3.2 Adaptive and Iterated Quadratures; Romberg's Method

3.2.1 Adaptive Quadratures

As seen so far, the errors in numerical integration methods depend not only on the size of the interval, but also on values of certain higher order derivatives of the function to be integrated. Newton-Cotes methods (including the three simple ones, that use low degree polynomial interpolation) work well for smooth integrands (even with a small number of nodes), but perform poorly for functions having large values of higher order derivatives – especially for functions having large oscillations on some subintervals or on the whole interval. As a simple example, consider

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

This integrand has infinite derivative in $x = 0$, but is smooth at points close to $x = 1$.

Generally, numerical integration schemes use evenly spaced nodes. When the function to be integrated has a singularity at some point $\alpha \in [a, b]$, this requires many nodes in the vicinity of that point, to reduce the errors caused by the chaotic behaviour of the function in that neighborhood. But this implies that many more nodes (more than necessary) are used throughout the *entire* interval of integration, increasing (unnecessarily) the computational cost of the method. Ideally, we want to use small subintervals where the derivatives are large, and larger subintervals where the derivatives are small and well-behaved.

A method that does this systematically is called **adaptive quadrature**. The general approach in an adaptive quadrature is to use two different methods on each subinterval, compare the results, and divide the interval when the differences are large. The structure of such an algorithm would be "Divide and conquer".

In Algorithm 3.1 we present an example of a general structure for a recursive adaptive quadrature. The parameter "met" is a function that implements a composite quadrature rule, such as the trapezoidal or Simpson's rule, and m is the number of subintervals.

Unlike other methods, that decide what amount of work is needed to achieve a desired precision, an adaptive quadrature computes only as much as is necessary.

Algorithm 3.1. [Adaptive quadrature]

```
function $I = adquad(f, a, b, \varepsilon, met, m)$
        $I1 = met(f, a, b, m)$;
        $I2 = met(f, a, b, 2m)$;
```

```
    if $|I1 - I2| < \varepsilon$ % success
      $I = I2$;
      return
    else   % recursive subdivision
      $I = adquad(f, a, \frac{a+b}{2}, \varepsilon, met, m) + adquad(f, \frac{a+b}{2}, b, \varepsilon, met, m)$;
    end
end
```

3.2.2 Richardson Extrapolation

Extrapolation is a method for generating high-accuracy numerical schemes using low-order formulas. The most widely used is *Richardson extrapolation.*

Consider the integral

$$I := \int_a^b f(x)\,dx$$

and a numerical integration scheme

$$I \approx I_n,$$

for which we have an asymptotic error formula of the form

$$I - I_n \approx \frac{c}{n^p}, \tag{3.1}$$

where c depends on a, b and the derivatives of a certain order of the function f on $[a, b]$. The difficulty in using this estimate is not knowing the value of the constant c. We can obtain a computable estimate of the error without needing to know c explicitly. We write (3.1) for a larger n:

$$I - I_{2n} \approx \frac{c}{(2n)^p} = \frac{c}{2^p n^p} \tag{3.2}$$

and eliminate the unknown c from relations (3.1)-(3.2). We obtain

$$I - I_n \approx 2^p\big(I - I_{2n}\big),$$

and then the approximation

$$I \approx \frac{2^p I_{2n} - I_n}{2^p - 1} = I_{2n} + \frac{I_{2n} - I_n}{2^p - 1} \overset{\text{not}}{=} R_{2n}, \tag{3.3}$$

called **Richardson's extrapolation formula**. From this, we can get another error estimate for I_{2n},

$$I - I_{2n} \approx \frac{I_{2n} - I_n}{2^p - 1}, \tag{3.4}$$

called **Richardson's error estimate**. The term R_{2n} is an improved estimate of I, based on using I_n, I_{2n}, p and the assumption (3.1). It is a more accurate approximation to I than is I_{2n}. How much more accurate it is depends on the validity of (3.1)–(3.2).

Example 3.2. Let us consider a few simple Newton-Cotes formulas.

