Lecture 14: Quadrature

This lecture is concerned with the evaluation of integrals

$$\int_a^b f(x)dx \tag{1}$$

over a finite interval $[a, b]$. The integrand $f(x)$ is assumed to be real-values and smooth. The approximation of an integral by a numerical method is commonly referred to as *quadrature*.

In Calculus one discusses many techniques for determining an antiderivative of $f(x)$. The integral is then computed by evaluating the antiderivative at the endpoints of the interval. However, many integrands of interest in science and engineering do not have a known antiderivative. Moreover, in some applications the integrand is only known at a few points in the interval. We would like to be able to determine accurate approximations of integrals also in these situations.

We will approximate integrals (1) by sums

$$\sum_{j=1}^{N} f(x_j)w_j. \tag{2}$$

These sums are referred to as *quadrature rules*. The x_j are the *nodes* and the w_j the *weights* of the quadrature rule. We are interested in determining weights so that the quadrature rules gives accurate approximations of the integral (1 for large classes of integrands $f(x)$. The nodes often cannot be chosen freely, because the integrand may only be known at certain points. For instance, the function $f(x)$ might not be explicitly known, only measured values $f(x_j)$ at the nodes x_j may be available.

We remark that you already encountered the approximation of integrals by sums in Calculus. There, however, one was less concerned with how small the error

$$E_N(f) = \int_a^b f(x)dx - \sum_{j=1}^{N} f(x_j)w_j \tag{3}$$

is for small values of N; it was sufficient that the error $E_N(f)$ converged to zero as N increased to infinity. In this lecture, we would like the error $E_N(f)$ to be small already for a small to modest number of terms N in the quadrature rule (2).

Method of undetermined coefficients

Let the nodes x_j, $1 \leq j \leq N$, be given. Throughout this lecture the nodes will be ordered so that

$$a \leq x_1 < x_2 < \ldots < x_N \leq b. \tag{4}$$

We would like to determine the weights w_j so that quadrature rule is exact for polynomials of as high degree as possible, i.e., we would like the weights be such that

$$\int_a^b p(x)dx = \sum_{j=1}^{N} p(x_j)w_j \tag{5}$$

for all polynomials $p(x)$ of as high degree as possible.

The powers $\{1, x, x^2, x^3, \dots\}$ form a basis for polynomials. Our objective therefore can be expressed as follows: We would like the equality (5) to hold for as many powers $p(x) = x^j$, $j = 0, 1, 2, \dots$, as possible. This requirement gives rise to the linear system of equations for the weights,

$$\begin{array}{lccccccccccc} p(x)=1: & w_1 & + & w_2 & +\ldots+ & w_N & = & \int_a^b dx & = & b-a, \\ p(x)=x: & x_1w_1 & + & x_2w_2 & +\ldots+ & x_Nw_N & = & \int_a^b xdx & = & \frac{1}{2}(b^2-a^2), \\ p(x)=x^2: & x_1^2w_1 & + & x_2^2w_2 & +\ldots+ & x_N^2w_N & = & \int_a^b x^2dx & = & \frac{1}{3}(b^3-a^3), \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ p(x)=x^{N-1}: & x_1^{N-1}w_1 & + & x_2^{N-1}w_2 & +\ldots+ & w_Nx_N^{N-1} & = & \int_a^b x^{N-1}dx & = & \frac{1}{N}(b^N-a^N). \end{array} \tag{6}$$

This system conveniently can be expressed in the form

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ \vdots & \vdots & & \vdots \\ x_1^{N-1} & x_2^{N-1} & \dots & x_N^{N-1} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} b-a \\ \frac{1}{2}(b^2-a^2) \\ \vdots \\ \frac{1}{N}(b^N-a^N) \end{bmatrix}. \tag{7}$$

We recognize the matrix as the transpose of a Vandermonde matrix. Vandermonde matrices were encountered in the fall semester in connection with polynomial interpolation, when we showed that Vandermonde matrices determined by distinct nodes are nonsingular. Determinants are invariant under transposition. This follows from the fact that determinants can be expanded either by rows or by columns. We conclude that the matrix in (7) is nonsingular when the nodes x_j are distinct. Hence the above linear system of equations has a unique solution. Thus, *given n distinct nodes, we can determine n unique weights w_j, such that the quadrature rule (2) integrates all polynomials of degree strictly less than n exactly.*

