- Exact Integration.
- Simple numerical methods.
- Advanced numerical methods.

- Standard techniques of integration – substitution, integration by parts or using identitites etc.

Exact Integration

- Standard techniques of integration – substitution, integration by parts or using identitites etc.
- Tables of integrals – consult various books.

Exact Integration

- Standard techniques of integration – substitution, integration by parts or using identitites etc.
- Tables of integrals – consult various books.
- Computer Algebra Systems (CAS).

Computer Algebra Systems

- Mathematica
- Matlab
- Sympy
- Magma
- Sagemath
- etc.

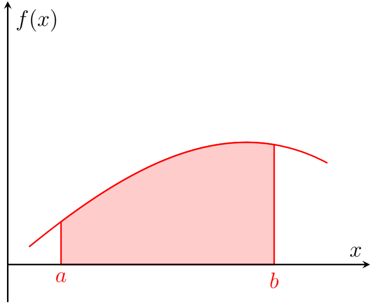
- Sometimes the integral that one is attempting to do is not readily available from the Tables of Integrals. Nor does the Computer Algebra System know how to give us an analytical result!

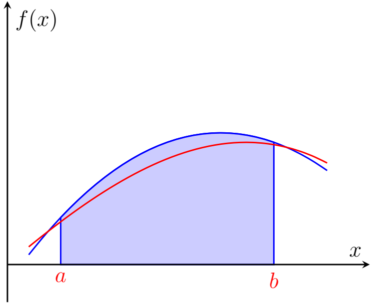
Numerical integration

- Sometimes the integral that one is attempting to do is not readily available from the Tables of Integrals. Nor does the Computer Algebra System know how to give us an analytical result!
- The data that we have collected and need to integrate is discrete data – ie is only known at some points (we may or maynot know the functional form).

Numerical integration can be based on fitting approximating functions (polynomials) to discrete data and integrating approximating functions.

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





- Function to be integrated is known only at a finite set of discrete points – ie the functional form of the integrand is not known.

Numerical Integration

- Function to be integrated is known only at a finite set of discrete points – ie the functional form of the integrand is not known.
- The only parameter under the users control is the degree of the approximating polynomial.

- Function to be integrated is known ie. the functional form of the integrand is known.

- Function to be integrated is known ie. the functional form of the integrand is known.
- The parameters under the users control are:

Numerical Integration

- Function to be integrated is known ie. the functional form of the integrand is known.
- The parameters under the users control are:
  - The degree of the approximating polynomial.

- Function to be integrated is known ie. the functional form of the integrand is known.
- The parameters under the users control are:
  - The degree of the approximating polynomial.
  - The total number of discrete points used in the calculation of the integral.

Numerical Integration

- Function to be integrated is known ie. the functional form of the integrand is known.
- The parameters under the users control are:
  - The degree of the approximating polynomial.
  - The total number of discrete points used in the calculation of the integral.
  - The location of the points where the function is discretized.

Direct fit polynomials

- This procedure is based on the idea that one can fit the data by a direct fit polynomial and integrate that polynomial:

$$f(x) \approx P_n(x) = a_0 + a_1 x + a_2 x^2 + \dots$$

Direct fit polynomials

- This procedure is based on the idea that one can fit the data by a direct fit polynomial and integrate that polynomial:

$$f(x) \approx P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots$$

then

$$I = \int_a^b f(x)dx \approx \int_a^b P_n(x)dx = \left[a_0 x + a_1 \frac{x^2}{2} + a_2 \frac{x^3}{3} + \ldots \right]_a^b$$

Direct fit polynomials

- This procedure is based on the idea that one can fit the data by a direct fit polynomial and integrate that polynomial:

$$f(x) \approx P_n(x) = a_0 + a_1 x + a_2 x^2 + \dots$$

then

$$I = \int_a^b f(x)dx \approx \int_a^b P_n(x)dx = \left[a_0 x + a_1 \frac{x^2}{2} + a_2 \frac{x^3}{3} + \dots \right]_a^b$$

- This procedure can be applied to data ($f(x)$) which is available for x that are equally spaced or unequally spaced.

