

CONTENTS

1. Numerical integration: basic rules	1
1.1. General formulas for the error on numerical quadrature	1
1.2. Particular cases	3
2. A birds'-eye view on interpolatory quadrature rules	5
3. Composite rules	6
3.1. Composite trapezoidal rule	6
4. The Romberg integration	9
4.1. Composite Simpson's rule	9
4.2. Obtaining composite Simpson's rule by Richardson extrapolation	10
5. Adaptive Simpson's rule	11
6. Gaussian quadrature	14
6.1. Maximizing the degree of exactness	15
6.2. Orthogonal polynomials	16
6.3. Three Term Recurrence Relationships	18
6.4. The exactness of Gaussian quadrature	20
6.5. Error estimate	22
6.6. Example: Gauss-Chebyshev quadrature	22
6.7. A naive way for finding nodes and weights	23
6.8. Golub-Welsch algorithm for finding nodes and weights	24

1. NUMERICAL INTEGRATION: BASIC RULES

References:

- S. D. Conte, Carl de Boor, *Elementary Numerical Analysis: an algorithmic approach, Second Edition, McGraw-Hill Book Company, 1972*

The problem of numerical integration, or numerical quadrature, is to estimate

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

This problem arises when the integration cannot be carried out exactly or when $f(x)$ is known only at a finite number of points.

Numerical integration is a stable process, i.e., if a function f is perturbed by δf , then the perturbation of the output of numerical integration over an interval $[a, b]$ is perturbed by at most $(b - a) \max_{[a,b]} |\delta f|$, i.e., the error in the output is bounded by the maximal error in input times the length of the interval of integration.

1.1. General formulas for the error on numerical quadrature. We will derive a collection of basic quadrature rules. We assume that the integrand $f(x)$ is sufficiently smooth on some interval $[c, d]$ containing $[a, b]$ so that we can write

$$f(x) = p_k(x) + f[x_0, \dots, x_k, x]\pi_{k+1}(x)$$

where $\pi_{k+1}(x) = (x - x_0) \dots (x - x_k)$ is the nodal polynomial. We estimate $I(f)$ by $I(p_k)$. Then the error of this estimate is

$$(1) \quad E(f) = I(f) - I(p_k) = \int_a^b f[x_0, \dots, f_k, x] \pi_{k+1}(x) dx.$$

The divided difference $f[x_0, \dots, f_k, x]$ is continuous and hence integrable function of x . In the chapter on interpolation we have learned that

$$(2) \quad f[x_0, \dots, f_k, x] = \frac{f^{(k+1)}(\zeta)}{(k+1)!}, \quad \zeta \in (a, b).$$

where $\zeta \in (a, b)$ is some point depending on x .

We consider two cases in which the integral in the right-hand side of (1) can be simplified.

Case 1: $\pi_{k+1}(x)$ is of one sign. The first such case is where the nodal polynomial $\pi_{k+1}(x)$ is of one sign. Then by the mean value theorem

$$(3) \quad \int_a^b f[x_0, \dots, x_k, x] \pi_{k+1}(x) dx = f[x_0, \dots, x_k, \xi] \int_a^b \pi_{k+1}(x) dx \text{ for some } \xi \in (a, b).$$

Exercise Think of a simple example showing that if $g(x)$ is not of one sign then

$$\int_a^b f(x)g(x)dx \neq f(\xi) \int_a^b g(x)dx \text{ for any } \xi \in (a, b).$$

If $f(x)$ is $k+1$ times continuously differentiable then, putting together (2) and (3), we obtain the integration error:

$$(4) \quad E(f) = \frac{f^{(k+1)}(\eta)}{(k+1)!} \int_a^b \pi_{k+1}(x) dx.$$

Case 2:

$$(5) \quad \int_a^b \pi_{k+1}(x) dx = 0 \text{ and } \pi_{k+2}(x) := (x - x_{k+1})\pi_{k+1}(x) \text{ is of one sign.}$$

Then we can make use of the identity

$$(6) \quad f[x_0, \dots, x_k, x] = f[x_0, \dots, x_k, x_{k+1}] + f[x_0, \dots, x_{k+1}, x](x - x_{k+1})$$

which is valid for an arbitrary x_{k+1} . This identity comes from the definition of the divided difference and the fact that divided differences are symmetric with respect to their arguments:

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

Exercise Show that the divided difference $f[x_0, \dots, x_k]$ is a symmetric function of its arguments, i.e., for an arbitrary permutation σ of $\{0, \dots, k\}$, $f[x_0, \dots, x_k] = f[x_{\sigma(0)}, \dots, x_{\sigma(k)}]$. *Hint: Think of the leading coefficient of Newton's interpolant.*

Using identity (36) we get

$$\begin{aligned} E(f) &= \int_a^b f[x_0, \dots, x_k, x] \pi_{k+1}(x) dx \\ &= \int_a^b f[x_0, \dots, x_k, x_{k+1}] \pi_{k+1}(x) dx + \int_a^b f[x_0, \dots, x_{k+1}, x] (x - x_{k+1}) \pi_{k+1}(x) dx. \end{aligned}$$

Note that $f[x_0, \dots, x_k, x_{k+1}]$ is independent of x . Hence we can take it outside the integral and use the fact that $\int_a^b \pi_{k+1}(x) dx = 0$ by assumption. Then we get the following expression for the integration error:

$$E(f) = \int_a^b f[x_0, \dots, x_{k+1}, x] (x - x_{k+1}) \pi_{k+1}(x) dx.$$

Now, if we choose x_{k+1} in such a way that

$$\pi_{k+2}(x) \equiv (x - x_{k+1}) \pi_{k+1}(x)$$

is of one sign on (a, b) , and if $f(x)$ is $k + 2$ times continuously differentiable, then it follows that

$$(7) \quad \boxed{E(f) = \frac{f^{(k+2)}(\eta)}{(k+2)!} \int_a^b \pi_{k+2}(x) dx.}$$

1.2. Particular cases. Now we go over some particular values of k and the corresponding quadrature rules.

1.2.1. Rectangle rules. We set $k = 0$. Then

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

Hence

$$I(p_0) = (b - a)f(x_0).$$

- If $x_0 = a$, we have the **left-hand rule**. Noting that the nodal polynomial is of one sign we obtain

$$(8) \quad \boxed{I(f) \approx L = (b - a)f(a), \quad E^L = f'(\eta) \int_a^b (x - a) dx = \frac{1}{2} f'(\eta)(b - a)^2.}$$

- If $x_0 = b$, we have the **right-hand rule**. Noting that the nodal polynomial is of one sign we obtain

$$(9) \quad \boxed{I(f) \approx R = (b - a)f(b), \quad E^R = f'(\eta) \int_a^b (x - b) dx = -\frac{1}{2} f'(\eta)(b - a)^2.}$$

1.2.2. *Midpoint rule.* We set $k = 0$ and pick $x_0 = \frac{a+b}{2}$. Then $\pi_1(x)$ is not of one sign but $\int_a^b \pi_1(x)dx = \int_a^b (x - x_0)dx = 0$, while $\int_a^b (x - x_0)^2 dx$ is of one sign. Hence the error in $I(p_0)$ can be computed by Eq. (7) with $x_1 = x_0$. This leads to the **midpoint rule**

$$(10) \quad \boxed{I(f) \approx M = (b - a)f\left(\frac{a + b}{2}\right), \quad E^M = \frac{1}{24}f''(\eta)(b - a)^3.}$$

1.2.3. *Trapezoid rule.* We set $\mathbf{k} = \mathbf{1}$. Then

$$f(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x]\pi_2(x).$$

Let $x_0 = a$ and $x_1 = b$. Then $\pi_2(x) = (x - a)(x - b)$ is of one sign on (a, b) . Hence

$$I(f) = \int_a^b \{f(a) + f[a, b](x - a)\}dx + \frac{1}{2}f''(\eta) \int_a^b (x - a)(x - b)dx.$$

Then we get the **trapezoid rule**

$$(11) \quad \boxed{I(f) \approx T = \frac{f(a) + f(b)}{2}(b - a), \quad E^T = -\frac{1}{12}f''(\eta)(b - a)^3.}$$

1.2.4. *Simpson's rule.* Let us choose $\mathbf{k} = \mathbf{2}$. Then

$$f(x) = p_2(x) + f[x_0, x_1, x_2, x]\pi_3(x).$$

Set $x_0 = a$, $x_1 = \frac{1}{2}(a + b)$, $x_2 = b$. Then

$$\int_a^b \pi_3(x)dx = 0.$$

If we pick $x_3 = x_1 = \frac{a+b}{2}$, then π_4 is of one sign and the error can be found by (7). As a result, we obtain **Simpson's rule**

$$(12) \quad \boxed{I(f) \approx S = \frac{b - a}{6} \left[f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right],}$$

$$(13) \quad \boxed{E^S = -\frac{1}{90}f^{(iv)}(\eta) \left(\frac{b - a}{2}\right)^5.}$$

