TOPIC 5 : NUMERICAL DIFFERENTIATION

Derivatives from divided difference tables.
We have seen that the interpolating polynomial of degree n − 1 passing through n data points 
\((x_i, f(x_i)), \ i = 1, 2, \ldots, n\) can be expressed as
\[ f(x) = f(x_1) + (x-x_1)f[x_1, x_2] + (x-x_1)(x-x_2)f[x_1, x_2, x_3] + (x-x_1)(x-x_2)(x-x_3)f[x_1, x_2, x_3, x_4] + \ldots \]

Now, the derivative of a product of n terms is a sum of n of these terms with one member of each term in
the sum replaced by its derivative; for example:
\[
\frac{d}{dx}(uvw) = u'vw + uv'w + uvw'
\]
So, we get this approximation for \( f'(x) \):
\[
f'(x) = f[x_1,x_2] \\
+ \{(x-x_2)+(x-x_1)\} f[x_1,x_2,x_3] \\
+ \{(x-x_2)(x-x_3)+(x-x_1)(x-x_3)+
( x-x_1)(x-x_2)\} f[x_1,x_2,x_3,x_4] \\
+ \ldots
\]
If we apply this formula to the table

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( f_i )</th>
<th>[( f[x_{i},x_{i+1}] )]</th>
<th>[( f[x_{i},x_{i+1},x_{i+2}] )]</th>
<th>[( f[x_{i},x_{i+1},x_{i+2},x_{i+3}] )]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
then the estimated value for \( f'(4.1) \) is
\[
4 + \{(4.1-3) + (4.1-2)\} \times 1 \\
+ \{(4.1-3)(4.1-5) + (4.1-2)(4.1-5) \\
+ (4.1-2)(4.1-3)\} \times 0
\]
= 7.2

Note that the table applies to the function \( f(x) = x^2 - x + 1 \), whose derivative is \( f'(x) = 2x - 1 \), and the value we have obtained at \( x = 4.1 \) is exact. This is expected, since our interpolating cubic is reduced to a quadratic as \( f[x_1,x_2,x_3,x_4] = 0 \).

When the x-values are evenly spaced, we can use ordinary forward differences instead of divided differences. For example, let us compute the derivative at \( x = 3.3 \) of a cubic polynomial created from the following table:
<table>
<thead>
<tr>
<th>$x$</th>
<th>$f(x)$</th>
<th>$\Delta f$</th>
<th>$\Delta^2 f$</th>
<th>$\Delta^3 f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.30</td>
<td>3.669</td>
<td>3.017</td>
<td>2.479</td>
<td>2.041</td>
</tr>
<tr>
<td>1.90</td>
<td>6.686</td>
<td>5.496</td>
<td>4.520</td>
<td>3.713</td>
</tr>
<tr>
<td>2.50</td>
<td>12.182</td>
<td>10.016</td>
<td>8.233</td>
<td>6.771</td>
</tr>
<tr>
<td>3.10</td>
<td>22.198</td>
<td>18.249</td>
<td>15.004</td>
<td>12.333</td>
</tr>
<tr>
<td>3.70</td>
<td>40.447</td>
<td>33.253</td>
<td>27.337</td>
<td></td>
</tr>
<tr>
<td>4.30</td>
<td>73.700</td>
<td>60.590</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.90</td>
<td>134.290</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$f'(x) = \frac{\Delta f(x_1)}{h} + \{(x-x_2)+(x-x_1)\} \frac{\Delta^2 f(x_1)}{2h^2} + \{(x-x_2)(x-x_3)+(x-x_1)(x-x_3)\} \frac{\Delta^3 f(x_1)}{6h^3} + (x-x_1)(x-x_2)\}$

To centre the $x$-value, we should take $x_1 = 2.50$
\[ f'(3.3) = \frac{10.016}{0.6} \]
\[ + \{(3.3-3.1) + (3.3-2.5)\} \times \frac{8.233}{2 \times 0.6^2} \]
\[ + \{(3.3-3.1)(3.3-3.7)\} + (3.3-2.5)(3.3-3.7) \]
\[ + \{(3.3-2.5)(3.3-3.1)\} \times \frac{6.771}{6 \times 0.6^3} \]
\[ = 26.875 \]

Note that the table applies to \( f(x) = e^x \), and that the correct \( f'(3.3) \) is 27.113 to 3D. If we use the "next-term rule" to estimate the error of our answer, we obtain 0.315 compared with the actual error of 0.238.