Riemann Integral

- If $f(x)$ is a continuous function defined for $a \leq x \leq b$ and one divides the interval $[a, b]$ into n subintervals of equal width, $\Delta x = \frac{b-a}{n}$ then the definite integral

$$I = \int_a^b f(x)dx = \lim_{n \to \infty} \sum_{i=1}^{n} f(x_i^*)\Delta x$$

where $f(x_i^*)$ is the value of the function at an arbitrary point, x_i^*, in the interval x_i and $x_i + \Delta x$.

Riemann Integral

- If $f(x)$ is a continuous function defined for $a \leq x \leq b$ and one divides the interval $[a, b]$ into n subintervals of equal width, $\Delta x = \frac{b-a}{n}$ then the definite integral

$$I = \int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\Delta x$$

  where $f(x_i^*)$ is the value of the function at an arbitrary point, x_i^*, in the interval x_i and $x_i + \Delta x$.
- The Riemann integral can be interpreted as the area under the curve $y = f(x)$ from a to b.

Left end-point Riemann sum

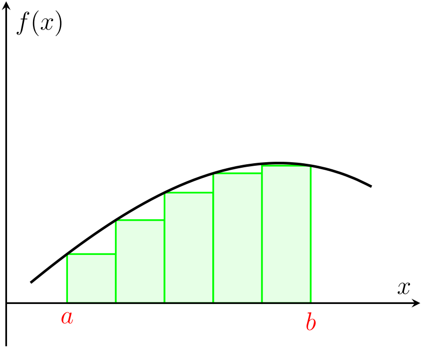
Simple integration methods

Left end-point Riemann sum

$$I = \int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_{i-1})\Delta x \quad \Delta x = \frac{b-a}{n}$$

Left end-point Riemann sum

$$I = \int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_{i-1})\Delta x \quad \Delta x = \frac{b-a}{n}$$



Right end-point Riemann sum

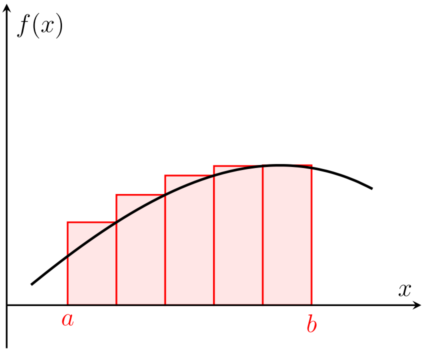
Simple integration methods

Right end-point Riemann sum

$$I = \int_a^b f(x)dx = \lim_{n \to \infty} \sum_{i=1}^{n} f(x_i)\Delta x \quad \Delta x = \frac{b-a}{n}$$

Right end-point Riemann sum

$$I = \int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\Delta x \quad \Delta x = \frac{b-a}{n}$$



