Numerical integration

Siltanen/Railo/Kaarnioja

Spring 2018
Applications of matrix computations

When computing integrals by hand, one usually employs the *fundamental theorem of calculus*.

Theorem

Let $f : [a, b] \to \mathbb{R}$ be a continuous function. If F is the antiderivative of f satisfying $F' = f$, then

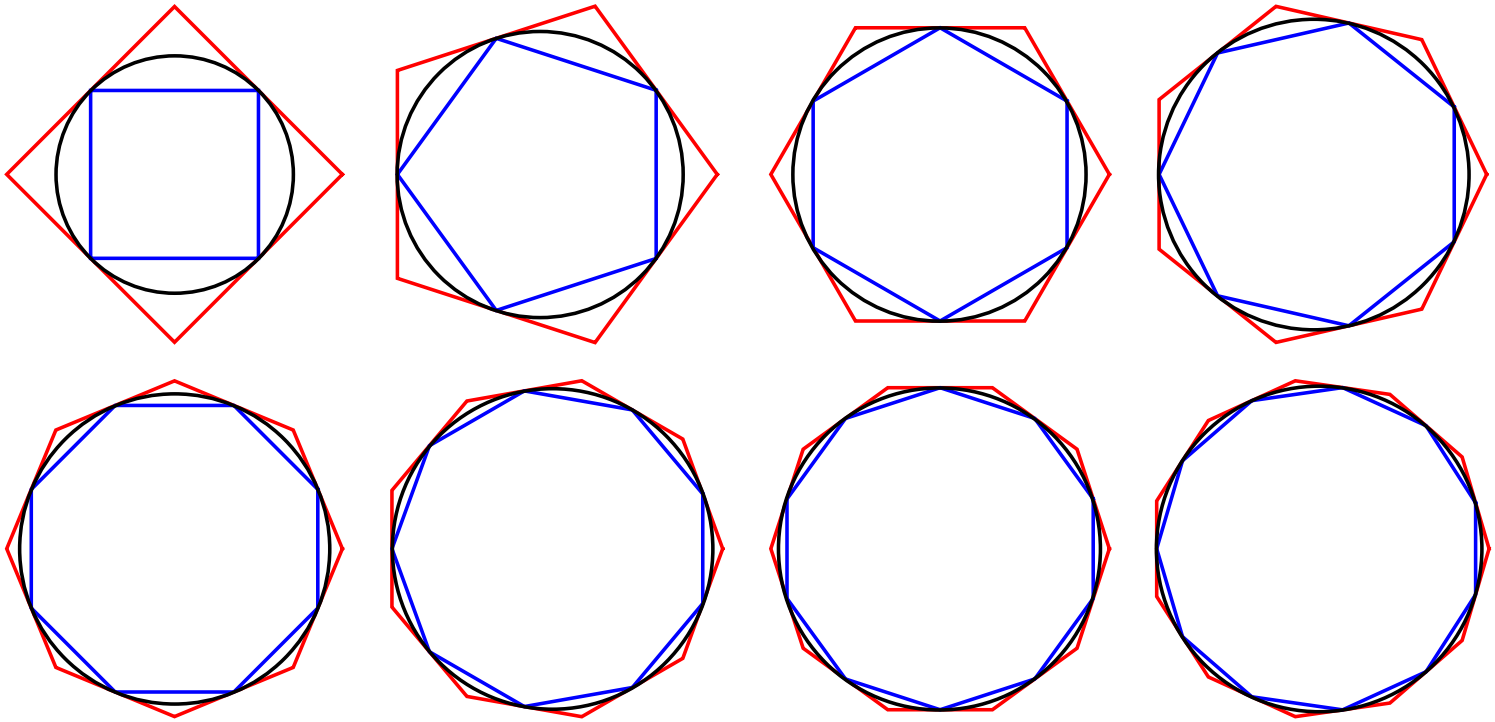
$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a).$$

However, in general there is no guarantee that the antiderivative F of an arbitrary function f even has a closed form expression. A famous example is given by $f(x) = \mathrm{e}^{-x^2}$. The situation becomes worse when the integrand f itself does not have a nice closed form expression: this is the case for example when f is the solution of an ODE or a PDE, and we are interested in computing its moments or marginalizing certain variables.

One can however always resort to numerical integration!

Quadrature of the circle

Numerical integration is also called *quadrature*, a term originating from the Ancient Greeks which refers to the estimation of the area of the unit disk using pen, ruler, and compass. Below using polygons:

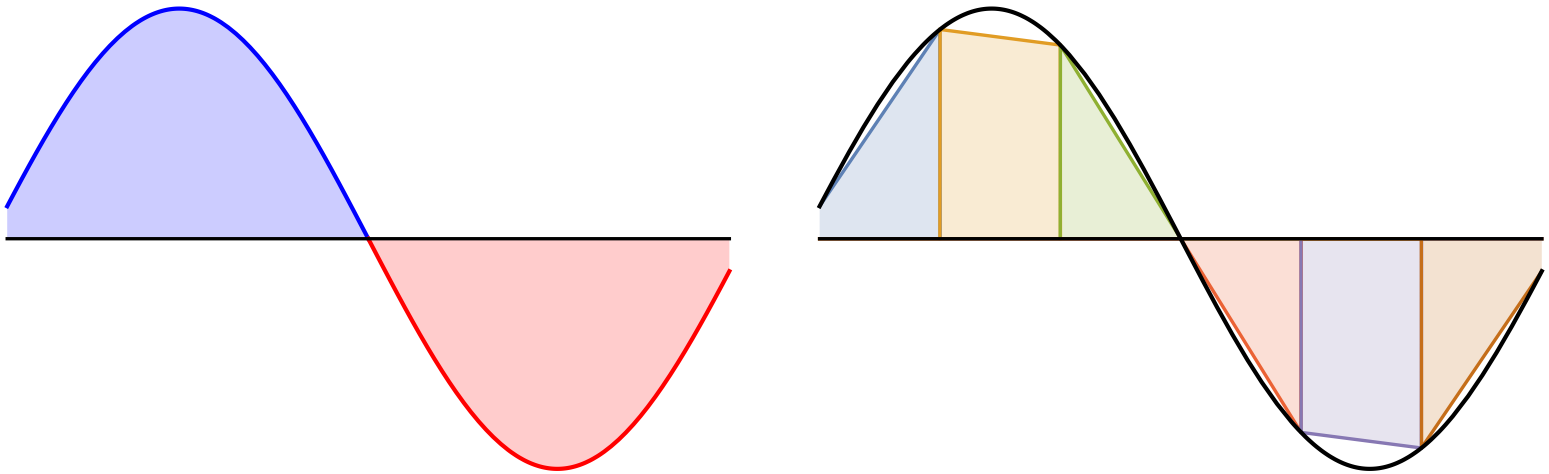


For the 11-sided polygons inscribing and circumscribing the unit disk with area π, we obtain the estimate $2.97 \leq \pi \leq 3.23$.

Geometrically, for a continuous function $f : [a, b] \to \mathbb{R}$ the value of the integral

$$\int_a^b f(x)\,\mathrm{d}x$$

is equal to the signed area between the graph $\{(x, f(x)) \mid x \in [a, b]\}$ and the x-axis.



We can obtain quadratures for integrals by partitioning the signed area using polygons and other shapes whose area we are able to calculate!

Polynomial interpolation

Before we tackle quadratures, let's briefly touch on a closely related topic of polynomial interpolation.

Let $f : [a, b] \to \mathbb{R}$ be a function and let $x_1, \ldots, x_n \in [a, b]$ be a set of mutually distinct nodes: $x_i \neq x_j$ whenever $i \neq j$.

