega),$$

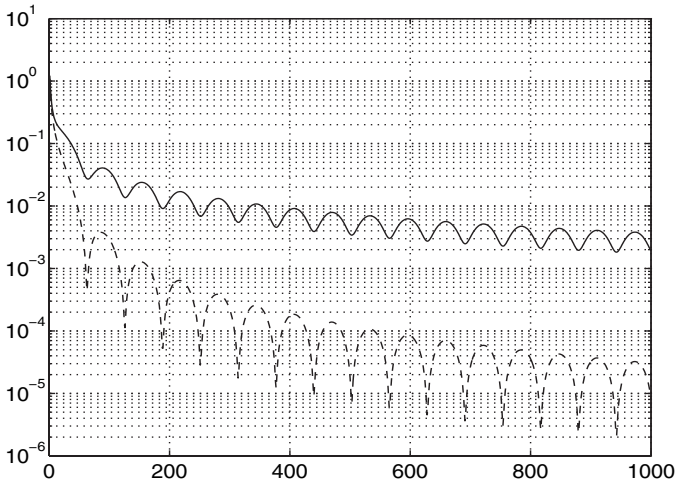

Figure 5.2.1. *The Filon-trapezoidal rule applied to the Fourier integral with $f(x) = e^x$, for $h = 1/10$, and $\omega = 1 : 1000$; solid line: exact integral; dashed line: absolute value of the error.*

and the absolute value of the error in the Filon-trapezoidal approximation for $h = 0.1$ and $\omega = 1 : 1000$. Clearly the error is small and becomes smaller as the characteristic frequency grows!

Sometimes convergence acceleration of a related series can be successfully employed for the evaluation of an integral with an oscillating integrand. Assume that the integral has the form

$$I[f] = \int_0^\infty f(x)\sin(g(x))\,dx,$$

where $g(x)$ is an increasing function and both $f(x)$ and $g(x)$ can be approximated by a polynomial. Set

$$I[f] = \sum_{n=0}^{\infty}(-1)^N u_n, \quad u_n = \int_{x_n}^{x_{n+1}} f(x)\,|\sin(g(x))|\,dx,$$

where $x_0, x_1, x_2, \ldots$ are the successive zeros of $\sin(g(x))$. The convergence of this alternating series can then be improved with the help of repeated averaging; see Sec. 3.4.3. Alternatively a sequence of partial sums can be computed, which then is accelerated by the epsilon algorithm. Sidi [316] has developed a useful extrapolation method for oscillatory integrals over an infinite interval.

Example 5.2.5 (*Gautschi* [146]).

The first problem in "The 100-digit Challenge"[174] is to compute the integral

$$I = \lim_{\epsilon\to 0}\int_\epsilon^1 t^{-1}\cos(t^{-1}\ln t)\,dt \tag{5.2.21}$$

[174] See [40] and `www.siam.org/books/100digitchallenge`.

to ten decimal places. Since the integrand is densely oscillating as $t \downarrow 0$ and at the same time the oscillations tend to infinity (see Figure 5.2.2), this is a challenging integral to compute numerically. (Even so the problem has been solved to an accuracy of 10,000 digits!)

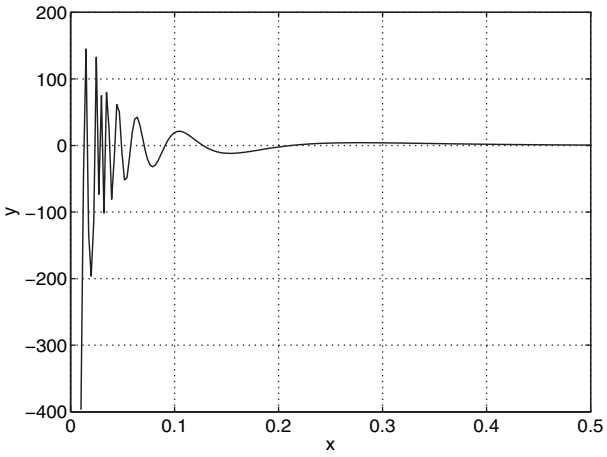

Figure 5.2.2. *The oscillating function* $x^{-1}\cos(x^{-1}\ln x)$.

With the change of variables $u = t^{-1}$, $du = -t^{-2}dt$, we get

$$I = \int_1^\infty u^{-1}\cos(u\ln u)\,du. \tag{5.2.22}$$

Making the further change of variables $x(u) = u\ln u$, we have $dx = (1+\ln u)du = (u+x)u^{-1}du$, and the integral becomes

$$I = \int_0^\infty \frac{\cos x}{x+u(x)}\,dx. \tag{5.2.23}$$

The inverse function $u(x)$ is smooth and relatively slowly varying, with $u(0) = 1, u'(0) = 1$. For $x > 0$, $u'(x)$ is positive and decreasing, while $u''(x)$ is negative and decreasing in absolute value. The function $u(x)$ is related to Lambert's W-function, which is the inverse of the function $x = we^w$ (see Problem 3.1.12). Clearly $u(x) = e^{w(x)}$.

The zeros of the integrand in (5.2.23) are at odd multiples of $\pi/2$. We split the interval of integration into intervals of constant sign for the integrand

$$I = \int_0^{\pi/2} \frac{\cos x}{x+u(x)}\,dx + \sum_{k=1}^{\infty} I_k, \quad I_k = \int_{(2k-1)\pi/2}^{(2k+1)\pi/2} \frac{\cos x}{x+u(x)}\,dx.$$

Changing variables $x = t + k\pi$ in the integrals I_k,

$$I_k = (-1)^k \int_{-\pi/2}^{\pi/2} \frac{\cos t}{t + k\pi + u(t+k\pi)}\,dt. \tag{5.2.24}$$

The terms form an alternating series with terms decreasing in absolute values. It is, however, slowly converging and for an error bound of $\frac{1}{2}10^{-5}$ about 116,000 terms would be needed. Accelerating the convergence using the epsilon algorithm, Gautschi found that using only 21 terms in the series suffices to give an accuracy of about 15 decimal digits:

$$I = 0.32336\,74316\,77779.$$

The integrand in the integrals (5.2.24) is regular and smooth. For computing these, for example, a Clenshaw–Curtis quadrature rule can be used after shifting the interval of integration to $[-1, 1]$; see also Problem 5.3.11.

5.2.4 Adaptive Quadrature

Suppose the integrand $f(x)$ (or some of its low-order derivatives) has strongly varying orders of magnitude in different parts of the interval of integration $[a, b]$. Clearly, one should then use *different step sizes in different parts of the integration interval*. If we write

$$\int_a^b = \int_a^{c_1} + \int_{c_1}^{c_2} + \cdots + \int_{c_1}^b,$$

then the integrals on the right-hand side can be treated as independent subproblems. In **adaptive quadrature methods** step sizes are automatically adjusted so that the approximation satisfies a prescribed error tolerance:

$$\left| I - \int_a^b f(x)\,dx \right| \le \epsilon. \tag{5.2.25}$$

A common difficulty is when the integrand exhibits one or several sharp peaks as exemplified in Figure 5.2.3. It should be realized that without further information about the location of the peaks all quadrature algorithms can fail if the peaks are sharp enough.

We consider first a fixed-order adaptive method based on Simpson's rule. For a subinterval $[a, b]$, set $h = (b - a)$ and compute the trapezoidal approximations

$$T_{00} = T(h), \qquad T_{10} = T(h/2), \qquad T_{20} = T(h/4).$$

The extrapolated values

$$T_{11} = (4T_{10} - T_{00})/3, \qquad T_{21} = (4T_{20} - T_{10})/3$$

are equivalent to (the composite) Simpson's rule with step length $h/2$ and $h/4$, respectively. We can also calculate

$$T_{22} = (16T_{21} - T_{11})/15,$$

which is Milne's method with step length $h/4$ with remainder equal to

$$(2/945)(h/4)^6(b - a)f^{(6)}(\xi).$$

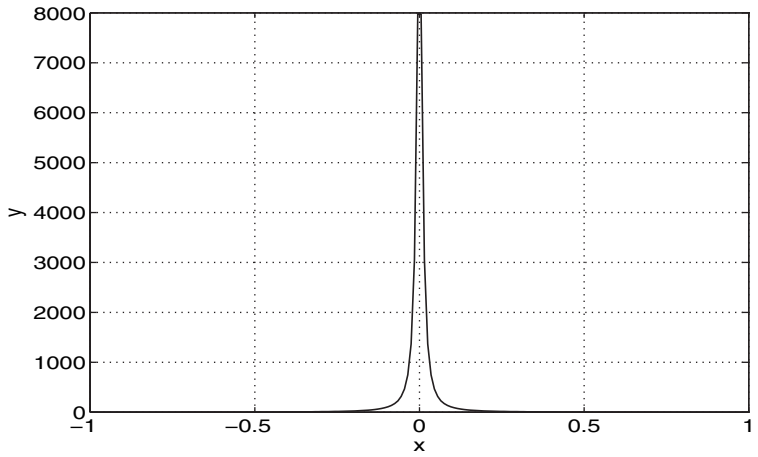

Figure 5.2.3. *A needle-shaped function.*

For T_{22} we can estimate the truncation error by $|T_{22} - T_{21}|$, which is usually a strong overestimate. We *accept* the approximation if

$$|T_{22} - T_{21}| < \frac{h_j \epsilon}{b - a}, \tag{5.2.26}$$

i.e., we require the error to be *less than* $\epsilon/(b - a)$ *per unit step*. Otherwise we *reject* the approximation, and subdivide the interval in two intervals, $[a_j, \frac{1}{2}(a_j + b_j)]$, $[\frac{1}{2}(a_j + b_j), b_j]$. The same rule is now applied to these two subintervals.

Note that if the function values computed previously are saved, these can be reused for the new intervals. We start with one interval $[a, b]$ and carry on subdivisions until the error criterion in (5.2.26) is satisfied for all intervals. Since the total error is the sum of errors for all subintervals, we then have the required error estimate:

$$R_T < \sum_j \frac{h_j \epsilon}{b - a} = \epsilon.$$

The possibility that a user might try to integrate a nonintegrable function (e.g., $f(x) = x^{-1}$ on $[0, 1]$) cannot be neglected. In principle it is not possible to decide whether a function $f(x)$ is integrable on the basis of a finite sample $f(x_1), \ldots, f(x_n)$ of function values. Therefore, it is necessary to impose

1. an upper limit on the number of function evaluation,

2. a lower limit on the size of the subregions.

This means that premature termination may occur even when the function is only close to being nonintegrable, for example, $f(x) = x^{-0.99}$.

Many different adaptive quadrature schemes exist. Here we shall illustrate one simple scheme based on a five-point closed Newton–Cotes' rule, which applies bisection in a

locally adaptive strategy. All function evaluations contribute to the final estimate. In many situations it might be preferable to specify a *relative error tolerance*:

$$tol = \eta \left| \int_a^b f(x)\,dx \right|.$$

A more complete discussion of the choice of termination criteria in adaptive algorithms is found in Gander and Gautschi [128].

ALGORITHM 5.3. *Adaptive Simpson.*

Let f be a given function to be integrated over $[a, b]$. The function `adaptsimp` uses a recursive algorithm to compute an approximation with an error less than a specified tolerance $\tau > 0$.

```
function [I,nf] = adaptsimp(f,a,b,tol);
% ADAPTSIMP calls the recursive function ADAPTREC to compute
% the integral of the vector-valued function f over [a,b];
% tol is the desired absolute accuracy; nf is the number of
% function evaluations.
%
ff = feval(f,[a, (a+b)/2, b]);
nf = 3; % Initial Simpson approximation
I1 = (b - a)*[1, 4, 1]*ff'/6;
% Recursive computation
[I,nf] =  adaptrec(f,a,b,ff,I1,tol,nf);

function [I,nf] = adaptrec(f,a,b,ff,I1,tol,nf);
h = (b - a)/2;
fm = feval(f, [a + h/2, b - h/2]);
nf = nf + 2;
% Simpson approximations for left and right subinterval
fR = [ff(2); fm(2); ff(3)];
fL = [ff(1); fm(1); ff(2)];
IL = h*[1, 4, 1]*fL/6;
IR = h*[1, 4, 1]*fR/6;
I2 = IL + IR;
I  = I2 + (I2 - I1)/15;   % Extrapolated approximation
if  abs(I - I2) > tol     % Refine both subintervals
   [IL,nf] = adaptrec(f,a,a+h,fL,IL,tol/2,nf);
   [IR,nf] = adaptrec(f,b-h,b,fR,IR,tol/2,nf);
   I  = IL + IR;
 end
```

Note that in a **locally adaptive** algorithm using a recursive partitioning scheme, the subintervals are processed from left to right until the integral over each subinterval satisfies some error requirement. This means that an a priori initial estimate of the whole integral,

needed for use in a relative local error estimate cannot be updated until all subintervals are processed and the computation is finished. Hence, if a relative tolerance is specified, then an estimate of the integral is needed before the recursion starts. This is complicated by the fact that the initial estimate might be zero, for example, if a periodic integrand is sampled at equidistant intervals. Hence a combination of relative and absolute criteria might be preferable.

Example 5.2.6.

This algorithm was used to compute the integral

$$\int_{-4}^{4} \frac{dx}{1+x^2} = 2.65163532733607$$

with an absolute tolerance 10^{-p}, $p = 4, 5, 6$. The following approximations were obtained.

I	tol	n	Error
2.65162 50211	10^{-4}	41	$1.0\,10^{-5}$
2.65163 52064	10^{-5}	81	$1.2\,10^{-7}$
2.65163 5327353	10^{-6}	153	$-1.7\,10^{-11}$

Note that the actual error is much smaller than the required tolerance.

So far we have considered adaptive routines, which use fixed quadrature rules on each subinterval but where the partition of the interval depends on the integrand. Such an algorithm is said to be **partition adaptive**. We can also consider **doubly adaptive** integration algorithms. These can choose from a sequence of increasingly higher-order rules to be applied to the current subinterval. Such algorithms use a selection criterion to decide at each stage whether to subdivide the current subinterval or to apply a higher-order rule. Doubly adaptive routines cope more efficiently with smooth integrands.

Many variations on the simple scheme outlined above are possible. For example, we could base the method on a higher-order Romberg scheme, or even try to choose an optimal order for each subinterval. Adaptive methods work even when the integrand $f(x)$ is badly behaved. But if f has singularities or unbounded derivatives, the error criterion may never be satisfied. To guard against such cases it is necessary to include some bound of the number of recursion levels that are allowed. It should be kept in mind that although adaptive quadrature algorithms are convenient to use they are in general less efficient than methods which have been specially adapted for a particular problem.

We finally warn the reader that *no automatic quadrature routine can always be guaranteed to work*. Indeed, any estimate of $\int_a^b f(x)\,dx$ based solely on the value of $f(x)$ on finitely many points can fail. The integrand $f(x)$ may, for example, be nonzero only on a small subset of $[a, b]$. An adaptive quadrature rule based only on samples $f(x)$ in a finite number of points theoretically may return the value zero in such a case!

We recall the remark that evaluation of the integral $\int_a^b f(x)\,dx$ is equivalent to solving an initial value problem $y' = f(x)$, $y(a) = 0$, for an ordinary differential equation. For such problems sophisticated techniques for adaptively choosing step size and order in the integration have been developed. These may be a good alternative choice for handling difficult cases.

Review Questions

5.2.1 (a) Give an account of the theoretical background of Romberg's method.

(b) For which values of k are the elements T_{kk} in the Romberg scheme identical to closed Newton–Cotes' formulas?

5.2.2 Romberg's method uses extrapolation of a sequence of trapezoidal approximations computed for a sequence of step sizes $h_0, h_1, h_2, \ldots$. What sequences have been suggested and what are their relative merits?

5.2.3 When the integrand has a singularity at one of the endpoints, many quadrature methods converge very slowly. Name a few possible ways to resolve this problem.

5.2.4 Romberg's method works only when the error of the trapezoidal rule has an expansion in even powers of h. If this is not the case, what other extrapolations methods should be tried?

5.2.5 Describe at least two methods for treating an integral with an oscillating integrand.

5.2.6 In partition adaptive quadrature methods the step sizes are locally adopted. Discuss how the division into subintervals can be controlled.