Mid point Riemann sum

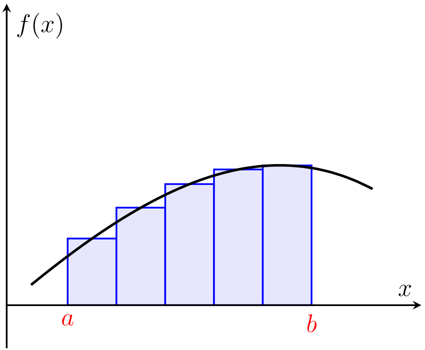
Simple integration methods

Mid point Riemann sum

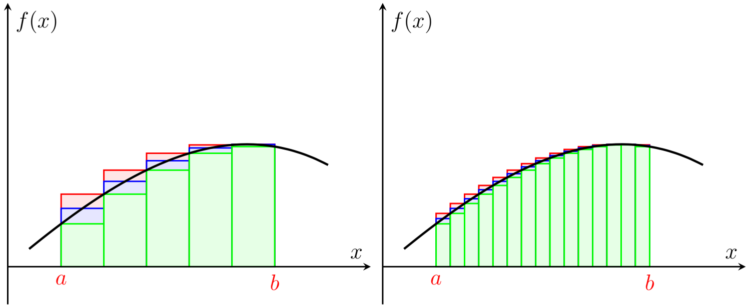
$$I = \int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(\frac{x_{i-1} + x_i}{2})\Delta x \quad \Delta x = \frac{b-a}{n}$$

Mid point Riemann sum

$$I = \int_a^b f(x)dx = \lim_{n\to\infty}\sum_{i=1}^{n} f(\frac{x_{i-1}+x_i}{2})\Delta x \quad \Delta x = \frac{b-a}{n}$$



Comparison of left, right and mid point



In principle, all three of the methods will converge to the same result – albiet very slowly!

Better method: Trapezoidal approximation

The area of the trapezoid that lies above the i^{th} subinterval:

$$S_i = \frac{\Delta x}{2}(f(x_{i-1}) + f(x_i))$$

Better method: Trapezoidal approximation

The area of the trapezoid that lies above the i^{th} subinterval:

$$S_i = \frac{\Delta x}{2}(f(x_{i-1}) + f(x_i))$$

then

$$I = \int_a^b f(x)dx = \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \dots$$
$$2f(x_{n-1}) + f(x_n)]$$

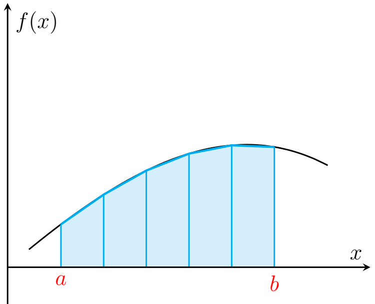
Better method: Trapezoidal approximation

The area of the trapezoid that lies above the i^{th} subinterval:

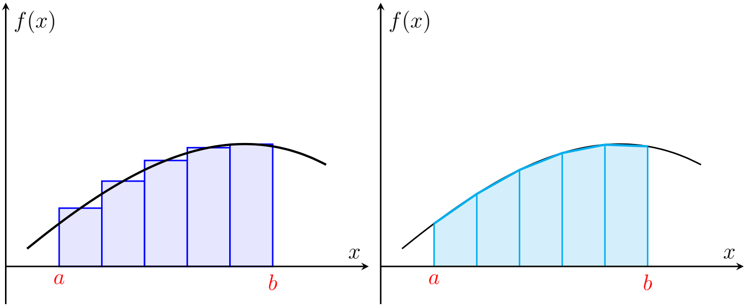
$$S_i = \frac{\Delta x}{2}(f(x_{i-1}) + f(x_i))$$

then

$$I = \int_a^b f(x)dx = \frac{\Delta x}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \dots \\ 2f(x_{n-1}) + f(x_n)]$$



Comparison of mid point and trapezoidal



First order interpolation

First order interpolation for the i^{th} subinterval:

$$\begin{aligned}
f(x) =& f(x_{i-1}) + f'(x_{i-1})x + higher\ order\ terms \\
=& f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{\Delta x}x + higher\ order\ terms.
\end{aligned}$$

First order interpolation

First order interpolation for the i^{th} subinterval:

$$\begin{aligned} f(x) =& f(x_{i-1}) + f'(x_{i-1})x + higher\ order\ terms \\ =& f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{\Delta x}x + higher\ order\ terms. \end{aligned}$$

then the integral for the i^{th} subinterval:

$$\begin{aligned} I = \int_{x_{i-1}}^{x_i} f(x)dx = \int_{x_{i-1}}^{x_i} &\left[f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{\Delta x}x \right] \\ =& \frac{\Delta x}{2}\left(f(x_{i-1}) + f(x_i)\right) \end{aligned}$$