1.2.5. *Corrected trapezoid rule.* Now we set $\mathbf{k} = \mathbf{3}$. In this case we have

$$f(x) = p_3(x) + f[x_0, x_1, x_2, x_3]\pi_4(x).$$

In order to make $\pi_4(x)$ of one sign on $[a, b]$ we set $x_0 = x_1 = a$ and $x_2 = x_3 = b$. Then

$$E(f) = \frac{1}{4!}f^{(iv)}(\eta) \int_a^b (x - a)^2(x - b)^2 dx = \frac{1}{720}f^{(iv)}(\eta)(b - a)^5.$$

Since $p_3^{(iv)}(x) = 0$, we can use the Simpson rule to evaluate $I(p_3)$ exactly. One can find $p_3(a) = f(a)$, $p_3(b) = b$, and

$$p_3\left(\frac{a+b}{2}\right) = \frac{1}{2}[f(a) + f(b)] + \frac{b-a}{8}[f'(a) - f'(b)].$$

Hence

$$(14) \quad I(f) \approx CT = \frac{b-a}{2}[f(a) + f(b)] + \frac{(b-a)^2}{12}[f'(a) - f'(b)],$$

$$(15) \quad E^{CT} = \frac{1}{720}f^{(iv)}(\eta)(b-a)^5.$$

This rule is called the **corrected trapezoidal rule**.

Exercise Work out the details of the derivation of the basic integration rules, Eqs. (9), (10), (11), (12) and (13), and (14) and (15).

2. A BIRDS'-EYE VIEW ON INTERPOLATORY QUADRATURE RULES

Reference:

- A. Gil, J. Segura, N. Temme, *Numerical Methods for Special Functions*, SIAM, 2007 (available online via UMD library). See Chapter 5.

Now let us take a more general look at the quadrature rules. All quadrature rules discussed in Section 1.2 except for the corrected trapezoid rule are of the form

$$(16) \quad \int_a^b f(x)dx \approx Q(f) = \sum_{i=0}^n w_i f(x_i).$$

The numbers w_i are called the **weights** and the values x_i are called the **nodes**. All of those rules are *interpolatory*.

Definition 1. We call a quadrature rule *interpolatory* if $f(x)$ is approximated by an n -th degree interpolating polynomial, and the rule is set up as below:

$$\int_a^b f(x)dx \approx \int_a^b p_n(x)dx = \int_a^b \sum_{i=0}^n f(x_i)L_i(x)dx = \sum_{i=0}^n w_i f(x_i),$$

where

$$w_i = \int_a^b L_i(x)dx \equiv \int_a^b \prod_{\substack{k=0 \\ k \neq i}}^n \frac{x - x_k}{x_i - x_k} dx.$$

Definition 2. The *Newton-Cotes quadrature rules* are the *interpolatory quadrature rules* with *equispaced interpolation nodes* (a. k. a. *abscissas*).

For $n = 1$ the Newton-Cotes rule is the trapezoidal rule. For $n = 2$ the Newton-Cotes rule is Simpson's rule. Considering Newton-Cotes rules for a high number of nodes is not very practical. Recall the Runge phenomenon, i.e., the fact that the interpolation polynomial with equispaced nodes does not necessarily converge to the function in the sense of the maximum norm as the number of nodes tends to infinity!

If the goal is to design a quadrature rule for fixed n that is exact for the highest degree polynomials, the *Gaussian quadrature* is the best choice. If one wants to reduce the error of integration, the better options are composite rules and Romberg integration.

Definition 3. A quadrature rule has degree of exactness m if it renders exact results when $f(x)$ is any polynomial of degree not larger than m but it is not exact for at least one polynomial of degree $m + 1$.

The degree of exactness of an n -point Newton-Cotes rule is at least $n - 1$. For the trapezoidal rule ($n = 2$) and the Simpson rule ($n = 3$), the degrees of exactness are 1 and 3 respectively. As we will see later, the degree of exactness of an n -point Gaussian quadrature is $2n - 1$.

3. COMPOSITE RULES

Reference:

- [A. Gil, J. Segura, N. Temme, Numerical Methods for Special Functions, SIAM, 2007](#) (available online via UMD library). See Chapter 5 for the basic facts and Chapter 11 for details on Bernoulli numbers and the Euler-Maclaren summation formula.

3.1. Composite trapezoidal rule. Let us consider an equally spaced partition of the interval $[a, b]$:

$$a = x_0 < x_1 < \dots < x_n = b, \quad x_k = a + kh, \quad h = \frac{b-a}{n}.$$

If we apply the trapezoidal rule to each subinterval, we obtain

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx \approx T_n(x) = \frac{h}{2}(f_0 + f_n) + h \sum_{i=1}^{n-1} f(x_i).$$

Theorem 1. Let $f(x) \in C^2[a, b]$ and let $a = x_0 < x_1 < \dots < x_n = b$, $x_k = a + kh$, $h = \frac{b-a}{n}$, be an equally spaced partition of $[a, b]$. Then

$$(17) \quad \int_a^b f(x)dx = \frac{h}{2}(f_0 + f_n) + h \sum_{i=1}^{n-1} f(x_i) + E_n^T,$$

where $f_i = f(x_i)$ and there is $\tau \in [a, b]$ such that

$$(18) \quad E_n^T = -\frac{(b-a)h^2}{12} f''(\tau).$$

Proof. It immediately follows from the trapezoidal rule that

$$E_n^T = -\frac{h^2}{12} \sum_{i=1}^n f''(\eta_i) = -\frac{(b-a)h^2}{12} \frac{1}{n} \sum_{i=1}^n f''(\eta_i),$$

where $\eta_i \in [x_{i-1}, x_i]$, $i = 1, 2, \dots, n$. Observing that

$$\min_{[a,b]} f''(x) \leq \frac{1}{n} \sum_{i=1}^n f''(\eta_i) \leq \max_{[a,b]} f''(x)$$

and applying the intermediate value theorem, we obtain that there is $\tau \in [a, b]$ such that $f''(\tau) = \frac{1}{n} \sum_{i=1}^n f''(\eta_i)$. Then the result follows. \square

One can check that the composite corrected trapezoidal rule is given by

$$\begin{aligned} \int_a^b f(x)dx &= CT_n(f) + E_n^{CT}(f) \\ (19) \quad &= \frac{h}{2}(f_0 + f_n) + h \sum_{i=1}^{n-1} f(x_i) + \frac{h^2}{12}[f'(a) - f'(b)] + \frac{b-a}{720}h^4 f^{(iv)}(\eta). \end{aligned}$$

Comparing equations (17)–(18) and (19) we conclude that the major contribution to the error of the composite trapezoidal rule is

$$E_n^T(f) \approx \frac{h^2}{12}[f'(a) - f'(b)].$$

Therefore, if

$$f \in C^4([a, b]) \quad \text{and} \quad f'(a) = f'(b)$$

then the error of integration by the composite trapezoid rule decays as $O(h^4)$ rather than $O(h^2)$!

Example The Bessel function $J_0(x)$ is defined by

$$(20) \quad \pi J_0(x) = \int_0^\pi \cos(x \sin(t))dt = h + h \sum_{j=1}^{n-1} \cos(x \sin(hj)) + E_n^T \quad h = \pi/n.$$

The relative errors of integration by the composite trapezoid rule are given in Table 1. We

TABLE 1. The relative errors using the composite trapezoidal rule for Eq. (20) for $x = 5$.

n	E_n^T
4	-0.12e-0
8	-0.48e-6
16	-0.11e-21
32	-0.13e-62
64	-0.13e-163
128	-0.53e-404

observe that the error is much smaller than we have expected taking into account the fact that $f'(0) = f'(\pi)$.

This example suggests that the composite trapezoidal rule can do much better than expected for certain types of integrals. The crucial result to understand this behavior is the **Euler-Maclaurin formula** (equation (21) below).

Theorem 2. *Let $f(x) \in C^{2m+2}[x_0, x_n]$. Then*

$$\int_{x_0}^{x_n} f(x)dx = T_n(f) + R_n(f)$$

where the truncation error R_n admits the expansion

$$(21) \quad R_n(f) = \sum_{l=1}^m \frac{B_{2l}}{(2l)!} h^{2l} \left(f^{(2l-1)}(x_0) - f^{(2l-1)}(x_n) \right) - \frac{B_{2m+2}}{(2m+2)!} (x_n - x_0) h^{2m+2} f^{(2m+2)}(\zeta)$$

where $\zeta \in [x_0, x_n]$ and B_k are the Bernoulli numbers.

