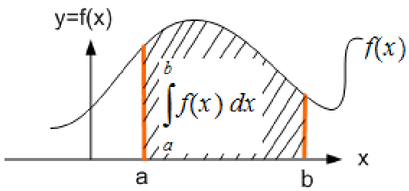
A brief note on numerical integration

In numerical integration we try an approximate solution to a definite integral up to desired precision



$$\mathcal{I} = \int_a^b f(x)\,dx \tag{1}$$

where $f(x)$ is a (piecewise) continuous, well-behaved (smooth) function over the interval $[a, b]$ of our interest. The definite integration is obviously the area under the curve as shown. There can be various reasons for doing integration numerically – (a) $f(x)$ may be known only at certain points, (b) $f(x)$ is known but analytical integration may be too difficult or at times impossible to carry out and (c) the integration can be carried out analytically but numerically it may be far more easier to a given accuracy. The numerical integration is often called as *numerical quadrature* or simply *quadrature* especially when it is applied to one dimensional integration *i.e.* integration over only one variable. At the most basic, the numerical integration amounts to evaluating the integrand at finite set of points (may or may not be equally spaced) bounded by the interval $[, b]$ and doing a weighted sum of these values to approximate the integral.

$$\mathcal{I} = \int_a^b f(x)\,dx \approx \sum_{n=1}^{N} w(x_n)\,f(x_n) \equiv \mathcal{I}_N \tag{2}$$

where $w(x_n)$ is the weight function and N is the number of integration points when the integration limit is sliced up. Choice of N depends on the maximum error $|\mathcal{I} - \mathcal{I}_N| < \epsilon$ we desire. We will discuss more about the errors later.

The expression in (2) follows from approximating a function $f(x)$ with Lagrange interpolating polynomial, often written in the form

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = f(x_0)\cdot l_0(x) + f(x_1)\cdot l_1(x) + \cdots + f(x_N)\cdot l_N(x) \tag{3}$$

where $l_n(x)$ are n-degree polynomials. Integrating the polynomial, we get,

$$\begin{aligned}
\mathcal{I}_N &= \int_a^b p(x)\,dx = \int_a^b \sum_{n=0}^{N} f(x_n)\cdot l_n(x)\,dx \\
&= \sum_{n=0}^{N} \left(f(x_n)\cdot \int_a^b l_n(x)\,dx \right) \equiv \sum_{n=0}^{N} f(x_n)\cdot w(x_n)
\end{aligned} \tag{4}$$

In this course, we will explore the following techniques of numerical integration –

- Midpoint rule
- Trapezoidal rule
- Simpson's rule
- Monte Carlo

in which the Monte Carlo technique is different from the rest in the sense the above discussion of numerical integration is not quite applicable to it. In the first three methods, we approximate the integrand $f(x)$ with a polynomial : $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots$. For Midpoint we only use the constant term a_0, for Trapezoidal the linear term $a_1 x$ and the quadratic term $a_2 x^2$ for Simpson.

Midpoint method

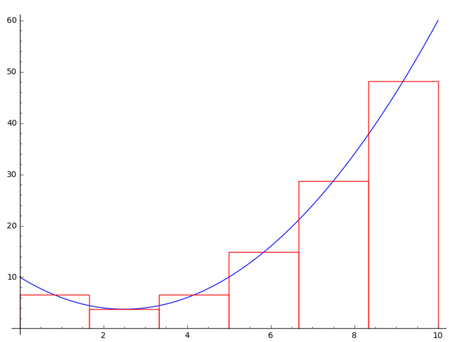
1. Let us divide the integration range $[a,\, b]$ in N equal parts of width h

$$h = (b - a)/N \tag{5}$$

2. Determine the midpoint of each intervals which are the integration points

$$x_1 = \frac{(a) + (a + h)}{2},\ x_2 = \frac{(a + h) + (a + 2h)}{2},\ x_3 = \frac{(a + 2h) + (a + 3h)}{2},\ \ldots \tag{6}$$

Hence, $f(x)$ is evaluated only at x_n and kept constant over the sub-interval and the assumption being the area of each rectangle evaluated at each integration point, $hf(x_n)$, approximates the area under the curve over that sub-interval.