First order interpolation

First order interpolation for the i^{th} subinterval:

$$\begin{aligned} f(x) =& f(x_{i-1}) + f'(x_{i-1})x + higher\ order\ terms \\ =& f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{\Delta x}x + higher\ order\ terms. \end{aligned}$$

then the integral for the i^{th} subinterval:

$$\begin{aligned} I = \int_{x_{i-1}}^{x_i} f(x)dx =& \int_{x_{i-1}}^{x_i} \left[f(x_{i-1}) + \frac{f(x_i) - f(x_{i-1})}{\Delta x}x \right] \\ =& \frac{\Delta x}{2}(f(x_{i-1}) + f(x_i)) \end{aligned}$$

Trapezoidal approximation is the application of first order interpolation for each subinterval.

Integration with second order interpolation

Using three point interpolation, one may write the Simpson's rule for integration:

Integration with second order interpolation

Using three point interpolation, one may write the Simpson's rule for integration:

$$I = \int_a^b f(x)dx = \frac{\Delta x}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots$$
$$2f(x_{n-1}) + 4f(x_{n-1}) + f(x_n)]$$

Integration with second order interpolation

Using three point interpolation, one may write the Simpson's rule for integration:

$$I = \int_a^b f(x)dx = \frac{\Delta x}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \ldots$$
$$2f(x_{n-1}) + 4f(x_{n-1}) + f(x_n)]$$

Number of n intervals should be even – if not then the last interval should be treated in some other way!

- Lagrange interpolation.
- Newton forward difference polynomials.
- Trapezoidal rule (revisited).
- Simpson's rule (revisited).
- Simpson's 3/8 rule.
- Integration error.

Lagrange Interpolation

Let us assume a set of numbers $x_0, \ldots, x_n$ and the corresponding function's ($f(x)$) values $f_0, \ldots, f_n$.

Lagrange Interpolation

Let us assume a set of numbers $x_0, \ldots, x_n$ and the corresponding function's ($f(x)$) values $f_0, \ldots, f_n$.

There exist a unique polynomial $P_n\left(x\right)$ of degree n such that $P_n\left(x_i\right) = f_i$ for all $i = 0, \ldots, n$.

Lagrange Interpolation

Let us assume a set of numbers $x_0, \ldots, x_n$ and the corresponding function's ($f(x)$) values $f_0, \ldots, f_n$.

There exist a unique polynomial $P_n(x)$ of degree n such that $P_n(x_i) = f_i$ for all $i = 0, \ldots, n$.

$$\begin{aligned} P_n(x) &= \sum_{i=0}^{n} C_i x^i \\ &= \sum_{i=0}^{n} a_i \prod_{i \neq j} (x - x_j) \end{aligned}$$

But as $P_n\left(x_i\right) = f_i$ for all $i = 0, \ldots, n$.

$$a_i = \frac{f_i}{\prod_{j \neq i}\left(x_i - x_j\right)}$$

But as $P_n(x_i) = f_i$ for all $i = 0, \ldots, n$.

$$a_i = \frac{f_i}{\prod_{j \neq i}(x_i - x_j)}$$

This polynomial is the Lagrange interpolation polynomial whose expression is given by :

$$\begin{aligned} P_n(x) &= \sum_{i=0}^{n} \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} f_i. \\ &= \prod_{i=0}^{n}(x - x_i) \sum_{i=0}^{n} \frac{f_i}{(x - x_i)\prod_{j \neq i}(x_i - x_j)}. \end{aligned}$$

Lagrange method is mostly a theoretical tool used for proving theorems. Not only it is not very efficient when a new point is added (which requires computing the polynomial again, from scratch), it is also numerically unstable.

