KTH Engineering Sciences

Numerical Integration

In these notes we discuss numerical methods for approximating the integral

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

The most common methods are *interpolatory*, i.e. they are derived from the strategy

1. Interpolate $f(x)$ by a polynomial $p(x)$ in some nodes,
2. Approximate $I(f)$ by the exact integral of $p(x)$,

$$I(f) \approx \int_a^b p(x)dx. \tag{1}$$

Suppose there are $n+1$ nodes $a \le x_0 < x_1 < \cdots < x_n \le b$ and that $p(x)$ is the unique interpolation polynomial[1] in $\mathbb{P}_n$, the space of polynomials of degree at most n. Then we can use the Lagrange form for the interpolation polynomial to get

$$I(f) \approx \int_a^b p(x)dx = \int_a^b \sum_{j=0}^n f(x_j)L_j(x)dx = \sum_{j=0}^n f(x_j)\left(\int_a^b L_j(x)dx\right).$$

Here $L_j(x) \in \mathbb{P}_n$ is the j-th Lagrange polynomial for the nodes (see Sauer 3.1.1). Setting

$$w_j = \int_a^b L_j(x)dx, \tag{2}$$

we see that interpolatory methods using $n+1$ nodes always lead to an explicit $n+1$-point *quadrature formula*,

$$I(f) \approx \sum_{j=0}^n w_j f(x_j) =: Q_{n+1}(f),$$

with some nodes, $\{x_j\}$, and *weights*, $\{w_j\}$, neither of which depend on f.

1 Newton–Cotes Formulae

The most simple interpolatory quadrature formulae are the Newton–Cotes formulae. They are based on *equidistant* polynomial interpolation. An n-point Newton–Cotes rule hence uses an $n-1$ degree polynomial in (1). The first few rules are well known under their nick names:

[1] The existence of $p(x)$ is guaranteed by Theorem 3.2 in Sauer, page 141.

0. The *midpoint* rule uses a zeroth order polynomial interpolating f in one point, $x_0 = (b+a)/2$. Then $p_0(x) \equiv f(x_1)$ and, denoting the rule $M(f)$, we get

$$M(f) := \int_a^b p_0(x)dx = (b-a)f\left(\frac{a+b}{2}\right) \approx I(f).$$

Thus $w_0 = b - a$.

1. The *trapezoidal* rule uses a first order polynomial interpolating f in two points, $x_0 = a$ and $x_1 = b$. Then

$$p_1(x) = \frac{b-x}{b-a}f(a) + \frac{x-a}{b-a}f(b)$$

and, denoting the rule $T(f)$, we get

$$T(f) := \int_a^b p_1(x)dx = (b-a)\frac{f(a)+f(b)}{2} \approx I(f).$$

Thus $w_0 = w_1 = (b-a)/2$.

2. *Simpson's* rule uses a second order polynomial interpolating f in three points, $x_0 = a$, $x_1 = (b+a)/2$ and $x_2 = b$. Then

$$p_2(x) = f(a) + [f(b) - f(a)]\frac{x-a}{b-a} - \left[f(a) - 2f\left(\frac{a+b}{2}\right) + f(b)\right]\frac{2(x-a)(b-x)}{(b-a)^2},$$

and, denoting the rule $S(f)$, we get

$$S(f) := \int_a^b p_2(x)dx = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \approx I(f).$$

Thus $w_0 = w_2 = (b-a)/6$ and $w_1 = 2(b-a)/3$.

Higher order Newton–Cotes are seldom used. The problems encountered for high order equidistant polynomial interpolation (see lecture notes on approximation theory) carry over also to interpolatory integration formulae. Instead one uses *composite* versions of low order Newton–Cotes rules: see below.

2 Simple and Composite Rules

For any quadrature rule we can also consider its *composite* version. Composite rules divide the interval $[a, b]$ into m (often equidistant) subintervals and apply the quadrature rule in each one of them. We call the original rule a *simple* rule to distinguish it from the composite version. If the simple rule is interpolatory, then the composite rule corresponds to exact integration of *piecewise polynomials*.