Problems and Computer Exercises

5.2.1 Is it true that (the short version of) Simpson's formula is a particular case of Gregory's formula?

5.2.2 Use Romberg's method to compute the integral $\int_0^4 f(x)\,dx$, using the following (correctly rounded) values of $f(x)$. Need all the values be used?

x	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
$f(x)$	−4271	−2522	−499	1795	4358	7187	10,279	13,633	17,247

5.2.3 (a) Suppose that the form of the error of Romberg's method is known, but not the error constant r_k. Determine r_k numerically for $k = 3$ and $k = 4$, by computing the Romberg scheme for $f(x) = x^{2k}$.

(b) Prove the formula for the error constant of Romberg's method.

5.2.4 Compute by the Euler–Maclaurin formula, or rather the composite trapezoidal rule,

$$\text{(a)}\ \int_0^\infty e^{-x^2/2}dx, \qquad \text{(b)}\ \int_0^\infty \frac{dx}{\cosh(\pi x)}$$

as accurately as you can with the normal precision of your computer (or software). Then find out empirically how the error depends on h. Make semilogarithmic plots on the same screen. How long a range of integration do you need?

5.2.5 (a) Use Romberg's method and Aitken acceleration to compute the integral

$$I[f] = \int_1^\infty \frac{1}{1+x^2}\,dx = \int_1^2 + \int_2^4 + \int_4^8 + \cdots.$$

Determine where to terminate the expansion, and then use Aitken acceleration to find $I[f]$. Compare with the exact result. Think of an error estimate that can be used if the exact result is not known.

(b) Treat in the same way

$$\int_1^\infty \frac{1}{\sqrt{x+x^3}}.$$

Compare the computational effort for the computation of the tail $\int_R^\infty$ by acceleration and by series expansion with the same accuracy.

5.2.6 Modify the MATLAB function `romberg` so that it uses rational extrapolation according to the recursion (5.2.7) instead of polynomial extrapolation. Use the modified program to compute the integral in Example 5.2.2. Compare the results for the two different extrapolation methods.

5.2.7 Apply the MATLAB program `romberg` in Sec. 5.2.2 and repeated averages on the integral

$$\int_0^{1000} x\cos(x^3)\,dx.$$

Try to obtain the results with 10 decimal places.

5.2.8 (a) Show the following series expansions for the coefficients in the Filon-trapezoidal formula:

$$w_0(\theta) = w_N(-\theta) = \frac{1}{2} - \frac{\theta^2}{24} + \frac{\theta^4}{720} - \cdots + i\Big(\frac{\theta}{6} - \frac{\theta^3}{120} + \frac{\theta^5}{5040} - \cdots\Big),$$
$$w(\theta) = w_0(\theta) + w_N(-\theta) = 1 - \frac{\theta^2}{12} + \frac{\theta^4}{360} - \cdots.$$

(b) For what value of θ should you switch to using the series expansions above, if you want to minimize an upper bound for the error in the coefficients?

5.3 Quadrature Rules with Free Nodes

5.3.1 Method of Undetermined Coefficients

We have previously seen how to derive quadrature rules using Lagrange interpolation or operator series. We now outline another general technique, the method of undetermined coefficients, for determining quadrature formulas of maximum order with both free and prescribed nodes.

Let L be a linear functional and consider approximation formulas of the form

$$Lf \approx \tilde{L}f = \sum_{i=1}^{p} a_i f(x_i) + \sum_{j=1}^{q} b_j f(z_j), \tag{5.3.1}$$

where the x_i are p given nodes, while the z_j are q **free nodes**. The latter are to be determined together with the weight factors a_i, b_j. The altogether $p + 2q$ parameters in the formula

are to be determined, if possible, so that the formula becomes exact for all polynomials of degree less than $N = p + 2q$. We introduce the two node polynomials

$$r(x) = (x - x_1) \cdots (x - x_p), \qquad s(x) = (x - z_1) \cdots (x - z_q) \tag{5.3.2}$$

of degree p and q, respectively.

Let $\phi_1, \phi_2, \ldots, \phi_N$ be a basis of the space of polynomials of degree less than N. We assume that the quantities $L\phi_k$, $k = 1 : p + 2q$ are known. Then we obtain the *nonlinear* system

$$\sum_{i=1}^{p} \phi_k(x_i)a_i + \sum_{j=1}^{q} \phi_k(z_j)b_j = L\phi_k, \quad k = 1, 2, \ldots, p + 2q, \tag{5.3.3}$$

for the $p + 2q$ parameters. This system is nonlinear in z_j, but of a very special type. Note that the free nodes z_j appear in a symmetric fashion; the system (5.3.3) is invariant with respect to permutations of the free nodes together with their weights. We therefore first ask for their **elementary symmetric functions**, i.e., for the coefficients g_j of the node polynomial

$$s(x) = \phi_{q+1}(x) - \sum_{j=1}^{q} g_j \phi_j(x) \tag{5.3.4}$$

that has the free nodes $z_1, \ldots, z_q$ as zeros. We change the basis to the set

$$\phi_1(x), \ldots, \phi_q(x), s(x)\phi_1(x), \ldots, s(x)\phi_{p+q}(x).$$

In the system (5.3.3), the equations for $k = 1 : q$ will not be changed, but the equations for $k = 1 + q : p + 2q$ become

$$\sum_{i=1}^{p} \phi_{k'}(x_i)s(x_i)a_i + \sum_{j=1}^{q} \phi_{k'}(z_j)s(z_j)b_j = L(s\phi_{k'}), \quad 1 \le k' \le p + q. \tag{5.3.5}$$

Here the second sum disappears since $s(z_j) = 0$ for all j. (This is the nice feature of this treatment.) Further, by (5.3.4),

$$L(s\phi_{k'}) = L(\phi_{k'}\phi_{q+1}) - \sum_{j=1}^{q} L(\phi_{k'}\phi_j)g_j, \quad 1 \le k' \le p + q. \tag{5.3.6}$$

We thus obtain the following *linear* system for the computation of the $p + q$ quantities, g_j, and $A_i = s(x_i)a_i$:

$$\sum_{j=1}^{q} L(\phi_{k'}\phi_j)g_j + \sum_{i=1}^{p} \phi_{k'}(x_i)A_i = L(\phi_{k'}\phi_{q+1}), \quad k' = 1 : p + q. \tag{5.3.7}$$

The weights of the fixed nodes are $a_i = A_i/s(x_i)$. The free nodes z_j are then determined by finding the q roots of the polynomial $s(x)$. Methods for computing roots of a polynomial are given in Sec. 6.5. Finally, with a_i and z_j known, the weights b_j are obtained by the solution of the first q equations of the system (5.3.3) which are linear in b_j.

The remainder term $Rf = (Lf - \tilde{L}f)$ of the method, exact for all polynomials of degree less than $N = p + 2q$, is of the form

$$Rf = R(f - P_N) \approx c_N f^{(N)}(\xi), \quad c_N = R(x^N)/N!,$$

where c_N is called the error constant. Note that $R(x^N) = R(\phi_{N+1})$, where $\phi_{(N+1)}$ is any monic polynomial of degree N, since $x^N - \phi_{(N+1)}$ is a polynomial of degree less than N. Hence, for the determination of the error constant we compute the difference between the right-hand and the left-hand sides of

$$\sum_{i=1}^{p} \phi_k(x_i)\, a_i + \sum_{j=1}^{q} \phi_k(z_j)\, b_j + N! c_N = L\phi_{N+1}, \quad N = p + 2q, \tag{5.3.8}$$

and divide by $(N)!$. If, for example, a certain kind of symmetry is present, then it can happen that $c_{p+2q} = 0$. The formula is then more accurate than expected, and we take $N = p + 2q + 1$ instead. The case that also $c_{p+2q+1} = 0$ may usually be ignored. It *can* occur if several of the given nodes are located, where free nodes would have been placed.

From a pure mathematical point of view all bases are equivalent, but equation (5.3.3) may be better conditioned with some bases than with others, and this turns out to be an important issue when $p + 2q$ is large. We mention three different situations.

(i) The most straightforward choice is to set $[a, b] = [0, 1]$ and use the monomial basis $\phi_k(x) = x^{k-1}$, $x \in (0, b)$ (b may be infinite). For this choice the condition number of (5.3.3) increases exponentially with $p + 2q$. Then the free nodes and corresponding weights may become rather inaccurate when $p + 2q$ is large. It is usually found, however, that unless the condition number is so big that the solution breaks down completely, the computed solution will satisfy equation (5.3.3) with a small residual. This is what really matters for the application of formula (5.3.1).

(ii) Take $[a, b] = [-1, 1]$, and assume that the weight function $w(x)$ and the given nodes x_i are symmetrical with respect to the origin. Then the weights a_i and b_i, and the free nodes z_j will also be symmetrically located, and with the monomial basis it holds that $L(\phi_k(x)) = 0$, when k is even. If $p = 2p'$ is even, the number of parameters will be reduced to $p' + q$ by the transformation $x = \sqrt{\xi}$, $\xi \in [0, b^2]$. Note that $w(x)$ will be replaced by $w(\sqrt{\xi})/\sqrt{\xi}$. If p is odd, one node is at the origin, and one can proceed in an analogous way. This should also reduce the condition number approximately to its square root, and it is possible to derive in a numerically stable way formulas with about twice as high an order of accuracy as in the unsymmetric case.

(iii) Taking ϕ_k to be the orthogonal polynomials for the given weight function will give a much better conditioned system for determining the weights. This case will be considered in detail in Sec. 5.3.5.

Example 5.3.1.

Consider the linear functional $L(f) = \int_0^1 f(x)\, dx$. Set $p = 0, q = 3$ and choose the monomial basis $\phi_i(x) = x^{i-1}$. Introducing the node polynomial

$$s(x) = (x - z_1)(x - z_2)(x - z_3) = x^3 - s_3x^2 - s_2x - s_1,$$

the linear system (5.3.6) becomes

$$\begin{pmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 1/5 \\ 1/6 \end{pmatrix}.$$

The exact solution is $s_1 = 1/20$, $s_2 = -3/5$, and $s_3 = 3/2$. The free nodes thus are the zeros of $s(x) = x^3 - 3x^2/2 + 3x/5 - 1/20$, which are $z_2 = 1/2$ and $z_{1,3} = 1/2 \pm \sqrt{3/20}$. The weights b_1, b_2, b_3 are then found by solving (5.3.3) for $k = 1:3$.

The matrix of the above system is a Hankel matrix. The reader should verify that when $p > 0$ the matrix becomes a kind of combination of a Hankel matrix and a Vandermonde matrix.

5.3.2 Gauss–Christoffel Quadrature Rules

Assume that the n nodes in a quadrature formula are chosen so that

$$(f, s) = \int_a^b p(x)s(x)w(x)\,dx = 0 \quad \forall\, p(x) \in \mathcal{P}_n, \tag{5.3.9}$$

where $s(x) = (x-x_1)(x-x_2)\cdots(x-x_n)$ is the node polynomial. Then, by Theorem 5.1.3, the corresponding interpolatory quadrature rule will have the maximum possible order $2n-1$.

We define an inner product with respect to a weight function $w(x) \geq 0$ by

$$(f, g) = \int_a^b f(x)g(x)w(x)\,dx, \tag{5.3.10}$$

and assume that the moments

$$\mu_k = (x^k, 1) = \int_a^b x^k w(x)\,dx \tag{5.3.11}$$

are defined for all $k \geq 0$, and $\mu_0 > 0$. This inner product has the important property that $(xf, g) = (f, xg)$. The condition (5.3.9) on the node polynomial can then be interpreted to mean that $s(x)$ is orthogonal to all polynomials in $\mathcal{P}_n$.

For the weight function $w(x) \equiv 1$ the corresponding quadrature rules were derived in 1814 by Gauss [133]. Formulas for more general weight functions were given by Christoffel [68] in 1858,[175] which is why these are referred to as **Gauss–Christoffel quadrature** rules.

The construction of Gauss–Christoffel quadrature rules is closely related to the theory of orthogonal polynomials. In Sec. 4.5.5 we showed how the orthogonal polynomials corresponding to the inner product (5.3.10) could be generated by a three-term recurrence formula. The zeros of these polynomials are the nodes in a Gauss–Christoffel quadrature formula. As for all interpolatory quadrature rules the weights can be determined by

[175] Elwin Bruno Christoffel (1829–1900) worked mostly in Strasbourg. He is best known for his work in geometry and tensor analysis, which Einstein later used in his theory of relativity.

integrating the elementary Lagrange polynomials (5.1.7)

$$w_i = \int_a^b \ell_i(x)w(x)\,dx, \quad \ell_i(x) = \prod_{\substack{j=1\\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}, \quad i = 1:n.$$

In Sec. 5.3.5 we will outline a more stable algorithm that determines the nodes and weights by solving the eigenvalue problem for a symmetric tridiagonal matrix defined by the coefficients in the recurrence relation.

We shall now prove some important properties of Gauss–Christoffel quadrature rules using the general theory of orthogonal polynomials.

Theorem 5.3.1.

The zeros x_i, $i = 1:n$, of the orthogonal polynomial polynomial $\varphi_{n+1}(x)$ of degree n, associated with the weight function $w(x) \geq 0$ on $[a, b]$, are real, distinct, and contained in the open interval (a, b).

Proof. Let $a < x_1 < x_2 < \cdots < x_m < b$ be the roots of $\varphi_{n+1}(x)$ of odd multiplicity, which lie in (a, b). At these roots $\varphi_{n+1}(x)$ changes sign and therefore the polynomial $q(x)\varphi_{n+1}(x)$, where

$$q(x) = (x-x_1)(x-x_2)\cdots(x-x_m),$$

has constant sign in $[a, b]$. Hence,

$$\int_a^b \varphi_{n+1}q(x)w(x)\,dx > 0.$$

But this is possible only if the degree of $q(x)$ is equal to n. Thus $m = n$ and the theorem follows. □

Corollary 5.3.2.

If $x_1, x_2, \ldots, x_n$ are chosen as the n distinct zeros of the orthogonal polynomial φ_{n+1} of degree n in the family of orthogonal polynomials associated with $w(x)$, then the formula

$$\int_a^b f(x)w(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i), \quad w_i = \int_a^b \ell_i(x)w(x)\,dx, \tag{5.3.12}$$

is exact for polynomials of degree $2n-1$.

Apart from having optimal degree of exactness equal to $2n-1$, Gaussian quadrature rules have several important properties, which we now outline.

Theorem 5.3.3.

All weights in a Gaussian quadrature rule are real, distinct, and positive.

Proof. Let

$$\ell_i(x) = \prod_{\substack{j=1\\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}, \quad i = 1:n,$$

be the Lagrange polynomials. Then the quadrature formula (5.3.12) is exact for $p(x) = (\ell_i(x))^2$, which is of degree $2(n-1)$. Further, $\ell_i(x_j) = 0$, $j \neq i$, and therefore

$$\int_a^b (\ell_i(x))^2 w(x)\,dx = w_i(\ell_i(x_i))^2 = w_i.$$

Since $w(x) > 0$ it follows that $w_i > 0$. □

Gaussian quadrature formulas can also be derived by Hermite interpolation on the nodes x_k, *each counted as a double node* and requiring that coefficients of the derivative terms should be zero. This interpretation gives a convenient expression for the error term in Gaussian quadrature.

Theorem 5.3.4.

The remainder term in Gauss' quadrature rule (5.3.12) *with n nodes is given by the formula*

$$I[f] - I_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!}\int_a^b \left[\prod_{i=1}^n (x - x_i)\right]^2 w(x)\,dx = c_n f^{(2n)}(\xi), \quad a < \xi < b. \tag{5.3.13}$$

The constant c_n can be determined by applying the formula to some polynomial of degree $2n$.

Proof. Denote by $q(x)$ the polynomial of degree $2n-1$ which solves the Hermite interpolation problem (see Sec. 4.3.1)

$$q(x_i) = f(x_i), \qquad q'(x_i) = f'(x_i), \quad i = 1:n.$$

The Gauss quadrature formula is exact for $q(x)$, and hence

$$\int_a^b q(x)w(x)\,dx = \sum_{i=1}^n w_i q(x_i) = \sum_{i=1}^n w_i f(x_i).$$

Thus

$$\sum_{i=1}^n w_i f(x_i) - \int_a^b f(x)w(x)\,dx = \int_a^b (q(x) - f(x))w(x)\,dx.$$

Using the remainder term (4.3.4) in Hermite interpolation gives

$$f(x) - q(x) = \frac{f^{(2n)}(\xi)}{(2n)!}(\varphi_n(x))^2, \quad \varphi_n(x) = \prod_{i=1}^n (x - x_i),$$

and the theorem now follows. □

Using Bernštein's approximation theorem (Theorem 3.2.5) we get the following corollary.