Finite difference approximations
Although finite difference formulae are inappropriate for "noisy" data or
“rough” functions, they are useful for approximating derivatives of a smooth function that is known analytically or that can be evaluated accurately for any given argument. Given a smooth function $f : \mathbb{R} \to \mathbb{R}$, we wish to approximate its first and second derivatives at a point $x$. Consider the Taylor series expansions

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2} h^2 + \frac{f'''(x)}{6} h^3 + \ldots$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2} h^2 - \frac{f'''(x)}{6} h^3 + \ldots$$
Solving for \( f'(x) \) in the first series:

\[
    f'(x) = \frac{f(x + h) - f(x)}{h} - \frac{f''(x)}{2} h + \ldots
\]

\[
    \approx \frac{f(x + h) - f(x)}{h}
\]

which gives an approximation that is first-order accurate since the dominant term in the remainder of the series is \( O(h) \).

From the second series:

\[
    f'(x) = \frac{f(x) - f(x - h)}{h} + \frac{f''(x)}{2} h + \ldots
\]

\[
    \approx \frac{f(x) - f(x - h)}{h}
\]

which is also first-order accurate.
Subtracting the second series from the first gives the centred difference formula

\[ f''(x) \approx \frac{2h}{f(x + h) - f(x - h)} - \frac{f''''(x)}{6} h^2 + \ldots \]

which is second-order accurate. Finally, adding the two series together gives a centred difference formula for the second derivative:

\[ f''(x) = \frac{f(x + h) - 2f(x) + f(x - h)}{h^2} - \frac{f^{IV}(x)}{12} h^2 + \ldots \]

\[ \approx \frac{2h}{f(x + h) - f(x - h)} h^2 \]

which is second-order accurate. By using function values at additional points, x ± 2h, x ± 3h,..., we can get finite difference approximations with higher accuracy or for higher-order derivatives. Consider the following example: For 

\[ f(x) = x e^{-\frac{1}{2}x} \]

estimate the value of \( f'(0.3) \) using a centred difference formula and \( h = 0.1, 0.05, 0.025 \).

\[ f'(0.3) \approx \frac{1}{2h} [f(0.3+h) - f(0.3-h)] \]

\begin{align*}
\text{h} & \quad \text{Value} & \quad \text{Error} \\
0.1 & \quad 0.73262 & \quad 0.00102 \\
0.05 & \quad 0.73186 & \quad 0.00026 \\
0.025 & \quad 0.73167 & \quad 0.00006
\end{align*}
Note that errors decrease in proportion to $h^2$ (successive errors are decreased by a factor of about 4). It is clear, then, that decreasing the value of $h$ increases the accuracy of our computation of the derivative, because this procedure reduces truncation error.

As $h$ is reduced, however, we are required to subtract function values that are almost equal, and this incurs a large error due to round-off. The table below shows that the increase in round-off error as $h$ gets smaller causes the best accuracy to occur at some intermediate point.
<table>
<thead>
<tr>
<th>$h$</th>
<th>$f'(x)$</th>
<th>$f''(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1E-00</td>
<td>0.1001663E 01</td>
<td>0.1000761E 01</td>
</tr>
<tr>
<td>0.1E-01</td>
<td>0.1000001E 01</td>
<td>0.9959934E 00</td>
</tr>
<tr>
<td>0.1E-02</td>
<td>0.9999569E 00</td>
<td>0.8940693E 00</td>
</tr>
<tr>
<td>0.1E-03</td>
<td>0.9959930E 00</td>
<td>-0.8344640E 02</td>
</tr>
<tr>
<td>0.1E-04</td>
<td>0.9775158E 00</td>
<td>-0.4768363E 04</td>
</tr>
<tr>
<td>0.1E-05</td>
<td>0.9834763E 00</td>
<td>-0.5960462E 05</td>
</tr>
<tr>
<td>0.1E-06</td>
<td>0.5960463E 00</td>
<td>-0.1192093E 08</td>
</tr>
<tr>
<td>0.1E-07</td>
<td>0.2980230E 01</td>
<td>-0.5960458E 09</td>
</tr>
<tr>
<td>0.1E-08</td>
<td>0.2980229E 02</td>
<td>-0.5960459E 11</td>
</tr>
<tr>
<td>0.1E-09</td>
<td>0.2980229E 03</td>
<td>-0.5960456E 13</td>
</tr>
</tbody>
</table>

**Results with single precision:**

$\frac{\Delta y}{\Delta x} = \frac{f(x + h) - f(x)}{h}$

**Results with double precision:**

$\frac{\Delta y}{\Delta x} = \frac{f(x + h) - f(x)}{h}$
So, numerical differentiation is basically an unstable operation. We shall see that the same problem does not occur with numerical integration, where function values are added together, producing an inherently stable situation. This is generally true of global computations such as integration, in contrast to those that are local in nature such as differentiation. In differentiation of “noisy” data (i.e., observations subject to errors of measurement), the errors can so influence the values of derivatives calculated by numerical procedures that these derivative values may be meaningless. The recommendation is to smooth the data first, and then
obtain the derivative of this approximation to the data i.e. we don’t try to represent the function by one that fits exactly to the data points because this fits to the errors as well as to the trend of the information; rather, we approximate with a smooth curve that is closer to the truth than is the data.

