Numerical Calculus: Numerical Integration

Pătrulescu Flavius

Technical University of Cluj-Napoca

2025

References

1. R.L. Burden, D.J. Faires, A.M. Burden, *Numerical Analysis*, 10th Ed., Cengage Learning, Boston, 2022.
2. G. Dahlquist, A. Björk, *Numerical Methods in Scientific Computing*, Vol. I, SIAM, Philadelphia, 2008.
3. A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, Springer-Verlag, New-York, 2000.
4. E. Süli, D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, Cambridge, 2003.

Heaviside unit step function (Dahlquist and Björk, 2008)

$$t_+ = \max\{t, 0\}, \quad t_+^k = (t_+)^k, \quad sign(t) = \begin{cases} 1, \ t > 0 \\ 0, \ t = 0 \\ -1, \ t < 0 \end{cases}$$

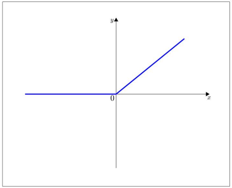
The function t_+^0 is often denoted $H(t)$ and is known as the *Heaviside unit step function*

$$H(t) = t_+^0 = \frac{1 + sign(t)}{2} = \begin{cases} 1, \ t > 0 \\ \frac{1}{2}, \ t = 0 \\ 0, \ t < 0 \end{cases}$$

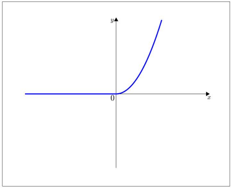
$$t_+^k \in C^{k-1}, \quad k \geqslant 1$$

Functions

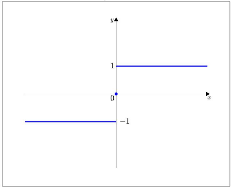
Function t_+



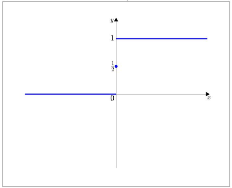
Function t_+^2



Sign function



Heaviside unit step function



Peano's Remainder Theorem (Dahlquist and Björk, 2008)

Let $\mathcal{R} : C^n[a,b] \to \mathbb{R}$ a linear functional with $\mathcal{R}(P) = 0$ for all $P \in \Pi_{n-1}$ $\left(ker(\mathcal{R}) = \Pi_{n-1}\right)$. Then, for all $f \in C^n[a,b]$

$$\mathcal{R}(f) = \int_a^b f^{(n)}(u) K(u)\, du,$$

where K represents the *Peano's kernel* of the functional $\mathcal{R}$

$$K(u) = \frac{1}{(n-1)!} \mathcal{R}_x((x-u)_+^{n-1}),\ u \in [a,b].$$

The subscript in $\mathcal{R}_x$ indicates that $\mathcal{R}$ acts on the variable x (not on the variable u).

Peano's Remainder Theorem (Dahlquist and Björk, 2008)

The function $K(u)$ vanishes outside $[a, b]$ because

- if $u > b$ then $u > x$; hence $(x-u)_+^{n-1} = 0$ and $K(u) = 0$
- if $u < a$, then $x > u$. It follows $(x-u)_+^{n-1} = (x-u)^{n-1} \in \Pi_{n-1}$; hence $K(u) = 0$ by $ker(\mathcal{R}) = \Pi_{n-1}$

If $K(u)$ does not change the sign then $\exists\, \xi \in [a, b]$ s.t.

$$R(f) = \frac{f^{(n)}(\xi)}{n!} R(x^n).$$

Numerical Integration (Burden *et al.*, 2022)

The basic method involved in approximating $\int_a^b f(x)\,dx$ is called *numerical quadrature*.

$$\int_a^b f(x)\,dx = \underbrace{\sum_{k=0}^{n} A_k f(x_k) + R(f)}_{I[f]:=}$$

$x_0, \ldots, x_n$ - nodes

$A_0, \ldots, A_n$ - coefficients/weights

$R(f)$ - error term / rest term

Weighted quadrature rule (Dahlquist and Björk, 2008)

If the integrand has a singularity, for example, it becomes infinite at some point in or near the interval of integration, some modification is necessary. Another complication arises when the interval of integration is infinite. In both cases it may be advantageous to consider a weighted quadrature rule:

$$\int_a^b w(x) f(x)\, dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

where $w(x) \geqslant 0$ is a *weight function or density function* (incorporates the singularity so that $f(x)$ can be well approximated by a polynomial).

Numerical Integration (Dahlquist and Bjork,2008)

The *kth moment with respect to w* is given by

$$\mu_k = \int_a^b x^k w(x)\, dx < \infty.$$

To ensure that the integral is well defined when $f(x)$ is a polynomial, we assume in the following that the moments μ_k are defined $\forall\, k \geqslant 0$ and $\mu_0 > 0$.
Often (but not always) all nodes lie in $[a, b]$. The weights A_i are usually determined so that the quadrature formula is exact for polynomials of as high degree as possible.