Corollary 5.3.5.

Let f real-valued for $z \in [-1, 1]$, and analytic and single-valued $|f(z)| \leq M$ in the region $z \in \mathcal{E}_R$, $R > 1$, where

$$\mathcal{E}_R = \{z : |z-1| + |z+1| \leq R + R^{-1}\},$$

be an ellipse with foci at 1 *and* -1. *Then the remainder term in a Gauss quadrature rule with n nodes for the interval* $[-1, 1]$ *satisfies*

$$|I[f] - I_n(f)| \leq \frac{2M\mu_0}{1 - 1/R} R^{-2n}. \tag{5.3.14}$$

This shows the rapid convergence of Gauss' quadrature rules for functions analytic in a region $\mathcal{E}_R$, with $R \gg 1$.

We now mention some classical Gauss–Christoffel quadrature rules, which are related to the orthogonal polynomials surveyed in Sec. 4.5.5. For an integral $\int_{-1}^{1} f(x)\,dx$, with uniform weight distribution $w(x) = 1$, the relevant orthogonal polynomials are the Legendre polynomials $P_n(x)$.

As a historical aside, Gauss derived his quadrature formula by considering the continued fraction

$$\frac{1}{2}\int_{-1}^{1} \frac{dx}{z-x} = \frac{1}{2}\ln\left(\frac{z+1}{z-1}\right) = \frac{1}{z-}\ \frac{1/3}{z-}\ \frac{4/(3\cdot 5)}{z-}\ \frac{9/(5\cdot 7)}{z-}\ \cdots, \tag{5.3.15}$$

which he had derived in an earlier paper. The nth convergent of this continued fraction is a rational function with a numerator of degree $n-1$ in z and denominator of degree n which is the $(n-1, n)$ Padé approximant to the function. Decomposing this fraction in partial fractions the residues and the poles can be taken as nodes of a quadrature formula. Using the accuracy properties of the Padé approximants Gauss showed that the quadrature formula will have order $2n-1$.

The reciprocal of the denominators' polynomials $P_n(z) = z^n Q_n(1/z)$ are precisely the Legendre polynomials; see Example 3.5.6. Recall that the monic Legendre polynomials satisfy the recurrence formula $P_0 = 1$, $P_1 = x$,

$$P_{n+1}(x) = x\,P_n(x) - \frac{n^2}{4n^2-1} P_{n-1}(x), \quad n \geq 1.$$

The first few monic Legendre polynomials are

$$P_2(x) = \frac{1}{3}(3x^2 - 1), \qquad P_3(x) = \frac{1}{5}(5x^3 - 3x),$$
$$P_4(x) = \frac{1}{35}(35x^4 - 30x^2 + 3), \qquad P_5(x) = \frac{1}{63}(63x^5 - 70x^3 + 15x), \ldots.$$

Example 5.3.2.

For a two-point Gauss–Legendre quadrature rule the two abscissae are the zeros of $P_2(x) = \frac{1}{3}(3x^2 - 1)$, i.e., $\pm 3^{-1/2}$. Note that they are symmetric with respect to the origin.

The weights can be determined by application of the formula to $f(x) = 1$ and $f(x) = x$, respectively. This gives

$$w_0 + w_1 = 2, \qquad -3^{-1/2} w_0 + 3^{-1/2} w_1 = 0,$$

with solution $w_0 = w_1 = 1$. Hence the formula

$$\int_{-1}^{1} f(x)\,dx \approx f(-3^{-1/2}) + f(3^{-1/2})$$

is exact for polynomials of degree ≤ 3. For a three-point Gauss formula, see Problem 5.3.1.

Abscissae and weights for Gauss formulas using $n = m + 1$ points, for $n = 2 : 10$, with 15 decimal digits and $n = 12, 16, 20, 24, 32, 40, 48, 64, 80$, and 96 with 20 digits are tabulated in [1, Table 25.4]; see Table 5.3.1 for a sample. Instead of storing these constants, it might be preferable to use a program that generates abscissae and weights as needed.

Table 5.3.1. *Abscissae and weight factors for some Gauss–Legendre quadrature from* [1, Table 25.4].

x_i	w_i
$n = 3$	
0.00000 00000 00000	0.88888 88888 88889
±0.77459 66692 41483	0.55555 55555 55556
$n = 4$	
±0.33998 10435 84856	0.65214 51548 62546
±0.86113 63115 94053	0.34785 48451 37454
$n = 5$	
0.00000 00000 00000	0.56888 88888 88889
±0.53846 93101 05683	0.47862 86704 99366
±0.90617 98459 38664	0.23692 68850 56189

For the weight function

$$w(x) = (1-x)^{\alpha}(1+x)^{\beta}, \quad x \in [-1, 1], \quad \alpha, \beta > -1,$$

the nodes are obtained from the zeros of the Jacobi polynomials $J_n(x; \alpha, \beta)$. In the special case when $\alpha = \beta = 0$ these equal the Legendre polynomials. The case $\alpha = \beta = -1/2$, which corresponds to the weight function $w(x) = 1/\sqrt{1-x^2}$, gives the Chebyshev polynomials $T_n(x)$ of the first kind. Similarly, $\alpha = \beta = 1/2$ gives the Chebyshev polynomials $U_n(x)$ of the second kind.

If a quadrature rule is given for the standard interval $[-1, 1]$, the corresponding formula for an integral over the interval $[a, b]$ is obtained by the change of variable $t = \frac{1}{2}((b-a)x + (a+b))$, which maps the interval $[a, b]$ onto the standard interval $[-1, 1]$:

$$\int_a^b f(t)dt = \frac{b-a}{2}\int_{-1}^{1} g(x)\,dx, \qquad g(x) = f\left(\frac{1}{2}\big((b-a)x + (a+b)\big)\right).$$

If $f(t)$ is a polynomial, then $g(x)$ will be a polynomial of the same degree, since the transformation is linear. Hence the order of accuracy of the formula is not affected.

Two other important cases of Gauss quadrature rules deal with infinite intervals of integration. The generalized Laguerre polynomials $L_n^{(\alpha)}(x)$ are orthogonal with respect to the weight function

$$w(x) = x^{\alpha}e^{-x}, \quad x \in [0, \infty], \quad \alpha > -1.$$

Setting $\alpha = 0$, we get the Laguerre polynomials $L_n^{(0)}(x) = L_n(x)$.

The Hermite polynomials are orthogonal with respect to the weight function

$$w(x) = e^{-x^2}, \quad -\infty < x < \infty.$$

Recall that weight functions and recurrence coefficients for the above monic orthogonal polynomials are given in Table 4.5.1.

Rather little is found in the literature on numerical analysis about densities on infinite intervals, except the classical cases above. It follows from two classical theorems of Hamburger in 1919 and M. Riesz in 1923 that the system of orthogonal polynomials for the density w over the infinite interval $[-\infty, \infty]$ is complete if, for some $\beta > 0$,

$$\int_{-\infty}^{\infty} e^{\beta|x|} w(x)\, dx < \infty;$$

see Freud [123, Sec. II.4–5]. For densities on $[0, \infty]$, x is to be replaced by $\sqrt{x}$ in the above result. (Note that a density function on the positive real x-axis can be mapped into an even density function on the whole real t-axis by the substitution $x = t^2$.

5.3.3 Gauss Quadrature with Preassigned Nodes

In many applications it is desirable to use Gauss-type quadrature where some nodes are preassigned and the rest chosen to maximize the order of accuracy. In the most common cases the preassigned nodes are at the endpoints of the interval. Consider a quadrature rule of the form

$$\int_a^b f(x)w(x)\, dx = \sum_{i=1}^{n} w_i f(x_i) + \sum_{j=1}^{m} b_j f(z_j) + R(f), \tag{5.3.16}$$

where z_j, $j = 1 : m$, are fixed nodes in $[a, b]$ and the x_i are determined so that the interpolatory rule is exact for polynomials of order $2n + m - 1$. By a generalization of Theorem 5.3.4 the remainder term is given by the formula

$$R(f) = \frac{f^{(2n+m)}(\xi)}{(2n)!} \int_a^b \prod_{i=1}^{m}(x - z_i)\Big[\prod_{i=1}^{n}(x - x_i)\Big]^2 w(x)\, dx, \quad a < \xi < b. \tag{5.3.17}$$

In **Gauss–Lobatto** quadrature both endpoints are used as abscissae, $z_1 = a$, $z_2 = b$, and $m = 2$. For the standard interval $[a, b] = [-1, 1]$ and the weight function $w(x) = 1$, the quadrature formula has the form

$$\int_{-1}^{1} f(x)\, dx = w_0 f(-1) + w_{n+1} f(1) + \sum_{i=1}^{n} w_i f(x_i) + E_L. \tag{5.3.18}$$

The abscissae $a < x_i < b$ are the zeros of the orthogonal polynomial ϕ_n corresponding to the weight function $\tilde{w}(x) = (1 - x^2)$, i.e., up to a constant factor equal to the Jacobi polynomial $J_n(x, 1, 1) = P'_{n+1}(x)$. The nodes lie symmetric with respect to the origin. The corresponding weights satisfy $w_i = w_{n+1-i}$, and are given by

$$w_0 = w_{n+1} = \frac{2}{(n+2)(n+1)}, \qquad w_i = \frac{w_0}{(P_{n+1}(x_i))^2}, \quad i = 1 : n. \tag{5.3.19}$$

The Lobatto rule (5.3.18) is exact for polynomials of order $2n+1$, and for $f(x) \in C^{2m}[-1, 1]$ the error term is given by

$$R(f) = -\frac{(n+2)(n+1)^3 2^{2n+3}(n!)^4}{(2n+3)[(2n+2)!]^3} f^{(2n+2)}(\xi), \quad \xi \in (-1, 1). \tag{5.3.20}$$

Nodes and weights for Lobatto quadrature are found in [1, Table 25.6].

In **Gauss–Radau** quadrature rules $m = 1$ and one of the endpoints is taken as the abscissa, $z_1 = a$ or $z_1 = b$. The remainder term (5.3.17) becomes

$$R(f) = \frac{f^{(2n+1)}(\xi)}{(2n)!} \int_a^b (x - z_1) \left[\prod_{i=1}^{n} (x - x_i) \right]^2 w(x)\, dx, \quad a < \xi < b. \tag{5.3.21}$$

Therefore, if the derivative $f^{(n+1)}(x)$ has constant sign in $[a, b]$, then the error in the Gauss–Radau rule with $z_1 = b$ will have opposite sign to the Gauss–Radau rule with $z_1 = a$. *Thus, by evaluating both rules we obtain lower and upper bounds for the true integral.* This has many applications; see Golub [160].

For the standard interval $[-1, 1]$ the Gauss–Radau quadrature formula with $z_1 = 1$ has the form

$$\int_{-1}^{1} f(x)\, dx = w_0 f(-1) + \sum_{i=1}^{n} w_i f(x_i) + E_{R1}. \tag{5.3.22}$$

The n free abscissae are the zeros of

$$\frac{P_n(x) + P_{n+1}(x)}{x - 1},$$

where $P_m(x)$ are the Legendre polynomials. The corresponding weights are given by

$$w_0 = \frac{2}{(n+1)^2}, \qquad w_i = \frac{1}{(n+1)^2} \frac{1 - x_i}{(P_n(x_i))^2}, \quad i = 1 : n. \tag{5.3.23}$$

The Gauss–Radau quadrature rule is exact for polynomials of order $2n$. If $f(x) \in C^{2m-1}[-1, 1]$, then the error term is given by

$$E_{R1}(f) = \frac{(n+1)2^{2n+1}}{[(2n+1)!]^3} (n!)^4 f^{(2n+1)}(\xi_1), \quad \xi_1 \in (-1, 1). \tag{5.3.24}$$

A similar formula can be obtained with the fixed point $+1$ by making the substitution $t = -x$.

By modifying the proof of Theorem 5.3.3 it can be shown that the weights in Gauss–Radau and Gauss–Lobatto quadrature rules are positive if the weight function $w(x)$ is nonnegative.

Example 5.3.3.

The simplest Gauss–Lobatto rule is Simpson's rule with $n = 1$ interior node. Taking $n = 2$ the interior nodes are the zeros of $\phi_2(x)$, where

$$\int_{-1}^{1} (1 - x^2)\phi_2(x)\, p(x)\, dx = 0 \quad \forall p \in P_2.$$

Thus, ϕ_2 is, up to a constant factor, the Jacobi polynomial $J_2(x, 1, 1) = (x^2 - 1/5)$. Hence the interior nodes are $\pm 1/\sqrt{5}$ and by symmetry the quadrature formula is

$$\int_{-1}^{1} f(x)\, dx = w_0(f(-1) + f(1)) + w_1(f(-1/\sqrt{5}) + f(1/\sqrt{5})) + R(f), \quad (5.3.25)$$

where $R(f) = 0$ for $f \in P_6$. The weights are determined by exactness for $f(x) = 1$ and $f(x) = x^2$. This gives $2w_0 + 2w_1 = 2$, $2w_0 + (2/5)w_1 = \frac{2}{3}$, i.e., $w_0 = \frac{1}{6}$, $w_1 = \frac{5}{6}$.

A serious drawback with Gaussian rules is that as we increase the order of the formula, *all interior abscissae change*, except that at the origin. Thus function values computed for the lower-order formula are not used in the new formula. This is in contrast to Romberg's method and Clenshaw–Curtis quadrature rules, where *all old function values* are used also in the new rule when the number of points is doubled.

Let G_n be an n-point Gaussian quadrature rule

$$\int_a^b f(x)w(x)\, dx \approx \sum_{i=0}^{n-1} a_i f(x_i),$$

where x_i, $i = 0 : n - 1$, are the zeros of the nth degree orthogonal polynomial $\pi_n(x)$. Kronrod [227, 228] considered extending G_n by finding a new quadrature rule

$$K_{2n+1} = \sum_{i=0}^{n-1} a_i f(x_i) + \sum_{i=0}^{n} b_i f(y_i), \quad (5.3.26)$$

where the new $n + 1$ abscissae y_i are chosen such that the degree of the rule K_{2n+1} is equal to $3n + 1$. The new nodes y_i should then be selected as the zeros of a polynomial $p_{n+1}(x)$ of degree $n + 1$, satisfying the orthogonality conditions

$$\int_a^b \pi_n(x)\, p_{n+1}(x) w(x)\, dx = 0. \quad (5.3.27)$$

If the zeros are real and contained in the closed interval of integration $[a, b]$ such a rule is called a **Kronrod extension** of the Gaussian rule. The two rules (G_n, K_{2n+1}) are called a **Gauss–Kronrod** pair. Note that the number of new function evaluations are the same as for the Gauss rule G_{n+1}.

It has been proved that a Kronrod extension exists for the weight function $w(x) = (1 - x^2)^{\lambda - 1/2}$, $\lambda \in [0, 2]$, and $[a, b] = [-1, 1]$. For this weight function the new nodes interlace the original Gaussian nodes, i.e.,

$$-1 \le y_0 < x_0 < y_1 < x_1 < y_2 < \cdots < x_{n-1} < y_n < 1.$$

This interlacing property can be shown to imply that all weights are positive. Kronrod considered extensions of Gauss–Legendre rules, i.e., $w(x) = 1$, and gives nodes and weights in [228] for $n \leq 40$.

It is not always the case that all weights are positive. For example, it has been shown that Kronrod extensions of Gauss–Laguerre and Gauss–Hermite quadrature rules with positive weights do not exist when $n > 0$ in the Laguerre case and $n = 3$ and $n > 4$ in the Hermite case. On the other hand, the Kronrod extensions of Gauss–Legendre rules can be shown to exist and have positive weights.

Gauss–Kronrod rules are one of most effective methods for calculating integrals. Often one takes $n = 7$ and uses the Gauss–Kronrod pair (G_7, K_{15}), together with the realistic but still conservative error estimate $(200|G_n - K_{2n+1}|)^{1.5}$; see Kahaner, Moler, and Nash [215, Sec. 5.5].