It is easy to see that the coordinates $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$ can be interpolated by a unique polynomial L_{n-1} with degree $\leq n - 1$ such that $f(x_i) = L_{n-1}(x_i)$, $i \in \{1, \ldots, n\}$.

To see this, write $L_{n-1}(x) = c_{n-1}x^{n-1} + \cdots + c_1 x + c_0$ with undetermined coefficients $c_0, \ldots, c_{n-1} \in \mathbb{R}$. Then the conditions $f(x_i) = L_{n-1}(x_i)$, $i \in \{1, \ldots, n\}$, can be used to form the following linear system of equations:

$$\begin{cases} c_0 + c_1 x_1 + \cdots + c_{n-1} x_1^{n-1} = f(x_1) \\ \vdots \\ c_0 + c_1 x_n + \cdots + c_{n-1} x_n^{n-1} = f(x_n). \end{cases}$$

In matrix form, $V^{\mathrm{T}}[c_0, c_1, \ldots, c_{n-1}]^{\mathrm{T}} = [f(x_1), \ldots, f(x_n)]^{\mathrm{T}}$, where $V_{i,j} = x_j^{i-1}$ is called the *Vandermonde matrix*.

Now the *Vandermonde determinant* is

$$\det V = \det V^{\mathrm{T}} = \prod_{1 \le i < j \le n} (x_j - x_i) \neq 0$$

whenever $x_i \neq x_j$ for all $i \neq j$. Thus, the solution vector $[c_0, c_1, \ldots, c_{n-1}]^{\mathrm{T}}$ of the above system is unique and, in consequence, the interpolating polynomial of degree $n - 1$ is unique as well.

The interpolating polynomial L_{n-1} is called the *Lagrange interpolating polynomial* and it is given by

$$L_{n-1}(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x),$$

where the auxiliary polynomials ℓ_j are given by

$$\ell_j(x) = \prod_{\substack{1 \le i \le n \\ i \neq j}} \frac{x - x_i}{x_j - x_i} \quad \text{for } j \in \{1, \ldots, n\}$$

and they satisfy $\ell_i(x_j) = \delta_{i,j}$. Here, $\delta_{i,j}$ denotes the Kronecker delta function and it is defined as unity whenever the indices coincide and vanishing otherwise.

Newton–Cotes formulae

Let $a = x_0 < x_1 < \cdots < x_n \leq b$ be an equispaced partition[1] of the interval $[a, b]$, i.e.,

$$x_k = a + \frac{b-a}{n} k \quad \text{for } k \in \{0, 1, \ldots, n\}.$$

We denote the mesh size by

$$h = \frac{b-a}{n}.$$

Suppose that $f : [a, b] \to \mathbb{R}$ is a continuous function. How to find approximations of the integral

$$\int_a^b f(x)\,\mathrm{d}x$$

using point evaluations of f?

[1] The basic tenets behind Newton–Cotes formulae work even with unequally spaced nodes, but the resulting formulae are a bit nicer when the partition is done in this way.

Midpoint rule

Let $a = x_0 < x_1 < \cdots < x_n = b$ be an equispaced partition of the interval $[a, b]$. The signed area of each rectangle is given by

$$R_i = (x_i - x_{i-1}) f\left(\frac{x_{i-1} + x_i}{2}\right).$$

We thus obtain the approximation

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} (x_i - x_{i-1}) f\left(\frac{x_{i-1} + x_i}{2}\right)$$
$$= h \sum_{k=1}^{n} f\left(a + \frac{b-a}{2n}(2k-1)\right).$$

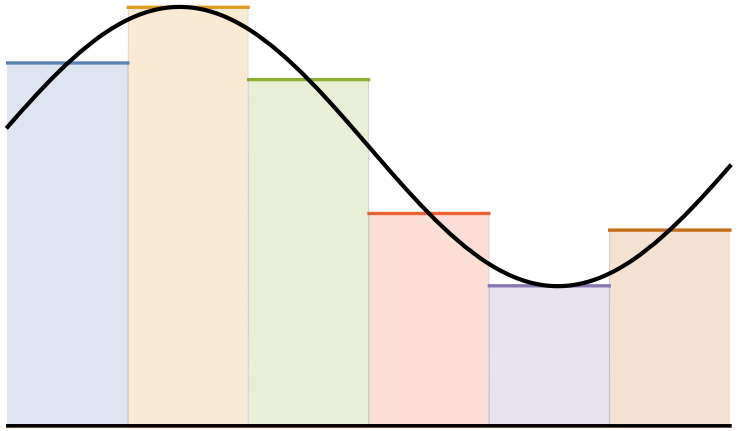
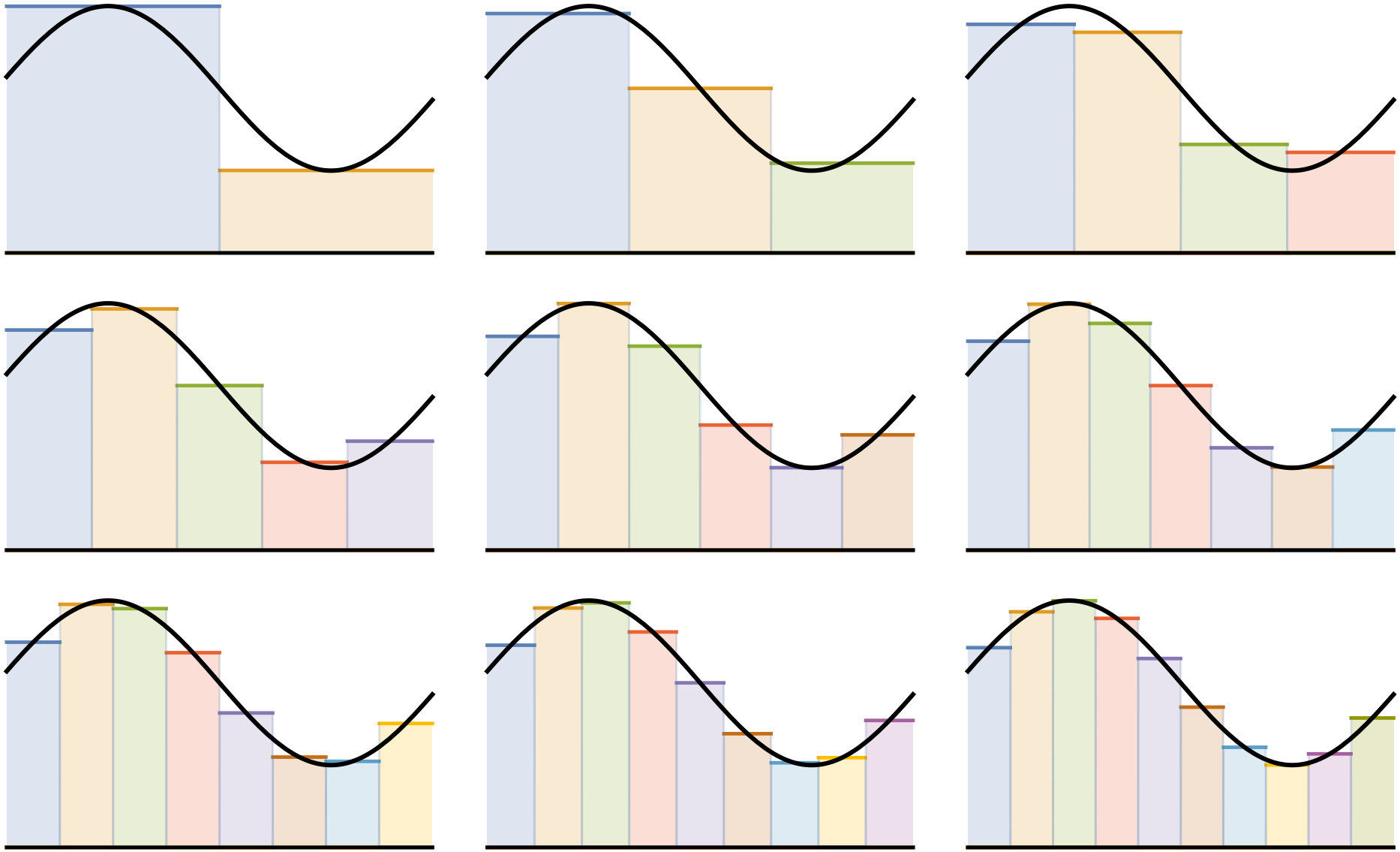


Figure: The midpoint rule can be seen as piecewise constant interpolation at the nodes $(\tilde{x}_i, f(\tilde{x}_i))$, where $\tilde{x}_i = \frac{x_{i-1}+x_i}{2}$.