Degree of accuracy (Burden *et al.*, 2022)

The *degree of accuracy, or precision*, of a quadrature formula is the largest positive integer r such that the formula is exact for x^j, for each $j = 0, 1, \dots, r$, *i.e.*

$$\int_a^b x^j \, dx = \sum_{k=0}^{n} A_k x_k^j \quad (R(x^j) = 0).$$

The degree of precision of a quadrature formula is r if and only if the error is zero for all polynomials of degree $j = 0, 1, \dots, r$, but is not zero for some polynomial of degree $r + 1$

$$R(x^j) = 0, \; \forall j = \overline{0, r}, R(x^{r+1}) \neq 0$$

$$\Rightarrow R(P) = 0, \forall \, P \in \Pi_r, \exists \, Q \in \Pi_{r+1} \text{ s.t. } R(Q) \neq 0.$$

Composite Formulae (Süli and Mayers, 2003)

A better approach to improving accuracy is to divide the interval $[a, b]$ into an increasing number of subintervals of decreasing size and then to use a numerical integration formula of fixed order n on each of the subintervals. The quadrature rules based on this approach are called *composite formulae*.

$$\int_a^b f(x)\,dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,dx =$$

$$= \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \ldots + \int_{x_{n-1}}^{x_n} f(x)\,dx$$

where

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b.$$

Numerical Integration (Burden *et al.*, 2022)

Quadrature methods using interpolation polynomials: the basic idea is to select a set of distinct nodes $\{x_0, x_1, \ldots, x_{n-1}, x_n\}$ from the interval $[a, b]$. Then integrate the Lagrange interpolating polynomial $L_n(x) = \sum_{k=0}^{n} f(x_k) l_k(x)$ and its truncation error term over $[a, b]$ to obtain

$$
\begin{aligned}
&\int_a^b f(x)\,dx = \\
&= \int_a^b \sum_{k=0}^{n} f(x_k) l_k(x)\,dx + \int_a^b \prod_{k=0}^{n} (x - x_k) \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\,dx = \\
&= \sum_{k=0}^{n} f(x_k) \underbrace{\int_a^b l_k(x)\,dx}_{A_k :=} + \frac{1}{(n+1)!} \int_a^b \prod_{k=0}^{n} (x - x_k) f^{(n+1)}(\xi(x))\,dx,
\end{aligned}
$$

where $\xi(x) \in [a, b]$.

Numerical Integration (Burden *et al.*, 2022)

$$\Rightarrow \int_a^b f(x)\,dx = \sum_{k=0}^{n} A_k f(x_k) +$$

$$+\frac{1}{(n+1)!}\int_a^b \prod_{k=0}^{n}(x - x_k) f^{(n+1)}(\xi(x))\,dx$$

The quadrature formula is, therefore,

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} A_k f(x_k),\; A_k = \int_a^b l_k(x)\,dx$$

with error term given by

$$R(f) = \frac{1}{(n+1)!}\int_a^b \prod_{k=0}^{n}(x - x_k) f^{(n+1)}(\xi(x))\,dx$$

The Trapezoidal Rule (Burden *et al.*, 2022)

To derive the trapezoidal rule, we use linear Lagrange polynomial. Let $x_0 = a$, $x_1 = b$, $h = b - a$.

$$f(x) = L_1(x) + \frac{f''(\xi(x))}{2!}(x-a)(x-b)$$

$$L_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) = \frac{x - b}{a - b} f(a) + \frac{x - a}{b - a} f(b)$$

$$\int_a^b f(x)\,dx = \int_a^b L_1(x)\,dx + \int_a^b \frac{f''(\xi(x))}{2!}(x-a)(x-b)\,dx$$

$$= \frac{b-a}{2}(f(a) + f(b)) + \frac{1}{2!}\int_a^b f''(\xi(x))(x-a)(x-b)\,dx$$

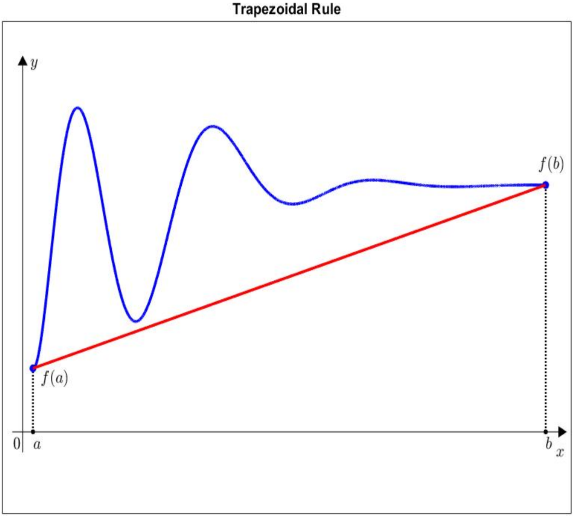
The Trapezoidal Rule (Burden *et al.*, 2022)

