Numeric methods of integration – Rectangular, Trapezium and Simpson rules

The *anti-differential* of a given function cannot always be determined *analytically*, that is, can be written in terms of basic Mathematical functions such as polynomials, trigomometric, exponentials, logarithms etc. Since the **integral** of a curve $y = f(x)$ (i.e. the area between the curve and the x axis) is the *difference of the anti-differential*, evaluated at the required limits, this means *not all areas under curves can be determined exactly*, *even if the curve itself is known*. This represents a major problem in Applied Mathematics, since the area under a velocity vs time graph is displacement, the area under a force vs displacement graph is work done, area under a force vs time graph is momentum change (i.e. impulse) etc.

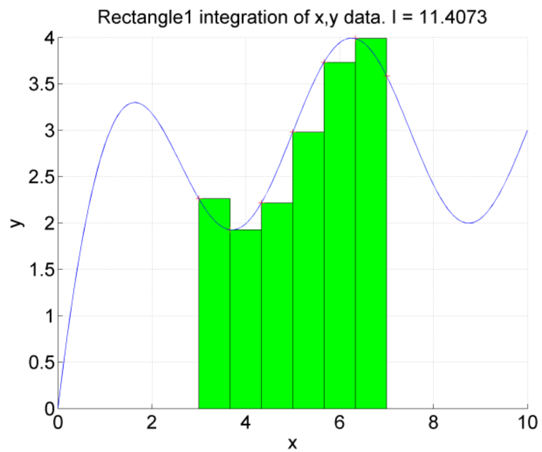
Given a set of x,y data points, **numeric integration methods** can be used to approximate an integral. We shall assume the x coordinates are equi-spaced and there are N +1 coordinate pairs.
i.e. our data set shall be:

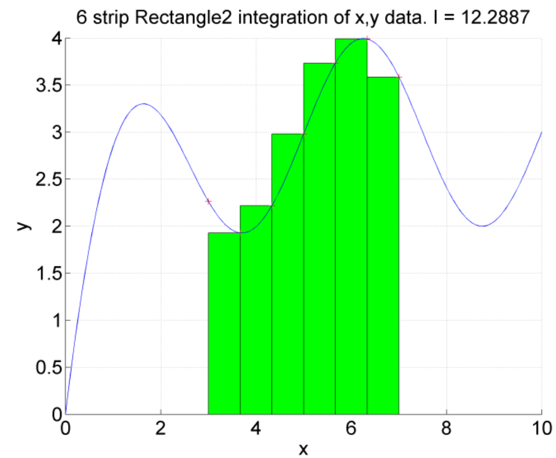
$$(x_0, y_0), (x_1, y_1), ...(x_N, y_N)$$
$$a = x_0, \;\; b = x_1$$

All numerical integration methods suggested here will assume *polynomials* fitted to a sequential set of data points. The key idea is that the polynomial coefficients change as one moves through the coordinate values.

Note in all examples we shall assume a constant x deviation (i.e. 'strip width') of $h = \dfrac{b-a}{N}$

i.e. N strips that cover the integral range of $a \le x \le b$





Rectangular method 1

$$\int_a^b f(x)dx \approx h\left\{ f(a) + f(a+h) + f(a+2h) + ... + f(a+(N-1)h) \right\}$$

$$\int_a^b f(x)dx \approx h\left\{ y_0 + y_1 + ... + y_{N-1} \right\}$$

Rectangular method 2

$$b = a + Nh$$

$$\int_a^b f(x)dx \approx h\left\{ f(a+h) + f(a+2h) + ... + f(a+Nh) \right\}$$

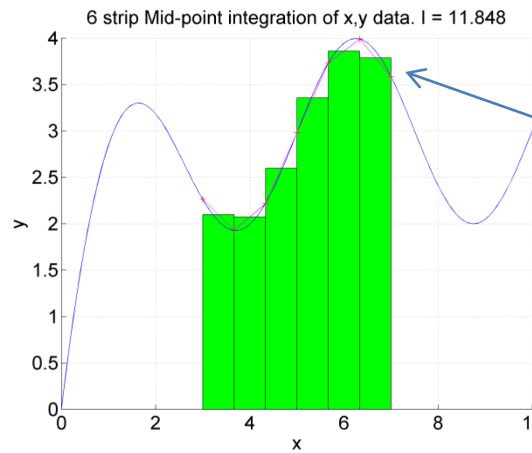
$$\int_a^b f(x)dx \approx h\left\{ y_1 + y_2 + ... + y_N \right\}$$

Mid-point method

$$\int_a^b f(x)dx \approx h\left\{ f(a+\tfrac{1}{2}h) + f(a+\tfrac{3}{2}h) + ... + f(a+\tfrac{1}{2}h+(n-1)h) + ... + f(a+\tfrac{1}{2}h+(N-\tfrac{1}{2})h) \right\}$$

$$\int_a^b f(x)dx \approx h\left\{ \tfrac{1}{2}(y_0+y_1) + \tfrac{1}{2}(y_1+y_2) + ... \tfrac{1}{2}(y_{N-1}+y_N) \right\}$$

$$\int_a^b f(x)dx \approx \tfrac{1}{2}h\left\{ y_0 + 2y_1 + 2y_2 + ... 2y_{N-1} + y_N \right\}$$



The purple lines are straight lines drawn between the data points. In this particular example, this represents rather a crude sampling of the underlying curve.

Note the mid-points are exactly half way along the purple line segments, and not necessarily along the curve between the data points.