Solution. For the composite trapezoidal rule, if f has continuous second order derivatives on $[a, b]$, we have

$$\begin{aligned}\int_a^b f(x)dx &= \frac{b-a}{n}\Big[f(a) + f(b) + \sum_{i=1}^{n-1} f\Big(a + \frac{b-a}{n}i\Big)\Big] - \frac{(b-a)^3}{12n^2}f''(\xi) \\ &= T_n + \frac{c}{n^2},\end{aligned}$$

hence, $p = 2$. Using Richardson extrapolation, we get

$$I \approx \frac{4T_{2n} - T_n}{3} = T_{2n} + \frac{1}{3}\big(T_{2n} - T_n\big) \tag{3.5}$$

and the error formula

$$I - T_{2n} \approx \frac{1}{3}\big(T_{2n} - T_n\big). \tag{3.6}$$

With Simpson's repeated rule, if f has continuous fourth order derivatives on $[a, b]$ and $f_j = f\Big(a + \frac{b-a}{2n}j\Big)$, $j = \overline{0, 2n}$, we have

$$\begin{aligned}\int_a^b f(x)dx &= \frac{b-a}{6n}\Big[f(a) + f(b) + 4\sum_{i=1}^{n} f_{2i-1} + 2\sum_{i=1}^{n-1} f_{2i}\Big] - \frac{(b-a)^5}{2880n^4}f^{(4)}(\xi) \\ &= S_n + \frac{c}{n^4},\end{aligned}$$

so in this case, $p = 4$. With Richardson extrapolation, we obtain

$$I \approx \frac{16S_{2n} - S_n}{15} = S_{2n} + \frac{1}{15}\big(S_{2n} - S_n\big) \tag{3.7}$$

and the error

$$I - S_{2n} \approx \frac{1}{15}\big(S_{2n} - S_n\big). \tag{3.8}$$

■

Example 3.3. Consider again the problem in Example 3.8 in Lecture 7: the approximation of the integral

$$\int_0^1 e^{-x^2} dx = 0.746824132812427,$$

using the composite trapezoidal rule.

Solution. Last time we obtained the errors

n	Error	Ratio
2	$1.55e-2$	
4	$3.84e-3$	4.02
8	$9.59e-4$	4.01
16	$2.40e-4$	4.00
32	$5.99e-5$	4.00

Table 1: Errors in repeated trapezoidal rule, Example 3.3

We have

$$\begin{aligned} T_2 &= 0.7313702518, \\ T_4 &= 0.7429840978. \end{aligned}$$

By (3.5), we get the approximation

$$I \approx R_4 = \frac{1}{3}\big(4T_4 - T_2\big) = 0.7468553798,$$

with absolute error

$$0.0000312 \;=\; 3.12e-5.$$

Notice in Table 1 that the error of R_4 is smaller than that of T_{32}, so R_4 (after only 2 steps) gives a better approximation of I than T_{32}, obtained after 5 steps!

Now, let us estimate the error in T_4 using Richardson extrapolation. By (3.6), we have

$$I - T_4 \;\approx\; \frac{1}{3}\big(T_4 - T_2\big) \;=\; 0.00387.$$

The actual error in T_4 is 0.00384, so we obtained a very accurate error estimate.

■

Remark 3.4. Richardson's extrapolation and error estimation are not always as accurate as this example might suggest, but it is usually a fairly accurate procedure. The main assumption that must be satisfied is (3.1), with a known p. And the extrapolation itself provides a way of testing whether this assumption is valid for the actual values of I_n being used: Continue the ideas in (3.1)–(3.3) and write successively

$$\begin{aligned} I - I_n &\approx 2^p\big(I - I_{2n}\big), \\ I - I_{2n} &\approx 2^p\big(I - I_{4n}\big). \end{aligned}$$

We get

$$\begin{aligned} I_{2n} - I_n &= \big(I - I_n\big) - \big(I - I_{2n}\big) \\ &\approx 2^p\big(I - I_{2n}\big) - \big(I - I_{2n}\big) \;=\; (2^p - 1)\big(I - I_{2n}\big). \end{aligned}$$