The blue text explains what are the Bernoulli numbers and where they come from. All what is important for us is that they are bounded – se the last line of the blue text.

The Bernoulli polynomials are defined by the generating function

$$(22) \quad \frac{ze^{xz}}{e^z - 1} = \sum_{n=0}^{\infty} \frac{B_n(x)}{n!} z^n, \quad |z| < 2\pi.$$

The Bernoulli numbers, also denoted by B_n but with no argument, are the values of the Bernoulli polynomials at $x = 0$. This means that they can be found from the equality obtained from Eq. (22) by plugging in $x = 0$:

$$(23) \quad \frac{z}{e^z - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} z^n, \quad |z| < 2\pi.$$

Observing that

$$\begin{aligned} \frac{z}{e^z - 1} &= \frac{1}{1 + \frac{z}{2} + \frac{z^2}{6} + \frac{z^3}{24} + \dots + \frac{z^k}{(k+1)!} + \dots} \\ &= 1 - \left(\sum_{k=1}^{\infty} \frac{z^k}{(k+1)!} \right) + \left(\sum_{k=1}^{\infty} \frac{z^k}{(k+1)!} \right)^2 - \left(\sum_{k=1}^{\infty} \frac{z^k}{(k+1)!} \right)^3 + \dots \\ &= 1 - \frac{z}{2} + \frac{z^2}{12} + \dots \\ &= B_0 + B_1 z + B_2 \frac{z^2}{2} + B_3 \frac{z^3}{6} + \dots, \end{aligned}$$

we find $B_0 = 1$, $B_1 = -\frac{1}{2}$, $B_2 = \frac{1}{6}$, \dots . Check that

$$\frac{z}{e^z - 1} = 1 + \frac{1}{2}z$$

is even. Therefore, we have

$$B_{2n+1} = 0, \quad n = 1, 2, 3, \dots$$

The first nonvanishing numbers are

$$B_0 = 1, \quad B_1 = -\frac{1}{2}, \quad B_2 = \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \dots$$

Theorem 2 has a very important implication of the convergence of the composite trapezoid rule.

Corollary 3. *If $f \in C^k(\mathbb{R})$ is periodic and the integral is taken over the full period, then*

$$|R_n| = O(n^{-k}), \quad k \rightarrow \infty.$$

If the function is infinitely differentiable on $[a, b]$ and either periodic with period $(b - a)$, or if the function vanishes at a and b together with all of its derivatives then the integration error of the composite trapezoid rule decays faster than any negative power of n where n is the number of nodes. In this case we say that *the error decays exponentially*.

4. THE ROMBERG INTEGRATION

The Romberg integration is Richardson's extrapolation applied to integration. We will show that composite Simpson's rule is the result of one step of Richardson extrapolation applied to the composite trapezoid rule. First we discuss composite Simpson's rule.

4.1. Composite Simpson's rule. Composite Simpson's rule is obtained and analyzed in a similar manner as the composite trapezoid rule.

Theorem 4. *Let $f(x) \in C^4[a, b]$ and let $a = x_0 < x_1 < \dots < x_n = b$, $x_i = x_0 + jh$, $i = 0, 1, 2, \dots, n = 2m$, $h = (b - a)/n$. Then*

$$(24) \quad \int_a^b f(x) dx = S_n(f) + E_n^S(f),$$

where

$$(25) \quad S_n(f) = \frac{1}{3}h(f_0 + f_n) + \frac{2}{3}h \sum_{j=1}^{m-1} f_{2j} + \frac{4}{3}h \sum_{j=1}^m f_{2j-1},$$

where $f_j = f(x_j)$ and there is $\tau \in [a, b]$ such that

$$(26) \quad E_n^S(f) = -\frac{(b-a)h^4}{180} f^{(iv)}(\tau).$$

Remark The error can be estimated for large n as

$$E_n^S(f) \approx \hat{E}_n^S(f) = \frac{h^4}{180} (f'''(a) - f'''(b))$$

in the sense that if $f'''(a) \neq f'''(b)$ then

$$\lim_{n \rightarrow \infty} \frac{\hat{E}_n^S(f)}{E_n^S(f)} = 1.$$

4.2. Obtaining composite Simpson's rule by Richardson extrapolation. Let $f(x)$ be a smooth function. Then for the composite trapezoid rule with step h we have:

$$(27) \quad T_h(f) \equiv \frac{1}{2}h[f(a) + f(b)] + h \sum_{j=1}^{n-1} f_j = \int_a^b f(x)dx + c_1h^2 + c_2h^4 + O(h^6),$$

where the coefficients c_1 and c_2 can be extracted from the Euler-Maclaurin summation formula:

$$c_1 = -\frac{1}{12}(f'(b) - f'(a)), \quad c_2 = \frac{1}{720}(f'''(b) - f'''(a)).$$

Suppose $h = (b - a)/n$ where n is even, i.e., $n = 2m$. Then for the composite trapezoid rule with step $2h$ we have

$$(28) \quad T_{2h}(f) \equiv \frac{1}{2}2h[f(a) + f(b)] + 2h \sum_{j=1}^{m-1} f_{2j} = \int_a^b f(x)dx + c_1(2h)^2 + c_2(2h)^4 + O(h^6),$$

Therefore, we obtain the following system of equations

$$\begin{aligned} T_h(f) &= I(f) + c_1h^2 + c_2h^4 + O(h^6), \\ T_{2h}(f) &= I(f) + 4c_1h^2 + 16c_2h^4 + O(h^6), \end{aligned}$$

Multiplying the first equation by 4 and subtracting the second one from it we get

$$4T_h(f) - T_{2h}(f) = 3I(f) - 12c_2h^4 + O(h^6).$$

Hence

$$I(f) = \frac{4}{3}T_h(f) - \frac{1}{3}T_{2h}(f) + 4c_2h^4 + O(h^6).$$

Plugging in the expressions for the trapezoid rule and the coefficient c_2 we obtain

$$\begin{aligned} I(f) &= \frac{3}{4} \left(\frac{1}{2}h[f(a) + f(b)] + h \sum_{j=1}^{m-1} f_{2j} + h \sum_{j=1}^m f_{2j-1} \right) \\ &\quad - \frac{1}{3} \left(h[f(a) + f(b)] + 2h \sum_{j=1}^{m-1} f_{2j} \right) + \frac{h^4}{180}(f'''(b) - f'''(a)) + O(h^6) \\ (29) \quad &= \frac{h}{3}[f(a) + f(b)] + \frac{2h}{3} \sum_{j=1}^{m-1} f_{2j} + \frac{4h}{3} \sum_{j=1}^m f_{2j-1} + \frac{1}{180}h^6(f'''(b) - f'''(a)) + O(h^6). \end{aligned}$$

The last expression coincides with composite Simpson's rule.

The fact that composite Simpson's rule is obtained from the composite trapezoid rule by Richardson's extrapolation proves that composite Simpson's rule also has an asymptotic error expansion, and its coefficients can be obtained from the Euler-Maclaurin formula. This means that all corollaries that we have drawn from the Euler-Maclaurin formula for the composite trapezoid rule are also valid for composite Simpson's rule.

5. ADAPTIVE SIMPSON'S RULE

Reference:

- [Wiki: Adaptive Simpson's method](#)

Adaptive Simpson's method, also called adaptive Simpson's rule, is a method of numerical integration proposed by G.F. Kuncir in 1962. Adaptive Simpson's method uses an estimate of the error we get from calculating a definite integral using Simpson's rule. If the error exceeds a user-specified tolerance, the algorithm subdivides the interval of integration in two and applies adaptive Simpson's method to each subinterval in a recursive manner. The technique is usually much more efficient than composite Simpson's rule since it uses fewer function evaluations in places where the function is well-approximated by a cubic function. A criterion for determining when to stop subdividing an interval, suggested by J.N. Lyness, is

$$(30) \quad \frac{1}{15} |S(a, c) + S(c, b) - S(a, b)| < \epsilon,$$

where $[a, b]$ is an interval with midpoint c , $S(a, b)$, $S(a, c)$, and $S(c, b)$ are the estimates given by Simpson's rule on the corresponding intervals and ϵ is the desired tolerance for the interval.