The product $(x - a)(x - b)$ does not change sign in $[a, b]$, so the Weighted Mean Value Theorem for integrals can be applied to the error term. For some $\xi \in (a, b)$,

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\Big(f(a) + f(b)\Big) - \frac{(b-a)^3}{12} f''(\xi)$$

The Trapezoidal Rule provides an error term which involves f'', so the method gives the exact result when applied to any function whose second derivative is identically zero, that is, any polynomial of degree one or less.

Trapezoidal Rule



Composite Trapezoidal Rule (Dahlquist and Björk, 2008)

To increase the accuracy we subdivide the interval $[a, b]$ and assume that $f(x_k)$ are known on a grid of equidistant points

$$x_k = a + kh,\ k = \overline{0, n},\ h = \frac{b-a}{n}$$

$$\int_a^b f(x)\,dx = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x)\,dx \approx \frac{h}{2}\left(f(x_0) + 2\sum_{k=1}^{n-1} f(x_k) + f(x_n)\right)$$

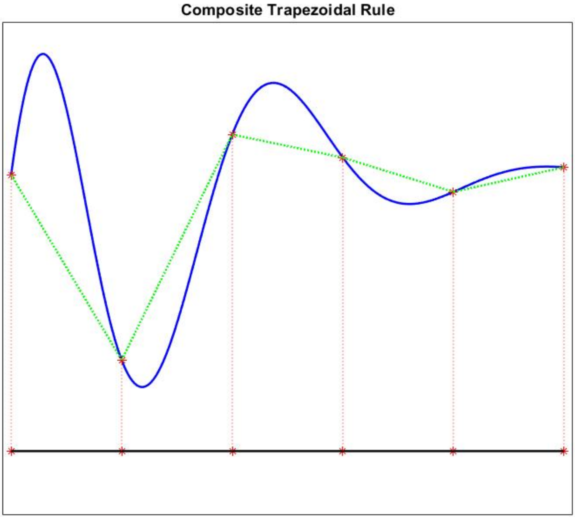
Concerning the global truncation error, using discrete mean-value theorem we deduce that there exists $\xi \in [a, b]$ s.t.

$$\int_a^b f(x)\,dx = \frac{h}{2}\left(f(x_0) + 2\sum_{k=1}^{n-1} f(x_k) + f(x_n)\right) - \frac{(b-a)h^2}{12} f^{(2)}(\xi) =$$

$$= \frac{b-a}{2n}\left(f(x_0) + 2\sum_{k=1}^{n-1} f(x_k) + f(x_n)\right) - \frac{(b-a)^3}{12n^2} f^{(2)}(\xi)$$

This shows that by choosing h small enough we can make the truncation error arbitrarily small. In other words, we have asymptotic convergence when $h \to 0$ ($n \to \infty$).

Composite Trapezoidal Rule



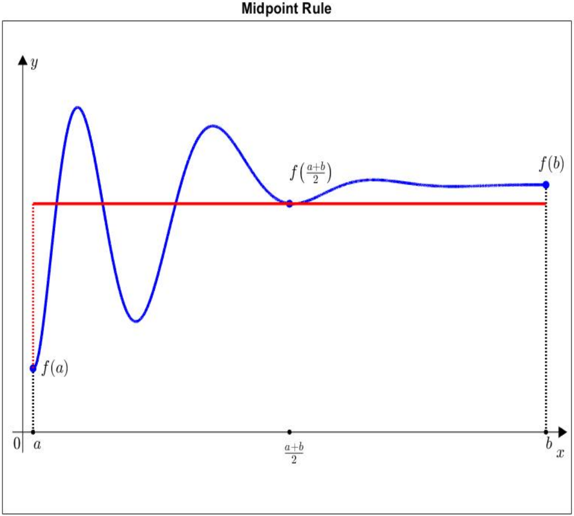
Midpoint Formula (Quarteroni *et al.*, 2000)

This formula is obtained by replacing f over $[a, b]$ with a constant function equal to the value achieved by f at the midpoint of $[a, b]$. For some $\xi \in (a, b)$ we have

$$\int_a^b f(x)\,dx = (b-a)f\left(\frac{a+b}{2}\right) + \frac{(b-a)^3}{24} f''(\xi)$$

The midpoint rule has degree of exactness equal to 1 (is exact for constant and affine functions, since in both cases $f''(\xi) = 0$ for any $\xi \in (a, b)$).