Trapezoidal rule

Let $a = x_0 < x_1 < \cdots < x_n = b$ be an equispaced partition of the interval $[a, b]$. The signed area of each trapezoid is given by

$$T_i = (x_i - x_{i-1}) \frac{f(x_i) + f(x_{i-1})}{2}.$$

We thus obtain the approximation

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} (x_i - x_{i-1}) \frac{f(x_{i-1}) + f(x_i)}{2}$$

$$= \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right).$$

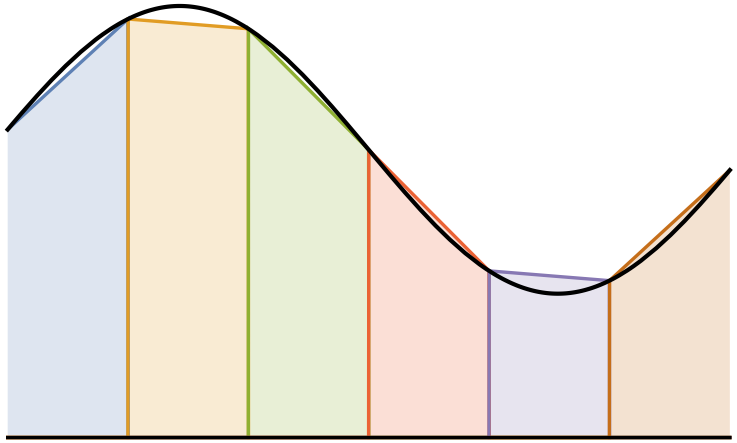
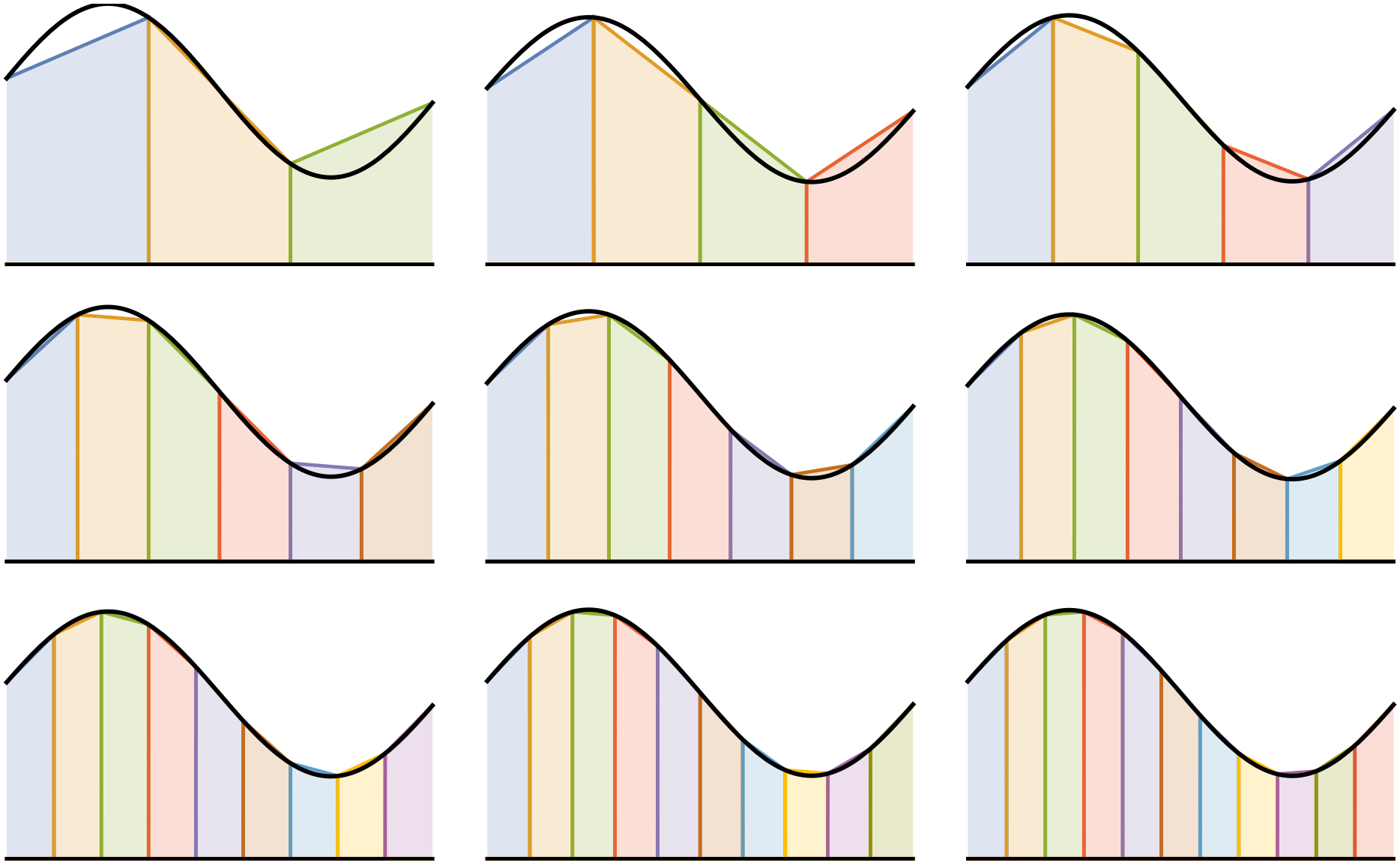


Figure: The trapezoidal rule can be seen as piecewise linear interpolation between the nodes $(x_{i-1}, f(x_{i-1}))$ and $(x_i, f(x_i))$.