Newton's divided differences Interpolation

If one writes the Lagrange interpolation polynomial slightly different basis functions, one obtains the Newton's interpolation formula given by:

$$
\begin{aligned}
P_n(x) = & \alpha_0 + \alpha_1 (x - x_0) + \alpha_2 (x - x_1)(x - x_0) + \dots \\
& + \alpha_n (x - x_{n-1}) \dots (x - x_1)(x - x_0)
\end{aligned}
$$

Newton's divided differences Interpolation

If one writes the Lagrange interpolation polynomial slightly different basis functions, one obtains the Newton's interpolation formula given by:

$$
\begin{aligned}
P_n(x) = &\alpha_0 + \alpha_1 (x - x_0) + \alpha_2 (x - x_1)(x - x_0) + \ldots \\
&+ \alpha_n (x - x_{n-1}) \ldots (x - x_1)(x - x_0)
\end{aligned}
$$

For $i = 0$:

$$f_0 = P_n(x_0) = \alpha_0$$

Newton's divided differences Interpolation

If one writes the Lagrange interpolation polynomial slightly different basis functions, one obtains the Newton's interpolation formula given by:

$$\begin{aligned} P_n\left(x\right) =& \alpha_0 + \alpha_1\left(x - x_0\right) + \alpha_2\left(x - x_1\right)\left(x - x_0\right) + \dots \\ & + \alpha_n\left(x - x_{n-1}\right)\dots\left(x - x_1\right)\left(x - x_0\right) \end{aligned}$$

For $i = 0$:

$$f_0 = P_n\left(x_0\right) = \alpha_0$$

For $i = 1$:

$$\begin{aligned} f_1 &= P_n\left(x_1\right) = \alpha_0 + \alpha_1(x_1 - x_0) \\ \alpha_1 &= \frac{f_1 - f_0}{x_1 - x_0} \end{aligned}$$

For $i = 2$:

$$f_2 = P_n(x_2) = \alpha_0 + \alpha_1(x_2 - x_0) + \alpha_2(x_2 - x_1)(x_2 - x_0)$$

$$\alpha_2 = \frac{(f_2 - f_1)/(x_2 - x_1) - (f_1 - f_0)/(x_1 - x_0)}{x_2 - x_0}$$

Similarly we can find $\alpha_3 ..... \alpha_{n-1}$.

Newton's divided differences Interpolation

For $i = 2$:

$$f_2 = P_n(x_2) = \alpha_0 + \alpha_1(x_2 - x_0) + \alpha_2(x_2 - x_1)(x_2 - x_0)$$

$$\alpha_2 = \frac{(f_2 - f_1)/(x_2 - x_1) - (f_1 - f_0)/(x_1 - x_0)}{x_2 - x_0}$$

Similarly we can find $\alpha_3.....\alpha_{n-1}$.
To express $\alpha_i, i = 0......n-1$ in a compact manner let us first define the following notation called divided differences:

$$\begin{aligned} f[x_k] &= f_k \\ f[x_k, x_{k+1}] &= \frac{f[x_{k+1}] - f[x_k]}{x_{k+1} - x_k} \\ f[x_k, x_{k+1}, x_{k+2}] &= \frac{f[x_{k+1}, x_{k+2}] - f[x_k, x_{k+1}]}{x_{k+2} - x_k} \\ f[x_k, x_{k+1}, \ldots, x_i, x_{i+1}] &= \frac{f[x_{k+1}, \ldots, x_{i+1}] - f[x_k, \ldots, x_i]}{x_{i+1} - x_k} \end{aligned}$$

Newton's divided differences Interpolation

With this notation:

$$
\begin{aligned}
\alpha_0 &= f[x_0] \\
\alpha_1 &= f[x_0, x_1] \\
\alpha_2 &= f[x_0, x_1, x_2] \\
\alpha_n &= f[x_0, x_1, \ldots, x_n]
\end{aligned}
$$

Newton's divided differences Interpolation

With this notation:

$$\alpha_0 = f[x_0]$$
$$\alpha_1 = f[x_0, x_1]$$
$$\alpha_2 = f[x_0, x_1, x_2]$$
$$\alpha_n = f[x_0, x_1, \ldots, x_n]$$

Now the polynomial can be rewritten as:

$$P_n(x) = \sum_{k=0}^{n} f[x_0, \ldots, x_k] \prod_{i=0}^{k-1} (x - x_i)$$

This is called as Newton's Divided Difference interpolation polynomial.