Kronrod extension of Gauss–Radau and Gauss–Lobatto rules can also be constructed. Kronrod extension of the Lobatto rule (5.3.25) is given by Gander and Gautschi [128] and used in an adaptive Lobatto quadrature algorithm. The simplest extension is the four-point Lobatto–Kronrod rule

$$\int_{-1}^{1} f(x)\,dx = \frac{11}{210}(f(-1)+f(1)) + \frac{72}{245}\left(f\left(-\sqrt{\frac{2}{3}}\right)+f\left(\sqrt{\frac{2}{3}}\right)\right) + \frac{125}{294}\left(f\left(-\frac{1}{\sqrt{5}}\right)+f\left(\frac{1}{\sqrt{5}}\right)\right)+\frac{16}{35}f(0)) + R(f). \tag{5.3.28}$$

This rule is exact for all $f \in \mathcal{P}_{10}$. Note that the Kronrod points $\pm\sqrt{2/3}$ and 0 interlace the previous nodes.

5.3.4 Matrices, Moments, and Gauss Quadrature

We first collect some classical results of Gauss, Christoffel, Chebyshev, Stieltjes, and others, with a few modern aspects and notations appropriate for our purpose.

Let $\{p_1, p_2, \ldots, p_n\}$, where p_j is of exact degree $j-1$, be a basis for the space $\mathcal{P}_n$ of polynomials of degree $n-1$. We introduce the row vector

$$\pi(x) = [p_1(x),\ p_1(x), \ldots, p_n(x)] \tag{5.3.29}$$

containing these basis functions. The **modified moments** with respect to the basis $\pi(x)$ are

$$\nu_k = (p_k, 1) = \int_a^b p_k(x)w(x)\,dx, \quad k = 1:n. \tag{5.3.30}$$

We define the two symmetric matrices

$$G = \int \pi(x)^T\pi(x)w(x)\,dx, \qquad \hat{G} = \int x\pi(x)^T\pi(x)w(x)\,dx \tag{5.3.31}$$

associated with the basis defined by π. These have elements

$$g_{ij} = (p_i, p_j) = (p_j, p_i), \qquad \hat{g}_{ij} = (xp_i, p_j) = (xp_j, p_i),$$

respectively. Here

$$G = \begin{pmatrix} (p_1, p_1) & (p_1, p_2) & \dots & (p_1, p_n) \\ (p_2, p_1) & (p_2, p_2) & \dots & (p_2, p_n) \\ \vdots & \vdots & \ddots & \vdots \\ (p_n, p_1) & (p_n, p_2) & \dots & (p_n, p_n) \end{pmatrix} \tag{5.3.32}$$

is called the **Gram matrix**.

In particular, for the power basis

$$\theta(x)(1, x, x^2, \dots, x^{n-1}) \tag{5.3.33}$$

we have $g_{ij} = (x^{i-1}, x^{j-1}) = \mu_{i+j-2}$, where $\mu_k = (x^k, 1) = \int_a^b x^k w(x)\,dx$ are the ordinary moments. In this case the matrices G and $\hat{G}$ are the Hankel matrices,

$$G = \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & \cdots & \vdots \\ \mu_{n-1} & \mu_n & \cdots & \mu_{2n-2} \end{pmatrix}, \qquad \hat{G} = \begin{pmatrix} \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & \cdots & \vdots \\ \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \end{pmatrix}.$$

In particular, for $w(x) \equiv 1$ and $[a, b] = [0, 1]$ we have $\mu_k = \int_0^1 x^{k-1}\,dx = 1/k$, and G is the notoriously ill-conditioned Hilbert matrix.

Let u and v be two polynomials in $\mathcal{P}_n$ and set

$$u(x) = \pi(x)u_\pi, \qquad v(x) = \pi(x)v_\pi,$$

where u_π, v_π are column vectors with the coefficients in the representation of u and v with respect to the basis defined by $\pi(x)$. Then

$$(u, v) = \int_a^b u_\pi^T \pi(x)^T \pi(x) v_\pi w(x)\,dx = u_\pi^T G v_\pi.$$

For $u = v \neq 0$ we find that $u_\pi^T G u_\pi = (u, u) > 0$, i.e., the Gram matrix G is positive definite. (The matrix $\hat{G}$ is, however, usually indefinite.)

A polynomial of degree n that is orthogonal to all polynomials of degree less than n can be written in the form

$$\phi_{n+1}(x) = xp_n(x) - \pi(x)c_n, \quad c_n \in \mathbf{R}^n. \tag{5.3.34}$$

Here c_n is determined by the linear equations

$$0 = (\pi(x)^T, \phi_{n+1}(x)) = (\pi(x)^T, xp_n(x)) - (\pi(x)^T, \pi(x))c_n,$$

or in matrix form

$$Gc_n = \hat{g}_n, \tag{5.3.35}$$

where $\hat{g}_n = \hat{G}e_n$ is the last column of the matrix $\hat{G}$. Further, there are coefficients $c_{k,j}$ depending on the basis only such that

$$xp_j(x) = \sum_{k=1}^{j+1} c_{k,j} p_k(x), \quad j = 1 : n-1.$$

Together with (5.3.34) this can be summarized in the (row) vector equation

$$x\pi(x) = \pi(x)\overline{C} + \phi_{n+1}(x)e_n^T, \quad \overline{C} = (C,\ c_n). \tag{5.3.36}$$

Here $e_n^T = (0, 0, \ldots, 1)$ and $C = (c_{k,j}) \in \mathbf{R}^{n\times(n-1]}$ is an upper Hessenberg matrix. Note that C depends on the basis only, while c_n also depends on the weight function.

For the power basis $p_j(x) = x^{j-1}$, the matrix C is a **shift matrix**; the only nonzero elements are ones in the first main subdiagonal. If the basis is some family of orthogonal polynomials (possibly with respect to weight function other than w) C is a tridiagonal matrix, obtained by means of the three-term recurrence relation for this family.

After multiplication of (5.3.36) by $\pi(x)^T w(x)$ and integration we obtain by (5.3.31)

$$G\overline{C} = \hat{G}, \tag{5.3.37}$$

where the last column of this matrix equation is the same as (5.3.35). Let G^*, C^* be defined like G, C, with n increased by one. Note that G and C are principal submatrices of G^* and C^*. Then $\hat{G}$ equals the n first rows of the product G^*C^*. Thus, no integrations are needed for g_n, except for the Gram matrix G.

Theorem 5.3.6.

Denote by R the matrix of coefficients of the expansions of the general basis functions $\pi(x) = [p_1(x),\ p_1(x), \ldots, p_n(x)]$ into the orthonormal basis polynomials with respect to the weight function w, i.e.,

$$\pi(x) = \varphi(x)R, \quad \varphi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_n(x)). \tag{5.3.38}$$

(Conversely, the coefficients of the expansions of the orthogonal polynomials into the original basis functions are found in the columns of R^{-1}.) Then $G = R^T R$, i.e., R is the upper triangular Cholesky factor of the Gram matrix G. Note that up to the mth row this factorization is the same for all $n \geq m$. Further, $\hat{G} = R^T J R$, where J is a symmetric tridiagonal matrix.

Proof. R is evidently an upper triangular matrix. Further, we have

$$\begin{aligned} G = \int \pi(x)^T \pi(x) w(x)\, dx &= \int R^T \varphi(x)^T \varphi(x) R w(x)\, dx \\ &= R^T I R = R^T R, \end{aligned}$$

since the elements of $\varphi(x)$ are an orthonormal system. This shows that R is the Cholesky factor of G. We similarly find that

$$\hat{G} = R^T J R, \quad J = \int x\varphi(x)^T \varphi(x) w(x)\, dx,$$

and thus J clearly is a symmetrical matrix. J is a particular case of $\hat{G}$ and from (5.3.37) and $G = I$ it follows that $J = \overline{C}$, a Hessenberg matrix. Hence J is a symmetric tridiagonal matrix. □

From (5.3.37) and Theorem 5.3.6 it follows that

$$\hat{G} = R^T J R = G\overline{C} = R^T R\overline{C}.$$

Since R is nonsingular we have $R\overline{C} = JR$, or

$$J = R\overline{C}R^{-1}. \tag{5.3.39}$$

This shows that the spectrum of $\overline{C}$ equals the spectrum of J, for every choice of basis. We shall see that it is equal to the set of zeros of the orthogonal polynomial ϕ_{n+1}. For the power basis $p_j(x) = x^{j-1}$ (5.3.34) reads

$$\phi_{n+1}(x) = x^n - \sum_{k=1}^{n} c_{n,k} x^{k-1},$$

and hence

$$\overline{C} = \begin{pmatrix} 0 & & & & c_{n,1} \\ 1 & 0 & & & c_{n,2} \\ & 1 & \ddots & & \vdots \\ & & \ddots & 0 & c_{n,n-1} \\ & & & 1 & c_{n,n} \end{pmatrix} \in \mathbf{R}^{n\times n}.$$

This is the companion matrix of $\phi_{n+1}(x)$, and it can be shown that (see Sec. 6.5.2)

$$\det(zI - \overline{C}) = \phi_{n+1}(x). \tag{5.3.40}$$

Thus the eigenvalues λ_j, $j = 1 : n$, of $\overline{C}$ are the zeros of $\phi_{n+1}(x)$, and hence the nodes for the Gauss–Christoffel quadrature formula.

It can be verified that the row eigenvector of $\overline{G}$ corresponding to λ_j is

$$\theta(\lambda_j) = (1, \lambda_j, \lambda_j^2, \ldots, \lambda_j^{n-1}); \tag{5.3.41}$$

i.e., it holds that

$$\theta(\lambda_j)\overline{C} = \lambda_j \theta(\lambda_j), \quad j = 1 : n. \tag{5.3.42}$$

This yields a diagonalization of $\overline{C}$, since, by the general theory of orthogonal polynomials (see Theorem 5.3.3), the roots are simple roots, located in the interior of the smallest interval that contains the weight distribution.

To summarize, we have shown that if C and the Gram matrix G are known, then c_n can be computed by performing the Cholesky decomposition $G = R^T R$ and then solving $R^T R c_n = \hat{g}_n$ for c_n. The zeros of $\phi_{n+1}(x)$ are then equal to the eigenvalues of $\overline{C} = (C, c_n)$ or, equivalently, the eigenvalues of the symmetric tridiagonal matrix $J = R\overline{C}R^{-1}$. *This is true for any basis $\pi(x)$.* Note that J can be computed by solving the matrix equation $JR = R\overline{C}$ or

$$R^T J = (R\overline{C})^T. \tag{5.3.43}$$

Here R^T is a lower triangular matrix and the right-hand side a lower Hessenberg matrix. This and the tridiagonal structure of J considerably simplifies the calculation of J. In the next section we show how the theory developed here leads to a stable and efficient algorithm for computing Gauss quadrature rules.

5.3.5 Jacobi Matrices and Gauss Quadrature

The computations are most straightforward for the power basis, $\theta(x)$, using the moments of the weight function as the initial data. But the condition number of the Gram matrix G, which in this case is a Hankel matrix, increases rapidly with n. This is related to the by now familiar fact that, when n is large, x^n can be accurately approximated by a polynomial of lower degree. Thus the moments for the power basis are not generally a good starting point for the numerical computation of the matrix J.

For the orthonormal basis $\varphi(x)$, we have $G = I$, and

$$\overline{C} = \hat{G} = J = \begin{pmatrix} \beta_1 & \gamma_1 & & & 0 \\ \gamma_1 & \beta_2 & \gamma_2 & & \\ & \gamma_2 & & \ddots & \\ & & \ddots & \ddots & \gamma_{n-1} \\ 0 & & & \gamma_{n-1} & \beta_n \end{pmatrix} \tag{5.3.44}$$

is a symmetric tridiagonal matrix with nonzero off-diagonal elements. Such a tridiagonal matrix is called a **Jacobi matrix** and has n real distinct eigenvalues λ_j. The row eigenvectors $\varphi(\lambda_j)$ satisfy

$$\varphi(\lambda_j)J = \lambda_j\varphi(\lambda_j), \quad j = 1:n, \tag{5.3.45}$$

and are mutually orthogonal. Setting

$$\Phi = (\varphi(\lambda_1)^T, \ldots, \varphi(\lambda_n)^T), \qquad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n),$$

we obtain by (5.3.45) and the symmetry of J the important matrix formula

$$J\Phi = \Phi\Lambda. \tag{5.3.46}$$

It also follows from (5.3.36) that for all x

$$x\varphi(x) = J\varphi(x) + \gamma_n\phi_{n+1}(x)e_n^T, \tag{5.3.47}$$

where γ_n is to be chosen so that $\|\phi_{n+1}\| = 1$. The last column of this equation gives

$$(x - \beta_n)\phi_n(x) = \gamma_{n-1}\phi_{n-1}(x) + \gamma_n\phi_{n+1}(x), \tag{5.3.48}$$

which is the three-term recurrence relation (4.5.36) for orthogonal polynomials.

Let V be an orthogonal matrix that diagonalizes J, i.e.,

$$JV = V\Lambda, \quad V^TV = VV^T = I,$$

where Λ is the diagonal in (5.3.46). It follows that $V = \Phi D$ for some diagonal matrix $D = \operatorname{diag}(d_i)$, and

$$VV^T = \Phi D^2\Phi^T = I,$$

i.e.,

$$\sum_{k=1}^{n} d_k^2\phi_i(\lambda_k)\phi_j(\lambda_k) = \delta_{ij} = (\phi_i, \phi_j), \quad i, j = 1:n.$$

This equality holds also for $i = n+1$, because $\phi_{n+1}(\lambda_k) = 0$, for all k, and $(\phi_{n+1}, \phi_j) = 0$, $j = 1:k$.

Since every polynomial p of degree less than $2n$ can be expressed as a linear combination of polynomials of the form $\phi_i\phi_j$ (in infinitely many ways) it follows that

$$\sum_{k=1}^{n} d_k^2 p(\lambda_k) = \int p(x)w(x)\,dx, \tag{5.3.49}$$

for any polynomial p of degree less than $2n$. This yields the **Gauss–Christoffel quadrature rule**:

$$\int f(x)w(x)\,dx = \sum_{k=1}^{n} d_k^2 f(\lambda_k) + R, \tag{5.3.50}$$

where

$$R = \int (f(x) - p(x))w(x)\,dx,$$

for any polynomial p of degree less than $2n$ such that $p(\lambda_k) = f(\lambda_k)$, $k = 1:n$.

The familiar form for the remainder term

$$R = k_n f^{(2n)}(\xi)/(2n)! \tag{5.3.51}$$

is obtained by choosing a Hermite interpolation polynomial for p and then applying the mean value theorem. The constant k_n is independent of f. The choice $f(x) = A_n^2 x^{2n} + \cdots$ gives $k_n = A_n^{-2}$. A recurrence relation for the leading coefficient A_j is obtained by (5.3.48). We obtain

$$A_0 = \mu_0^{-1/2}, \qquad A_{k+1} = A_k/\gamma_k. \tag{5.3.52}$$

The mean value form for R may be inappropriate when the interval is infinite. Some other estimate of the above integral for R may then be more adequate, for example, in terms of the best approximation of f by a polynomial in some weighted L_p-norm.

A simple formula for the weights d_k^2, due to Golub and Welsch, is obtained by matching the first rows of the equality $V = \Phi D$. Since the elements in the first row of Φ are all equal to the constant $\phi_1 = \mu_0^{-1/2}$, we obtain

$$e_1^T V = \mu_0^{-1/2} d^T, \qquad d_k^2 = \mu_0 v_{1,k}^2, \quad k = 1:n. \tag{5.3.53}$$

The well-known fact that the weights are positive and their sum equals μ_0 follows immediately from this simple formula for the weights. We summarize these results in the following theorem.

When the three-term recurrence relation for the orthonormal polynomials associated with the weight function $w(x)$ is known, or can be computed by the Stieltjes procedure in Sec. 4.5.5, the Gauss–Christoffel rule can be obtained elegantly as follows. The nodes of the Gauss–Christoffel rule are the eigenvalues of the tridiagonal matrix J, and by (5.3.53) the weights equal the square of the first components of the corresponding eigenvectors. These quantities can be computed in a stable and efficient way by the QR algorithm; see Volume II. In [159] this scheme is extended to the computation of nodes and weights for Gauss–Radau and Gauss–Lobatto quadrature rules.

When the coefficients in the three-term relation cannot be obtained by theoretical analysis or numerical computation, we consider the matrices $\overline{C}$ and $G = R^T R$ as given data about the basis and weight function. Then J can be computed by means of (5.3.39) and the nodes and weights are computed according to the previous case. Note that R and J can be determined simultaneously for all $k \leq n$; just take the submatrices of the largest ones.