Simpson's rule is an interpolatory quadrature rule with three points: the endpoints a and b , and the midpoint c . Subdividing $[a, b]$ to two equal subintervals $[a, c]$ and $c, b]$ and using one step of Richardson extrapolation, we obtain a more accurate five-point quadrature rule. Using Richardson extrapolation, a more accurate estimate for the integral $I(f)$ involving five function values is combined with a less accurate estimate for three function values by applying the correction (see Fig. 1). Simpson's rule is exact for polynomials of degree ≤ 3 . The resulting five-point rule is exact for polynomials of degree ≤ 5 .

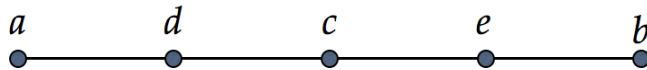


FIGURE 1. The interval $[a, b]$, $[a, c]$ and $[c, b]$ and their midpoints for adaptive Simpson's rule.

Now we derive the criterion (30) and justify the the statement about exactness for the polynomials of degree ≤ 5 . Indeed, since the Simpson rule is obtained from the trapezoidal rule by one step of Richardson extrapolation, it has error expansion of the form

$$(31) \quad I(f) := \int_a^b f(x)dx = S(a, b) + \alpha_4 h^4 + \alpha_6 h^6 + \dots,$$

where $h = b - a$,

$$S(a, b) = \frac{1}{6}h[f(a) + 4f(c) + f(b)],$$

and the coefficients α_n are of the form

$$\alpha_n = \gamma_n(f^{(n)}(a) - f^{(n)}(b)).$$

Making the step twice as small we get

$$(32) \quad I(f) := \int_a^b f(x)dx = S(a, c) + S(c, b) + \alpha_4 \frac{h^4}{16} + \alpha_6 \frac{h^6}{64} + \dots$$

Multiplying Eq. (37) by 16 and subtracting Eq. (36) from it we obtain

$$15I(f) = 15[S(a, c) + S(c, b)] + [S(a, c) + S(c, b) - S(a, b)] + \hat{\alpha}_6 h^6 + \dots$$

Therefore

$$(33) \quad I(f) = S(a, c) + S(c, b) + \frac{1}{15}[S(a, c) + S(c, b) - S(a, b)] + \beta_6 h^6 + \dots$$

Comparing Eqs. (37) and (43) we see that the error in Eq. (37) is approximately given by $\frac{1}{15}[S(a, c) + S(c, b) - S(a, b)]$. Therefore, Eq. (30) is justified. Furthermore, the coefficient β_n is of the form

$$\beta_n = \delta_n(f^{(n)}(a) - f^{(n)}(b)).$$

Hence, the five point integration rule given by

$$(34) \quad Q(f) = S(a, c) + S(c, b) + \frac{1}{15}[S(a, c) + S(c, b) - S(a, b)]$$

is exact on all polynomials of degree ≤ 5 .

The program `AdaptiveSimpson.m` (see below) implements adaptive Simpson's method. This is a recursive algorithm refining intervals whenever the error tolerance is not satisfied. To avoid infinite recursion, the maximal recursion depth is introduced. Integration nodes of $f(x) = \sin(1 - 30x^2)$ over the interval $[0, 1]$ with the tolerance `tol=1.0e-4` is shown in Fig. 2. There is a total of 109 nodes. They are shown both on the graph and on the x -axis.

```
% This program integrates the function that you define in myf(x)
% using the adaptive Simpson rule.
%
function AdaptiveSimpson()
clear all
global count nodes
a = -5; b = 5; % the endpoints of the interval
tol = 1.0e-4; % tolerance
maxRecursionDepth = 40; % Maximal depth of recursion
    nodes(1) = a;
    nodes(2) = b;
count = 2; % count of the number of nodes
c = (a + b)/2;
```

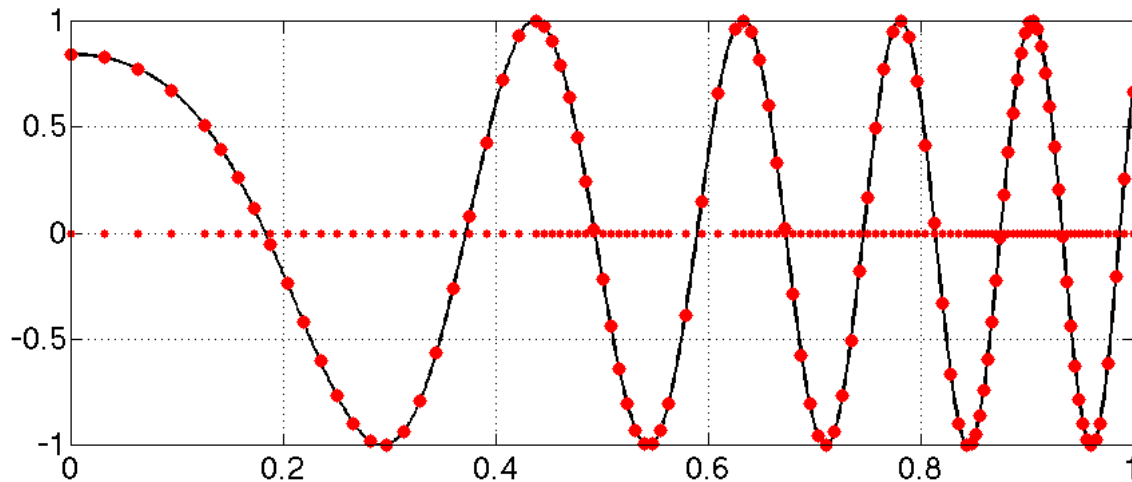


FIGURE 2. Adaptive Simpson's method applied to integrate $f(x) = \sin(1 - 30x^2)$ over the interval $[0, 1]$.

```

    h = b - a;
fa = myf(a);
    fb = myf(b);
    fc = myf(c);
S = (h/6)*(fa + 4*fc + fb);
count=count+1;
    nodes(count) = c;
I = adaptiveSimpsonsAux( a, b, tol, S, fa, fb, fc, maxRecursionDepth );
fprintf('I = %.16e\n',I); % print the result
fprintf('# of nodes = %i\n',count);
    %% plot the graph
    figure;
    hold on;
    grid;
    x = linspace(a,b,1000);
    y = myf(x);
    plot(x,y,'LineWidth',2);
    plot(nodes,myf(nodes),'o','MarkerEdgeColor','k',...
        'MarkerSize',9,'MarkerFaceColor','r')
    plot(nodes,zeros(size(nodes)),'o','MarkerEdgeColor','k',...
        'MarkerSize',9,'MarkerFaceColor','b')
    set(gca,'FontSize',16);

```

14

end

```
%% Recursive auxiliary function adaptiveSimpsonsAux
function I = adaptiveSimpsonsAux(a, b, tol, S, fa, fb, fc, bottom)
global count nodes
c = (a + b)/2;
    h = b - a;
d = (a + c)/2;
    e = (c + b)/2;
fd = myf(d);
    fe = myf(e);
% fprintf('%.12e\n%.12e\n',d,e);
    count = count + 1;
    nodes(count) = d;
    count = count + 1;
    nodes(count) = e;
Sleft = (h/12)*(fa + 4*fd + fc);
Sright = (h/12)*(fc + 4*fe + fb);
S2 = Sleft + Sright;
if bottom <= 0 | abs(S2 - S) <= 15*tol
if( bottom <= 0 )
        fprintf('bottom is reached\n');
        end
I = S2 + (S2 - S)/15;
    else
I = adaptiveSimpsonsAux( a, c, tol/2, Sleft, fa, fc, fd, bottom-1) + ...
    adaptiveSimpsonsAux( c, b, tol/2, Sright, fc, fb, fe, bottom-1);
    end
end

%% The function to be integrated
function y = myf(x)
y = 1./(x.^2+1);
end
```

6. GAUSSIAN QUADRATURE

Reference:

- [A. Gil, J. Segura, N. Temme, Numerical Methods for Special Functions, SIAM, 2007](#)(available online via UMD library). See Chapter 5 starting from Section 5.3.

Gaussian quadrature is designed for computing integrals of the form

$$(35) \quad I(f) = \int_a^b f(x)w(x)dx,$$

where $w(x)$ is a weight function on $[a, b]$ ($a = -\infty$ and $b = \infty$ are accepted) which means that it is positive in any open interval in $[a, b]$ and

$$\int_a^b w(x)|x|^n dx < \infty, \quad n = 0, 1, 2, \dots$$

Note that if $w(x) \equiv 1$, integral (35) is reduced to the one considered in the previous sections. Integrals of the form (35) with nontrivial $w(x)$ arise in various applications involving special functions. See, for example, [Gamma function](#). The key idea behind the Gaussian quadrature is a clever choice of interpolation nodes in $[a, b]$ that maximizes the degree of exactness of the quadrature rule.