Example 1. Consider the midpoint quadrature rule

$$\int_a^b f(x)dx \approx f(x_1)w_1, \qquad x_1 = \frac{1}{2}(a+b).$$

We only have one weight to determine. The linear system of equations (6) reduces to

$$p(x) = 1: \quad w_1 = \int_a^b dx = b - a.$$

This determines the weight w_1. $\Box$

Example 2. Apply the midpoint rule to integrate the function

$$f(x) = \exp(\sqrt{1-(x/2)^2}) \tag{8}$$

on the interval $[0, 1]$. The midpoint rule approximates the integrand by the constant function $f(1/2) = \exp(\sqrt{7/8})$ on the interval $[a, b]$ and integrates the latter function exactly. The blue graph of Figure 1 shows $f(x)$ and the value of the integral is the area below this graph. The dashed red line displayes the constant function that is integrated exactly by the midpoint rule. The value determined by the midpoint rule is the area below the dashed red line. $\Box$

Sometimes N-node quadrature rules determined by solving the linear system of equations (7) also integrate powers x^k for $k \geq N$ exactly. The following example illustrates this.

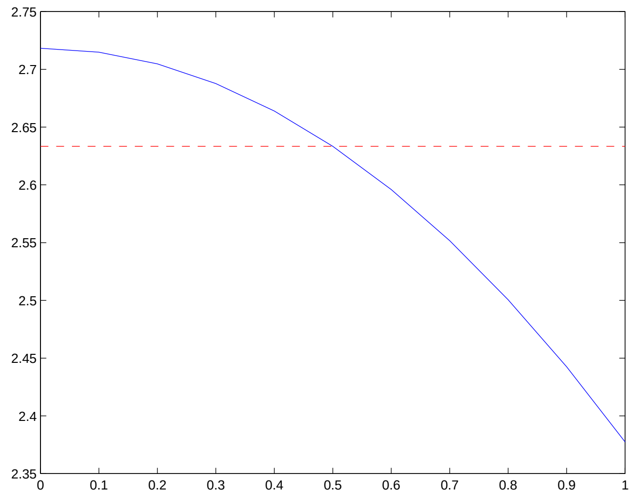

Figure 1: Integrand of Example 2 and approximation used by the midpoint rule.

Example 3. Consider the midpoint quadrature rule of Example 1. Application of this rule to $f(x) = x$ yields

$$x_1 w_1 = \frac{1}{2}(a+b)(b-a) = \frac{1}{2}(b^2 - a^2).$$

The right-hand side equals $\int_a^b x dx$, i.e., the midpoint rule integrates not only constants but also linear polynomials exactly. □

Exercise 1. Determine the weights of the 2-node quadrature rule

$$\int_a^b f(x)dx \approx f(x_1)w_1 + f(x_2)w_2, \qquad x_1 = a, \quad x_2 = b.$$

This rule is known as the trapezoidal rule. By construction, it integrates linear polynomials exactly. Does it integrate polynomials of higher degree exactly? Justify your answer. □

Exercise 2. Determine the weights of the 3-node quadrature rule

$$\int_0^1 f(x)dx \approx f(x_1)w_1 + f(x_2)w_2 + f(x_3)w_3, \qquad x_1 = 0,\ x_2 = 1/2,\ x_3 = 1.$$

This rule is known as Simpson's rule. It integrates quadratic polynomials exactly by construction. Does it integrate polynomials of higher degree exactly? Justify your answer. What is the analogous quadrature rule for the interval $[1, 2]$? Hint: Do not recompute the quadrature rule, just make a change of variables. □

Integrating Lagrange polynomials

When discussing polynomial interpolation last fall, we considered different polynomial bases, such as monomials, Lagrange polynomials, and Newton polynomials. We can also in the present context use bases different from the monomial one. For instance, the nodes $x_1, x_2, \ldots, x_N$ determine the Lagrange polynomials

$$\ell_k(x) = \prod_{\substack{j=1 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}, \qquad k = 1, 2, \ldots, N, \tag{9}$$

which form a basis for all polynomials of degree at most $N-1$. We therefore may require that an N-node quadrature rule integrates the Lagrange polynomials (9) exactly. Substituting $p(x) = \ell_k(x)$ into (5) yields

$$\int_a^b \ell_k(x)dx = \sum_{j=1}^{N} \ell_k(x_j)w_j = \ell_k(x_k)w_k = w_k, \qquad k = 1, 2, \ldots, N,$$

where the simplifications of the right-hand side follow from the fact that

$$\ell_k(x_j) = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$$

Thus, the weights can be obtained by integrating the Lagrange polynomials. However, the evaluation of these integrals is tedious unless N is small.

Composite quadrature rules and Taylor expansion

Assume that the midpoint rule of Example 2 does not yield a sufficiently accurate approximation of the integral. We may then subdivide the interval $[0, 1]$ into subintervals and apply the midpoint rule on each subinterval. This defines the composite midpoint rule.