Midpoint Rule



Composite Midpoint Rule (Dahlquist and Björk, 2008)

Let the grid of equidistant points

$$x_k = a + kh,\ k = \overline{0, n},\ h = \frac{b-a}{n}$$

and

$$x_{k+1/2} = \frac{x_k + x_{k+1}}{2} = a + \frac{2k+1}{2}h,\ k = \overline{0, n-1}$$

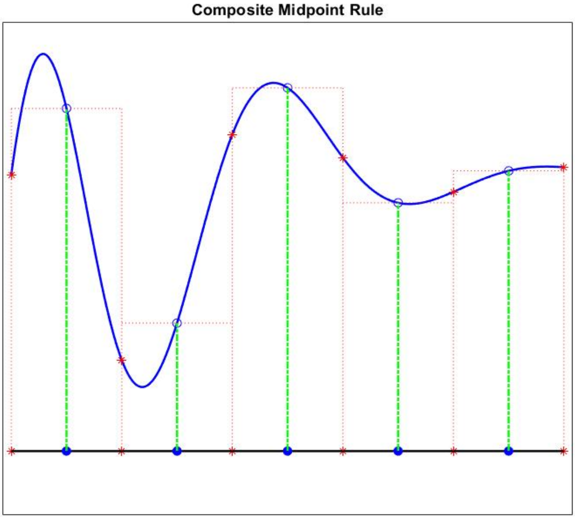
$$\int_a^b f(x)\,dx = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x)\,dx \approx h \sum_{k=0}^{n-1} f(x_{k+1/2})$$

Concerning the global truncation error there exists $\xi \in [a, b]$ s.t.

$$\int_a^b f(x)\,dx = h \sum_{k=0}^{n-1} f(x_{k+1/2}) + \frac{(b-a)h^2}{24} f^{(2)}(\xi)$$

$$= \frac{b-a}{n} \sum_{k=0}^{n-1} f(x_{k+1/2}) + \frac{(b-a)^3}{24n^2} f^{(2)}(\xi)$$

Composite Midpoint Rule



Simpson's Rule (Burden *et al.*, 2022)

Simpson's rule results from integrating over $[a, b]$ the quadratic Lagrange polynomial with equally spaced nodes
$x_0 = a$, $x_1 = a + h = \frac{a+b}{2}$, $x_2 = a + 2h = b$, $h = \frac{b-a}{2}$

$$f(x) = L_2(x) + \frac{f^{(3)}(\xi(x))}{3!}(x-a)(x-\frac{a+b}{2})(x-b)$$

$$L_2(x) = \frac{(x-\frac{a+b}{2})(x-b)}{(a-\frac{a+b}{2})(a-b)}f(a) + \frac{(x-a)(x-b)}{(\frac{a+b}{2}-a)(\frac{a+b}{2}-b)}f(\frac{a+b}{2}) + \frac{(x-a)(x-\frac{a+b}{2})}{(b-a)(b-\frac{a+b}{2})}f(b)$$

$$\int_a^b f(x)\,dx = \int_a^b L_2(x)\,dx +$$

$$+\int_a^b \frac{f^{(3)}(\xi(x))}{3!}(x-a)(x-\frac{a+b}{2})(x-b)\,dx$$

Deriving Simpson's rule in this manner, however, provides only an $O(h^4)$ error term involving $f^{(3)}$.

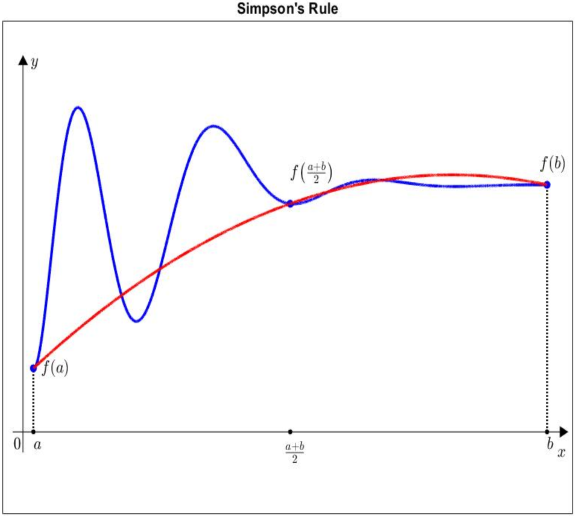
Simpson's Rule (Burden *et al.*, 2022)

Using Peano's Remainder Theorem a higher-order term involving $f^{(4)}$ can be derived. For $\xi \in (a, b)$

$$\int_a^b f(x)\,dx = \frac{b-a}{6}\Big(f(a) + 4f\big(\frac{a+b}{2}\big) + f(b)\Big) - \frac{(b-a)^5}{2880} f^{(4)}(\xi).$$

The error term in Simpson's rule involves the fourth derivative of f, so it gives exact results when applied to any polynomial of degree three or less.