6.1. Maximizing the degree of exactness. In Gaussian quadrature, the endpoints a and b are often not chosen as interpolation nodes. For this reason, we will start the enumeration of interpolation nodes from 1 rather than from 0. As soon as the interpolation nodes $x_1 < x_2 < \dots < x_n$, $x_j \in [a, b]$, $j = 1, 2, \dots, n$, are picked, one can approximate the integral using the standard recipe for interpolating quadrature rules:

$$(36) \quad I(f) = \int_a^b f(x)w(x)dx \approx Q(f) = \int_a^b w(x) \sum_{j=1}^n f(x_j)L_j(x) = \sum_{j=1}^n w_j f(x_j)$$

where

$$(37) \quad w_j = \int_a^b L_j(x)w(x)dx, \quad L_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^n \frac{(x - x_k)}{(x_j - x_k)}.$$

Independent of the choice of nodes, such a rule is exact for polynomials of degree $\leq n - 1$.

Definition 4. We say a quadrature rule of the form of Eq. (36) has degree of exactness m if it is exact for all polynomials $f(x)$ of degree $\leq m$ but not exact for all polynomials of degree $m + 1$.

The quadrature rule of the form (36) has $2n$ parameters: n nodes and n weights. Hence we can hope to make it exact for all polynomials of degree $2n - 1$ as they have $2n$ coefficients. The nodes and weights for such a rule will be a solution to the system of $2n - 1$ equations

$$\sum_{j=1}^n w_j x_j^k = \int_a^b x^k w(x) dx, \quad k = 0, 1, \dots, 2n - 1.$$

Unfortunately, this system is hard to solve. It is nonlinear and ill-conditioned. Therefore, this approach is not practical. A much better idea is to generate a set of orthogonal polynomials $\{p_k(x)\}_{k=0}^n$ on $[a, b]$ with respect to the inner product with the weight function $w(x)$ using the three-term recurrence relations (TTRR). Then the nodes coinciding with

the zeros of the polynomial $p_n(x)$ for an interpolatory quadrature rule will render it exact for all polynomials of degree up to $2n - 1$.

6.2. Orthogonal polynomials. We consider a series of polynomials orthogonal with respect to the inner product of the form

$$(38) \quad (f, g) = \int_a^b f(x)g(x)w(x)dx.$$

The interval $[a, b]$ can be finite or infinite. The weight function $w(x)$ is satisfies the following requirements.

Definition 5. We say that $w(x)$ is a weight function on $[a, b]$ if it is positive in any open interval in $[a, b]$ and

$$(39) \quad \int_a^b |x|^n w(x)dx < \infty, \quad n = 1, 2, \dots$$

For any given inner product of the form (38) one can obtain a series of orthogonal polynomials e.g. using the Gram-Schmidt orthogonalization procedure. Let $\{q_k\}_{k=0}^N$ be a basis in \mathbb{P}_N , the space of polynomials of degree less or equal to N . For example, $q_k(x) = x^k$, $k = 0, 1, \dots$

Algorithm 1: Gram-Schmidt orthogonalization

Input : a basis $\{q_k\}_{k=0}^N$ in \mathbb{P}_N .

Output: an orthogonal basis $\{p_k\}_{k=0}^N$ and an orthonormal basis $\{\tilde{p}_k\}_{k=0}^N$.

$p_0 = q_0$;

$\tilde{p}_0 = \frac{q_0}{\|q_0\|}$;

for $k = 1, \dots, N$ **do**

$p_k = q_k - \sum_{j=0}^{k-1} (q_k, \tilde{p}_j)\tilde{p}_j$;

$\tilde{p}_k = \frac{p_k}{\|p_k\|}$;

end

Remark The Gram-Schmidt orthogonalization is good for theoretical purposes but not for numerical ones. When it is used for computing orthogonal vectors, the generated vectors quickly become non-orthogonal due to the accumulation of roundoff errors. A somewhat better algorithm is the modified Gram-Schmidt procedure.

Further we will denote by $\{p_k\}_{k=0}^N$ the set of monic orthogonal polynomials (with the leading coefficient equal to one, i.e., $p_k(x) = x^k + \dots$) and by $\{\tilde{p}_k\}_{k=0}^N$ the set of orthonormal polynomials. Any polynomial of degree m , $h_m(x)$ can be written uniquely as a linear combination of the set of orthonormal polynomials $\{\tilde{p}_k\}_{k=0}^m$ in the following way

$$h_m(x) = \sum_{k=0}^m (h_m, \tilde{p}_k)\tilde{p}_k(x).$$

In addition,

$$(h_m, p_n) = 0 \text{ for any } n > m.$$

Now we establish the following fundamental result in the theory of orthogonal polynomials.

Theorem 5. (1) *Given $w(x)$, a weight function on $[a, b]$, there exists a unique family of monic polynomials $\{p_k\}$, k being the degree of polynomials, such that*

$$(40) \quad \int_a^b p_m(x)p_n(x)w(x)dx \neq 0 \text{ iff } m = n,$$

and we say that $\{p_k\}_{n \in \mathbb{N}}$ is the family of monic polynomials corresponding to the weight function $w(x)$.

- (2) *Furthermore, p_n is the only n -th degree monic polynomial which is orthogonal to all polynomials of degree smaller than n .*
 (3) *All zeros of the polynomials of this family are real and lie in (a, b) .*

Proof. (1) The existence of the family of monic orthogonal polynomials follows from the Gram-Schmidt orthogonalization procedure. Let us prove its uniqueness. Suppose that there are two families of monic orthogonal polynomials $\{p_k\}$ and $\{q_k\}$. Since the polynomials are monic, $p_0 = q_0 = 1$. Let m be the smallest integer such that $p_m \neq q_m$. Then q_m can be written as

$$q_m = \sum_{k=0}^m \frac{(q_m, p_k)}{(p_k, p_k)} p_k.$$

Since $q_k = p_k$ for $k < m$ we have

$$q_m = \sum_{k=0}^{m-1} \frac{(q_m, q_k)}{(q_k, q_k)} q_k + \frac{(p_m, q_m)}{(p_m, p_m)} p_m = \frac{(p_m, q_m)}{(p_m, p_m)} p_m$$

due to orthogonality. Therefore, q_m is proportional to p_m . Since q_m and p_m are monic, the coefficient of proportionality must be 1. Hence $p_m = q_m$ which contradicts to the hypothesis that $p_m \neq q_m$.

- (2) Any polynomial h_m of degree $m < n$ can be written as

$$h_m(x) = \sum_{k=0}^m \frac{(h_m, p_k)}{(p_k, p_k)} p_k(x).$$

Hence $(h_m, p_n) = 0$ for any $n > m$. Suppose another monic polynomial q_n of degree n is orthogonal to all smaller degree polynomials. Then q_n is orthogonal to $\{p_k\}_{k=0}^{n-1}$. Then using the argument from the proof of uniqueness we can show that q_n coincides with p_n .

- (3) Since $(p_n, p_0) = 0$, $n \geq 1$ we have

$$\int_a^b p_n(x)w(x)dx = 0.$$

Hence $p_n(x)$ changes sign in (a, b) at least once. Suppose that $p_n(x)$ changes sign in (a, b) k times at the points $x_1 < x_2 < \dots < x_k$. Consider a polynomial $q_k =$

$(x - x_1)(x - x_2) \dots (x - x_k)$. By construction, $p_n q_k > 0$ on (a, b) , hence

$$\int_a^b p_n(x) q_k(x) w(x) dx > 0.$$

On the other hand, $(p_n, q_k) = 0$ if $k < n$. Hence $k \geq n$. But an n -th degree polynomial changes sign at most n times. Hence $k = n$. □

6.3. Three Term Recurrence Relationships. The next question is how to generate orthogonal polynomials efficiently. The Gram-Schmidt orthogonalization procedure requires a lot of numerical integration and hence leads to integration errors apart from its stability issues. A better way to generate a family of orthogonal polynomials involves three-term recurrence relations (TTRRs).