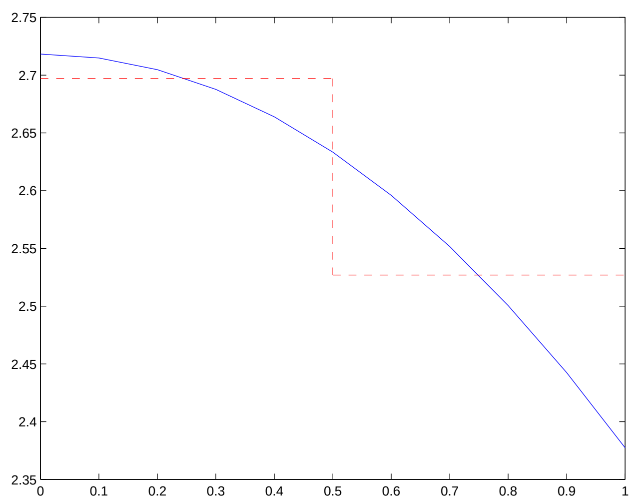

Figure 2: Integrand of Example 4 and approximation used by the composite midpoint rule obtained by dividing the interval $[0, 1]$ into two subintervals of equal length.

Example 4. Divide the interval $[0, 1]$ into the subintervals $[0, 1/2]$ and $[1/2, 1]$ and apply the midpoint rule on each subinterval to the integrand $f(x)$ defined by (8). This yields the composite midpoint rule

$$\int_0^1 f(x)dx \approx f(x_1)w_1 + f(x_2)w_2$$

with $x_1 = 1/4$, $x_2 = 3/4$, and $w_1 = w_2 = 1/2$. The integral is approximated by the area below the dashed curve of Figure 2. Comparing the graphs of Figures 1 and 2 suggests that subdivision of the interval should increase the accuracy of the computed approximation. $\square$

While the computation of the weights by solving the linear system of equations (7) is easily done in MATLAB or Octave, this approach does not shed any light on how the error behaves when we increase the number of nodes. Further insight on the behavior of quadrature rules can be gained by expanding the integrand into a Taylor series.

Consider the approximation of the integral $\int_0^h f(x)dx$ by the midpoint rule and use the Taylor expansion of $f(x)$ at $x = h/2$. Here $x - h/2$ should be thought of as fairly small, and we assume that $f(x)$ is continuously differentiable as many times as required. Then

$$f(x) = f\left(\frac{h}{2}\right) + f'\left(\frac{h}{2}\right)\left(x - \frac{h}{2}\right) + \frac{f''\left(\frac{h}{2}\right)}{2!}\left(x - \frac{h}{2}\right)^2 + \frac{f'''\left(\frac{h}{2}\right)}{3!}\left(x - \frac{h}{2}\right)^3 + \frac{f''''\left(\frac{h}{2}\right)}{4!}\left(x - \frac{h}{2}\right)^4 + \dots$$

Integrating the left-hand and right-hand sides from 0 to h yields

$$\begin{aligned}\int_0^h f(x)dx &= f\left(\frac{h}{2}\right)\int_0^h dx + f'\left(\frac{h}{2}\right)\int_0^h \left(x - \frac{h}{2}\right)dx + \frac{f''\left(\frac{h}{2}\right)}{2!}\int_0^h \left(x - \frac{h}{2}\right)^2 dx \\ &\quad + \frac{f'''\left(\frac{h}{2}\right)}{3!}\int_0^h \left(x - \frac{h}{2}\right)^3 dx + \frac{f''''\left(\frac{h}{2}\right)}{4!}\int_0^h \left(x - \frac{h}{2}\right)^4 dx + \dots\end{aligned}$$

The integral over odd powers of $x - h/2$ vanishes due to symmetry. The right-hand side therefore simplifies to

$$\begin{aligned}\int_0^h f(x)dx &= f\left(\frac{h}{2}\right)h + \frac{f''\left(\frac{h}{2}\right)}{2!}\int_0^h \left(x - \frac{h}{2}\right)^2 dx + \frac{f''''\left(\frac{h}{2}\right)}{4!}\int_0^h \left(x - \frac{h}{2}\right)^4 dx + \dots \\ &= f\left(\frac{h}{2}\right)h + \frac{f''\left(\frac{h}{2}\right)}{12}h^3 + \frac{f''''\left(\frac{h}{2}\right)}{80}h^5 + \dots\end{aligned}$$

The midpoint rule applied to the integral on the left-hand side gives the first term in the right-hand side. The remaining terms express the quadrature error. Hence, this error is given by

$$\int_0^h f(x)dx - f\left(\frac{h}{2}\right)h = \frac{f''\left(\frac{h}{2}\right)}{12}h^3 + \frac{f''''\left(\frac{h}{2}\right)}{80}h^5 + \dots \tag{10}$$

Since all derivatives of order 2 and higher of a linear function vanish, the right-hand side vanishes for such functions. It follows that the midpoint rule is exact for linear functions. We know this already, and here it is a consequence of the expansion of the integral in powers of h.