3. The sum of all such rectangular area is

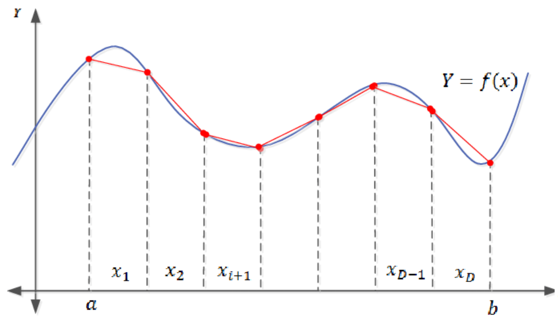
$$\mathcal{M}_N = \sum_{n=1}^{N} h\, f(x_n) \quad \Rightarrow \quad \lim_{N\to\infty} \mathcal{M}_N = \int_a^b f(x)\, dx \tag{7}$$

where the weight function $w(x_n) = w = 1$ or h for all x_n *i.e.* constant.

We will discuss later how to choose N so that the integration in (7) is accurate enough.

Trapezoidal rule

The trapezoidal rule for estimating definite integrals uses straight lines connecting consecutive $f(x_n)$ to generate trapezoids whose areas are expected to better approximate the area under the curve over each interval.



1. As before in (5), we have N intervals each of width h. This h forms the width of each trapezoid.

2. Let the endpoints of each interval be at x_0, x_1, x_2, $\ldots$ x_N where

$$x_0 = a,\ x_1 = x_0 + h,\ x_2 = x_0 + 2h,\ \ldots\ x_n = x_0 + Nh = b \tag{8}$$

3. Evaluate the $f(x_n)$ and calculate the area of each trapezoid,

$$\mathcal{T}_n = \frac{h}{2}\Big(f(x_{n-1}) + f(x_n)\Big) \tag{9}$$

4. Sum over all the $\mathcal{T}_n$'s to approximate the integral

$$\begin{aligned}
\mathcal{T}_N &= \sum_{n=1}^{N} \mathcal{T}_n = \frac{h}{2}\Big([f(x_0) + f(x_1)] + [f(x_1) + f(x_2)] + \cdots + [f(x_{N-1}) + f(x_N)]\Big) \\
&= \frac{h}{2}\Big(f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{N-1}) + f(x_N)\Big) \qquad (10) \\
&= \sum_{n=1}^{N} w(x_n)\, f(x_n) \ \Rightarrow\ \lim_{N\to\infty} \mathcal{T}_N = \int_a^b f(x)\, dx
\end{aligned}$$

where $w(x_0) = w(x_N) = 1$ or $h/2$ and $w(x_1) = w(x_2) = \cdots = w(x_{N-1}) = 2$ or h. Once again, we will come back to the accuracy question later. But contrary to the expectation, trapezoidal rule tends to be less accurate than the midpoint rule, particularly when the curve is strictly concave or convex over the integration limits.

Simpson's rule

In Midpoint method the area under the curve is estimated by rectangles *i.e.* piecewise constant functions. In Trapezoidal rule the area is estimated by trapeziums *i.e.* piecewise linear functions. In the Simpson's rule we will approximate the curves in an interval with a quadratic functions $\propto x^2$. We, therefore, need three points to describe a quadratic curve – the obvious two are the (sub) interval boundaries and the other is taken to be the average of these two *i.e.* the midpoint.

$$\text{For } \int_{x_0}^{x_2} f(x)\,dx \text{ we need } (x_0,\, f(x_0)),\, (x_1,\, f(x_1)) \text{ and } (x_2,\, f(x_2)) \tag{11}$$

where $x_1 = (x_0 + x_2)/2$. The calculation of the integral above in (11) proceeds as,

$$\begin{aligned}
\int_{x_0}^{x_2} f(x)\,dx &\approx \int_{x_0}^{x_2} \Big(a_2 x^2 + a_1 x + a_0\Big)^2 dx \\
&= \Big(\frac{a_2}{3}x^3 + \frac{a_1}{2}x^2 + a_0 x\Big)\Big|_{x_0}^{x_2} \\
&= \frac{a_2}{3}\left(x_2^3 - x_0^3\right) + \frac{a_1}{2}\left(x_2^2 - x_0^2\right) + a_0\left(x_2 - x_0\right) \\
&= \frac{x_2 - x_0}{6}\Big(2a_2\left(x_2^2 + x_2 x_0 + x_0^2\right) + 3a_1\left(x_2 + x_0\right) + 6a_0\Big) \qquad (12)
\end{aligned}$$