The following concise and applicable result was found independently by Golub and Meurant (see [162, Theorem 3.4]) and the first-named author (see [86, Theorem 2.2]).

Theorem 5.3.7.

Let J be the symmetric tridiagonal $n \times n$ matrix that contains the coefficients in the three-term recurrence relation for the orthogonal polynomials associated with a positive weight function $w(x)$ (with any sequence of leading coefficients). Let $e_1 = (1, 0, 0, \ldots, 0)^T$ and f be an analytic function in a domain that contains the spectrum of J.

Then the formula

$$\frac{1}{\mu_0} \int f(x) w(x)\, dx \approx e_1^T f(J) e_1 \tag{5.3.54}$$

is exact when f is a polynomial of degree less than $2n$.

Proof. If $J = V \Lambda V^T$ is the spectral decomposition of J, then we have

$$f(J) = V^T \operatorname{diag}\left(f(\lambda_1), \ldots, f(\lambda_n)\right) V.$$

Let p be a polynomial of degree less than $2n$. We obtain using (5.3.53)

$$e_1^T V \Lambda V^T e_1^T = \mu_0^{-1/2} d^T p(\Lambda) \mu_0^{-1/2} d = \mu_0^{-1} \sum_{j=1}^{n} p(\lambda_j) d_j^2 = \mu_0^{-1} \int p(x) w(x)\, dx,$$

since Gauss–Christoffel quadrature is exact for p. □

If $f(J)$ is evaluated by means of the diagonalization of J, (5.3.54) becomes exactly the Gauss–Christoffel rule, but it is noteworthy that $e_1^T V^T f(\Lambda) V e_1$ can sometimes be evaluated without a diagonalization of J. The accuracy of the estimate of the integral still depends on how well $f(z)$ can be approximated by a polynomial of degree less than twice the size of J in the weighted L_1-norm with weight function $w(x)$.

In many important cases the weight function $w(x)$ is *symmetric about the origin*. Then the moments of odd order are zero, and the orthogonal polynomials of odd (even) degree are odd (even) functions. By Theorem 4.5.18 the coefficients $\beta_k = 0$ for all k, i.e., *the matrix J will have a zero diagonal*. The eigenvalues of J will then appear in pairs, $\pm\lambda_k$. If n is odd, there is also a simple zero eigenvalue. The weights are symmetric so that the weights corresponding to the two eigenvalues $\pm\lambda_i$ are the same.

We shall see that in the symmetric case the eigenvalue problem for the tridiagonal matrix $J \in \mathbf{R}^{n\times n}$ can be reduced to a singular value problem for a smaller bidiagonal matrix B, where

$$B \in \begin{cases} \mathbf{R}^{n/2 \times n/2} & \text{if } n \text{ even}, \\ \mathbf{R}^{(n+1)/2 \times (n-1)/2} & \text{if } n \text{ odd}. \end{cases}$$

We permute rows and columns in J, by an odd-even permutation; for example, if $n = 7$, then $(1, 2, 3, 4, 5, 6, 7) \mapsto (1, 3, 5, 7, 2, 4, 6)$, and

$$\tilde{J} = T^{-1}JT = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \gamma_1 & 0 & 0 \\ \gamma_2 & \gamma_3 & 0 \\ 0 & \gamma_4 & \gamma_5 \\ 0 & 0 & \gamma_6 \end{pmatrix},$$

where T is the permutation matrix effecting the permutation. Then, J and $\tilde{J}$ have the same eigenvalues. If the orthogonal matrix V diagonalizes J, i.e., $J = V\Lambda V^T$, then $\tilde{V} = T^{-1}V$ diagonalizes $\tilde{J} = T^T JT$, i.e., $\tilde{J} = T^{-1}JT = T^{-1}V\lambda V^T T$. Note that the first row of V is just a permutation of $\tilde{V}$. We can therefore substitute $\tilde{V}$ for V in equation (5.3.53), which gives the weights in the Gauss–Christoffel formula.

The following relationship between the singular value decomposition (SVD) and a Hermitian eigenvalue problem, exploited by Lanczos [231, Chap. 3], can easily be verified.

Theorem 5.3.8.

Let the SVD of $B \in \mathbf{R}^{m\times n}$ $(m \geq n)$ be $B = P\Sigma Q^T$, where

$$\Sigma = \operatorname{diag}(\Sigma_1,\ 0), \quad \Sigma_1 = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n),$$

and

$$P = (P_1,\ P_2) \in \mathbf{C}^{m\times m}, \quad P_1 \in \mathbf{C}^{m\times n}, \quad Q \in \mathbf{C}^{n\times n}.$$

Then the symmetric matrix $C \in \mathbf{R}^{(m+n)\times(m+n)}$ has the eigendecomposition

$$C = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix} = V \begin{pmatrix} \Sigma_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\Sigma_1 \end{pmatrix} V^T, \tag{5.3.55}$$

where $V \in$ is orthogonal:

$$V = \frac{1}{\sqrt{2}} \begin{pmatrix} P_1 & \sqrt{2}\,P_2 & P_1 \\ Q & 0 & -Q \end{pmatrix}^T. \tag{5.3.56}$$

Hence the eigenvalues of C are $\pm\sigma_1, \pm\sigma_2, \ldots, \pm\sigma_r$, and zero repeated $(m - n)$ times.

The QR algorithm for symmetric tridiagonal matrices can be adopted to compute the singular values σ_i and the first components of the matrix $P = (P_1, P_2)$ of left singular vectors of the bidiagonal matrix B; see Volume II.

Example 5.3.4.

The monic Legendre polynomials are symmetric around the origin, and thus $\beta_n = 0$ for all n and $\mu_0 = 2$. According to (4.5.55) we have

$$\gamma_n = \frac{n}{\sqrt{4n^2 - 1}} = \frac{1}{\sqrt{4 - n^{-2}}}.$$

ALGORITHM 5.4. *Gauss–Legendre Quadrature.*

The following MATLAB function computes the nodes and weights of the Gauss–Legendre rule with n points by generating the bidiagonal matrix B and its SVD.

```
function [x,w] = legendre(n);
% LEGENDRE(n) computes the nodes and weights in the
% Gauss-Legendre quadrature rule with n+1 nodes (n > 1).
%
gamma = 1./sqrt(4 - [1:n].^(-2));
gamma(n+1) = 0;
b0(1) = gamma(1:2:n+1);
b1(k) = gamma(2:2:n);
B = diag(b0,0) + diag(b1,1);
[P,S,Q] = svd(B);
x = diag(S); [x,i] = sort(x);
w = P(1,i).^2;
if rem(n,2) == 0 w(1) = 2*w(1); end
```

For $n = 6$ the upper bidiagonal matrix becomes

$$B = \begin{pmatrix} 1/\sqrt{3} & 2/\sqrt{15} & & & \\ & 3/\sqrt{35} & 4/\sqrt{63} & & \\ & & 5/\sqrt{99} & 6/\sqrt{143} & \\ & & & 0 & \end{pmatrix} \in \mathbf{R}^{4\times 4},$$

and we obtain the nonnegative nodes (cf. Table 5.3.1) $x_1 = 0$,

$$x_2 = 0.40584515137740, \qquad x_3 = 0.74153118559939, \qquad x_4 = 0.94910791234276.$$

The first row of $P = (\, P_1 \quad P_2\,)$ is

$$-0.45714285714286,\ -0.61792398440675,\ 0.52887181007242,\ -0.35984019532130,$$

where the first entry corresponds to node $x_1 = 0$. Dividing the last three components by $\sqrt{2}$, squaring, and multiplying with $\mu_0 = 2$, gives the weights

$$w_1 = 0.41795918367347,\ \ w_2 = 0.38183005050512,\ \ w_3 = 0.27970539148928,$$
$$w_4 = 0.12948496616887.$$

We remark that the given program is inefficient in that the full matrices of left and right singular vectors are computed. Unless n is very large the execution time is negligible anyway.

In the computation of harmonic transforms used in spectral weather analysis, Gauss–Legendre quadrature rules with values of n in excess of 1000 are required. Methods for computing points and weights accurate to double precision for such high values of n are discussed by Swarztrauber in [338].

Review Questions

5.3.1 What increase in order of accuracy can normally be achieved by a judicious choice of the nodes in a quadrature formula?

5.3.2 What are orthogonal polynomials? Give a few examples of families of orthogonal polynomials together with the three-term recursion formula, which its members satisfy.

5.3.3 Formulate and prove a theorem concerning the location of zeros of orthogonal polynomials.

5.3.4 Give an account of Gauss quadrature formulas, including accuracy and how the nodes and weights are determined. What important properties are satisfied by the weights?

5.3.5 What is the orthogonality property of the Legendre polynomials?

Problems and Computer Exercises

5.3.1 Prove that the three-point quadrature formula

$$\int_{-1}^{1} f(x)\,dx \approx \frac{1}{9}\left(5f\left(-\sqrt{\frac{3}{5}}\right) + 8f(0) + 5f\left(\sqrt{\frac{3}{5}}\right)\right)$$

is exact for polynomials of degree five. Apply it to the computation of

$$\int_0^1 \frac{\sin x}{1+x}\,dx,$$

and estimate the error in the result.

5.3.2 (a) Calculate the Hermite polynomials H_n for $n \le 4$ using the recurrence relation.
(b) Express, conversely, $1, x, x^2, x^3, x^4$ in terms of the Hermite polynomials.

5.3.3 (a) Determine the orthogonal polynomials $\phi_n(x)$, $n = 1, 2, 3$, with leading coefficient 1, for the weight function $w(x) = 1 + x^2$, $x \in [-1, 1]$.
(b) Give a two-point Gaussian quadrature formula for integrals of the form

$$\int_{-1}^{1} f(x)(1+x^2)\,dx$$

which is exact when $f(x)$ is a polynomial of degree three.

Hint: Use either the method of undetermined coefficients taking advantage of symmetry, or the three-term recurrence relation in Theorem 5.3.1.

5.3.4 (W. Gautschi)
(a) Construct the quadratic polynomial ϕ_2 orthogonal on $[0, \infty]$ with respect to the weight function $w(x) = e^{-x}$. *Hint:* Use $\int_0^\infty t^m e^{-t}\,dt = m!$.

(b) Obtain the two-point Gauss–Laguerre quadrature formula

$$\int_0^\infty f(x)e^{-x}\,dx = w_1 f(x_1) + w_2 f(x_2) + E_2(f),$$

including a representation for the remainder $E_2(f)$.

(c) Apply the formula in (b) to approximate

$$I = \int_0^\infty (x+1)^{-1} e^{-x}\,dx.$$

Use the remainder term to estimate the error, and compare your estimate with the true error ($I = 0.596347361\ldots$).

5.3.5 Show that the formula

$$\int_{-1}^{1} f(x)(1-x^2)^{-1/2}\,dx = \frac{\pi}{n}\sum_{k=1}^{n} f\left(\cos\frac{2k-1}{2n}\pi\right)$$

is exact when $f(x)$ is a polynomial of degree at most $2n-1$.

5.3.6 (a) Use the MATLAB program in Example 5.3.4 to compute nodes and weights for the Gauss–Hermite quadrature rule. Use it to compute a 10-point rule; check the result using a table.

(b) Write a program for computing nodes and weights for Gauss quadrature rules when $w(x)$ is not symmetric. In MATLAB use the function `[v,d] = eig(J)` to solve the eigenvalue problems. Use the program to compute some Gauss–Laguerre quadrature rules.

5.3.7 Derive the Gauss–Lobatto quadrature rule in Example 5.3.3, with two interior points by using the Ansatz

$$\int_{-1}^{1} f(x)\,dx = w_1(f(-1)+f(1)) + w_2(f(-x_1)+f(x_1)),$$

and requiring that it be exact for $f(x) = 1, x^2, x^4$.

5.3.8 Compute an approximate value of

$$\int_{-1}^{1} x^4 \sin^2 \pi x\,dx = 2\int_0^1 x^4 \sin^2 \pi x\,dx,$$

using a five-point Gauss–Legendre quadrature rule on [0, 1] for the weight function $w(x) = 1$. For nodes and weights see Table 5.3.1 or use the MATLAB function `legendre(n)` given in Example 5.3.4. (The true value of the integral is 0.11407 77897 39689.)

5.3.9 (a) Determine exactly the Lobatto formulas with given nodes at -1 and 1 (and the remaining nodes free), for the weight functions

$$w(x) = (1-x^2)^{-1/2}, \quad x \in [-1, 1].$$

Determine for this weight function also the nodes and weights for the Gauss quadrature formula (i.e., when all nodes are free).

Hint: Set $x = \cos\phi$, and formulate equivalent problems on the unit circle. Note that you obtain (at least) two different discrete orthogonality properties of the Chebyshev polynomials this way.

(b) Lobatto–Kronrod pairs are useful when a long interval has been divided into several shorter intervals (cf. Example 5.3.3). Determine Lobatto–Kronrod pairs (exactly) for $w(x) = (1 - x^2)^{-1/2}$.

5.3.10 Apply the formulas in Problem 5.3.9 to the case $w(x) = 1$, $x \in [-1, 1]$, and some of the following functions.

(a) $f(x) = e^{kx}$, $k = 1, 2, 4, 8, \ldots$; (b) $f(x) = 1/(k + x)$, $k = 1, 2, 1.1, 1.01$;

(c) $f(x) = k/(1 + k^2x^2)$, $k = 1, 4, 16, 64$.

Compare the actual errors with the error estimates.

5.3.11 For $k = 1$ the integral (5.2.24) in Example 5.2.5 is

$$\int_{-\pi/2}^{\pi/2} \frac{\cos t}{t + \pi + u(t + \pi)}\, dt.$$

Compute this integral with at least ten digits of accuracy, using a Gauss–Legendre rule of sufficiently high order. Use the MATLAB function `legendre(n)` given in Example 5.3.4 to generate the nodes and weights.

5.3.12 Write a MATLAB function for the evaluation of the Sievert[176] integral,

$$S(x, \theta) = \int_0^{\theta} e^{-x/\cos\phi}\, d\phi,$$

for any $x \geq 0$, $x \leq \theta \leq 90°$, with at least six decimals relative accuracy. There may be useful hints in [1, Sec. 27.4].

5.4 Multivariate Integration

Numerical integration formulas in several dimensions, sometimes called **numerical cubature**, are required in many applications. Several new difficulties are encountered in deriving and applying such rules.

In one dimension any finite interval of integration $[a, b]$ can be mapped by an affine transformation onto $[-1, 1]$ (say). Quadrature rules need therefore only be derived for this standard interval. The order of accuracy of the rule is preserved since affine transformations preserve the degree of the polynomial. In d dimensions the boundary of the region of integration has dimension $d - 1$, and can be complicated manifold. For any dimension $d \geq 2$ there are infinitely many connected regions in $\mathbf{R}^d$ which cannot be mapped onto each other using affine transformations. Quadrature rules with a certain polynomial accuracy designed for any of these regions are fundamentally different than for any other region.

[176] Sievert was a Swedish radiophysicist, was so revered that doses of radiation are measured in millisieverts, or even microsieverts, all over the world.

The number of function values needed to obtain an acceptable approximation tends to increase exponentially in the number of dimensions d. That is, if n points are required for an integral in one dimension, then n^d points are required in d dimensions. Thus, even for a modest number of dimensions, achieving an adequate accuracy may be an intractable problem. This is often referred to as **the curse of dimensionality**, a phrase coined by Richard Bellman.[177]

5.4.1 Analytic Techniques

It is advisable to try, if possible, to reduce the number of dimensions by applying analytic techniques to parts of the task.

Example 5.4.1.

The following triple integral can be reduced to a single integral:

$$\int_0^\infty \int_0^\infty \int_0^\infty e^{-(x+y+z)} \sin(xz) \sin(yx)\, dx dy dz$$
$$= \int_0^\infty e^{-x}\, dx \int_0^\infty e^{-y} \sin(yx) dy \int_0^\infty e^{-z} \sin(zx)\, dz = \int_0^\infty \Big(\frac{x}{1+x^2}\Big)^2 e^{-x}\, dx.$$

This is possible because

$$\int_0^\infty e^{-z} \sin(zx) dz = \int_0^\infty e^{-y} \sin(yx) dz = \frac{x}{1+x^2}.$$

The remaining single integral is simply evaluated by the techniques previously studied.