The simple rules as we derived them above are only defined for a specific interval $[a, b]$; the weights w_j and the nodes x_j depend on the interval end points a and b. We can however easily apply them to a general interval $[\alpha, \beta]$ by rescaling the x-variable. Let us write $Q_k(f, a, b)$ to indicate that the quadrature formula is applied to the interval $[a, b]$ and suppose that the original rule is the $k+1$-point formula

$$Q_{k+1}(f, a, b) = \sum_{j=0}^{k} w_j f(x_j) \approx \int_a^b f(x)dx.$$

Then, by the change of variables $x \mapsto \rho(x-a)+\alpha$ with $\rho = (\beta-\alpha)/(b-a)$, we have

$$\int_\alpha^\beta f(x)dx = \rho \int_a^b f\Big(\rho(x-a)+\alpha\Big)dx.$$

The last integral is over the interval $[a,b]$ and we use $Q_{k+1}(f,a,b)$ to approximate it

$$\int_\alpha^\beta f(x)dx \approx \sum_{j=0}^{k} w_j \rho f\Big(\rho(x_j-a)+\alpha\Big) =: Q_{k+1}(f,\alpha,\beta).$$

This hence defines $Q_{k+1}(f,\alpha,\beta)$ for general intervals $[\alpha,\beta]$. We are now ready to define the composite rule corresponding to a simple k-point rule $Q_k(t,a,b)$. We divide the interval $[a,b]$ into m subintervals separated by $m+1$ points $\{y_i\}$, which we for simplicity take to be equidistant,

$$y_i = a+ih, \qquad h = \frac{b-a}{m}. \tag{3}$$

Then, clearly

$$\int_a^b f(x)dx = \sum_{i=0}^{m-1} \int_{y_i}^{y_{i+1}} f(x)dx,$$

which motivates the composite rule $\mathbb{Q}_n$,

$$\mathbb{Q}_n(f,a,b) := \sum_{i=0}^{m-1} Q_k(f,y_i,y_{i+1}). \tag{4}$$

Here n represents the *total* number of nodes (function evaluations) used in the composite rule, which is of the order k (points per subinterval) multiplied by m (number of subintervals). More precisely, $n=(k-1)m+1$ if Q_k is a closed rule (the interval end points belongs to the nodes) and $n=km$ if Q_k is an open rule (the interval end points do not belong to the nodes).

Example. The composite versions of the first Newton–Cotes rules that we introduced above are as follows.

0. The *composite midpoint* rule. The midpoint rule is open so $m=n$. Therefore,

$$\mathbb{M}_n(f) = \sum_{i=0}^{n-1} M(f,y_i,y_{i+1}) = \sum_{i=0}^{n-1} (y_{i+1}-y_i) f\left(\frac{y_i+y_{i+1}}{2}\right).$$

   We thus have

$$\mathbb{M}_n(f) = h\Big(f(x_0)+\cdots+f(x_{n-1})\Big), \qquad x_j = a+(j+1/2)h, \qquad h=\frac{b-a}{n}.$$

1. The *composite trapezoidal* rule. Here $n=m+1$ and

$$\begin{aligned}\mathbb{T}_{n+1}(f) &= \sum_{i=0}^{n-1} T(f,y_i,y_{i+1}) = \sum_{i=0}^{n-1}(y_{i+1}-y_i)\frac{f(y_i)+f(y_{i+1})}{2} \\ &= \frac{h}{2}f(y_0)+\frac{h}{2}f(y_n)+h\sum_{i=1}^{n-1} f(y_i),\end{aligned}$$

which gives

$$\mathbb{T}_{n+1}(f) = h\left(\frac{1}{2}f(x_0) + f(x_1) + \cdots + f(x_{n-1}) + \frac{1}{2}f(x_n)\right), \tag{5}$$

with $x_j = a + jh$ and $h = \frac{b-a}{n}$.

2. The *composite Simpson's* rule. Here $n = 2m + 1$ and

$$\begin{aligned}\mathbb{S}_{2m+1}(f) &= \sum_{i=0}^{m-1} S(f, y_i, y_{i+1}) = \sum_{i=0}^{m-1} \frac{y_{i+1} - y_i}{6}\left(f(y_i) + 4f\left(\frac{y_i + y_{i+1}}{2}\right) + f(y_{i+1})\right) \\ &= \frac{h}{6}f(y_0) + \frac{h}{6}f(y_m) + \frac{h}{3}\sum_{i=1}^{m-1} f(y_i) + \frac{2h}{3}\sum_{i=1}^{m-1} f\left(\frac{y_i + y_{i+1}}{2}\right).\end{aligned}$$

This eventually gives

$$\mathbb{S}_{n+1}(f) = \frac{\tilde{h}}{3}\left(f(x_0) + 4\sum_{\substack{j \text{ odd} \\ 0<j<n}} f(x_j) + 2\sum_{\substack{j \text{ even} \\ 0<j<n}} f(x_j) + f(x_n)\right),$$

with $x_j = a + j\tilde{h}$ and $\tilde{h} = \frac{h}{2} = \frac{b-a}{n}$. (Note that the total number of points $n + 1$ must be an odd number here.)