Some functions are inherently rough i.e. the function values change rapidly. A set of data points may not reflect this, and indicate a smoother function than actually exists. In this situation, valid values of derivatives cannot be obtained.

Richardson extrapolation
In the computation of a derivative above, we have computed an
approximate value for the derivative based on stepsize. Ideally, we would like to obtain the limiting value as the stepsize approaches zero. Based on values for nonzero stepsize, Richardson’s extrapolation estimates what the value would be for a stepsize of zero.

Let \( F(h) \) be the value obtained with stepsize \( h \), and suppose that

\[
F(h) = a_0 + a_1 h^p + O(h^r), \quad r > p
\]

where \( a_0 \) and \( a_1 \) are unknown (in fact, \( F(0) = a_0 \) is the quantity we seek).

Suppose that \( F \) has been computed for two stepsizes, say \( h \) and \( qh \).

Then

\[
F(h) = a_0 + a_1 h^p + O(h^r)
\]

and

\[
F(qh) = a_0 + a_1 (qh)^p + O(h^r)
\]
from which
\[ a_0 = F(h) + \frac{F(h) - F(qh)}{q^p - 1} + O(h^r) \]
Let us use Richardson extrapolation to find an approximation to the value of the derivative of \( \sin x \) at \( x = 1 \). Using the first order accurate difference formula
\[ f'(x) = \frac{f(x + h) - f(x)}{h} \]
then
\[ F(h) = \frac{\sin 1.25 - \sin 1}{0.25} = 0.430055 \]
and
\[ F(2h) = \frac{\sin 1.5 - \sin 1}{0.5} = 0.312048 \]
and
\[ F(h) = a_0 + a_1 h + O(h^r) \]
(i.e. \( p = 1 \) and \( r = 2 \))
then \( F(0) = a_0 \)

\[ = F(h) + \frac{F(h) - F(2h)}{2 - 1} \]

\[ = 2 F(h) - F(2h) \]

\[ = 0.548061 \]

c.f. \( \cos 1 = 0.540302 \)
TOPIC 5: NUMERICAL INTEGRATION

If we need to evaluate a definite integral involving a function whose antiderivative cannot be found (e.g. \( \sqrt{1-x^3} \), \( \frac{\cos x}{x} \), \( e^{x^2} \)), we must resort to an approximation technique. The numerical approximation of definite integrals is known as numerical quadrature.

In what follows, we assume that the integrand is continuous and smooth, and that the interval of integration is finite. Nonetheless, an integral is an infinite summation, which we will approximate by a finite sum. Such a finite sum, in which the function to be integrated is sampled at a finite
number of points in the interval of integration, is called a quadrature rule. The questions are: How do we choose the sample points and how do we weight their contributions to the quadrature formula so that we obtain the desired level of accuracy at a reasonable computational cost (note that, in numerical quadrature, computational work can be measured by the number of evaluations of the function to be integrated). An n-point quadrature rule has the form

\[ I(f) = \int_a^b f(x) \, dx = \sum_{i=1}^{n} w_i f(x_i) + R_n; \]
the points $x_i$ at which the function $f$ is evaluated are called the nodes or abscissas, the multipliers $w_i$ are called the weights, and $R_n$ is the remainder or error. $R_n$ usually involves information such as a higher degree of $f$ which is either inconvenient or impossible to obtain, so we usually settle for estimating the possible error in using a given rule. Quadrature rules are based on polynomial interpolation. The function to be integrated is sampled at some number of points, the polynomial that interpolates the function at these points is found, and the integral of the interpolating polynomial is taken as an approximation to the integral of the
original function. In practice, however, the interpolating polynomial is not determined explicitly each time a particular integral is to be evaluated. Instead, polynomial interpolation is used to determine the weights corresponding to the chosen nodes in a quadrature rule, which are then used to approximate any integral over the interval. If the nodes $x_i$ are equally spaced on the interval $[a,b]$, the quadrature rule is known as a Newton-Cotes quadrature rule; such a rule is said to be closed if its nodes include the endpoints $a$ and $b$, otherwise the rule is said to be open.
Interpolation at one, two, and three equally spaced points gives the first three Newton-Cotes quadrature rules:

- Interpolating the function value at the midpoint of the interval by a constant (i.e. a polynomial of degree 0) gives a one-point quadrature rule called the midpoint rule or the rectangle rule:

\[ I(f) \approx m(f) = (b - a)f\left(\frac{a + b}{2}\right) \]