Simpson's Rule



Composite Simpson's Rule (Burden *et al.*, 2022)

To obtain the composite form, we subdivide the interval $[a, b]$ into n subintervals and apply Simpson's rule to each consecutive pair of subintervals. With $h = \frac{b-a}{n}$ and $x_k = a + kh$, $\forall\, k = \overline{0, n}$, we have

$$\int_a^b f(x)\,dx = \sum_{k=1}^{(n/2)} \int_{x_{2k-2}}^{x_{2k}} f(x)\,dx \approx$$

$$\approx \frac{h}{3}\Big(f(x_0) + 2\sum_{k=1}^{(n/2)-1} f(x_{2k}) + 4\sum_{k=1}^{(n/2)} f(x_{2k-1}) + f(x_n) \Big).$$

Composite Simpson's Rule (Burden *et al.*, 2022)

Concerning the global truncation error, there exists $\xi \in [a, b]$ s.t.

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

$$= \frac{h}{3}\Big(f(x_0) + 2\sum_{k=1}^{(n/2)-1} f(x_{2k}) + 4\sum_{k=1}^{(n/2)} f(x_{2k-1}) + f(x_n)\Big) -$$

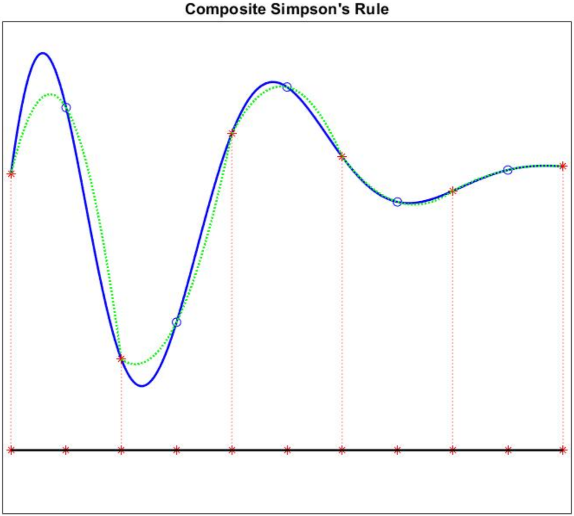
$$-\frac{b-a}{180}h^4 f^{(4)}(\xi) =$$

$$= \frac{b-a}{3n}\Big(f(x_0) + 2\sum_{k=1}^{(n/2)-1} f(x_{2k}) + 4\sum_{k=1}^{(n/2)} f(x_{2k-1}) + f(x_n)\Big) -$$

$$-\frac{(b-a)^5}{180n^4} f^{(4)}(\xi)$$

For the Composite Simpson's rule n is an even integer.

Composite Simpson's Rule



Simpson's 3/8 rule (Burden *et al.*, 2022)

Simpson's 3/8 rule results from integrating over $[a, b]$ the cubic Lagrange polynomial with equally spaced nodes

$$x_0 = a,\ x_1 = a + h = \frac{2a+b}{3},\ x_2 = a + 2h = \frac{a+2b}{3},$$
$$x_3 = a + 3h = b,\ h = \frac{b-a}{3}$$

$$f(x) = L_3(x) + \frac{f^{(4)}(\xi(x))}{4!}(x-a)(x-\frac{2a+b}{3})(x-\frac{a+2b}{3})(x-b)$$

$$L_3(x) = \frac{(x-\frac{2a+b}{3})(x-\frac{a+2b}{3})(x-b)}{(a-\frac{2a+b}{3})(a-\frac{a+2b}{3})(a-b)}f(a) +$$

$$+\frac{(x-a)(x-\frac{a+2b}{3})(x-b)}{(\frac{2a+b}{3}-a)(\frac{2a+b}{3}-\frac{a+2b}{3})(\frac{2a+b}{3}-b)}f(\frac{2a+b}{3}) +$$
$$+\frac{(x-a)(x-\frac{2a+b}{3})(x-b)}{(\frac{a+2b}{3}-a)(\frac{a+2b}{3}-\frac{2a+b}{3})(\frac{a+2b}{3}-b)}f(\frac{a+2b}{3}) +$$

$$+\frac{(x-a)(x-\frac{2a+b}{3})(x-\frac{a+2b}{3})}{(b-a)(b-\frac{2a+b}{3})(b-\frac{a+2b}{3})}f(b)$$

Simpson's 3/8 rule (Burden *et al.*, 2022)