Similarly, we have

$$\begin{aligned} I_{4n} - I_{2n} &= \big(I - I_{2n}\big) - \big(I - I_{4n}\big) \\ &\approx \big(I - I_{2n}\big) - 2^{-p}\big(I - I_{2n}\big) \;=\; (1 - 2^{-p})\big(I - I_{2n}\big). \end{aligned}$$

Then,

$$\frac{I_{2n} - I_n}{I_{4n} - I_{2n}} \;\approx\; \frac{2^p - 1}{1 - \dfrac{1}{2^p}} \;=\; 2^p.$$

We obtained the (computable) estimate

$$2^p \approx \frac{I_{2n} - I_n}{I_{4n} - I_{2n}},$$

or

$$p \approx \log_2 \left(\frac{I_{2n} - I_n}{I_{4n} - I_{2n}} \right) = \frac{1}{\ln 2} \ln \left(\frac{I_{2n} - I_n}{I_{4n} - I_{2n}} \right). \tag{3.9}$$

This gives a practical means of checking/finding the value of p in (3.1), using three successive values I_n, I_{2n}, I_{4n}.

3.2.3 Iterated Quadratures; Romberg's Method

Just like in the case of Lagrange interpolation, we want algorithms for which it is easy to go from one step (iteration) to the next, by using previously computed values of the function.

One drawback of adaptive quadratures is that they compute repeatedly the function values at the nodes and when such an algorithm is executed, there is an extra computational cost due to recursion. *Iterated quadratures* overcome this shortcoming. They apply at the first step a composite quadrature rule and then divide the interval into equal parts using at each step the previously computed approximations. **Romberg's method** is such an iterative algorithm, starting with the composite trapezoidal (or midpoint) rule and then improving the convergence by using Richardson extrapolation.

The initial approximations are obtained by applying either the trapezoid or midpoint rule with $n_k = 2^{k-1}, k \in \mathbb{N}$. Then the value of the step h_k is

$$h_k = \frac{b-a}{n_k} = \frac{b-a}{2^{k-1}}.$$

With these notations, we have (for the trapezoidal rule)

$$\int_a^b f(x)dx = \frac{h_k}{2} \left[f(a) + f(b) + 2 \sum_{i=1}^{2^{k-1}-1} f\big(a + ih_k\big) \right] - \frac{b-a}{12} h_k^2 f''(\xi_k), \ \xi_k \in [a,b].$$

Denote by $R_{k,1}$ the approximation above, i.e.,

$$R_{1,1} = \frac{h_1}{2} \Big[f(a) + f(b) \Big] = \frac{b-a}{2} \Big[f(a) + f(b) \Big], \tag{3.10}$$

$$
\begin{aligned}
R_{2,1} &= \frac{h_2}{2}\Big[f(a)+f(b)+2f\big(a+h_2\big)\Big] \\
&= \frac{b-a}{4}\Big[f(a)+f(b)+2f\Big(a+\frac{1}{2}h_1\Big)\Big] \\
&= \frac{1}{2}\Big[\frac{b-a}{2}\Big(f(a)+f(b)\Big)+(b-a)f\Big(a+\frac{1}{2}h_1\Big)\Big] \\
&= \frac{1}{2}\Big[R_{1,1}+h_1 f\Big(a+\frac{1}{2}h_1\Big)\Big] \\
&\cdots \\
R_{k,1} &= \frac{1}{2}\Big[R_{k-1,1}+h_{k-1}\sum_{i=1}^{2^{k-2}} f\Big(a+\big(i-\frac{1}{2}\big)h_{k-1}\Big)\Big],\ k=\overline{2,n}.
\end{aligned}
\tag{3.11}
$$