- Interpolating the function values at the two endpoints of the interval by a straight line (i.e. a polynomial of degree 1) gives a two-point quadrature rule called the trapezoid rule:

\[ I(f) \approx T(f) = \left(\frac{b - a}{2}\right)[f(a) + f(b)] \]
• Interpolating the function values at three points (two endpoints and the midpoint) by a quadratic gives a three-point quadrature rule called Simpson’s rule:

\[ I(f) \approx S(f) = \left( \frac{b - a}{6} \right) [f(a) + 4f\left( \frac{a + b}{2} \right) + f(b)] \]

To derive the trapezoid rule for more than one strip we begin by dividing the interval \( a \leq x \leq b \) into \( n \) equal subintervals, each of width \( \left( \frac{b - a}{n} \right) \), such that

\[ a = x_0 < x_1 < x_2 < \ldots < x_n = b \]

For the trapezoid rule, we form a trapezium for each subinterval:
The sum of the areas of the \( n \) trapeziums is

\[
\frac{b-a}{2n} \left[ (f(x_0) + f(x_1)) + (f(x_1) + f(x_2)) + \ldots + (f(x_{n-1}) + f(x_n)) \right]
\]

\[
= \frac{b-a}{2n} \left[ f(x_0) + 2f(x_1) + \ldots + 2f(x_{n-1}) + f(x_n) \right]
\]

\[
= \frac{h}{2} \left[ f(x_0) + 2f(x_1) + \ldots + 2f(x_{n-1}) + f(x_n) \right]
\]
where \( h = \frac{b - a}{n} \) = width of each subinterval.

Let us use the trapezoid rule to find an approximation for

\[
\int_{0}^{\pi} x \sin x \, dx
\]

using 6 strips:

<table>
<thead>
<tr>
<th>( x )</th>
<th>( f(x) = x \sin x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \pi/6 )</td>
<td>( \pi/12 )</td>
</tr>
<tr>
<td>( \pi/3 )</td>
<td>( \pi/2\sqrt{3} )</td>
</tr>
<tr>
<td>( \pi/2 )</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>( 2\pi/3 )</td>
<td>( \pi/\sqrt{3} )</td>
</tr>
<tr>
<td>( 5\pi/6 )</td>
<td>5( \pi/12 )</td>
</tr>
<tr>
<td>( \pi )</td>
<td>0</td>
</tr>
</tbody>
</table>
\[
\int_0^\pi x \sin x \, dx \\
\approx \frac{1}{2} \frac{\pi}{6} \left[ 0 + 2 \cdot \frac{\pi}{12} + 2 \cdot \frac{\pi}{2\sqrt{3}} + 2 \cdot \frac{\pi}{2} \right. \\
+ 2 \cdot \frac{\pi}{\sqrt{3}} + 2 \cdot \frac{5\pi}{12} + 0 \bigg] \\
\frac{\pi^2}{12} (2 + \sqrt{3}) \\
= 3.0694, \text{ to } 4D
\]

The exact value for the integral is \( \pi \), i.e. 3.1416 to 4D, so that the error is 0.0722 to 4D. In general, this error can be decreased by increasing the number of strips (i.e. by decreasing the width of the subintervals), but we have to be careful that the increase in the accuracy of the approximation
which results from this procedure is not outweighed by an increase in round-off error. The formula for estimating the maximum error involved in the use of the trapezoid rule is as follows: If $f$ has a continuous second derivative on the interval $a \leq x \leq b$, then the upper bound on the magnitude of the error involved in approximating

$$
\int_a^b f(x)\,dx \text{ by the trapezoid rule is}
$$

$$
\frac{(b - a)^3}{12n^2} \cdot \max |f''(x)|
$$

$$
= \frac{(b - a)}{12} h^2 \max |f''(x)|
$$
(Note that the local error of the trapezoid rule, i.e. the error for 1 strip, is of magnitude
\[ \frac{1}{12} h^3 f''(\varepsilon_1), x_0 < \varepsilon_1 < x_1 \]
To develop the global error for n strips, we sum the local errors to obtain
\[ \frac{1}{12} h^3 [f''(\varepsilon_1) + f''(\varepsilon_2) + \ldots + f''(\varepsilon_n)] \]
If we assume that \( f''(x) \) is continuous on \( a \leq x \leq b \), there is some value of \( x \), say \( x = \varepsilon \), on the interval \([a,b] \) at which the sum above is equal to
\[ n f''(\varepsilon). \] As \( nh = b - a \), the magnitude of the global error
becomes
\[
\frac{1}{12} (b - a) h^2 f''(\varepsilon). \]

Let us use the formula
\[
\frac{(b - a)^3}{12n^2} \times \max |f''(x)|
\]
to find the value of \( n \) so that the trapezoid rule will approximate the value of
\[
\frac{1}{\sqrt{1 + x^2}} dx
\]
with an error less than 0.01:

If \( f(x) = \sqrt{1 + x^2} \)

then \( f''(x) = \frac{1}{(1 + x^2)^{3/2}} \)
and the maximum value of $|f''(x)|$ on the interval $0 \leq x \leq 1$ is 1.