Simpson's rule

Let $a = x_0 < x_1 < \cdots < x_n = b$ be an equispaced partition of the interval $[a, b]$ with $2|n$. The signed area trapped by each quadratic is given by

$$Q_i = \frac{h}{3}(f(x_{i-1}) + 4f(x_i) + f(x_{i+1})).$$

We thus obtain the approximation

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x \\ \approx \frac{h}{3}\big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) \\ + \cdots + 4f(x_{n-1}) + f(x_n)\big).\end{aligned}$$

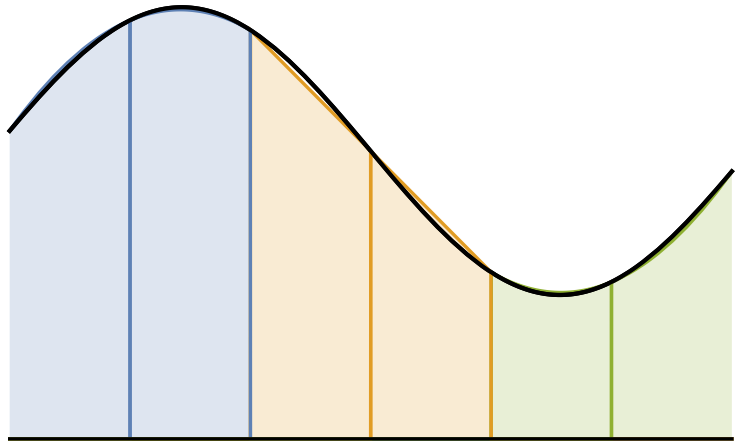
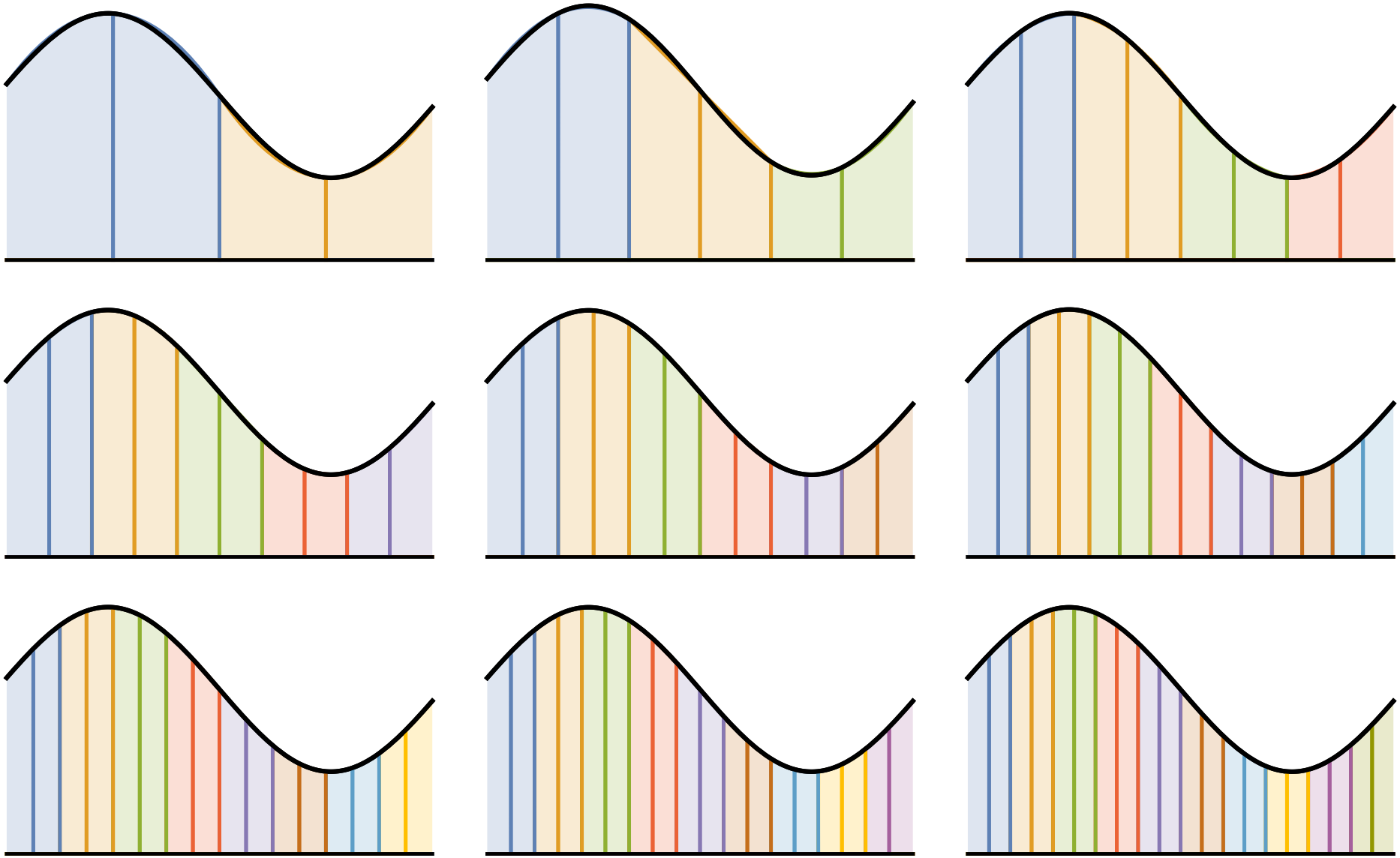


Figure: Simpson's rule can be seen as piecewise quadratic interpolation between the nodes $(x_{i-1}, f(x_{i-1}))$, $(x_i, f(x_i))$, and $(x_{i+1}, f(x_{i+1}))$ for $i \in \{1, 3, 5, \ldots\}$.



Simpson's $\frac{3}{8}$ rule

Let $a = x_0 < x_1 < \cdots < x_n = b$ be an equispaced partition of the interval $[a, b]$ with $3|n$. The signed area trapped by each cubic is given by

$$C_i = \frac{3h}{8}(f(x_{i-1}) + 3f(x_i) + 3f(x_{i+1}) + f(x_{i+2})).$$

We thus obtain the approximation

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \frac{3h}{8}\big(f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) \\ &\quad + 3f(x_4) + 3f(x_5) + 2f(x_6) \\ &\quad + \cdots + f(x_n)\big).\end{aligned}$$