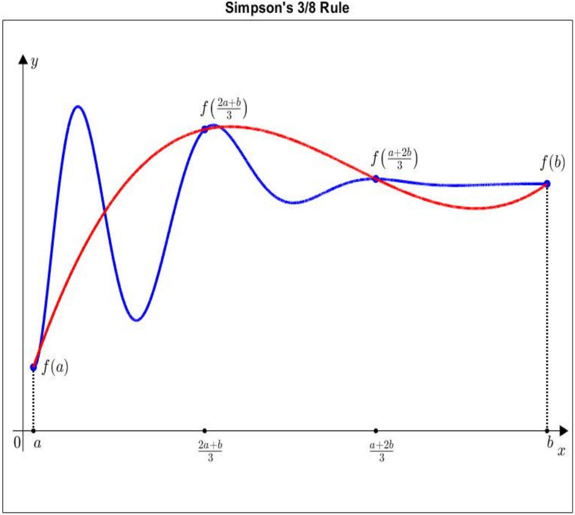
Using Peano's Remainder Theorem, a rest term involving $f^{(4)}$ can be derived. For $\xi \in (a, b)$

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

$$= \frac{b-a}{8}\Big(f(a) + 3f\big(\frac{2a+b}{3}\big) + 3f\big(\frac{a+2b}{3}\big) + f(b)\Big) -$$

$$-\frac{(b-a)^5}{6480} f^{(4)}(\xi).$$

Simpson's 3/8 Rule



Composite Simpson's 3/8 Rule (Burden *et al.*, 2022)

To obtain the composite form, we subdivide the interval $[a, b]$ into n subintervals and apply Simpson's 3/8 rule to each consecutive pair of subintervals. With $h = \frac{b-a}{n}$ and $x_k = a + kh$, $\forall\, k = \overline{0, n}$, we have

$$\int_a^b f(x)\,dx = \sum_{k=1}^{(n/3)} \int_{x_{3k-3}}^{x_{3k}} f(x)\,dx \approx$$

$$\approx \frac{3h}{8}\left(f(x_0) + 3 \sum_{k=1, k\nmid 3}^{n-1} f(x_k) + 2 \sum_{k=1}^{(n/3)-1} f(x_{3k}) + f(x_n) \right)$$

For the Composite Simpson's 3/8 rule n is a multiple of three.

Composite Simpson's 3/8 Rule (Burden *et al.*, 2022)

Concerning the global truncation error, there exists $\xi \in [a, b]$ s.t.

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

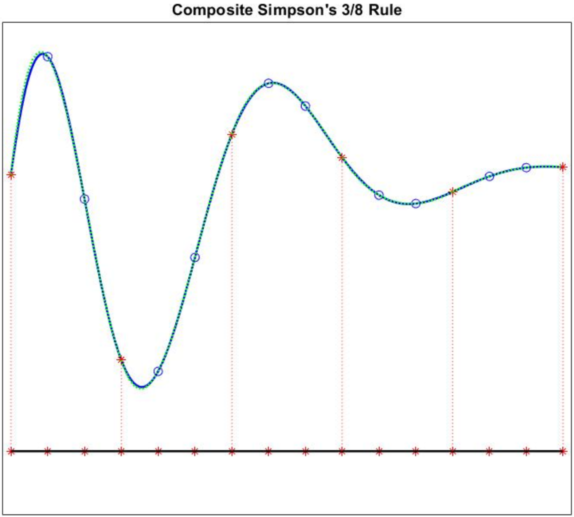
$$= \frac{3h}{8}\left(f(x_0) + 3\sum_{k=1, k\nmid 3}^{n-1} f(x_k) + 2\sum_{k=1}^{(n/3)-1} f(x_{3k}) + f(x_n)\right) -$$

$$-\frac{b-a}{80}h^4 f^{(4)}(\xi)$$

$$= \frac{3(b-a)}{8n}\left(f(x_0) + 3\sum_{k=1, k\nmid 3}^{n-1} f(x_k) + 2\sum_{k=1}^{(n/3)-1} f(x_{3k}) + f(x_n)\right) -$$

$$-\frac{(b-a)^5}{80n^4} f^{(4)}(\xi)$$

Composite Simpson's 3/8 Rule



Newton-Cotes formulas (Dahlquist and Björk, 2008)

The classical *Newton–Cotes quadrature rules*

$$\int_a^b f(x)\,dx = \sum_{k=0}^{n} A_k f(x_k) + R(f)$$

are interpolatory rules obtained for $w \equiv 1$ and equidistant points. There are two classes: *closed formulas*, where the endpoints of the interval belong to the nodes, and *open formulas*, where all nodes lie strictly in the interior of the interval.