Since $b - a = 1$, the upper bound to the error is of magnitude

\[
\frac{1^3}{12n^2} \cdot 1
\]

and this needs to be less than 0.01; hence

\[
\frac{1}{12n^2} < \frac{1}{100}
\]

\[
12n^2 > 100
\]

\[
n^2 > \frac{100}{12}
\]

\[
n > 2.89 \text{ (n must be + ve)}
\]
i.e. we must use 3 strips.
An algorithm for trapezoid rule integration follows:

**An Algorithm for Composite Trapezoidal Rule Integration**

Given a function $f(x)$:

(Get user inputs)

INPUT

- $a, b =$ endpoints of interval,
- $n =$ number of intervals.

(Do the integration)

SET $h = (b - a)/n$.

SET SUM = 0.

DO FOR $i = 1$ TO $n - 1$ STEP 1

- SET $x = a + h \times i$.
- SET SUM = SUM + 2 * $f(x)$

ENDDO (FOR $i$).

SET SUM = SUM + $f(a) + f(b)$.

SET ANS = SUM * h/2.

The value of the integral is given by ANS.

Let us now move on to Simpson’s Rule for estimating $\int_{a}^{b} f(x)dx$.

Whereas the Trapezoid Rule approximates $f$ on each subinterval by a first degree polynomial,
Simpson’s Rule approximates $f$ by a quadratic $Ax^2 + Bx + C$. We begin by showing that

$$
\int_{a}^{b} p(x) \, dx = \frac{b-a}{6} \left[ p(a) + 4p\left(\frac{a+b}{2}\right) + p(b) \right]
$$

where $p(x) = Ax^2 + Bx + C$.

$$
\int_{a}^{b} p(x) \, dx
$$

$$
= \int_{a}^{b} (Ax^2 + Bx + C) \, dx
$$

$$
= \left[ \frac{Ax^3}{3} + \frac{Bx^2}{2} + Cx \right]_{a}^{b}
$$

$$
= \frac{A(b^3 - a^3)}{3} + \frac{B(b^2 - a^2)}{2} + C(b - a)$$
\[ \frac{b-a}{6} \left[ 2A(a^2 + ab + b^2) + 3B(b+a) + 6C \right] \]

\[ = \frac{b-a}{6} \left[ (Aa^2 + Ba + C) + 4 \left\{ A \left( \frac{b+a}{2} \right)^2 + B \left( \frac{b+a}{2} \right) + C \right\} + (Ab^2 + Bb + C) \right] \]

\[ = \frac{b-a}{6} \left[ p(a) + 4p \left( \frac{a+b}{2} \right) + p(b) \right] \]
If we divide the interval \( a \leq x \leq b \) into \( n \) equal subintervals each of width \( h \), \( n \) is required to be even, and the subintervals grouped in pairs (called panels) such that

\[
a = x_0 < x_1 < x_2 < x_3 < x_4 < ... < x_{n-2} < x_{n-1} < x_n = b
\]

On the interval \( x_0 \leq x \leq x_2 \), the polynomial \( p \) passes through the points \((x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2))\). \( p \) is an approximation for \( f \) on this subinterval; hence

\[
\int_{x_0}^{x_2} f(x) \, dx \\
\approx \int_{x_0}^{x_2} p(x) \, dx
\]
\[
= \frac{x_2 - x_0}{6} [p(x_0) + 4p\left(\frac{x_0 + x_2}{2}\right) + p(x_2)] = \frac{2[(b - a) / n]}{6} [p(x_0) + 4p(x_1) + p(x_2)]
\]

\[
\approx \frac{h}{3} \left[f(x_0) + 4f(x_1) + f(x_2)\right]
\]

Repeating this procedure on the entire interval \(a \leq x \leq b\) produces Simpson’s Rule:

\[
\int_{a}^{b} f(x) \, dx \approx \frac{h}{3} \left[f(x_0) + 4f(x_1) + 2f(x_2) + \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right]
\]