Often a transformation of variable is needed for such a reduction. Given a region D in the (x, y)-plane, this is mapped onto a region D' in the (u, v)-plane by the variable transformation

$$x = \phi(u, v), \qquad y = \psi(u, v). \tag{5.4.1}$$

If ϕ and ψ have continuous partial derivatives and the Jacobian

$$J(u, v) = \begin{vmatrix} \partial\phi/\partial u & \partial\phi\partial u \\ \partial\psi\partial u & \partial\psi\partial u \end{vmatrix} \tag{5.4.2}$$

does not vanish in D', then

$$\int\int_D f(x, y)\, dx\, dy = \int\int_{D'} f(\phi(x, y), \psi(x, y))|J(u, v)|\, du\, dv. \tag{5.4.3}$$

It is important to take into account any symmetries that the integrand can have. For example, the integration of a spherically symmetric function over a spherical region reduces in polar coordinates to a one-dimensional integral.

[177] Richard Ernest Bellman (1920–1984) was an American mathematician. From 1949 to 1965 he worked at the Rand Corporation and made important contributions to operations research and dynamic programming.

Example 5.4.2.

To evaluate the integral

$$I = \int\int_D \frac{y \sin(ky)}{x^2 + y^2}\, dx\, dy,$$

where D is the unit circle $x^2 + y^2 \leq 1$, we introduce polar coordinates (r, φ), $x = r\cos\varphi$, $y = r\sin\varphi$, $dx\,dy = r\,dr\,d\varphi$. Then, after integrating in the r variable, this integral is reduced to the single integral

$$I = \frac{1}{k}\int_0^{2\pi} [1 - \cos(k\sin\varphi)]\, d\varphi.$$

This integral is not expressible in finite terms of elementary functions. Its value is in fact $(1 - J_0(k))2\pi/k$, where J_0 is a Bessel function. Note that the integrand is a periodic function of φ, in which the trapezoidal rule is very efficient (see Sec. 5.1.4). This is a useful device for Bessel functions and many other transcendental functions which have integral representations.

If the integral cannot be reduced, then several approaches are possible:

(a) Tensor products of one-dimensional quadrature rules can be used. These are particularly suitable if the boundary of the region is composed of straight lines. Otherwise numerical integration in one direction at a time can be used; see Sec. 5.4.3.

(b) For more general boundaries an irregular triangular grid can be used; see Sec. 5.4.4.

(c) Monte Carlo or quasi–Monte Carlo methods can be used, mainly for problems with complicated boundaries and/or a large number of dimensions; see Sec. 5.4.5.

5.4.2 Repeated One-Dimensional Integration

Consider a double integral (5.4.7) over a region D in the (x, y)-plane such that lines parallel with the x-axis have at most one segment in common with D (see Figure 5.4.1). Then J can be written in the form

$$I = \int_a^b \left(\int_{c(x)}^{d(x)} f(x, y) dy \right) dx,$$

or

$$I = \int_a^b \varphi(x)\, dx, \qquad \varphi(x) = \int_{c(x)}^{d(x)} f(x, y) dy. \tag{5.4.4}$$

The one-dimensional integral $\varphi(x)$ can evaluated for the sequence of abscissae x_i, $i = 1, \ldots, n$, used in another one-dimensional quadrature rule for J. Note that if D is a more general domain, it might be possible to decompose D into the union of simpler domains on which these methods can be used.

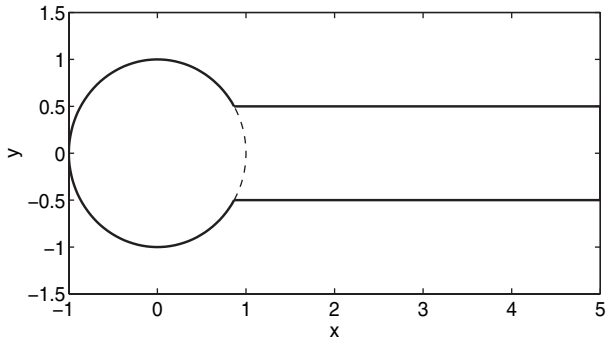

Figure 5.4.1. *Region D of integration.*

Example 5.4.3.

Compute

$$I = \int\int_D \sin^2 y \sin^2 x (1 + x^2 + y^2)^{-1/2}\, dx\, dy,$$

where

$$D = \{(x, y) \mid x^2 + y^2 \le 1\} \cup \{(x, y) \mid 0 \le x \le 3, |y| \le 0.5\}$$

is a composite region (see Figure 5.4.1). Then

$$I = \int_{-1}^{3} \sin^2 x\, \varphi(x)\, dx, \tag{5.4.5}$$

$$\varphi(x) = \int_{-c(x)}^{c(x)} \sin^2 y\, (1 + x^2 + y^2)^{-1/2} dy, \tag{5.4.6}$$

where

$$c(x) = \begin{cases} (1 - x^2)^{1/2}, & x < \frac{1}{2}\sqrt{3}, \\ \frac{1}{2}, & x \ge \frac{1}{2}\sqrt{3}. \end{cases}$$

Values of $\varphi(x)$ were obtained by the application of Romberg's method to (5.4.6) and numerical integration applied to the integral (5.4.5) yielded the value of $I = 0.13202 \pm 10^{-5}$. Ninety-six values of x were needed, and for each value of x, 20 function evaluations used, on the average. The grid is chosen so that $x = \frac{1}{2}\sqrt{3}$, where $\varphi'(x)$ is discontinuous, is a grid point.

5.4.3 Product Rules

In $d = 2$ dimensions, common boundaries are a rectangle, circle, or triangle, or a combination of these. Consider a **double integral** over a rectangular region

$$I = \int\int_D u(x, y)\, dxdy, \quad D = \{(x, y) \mid a \le x \le b,\ c \le y \le d\}. \tag{5.4.7}$$

Introduce an equidistant **rectangular grid** in the (x, y)-plane, with grid spacings h and k in the x and y directions,

$$x_i = a + ih, \quad y_j = c + jk, \quad h = (b-a)/n, \quad k = (d-c)/m,$$

and set $u_{ij} = u(x_i, y_j)$. Then the following **product rule** for the double integral generalizes the compound midpoint rule:

$$I \approx hk \sum_{i=1}^{m} \sum_{j=1}^{n} u_{i-1/2, j-1/2}. \tag{5.4.8}$$

The product trapezoidal rule is

$$\begin{aligned} I &\approx hk \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{1}{4}(u_{i-1,j-1} + u_{i-1,j} + u_{i,j-1} + u_{i,j}) \\ &= hk \sum_{i=0}^{m} \sum_{j=0}^{n} w_{ij} u_{ij}. \end{aligned} \tag{5.4.9}$$

Here $w_{ij} = 1$ for the interior grid points, i.e., when $0 < i < m$, and $0 < j < n$. For the trapezoidal rule $w_{ij} = \frac{1}{4}$ for the four corner points, while $w_{ij} = \frac{1}{2}$ for the other boundary points. Both formulas are exact for all **bilinear functions** $x^i y^j$, $0 \le i, j \le 1$. The error can be expanded in even powers of h and k so that Romberg's method can be used to get more accurate results. The generalization to integrals over the hypercube $[0, 1]^d$ is straightforward.

It is not necessary to use the same quadrature rule in both dimensions. Suppose we have the two one-dimensional quadrature rules

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i) + (b-a)R_1, \qquad \int_c^d g(y)\,dy \approx \sum_{j=1}^{m} v_j g(y_j) + (d-c)R_2. \tag{5.4.10}$$

Combining these two rules over the rectangular region D gives the product rule

$$\begin{aligned} &\int_a^b \int_c^d u(x, y)\,dx\,dy \approx \int_a^b \left(\sum_{j=1}^{m} v_j u(x, y_j) + (d-c)R_2 \right) dx \\ &= \sum_{j=1}^{m} v_j \int_a^b u(x, y_j)\,dx + \int_a^b (d-c)R_2\,dx \approx \sum_{i=1}^{n} \sum_{j=1}^{m} w_i v_j u(x_i, y_j) + R, \end{aligned}$$

where

$$R = (d-c) \int_a^b R_2\,dx + (b-a) \sum_{j=1}^{m} v_j R_1 \approx (b-a)(d-c)(R_1 + R_2).$$

The following property of product rules follows easily.

Theorem 5.4.1.

If the two one-dimensional rules (5.4.10) *integrate $f(x)$ exactly over $[a, b]$ and $g(y)$ exactly over $[c, d]$, then the product rule* (5.4.11) *integrates $u(x, y) = f(x)g(y)$ exactly over the region $[a, b] \times [c, d]$.*

If the one-dimensional rules are exact for polynomials of degree d_1 and d_2, respectively, then the product rule will be exact for all bivariate polynomials $x^p y^q$, where $p \le d_1$ and $q \le d_2$.

Example 5.4.4.

The product Simpson's rule for the square $|x| \le h$, $|y| \le h$ has the form

$$\int_{-h}^{h}\int_{-h}^{h} u(x, y)\,dxdy = 4h^2 \sum_{j=-1}^{1}\sum_{i=-1}^{1} w_{i,j}u(x_i, y_j).$$

It uses $3^2 = 9$ function values, with abscissae and weights given by

(x_i, y_j)	(0,0)	$(\pm h, \pm h)$	$(\pm h, 0)$	$(0, \pm h)$
$w_{i,j}$	4/9	1/36	1/9	1/9

Of similar accuracy is the product rule obtained from a two-point Gauss–Legendre rule, which uses the four points

$$(x_i, y_i) = \left(\pm\frac{h}{\sqrt{3}}, \pm\frac{h}{\sqrt{3}}\right) \quad w_i = \frac{1}{4}.$$

For both rules the error is $O(h^4)$. Note that for the corresponding composite rules, the functions values at corner points and midpoints in the product Simpson's rule are shared with other subsquares. Effectively this rule also uses four function values per subsquare.

Higher-accuracy formulas can also be derived by **operator** techniques, based on an operator formulation of Taylor's expansion (see (4.8.2)),

$$u(x_0 + h, y_0 + k) = e^{(hD_x + kD_y)}u(x_0, y_0). \tag{5.4.11}$$

For regions D, such as a square, cube, cylinder, etc., which are the Cartesian product of lower-dimensional regions, product integration rules can be developed by multiplying together the lower-dimensional rules. Product rules can be used on nonrectangular regions, if these can be mapped into a rectangle. This can be done, for example, for a triangle, but product rules derived in this way are often not very efficient and are seldom used.

For nonrectangular regions, the rectangular grid may also be bordered by triangles or "triangles" with one curved side, which may be treated with the techniques outlined in the next section.

So far we have restricted ourselves to the two-dimensional case. But the ideas are more general. Let $(x_1, \ldots, x_r) \in C$, where C is a region in $\mathbf{R}^r$ and $(y_1, \ldots, y_s) \in D$, where D is a region in $\mathbf{R}^r$. Let $C \times D$ denote the Cartesian product of C and D, i.e., the region in $\mathbf{R}^{r+s}$ consisting of points (using vector notations) $(\mathbf{x}, \mathbf{y})$ such that $\mathbf{x} \in C$ and $\mathbf{y} \in D$.

Suppose we have two quadrature rules for the regions C and D

$$\int_C f(\mathbf{x})\,d\mathbf{x} \approx \sum_{i=1}^{n} w_i f(\mathbf{x}_i), \qquad \int_D g(\mathbf{y})\,d\mathbf{y} \approx \sum_{j=1}^{m} v_i g(\mathbf{y}_i). \tag{5.4.12}$$

We can combine these two rules to give a product rule for the region $C \times D$:

$$\int_{C\times D} u(\mathbf{x},\mathbf{y})\,d\mathbf{x}\,d\mathbf{y} \approx \sum_{i=1}^{n}\sum_{i=1}^{n} w_i v_j u(\mathbf{x}_i,\mathbf{y}_j). \tag{5.4.13}$$

Product rules are not necessarily the most economical rules. More efficient quadrature rules exist, which are not the result of applying one-dimensional rules to several dimensions. We could try to determine such rules by selecting n nodes and weights so that the rule integrates bivariate polynomials of as high a degree as possible. This is much more difficult in several dimensions than in one dimension, where this approach led to Gaussian rules. The solution is in general not unique; there may be several rules with different nodes and weights. For most regions it is not known what the best rules are. Some progress has been made in developing nonproduct quadrature rules of optimal order for triangles.

Some simple quadrature rules for circles, triangles, hexagons, spheres, and cubes are given in Abramowitz–Stegun [1, pp. 891–895], for example, the following quadrature rule for a double integral over a disk $C = \{(x, y) \in C \mid x^2 + y^2 \le h^2\}$:

$$\int\int_C f(x, y)\,dxdy = \pi h^2 \sum_{i=1}^{4} w_i f(x_i, y_i) + O(h^4),$$

where

$$(x_i, y_i) = (\pm h/2, \pm h/2), \quad w_i = 1/4, \quad i = 1:4.$$

This four-point rule has the same order of accuracy as the four-point Gaussian product for the square given in Example 5.4.4. A seven-point $O(h^6)$ rule uses the points (see Figure 5.4.2)

$$\begin{aligned}(x_1, y_1) = (0, 0), \quad (x_i, y_i) &= (\pm h\sqrt{2/3}, 0), \quad i = 2, 3\\ (x_i, y_i) &= (\pm h/\sqrt{6}, \pm h/\sqrt{2}), \quad i = 4:7,\end{aligned}$$

with weights $w_1 = 1/4$, and $w_i = 1/8$, $i = 2:7$.

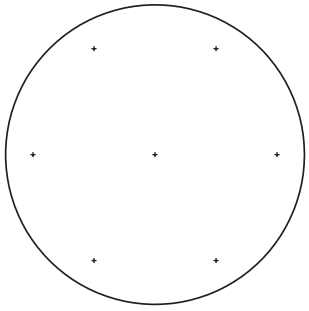

Figure 5.4.2. *A seven-point $O(h^6)$ rule for a circle.*

Example 5.4.5.

We seek a quadrature rule

$$I = \int\int_T f(x, y)\, dxdy = A \sum_{i=1}^{n} w_i f(x_i, y_i) + R, \tag{5.4.14}$$

where T is an equilateral triangle with sides of length h and area $A = h^2\sqrt{3}/4$. We use function values at the "center of mass" $(x_1, y_1) = (0, h/(2\sqrt{3}))$ of the triangle and at the corner nodes

$$(x_i, y_i) = (\pm h/2, 0), \quad i = 2, 3, \quad \text{and} \quad (x_4, y_4) = (0, h\sqrt{3}/2).$$

Then, taking $w_1 = 3/4$, and $w_i = 1/12$, $i = 2 : 4$, we get a four-point rule with error $R = O(h^3)$.

Adding nodes at the midpoint of the sides

$$(x_5, y_5) = (0, 0) \quad \text{and} \quad (x_i, y_i) = (\pm h/4, h\sqrt{3}/4), \quad i = 6, 7,$$

and using weights $w_1 = 9/20$, and $w_i = 1/20$, $i = 2 : 4$, $w_i = 2/15$, $i = 5 : 7$, gives a seven-point rule for which $R = O(h^4)$ in (5.4.14).

5.4.4 Irregular Triangular Grids

A grid of triangles of arbitrary form is a convenient means for approximating a complicated plane region. It is fairly easy to program a computer to refine a coarse triangular grid automatically; see Figure 5.4.3. It is also easy to adapt the density of points to the behavior of the function.

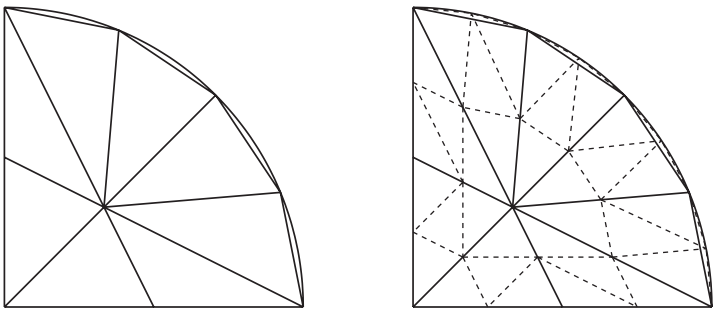

Figure 5.4.3. *Refinement of a triangular grid.*

Triangular grids are thus more flexible than rectangular ones. On the other hand, the administration of a rectangular grid requires less storage and a simpler program. Sometimes the approximation formulas are also a little simpler. Triangular grids are used, for example, in the **finite element method** (FEM) for problems in continuum mechanics and other applications of partial differential equations; see [111].