Since $h_k = \frac{1}{2}h_{k-1}$, each successive level of improvement increases the order of the error term from $O(h^{2k-2})$ to $O(h^{2k})$, so by

$$O(h^2) \;=\; O\Big(\frac{1}{n^2}\Big).$$

Then we can use Richardson extrapolation with $p = 2$, by eliminating the term in h_k^2 from the approximation of I by $R_{k-1,1}$ and $R_{k,1}$, respectively. We obtain

$$I \;=\; \frac{4R_{k,1}-R_{k-1,1}}{3}+O(h_k^4)$$

and define

$$R_{k,2} \;=\; \frac{4R_{k,1}-R_{k-1,1}}{3}. \tag{3.12}$$

We apply Richardson extrapolation to these values, too. In general, if $f \in C^{2n+2}[a,b]$, then, for $k = \overline{1,n}$, we can write

$$\int_a^b f(x)dx \;=\; \frac{h_k}{2}\Big[f(a)+f(b)+2\sum_{i=1}^{2^{k-1}-1} f\big(a+ih_k\big)\Big]+\sum_{i=1}^{k}K_i h_k^{2i}+O(h_k^{2k+2}),$$

where K_i does not depend on h_k.

Successively eliminating the powers of h from the relation above, we get

$$R_{k,j} \quad = \quad \frac{4^{j-1}R_{k,j-1} - R_{k-1,j-1}}{4^{j-1}-1},\ k = \overline{2,n},\ j = \overline{2,k}. \tag{3.13}$$

The computations can be arranged in a table (from (3.11) and (3.13)):

$$\begin{array}{lllll}
R_{1,1} & & & & \\
R_{2,1} & R_{2,2} & & & \\
R_{3,1} & R_{3,2} & R_{3,3} & & \\
\vdots & \vdots & \vdots & \ddots & \\
R_{n,1} & R_{n,2} & R_{n,3} & \dots & R_{n,n}
\end{array}$$

Since the sequence $\{R_{n,1}\}_n$ converges, so does $\{R_{n,n}\}_n$, at a faster rate. We can use the stopping criterion

$$|R_{n-1,n-1} - R_{n,n}| \quad < \quad \varepsilon.$$

Remark 3.5. The second column in Romberg's method corresponds to Simpson's composite rule. We introduce the notation

$$S_{k,1} \quad = \quad R_{k,2}.$$

Then, the values in the third column are

$$R_{k,3} \quad = \quad \frac{4^2 R_{k,2} - R_{k-1,2}}{4^2 - 1} \quad = \quad \frac{16S_{k,1} - S_{k-1,1}}{15},$$

which is Richardson's extrapolation for Simpson's rule. The relation

$$S_{k,1} \quad = \quad \frac{16S_{k,1} - S_{k-1,1}}{15} \tag{3.14}$$

is at the core of a well-known (and oftenly used) adaptive quadrature algorithm (due to Gander and Gautschi).

Example 3.6. Approximate

$$\int_0^\pi \sin x \, dx$$

by Romberg's method, with precision $\varepsilon = 10^{-1}$.

Solution. The exact value of the integral is

$$I = -\cos x\Big|_0^\pi = 2.$$

Using the repeated trapezoidal rule with $n_1 = 2^0, h_1 = \pi$ (i.e. nodes $x_0 = 0, x_1 = \pi$) and $n_2 = 2^1, h_2 = \pi/2$ (so nodes $x_0 = 0, x_1 = \pi/2, x_2 = \pi$), we get

$$\begin{aligned} R_{1,1} &= \frac{\pi}{2}(\sin 0 + \sin\pi) = 0, \\ R_{2,1} &= \frac{1}{2}\Big(R_{1,1} + h_1 \sin\frac{\pi}{2}\Big) = \frac{\pi}{2} = 1.5708. \end{aligned}$$

Richardson extrapolation is next:

$$R_{2,2} = \frac{4R_{2,1} - R_{1,1}}{3} = \frac{2\pi}{3} = 2.0944.$$

We have

$$|R_{2,2} - R_{1,1}| = 2.0944 > 0.1,$$

so we continue. We compute

$$\begin{aligned} R_{3,1} &= \frac{1}{2}\Big[R_{2,1} + h_2\Big(\sin\frac{\pi}{4} + \sin\frac{3\pi}{4}\Big)\Big] = 1.8961, \\ R_{3,2} &= \frac{4R_{3,1} - R_{2,1}}{3} = 2.0046, \\ R_{3,3} &= \frac{16R_{3,2} - R_{2,2}}{15} = 1.9986 \end{aligned}$$

and

$$|R_{3,3} - R_{2,2}| = 0.0958 < 0.1.$$

Hence, we obtained the approximation

$$I \approx R_{3,3} = 1.9986,$$

(with an error of $1.4e-3$), which is obviously better than the trapezoidal rule with $n = 4$, $R_{3,1}$ (with the error of 0.1039). Also, it is more accurate than Simpson's approximation with 4 nodes,

$I \approx 2.005$ (with error $5e-3$).