Let us use Simpson’s Rule to again find an approximation for \(\int_{0}^{\pi} x \sin x \, dx\)
using 6 strips:
From our earlier table
\[
\int_0^\pi x \sin x \, dx \\
\approx \frac{1}{36} [0 + 4 \cdot \frac{\pi}{12} + 2 \cdot \frac{\pi}{2\sqrt{3}} + 4 \cdot \frac{\pi}{2} + 2 \cdot \frac{\pi}{\sqrt{3}} + 4 \cdot \frac{5\pi}{12} + 0]
\]
\[
= \frac{\pi^2}{18} [4 + \sqrt{3}]
\]
= 3.1429, to 4D

c.f. exact value of 3.1416 to 4D
i.e. error is 0.0013 to 4D

The maximum error involved in the use of Simpson’s Rule is as follows:
If $f$ has a continuous fourth derivative on the interval $a \leq x \leq b$, then the upper bound on the magnitude of the error involved in the approximation of

$$\int_{a}^{b} f(x) \, dx$$

is

$$\frac{(b - a)^5}{180n^4} \max |f^{IV}(x)|$$

$$\frac{(b - a)}{180} h^4 \max |f^{IV}(x)|$$

(Note that, as in the case of the trapezoid rule, this global error comes from summing local errors; for Simpson’s Rule, the local error has a magnitude of $\frac{1}{90} h^5 f^{IV}(\epsilon)$. )
An algorithm for Simpson's Rule integration follows:

**Algorithm for Simpson's \( \frac{1}{3} \) Rule Integration**

Given a function, \( f(x) \):

(Get user inputs)

**INPUT:**

\[ a, b = \text{endpoints of interval.} \]
\[ n = \text{number of intervals (n must be even).} \]

(Do the integration)

\[ \text{SET } h = (b - a)/n. \]
\[ \text{SET } \text{SUM} = 0. \]

DO FOR \( i = 1 \) TO \( n/2 \) STEP 1

\[ \text{SET } x = a + h + 2 \cdot h \cdot i. \]
\[ \text{SET } \text{SUM} = \text{SUM} + 4 \cdot f(x). \]

IF \( i \neq n/2 \) THEN:

\[ \text{SET } \text{SUM} = \text{SUM} + 2 \cdot f(x + h). \]

ENDIF.

ENDDO (FOR i).

\[ \text{SET } \text{SUM} = \text{SUM} + f(a) + f(b). \]
\[ \text{SET } \text{ANS} = \text{SUM} \cdot h/3. \]

The value of the integral is given by \( \text{ANS} \).

The accuracy of a quadrature rule is often characterized by the notion of polynomial degree. A quadrature rule is said to be of polynomial degree \( d \) if it is exact (i.e. its
remainder is zero) for every polynomial of degree \( d \) but is not exact for some polynomial of degree \( d + 1 \). Since an \( n \)-point Newton-Cotes rule is based on an interpolating polynomial of degree \( n - 1 \), we would expect such a rule to have a polynomial degree of \( n - 1 \). Thus we would expect the trapezoid rule to have polynomial degree 1 and Simpson’s Rule to have polynomial degree 2. We have seen, however, that the error bound for Simpson’s Rule depends on the fourth derivative, which vanish for cubics, so that Simpson’s Rule is of polynomial degree 3 rather than 2. In general, an odd-order Newton-Cotes rule gains an extra degree beyond
that of the polynomial interpolant on which it is based.

**Gaussian Quadrature**

The Newton-Cotes formulae are based on evaluations of a function at equally spaced values of the independent variable. With a formula of (say) three terms, there were three parameters to be found viz. the coefficients applied to each of the functional values. If we remove the requirement that the function be evaluated at pre-determined x-values, a three-term formula now requires the evaluation of six parameters viz. the three x-values plus the three coefficients. Formulas
based on this principle are called Gaussian quadrature formulas.

Gaussian integration formulae are usually expressed in terms of the interval of integration [-1,1]; for other intervals, a change of variable is needed to transform the problem so that it utilizes the interval [-1,1].

Let us determine the Gaussian quadrature rule for two points, which is exact for polynomials up to and including degree 3. We need to determine the parameters \( a, b, x_1, \) and \( x_2 \) in the formula

\[
\int_{-1}^{1} f(x) \, dx \approx a \, f(x_1) + b \, f(x_2)
\]
For $f(x) = x^3$:  
\[ \int_{-1}^{1} f(x) \, dx = 0 \]

\[ = a x_1^3 + b x_2^3 \]

For $f(x) = x^2$:  
\[ \int_{-1}^{1} f(x) \, dx = \frac{2}{3} \]

\[ = a x_1^2 + b x_2^2 \]

For $f(x) = x$:  
\[ \int_{-1}^{1} f(x) \, dx = 0 \]

\[ = a x_1 + b x_2 \]

For $f(x) = 1$:  
\[ \int_{-1}^{1} dx = 2 \]

\[ = a + b \]

Multiplying the third equation by $x_1^2$ and subtracting from the first:  

\[ 0 = b(x_2^3 - x_2x_1^2) \]
\[ 0 = bx_2(x_2 - x_1)(x_2 + x_1) \]
from which
\[ b = 0 \text{ or } x_2 = 0 \text{ or } x_1 = x_2 \text{ or } x_1 = -x_2. \]
Only the last of these is satisfactory (the others are invalid or reduce our formula to a single term).