Now, let us take $h = (x_2 - x_0)/2$ and rearranging the terms using $x_1 = (x_2 + x_0)/2$ in (12) we get

$$\begin{aligned}
\int_{x_0}^{x_2} f(x)\,dx &\approx \frac{h}{3}\Big(\left(a_2 x_2^2 + a_1 x_2 + a_0\right) + \left(a_2 x_0^2 + a_1 x_0 + a_0\right) + a_2\left(x_2^2 + 2x_2 x_0 + x_0^2\right) \\
&\quad +2a_1\left(x_2 + x_0\right) + 4a_0\Big) \\
&= \frac{h}{3}\Big(f(x_2) + f(x_0) + a_2\left(2x_1\right)^2 + 2a_1\left(2x_1\right) + 4a_0\Big) \\
&= \frac{h}{3}\Big(f(x_2) + f(x_0) + 4(a_2 x_1^2 + a_1 x_1 + a_0)\Big) \\
&= \frac{h}{3}\Big(f(x_0) + 4f(x_1) + f(x_2)\Big) \qquad (13)
\end{aligned}$$

Similarly for $\int_{x_2}^{x_4} f(x)dx = h(f(x_2) + 4f(x_3) + f(x_4))$ and so on. Therefore, in the final step we get the answer from Simpson's rule as,

$$\begin{aligned}
\mathcal{S}_N &= \frac{h}{3}\Big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)\Big) \qquad (14) \\
&= \sum_{n=1}^{N} w(x_n)\, f(x_n) \;\Rightarrow\; \lim_{N\to\infty} \mathcal{S}_N = \int_a^b f(x)\,dx
\end{aligned}$$

where, the weight functions are $w(x_0) = w(x_N) = 1$, for odd i the $w(x_n) = 4$ and for the even n the $w(x_i) = 2$. It is interesting to note that

$$\mathcal{S}_{2N} = \frac{2}{3}\,\mathcal{M}_N + \frac{1}{3}\,\mathcal{T}_N \tag{15}$$

Let us now address the problem of accuracy of the above numerical integrations. If we already know the actual answer of the integration $\mathcal{I}$ in (1), then the absolute and relative errors obviously are

$$\text{Abosulte error: } |\mathcal{I}_N - I| \quad \text{and Relative error: } \left|\frac{\mathcal{I}_N - I}{I}\right| \times 100\% \tag{16}$$

where $\mathcal{I}_N \in \mathcal{M}_N,\, \mathcal{T}_N,\, \mathcal{S}_N$. But often, the very reason for doing numerical integration is that we do not have or cannot calculate an $\mathcal{I}$, and in that case all we can calculate is the **upper bound** of the error that each method will yield. If $f(x)$ is a continuous function over $[a,\, b]$, having a second derivative $f''(x)$ for Midpoint and Trapezoidal and fourth derivative $f''''(x)$ for Simpson over this interval, then the estimated upper bounds for the error in using numerical integration schemes to estimate $\int_a^b f(x)\,dx$ are

$$\begin{aligned} \text{Midpoint :}\quad \text{Error in } M_N &\leq \frac{(b-a)^3}{24N^2}\,|f''(x)|_{\max} && (17)\\ \text{Trapezoidal :}\quad \text{Error in } T_N &\leq \frac{(b-a)^3}{12N^2}\,|f''(x)|_{\max} && (18)\\ \text{Simpson :}\quad \text{Error in } S_N &\leq \frac{(b-a)^5}{180N^4}\,|f''''(x)|_{\max} && (19) \end{aligned}$$

The requirement of upper bound for the error determines N. Two things to keep in mind – (i) N chosen should be the smallest integer value greater than or equal to N (basically `ceil(N)`), (ii) the actual estimate may, in fact, be much better approximation than is indicated by the error upper bound. It may sound strange, but it is often true, that taking N larger than what you obtain from the error bounds can actually detoriate the accuracy, contrary to what the limit $N \to \infty$ suggests.