Closed Newton-Cotes formulas (Burden *et al.*, 2022)

The $(n+1)$-*point closed Newton-Cotes formula* used the nodes

$$x_k = a + kh,\ k = \overline{0, n},\ h = \frac{b-a}{n}.$$

It is called closed because the endpoints of the closed interval $[a, b]$ are included as nodes ($x_0 = a$, $x_n = b$). The formula assumes the form

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

where

$$A_k = \int_a^b l_k(x)\,dx = \int_a^b \prod_{i=0, i\neq k}^{n} \frac{x - x_i}{x_k - x_i}.$$

The weights satisfy

$$A_k = A_{n-k} \text{ and } \sum_{k=0}^{n} A_k = n.$$

Closed Newton-Cotes formulas (Burden *et al.*, 2022)

There exists $\xi \in (a, b)$ for which

$$\int_a^b f(x)\,dx = \sum_{k=0}^{n} A_k f(x_k) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1)\dots(t-n)\,dt$$

if n is even and $f \in C^{n+2}[a, b]$, and

$$\int_a^b f(x)\,dx = \sum_{k=0}^{n} A_k f(x_k) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1)\dots(t-n)\,dt$$

if n is odd and $f \in C^{n+1}[a, b]$.
Note that when n is an even integer, the degree of precision is $n+1$, although the interpolation polynomial is of degree at most n. When n is odd, the degree of precision is only n.

Closed Newton-Cotes formulas (Burden *et al.*, 2022)

$n = 1$ Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}\Big[f(x_0) + f(x_1)\Big] - \frac{h^3}{12} f''(\xi)$$

$n = 2$ Simpson's rule

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}\Big[f(x_0) + 4f(x_1) + f(x_2)\Big] - \frac{h^5}{90} f^{(4)}(\xi)$$

$n = 3$ Simpson's 3/8 rule

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}\Big[f(x_0)+3f(x_1)+3f(x_2)+f(x_3)\Big]-\frac{3h^5}{80} f^{(4)}(\xi)$$

Closed Newton-Cotes formulas (Burden *et al.*, 2022)

$n = 4$ Boole's rule

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2h}{45}\Big[7f(x_0) + 32f(x_1) + 12f(x_2) +$$

$$+32f(x_3) + 7f(x_4)\Big] - \frac{8h^7}{945} f^{(6)}(\xi)$$

$n = 5$

$$\int_{x_0}^{x_5} f(x)\,dx = \frac{5h}{288}\Big[19f(x_0) + 75f(x_1) + 50f(x_2) +$$

$$+50f(x_3) + 75f(x_4) + 19f(x_5)\Big] - \frac{275h^7}{12096} f^{(6)}(\xi)$$

$n = 6$ Weddle's rule

$$\int_{x_0}^{x_6} f(x)\,dx = \frac{h}{140}\Big[41f(x_0) + 236f(x_1) + 27f(x_2) +$$

$$+272f(x_3) + 27f(x_4) + 236f(x_5) + 41f(x_6)\Big] - \frac{9h^9}{1400} f^{(8)}(\xi)$$

Open Newton-Cotes formulas (Burden *et al.*, 2022)

The *open Newton-Cotes formulas* do not include the endpoints of $[a, b]$ as nodes. They use the nodes

$$x_k = x_0 + kh,\ k = \overline{0, n},\ h = \frac{b-a}{n+2},\ x_0 = a + h$$

This implies that $x_n = b - h$. We label the endpoints by setting $x_{-1} = a$ and $x_{n+1} = b$. Open formulas contain all the nodes used for the approximation within the open interval (a, b). The formulas become

$$\int_a^b f(x)\,dx = \int_{x_{-1}}^{x_{n+1}} f(x)\,dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

where $A_k = \displaystyle\int_a^b l_k(x)\,dx$

Open Newton-Cotes formulas (Burden *et al.*, 2022)

There exists $\xi \in (a, b)$ for which

$$\int_a^b f(x)\,dx = \sum_{k=0}^{n} A_k f(x_k) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_{-1}^{n+1} t^2(t-1)\dots(t-n)\,dt$$

if n is even and $f \in C^{n+2}[a, b]$ and

$$\int_a^b f(x)\,dx = \sum_{k=0}^{n} A_k f(x_k) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_{-1}^{n+1} t(t-1)\dots(t-n)\,dt$$

if n is odd and $f \in C^{n+1}[a, b]$.
Notice, as in the case of the closed methods, we have the degree of precision comparatively higher for the even methods than for the odd methods.