Sometimes in practice the data points x_i are equally spaced points:

$$x_i = x_0 + i \cdot h, \quad i = 0, 1, 2, \ldots, n$$

where x_0 is the starting point and h is the step size.

Sometimes in practice the data points x_i are equally spaced points:

$$x_i = x_0 + i \cdot h, \quad i = 0, 1, 2, \ldots, n$$

where x_0 is the starting point and h is the step size.

In this case, it is enough to calculate simple differences rather than the divided differences as in the non-uniformly placed data set case.

Sometimes in practice the data points x_i are equally spaced points:

$$x_i = x_0 + i \cdot h, \quad i = 0, 1, 2, \ldots, n$$

where x_0 is the starting point and h is the step size.

In this case, it is enough to calculate simple differences rather than the divided differences as in the non-uniformly placed data set case.

These simple differences can be forward differences (Δf_i) or backward differences (∇f_i). We will first look at forward differences and the interpolation polynomial based on forward differences.

The first order forward difference Δf_i is defined as

$$\Delta f_i = f_{i+1} - f_i$$

The first order forward difference Δf_i is defined as

$$\Delta f_i = f_{i+1} - f_i$$

The second order forward difference $\Delta^2 f_i$ is defined as

$$\Delta^2 f_i = \Delta f_{i+1} - \Delta f_i$$

Newton's divided differences Interpolation

The first order forward difference Δf_i is defined as

$$\Delta f_i = f_{i+1} - f_i$$

The second order forward difference $\Delta^2 f_i$ is defined as

$$\Delta^2 f_i = \Delta f_{i+1} - \Delta f_i$$

The k^{th} order forward difference $\Delta^k f_i$ is defined as

$$\Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i$$

Then the first divided difference $f[x_0, x_1]$,

$$f[x_0, x_1] = \frac{f_1 - f_0}{h} = \frac{\Delta f_0}{h}$$

$\therefore \Delta f_0 = hf[x_0, x_1]$

Then the first divided difference $f[x_0, x_1]$,

$$f[x_0, x_1] = \frac{f_1 - f_0}{h} = \frac{\Delta f_0}{h}$$

$\therefore \Delta f_0 = hf[x_0, x_1]$

By the definition of second order forward difference $\Delta^2 f_0$, we get

$$\begin{aligned}
\Delta^2 f_0 &= \Delta f_1 - \Delta f_0 \\
&= h\{f[x_1, x_2] - f[x_0, x_1]\} \\
&= h * 2h\{(f[x_1, x_2] - f[x_0, x_1])/(x_2 - x_0)\} \\
&= 2h^2 f[x_0, x_1, x_2]
\end{aligned}$$

Newton's divided differences Interpolation

Then the first divided difference $f[x_0, x_1]$,

$$f[x_0, x_1] = \frac{f_1 - f_0}{h} = \frac{\Delta f_0}{h}$$

$\therefore \Delta f_0 = hf[x_0, x_1]$

By the definition of second order forward difference $\Delta^2 f_0$, we get

$$\begin{aligned}
\Delta^2 f_0 &= \Delta f_1 - \Delta f_0 \\
&= h\{f[x_1, x_2] - f[x_0, x_1]\} \\
&= h * 2h\{(f[x_1, x_2] - f[x_0, x_1])/(x_2 - x_0)\} \\
&= 2h^2 f[x_0, x_1, x_2]
\end{aligned}$$