Hence \( a = b = 1 \)
\[ x_1 = -x_2 = \sqrt{\frac{1}{3}} \approx 0.5774 \]
and so
\[ \int_{-1}^{1} f(x)dx \approx f(-0.5774) + f(0.5774) \]
Consider the example:
\[
\int_{-1}^{1} e^{-x^2} \, dx \\
\approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \\
\quad \left(-\frac{1}{\sqrt{3}}\right)^2 + \left(\frac{1}{\sqrt{3}}\right)^2 \\
= e^{-\frac{1}{3}} + e^{-\frac{1}{3}} \\
= e^{\frac{-1}{3}} + e^{\frac{-1}{3}} \\
= 1.433 \quad \text{to 4S}
\]

If we have an integral on the interval \([a, b]\), we start by writing the desired integral in terms of some variable other than \(x\), say \(t\):

\[
\int_{a}^{b} f(t) \, dt
\]
The change of variable that is required to convert an integral on the interval \( t \in [a,b] \) to the interval \( x \in [-1,1] \) is
\[
t = \frac{(b - a)x + (b + a)}{2}
\]
So \( \int_{a}^{b} f(t) \, dt \)
\[
= \frac{b - a}{2} \int_{-1}^{1} f \left[ \frac{(b - a)x + (b + a)}{2} \right] \, dx
\]
Consider the example \( \int_{0}^{2} e^{-t^2} \, dt \)
Here \( a = 0, \ b = 2 \),
so \( t = x + 1 \) and \( dt = dx \)
and we seek
\[ \int_{-1}^{1} e^{-(x+1)^2} \, dx \]
\[ \approx e^{-1.5774^2} + e^{-0.4226^2} \]
\[ = 0.9195 \quad \text{to 4S} \]

The power of the Gaussian method is that it gives results of somewhat better accuracy with Simpson’s Rule (0.8299, n = 2; 0.8818, n = 4; to 4D), and requires less function evaluations.

Gaussian quadrature can be extended beyond two terms e.g. the Gauss quadrature rule for n = 3 evaluation points, which is exact for polynomials up to and including degree 2n – 1 = 5, is
\[ \int_{-1}^{1} f(x) \, dx \approx \frac{5}{9} f\left(-\sqrt{\frac{3}{5}} \right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}} \right) \]

\[
\approx \frac{5}{9} f(-0.7746) + \frac{8}{9} f(0) + \frac{5}{9} f(0.7746)
\]

For a given number of evaluation points, the x-values are the roots of a Legendre polynomial. Legendre polynomials are defined as follows:

\((n+1)L_{n+1}(x)-(2n+1)xL_n(x)+nL_{n-1}(x)=0\)

\(L_0(x) = 1\)

\(L_1(x) = x\)
So $L_2(x) = \frac{3xL_1(x) - 1L_0(x)}{2}$

$= \frac{3x^2 - 1}{2}$

$= 0$ if $x = \pm \sqrt{\frac{1}{3}}$, the x-values for $n=2$

And $L_3(x) = \frac{5x^3 - 3x}{2}$

$= 0$ if $x = 0, \pm \sqrt{\frac{3}{5}}$, the x-values for $n=3$

And so on.

It is clear that the evaluation points (and the weights for $n>3$) are irrational numbers even though the integration limits are rational. This makes Gaussian quadrature
inconvenient for hand computation, though not for computers, when the evaluation points and the weights can be stored in advance and called when required.
Automatic Quadrature

We have seen that composite trapezoid and Simpson’s rules can be used to produce an automatic quadrature procedure – simply reduce the strip width (usually by half) until the agreement between two successive computations falls below the required tolerance. As an example, consider the evaluation of

\[ \int_{0.2}^{1} \frac{1}{x^2} \, dx \]

using Simpson’s Rule with a tolerance of 0.02:
| No of strips, n | h = \frac{1 - 0.2}{n} | S_n | |S_{n+2} - S_n| |
|----------------|----------------------|-----|------------------|
| 2              | 0.4                  | 4.948148 | 0.761111 |
| 4              | 0.2                  | 4.187037 | 0.162819 |
| 8              | 0.1                  | 4.024218 | 0.022054 |
| 16             | 0.05                 | 4.002164 | 0.002010 |
| 32             | 0.025                | 4.000154 |                 |

The table shows that at \( n = 32 \) we have met the tolerance criterion.
Adaptive Quadrature

The disadvantage of using a composite quadrature rule automatically is that the value of $h$ is the same over the entire interval of integration, whereas the behaviour of the $f(x)$ may not require this. In the diagram below, for example, $h$ can be much larger on the subinterval $[c,b]$ than on the subinterval $[a,c]$ where the graph is rough.
In adaptive integration, the complete interval \([a,b]\) is broken into several subintervals, with different values of \(h\) within each of them. As an example, consider the application of Simpson’s Rule to find

\[
\int_{0.2}^{1} \frac{1}{x^2} \, dx,
\]

with a tolerance level of 0.02.