Exercise 3. Use Taylor expansions around $x = 0$ and $x = h$ to determine how the approximation of the integral $\int_0^h f(x)dx$ determined by the trapezoidal rule depends on h. Thus, express the quadrature error

$$\int_0^h f(x)dx - \frac{h}{2}(f(0) + f(h))$$

as a function of h in a similarly manner as (10). $\square$

Consider the approximation of the integral (1) by the N-point composite midpoint rule with the nodes and weights

$$x_j = a + \left(j - \frac{1}{2}\right) h, \quad h = \frac{b-a}{N - \frac{1}{2}}, \qquad w_j = h, \qquad j = 1, 2, \dots, N. \tag{11}$$

Thus, the quadrature rule is given by

$$M_h(f) = h \sum_{j=1}^{N} f(x_j).$$

Analogously to (10), we obtain for each subinterval $[x_j - h/2, x_j + h/2]$ of length h the expression

$$\int_{x_j - h/2}^{x_j + h/2} f(x)dx - f(x_j)h = \frac{f''(x_j)}{12} h^3 + \dots \ , \tag{12}$$

and summing over $j = 1, 2, \dots, N$ yields

$$\int_a^b f(x) - M_h(f) = \sum_{j=1}^{N} \frac{f''(x_j)}{12} h^3 + \dots \ = \frac{1}{N} \sum_{j=1}^{N} f''(x_j) \frac{Nh^3}{12} + \dots \ . \tag{13}$$

The average of the second derivative values $\frac{1}{N} \sum_{j=1}^{N} f''(x_j)$ is not smaller than the smallest of the values $f''(x_j)$. Similarly, the average is not larger than largest of the values $f''(x_j)$. This is expressed by the inequalities

$$\min_{1 \le j \le N} f''(x_j) \le \frac{1}{N} \sum_{j=1}^{N} f''(x_j) \le \max_{1 \le j \le N} f''(x_j).$$

We assumed $f''(x)$ to be continuous. Therefore there is a value ξ in $[a, b]$, such that

$$f''(\xi) = \frac{1}{N} \sum_{j=1}^{N} f''(x_j).$$

Substituting this expression into the right-hand side of (13) yields

$$\int_a^b f(x) - M_h(f) = f''(\xi) \frac{Nh^3}{12} + \dots \ . \tag{14}$$

Multiplying the expression for h in (11) by the denominator gives

$$Nh = b - a + \frac{h}{2},$$

which when substituted into (14) yields

$$\int_a^b f(x) - M_h(f) = f''(\xi) \frac{b-a}{12} h^2 + \dots \tag{15}$$

The right-hand side indicates that we can expect the error to decrease by a factor 4 when h is halved (and the number of nodes x_j is doubled).

Generally, one does not know in advance how many nodes to use in order to achieve desired accuracy. We therefore may be interested in halving h until consecutive approximations $M_h(f), M_{h/2}(f), M_{h/4}(f), \ldots$, of the integral do not vary much when h is further reduced.

Exercise 4. In applications with complicated functions, the evaluation of the function values may dominate the arithmetic work required to evaluate quadrature rules. The computation of $M_h(f)$ requires the evaluation of N function values. How many of these function values can be used again when evaluating $M_{h/2}(f)$? □

Exercise 5. Consider the N-point composite trapezoidal rule for the approximation of (1),

$$T_h(f) = h\left(\frac{1}{2}f(x_1) + f(x_2) + f(x_3) + \ldots + f(x_{N-1}) + \frac{1}{2}f(x_N)\right)$$

with

$$x_j = a + (j-1)h, \quad h = \frac{b-a}{N-1}, \qquad j = 1, 2, \ldots, N.$$

a) Use the representation

$$T_h(f) = \sum_{j=1}^{N-1} \frac{h}{2}(f(x_j) + f(x_{j+1}))$$

and the result of Exercise 3 to determine an expansion of the quadrature error similar to (13).

b) What is the analog of formula (15)? How much is the error reduced when h is halved?

c) Assume that $T_h(f)$ has been evaluated, and we would like to compute $T_{h/2}(f)$. How many additional function evaluations are required? □

Singular integrands, infinite intervals, and adaptive quadrature rules

The composite midpoint and trapezoidal rules can be used also for functions that are not twice continuously differentiable. However, the error then may converge to zero slower when h is reduced and many nodes might be required to give a small quadrature error. We discuss an approach to remedy this situation.