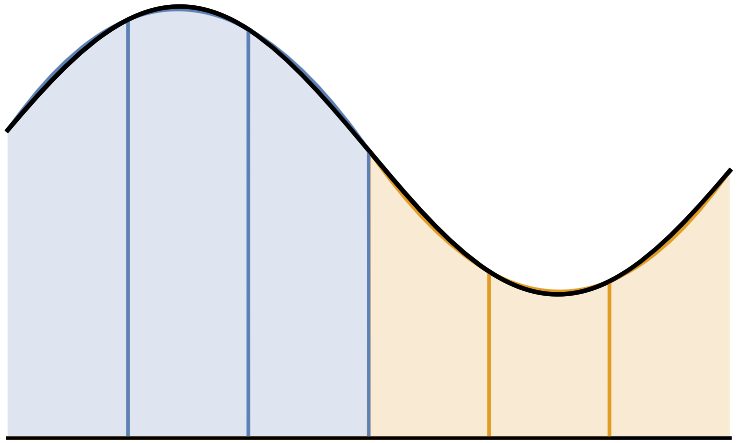
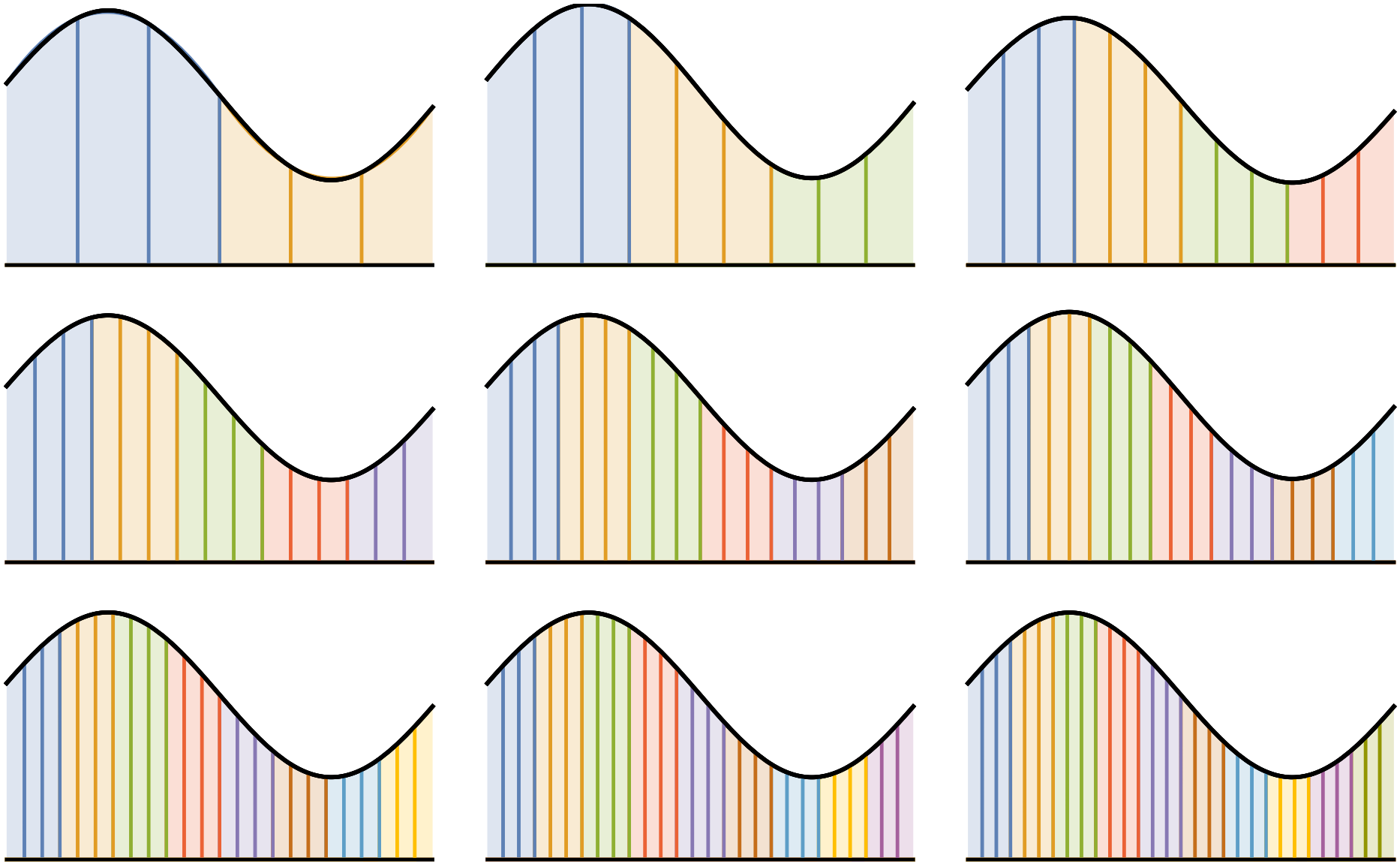


Figure: Simpson's $\frac{3}{8}$ rule can be seen as piecewise cubic interpolation between the nodes $(x_{i-1}, f(x_{i-1}))$, $(x_i, f(x_i))$, $(x_{i+1}, f(x_{i+1}))$, and $(x_{i+2}, f(x_{i+2}))$ for $i \in \{1, 4, 7, \ldots\}$.



The rule constructed by using piecewise quartic interpolation is called Boole's rule.

General construction of Newton–Cotes rules: Let $a = x_1 < x_2 < \cdots < x_n = b$ be a partition of the interval $[a, b]$. Let L_{n-1} be the unique Lagrange interpolating polynomial of f such that $L_{n-1}(x_i) = f(x_i)$ for all $i \in \{1, \ldots, n\}$. Then the area trapped by L_{n-1} can be used to approximate the integral of f over $[a, b]$:

$$\int_a^b f(x)\,\mathrm{d}x \approx \int_a^b L_{n-1}(x)\,\mathrm{d}x = \int_a^b \sum_{i=1}^{n} f(x_i)\ell_i(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i f(x_i),$$

where the *quadrature weights* are given by

$$w_i = \int_a^b \ell_i(x)\,\mathrm{d}x.$$

As before, the interval $[a, b]$ can be divided into multiple subpartitions containing n evaluation points each.

In general, if your integrand is smooth, then using a Newton–Cotes formula of higher order yields better accuracy.

Error formulae for the integration rules can be deduced using, e.g., the error formulae of the Lagrange interpolating polynomial or the Taylor expansion. For example, for the trapezoidal rule the error is

$$-\frac{(b-a)h^2}{12}f''(\xi), \quad \text{where } \xi \in [a,b],$$

and for Simpson's rule

$$-\frac{(b-a)h^4}{180}f^{(4)}(\xi), \quad \text{where } \xi \in [a,b],$$

provided that f is twice or four times continuously differentiable, respectively.

Further treatment of Newton–Cotes formulae can be found, e.g., in [Davis], [NIST], [Abramowitz and Stegun], or [Wikipedia].

Gaussian quadratures

Gaussian quadratures are numerical integration rules that have *the highest possible degree of polynomial exactness* for an integration rule consisting of n nodes.

We shall inspect Gaussian quadrature rules to evaluate integrals of the form[2]

$$\int_{\mathcal{I}} W(x) f(x) \, \mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i),$$

where $\mathcal{I} \subset \mathbb{R}$ is an interval (not necessarily finite!) and $W : \mathcal{I} \to \mathbb{R}_+$ is an a priori known weight function with finite moments (we'll define this in a second).

Here we **don't** have control over the choice of the quadrature nodes $a < x_1 < x_2 < \cdots < x_n < b$. They must be chosen as the roots of an appropriate orthogonal polynomial in order to ensure optimality of the quadrature rule. (The weights are deduced deterministically afterwards!)

[2] We omit the generalization to integrals $\int_{\mathbb{R}} f(x) \, \mathrm{d}\lambda(x)$, see, e.g., [Gautschi].

We're interested in finding approximations to the integrals

$$\int_{\mathcal{I}} W(x)f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i).$$

Here, the function W is assumed to be known a priori and it has the following properties.

Definition

Let $\mathcal{I} \subset \mathbb{R}$ be an interval (possibly infinite). We call a mapping $W : \mathcal{I} \to \mathbb{R}$ a *weight function* if it has the following properties:

(i) $W(x) \geq 0$ for a.e. $x \in \mathcal{I}$,

(ii) $\displaystyle\int_{\mathcal{I}} W(x)x^k\,\mathrm{d}x < \infty$ for all $k \in \mathbb{N}$.

We first need to define the relationship between the weight function W and the nodes and weights $((x_i, w_i))_{i=1}^n$ of the corresponding quadrature rule. To this end, we invoke the theory of orthogonal polynomials.

Let Π_n denote the space of all real polynomials with degree $\leq n$. We can endow Π_n with the inner product $\langle \cdot, \cdot \rangle_W : \Pi_n \times \Pi_n \to \mathbb{R}$ defined by setting

$$\langle p, q \rangle_W := \int_{\mathcal{I}} W(x) p(x) q(x) \, \mathrm{d}x, \quad p, q \in \Pi_n.$$

A sequence $(p_k)_{k=0}^{\infty}$ of polynomials is called *orthogonal* with respect to $\langle \cdot, \cdot \rangle_W$ if

$$\langle p_i, p_j \rangle_W = 0 \quad \text{for all } i \neq j.$$

Note that $\dim \Pi_n = n + 1$ since each $p \in \Pi_n$ may be written uniquely as

$$p(x) = \sum_{i=0}^{n} c_i x^i, \quad c_i \in \mathbb{R}.$$

The sequence $(1, x, x^2, \ldots, x^n)$ is called the *monomial basis* of Π_n.

Now to obtain an orthogonal basis for Π_n with respect to the inner product $\langle \cdot, \cdot \rangle_W$, we may use Gram–Schmidt orthogonalization.