Let the points P_j, $j = 1, 2, 3$, with coordinates $p_j = (x_j, y_j)$, be the vertices of a triangle T with area $Y > 0$. Then any point $p = (x, y)$ in the plane can be uniquely expressed by the vector equation

$$p = \theta_1 p_1 + \theta_2 p_2 + \theta_3 p_3, \quad \theta_1 + \theta_2 + \theta_3 = 1. \tag{5.4.15}$$

The θ_i, which are called homogeneous **barycentric coordinates** of P, are determined from the following nonsingular set of equations:

$$\begin{aligned} \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 &= x, \\ \theta_1 y_1 + \theta_2 y_2 + \theta_3 y_3 &= y, \\ \theta_1 + \theta_2 + \theta_3 &= 1. \end{aligned} \tag{5.4.16}$$

Barycentric coordinates were discovered by Möbius[178] in 1827; see Coxeter [83, Sec. 13.7]. In engineering literature the barycentric coordinates for a triangle are often called **area coordinates** since they are proportional to the area of the three subtriangles induced by P; see Figure 5.4.4.

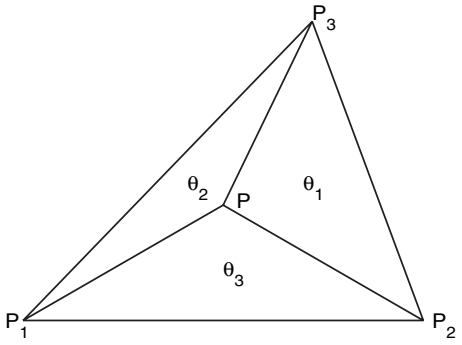

Figure 5.4.4. *Barycentric coordinates of a triangle.*

The interior of the triangle is characterized by the inequalities $\theta_i > 0$, $i = 1, 2, 3$. In this case P is the center of mass (centroid) of the three masses $\theta_1, \theta_2, \theta_3$ located at the vertices of the triangle. This explains the term "barycentric coordinates." The equation for the side P_2P_3 is $\theta_1 = 0$; similarly $\theta_2 = 0$ and $\theta_3 = 0$ describe the other two sides. Note that if θ and θ' ($i = 1, 2, 3$) are the barycentric coordinates of the points P_i and P_j, respectively, then the barycentric coordinates of $\alpha P + (1 - \alpha) P'$ are $\alpha\theta + (1 - \alpha)\theta'$.

Barycentric coordinates are useful also for $d > 2$ dimensions. By a **simplex** in $\mathbf{R}^d$ we mean the convex hull of $(d + 1)$ points $p_j = (p_{1j}, p_{2j}, \ldots, p_{dj})^T \in \mathbf{R}^d$, which are called the vertices of the simplex. We assume that the vertices are not contained in a hyper-plane. This is the case if and only if the $(d + 1) \times (d + 1)$ matrix

$$A = \begin{pmatrix} p_1 & p_2 & \cdots & p_{d+1} \\ 1 & 1 & \cdots & 1 \end{pmatrix} \tag{5.4.17}$$

is nonsingular. For $d = 2$ the simplex is a triangle and for $d = 3$ a tetrahedron.

[178] August Ferdinand Möbius (1790–1868) was a German astronomer and mathematician, and a professor at the University of Leipzig. His 1827 work *Barycentric Calculus* became a classic and played an important role in the development of projective geometry.

The barycentric coordinates of a point p are the unique vector $\theta \in \mathbf{R}^{d+1}$ such that

$$(p_1 \ldots, p_{d+1})\theta = p, \quad e^T\theta = 1 \tag{5.4.18}$$

or, equivalently, $\theta = A^{-1}\binom{p}{1}$. The center of gravity of the simplex is the point with coordinates $\theta_i = 1/(d+1)$, $i = 1 : d+1$.

If u is a *nonhomogeneous linear function* of p, i.e., if

$$u(p) = a^T p + b = (a^T, b)\begin{pmatrix} p \\ 1 \end{pmatrix},$$

then the reader can verify that

$$u(p) = \sum_{j=1}^{d+1} \theta_j u(p_j), \quad u(p_j) = a^T p_j + b. \tag{5.4.19}$$

This is a form of *linear interpolation* and shows that a linear function is uniquely determined by its values at the vertices.

Using also the midpoints of the edges $p_{ij} = \frac{1}{2}(p_i + p_j)$ a *quadratic interpolation* formula can be obtained.

Theorem 5.4.2.

Define

$$\Delta''_{ij} = u(p_i) + u(p_j) - 2u\left(\frac{1}{2}(p_i + p_j)\right), \quad i < j. \tag{5.4.20}$$

Then the interpolation formula

$$u(p) = \sum_j \theta_j u(p_j) - 2\sum_{i<j} \theta_i\theta_j \Delta''_{ij}, \tag{5.4.21}$$

where the summation indices i, j are assumed to take all values $1 : d+1$ unless otherwise specified, is exact for all quadratic functions.

Proof. The right-hand is a quadratic function of p, since it follows from (5.4.16) that the θ_i are (nonhomogeneous) linear functions of the coordinates of p. It remains to show that the right-hand side is equal to $u(p)$ for $p = p_j$, and $p = (p_i + p_j)/2$, $i, j = 1 : d+1$.

For $p = p_j$, $\theta_j = 1$, $\theta_i = 0$, $i \neq j$, hence the right-hand side equals u_i. For $p = (p_i + p_j)/2$, $\theta_i = \theta_j = 1/2$, $\theta_k = 0$, $k \neq i, j$, and hence the right-hand side becomes

$$\frac{1}{2}(u_i + u_j) - 2 \cdot \frac{1}{2}\left(u_i + u_j - 2u\left(\frac{1}{2}(p_i + p_j)\right)\right) = u\left(\frac{1}{2}(p_i + p_j)\right). \qquad \square$$

The following theorem for triangles ($d = 2$) is equivalent to a rule which has been used in mechanics for the computation of moments of inertia since the nineteenth century.

Theorem 5.4.3.

Let T be a triangle with vertices p_1, p_2, p_3 and area Y. Then the integration formula

$$\int_T u(x,y)\,dxdy = \frac{Y}{3}\left(u\left(\frac{1}{2}(p_1+p_2)\right)+u\left(\frac{1}{2}(p_1+p_3)\right)+u\left(\frac{1}{2}(p_2+p_3)\right)\right) \tag{5.4.22}$$

is exact for all quadratic functions.

Proof. Using the interpolation formula (5.4.21), the integral equals

$$\int_T u(x,y)\,dxdy = \sum_j u(p_j)\int_T \theta_j dxdy - 2\sum_{i<j}\Delta''_{ij}\int_T \theta_i\theta_j dxdy.$$

By symmetry, $\int_T \theta_i\,dxdy$ is the same for $i = 1, 2, 3$. Similarly, $\int_T \theta_i\theta_j\,dxdy$ is the same for all $i < j$. Hence using (5.4.20)

$$\begin{aligned}\int_T u(x,y)\,dxdy &= a(u_1+u_2+u_3) - 2b(\Delta''_{23}+\Delta''_{13}+\Delta''_{12})\\ &= (a-4b)(u_1+u_2+u_3)\\ &\quad + 4b\left(u\left(\frac{1}{2}(p_1+p_2)\right)+u\left(\frac{1}{2}(p_2+p_3)\right)+u\left(\frac{1}{2}(p_3+p_1)\right)\right),\end{aligned} \tag{5.4.23}$$

where

$$a = \int_T \theta_1\,dxdy, \qquad b = \int_T \theta_1\theta_2\,dxdy.$$

Using θ_1, θ_2 as new variables of integration, we get by (5.4.16) and the relation $\theta_3 = 1-\theta_1-\theta_2$

$$\begin{aligned}x &= \theta_1(x_1-x_3)+\theta_1(x_1-x_3)+x_3,\\ y &= \theta_1(y_1-y_3)+\theta_1(y_1-y_3)+y_3.\end{aligned}$$

The functional determinant is equal to

$$\begin{vmatrix} x_1-x_3 & x_2-x_3 \\ y_1-y_3 & y_2-y_3 \end{vmatrix} = 2Y,$$

and (check the limits of integration!)

$$\begin{aligned}a &= \int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1-\theta_1} 2\theta_1 d\theta_1 d\theta_2 = 2Y\int_0^1 \theta_1(1-\theta_1)d\theta_1 = \frac{Y}{3},\\ b &= \int_{\theta_1=0}^{1}\int_{\theta_2=0}^{1-\theta_1} 2\theta_1\theta_2 d\theta_1 d\theta_2 = 2A\int_0^1 \theta_1\frac{(1-\theta_1)^2}{2}d\theta_1 = \frac{Y}{12}.\end{aligned}$$

The results now follow by insertion of this into (5.4.23). □

A numerical method for a two-dimensional region can be based on Theorem 5.4.2, by covering the domain D by triangles. For each curved boundary segment (Figure 5.4.5) the correction

$$\frac{4}{3} f(S) A(PRQ) \tag{5.4.24}$$

is to be added, where $A(PRQ)$ is the area of the triangle with vertices P, R, Q. The error of the correction can be shown to be $O(\|Q - P\|^5)$ for each segment, if R is close to the midpoint of the arc PQ. If the boundary is given in parametric form, $x = x(x)$, $y = y(x)$, where x and y are twice differentiable on the arc PQ, then one should choose $t_R = \frac{1}{2}(t_P + t_Q)$. Richardson extrapolation can be used to increase the accuracy; see the examples below.

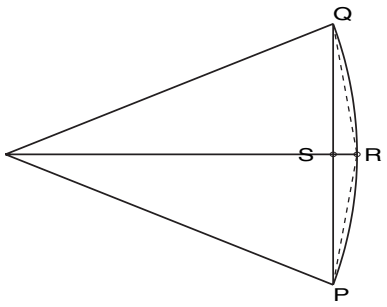

Figure 5.4.5. *Correction for curved boundary segment.*

Example 5.4.6.

Consider the integral

$$I = \int\int_D (x^2 + y^2)^k \, dxdy,$$

where the region D and the grids for I_4 and I_{16} are shown in Figure 5.4.6 and I_n denotes the result obtained with n triangles. Because of symmetry the error has an expansion in even powers of h. Therefore, we can use repeated Richardson extrapolation and put

$$R'_n = I_{4n} + \frac{1}{15}(I_{4n} - I_n), \qquad R''_n = R'_{4n} + \frac{1}{63}(R'_{4n} - R'_n).$$

The results are shown in the table below. In this case the work could be reduced by a factor of four, because of symmetry.

k	I_4	I_{16}	I_{64}	R'_4	R'_{16}	R''_4	Correct
2	0.250000	0.307291	0.310872	0.311111	0.311111	0.311111	28/90
3	0.104167	0.161784	0.170741	0.165625	0.171338	0.171429	0.171429
4	0.046875	0.090678	0.104094	0.093598	0.104988	0.105169	0.105397

It is seen that R'-values have full accuracy for $k = 2$ and that the R''-values have high accuracy even for $k = 4$. In fact, it can be shown that R'-values are exact for any

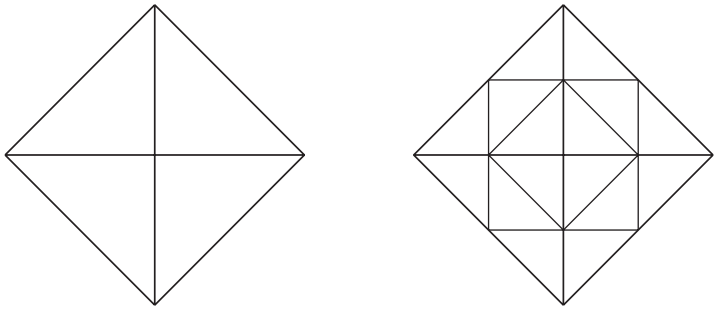

Figure 5.4.6. *The grids for I_4 and I_{16}.*

fourth-degree polynomial and R''-values are exact for any sixth-degree polynomial, when the region is covered exactly by the triangles.

Example 5.4.7.

The integral

$$a \int\int (a^2 - y^2)^{-1/2}\, dxdy,$$

over a quarter of the unit circle $x^2+y^2 \leq 1$, is computed with the grids shown in Figure 5.4.6, and with boundary corrections according to (5.4.15). The following results, using the notation of the previous example, were obtained and compared with the exact values.

a	I_8	I_{32}	R'_8	Correct
2	0.351995	0.352077	0.352082	0.352082
4	0.337492	0.337608	0.337615	0.337616
6	0.335084	0.335200	0.335207	0.335208
8	0.334259	0.334374	0.334382	0.334382

Note, however, that Richardson extrapolation may not always give improvement, for example, when the rate of convergence of the basic method is *more rapid* than usual.

5.4.5 Monte Carlo Methods

Multidimensional integrals arise frequently in physics, chemistry, computational economics,[179] and other branches of science. If a product rule is used to evaluate a multivariate integral in d dimensions the work will increase exponentially with the number of dimensions d. For example, the product rule of an 8-point one-dimensional rule will require $(8)^8 = 2^{24} \approx 1.6 \cdot 10^7$ function evaluations in eight dimensions. This means that the problem may quickly become intractable when d increases.

One important application of the Monte Carlo method described in Section 1.4.2 is the numerical calculation of integrals of high dimension. For the Monte Carlo method

[179] The valuation of financial derivatives can require computation of integrals in 360 dimensions!

the accuracy achieved is always proportional to $1/\sqrt{n}$, where n is the number of function evaluations *independent of the dimension* d. Thus, if approached randomly multivariate integration becomes tractable! The Monte Carlo method can be said to break "the curse of dimension" inherent in other methods. (For smooth integrands the Monte Carlo method is, however, not of optimal complexity.)

We shall briefly describe some ideas used in integration by the Monte Carlo method. For simplicity, we first consider integrals in *one* dimension, even though the Monte Carlo method cannot really compete with traditional numerical methods for this problem.

Let R_i, $i = 1 : N$, be a sequence of random numbers rectangularly distributed on $[0, 1]$, and set

$$I = \int_0^1 f(x)\,dx \approx I_1, \quad I_1 = \frac{1}{N}\sum_{i=1}^{N} f(R_i).$$

Then the expectation of the variable I_1 is I and its standard deviation decreases as $N^{-1/2}$. I_1 can be interpreted as a stochastic estimate of the mean value of $f(x)$ in the interval $[0, 1]$. This generalizes directly to multivariate integrals over the unit hypercube. Let $R_i \in \mathbf{R}^d$, $i = 1 : N$, be a sequence of random points uniformly distributed on $[0, 1]^d$. Then

$$I = \int_{[0,1]^d} f(x)\,dx \approx I_1, \quad I_1 = \frac{1}{N}\sum_{i=1}^{N} f(R_i). \tag{5.4.25}$$

If the integral is to be taken over a subregion $\mathcal{D} \subset [0, 1]^d$, we can simply set $f(x) = 0$, $x \notin \mathcal{D}$. In contrast to interpolatory quadrature methods smooth functions are not integrated more efficiently than discontinuous functions. According to the law of large numbers, the convergence

$$I_N(f) \to \mathrm{vol}(\mathcal{D})\mu(f) \quad \text{as} \quad N \to \infty,$$

where $\mu(f)$ is the mean value of $f(X)$, where X is a continuous random variable uniformly distributed in $\mathcal{D} \subset [0, 1]^d$.

A probabilistic error estimate can be obtained by estimating the standard deviation of $\mu(f)$ by the empirical standard deviation $s_N(f)$, where

$$s_N(f)^2 = \frac{1}{N-1}\sum_{i=1}^{N} \left(f(R_i) - I_N(f)\right)^2. \tag{5.4.26}$$

If the integral is over a subregion $\mathcal{D} \subset [0, 1]^d$, we should use the mean value over $\mathcal{D}$, that is, neglect all points $R_i \notin \mathcal{D}$.

The standard deviation of the Monte Carlo estimate in (5.4.25) decreases as $N^{-1/2}$. This is very slow even compared to the trapezoidal rule, for which the error decreases as N^{-2}. To get one extra decimal place of accuracy we must increase the number of points by a factor of 100. To get three-digit accuracy the order of one million points may be required!