3 Quadrature Error

The *quadrature error*,

$$E(f) = I(f) - Q_n(f),$$

measures the error in the numerical approximation. Evaluating f at many points may be computationally expensive, or difficult for other reasons, e.g. if f is given by measurements. It is therefore important to consider how the error E depends on n. Obviously, one would like E to become small when n becomes large. There are two strongly related ways of characterizing this dependence: (1) The *order of accuracy* and (2) the *degree of precision* of the quadrature rule.

Order of accuracy

The order of accuracy is mainly used for (equidistant) composite rules. It shows how fast $|E|$ decays to zero when we take a smaller distance h between nodes. Since $n \sim 1/h$ in these rules (in 1D), it similarly describes how fast $E \to 0$ when $n \to \infty$. We define the order of accuracy as the largest number p for which there exists a constant C, independent of h (and n) for which

$$|E(f)| \leq C\, h^p. \tag{6}$$

The order of accuracy usually depends both on the quadrature rule Q_n and on the regularity of the integrand f, i.e. how many continuous derivatives f has. When Q_n gives an order of accuracy of p for all sufficiently smooth f (typically all $f \in C^p$) we say that Q_n is a p-th order (accurate) method.

Degree of precision

The degree of precision is used for simple quadrature rules. It shows for which polynomials the rule is *exact*. A quadrature formula has degree of precision d if it integrates all polynomials up to degree d exactly (but not all of degree $d+1$). It can equivalently be defined as the largest number d for which the quadrature formula is exact when applied to monomials 1, x, x^2, ..., x^d, but not exact for x^{d+1} (see Appendix). The degree of precision is thus easy to determine for a given quadrature formula. We just need to check which monomials it integrates exactly. For the trapezoidal rule we have for instance

$$T(1) = \int_a^b dx = b-a, \qquad T(x) = \int_a^b x dx = \frac{1}{2}(b^2-a^2), \qquad T(x^2) \neq \int_a^b x^2 dx = \frac{1}{3}(b^3-a^3).$$

We then conclude that the degree is $d = 1$. Furthermore, it is easy to check that also the midpoint rule has degree $d = 1$ and that Simpson's rule has degree $d = 3$. Note that for any interpolatory $n+1$-point quadrature formula d is at least n. This is because if $f \in \mathbb{P}_n$, then, by the uniqueness of the interpolation polynomial, $p(x)$ in (1) and $f(x)$ coincide, so

$$Q_{n+1}(f,a,b) := \int_a^b p(x)dx = \int_a^b f(x)dx = I(f), \qquad \forall f \in \mathbb{P}_n.$$

As we indicated above, the order of accuracy and the degree of precision are related. Essentially they are two sides of the same coin. We have the following theorem.

Theorem 1 *If a simple interpolatory quadrature formula has degree of precision d, then the corresponding (equidistant) composite formula is $d+1$ order accurate for integrands $f \in C^{d+1}$.*

(The proof (secondary importance) is given in the Appendix.) From the theorem it follows that the composite versions of the midpoint and the trapezoidal rules have the same convergence properties: order 2 when $f \in C^2$. For Simpson's rule the composite version is of order 4 when $f \in C^4$. For a general n-point Newton–Cotes formulae we have

$$\text{degree} = \begin{cases} n-1, & n \text{ even}, \\ n, & n \text{ odd}, \end{cases} \qquad \text{order of accuracy} = \begin{cases} n, & n \text{ even}, \ f \in C^n, \\ n+1, & n \text{ odd}, \ f \in C^{n+1}. \end{cases}$$

Some remarks:

- The integrand f must be smooth enough to get the stated order of accuracy: $f \in C^{d+1}$ is sufficient for a simple degree d formula. If f is less regular, and we only have $f \in C^s$ for some $s < d+1$, but $f \notin C^{d+1}$, then the order of accuracy will typically be less than $d+1$. There is thus usually no point in using high order methods for functions with low regularity.
- When f is *periodic* and smooth the Newton–Cotes formulae are *considerably* more accurate than indicated above. In fact the quadrature error $E(f)$ typically decays *exponentially* with $n \sim 1/h$ rather than just algebraically, as in (6) even for the midpoint and trapezoidal rule. This is because when f is periodic, exact integration of interpolating piecewise constant/linear polynomials agrees with exact integration of interpolating trigonometric functions (Fourier interpolation), which is very accurate when f is periodic and smooth.

4 Numerical Integration in Higher Dimensions

For multiple integrals the one-dimensional quadrature rules can be generalized in several different ways. How it is done best depends on many factors including the shape of the integration domain and the information available about the integrand. Here we discuss a few particular cases.