Consider the evaluation of the integral

$$\int_0^1 f(x)\ln(x)dx \tag{16}$$

where $f(x)$ is assumed to be a smooth function. We note that the trapezoidal rule cannot be used since the integrand is infinite at $x = 0$. Nevertheless, the integral exists for many smooth functions $f(x)$. We therefore may consider using a composite midpoint rule; see Exercise 6.

Alternatively, the method of undetermined coefficients can be applied to the function $f(x)$ only. Thus, we would like to determine a quadrature rule of the form

$$\sum_{j=1}^{N} f(x_j)w_j$$

with given nodes, and we seek to determine weights w_j so that the relation

$$\int_a^b p(x)\ln(x)dx = \sum_{j=1}^{N} p(x_j)w_j \tag{17}$$

holds for all polynomials of as high degree as possible. This yields the linear system of equations, which is analogous to (6),

$$
\begin{array}{lccccccccc}
p(x) = 1: & w_1 & + & w_2 & +\ldots+ & w_N & = & \int_a^b \ln(x)dx, \\
p(x) = x: & x_1 w_1 & + & x_2 w_2 & +\ldots+ & x_N w_N & = & \int_a^b x\ln(x)dx, \\
p(x) = x^2: & x_1^2 w_1 & + & x_2^2 w_2 & +\ldots+ & x_N^2 w_N & = & \int_a^b x^2\ln(x)dx, \\
\vdots & \vdots & & \vdots & & \vdots & & \vdots \\
p(x) = x^{N-1}: & x_1^{N-1} w_1 & + & x_2^{N-1} w_2 & +\ldots+ & w_N x_N^{N-1} & = & \int_a^b x^{N-1}\ln(x)dx.
\end{array}
\tag{18}
$$

The right-hand side expressions can be evaluated by integration by parts. Note that the allocation of nodes x_j is quite arbitrary; in particular, we may put a node at the origin.

Exercise 6. Compute an approximation of (16) with $f(x) = \exp(x^2)$ by the 2-node composite midpoint rule. Use the MATLAB function `quad` to determine the "exact" value. Note that the MATLAB function for the integrand has to written to allow vector arguments, e.g.,

```
f=exp(x.^2).*log(x);
```

How large is the error? The MATLAB function `quad` applies a composite Simpson rule. □

The MATLAB function `quad` is an example of an adaptive quadrature rule. Adaptive rules are composite rules that estimate the quadrature error by comparing the result obtained with different mesh sizes h. New nodes are allocated in subintervals $[x_j, x_{j+1}]$ in which the quadrature error is deemed to be larger than a prescribed tolerance.

Exercise 7. Determine a 2-point quadrature rule with the same nodes as in Exercise 6 by solving a linear system of equations of the form (18). Apply it to the function $f(x)$ of Exercise 6. Is this rule more accurate than the one in Example 6? □

Exercise 8. Evaluate the integral $\int_0^\infty \exp(-x^3)dx$ with at least 5 correct decimal digits using the MATLAB function `quad`. This function requires the interval of integration to be finite. Therefore the integral has to be split,

$$\int_0^\infty \exp(-x^3)dx = \int_0^c \exp(-x^3)dx + \int_c^\infty \exp(-x^3)dx,$$

where the constant $c > 0$ is chosen large enough so that the second integral can be neglected and the first integral is evaluated with the function `quad`. For instance, we may choose c large enough so that

$$\int_c^\infty \exp(-x^3)dx \leq 5 \cdot 10^{-6}.$$

Determine such a value of c and justify your choice. The error bound for the integral evaluated with `quad` then should not be larger than $5 \cdot 10^{-6}$. □

In this lecture, we have fixed the nodes and then determined the weights so that we integrate polynomials of as high degree as possible. A clever choice of nodes as well as weights in an N-point quadrature rule makes it possible to integrate polynomials of degree up to $2N - 1$ exactly. The midpoint rule is an example. These rules are known as Gaussian quadrature rules. The nodes and weights generally are not known in closed form, but they can be computed quite efficiently by solving an eigenvalue problem a symmetric $N \times N$ tridiagonal matrix. This is often done with the QR-algorithm or a modification thereof.