In fact, for this example, the algorithm converges very fast, as seen below:

$$\begin{array}{llll} 0 & & & \\ 1.5708 & 2.0944 & & \\ 1.8961 & 2.0046 & 1.9986 & \\ 1.9742 & 2.0003 & 2.0000 & 2.0000 \end{array}$$

■

3.3 Gaussian Quadratures

3.3.1 General Framework

Definition 3.7. *An interpolatory formula of the form*

$$\int_a^b w(x) f(x)\, dx = \sum_{k=1}^{m} A_k f(x_k) + R_m(f) \tag{3.15}$$

*is called **Gaussian quadrature** if it has maximum degree of precision,* $d = 2m-1$.

The function $w : (a,b) \to \mathbb{R}_+$ is a *weight function*, a function for which the *moments*

$$\mu_j = \int_a^b w(x) x^j\, dx \tag{3.16}$$

exist and are finite for each $j \in \mathbb{N}$. The purpose of a weight function is to "absorb" some singularities of the integrand.

We want to determine the coefficients A_k and the nodes x_k such that

$$R_m(e_0) = R_m(e_1) = \ldots = R_m(e_{2m-1}) = 0. \tag{3.17}$$

Let us start with a simple example. Consider the integral

$$\int_{-1}^{1} f(x)\, dx. \tag{3.18}$$

So, in this case,

$$w(x) \equiv 1.$$

Case $m = 1$.

We seek a numerical integration formula

$$\int_{-1}^{1} f(x)\,dx \approx A_1 f(x_1).$$

From the first two relations in (3.17), we get

$$\begin{aligned} A_1 &= 2, \\ A_1 x_1 &= 0, \end{aligned}$$

and, thus, the formula

$$\int_{-1}^{1} f(x)\,dx \approx 2f(0),$$

which is the midpoint (rectangle) rule.

Case $m = 2$.

Now, we want a quadrature of the form

$$\int_{-1}^{1} f(x)\,dx \approx A_1 f(x_1) + A_2 f(x_2),$$

with 4 unknowns, which are determined from the first 4 relations (3.17). This leads to the system

$$\begin{aligned} A_1 + A_2 &= 2 \\ A_1 x_1 + A_2 x_2 &= 0 \\ A_1 x_1^2 + A_2 x_2^2 &= \tfrac{2}{3} \\ A_1 x_1^3 + A_2 x_2^3 &= 0 \end{aligned} \tag{3.19}$$

with solution

$$A_1 = A_2 = 1,\ x_1 = -\frac{\sqrt{3}}{3},\ x_2 = \frac{\sqrt{3}}{3}. \tag{3.20}$$

Hence, we found the quadrature formula

$$\int_{-1}^{1} f(x)\,dx \;\approx\; f\Big(-\frac{\sqrt{3}}{3}\Big) + f\Big(\frac{\sqrt{3}}{3}\Big), \tag{3.21}$$

with degree of precision $d = 3$.

General case $m > 2$.
In a similar fashion, we obtain the system

$$\begin{aligned} A_1 + A_2 + \cdots + A_{2m-1} &= 2 \\ A_1x_1 + A_2x_2 + \cdots + A_{2m-1}x_{2m-1} &= 0 \\ A_1x_1^2 + A_2x_2^2 + \cdots + A_{2m-1}x_{2m-1}^2 &= \frac{2}{3} \\ A_1x_1^3 + A_2x_2^3 + \cdots + A_{2m-1}x_{2m-1}^3 &= 0 \\ \cdots &\cdots \\ A_1x_1^{2m-2} + A_2x_2^{2m-2} + \cdots + A_{2m-1}x_{2m-1}^{2m-2} &= \frac{2}{2m-1} \\ A_1x_1^{2m-1} + A_2x_2^{2m-1} + \cdots + A_{2m-1}x_{2m-1}^{2m-1} &= 0. \end{aligned} \tag{3.22}$$