4.1 Trapezoidal Rule in 2D on a Rectangle

We consider double integrals

$$I = \iint_\Omega f(x,y)dxdy,$$

where Ω is the rectangular domain $[a,b]\times[c,d]$. In this case the integral can be written in iterated form as,

$$I = \int_a^b \left(\int_c^d f(x,y)dy \right) dx.$$

Numerically we can then do the same thing. The inner integral is approximated by a one-dimensional quadrature rule where x is held fixed. The values obtained are then used to approximate the outer integral, also by a one-dimensional rule. To make this more precise, let us consider the trapezoidal rule in two dimensions. Set

$$F(x) := \int_c^d f(x,y)dy \quad \Rightarrow \quad I = \int_a^b F(x)dx.$$

Then we approximate I with the one-dimensional (composite) trapezoidal rule,

$$I \approx \mathbb{T}_{n+1}(F) = h_x \left(\frac{1}{2}F(x_0) + F(x_1) + \cdots + F(x_{n-1}) + \frac{1}{2}F(x_n) \right),$$

where

$$x_j = a + jh_x, \qquad h_x = \frac{b-a}{n}.$$

Hence n is the number of points used in the x-direction and h_x is the distance between x-points. In order to compute $F(x_j)$ we approximate again, this time in the y-direction, with m points,

$$\begin{aligned} F(x_j) &= \int_c^d f(x_j,y)dy \approx \mathbb{T}_{m+1}(f(x_j,\,\cdot\,)) \\ &= h_y \left(\frac{1}{2}f(x_j,y_0) + f(x_j,y_1) + \cdots + f(x_j,y_{m-1}) + \frac{1}{2}f(x_j,y_m) \right), \end{aligned}$$

where

$$y_j = c + jh_y, \qquad h_y = \frac{d-c}{m}.$$

The setup and notation is shown in Figure 1. In general $n \neq m$ and $h_x \neq h_y$ here. However, typically n and m are chosen such that $h_x \approx h_y$ as this tends to minimize the computational cost for a fixed accuracy, when the integrand $f(x,y)$ is smooth and slowly varying. Putting together the formulae we obtain

$$I \approx \sum_{j=0}^{n} w_j^x F(x_j)h_x = \sum_{j=0}^{n}\sum_{k=0}^{m} w_j^x w_k^y f(x_j,y_k)h_x h_y,$$

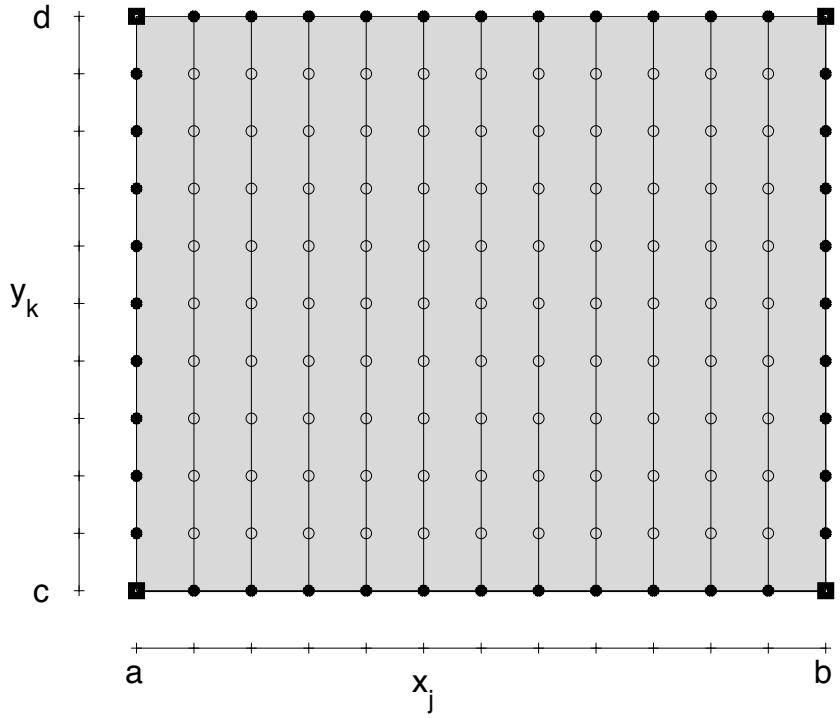

Figure 1. Notation for the two-dimensional trapezoidal rule on a rectangle. The shaded domain is Ω. Circles indicate the points (x_j, y_k).

where the weights equal $1/2$ in the end points of the intervals, $w_0^x = w_n^x = w_0^y = w_m^y = 1/2$ and equal 1 elsewhere, $w_j^x = 1$ for $j = 1, \ldots, n$ and $w_j^y = 1$ for $j = 1, \ldots, m$. The sum can be further simplified to

$$I \approx \sum_{j=0}^{n} \sum_{k=0}^{n_j} w_{j,k} f(x_j, y_k) h_x h_y,$$

where $w_{j,k}$ is 1 for points in the interior (open circles in Figure 1), $1/2$ along the outer edges (filled circles) and $1/4$ in the corner points (filled squares).