In general,

$$\Delta^k f_i = k!h^k f[x_i, x_{i+1}, x_{i+2} ... x_{i+k}]$$

Newton's divided differences Interpolation

The Newton forward difference interpolation polynomial may be written as follows:

$$P_n(x) = \sum_{k=0}^{n} \frac{\Delta^k f_0}{k! h^k} \prod_{i=0}^{k-1} (x - x_i)$$

Newton's divided differences Interpolation

The Newton forward difference interpolation polynomial may be written as follows:

$$P_n(x) = \sum_{k=0}^{n} \frac{\Delta^k f_0}{k! h^k} \prod_{i=0}^{k-1} (x - x_i)$$

We can rewrite the above in a simpler way:

$$x = x_0 + sh, \;\; p_n(s) = P_n(x)$$
$$x_k = x_0 + kh$$
$$x - x_k = (s - k)h$$

Then

$$
\begin{aligned}
p_n(s) &= \sum_{k=0}^{n} \frac{\Delta^k f_0}{k!h^k} \prod_{i=0}^{k-1} (s-i)h \\
&= \sum_{k=0}^{n} \frac{\Delta^k f_0}{k!h^k} \quad [s(s-1).......(s-k+1)]h^k \\
&= \sum_{k=0}^{n} \binom{s}{k} \Delta^k f_0
\end{aligned}
$$

Error in interpolation

When interpolating a given function f by a polynomial of degree n at the nodes $x_0, \ldots, x_n$ we get the error

$$f(x) - P_n(x) = f[x_0, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i)$$

If f is $n+1$ times continuously differentiable then for each x in the interval there exists ξ in that interval such that

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

The Trapezoidal Rule (revisited)

A first degree polynomial for a single interval (two points):

$$\Delta I = h \int_0^1 (f_0 + s\Delta f_0) ds = h \left[s f_0 + \frac{s^2}{2} \Delta f_0 \right]_0^1$$

The Trapezoidal Rule (revisited)

A first degree polynomial for a single interval (two points):

$$\Delta I = h \int_0^1 (f_0 + s\Delta f_0) ds = h \left[s f_0 + \frac{s^2}{2} \Delta f_0 \right]_0^1$$

$$\Delta I = h(f_0 + \frac{1}{2}\Delta f_0) = \frac{h}{2}(f_0 + f_1)$$

The Trapezoidal Rule (revisited)

A first degree polynomial for a single interval (two points):

$$\Delta I = h\int_0^1 (f_0 + s\Delta f_0)ds = h\left[sf_0 + \frac{s^2}{2}\Delta f_0\right]_0^1$$

$$\Delta I = h(f_0 + \frac{1}{2}\Delta f_0) = \frac{h}{2}(f_0 + f_1)$$

Applying over all intervals:

$$I = \sum_{i=0}^{n-1} \Delta I_i = \sum_{i=0}^{n-1} \frac{h}{2}(f_i + f_{i+1}) = \frac{h}{2}(f_0 + 2f_1 + \ldots + 2f_{n-1} + f_n)$$

The Trapezoidal Rule (revisited)

The error estimation can be done by integrating the error term. For a single interval:

$$E = h\int_0^1 \frac{s(s-1)}{2} h^2 f''(\xi) ds = -\frac{1}{12} h^3 f''(\xi) \sim \mathcal{O}(h^3)$$

The Trapezoidal Rule (revisited)

The error estimation can be done by integrating the error term. For a single interval:

$$E = h\int_0^1 \frac{s(s-1)}{2} h^2 f''(\xi) ds = -\frac{1}{12} h^3 f''(\xi) \sim \mathcal{O}(h^3)$$

Total error:

$$E_T = \sum_{i=0}^{n-1} E_i = -\frac{1}{12}(x_n - x_0) h^2 f''(\xi) \sim \mathcal{O}(h^2)$$

where $x_0 \leq \xi \leq x_n$.