Example 3.8. Consider again the integral in Example3.3,

$$I \;=\; \int_{0}^{1} e^{-x^2} dx \;=\; 0.746824132812427.$$

Solution. The linear change of variables

$$x \;=\; \frac{b + a + t(b-a)}{2}$$

maps the interval $[-1, 1]$ to $[a, b]$. So, with the substitution

$$x \;=\; \frac{1+t}{2}, \;\; t \;=\; 2x - 1,$$

we get

$$I \;=\; \frac{1}{2}\int_{-1}^{1} e^{-\frac{1}{4}(1+t)^2}\, dt.$$

We apply Gaussian quadratures to the above integral. The errors are given in Table2.

m	Error
2	$2.29e-4$
3	$9.55e-6$
4	$3.35e-7$
5	$6.05e-9$
6	$7.77e-11$
7	$7.89e-13$

Table 2: Gaussian quadratures errors, Example 3.8

Comparing these with the ones given by the composite trapezoid or Simpson's rules (even with extrapolation), we see that these approximations are much more accurate, with fewer nodes.

■

Gaussian quadrature formulas for a general weight function w can be found completely similarly. From relations (3.17), we obtain the system

$$\begin{array}{rcl} A_1 + A_2 + \cdots + A_{2m-1} & = & \mu_1 \\ A_1x_1 + A_2x_2 + \cdots + A_{2m-1}x_{2m-1} & = & \mu_2 \\ A_1x_1^2 + A_2x_2^2 + \cdots + A_{2m-1}x_{2m-1}^2 & = & \mu_2 \\ \cdots & \cdots & \\ A_1x_1^{2m-1} + A_2x_2^{2m-1} + \cdots + A_{2m-1}x_{2m-1}^{2m-1} & = & \mu_{2m-1}. \end{array} \tag{3.23}$$

This system is not linear in the nodes. It is linear in the coefficients, but with a Vandermonde system matrix, which is known to have *conditioning* (stability) problems. Solving this system is not an easy task. Even when a solution can be found (numerically), it is possible that some of the nodes are complex, or have values outside the interval $[a, b]$. Which is why we use another approach, one that involves orthogonal polynomials.

3.3.2 Orthogonal Polynomials

The use of orthogonal polynomials is justified by the following result.

Theorem 3.9. *Let* $u(x) = (x - x_1)(x - x_2)\dots(x - x_m)$. *Then, the quadrature formula* (3.15) *is exact for all polynomials* $p \in \mathbb{P}_{2m-1}$ *if and only if* u *is orthogonal to the set* $\mathbb{P}_{m-1}$, $u \perp \mathbb{P}_{m-1}$, *with* *respect to the inner product*

$$< f, g >_w \;=\; \int_a^b w(x)f(x)g(x)\,dx. \tag{3.24}$$

Proof.
"$\Rightarrow$"
Let $p \in \mathbb{P}_{m-1}$. Since u has degree m, it follows that $up \in \mathbb{P}_{2m-1}$, so formula (3.15) is exact for up,

$$\int_a^b w(x)u(x)p(x)\,dx \;=\; \sum_{k=1}^{m} A_k u(x_k)p(x_k) \;=\; 0,$$

because $u(x_k) = 0, \forall k = \overline{1,m}$. Hence, $u \perp p$ and, further, $u \perp \mathbb{P}_{m-1}$.

"$\Leftarrow$"
Let $f \in \mathbb{P}_{2m-1}$, arbitrary. By the division algorithm, there exist $q, r \in \mathbb{P}_{m-1}$ such that $f \;=\; uq + r$. Thus, we have

$$\int_a^b w(x)f(x)\,dx \;=\; \int_a^b w(x)u(x)q(x)\,dx + \int_a^b w(x)r(x)\,dx \;=\; 0 + \int_a^b w(x)r(x)\,dx,$$

since $u \perp q$.