The order of accuracy is two and

$$\text{error} = \mathcal{O}(h_x^2 + h_y^2).$$

4.2 Trapezoidal Rule in 2D: More General Domains

We now consider a more general domain, given as the area between two function graphs $r(x)$ and $s(x)$,

$$\Omega = \left\{ (x, y) \in \mathbb{R}^2 \ : \ a \leq x \leq b, \ r(x) \leq y \leq s(x) \right\}.$$

The integral can again be written in iterated form as,

$$I = \int_a^b \left(\int_{r(x)}^{s(x)} f(x, y) dy \right) dx.$$

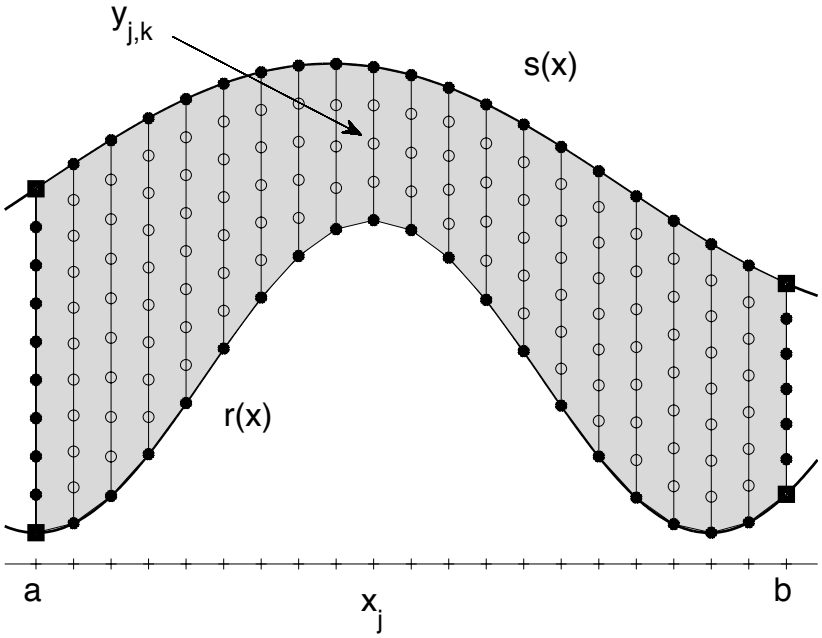

Figure 2. Notation for the two-dimensional trapezoidal rule on a general domain. The shaded domain is Ω. Circles indicate the points $(x_j, y_{j,k})$.

We can treat this in the same way as the rectangular domain. In the x-direction, we do almost exactly the same thing as before. We set

$$F(x) := \int_{r(x)}^{s(x)} f(x,y)dy \quad \Rightarrow \quad I = \int_a^b F(x)dx,$$

and then approximate using n points

$$I \approx \mathbb{T}_{n+1}(F) = h_x \left(\frac{1}{2}F(x_0) + F(x_1) + \cdots + F(x_{n-1}) + \frac{1}{2}F(x_n) \right),$$

where $x_j = a + jh_x$ and $h_x = (b-a)/n$. However, in the y-direction, we will need a discretization that changes from one x-point to another. The difference is illustrated in Figure 2. At $x = x_j$ we approximate using m_j points, and a step length of $h_{y,j}$

$$\begin{aligned} F(x_j) &= \int_{r(x_j)}^{s(x_j)} f(x_j, y)dy \approx \mathbb{T}_{m_j+1}(f(x_j, \,\cdot\,)) \\ &= h_{y,j} \left(\frac{1}{2}f(x_j, y_{j,0}) + f(x_j, y_{j,1}) + \cdots + f(x_j, y_{j,n_j-1}) + \frac{1}{2}f(x_j, y_{j,n_j}) \right), \end{aligned}$$

where

$$y_{j,k} = r(x_j) + kh_{y,j}, \qquad h_{y,j} = \frac{s(x_j) - r(x_j)}{m_j}.$$

Note that the number of points in the y-direction, m_j, and the step length, $h_{y,j}$, are in general different for different j. As in the rectangular case, the choice of m_j is usually taken such that $h_x \approx h_{y,j}$ for all j. The total formula becomes

$$I \approx \sum_{j=0}^{n} \sum_{k=0}^{m_j} w_{j,k} f(x_j, y_{j,k}) h_x h_{y,j},$$

where, as before, $w_{j,k}$ is 1 for points in the interior (open circles in Figure 2), 1/2 along the outer edges (filled circles) and 1/4 in the corner points (filled squares).