But if we consider, for example, a six-dimensional integral this is not exorbitant. Using a product rule with ten subdivisions in each dimension would also require 10^6 points.

The above Monte Carlo estimate is a special case of a more general one. Suppose X_i, $i = 1 : N$, has density function $g(x)$. Then

$$I_2 = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{g(X_i)}$$

has expected value I, since

$$E\left(\frac{f(X_i)}{g(X_i)}\right) = \int_0^1 \frac{f(x)}{g(x)} f(x)\, dx = \int_0^1 f(x)\, dx = I.$$

If one can find a frequency function $g(x)$ such that $f(x)/g(x)$ fluctuates less than $f(x)$, then I_2 will have smaller variance than I_1. This procedure is called **importance sampling**; it has proved very useful in particle physics problems, where important phenomena (for example, dangerous radiation which penetrates a shield) are associated with certain events of low probability.

We have previously mentioned the method of using a simple comparison problem. The Monte Carlo variant of this method is called the **control variate method**. Suppose that $\varphi(x)$ is a function whose integral has a known value K, and suppose that $f(x) - \varphi(x)$ fluctuates much less than $f(x)$. Then

$$I = K + \int_0^1 (f(x) - \varphi(x))\, dx,$$

where the integral to the right can be estimated by

$$I_3 = \frac{1}{N} \sum_{i=1}^{N} (f(R_i) - \varphi(R_i)),$$

which has less variance than I_1.

5.4.6 Quasi–Monte Carlo and Lattice Methods

In Monte Carlo methods the integrand is evaluated at a sequence of points which are supposed to be a sample of independent random variables. In **quasi–Monte Carlo** methods the accuracy is enhanced by using specially chosen deterministic points not necessarily satisfying the statistical tests discussed in Sec. 1.6.2. These points are constructed to be approximately equidistributed over the region of integration.

If the region of integration D is a subset of the d-dimensional unit cube $C_n = [0, 1]^d$ we set $f(x) \equiv 0$, for $x \notin D$. We can then always formulate the problem as the approximation of an integral over the d-dimensional unit cube $C_n = [0, 1]^d$:

$$I[f] = \int_{C_n} f(x_1, \ldots, x_d)\, dx_1 \ldots dx_d. \tag{5.4.27}$$

An infinite sequence of vectors $x_1, x_2, x_3, \ldots$ in $\mathbf{R}^d$ is said to be **equidistributed** in the cube $[0, 1]^d$ if

$$I[f] = \lim_{N\to\infty} Q_N(f), \quad Q_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(x_i), \tag{5.4.28}$$

for all Riemann integrable functions $f(x)$. The quadrature rules Q_N are similar to those used in Monte Carlo methods; this explains the name quasi–Monte Carlo methods.

In the **average case setting** the requirement that the worst case error is smaller than ϵ is replaced by the weaker guarantee that the expected error is at most ϵ. This means that we make some assumptions about the distribution of the functions to be integrated. In this setting the complexity of multivariate integration has been shown to be proportional to $1/\epsilon$, compared to $(1/\epsilon)^2$ for the Monte Carlo method. Hence the Monte Carlo method is not optimal.

The convergence of the quadrature rules Q_N in (5.4.28) depends on the variation of f and the distribution of the sequence of points $x_1, \ldots, x_N$. The **discrepancy** of a finite sequence of points $x_1, x_2, \ldots, x_N$ is a measure of how much the distribution of the sequence deviates from an equidistributed sequence. The deterministic set of points used in quasi–Monte Carlo are constructed from **low discrepancy sequences**, which are, roughly speaking, uniformly spread as $N \to \infty$; see Niederreiter [270].

Let $0 < a_i \le 1$, $i = 1 : d$, and restrict $f(x)$, $x \in \mathbf{R}^n$, to the class of functions

$$f(x) = \begin{cases} 1 & \text{if } 0 \le x_i \le a_i, \\ 0 & \text{otherwise.} \end{cases}$$

We require the points to be such that every $Q_N(f)$ gives a good approximation of the integral of $f(x)$ over the hypercube $[0, 1]^d$ for all functions in this class.

Low discrepancy sequences are usually generated by algorithms from number theory, a branch of mathematics seemingly far removed from analysis. Recall from Sec. 2.2.1 that each integer i has a unique representation $d_k \cdots d_2 d_1 d_0$ with respect to a integer basis $b \ge 2$. The **radical inverse function** φ_b maps an integer i onto the real number

$$\varphi_b(i) = 0.d_0 d_1 d_2 \cdots d_k \in [0, 1).$$

The **Van der Corput sequence** (see [358]) with respect to the base b is the infinite sequence defined by

$$x_i = \varphi_b(i), \quad i = 1, 2, 3, \ldots.$$

These sequences can be shown to have an asymptotic optimal discrepancy. The first few elements in the sequence for $b = 2$ are shown in the table below.

i			$\varphi_2(i)$
1	1	.1	0.5
2	10	.01	0.25
3	11	.11	0.75
4	100	.001	0.125
5	101	.101	0.625
6	110	.011	0.375
7	111	.111	0.875

Halton sequences (see [176]) are multidimensional extensions of Van der Corput sequences.

Definition 5.4.4.
Let the bases $b_1, b_2, \ldots, b_d$ be pairwise relative prime. Then the Halton sequence $x_i \in [0,1]^d$ with respect to these bases is defined by

$$x_i = (\varphi_{b_1}(i), \varphi_{b_2}(i), \ldots, \varphi_{b_d}(i))^T, \quad i = 0, 1, 2, \ldots, \tag{5.4.29}$$

where $\varphi_{b_k}(i)$ is the radical inverse function with respect to the basis b_k, $k = 1 : d$.

The **Hammersley sequences** [179] are similar to Halton sequences but are finite and differ in the definition of the first component. The N-point Hammersley sequence with respect to the bases $b_1, b_2, \ldots, b_{d-1}$ is the sequence of points $x_i \in [0,1]^d$ defined by

$$x_i = \left(i/N, \varphi_{b_1}(i), \varphi_{b_2}(i), \ldots, \varphi_{b_{d-1}}(i)\right)^T, \quad i = 0 : N-1. \tag{5.4.30}$$

The Hammersley points in $[0,1]^2$ for $N = 512$ and $b_1 = 2$ are shown in Figure 5.4.7.

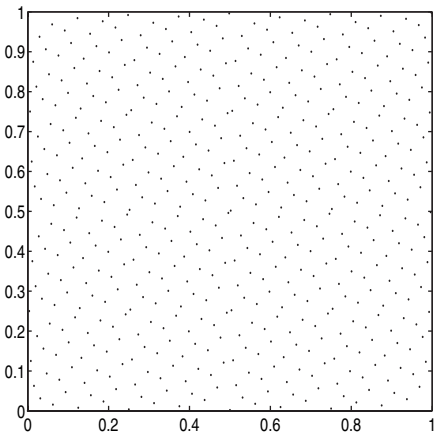

Figure 5.4.7. *Hammersley points in* $[0,1]^2$.

Wozniakowski [376] showed that the Hammersley points are optimal sampling points for multivariate integration. With n function evaluations a worst case error of quasi–Monte Carlo methods is bounded by a multiple of $(\log n)^d/n$, which can be compared to $n^{-1/2}$ for Monte Carlo methods.

ALGORITHM 5.5. *Hammersley Points.*

The following MATLAB program generates N Hammersley points in the two-dimensional square $[0, 1]^2$ for $b_1 = 2$.

```
[x,y] = Hammersley(N);
n = ceil(log2(N));
for i = 1:N
  x(i) = (i-1)/N;
  j = i-1;
  for p = 1:n
    j = j/2; d(p) = 0;
    if j > floor(j)
      d(p) = 1; j = floor(j);
    end;
  end;
  phi = d(n)/2;
  for p = n-1:-1:1
    phi = (phi + d(p))/2;
  end;
  y(i) = phi;
end;
```

Using harmonic analysis it was shown in Sec. 5.1.4 that the composite trapezoidal rule can be very accurate for periodic integrands. These results can be extended to multivariate integration of periodic integrands. A **lattice rule** for the numerical integration over the d-dimensional unit cube $C_n = [0, 1]^d$ is an equal weight rule,

$$Q_N(f) = \frac{1}{N} \sum_{j=0}^{N-1} f(x_i), \tag{5.4.31}$$

where the sampling points x_i, $i = 0 : N - 1$, are points of an integration lattice in the cube $[0, 1]^d$. A multivariate extension of the compound trapezoidal rule is obtained by taking

$$x_i = \text{fraction}\left(\frac{i}{N^p}\right),$$

where fraction(x) returns the fractional part of x. Lattice rules can be studied by expanding the integrand in a Fourier series. The "curse of dimension" can be lifted by using a class of randomly shifted lattice rules introduced by Ian H. Sloane.

Review Questions

5.4.1 What is meant by a product integration rule for computing a multivariate integral? What is the drawback with such rules for high dimensions?

5.4.2 Give the generalization of the composite trapezoidal and midpoint rules for a two-dimensional rectangular grid.

5.4.3 Define barycentric coordinates in two dimensions. Give a formula for linear interpolation on a triangular grid.

5.4.4 For high-dimensional integrals and difficult boundaries Monte Carlo methods are often used. How does the accuracy of such methods depend on the number n of evaluations of the integrand?

5.4.5 How do quasi–Monte Carlo methods differ from Monte Carlo methods?

Problems and Computer Exercises

5.4.1 Let E be the ellipse $\{(x, y) \mid (x/a)^2 + (y/b)^2 \le 1\}$. Transform

$$I = \int\int_E f(x, y)\, dxdy$$

into an integral over a rectangle in the (r, t)-plane with the transformation $x = a\, r \cos t$, $y = b\, r \sin t$.

5.4.2 Consider the integral I of $u(x, y, z)$ over the cube $|x| \le h$, $|y| \le h$, $|z| \le h$. Show that the rule

$$I \approx \frac{4}{3}h^3 \sum_{i=1}^{6} f(x_i, y_i, z_i),$$

where $(x_i, y_i, z_i) = (\pm h, 0, 0)$, $(0, \pm h, 0)$, $(0, 0, \pm h)$, is exact for all monomials $x^i y^j$, $0 \le i, j \le 1$.

5.4.3 (a) In one dimension Simpson's rule can be obtained by taking the linear combination $S(h) = (T(h) + 2M(h))/3$ of the trapezoidal and midpoint rule. Derive a quadrature rule

$$\int_{-h}^{h}\int_{-h}^{h} f(x, y)\, dxdy = \frac{4h^2}{6}(f_{1,0} + f_{0,1} + f_{-1,0} + f_{0,-1} + 2f_{0,0})$$

for the square $[-h, h]^2$ by taking the same linear combination of the *product* trapezoidal and midpoint rules. Note that this rule is not equivalent to the product Simpson's rule.

(b) Show that the rule in (a) is exact for all cubic polynomials. Compare its error term with that of the product Simpson's rule.

(c) Generalize the midpoint and trapezoidal rules to the cube $[-h, h]^3$. Then derive a higher-order quadrature rule using the idea in (a).

5.4.4 Is a quadratic polynomial uniquely determined, given six function values at the vertices and midpoints of the sides of a triangle?

5.4.5 Show that the boundary correction of (5.4.15) is exact if $f \equiv 1$, and if the arc is a parabola where the tangent at R is parallel to PQ.

5.4.6 Formulate generalizations to several dimensions of the integral formula of Theorem 5.4.2, and convince yourself of their validity.

Hint: The formula is most simply expressed in terms of the values in the vertices and in the centroid of a simplex.

5.4.7 (a) Write a program which uses the Monte Carlo method to compute $\int_0^1 e^x\,dx$. Take 25, 100, 225, 400, and 635 points. Plot the error on a log-log scale. How does the error depend (approximately) on the number of points?

(b) Compute the integral in (a) using the control variate method. Take $\varphi(x) = 1 + x + x^2/2$. Use the same number of points as in (a).

5.4.8 Use the Monte Carlo method to estimate the multiple integral

$$I = \int_{[0,1]^n} \prod |x_k - 1/3|^{1/2} \approx 0.49^n,$$

for $n = 6$. What accuracy is attained using $N = 10^3$ random points uniformly distributed in six dimensions?

5.4.9 Write a program to generate Halton points in two dimensions for $b_1 = 2$ and $b_2 = 5$. Then plot the first 200 points in the unit square.

Hint: For extracting the digits to form x_i, see Algorithm 2.1. You can also use TOMS Algorithm 247; see [177].

Notes and References

Numerical integration is a mature and well-understood subject. There are several comprehensive monographs devoted to this area; see in particular Davis and Rabinowitz [91] and Engels [110]. Examples of integrals arising in practice and their solution are found in [91, Appendix 1]. Newton–Cotes' and other quadrature rules can also be derived using computer algebra systems; see [129]. A collection of numerical quadrature rules is given in the Handbook [1, Sec. 25].

The idea of adaptive Simpson quadrature is old and treated fully by Lyness [246]. Further schemes, computer programs, and examples are given in [91]. A recent discussion of error estimates and reliability of different codes is given by Espelid [113].

The literature on Gauss–Christoffel quadrature and its computational aspects is extensive. Gauss–Legendre quadrature was derived by Gauss in 1814 using a continued fraction expansion. In 1826 Jacobi showed that the nodes were the zeros of the Legendre polynomials and that they were real, simple, and in $[-1, 1]$. The convergence of Gaussian quadrature methods was first studied by Stieltjes in 1884. More on the history can be found in Gautschi [140]. Recent results by Trefethen [352] suggest that the Clenshaw–Curtis rule may be as accurate as Gauss–Legendre quadrature with an equal number of nodes.

The importance of the eigenvalues and eigenvectors of the Jacobi matrices for computing Gauss' quadrature rules was first elaborated by Golub and Welsch [167]. The generalization to Radau and Lobatto quadrature was outlined in Golub [159] and further generalized by Golub and Kautsky [161].

The presentation in Sec. 5.3.4 was developed in Dahlquist [86]. Related ideas can be found in Gautschi [137, 143] and Mysovskih [268]. The encyclopedic book by Gautschi [145]

describes the current state-of-the-art of orthogonal polynomials and Gauss–Christoffel quadrature computation; see also the survey by Laurie [233].

There is an abundance of tables giving abscissae and weights for various quadrature rules. Abramowitz and Stegun [1, Sec. 25] give tables for Newton–Cotes' and several Gauss–Christoffel rules. Many Gaussian quadrature formulas with various weight functions are tabulated in Stroud and Secrest [336]. A survey of other tables is given in [91, Appendix 4].

Many algorithms and codes for generating integration rules have appeared in the public domain. In [91, Appendices 2, 3] several useful Fortran programs are listed and a bibliography of Algol and Fortran programs published before 1984 is given. Kautsky and Elhay [219] have developed algorithms and a collection of Fortran subroutines called IQPACK [108] for computing weights of interpolatory quadratures. QUADPACK is a collection of Fortran 77 and 90 subroutines for integration of functions available at `www.netlib.org`. It is described in the book by R. Piessens et al. [285].

A software package in the public domain by Gautschi [142] includes routines for generating Gauss-type arbitrary weight functions. A package QPQ consisting of MATLAB programs for generating orthogonal polynomials as well as dealing with applications is available at `www.cs.purdue.edu/archives/2002/wxg/codes`. Part of these programs are described in Gautschi [147]. Maple programs for Gauss quadrature rules are given by von Matt [257]. An overview of results related to Gauss–Kronrod rules is given by Monegato [265]. The calculation of Gauss–Kronrod rules is dealt with in [232, 61].

Gaussian product rules for integration over the n-dimensional cube, sphere, surface of a sphere, and tetrahedron are derived in Stroud and Secrest [336, Ch. 3]. Some simple formulas of various accuracy are tabulated in [1, Sec. 25]. The derivation of such formulas are treated by Engels [110]. Nonproduct rules for multidimensional integration are found in Stroud [335].

A good introduction to multidimensional integration formulas and Monte Carlo methods is given by Ueberhuber [357, Chap. 12]. Construction of fully symmetric numerical multidimensional integration formulas over the hypercube $[-h, h]^d$ using a rectangular grid is treated by McNamee and Stenger [258]. For a survey of recent results on sparse grids and breaking "the curse of dimensionality," see Bungartz and Griebel [59]. The efficiency of quasi–Monte Carlo methods are discussed in [321]. Lattice rules are treated in the monograph by Sloan and Joe [320]. For an introduction see also [357, Sec. 12.4.5].