Now, formula (3.15) is an interpolatory one, and as such, has degree of exactness at least $d = m-1$. Since $r \in \mathbb{P}_{m-1}$, we have

$$\int_a^b w(x)r(x)\,dx \;=\; \sum_{k=1}^{m} A_k r(x_k).$$

But for any $k = \overline{1,m}$, $f(x_k) \;=\; u(x_k)q(x_k) + r(x_k) \;=\; r(x_k)$ and, thus,

$$\int_a^b w(x)f(x)\,dx \;=\; \sum_{k=1}^{m} A_k f(x_k),$$

i.e. formula (3.15) is exact for every $f \in \mathbb{P}_{2m-1}$. □

Remark 3.10. So we now know that the nodes of a Gaussian quadrature are the roots of a poly-

nomial orthogonal to $\mathbb{P}_{m-1}$ with respect to the weight w. Such families of orthogonal polynomials have been studied extensively. Table 3 contains such examples. A few immediate conclusions:
1. A first consequence is the fact that all the nodes in (3.15) are real, distinct and interior to the interval (a, b).
2. There exists a linear recurrence relation between 3 consecutive monic orthogonal polynomials on the interval $[a, b]$ with respect to the weight w:

$$\pi_{k+1}(t) \;=\; (t-\alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t),\; k=0,1,\dots,\;\; \pi_{-1}(t) \;=\; 0,\;\; \pi_0(t) \;=\; 1, \quad (3.25)$$

where

$$\alpha_k \;=\; \frac{< t\pi_k, \pi_k >}{||\pi_k||^2},\; k=0,1,\dots,\;\; \beta_k \;=\; \frac{||\pi_k||^2}{||\pi_{k-1}||^2},\; k=1,2,\dots,\;\; \beta_0 \;=\; \mu_0. \quad (3.26)$$

Name	**Notation**	**Polynomial**	**Weight fn.**	**Interval**	$\boldsymbol{\alpha_k}$	$\boldsymbol{\beta_k}$
Legendre	l_m	$\left[(x^2-1)^m\right]^{(m)}$	1	$[-1,1]$	0	$\beta_0 = 2$, $\beta_k = (4-k^{-2})^{-1}, k \geq 1$
Chebyshev 1st	T_m	$\cos(m \arccos x)$	$(1-x^2)^{-\frac{1}{2}}$	$[-1,1]$	0	$\beta_0 = \pi$, $\beta_1 = \frac{1}{2}$, $\beta_k = \frac{1}{4}, k \geq 2$
Chebyshev 2nd	Q_m	$\dfrac{\sin[(m+1)\arccos x]}{\sqrt{1-x^2}}$	$(1-x^2)^{\frac{1}{2}}$	$[-1,1]$	0	$\beta_0 = \frac{\pi}{2}$, $\beta_k = \frac{1}{4}, k \geq 1$
Laguerre	L_m^a	$x^{-a}e^x\left(x^{m+a}e^{-x}\right)^{(m)}$	$x^a e^{-x},\ a > -1$	$[0,\infty)$	$2k+a+1$	$\beta_0 = \Gamma(1+a)$, $\beta_k = k(k+a), k \geq 1$
Hermite	H_m	$(-1)^m e^{x^2}\left(e^{-x^2}\right)^{(m)}$	e^{-x^2}	$\mathbb{R}$	0	$\beta_0 = \sqrt{\pi}$, $\beta_k = \frac{k}{2}, k \geq 1$

Table 3: Orthogonal polynomials and recurrence coefficients

Example 3.11. Now we can solve system (3.19) much easier.

Solution. We know that the nodes are the roots of the Legendre polynomial

$$l_2(x) \;=\; \left[(x^2-1)^2\right]'' \;=\; \left[4x(x^2-1)\right]' \;=\; 4(3x^2-1),$$

i.e. $\pm\dfrac{\sqrt{3}}{3}$. Then the coefficients $A_0 = A_1 = 1$ are immediately found.