The order of accuracy is two and

$$\text{error} = \mathcal{O}(h_x^2 + \max_j h_{y,j}^2).$$

4.3 Trapezoidal Rule in 3D

For triple integrals

$$I = \iiint_\Omega f(x,y,z)dxdydz,$$

we can use the same strategy as in 2D if Ω is the volume between two function graphs over a simple 2D domain. Here we take a rectangle and consider Ω of the type

$$\Omega = \left\{(x,y,z) \in \mathbb{R}^3 \ : \ a \le x \le b, \quad c \le y \le d, \quad r(x,y) \le z \le s(x,y)\right\}.$$

The iterated form of the integral is

$$I = \int_a^b \left(\int_c^d \left(\int_{r(x,y)}^{s(x,y)} f(x,y,z)dz \right) dy \right) dx.$$

As above, define

$$F(x,y) := \int_{r(x,y)}^{s(x,y)} f(x,y,z)dz \quad \Rightarrow \quad I = \int_a^b \left(\int_c^d F(x,y)dy \right) dx.$$

We can now use the 2D trapezoidal rule on a rectangle for this integral,

$$I \approx \sum_{j=0}^{n}\sum_{k=0}^{m} w_{j,k}F(x_j,y_k)h_x h_y, \qquad h_x = \frac{b-a}{n}, \quad h_y = \frac{d-c}{m},$$

where x_j, y_k and the weights $w_{j,k}$ were defined above. Finally, $F(x,y)$ is approximated with the one-dimensional trapezoidal rule for fixed (x,y), in the same way as in the 2D case,

$$\begin{aligned} F(x_j,y_k) &\approx \mathbb{T}_{p_{j,k}+1}(f(x_j,y_k,\,\cdot\,)) \\ &= h_{z,j,k}\left(\frac{1}{2}f(x_j,y_k,z_{j,k,0}) + f(x_j,y_k,z_{j,k,1}) + \cdots + f(x_j,y_k,z_{j,k,p_{j,k}-1}) + \frac{1}{2}f(x_j,y_k,z_{j,k,p_{j,k}})\right). \end{aligned}$$

Here $p_{j,k}$ is the number of points used in the z-direction at (x_j,y_k) and $h_{z,j,k}$ is the step length,

$$z_{j,k,\ell} = s(x_j,y_k) + \ell h_{z,j,k}, \qquad h_{z,j,k} = \frac{s(x_j,y_k) - r(x_j,y_k)}{p_{j,k}}.$$

As before we aim to choose n, m and $p_{j,k}$ such that $h_{z,j,k} \approx h_x \approx h_y$. The final formula is

$$I \approx \sum_{j=0}^{n}\sum_{k=0}^{m}\sum_{\ell=0}^{p_{j,k}} w_{j,k,\ell}f(x_j,y_k,z_{j,k,\ell})h_x h_y h_{z,j,k},$$

with $w_{j,k,\ell}$ being 1 for points in the interior, 1/2 for points in the interior of the bounding surfaces, 1/4 on the boundary edges and 1/8 in the corner points. The order of accuracy is two and

$$\text{error} = \mathcal{O}(h_x^2 + h_y^2 + \max_{j,k} h_{j,k}^2).$$

4.4 Quadrature Rules on Triangulated Surfaces

Another very common approach to numerical integration in two dimensions is to make a *triangulation* of the domain Ω, see Figure 3 for an example. This makes it possible to treat more complicated domains. Moreover, when solving PDEs the function values are often only available in this form.

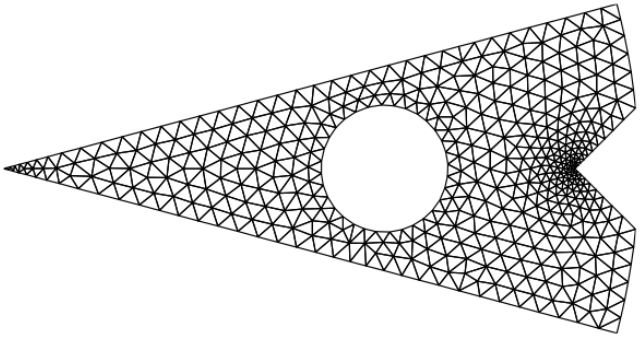

Figure 3. Example of a triangulated domain.[1]

The strategy to devise quadrature rules on triangulated domains is similar to the one-dimensional case, with the small triangles taking the role of the small intervals in 1D. First, simple rules are defined for integration over a single triangle. The degree of precision d is defined as the highest degree of polynomials (in two variables) that are integrated exactly by the rule. Second, composite rules are constructed by taking the sum of the simple rule over all small triangles. The order of accuracy is again $d+1$ when the integrand is smooth enough, and the error is $\mathcal{O}(h^{d+1})$, where h is the largest radius of the circumscribed circles of all triangles.