Example

Let $\mathcal{I} = [-1, 1]$ and $W(x) = 1$. By carrying out the Gram–Schmidt procedure on the monomial basis $(1, x, x^2, \ldots, x^n)$ with respect to $\langle \cdot, \cdot \rangle_W$, we obtain

$$p_0(x) = 1,$$

$$p_1(x) = x - \frac{\langle p_0, x \rangle_W}{\langle p_0, p_0 \rangle_W} p_0(x) = x - \frac{\int_{-1}^{1} x \, \mathrm{d}x}{\int_{-1}^{1} \mathrm{d}x} \cdot 1 = x,$$

$$p_2(x) = x^2 - \frac{\langle p_0, x^2 \rangle_W}{\langle p_0, p_0 \rangle_W} p_0 - \frac{\langle p_1, x^2 \rangle_W}{\langle p_1, p_1 \rangle_W} p_1 = x^2 - \frac{\int_{-1}^{1} x^2 \, \mathrm{d}x}{\int_{-1}^{1} \mathrm{d}x} - \frac{\int_{-1}^{1} x^3 \, \mathrm{d}x}{\int_{-1}^{1} x^2 \, \mathrm{d}x} x$$

$$= x^2 - \tfrac{1}{3},$$

$$p_3(x) = x^3 - \tfrac{3}{5}x, \quad p_4(x) = x^4 - \tfrac{6}{7}x^2 + \tfrac{3}{35}, \ldots$$

This is called the family of (monic) *Legendre polynomials*.

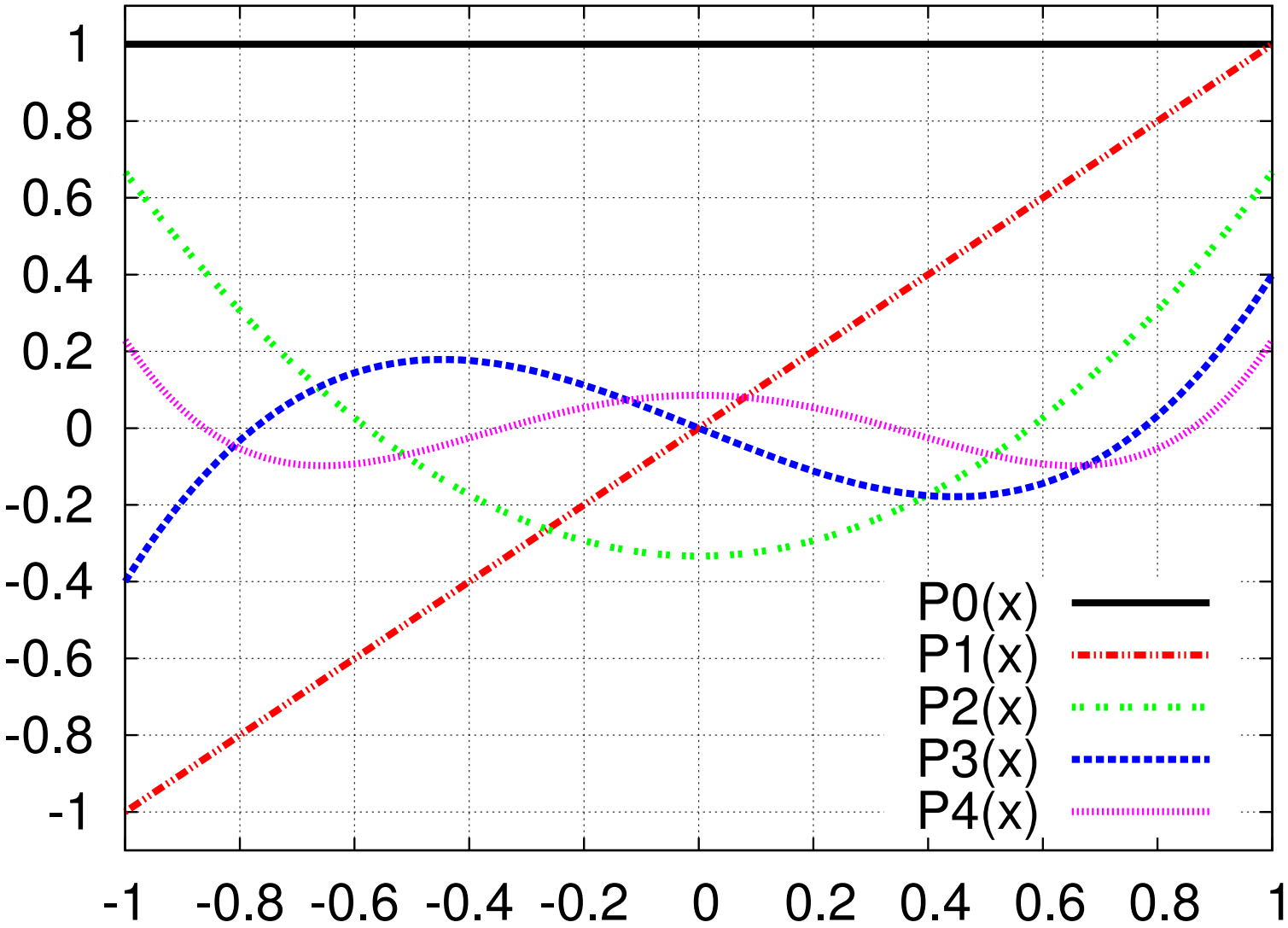


Figure: The monic Legendre polynomials P_0, P_1, P_2, P_3, and P_4.

Different weight functions and intervals correspond to different families of orthogonal polynomials.

orthogonal polynomials	weight function	interval
Legendre polynomials P_k	$W(x) = 1$	$\mathcal{I} = [-1, 1]$
Chebyshev polynomials T_k	$W(x) = (1 - x^2)^{-1/2}$	$\mathcal{I} = (-1, 1)$
Laguerre polynomials L_k	$W(x) = \mathrm{e}^{-x}$	$\mathcal{I} = [0, \infty)$
Hermite polynomials H_k	$W(x) = \mathrm{e}^{-x^2}$	$\mathcal{I} = (-\infty, \infty)$

Gaussian quadratures are used to evaluate integrals of the form

$$\int_{\mathcal{I}} W(x) f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i).$$

Let $(p_k)_{k=0}^{\infty}$ be a family of orthogonal polynomials with respect to $\langle \cdot, \cdot \rangle_W$. It turns out that the highest possible degree of polynomial exactness is obtained by taking the roots of p_n as the quadratures nodes $\{x_i\}_{i=1}^{n}$.

Remark. It can be shown that any orthogonal polynomial p_n w.r.t. $\langle \cdot, \cdot \rangle_W$ has n real, distinct roots that lie in the interval $\mathcal{I}$. We omit the proof of this statement, but this result can be deduced, e.g., as an application of the three-term recurrence relation (cf. Appendix).

Theorem

$$\int_{\mathcal{I}} W(x) f(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i f(x_i) \quad \textit{for all } f \in \Pi_{2n-1},$$

where $x_1 < \cdots < x_n$ *are the roots of* p_n *and the weights are*

$$w_i = \int_{\mathcal{I}} W(x)\ell_i(x)\,\mathrm{d}x \quad \textit{for } i \in \{1, \ldots, n\},$$

where

$$\ell_j(x) = \prod_{\substack{1 \le i \le n \\ i \neq j}} \frac{x - x_i}{x_j - x_i} \quad \textit{for } j \in \{1, \ldots, n\}.$$

Proof. Recall that the Lagrange interpolation polynomial of $f \in \Pi_{2n-1}$ at nodes $x_1 < \cdots < x_n$ is given by

$$L_{n-1}(x) = \sum_{i=1}^{n} f(x_i)\ell_i(x),$$

where

$$\ell_j(x) = \prod_{\substack{1 \le i \le n \\ i \neq j}} \frac{x - x_i}{x_j - x_i} \quad \text{for } j \in \{1, \ldots, n\}$$

satisfies $\ell_i(x_j) = \delta_{i,j}$. Hence $L_{n-1}(x_i) = f(x_i)$ and $L_{n-1} \in \Pi_{n-1}$.