Theorem 6. *The monic orthonormal polynomials $\{p_k\}$ associated with the weight function $w(x)$ on the interval $[a, b]$ satisfy the recurrence relation*

$$\begin{aligned} p_1(x) &= (x - B_0)p_0(x), \\ p_{k+1}(x) &= (x - B_k)p_k(x) - A_k p_{k-1}(x), \quad k = 1, 2, \dots, \end{aligned}$$

where

$$A_k = \frac{\|p_k\|^2}{\|p_{k-1}\|^2}, \quad k \geq 1, \quad B_k = \frac{(xp_k, p_k)}{\|p_k\|^2}, \quad k \geq 0.$$

Proof. Consider the polynomial $p_{k+1} - xp_k$. It is at most of degree k . Therefore,

$$p_{k+1} - xp_k = \sum_{i=0}^k \xi_i p_i(x).$$

Taking the inner product with p_j 's we get

$$-(xp_k, p_j) = \xi_j \|p_j\|^2.$$

If $j \leq k - 2$ we have

$$(xp_k, p_j) = (p_k, xp_j) = 0.$$

Hence only ξ_k and ξ_{k-1} can be nonzero. Therefore

$$p_{k+1} - xp_k = \xi_k p_k + \xi_{k-1} p_{k-1},$$

where

$$B_k = -\xi_k = \frac{(xp_k, p_k)}{\|p_k\|^2}.$$

and

$$A_k = -\xi_{k-1} = \frac{(xp_k, p_{k-1})}{\|p_{k-1}\|^2} = \frac{(p_k, xp_{k-1})}{\|p_{k-1}\|^2} = \frac{\|p_k\|^2}{\|p_{k-1}\|^2}.$$

The last equality comes from the fact that $x p_{k-1}$ is a monic polynomial of degree k and hence

$$x p_{k-1}(x) = p_k + \sum_{j=0}^{k-1} \eta_j p_j(x).$$

□

6.3.1. Orthonormal polynomials.

Exercise Show that the orthonormal set of polynomials $\{\tilde{p}_n\}$ also satisfy a TTRR of the form

$$(41) \quad \begin{aligned} x \tilde{p}_0 &= \alpha_1 \tilde{p}_1 + \beta_0 \tilde{p}_0, \\ x \tilde{p}_k &= \alpha_{k+1} \tilde{p}_{k+1} + \beta_k \tilde{p}_k + \alpha_k \tilde{p}_{k-1}, \quad k = 1, 2, \dots, \end{aligned}$$

where $\alpha_k = \sqrt{A_k} = \|p_k\|/\|p_{k-1}\|$ and $\beta_k = B_k$.

Hint: Observe that $p_k = \lambda_k \tilde{p}_k$, where $\lambda_k = \|p_k\|$.

The coefficients A_k and B_k can be hard to find. However, there are some special cases (called classic cases) for which the recursion coefficients can be given in an explicit analytical form. The three main families correspond to Jacobi, Hermite, and the generalized Laguerre polynomials. The standard definitions of the polynomials in these families correspond to neither monic nor orthonormal series.

6.3.2. Jacobi Polynomials. Notation: $P_n^{(\alpha, \beta)}(x)$.

Interval: $[-1, 1]$.

Weight function: $w(x) = (1-x)^\alpha(1+x)^\beta$, $\alpha, \beta > -1$.

Important particular cases:

- Chebyshev polynomials: $\alpha = \beta = -1/2$, $w(x) = (1-x^2)^{-1/2}$.
- Legendre polynomials: $\alpha = \beta = 0$, $w(x) = 1$.

TTRR:

$$\begin{aligned} a_0 P_1(x) + b_0 P_0(x) &= x P_0(x), \\ a_n P_{n+1}(x) + b_n P_n(x) + c_n P_{n-1}(x) &= x P_n(x), \end{aligned}$$

where

$$\begin{aligned} a_0 &= \frac{2}{\alpha + \beta + 2}, & b_0 &= \frac{\beta - \alpha}{\alpha + \beta + 2}, \\ a_n &= \frac{2(n+1)(n+\alpha+\beta+1)}{(L_n+1)(L_n+2)}, \\ b_n &= \frac{\beta^2 - \alpha^2}{L_n(L_n+2)}, \\ c_n &= \frac{2(n+\alpha)(n+\beta)}{L_n(L_n+1)}, & n &\geq 1, \\ L_n &= 2n + \alpha + \beta. \end{aligned}$$

6.3.3. *Generalized Laguerre polynomials.* Notation: $L_n^{(\alpha)}(x)$.

Interval: $[0, \infty)$.

Weight function: $w(x) = x^\alpha e^{-x}$, $\alpha > -1$.

TTRR:

$$\begin{aligned} -L_1^{(\alpha)}(x) + (\alpha + 1)L_0^{(\alpha)}(x) &= xL_0^{(\alpha)}(x), \\ -(n+1)L_{n+1}^{(\alpha)}(x) + (2n + \alpha + 1)L_n^{(\alpha)}(x) - (n + \alpha)L_{n-1}^{(\alpha)}(x) &= xL_n^{(\alpha)}(x). \end{aligned}$$

Remark Laguerre polynomials: $\alpha = 0$, $w(x) = e^{-x}$.

6.3.4. *Hermite polynomials.* Notation: $H_n(x)$.

Interval: $(-\infty, \infty)$.

Weight function: $w(x) = e^{-x^2}$.

TTRR:

$$\begin{aligned} \frac{1}{2}H_1(x) &= xH_0(x), \\ \frac{1}{2}H_{n+1}(x) + nH_{n-1}(x) &= xH_n(x). \end{aligned}$$

6.4. The exactness of Gaussian quadrature.

Theorem 7. Let $\{p_k(x)\}_{k=0}^n$ be a set of monic orthogonal polynomials on $[a, b]$ with respect to the inner product

$$(f, g) = \int_a^b f(x)g(x)w(x)dx.$$

Let x_j , $j = 1, 2, \dots, n$, be zeros of the polynomial $p_n(x)$. Then the quadrature rule given by

$$(42) \quad Q(f) = \sum_{j=1}^n w_j f(x_j), \quad w_j = \int_a^b \prod_{\substack{k=1 \\ k \neq j}}^n \frac{(x - x_k)}{(x_j - x_k)} w(x) dx$$

has degree of exactness $2n - 1$.

Proof. The key idea is the use of the division with residual of the polynomial $f(x)$ by the polynomial $p_n(x)$. Let $f(x)$ be a polynomial of degree $\leq 2n - 1$. Then

$$f(x) = p_n(x)q(x) + r(x), \quad \text{where } q(x), r(x) \in \mathbb{P}_{n-1},$$

i.e, $q(x)$ and $r(x)$ are polynomials of degree $\leq n - 1$. Since the rule (42) is interpolatory, it is exact for all polynomials of degree $\leq n - 1$, in particular, for $r(x)$. Then we have:

$$I(f) = I(qp_n + r) = \int_a^b q(x)p_n(x)w(x)dx + \int_a^b r(x)w(x)dx = \int_a^b r(x)w(x)dx$$

The last equality is due to the fact that the polynomial $p_n(x)$ is orthogonal to all polynomials of degree $\leq n - 1$, in particular, to $q(x)$. We continue:

$$I(f) = \int_a^b r(x)w(x)dx = I(r) = Q(r),$$

because the quadrature rule is exact for all polynomials of degree $\leq n - 1$, hence for $r(x)$. Next, since x_j 's are zeros of p_n , we write:

$$I(f) = Q(r) = \sum_{j=1}^n w_j r(x_j) = \sum_{j=1}^n w_j (p_n(x_j)q(x_j) + r(x_j)) = Q(f),$$

which shows that the quadrature rule is exact for all polynomials of degree $\leq (2n - 1)$.

It remains to prove that the rule (42) is not exact for all polynomials of degree $2n$. Let $f(x)$ be a polynomial of degree $2n$. Then $q(x)$ is of degree n and $r(x)$ is of degree $\leq n - 1$. On one hand,

$$I(f) = I(qp_n + r) = \int_a^b q(x)p_n(x)w(x)dx + \int_a^b r(x)w(x)dx = Q(r) + (q, p_n).$$

Let us show that $(q, p_n) \neq 0$. Indeed, expanding $q(x) = a_n x^n + \dots + a_0$ via the monic orthogonal polynomials p_k , we obtain:

$$q(x) = a_n p_n + \sum_{k=0}^{n-1} c_k p_k(x).$$

Hence

$$(q, p_n) = a_n \|p_n\|^2 \neq 0,$$

as $a_n \neq 0$ and $\|p_n\| \neq 0$. On the other hand,

$$Q(f) = Q(p_n q + r) = Q(r),$$

as x_j 's are the zeros of $p_n(x)$. Hence $I(f) \neq Q(f)$ for all polynomials f of degree $2n$. \square