As the examples suggest, the *convexity* (essentially the way the gradient changes as x increases) of the function being integrated will dictate which of the first two 'rectangular' methods is an underestimate or overestimate.

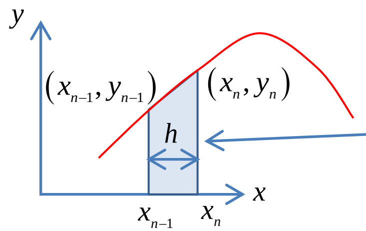
The **Mid-Point method** is often more precise as it is essentially the average of the two. If we don't actually evaluate the function $f(x)$ at the mid-point, but just use the actual data points, we indeed perform the mean average of the two rectangular methods

Trapezium Rule. In this case we assume a straight line between sequential data points, and integrate this exactly. This corresponds to the addition of the areas of trapeziums.

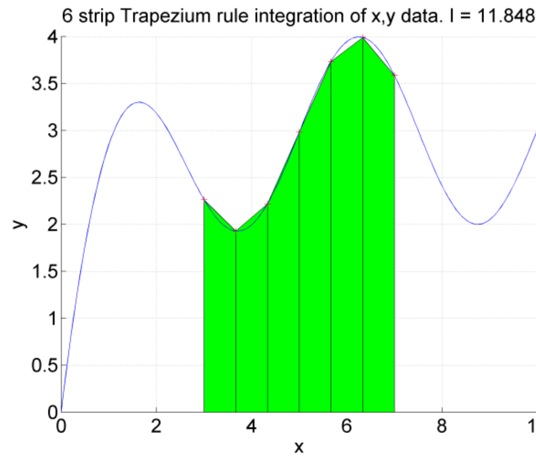
$$\int_a^b f(x)dx \approx \tfrac{1}{2}(y_0+y_1)h + \tfrac{1}{2}(y_1+y_2)h + ... + \tfrac{1}{2}(y_{N-1}+y_N)h$$

$$\boxed{\int_a^b f(x)dx \approx \tfrac{1}{2}h\{y_0+2y_1+2y_2+...2y_{N-1}+y_N\}}$$

i.e. the *same result* as the **Mid-Point rule**



Area of trapezium strip is $\tfrac{1}{2}h(y_{n-1}+y_n)$



$y = \sin\left(\tfrac{2}{5}\pi x\right) - 3e^{-x} + 3$

$$\boxed{\begin{aligned} &x_{n-1} \le x \le x_n \\ &m_n = \frac{y_n - y_{n-1}}{h} \\ &c_n = y_n - m_n x_n \end{aligned}}$$

Between each pair of points is the straight line $y = m_n x + c_n$

Comparing the different methods (6 strips) for the integral in the illustrative curve

Method	Fractional error
Rectangle 1	-3.76%
Rectangle 2	3.67%
Mid-point	-0.0455%
Trapezium	-0.0455%
Simpson	-0.0013%

Simpson's Rule. In this case we assume a **quadratic curve** between *three* sequential data points, and integrate this exactly.

This form is called **Lagrange Polynomial Approximation** which defines a quadratic which will pass through all the points

$$(x_n, y_n), (x_{n+1}, y_{n+1}), (x_{n+2}, y_{n+2})$$

Form of the quadratic curve segments

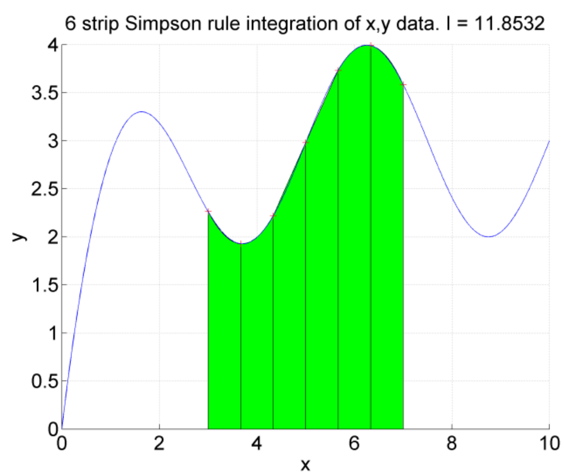
$$x_n \le x \le x_{n+2}$$

$$a_0 = x_n$$

$$a_1 = x_{n+1} = x_n + h$$

$$a_2 = x_{n+2} = x_n + 2h$$

$$y = y_n \frac{(x-a_1)(x-a_2)}{(a_0-a_1)(a_0-a_2)} + y_{n+1}\frac{(x-a_0)(x-a_2)}{(a_1-a_0)(a_1-x_2)} + y_{n+2}\frac{(x-a_0)(x-a_1)}{(a_2-a_0)(a_2-a_1)}$$



To determine the contributions from each quadratic segment we must integrate each piece. This is rather tedious algebraically (!) but leads to the following formula

$$\boxed{\int_a^b f(x)dx \approx \tfrac{1}{3}h\{y_0+4y_1+2y_2+4y_3+2y_4+...4y_{N-1}+y_N\}}$$

Note for Simpson's Rule the number of strips N *must be even*. Note also that Simpson's rule is *exact* if the function is a cubic, quadratic or linear polynomial.

There are a wide variety of higher order polynomial methods. In MATLAB, 'cubic splines' or 'Hermite polynomials' are the built-in options for the `interp1` function, which can be used to fit a *piecewise polynomial* to a set of *x*,*y* data.

These polynomials can be integrated (or differentiated) *exactly* so a curve fit can mean that integration or differentiation operations can be readily applied to a set of data.