In consequence, each root of $p_n \in \Pi_n$ is also a root of $f - L_{n-1} \in \Pi_{2n-1}$. These polynomials are thus divisible!

$$\frac{f - L_{n-1}}{p_n} = q \quad \text{for some } q \in \Pi_{n-1}.$$

We write this as $f(x) = L_{n-1}(x) + p_n(x)q(x)$.

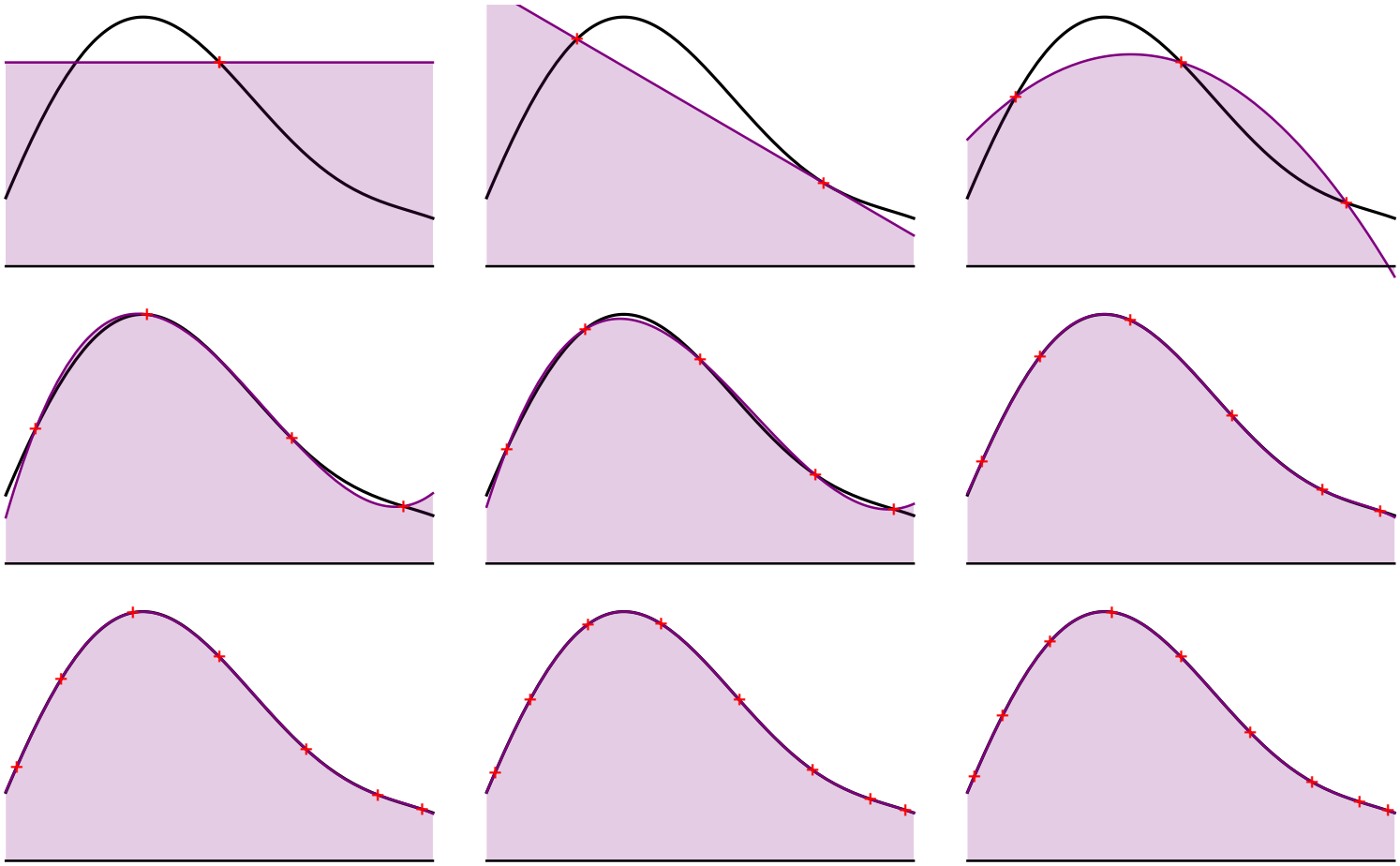
On the other hand, $(p_k)_{k=0}^{n-1}$ form a basis for Π_{n-1}. We can express q in this basis as

$$q(x) = \sum_{i=0}^{n-1} c_i p_i(x) \quad \text{for some } c_i \in \mathbb{R}.$$

Hence

$$\int_{\mathcal{I}} W(x)f(x)\,\mathrm{d}x = \int_{\mathcal{I}} W(x)L_{n-1}(x)\,\mathrm{d}x + \overbrace{\int_{\mathcal{I}} W(x)p_n(x)q(x)\,\mathrm{d}x}^{=0 \text{ since } \langle p_n, q\rangle_W = 0}$$

$$= \int_{\mathcal{I}} W(x)L_{n-1}(x)\,\mathrm{d}x = \sum_{i=1}^{n} f(x_i) \underbrace{\int_{\mathcal{I}} W(x)\ell_i(x)\,\mathrm{d}x}_{=w_i}. \quad \square$$



Gaussian quadrature of a function f in $[-1, 1]$. For the n-point rule, the quadrature nodes $(x_i)_{i=1}^n$ are taken as the roots of the Legendre polynomial P_n. The approximation of the integral is the signed area trapped by the Lagrange interpolating polynomial of f, when interpolation is carried out with respect to the Legendre roots $x_1 < \cdots < x_n$.

Efficient generation of Gaussian quadrature rules

Three-term recurrence of orthogonal polynomials

Gram–Schmidt orthogonalization is an ineffective way to generate orthogonal polynomials. It turns out that there exists a *three-term recurrence relation*

$$\begin{aligned} p_0(x) &= 1, \\ p_1(x) &= (x - \alpha_1)p_0(x), \\ p_{k+1}(x) &= (x - \alpha_{k+1})p_k(x) - \beta_{k+1}p_{k-1}(x), \end{aligned}$$

where

$$\alpha_{k+1} = \frac{\langle xp_k, p_k \rangle_W}{\langle p_k, p_k \rangle_W} \quad \text{and} \quad \beta_{k+1} = \frac{\langle p_k, p_k \rangle_W}{\langle p_{k-1}, p_{k-1} \rangle_W}.$$

Example (Legendre polynomials)

$$\alpha_k = 0 \ \forall k \quad \text{and} \quad \beta_k = \frac{(k-1)^2}{4k^2 - 8k + 3}, \quad k \geq 2.$$

Example (Hermite polynomials)

$$\alpha_k = 0 \ \forall k \quad \text{and} \quad \beta_k = \frac{k-1}{2}, \quad k \geq 2.$$

Similar formulae exist for most known families of orthogonal polynomials.

Let's assume that the family of orthogonal polynomials $(p_k)_{k=0}^{\infty}$ with respect to $\langle \cdot, \cdot \rangle_W$ is characterized by the three-term recurrence coefficients $(\alpha_k)_{k=1}^{\infty}$ and $(\beta_k)_{k=2}^{\infty}$.

The following algorithm for the generation of Gaussian quadratures was introduced in [Golub and Welsch].