In this way, generalization of the standard Newton–Cotes formulae can be obtained. For example, in the simple trapezoidal rule the integrand f is approximated by the plane passing through the function values in the triangle corners. Then the composite rule integrates exactly a piecewise linear interpolant of f. In the simple midpoint rule, f is approximated by a constant value, namely f evaluated at the barycenter of the triangle. The composite version integrates exactly a piecewise constant interpolant.

In three dimensions, domains can be divided into a large collection of tetrahedra, see Figure 4. Simple and composite quadrature rules can then be built using the tetrahedron as the smallest unit, in the same way as the triangle is used in 2D and the interval in 1D.

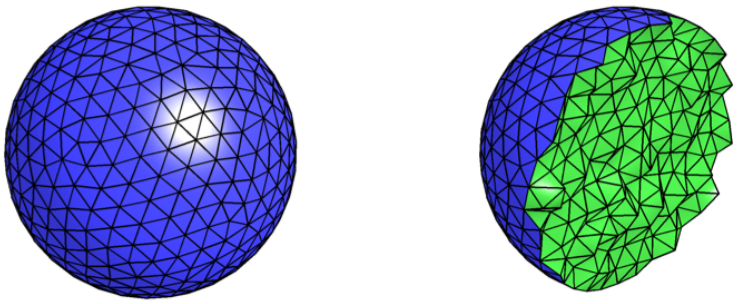

Figure 4. Example of using tetrahedra to describe a sphere in 3D.[1]

[1]Figure 3 and Figure 4 taken from P.-O. Persson, G. Strang, A Simple Mesh Generator in MATLAB. *SIAM*

4.5 Monte–Carlo Integration

Numerical integration with the type of quadrature rules described above becomes more expensive as the dimension of the integral increases. The error is always proportional to the longest distance h between the points in which the integrand is evaluated, taken to some power p (the order of accuracy). For instance, in the different versions of the trapezoidal rule the error is $\mathcal{O}(h^2)$, so $p = 2$. On the other hand, the total number of function evaluations N grows as h^{-d} in d dimensions. This means that, in terms of N the error is $\mathcal{O}(N^{-p/d})$, and thus decreases slowly with N when d is large. As a result, the computational cost to achieve a certain given accuracy grows as d increases.

There are many applications, from quantum mechanics, systems biology to finance, where numerical integration needs to be done in very high dimensions; d could easily be in the order of hundreds. For problems of that type other methods must be used. One such method is Monte–Carlo integration. It is based on the observation that if X is a stochastic variable, uniformly distributed in $\Omega \subset \mathbb{R}^d$, then

$$\frac{1}{|\Omega|} \int_\Omega f(\boldsymbol{x}) d\boldsymbol{x} = \mathbb{E}(f(X)),$$

where $|\Omega|$ is the volume of Ω, and $\mathbb{E}(Y)$ denotes the expected value (mean value) of Y. Hence, the integral corresponds to the expected value of the stochastic variable $f(X)$. Consequently, the integral can be approximated by taking randomly chosen points $X_1, X_2, \ldots, X_N \in \Omega$ and computing the sample mean of $f(X_j)$,

$$\int_\Omega f(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{|\Omega|}{N} \sum_{j=1}^{N} f(X_j). \tag{7}$$

From probability theory we know that this estimate of the expected value has a standard deviation which is proportional to $N^{-1/2}$. With high probability, the error in the estimate (7) is therefore of the order $\mathcal{O}(N^{-1/2})$. The great advantage of Monte–Carlo integration is that this error is *independent* of the dimension. Thus, compared to a standard method with order of accuracy p, the error in Monte–Carlo integration dies off faster with N if $p/d < 1/2$, i.e. $d > 2p$.

A More on quadrature errors

A.1 Degree of precision definitions

The degree of precision of a quadrature formula was defined as d if it integrates all polynomials of degree at most d exactly, but not all polynomials of degree $d+1$. The implication of the next theorem is that this is equivalent to integrating the monomials 1, x, x^2, …, x^d exactly, but not x^{d+1}.