<table>
<thead>
<tr>
<th>Interval</th>
<th>(h)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.2,1]</td>
<td>(h_1 = 0.4)</td>
<td>(S_1 [0.2,1] = 4.94814815)</td>
</tr>
<tr>
<td>[0.2,0.6]</td>
<td>(h_2 = 0.2)</td>
<td>(S_2 [0.2,0.6] = 3.51851852)</td>
</tr>
<tr>
<td>[0.6,1]</td>
<td>(h_2 = 0.2)</td>
<td>(S_3 [0.6,1] = 0.66851852)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(S_1 [0.2,1] - (S_2 [0.2,0.6] + S_2 [0.6,1]) = 0.76111111 &gt; 0.02)</td>
</tr>
<tr>
<td>[0.6,0.8]</td>
<td>(h_3 = 0.1)</td>
<td>(S_3 [0.6,0.8] = 0.41678477)</td>
</tr>
<tr>
<td>[0.8,1]</td>
<td>(h_3 = 0.1)</td>
<td>(S_3 [0.8,1] = 0.25002572)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(S_2 [0.6,1] - (S_3 [0.6,0.8] + S_3 [0.8,1]) = 0.00170803 &lt; 0.01)</td>
</tr>
<tr>
<td>[0.2,0.4]</td>
<td>(h_3 = 0.1)</td>
<td>(S_3 [0.2,0.4] = 2.52314815)</td>
</tr>
<tr>
<td>[0.4,0.6]</td>
<td>(h_3 = 0.1)</td>
<td>(S_3 [0.4,0.6] = 0.83425926)</td>
</tr>
</tbody>
</table>
Adding gives the final result

\[ S_5[0.2,0.25] + S_5 [0.25,0.3] + S_5 [0.3,0.35] + S_5 [0.35,0.4] + S_4[0.4,0.5] + S_4 [0.5,0.6] + S_3 [0.6,0.8] + S_3 [0.8,1] \]
= 1.00010288 + 0.66669728
  + 0.47620166 + 0.35714758
  + 0.50005144 + 0.33334864
  + 0.41678466 + 0.25002572
  
  = 4.000360

Using adaptive integration to solve this problem requires about one-half the number of function evaluations which we used previously – 17 vs 33.
Romberg Integration

Suppose that $I$ is the exact value of an integral and that $S_n$ is the approximate value of the integral obtained using Simpson’s rule with $n$ strips. Noting that error terms are expressed in powers of $h^{2i}$, $i = 2, 3, \ldots$ we write

$$I = S_n + c_1 h^4 + c_2 h^6 + c_3 h^8 + \ldots$$

Doubling the number of strips i.e. halving $h$

$$I = S_{2n} + c_1 \left(\frac{h}{2}\right)^4 + c_2 \left(\frac{h}{2}\right)^6 + c_3 \left(\frac{h}{2}\right)^8 + \ldots$$
Eliminating the terms in $h^4$ by subtracting the first of these equations from 16 times the second:

$$I = \frac{16S_{2n} - S_n}{15} + k_2 h^6 + k_3 h^8 + \ldots$$

The dominant term in the truncation error is now of order $h^6$, and so an improved approximation for $I$ has been obtained.

This procedure can be extended to more than one level. At the second level,

$$I = \frac{64I\left(\frac{h}{2}\right) - I(h)}{64 - 1}$$

and so on.
If Romberg integration is applied to the trapezoid rule, you need to be able to show that, at level $k$:

$$I = \frac{4^k I\left(\frac{h}{2}\right) - I(h)}{4^k - 1}, \quad k = 1, 2, 3, \ldots$$

Note that, at level 1, the trapezoid rule gives

$$I = \frac{4I\left(\frac{h}{2}\right) - I(h)}{3}$$

$$= \frac{1}{3}\left[4 \frac{h/2}{2}\left(f_0 + 2f_1 + 2f_2 + \ldots + f_{2n}\right) - \frac{h}{2}\left(f_0 + 2f_2 + \ldots + f_{2n}\right)\right]$$
\[ = \frac{h}{3} \left[ (f_0 + 2f_1 + 2f_2 \ldots + f_{2n}) \right. \\
\left. - \frac{1}{2} (f_0 + 2f_2 + \ldots + f_{2n}) \right] \]

\[ = \frac{h}{6} \left[ f_0 + 4f_1 + 2f_2 + \ldots + f_{2n} \right] \]

= Simpson’s Rule for 2n strips!