6.5. Error estimate. Next we obtain an error estimate for the Gaussian quadrature.

Theorem 8. *Let $w(x)$ be a weight function and $p_n(x)$ be the monic polynomial of degree n orthogonal to all polynomials of smaller degrees. Let $x_j, j = 1, 2, \dots, n$, be zeros of $p_n(x)$. Let $Q(f)$ be the quadrature rule defined in Eqs. (36) and (37). Suppose $f \in C^{2n}[a, b]$. Then one can find $\lambda \in (a, b)$ such that*

$$(43) \quad \int_a^b f(x)w(x)dx = Q(f) + \gamma_n \frac{f^{(2n)}(\lambda)}{(2n)!}, \quad \text{where} \quad \gamma_n = \int_a^b p_n^2(x)w(x)dx.$$

Proof. We use the Hermite interpolation to prove the error estimate. There exists a unique polynomial $h_{2n-1}(x)$ of degree $2n - 1$ such that

$$h_{2n-1}(x_j) = f(x_j) \quad \text{and} \quad h'_{2n-1}(x_j) = f'(x_j).$$

In addition, there exists $\zeta \in (a, b)$ depending of x such that

$$f(x) = h_{2n-1}(x) + \frac{f^{(2n)}(\zeta)}{(2n)!}(x - x_1)^2 \dots (x - x_n)^2.$$

Note that $(x - x_1) \dots (x - x_n) = p_n(x)$ as p_n is monic and $x_j, j = 1, \dots, n$, are its roots. Therefore,

$$\int_a^b f(x)w(x)dx = \int_a^b h_{2n-1}(x)w(x)dx + \int_a^b \frac{f^{(2n)}(\zeta)}{(2n)!} p_n^2(x)w(x)dx.$$

Since the quadrature is exact for polynomials of degree $2n - 1$, it is exact for $h_{2n-1}(x)$. Hence

$$\int_a^b h_{2n-1}(x)w(x)dx = Q(h_{2n-1}) = \sum_{j=1}^n h_{2n-1}(x_j)w_j = \sum_{j=1}^n f(x_j)w_j = Q(f).$$

On the other hand, because $p_n^2(x)w(x) \geq 0$ on $[a, b]$, we can apply the mean value theorem and get

$$\int_a^b \frac{f^{(2n)}(\zeta)}{(2n)!} p_n^2(x)dx = \frac{f^{(2n)}(\lambda)}{(2n)!} \int_a^b p_n^2(x)w(x)dx$$

for some $\lambda \in (a, b)$. This completes the proof. \square

6.6. Example: Gauss-Chebyshev quadrature. For the Gauss-Chebyshev quadrature, $[a, b] = [-1, 1]$, the weight function is

$$w(x) := \frac{1}{\sqrt{1-x^2}},$$

and the nodes are zeros of the Chebyshev polynomial $T_n(x)$:

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

and the weights w_n are all equal to π/n . Below we show how one can find the weights w_n .

As we know, the $n - 1$ -st degree Chebyshev interpolant of $f(x)$ is

$$f(x) \approx f_{n-1} = \frac{c_0}{2} + \sum_{i=1}^{n-1} c_i T_i(x),$$

where

$$c_i = \frac{2}{n} \sum_{k=1}^n f(x_k) T_i(x_k), \quad x_k = \cos\left(\frac{\pi(k - \frac{1}{2})}{n}\right), \quad i = 1, \dots, n.$$

The Gauss-Chebyshev rule consists in the approximation

$$I(f) = \int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx \approx Q(f) = \int_{-1}^1 \frac{f_{n-1}(x)}{\sqrt{1-x^2}} dx \equiv (f_{n-1}, T_0),$$

as $T_0(x) \equiv 1$. Recall that

$$(T_0, T_0) = \int_{-1}^1 (1-x^2)^{-1/2} dx = \int_{-1}^1 d(\arcsin(x)) = \arcsin(1) - \arcsin(-1) = \pi.$$

Plugging in the expression for $f_{n-1}(x)$ and using the facts that $(T_i, T_k) = 0$ for $i \neq k$, we get

$$Q(f) = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} \left(\frac{c_0}{2} + \sum_{i=1}^{n-1} c_i T_i(x) \right) dx = \frac{c_0}{2} (T_0, T_0) + \sum_{i=1}^{n-1} c_i (T_0, T_i) = \frac{\pi c_0}{2}.$$

Now, using the expression for c_0 we obtain

$$Q(f) = \frac{\pi c_0}{2} = \frac{\pi}{2} \frac{2}{n} \sum_{k=1}^n f(x_k) = \frac{\pi}{n} \sum_{k=1}^n f(x_k).$$

Therefore, the weights for the Gauss-Chebyshev quadrature are all equal to π/n :

$$w_i = \frac{\pi}{n}, \quad i = 1, \dots, n.$$

6.7. A naive way for finding nodes and weights. If the number of nodes n is small, one can find the nodes as the roots of p_n and calculate the weights directly by setting the rule to be exact for $1, \dots, x^{n-1}$.

Problem Consider quadrature rules for evaluating integrals on the interval $[0, +\infty)$ of the form

$$I(f) := \int_0^\infty f(x) e^{-x} dx \approx Q(f) := \sum_{j=1}^n w_j f(x_j).$$

Suppose you wish to find such a rule that is exact for all cubic polynomials.

- (1) Show that it suffices to devise a two-point rule: $Q(f) = w_1 f(x_1) + w_2 f(x_2)$.
- (2) Find w_j and x_j , $j = 1, 2$, for this rule.

Hint: the Laguerre polynomials orthogonal with respect to the inner product

$$(f, g) = \int_0^{\infty} f(x)g(x)e^{-x}dx$$

can be found from the recurrence relationship

$$L_0(x) = 1, \quad L_1(x) = 1 - x, \quad L_{k+1} = \frac{1}{k+1}[(2k+1-x)L_k(x) - kL_{k-1}(x)].$$

Solution (1) Set x_1 and x_2 to be the roots of $L_2(x)$. Let $p(x)$ be a polynomial of at most 3-rd degree. Then $p(x) = L_2(x)q(x) + r(x)$ where the degrees of the polynomials $q(x)$ and $r(x)$ do not exceed 1. Then we have

$$\begin{aligned} I(p) &= I(L_2(x)q(x) + r(x)) = I(L_2(x)q(x)) + I(r(x)) \\ &= I(r) = Q(r) = Q(L_2(x)q(x) + r(x)) = Q(p). \end{aligned}$$

We note that $I(L_2(x)q(x)) = 0$ due to orthogonality of $L_2(x)$ to all of the polynomials of smaller degrees, $I(r) = Q(r)$ due to exactness of the quadrature for all polynomials of degree less than 2 by construction, and $Q(L_2(x)q(x)) = 0$ by construction.

(2) Nodes $x_1 = 2 - \sqrt{2}$ and $x_2 = 2 + \sqrt{2}$ are the roots of the polynomial

$$L_2(x) = \frac{1}{2}(x^2 - 4x + 2).$$

To find the weights, we set the following system of equations:

$$\begin{aligned} I(1) &= \int_0^{\infty} e^{-x}dx = 1 = Q(1) = 1 \cdot w_1 + 1 \cdot w_2, \\ I(x) &= \int_0^{\infty} xe^{-x}dx = 1 = Q(x) = x_1w_1 + x_2w_2. \end{aligned}$$

Solving it we get

$$w_1 = \frac{1 + \sqrt{2}}{2\sqrt{2}}, \quad w_2 = \frac{\sqrt{2} - 1}{2\sqrt{2}}.$$

The resulting quadrature rule is

$$Q(f) = \frac{1 + \sqrt{2}}{2\sqrt{2}} f(2 - \sqrt{2}) + \frac{\sqrt{2} - 1}{2\sqrt{2}} f(2 + \sqrt{2}).$$

6.8. Golub-Welsch algorithm for finding nodes and weights. When n is not so small, there is a better way to compute the nodes and weights using the TTRRs. It is known as the Golub-Welsch algorithm.