Theorem 2 *A simple interpolatory quadrature formula is exact for all $p \in \mathbb{P}_d$ if and only if it is exact for the monomials 1, x, x^2, …, x^d.*

Proof: This is a simple consequence of the linearity of interpolatory rules. Take a polynomial $p \in \mathbb{P}_d$, which has the general form

$$p(x) = \sum_{k=0}^{d} c_k x^k.$$

Review, 46(2), pp. 329–345, 2004.

Then

$$E(p) = \int_a^b p(x)dx - Q_n(p) = \int_a^b p(x)dx - \sum_{j=0}^{n} w_j p(x_j) = \sum_{k=0}^{d} c_k \int_a^b p(x)dx - \sum_{j=0}^{n}\sum_{k=0}^{d} c_k w_j x_j^k$$
$$= \sum_{k=0}^{d} c_k \left(\int_a^b x^k dx - Q_n(x^k) \right) = \sum_{k=0}^{d} c_k E(x^k).$$

Hence, Q_n is exact for all $p \in \mathbb{P}_d$ if and only if Q_n is exact for $\{x^k\}_{k=0}^d$. $\square$

A.2 Proof of Theorem 1

Suppose the simple rule is a $(k+1)$-point quadrature formula

$$Q_{k+1}(f,a,b) = \sum_{j=0}^{k} w_j f(x_j),$$

with degree of precision d. We start by considering the general interval $[\alpha,\beta] \subset [a,b]$. Let $p_T \in \mathbb{P}_d$ be the d-th degree Taylor polynomial for f around α,

$$p_T(x) = f(\alpha) + (x-\alpha)f'(\alpha) + \frac{(x-\alpha)^2}{2}f''(\alpha) + \cdots + \frac{(x-\alpha)^d}{d!}f^{(d)}(\alpha).$$

Since $f \in C^{d+1}([a,b])$ we can estimate the remainder term by Taylor's formula,

$$\sup_{\alpha\le x\le\beta} |f(x) - p_T(x)| \le \sup_{\alpha\le x\le\beta} \frac{||f^{(d+1)}||_\infty}{(d+1)!}|x-\alpha|^{d+1} \le C|\beta-\alpha|^{d+1},$$

where the constant C does not depend on α or β. Let $\tilde{x}_j = \rho(x_j - a) + \alpha \in [\alpha,\beta]$ and recall that $\rho = (\beta-\alpha)/(b-a)$. Since also the scaled p_T, i.e. $p_T(\rho(x-a)+\alpha)$ belong to $\mathbb{P}_d$, the quadrature formula $Q_{k+1}(p_T,\alpha,\beta)$ is exact and

$$\begin{aligned}
\left| \int_\alpha^\beta f(x)dx - Q_{k+1}(f,\alpha,\beta) \right| &= \left| \int_\alpha^\beta f(x) - p_T(x)dx - Q_{k+1}(f-p_T,\alpha,\beta) \right| \\
&\le \int_\alpha^\beta |f(x) - p_T(x)|dx + \frac{\beta-\alpha}{b-a}\sum_{j=0}^{k} |w_j|\,|f(\tilde{x}_j) - p_T(\tilde{x}_j)| \\
&\le C|\beta-\alpha|^{d+2}\left(1 + \sum_{j=0}^{k}\frac{|w_j|}{b-a}\right) = C'|\beta-\alpha|^{d+2},
\end{aligned}$$

where C' is another constant, independent of α and β. From the definition of the composite rule in (3) and (4) we now have,

$$\begin{aligned}
\left| \mathbb{Q}_n(f,a,b) - \int_a^b f(x)dx \right| &\le \sum_{i=0}^{m-1} \left| Q_{k+1}(f,y_i,y_{i+1}) - \int_{y_i}^{y_{i+1}} f(x)dx \right| \\
&\le \sum_{i=0}^{m-1} C'|y_{i+1}-y_i|^{d+2} = C'\,mh^{d+2} = C'\,\frac{(b-a)^{d+2}}{m^{d+1}},
\end{aligned}$$

where $n = km + 1$ for closed rules, and $n = (k+1)m$ for closed rules. Then, recalling that $h = (b-a)/n$, we get, since $(b-a)/m \leq 2kh$ when $n \geq 2$ (both for closed and open formulae),

$$\left| \mathbb{Q}_n(f, a, b) - \int_a^b f(x)dx \right| \leq C'' \; h^{d+1}$$

for some constant C'' independent of h. This shows that the order of accuracy is at least $d+1$. By considering a polynomial in $\mathbb{P}_{d+1}$ which Q_{k+1} does not integrate exactly, one can also show that the order is not higher. This is left as an exercise for the reader.