Open Newton-Cotes formulas (Burden *et al.*, 2022)

$n = 0$ Midpoint rule

$$\int_{x_{-1}}^{x_1} f(x)\,dx = 2hf(x_0) + \frac{h^3}{3} f^{(2)}(\xi)$$

$n = 1$

$$\int_{x_{-1}}^{x_2} f(x)\,dx = \frac{3h}{2}\Big[f(x_0) + f(x_1)\Big] + \frac{3h^3}{4} f^{(2)}(\xi)$$

$n = 2$

$$\int_{x_{-1}}^{x_3} f(x)\,dx = \frac{4h}{3}\Big[2f(x_0) - f(x_1) + 2f(x_2)\Big] + \frac{14h^5}{45} f^{(4)}(\xi)$$

Gauss Quadrature Rule (Burden *et al.*, 2022)

The Newton-Cotes formulas were derived by integrating interpolating polynomials. The error term in the interpolating polynomial of degree n involves the $(n+1)$st derivative of the function to be approximated, so a Newton-Cotes formula is exact when approximating the integral of any polynomial of degree less than or equal to n. All Newton-Cotes formulas use values of the function at equally spaced points. This restriction is convenient when the formulas are combined to form the composite rules, but it can significantly decrease the accuracy of the approximation. In Gaussian quadrature, the points for evaluation are chosen in an optimal rather than an equally spaced way.

Gauss Quadrature Rule (Süli and Mayers, 2003)

Gauss quadrature rule is

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

where the *quadrature weights* are

$$A_k = \int_a^b w(x)(l_k(x))^2\, dx$$

and the *quadrature points* x_k are chosen as the zeros of the polynomial of degree n from a system of orthogonal polynomials over the interval (a, b) with respect to the weight function w.

Gauss Quadrature Rule (Süli and Mayers, 2003)

The calculation of the weights and the quadrature points in a Gauss quadrature rule requires little work when the system of orthogonal polynomials is already known. If this is not known, it is necessary to construct the polynomial whose roots are the quadrature points.

Error estimation for Gauss quadrature: suppose that $f \in C^{2n+2}[a,b]$, $n \geqslant 0$. Then, $\exists\, \eta \in [a,b]$ s.t.

$$\int_a^b w(x) f(x) = \sum_{k=0}^{n} A_k f(x_k) + \frac{f^{(2n+2)}(\eta)}{(2n+2)!} \int_a^b w(x) \big(l(x)\big)^2 \, dx,$$

where

$$l(x) = (x - x_0) \ldots (x - x_n)$$

Gauss Quadrature Rule (Süli and Mayers, 2003)

Note that the Gauss quadrature rule gives the exact value of the integral when f is a polynomial of degree $2n+1$ or less, which is the highest possible degree that one can hope for with the $2n+2$ free parameters consisting of the quadrature weights A_k and the quadrature points x_k.
The maximum degree of exactness of the Gauss quadrature formula is $2n+1$.

Gaussian Quadrature on Arbitrary Intervals (Burden *et al.*, 2022)

An integral $\int_a^b f(x)\,dx$ over an arbitrary $[a, b]$ can be transformed into an integral over $[-1, 1]$ by using the change of variables

$$t = \frac{2x - a - b}{b - a} \Leftrightarrow x = \frac{a + b}{2} + \frac{b - a}{2}t$$

This permits Gaussian quadrature to be applied to any interval $[a, b]$ because

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f\left(\frac{a + b}{2} + \frac{b - a}{2}t\right)\frac{b - a}{2}\,dt$$

Gauss-Chebyshev I Quadrature (Quarteroni *et al.*, 2000)

If Gaussian quadratures are considered with respect to the Chebyshev I weight

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

then the nodes and coefficients are given by

$$x_{k-1} = \cos\left(\frac{2k-1}{2n+2}\pi\right),\ A_{k-1} = \frac{\pi}{n+1},\ k = \overline{1, n+1}.$$

We obtain

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)\,dx = \frac{\pi}{n+1}\sum_{k=0}^{n} f\left(\cos\left(\frac{2k-1}{2n+2}\pi\right)\right) +$$

$$+\frac{\pi}{(2n+2)!2^{2n+1}} f^{(2n+2)}(\xi),\ -1 < \xi < 1$$

Gauss-Chebyshev II Quadrature (Quarteroni *et al.*, 2000)

If Gaussian quadratures are considered with respect to the Chebyshev II weight

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

then the nodes and coefficients are given by

$$x_{k-1} = \cos\frac{k\pi}{n+2},\ A_{k-1} = \frac{\pi}{n+2}\sin^2\frac{k\pi}{n+2},\ k = \overline{1, n+1}$$

We obtain

$$\int_{-1}^{1}\sqrt{1-x^2}f(x)\,dx = \frac{\pi}{n+2}\sum_{k=0}^{n}\sin^2\frac{k\pi}{n+2}f\left(\cos\left(\frac{k\pi}{n+2}\right)\right) +$$

$$+\frac{\pi}{(2n+2)!2^{2n+3}}f^{(2n+2)}(\xi),\ -1 < \xi < 1.$$