6.8.1. *Orthonormal polynomials.* The TTRRs for the three families of orthogonal polynomials in Sections 6.3.2–6.3.4 are traditionally given in the form

$$(44) \quad xp_k(x) = a_k p_{k+1}(x) + b_k p_k(x) + c_k p_{k-1}(x), \quad c_0 p_{-1}(x) = 0.$$

The norms of the polynomials obtained from these relationships are different from 1. We need to convert TTRR (44) to the TTRR for the orthonormal family. Since the orthonormal polynomials $\tilde{p}_k(x)$ and the orthogonal polynomials $p_k(x)$ relate via

$$p_j(x) = \lambda_j \tilde{p}_j(x),$$

we have:

$$(45) \quad x\lambda_k \tilde{p}_k(x) = a_k \lambda_{k+1} \tilde{p}_{k+1}(x) + b_k \lambda_k \tilde{p}_k(x) + c_k \lambda_{k-1} \tilde{p}_{k-1}(x).$$

Hence

$$\alpha_{k+1} = a_k \frac{\lambda_{k+1}}{\lambda_k}, \quad \beta_k = b_k, \quad \alpha_k = c_k \frac{\lambda_{k-1}}{\lambda_k}.$$

Therefore,

$$\alpha_k = a_{k-1} \frac{\lambda_k}{\lambda_{k-1}} = c_k \frac{\lambda_{k-1}}{\lambda_k} = \sqrt{a_{k-1} c_k},$$

In summary, we obtain α_k 's and β_k 's for orthonormal families from a_k 's, b_k 's, and c_k 's for the non-orthonormal ones according to:

$$(46) \quad \alpha_k = \sqrt{a_{k-1} c_k}, \quad \beta_k = b_k.$$

6.8.2. *Derivation of the Golub-Welsh algorithm.* The starting point for computing nodes and weights of a Gaussian n -point rule is the TTRR for the orthonormal polynomials (see Section 6.8.1 above). Let x_j be the one of the nodes, i.e., roots of $\tilde{p}_n(x)$. Then

$$(47) \quad \begin{aligned} \alpha_1 \tilde{p}_1(x_j) + \beta_0 \tilde{p}_0(x_j) &= x_j \tilde{p}_0(x_j) \\ \alpha_2 \tilde{p}_2(x_j) + \beta_1 \tilde{p}_1(x_j) + \alpha_1 \tilde{p}_0(x_j) &= x_j \tilde{p}_1(x_j) \\ \vdots & \\ \beta_{n-1} \tilde{p}_{n-1}(x_j) + \alpha_{n-1} \tilde{p}_{n-2}(x_j) &= x_j \tilde{p}_{n-1}(x_j). \end{aligned}$$

In the matrix form Eq. (47) is

$$(48) \quad \begin{bmatrix} \beta_0 & \alpha_1 & 0 & \dots & 0 \\ \alpha_1 & \beta_1 & \alpha_2 & & \\ 0 & \alpha_2 & \beta_2 & & \\ \vdots & & & & \\ 0 & \dots & 0 & \alpha_{n-1} & \beta_{n-1} \end{bmatrix} \begin{bmatrix} \tilde{p}_0(x_j) \\ \tilde{p}_1(x_j) \\ \tilde{p}_2(x_j) \\ \vdots \\ \tilde{p}_{n-1}(x_j) \end{bmatrix} = x_j \begin{bmatrix} \tilde{p}_0(x_j) \\ \tilde{p}_1(x_j) \\ \tilde{p}_2(x_j) \\ \vdots \\ \tilde{p}_{n-1}(x_j) \end{bmatrix},$$

or

$$(49) \quad JP(x_j) = x_j P(x_j),$$

where $P(x_j) = [\tilde{p}_0(x_j), \dots, \tilde{p}_{n-1}(x_j)]^\top$. Therefore, *the nodes that we need for the n -point Gaussian quadrature are the eigenvalues of the matrix J !*

Now we need to find a way to obtain the weights. We will use the facts that

- the Gaussian quadrature is exact for $\tilde{p}_0, \tilde{p}_1, \dots, \tilde{p}_{n-1}$,
- $P(x_j)$ is the eigenvector with the eigenvalue x_j , and
- \tilde{p}_0 is constant and we can easily find it and set the proper scaling for the rest of the entries of P .

Since

$$(\tilde{p}_i, \tilde{p}_k) = \int_a^b \tilde{p}_i(x)\tilde{p}_k(x)w(x)dx = \delta_{ik}, \quad 0 \leq i, k \leq n-1,$$

and the Gaussian quadrature is exact on these polynomials, we have

$$\delta_{ik} = (\tilde{p}_i, \tilde{p}_k) = \sum_{j=1}^n \tilde{p}_i(x_j)\tilde{p}_k(x_j)w_j.$$

In the matrix form this expression is

$$\begin{aligned} & \begin{bmatrix} \tilde{p}_0(x_1) & \dots & \tilde{p}_0(x_n) \\ \vdots & & \vdots \\ \tilde{p}_{n-1}(x_1) & \dots & \tilde{p}_{n-1}(x_n) \end{bmatrix} \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix} \begin{bmatrix} \tilde{p}_0(x_1) & \dots & \tilde{p}_{n-1}(x_1) \\ \vdots & & \vdots \\ \tilde{p}_0(x_n) & \dots & \tilde{p}_{n-1}(x_n) \end{bmatrix} \\ &= \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}, \end{aligned}$$

or, more compactly,

$$(50) \quad PWP^\top = I,$$

where $W = \text{diag}(w_1, w_2, \dots, w_n)$ and $P = [P(x_1), P(x_2), \dots, P(x_n)]$, i.e., P is the matrix whose columns are the properly scaled eigenvectors of J .

It follows from Eq. (50) that P is invertible, hence

$$W = P^{-1}P^{-\top} = (P^\top P)^{-1}.$$

Thus,

$$W^{-1} = P^\top P,$$

which means that

$$(51) \quad \frac{1}{w_j} = \sum_{k=0}^{n-1} (\tilde{p}_k(x_j))^2 = \|P(x_j)\|^2,$$

where $\|\cdot\|$ is the Euclidean norm.

On the other hand, given an eigenvector $v^{(j)} = [v_1^{(j)}, \dots, v_n^{(j)}]^\top$ of the matrix J , there exists a constant C such that

$$v^{(j)} = CP(x_j) = C[\tilde{p}_0(x_j), \tilde{p}_1(x_j), \dots, \tilde{p}_{n-1}(x_j)]^\top.$$

The value of C can be obtained by considering

$$1 = (\tilde{p}_0, \tilde{p}_0) = \tilde{p}_0^2 \int_a^b w(x)dx = \tilde{p}_0^2 \mu_0.$$

It follows that

$$\tilde{p}_0 = \frac{1}{\sqrt{\mu_0}}.$$

Hence

$$v_1^{(j)} = C\tilde{p}_0 = \frac{C}{\sqrt{\mu_0}}.$$

Then

$$C = v_1^{(j)}\sqrt{\mu_0}.$$

Therefore,

$$P(x_j) = \frac{1}{C}v^{(j)} = \frac{1}{v_1^{(j)}\sqrt{\mu_0}}v^{(j)}.$$

Finally, we obtain the weight w_j associated with the node x_j :

$$(52) \quad w_j = \frac{1}{\|P(x_j)\|^2} = \mu_0 \frac{(v_1^{(j)})^2}{\|v^{(j)}\|^2}.$$

6.8.3. *The Golub-Welsch algorithm. Input:* $\mu_0 = \int_a^b w(x)dx$

Output: $x_1, \dots, x_n; w_1, \dots, w_n$.

- Build J from $\alpha_1, \dots, \alpha_{n-1}, \beta_0, \dots, \beta_{n-1}$.
- Compute eigenvalues ρ_1, \dots, ρ_n and eigenvectors v_1, \dots, v_n of J .
- for $i = 1 : n$

$$\begin{aligned} x_i &= \rho_i, \\ w_i &= \mu_0 \frac{v_i(1)^2}{\|v_i\|^2}. \end{aligned}$$

end

The following subroutine implements the Golub-Welsch algorithm in matlab for computing the Gamma function. It tests that for $d \in \mathbb{N}$,

$$\Gamma(d+1) = \int_0^\infty x^d e^{-x} dx = d!$$

```
function golubwelsch()
global deg
deg = 7;
n=ceil((deg + 1)/2); % the number of nodes
b=zeros(n,1);
a=zeros(n,1);
b(1)=1;
for j=2:n
    k=j-1;
    b(j)=b(k)+2;
    a(j)=k;
end
mu0=1;
%% form J
```

28

```
J=zeros(n);
J(1,1)=b(1);
J(1,2)=a(2);
for j=2:n-1
    J(j,j)=b(j);
    J(j,j-1)=a(j);
    J(j,j+1)=a(j+1);
end
J(n,n)=b(n);
J(n,n-1)=a(n);

%% find eigenvalues and eigenvectors
[V E]=eig(J);

%% find weights
x=zeros(n,1);
w=zeros(n,1);
for j=1:n
    x(j)=E(j,j);
    w(j)=mu0*V(1,j)^2/norm(V(:,j))^2;
end

%% compute integral

I=sum(myf(x).*w);
Iexact = factorial(deg);
fprintf('n = %d, I = %d, Iexact = %d, I - Iexact = %d\n',...
    n, I,Iexact,I-Iexact);
end
%%
function y=myf(x)
global deg
p=zeros(deg + 1,1);
p(1)=1;
y=polyval(p,x);
end
```