Algorithm (Golub–Welsch)

(i) *Construct the tridiagonal $n \times n$ matrix*

$$A = \begin{bmatrix} \alpha_1 & \sqrt{\beta_2} & & & \\ \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & & \\ & \sqrt{\beta_3} & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_n} \\ & & & \sqrt{\beta_n} & \alpha_n \end{bmatrix}.$$

(ii) **Fact:** *The eigenvalues $x_1, \ldots, x_n$ of A are precisely the roots of p_n.*

(iii) **Fact:** *Let $\mathbf{q}_j = [q_{1,j}, \ldots, q_{n,j}]^{\mathrm{T}}$ be the normalized eigenvector corresponding to eigenvalue x_j. Then $w_j = q_{1,j}^2 \int_{\mathcal{I}} W(x)\,\mathrm{d}x$.*

(iv) *Compute the Gaussian quadrature*

$$\int_{\mathcal{I}} W(x) f(x)\,\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i).$$

Multidimensional integration

Multidimensional integration over hypercubes $[a, b]^d$

Let $((w_i, x_i))_{i=1}^n$ be the weights and nodes of your favorite univariate quadrature rule:

$$\int_a^b f(x)W(x)\,\mathrm{d}x \approx \sum_{i=1}^n w_i f(x_i).$$

Let $g : [a, b]^d \to \mathbb{R}$ be a d-variate function. Then you can integrate over the hypercube $[a, b]^d$ by composing your favorite quadrature rule over all axes:

$$\int_{[a,b]^d} g(x_1, \ldots, x_d)W(x_1)\cdots W(x_d)\,\mathrm{d}x_1\cdots\mathrm{d}x_d$$

$$\approx \sum_{i_1=1}^n \cdots \sum_{i_d=1}^n w_{i_1}\cdots w_{i_d} g(x_{i_1}, \ldots, x_{i_d}).$$

In fact, you can choose separate quadrature rules corresponding to different intervals and weights for all axes! Cost: n^d function evaluations!

Multidimensional Monte Carlo integration

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. We wish to integrate

$$\int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

Idea: $\mathrm{d}\mathbf{x}/\mathrm{vol}(\Omega)$ is the probability density of the uniform distribution; hence

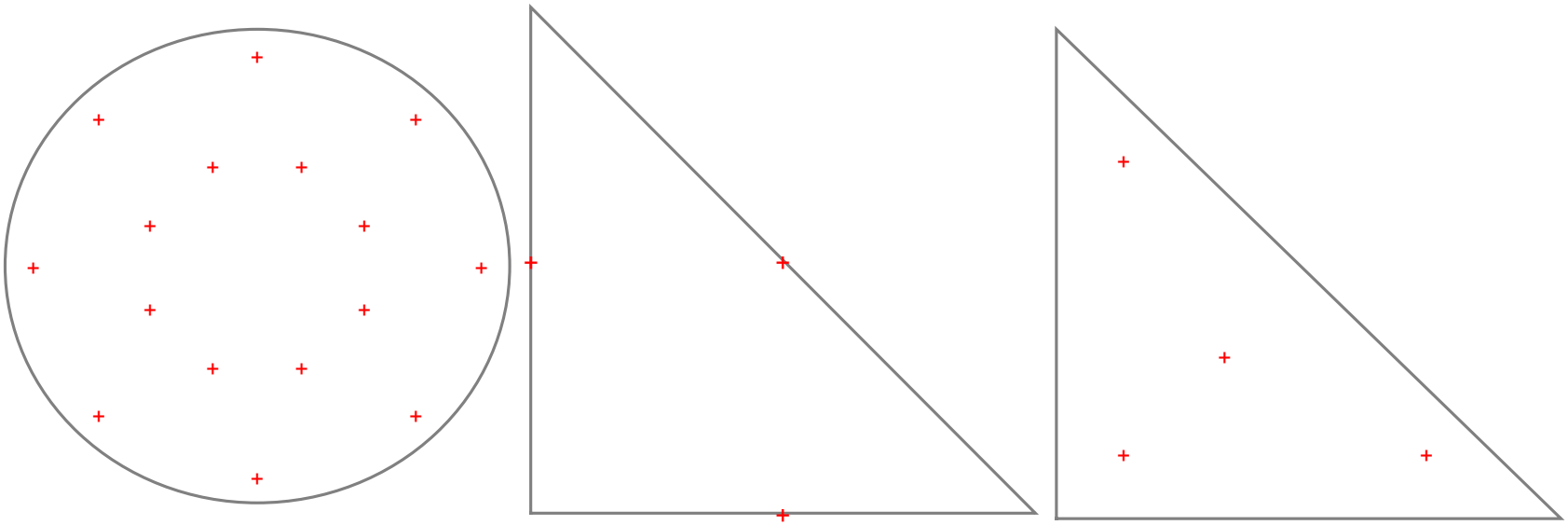
$$\int_\Omega f(\mathbf{x})\,\mathrm{d}\mathbf{x} \approx \frac{\mathrm{vol}(\Omega)}{N}\sum_{i=1}^{N} f(\mathbf{x}_i), \quad \mathrm{vol}(\Omega) = \int_\Omega \mathrm{d}\mathbf{x},$$

where $(\mathbf{x}_i)_{i=1}^N$ is a random sample of points in Ω.

Convergence rate: $\mathcal{O}(1/\sqrt{N})$ according to the Central Limit Theorem.
Independent of dimension d, but extremely slow nonetheless!

Integration rules for canonical domains

For many canonical domains, there exist various integration formulae in the literature.



Optimal Gaussian cubatures, however, remain elusive in the case $d > 1$ and have been a popular topic of research in the discipline of algebraic geometry.

Bibliography

On polynomial interpolation and Newton–Cotes formulae:


P.J. Davis. *Interpolation and Approximation*. Dover, New York, 1976.

On orthogonal polynomials and Gaussian quadratures:


W. Gautschi. *Orthogonal polynomials: Computation and Approximation.* Oxford University Press, 2004.


J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer New York, 2002.


H. Wilf. *Mathematics for the Physical Sciences*. Wiley, 1962.


F.B. Hildebrand. *Introduction to Numerical Analysis*. McGraw–Hill, 1974.


G.H. Golub and J.H. Welsch. Calculation of Gauss Quadrature Rules. *Math. Comp.* **23**(106):221–230, 1969. DOI:10.1090/S0025-5718-69-99647-1

Appendix

Some consequences of the three-term recurrence

The three-term recurrence on page 35 can be recast in matrix form as

$$x\boldsymbol{\psi}(x) = S\boldsymbol{\psi}(x) + p_n(x)\mathbf{e}_n, \tag{1}$$

where $\boldsymbol{\psi}(x) = [p_0(x), \ldots, p_{n-1}(x)]^{\mathrm{T}}$, $\mathbf{e}_n = [0, 0, \ldots, 0, 1]^{\mathrm{T}}$, and

$$S = \begin{bmatrix} \alpha_1 & 1 & & & \\ \beta_2 & \alpha_2 & 1 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & \beta_n & \alpha_n \end{bmatrix}.$$

- What happens when you choose x to be equal to one of the roots of p_n in (1)?
- Let $D = \operatorname{diag}(\sqrt{\beta_2}, 1, \frac{1}{\sqrt{\beta_3}}, \frac{1}{\sqrt{\beta_3\beta_4}}, \cdots, \prod_{k=3}^{n} \frac{1}{\sqrt{\beta_k}})$. Observe that S and DSD^{-1} have the same eigenvalues ([similarity transformation]) and, moreover, that DSD^{-1} is equal to matrix A on page 38.
- Conclude that the eigenvalues of matrix A in the Golub–Welsch algorithm coincide with the roots of p_n. In addition, due to A being symmetric, the eigenvalues of orthogonal polynomials must be real!