$\blacksquare$

Other properties of Gauss quadratures:

Proposition 3.12. *The coefficients* $A_k, k = \overline{1,m}$ *in* (3.15) *are positive.*

Proof. Recall Lagrange fundamental polynomials

$$l_i(x) \;=\; \frac{u_i(x)}{u_i(x_i)} \;=\; \frac{(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_m)}{(x_i-x_1)\dots(x_i-x_{i-1})(x-x_{i+1})\dots(x_i-x_m)},\ i=1,\dots,m.$$

This polynomial has degree $m-1$ and $l_i(x_k) = \delta_{i,k}$. Then the degree of l_i^2 is $2m-2$ and $l_i^2(x_k) = \delta_{i,k}$. Hence, formula (3.15) is exact for $f = l_i^2, i = \overline{1,m}$ and we have

$$0 \;<\; \int\limits_a^b w(x)l_i^2(x)\,dx \;=\; \sum_{k=1}^m A_k l_i^2(x_k) \;=\; A_i.$$

□

Regarding the convergence of Gaussian quadratures, we have:

Theorem 3.13. *If* $[a,b]$ *is bounded and* $f \in C[a,b]$*, then the Gaussian formula* (3.15) *converges,* $R_m(f) \to 0,\ m \to \infty$.

The proof is based on Weierstrass' theorem.

For the remainder of the quadrature formula, the following holds:

Proposition 3.14. *If* $f \in C^{2m}[a,b]$*, then there exists* $\xi \in (a,b)$ *such that*

$$R_m(f) \;=\; \frac{f^{(2m)}(\xi)}{(2m)!}\int\limits_a^b w(x)u^2(x)\,dx. \tag{3.27}$$

Proof. Consider the Hermite interpolation polynomial at the double nodes $x_1,\dots,x_m$ (so a polynomial of degree $2m-1$, for which the Gaussian formula is exact) and its remainder in Newton's form. We have

$$f(x) \;=\; (H_{2m-1}f)(x) + f[x,x_1,x_1,\dots,x_m,x_m]u^2(x).$$

Multiply by $w(x)$ and integrate, keeping in mind that $(H_{2m-1}f)(x_k) = f(x_k), k = \overline{1,m}$.

$$\int\limits_a^b w(x)f(x)\,dx \;=\; \int\limits_a^b w(x)(H_{2m-1}f)(x)\,dx + \int\limits_a^b w(x)u^2(x)f[x,x_1,x_1,\dots,x_m,x_m]\,dx$$

$$= \sum_{k=1}^{m} A_k f(x_k) + \frac{f^{(2m)}(\xi)}{(2m)!} \int_a^b w(x) u^2(x)\, dx,$$

since $w(x)u^2(x) > 0$ on (a, b), so the mean value formula can be used. It follows that the last term represents the remainder of the quadrature formula.

□

Now we have procedures for finding the nodes and coefficients of a Gaussian quadrature formula. However, they are still not enough for an efficient implementation. For that, we will make use of the recurrence relation and the parameters in (3.25)–(3.26). With these, we define

$$J_m(w) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{m-1}} \\ 0 & & & \sqrt{\beta_{m-1}} & \alpha_{m-1} \end{bmatrix}, \tag{3.28}$$

called the **Jacobi matrix** of order m for the weight function w on the interval $[a, b]$. The following holds:

Theorem 3.15. *The nodes $\{x_k\}_{k=1}^m$ of the Gaussian formula* (3.15) *are the eigenvalues of J_m,*

$$J_m v_k = x_k v_k, \; v_k^T v_k = 1, \; k = 1, \ldots, m, \tag{3.29}$$

while the coefficients $\{A_k\}_{k=1}^m$ are given by

$$A_k = \beta_0 v_{k,1}^2, \; k = 1, \ldots, m, \tag{3.30}$$

where $v_{k,1}$ is the first component of the normalized ($||v_k|| = 1$) eigenvector associated with the eigenvalue x_k.

This is easily proved by writing the recurrence relation (3.25) in matrix (vector) form.

Remark 3.16. Thus, the problem of determining a Gauss numerical integration formula is now reduced to that of finding e-values and e-vectors for a *symmetric* and *tridiagonal* matrix. This problem has been studied extensively in linear algebra, there is a vast literature on it and there are many very efficient methods for solving it.