The Simpson's Rule (revisited)

A second degree polynomial for two intervals (three points):

$$\Delta I = h \int_0^2 (f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0) ds$$

The Simpson's Rule (revisited)

A second degree polynomial for two intervals (three points):

$$\Delta I = h \int_0^2 (f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0) ds$$

$$\Delta I = \frac{h}{3}(f_0 + 4f_1 + f_2)$$

The Simpson's Rule (revisited)

A second degree polynomial for two intervals (three points):

$$\Delta I = h \int_0^2 (f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0) ds$$

$$\Delta I = \frac{h}{3}(f_0 + 4f_1 + f_2)$$

Applying over all intervals:

$$I = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 4f_{n-1} + f_n)$$

The Simpson's Rule (revisited)

The error estimation can be done by integrating the error term. For an interval:

$$E = h \int_0^2 \frac{s(s-1)(s-2)}{6} h^3 f'''(\xi) ds = 0$$

The Simpson's Rule (revisited)

The error estimation can be done by integrating the error term. For an interval:

$$E = h\int_0^2 \frac{s(s-1)(s-2)}{6} h^3 f'''(\xi) ds = 0$$

$$E = h\int_0^2 \frac{s(s-1)(s-2)(s-3)}{24} h^4 f''''(\xi) ds = -\frac{1}{90} h^5 f''''(\xi)$$

The Simpson's Rule (revisited)

The error estimation can be done by integrating the error term. For an interval:

$$E = h\int_0^2 \frac{s(s-1)(s-2)}{6} h^3 f'''(\xi) ds = 0$$

$$E = h\int_0^2 \frac{s(s-1)(s-2)(s-3)}{24} h^4 f''''(\xi) ds = -\frac{1}{90} h^5 f''''(\xi)$$

Total error:

$$E_T \sim \mathcal{O}(h^4)$$

The Simpson's 3/8 Rule

A third degree polynomial for three intervals (four points):

$$\Delta I = h\int_0^3 (f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{6}\Delta^3 f_0)ds$$

The Simpson's 3/8 Rule

A third degree polynomial for three intervals (four points):

$$\Delta I = h \int_0^3 (f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{6}\Delta^3 f_0) ds$$

$$\Delta I = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3)$$

The Simpson's 3/8 Rule

A third degree polynomial for three intervals (four points):

$$\Delta I = h \int_0^3 (f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{6}\Delta^3 f_0) ds$$

$$\Delta I = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3)$$

Applying over all intervals:

$$I = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + 2f_3 + \ldots + 3f_{n-1} + f_n)$$

The Simpson's 3/8 Rule

The error estimation can be done by integrating the error term. For an interval:

$$E = h\int_0^3 \frac{s(s-1)(s-2)(s-3)}{24} h^4 f''''(\xi) ds = -\frac{3}{80} h^5 f''''(\xi)$$

The Simpson's 3/8 Rule

The error estimation can be done by integrating the error term. For an interval:

$$E = h\int_0^3 \frac{s(s-1)(s-2)(s-3)}{24} h^4 f''''(\xi) ds = -\frac{3}{80} h^5 f''''(\xi)$$

Total error:

$$E_T \sim \mathcal{O}(h^4)$$

Same order as Simpson's 1/3 rule!

Other view on numerical quadrature

Quadrature isa weighted sum of finite number of sample values of the intergrand.

$$\int_a^b f(x)dx = \sum_{i=1}^{n} f(x_i)w_i$$

name	degree	Weights
Trapezoid	1	(h/2, h/2)
Simpson's	2	(h/3, 4h/3, h/3)
3/8	3	(3h/8, 9h/8, 9h/8, 3h/8)
Milne	4	(14h/45, 64h/45, 24h/45, 64h/45, 14h/45)

Integration error

- The best numerical evaluation of an integral can be done with relatively small number of sub-intervals ($n \sim 1000 - 10000$).
- It is possible to get errors close to machine precision with Simpson's rule and other higher order methods.