Example : Let us estimate numerically the following integral whose exact analytical answer is available,

$$\int_0^1 x^2\,dx = \frac{1}{3} = 0.333 \tag{20}$$

Without worrying about error upper bound, let us divide up the integration limit $[0, 1]$ in $N = 4$ intervals – $[0, 1/4]$, $[1/4, 2/4]$, $[2/4, 3/4]$ and $[3/4, 1]$, the interval length being $h = 1/4$. For Midpoint method, we need the midpoints of these subintervals:

1/8, 3/8, 5/8, 7/8. Hence, from (7),

$$\begin{aligned} M_4 &= \frac{1}{4}\left[f\left(\frac{1}{8}\right)+f\left(\frac{3}{8}\right)+f\left(\frac{5}{8}\right)+f\left(\frac{1}{8}\right)\right] \\ &= \frac{1}{4}\left[\left(\frac{1}{8}\right)^2+\left(\frac{3}{8}\right)^2+\left(\frac{5}{8}\right)^2+\left(\frac{7}{8}\right)^2\right]=\frac{21}{64}=0.328 \end{aligned} \tag{21}$$

The absolute error is $|0.333-0.328|\approx 0.0052$ and relative error is 1.56%. Let us now make use of (17) to determine N, first the $|f^{"}(x)|_{\max}$,

$$f(x)=x^2 \;\Rightarrow\; f^{'}(x)=2x \;\Rightarrow\; f^{"}(x)=2 \;\Rightarrow\; |f^{"}(x)|_{\max}=2 \tag{22}$$

So, N required for maximum error bound of 0.001 is

$$0.001=\frac{(1-0)^3}{24N^2}2 \;\Rightarrow\; N=9. \tag{23}$$

Funnily, Trapezoidal method for $N=4$ yields 0.3437 and thus the absolute and relative errors are 0.104 and 3.12% respectively. And for maximum error bound of 0.001, Trapezoidal rule requires $N=13$.

Monte Carlo integration

Monte Carlo method is a class of computational algorithm that involves repeated random sampling to estimate an integration numerically. It is a widely used method to estimate particularly higher dimensional integrals. The Monte Carlo integration starts with choosing random numbers X_i with a *probability distribution function* (PDF) $p(x)$. Suppose the *domain* of X is discrete, as in tossing of coin $X \in \{h,t\}$ or rolling of a dice $X \in \{1,2,3,4,5,6\}$. Then the $p(x)$ gives the probability or relative frequency with which a particular X occurs, $p(x)=\text{Prob}\,(X=x)$. For a continuous domain, however, we consider $p(x)\,dx$ to be the probability for X to assume any value within an interval dx around x ($[x \pm dx]$). Two important properties of PDF $p(x)$ are

$$0 \le p(x) \le 1 \quad \text{and} \quad \sum_{x_i\in D} p(x_i)=1 \;\text{ or }\; \int_D p(x)\,dx=1 \tag{24}$$

where D is the domain of x. As an example, consider $p(x)=$ constant *i.e.* uniformly distributed over a domain $D=[a,b]$

$$\begin{aligned} \int_a^b p(x)\,dx &= \int_a^b \mathcal{C}\,dx = 1 \\ \Rightarrow \quad p(x) &= \mathcal{C} = \frac{1}{b-a} \end{aligned} \tag{25}$$

Another widely used PDF is Gaussian distribution $\mathcal{N}(\mu, \sigma)$,

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{26}$$

Suppose $f(x)$ is a function on the, say discrete, domain of X whose PDF is $p(x)$ and evaluated over M random X, then

$$\langle f \rangle = \frac{1}{M}\sum_{i=1}^{M} f(x_i)\, p(x_i) \tag{27}$$

$$\sigma_f^2 = \frac{1}{M}\sum_{i=1}^{M} \Big(f(x_i) - \langle f \rangle\Big)^2 p(x_i) = \langle f^2 \rangle - \langle f \rangle^2 \tag{28}$$

Suppose we assemble N such independent $\langle f \rangle$ and their corresponding σ_f, then the *global* average and variance will be

$$\langle\langle f \rangle\rangle = \frac{1}{N}\sum_{i=1}^{N} \langle f \rangle_i \quad \text{and} \quad \sigma_N^2 = \frac{\sigma_f^2}{N} \quad \Rightarrow \quad \sigma_N \sim \frac{1}{\sqrt{N}} \tag{29}$$

The error on the measurement of f thus decreases as $1/\sqrt{N}$. So if we want Monte Carlo estimate of $\int_a^b f(x)dx$, then the method is certainly at disadvantage when compared to, say, Trapezoidal or Simpson where errors fall as $1/N^2$ and $1/N^4$ respectively. But it is true for one or fewer dimensions, as we move to higher dimensions (*i.e.* large number of variables) Monte Carlo becomes significantly efficient.

In Monte Carlo integration, we wish to estimate the integral $\int_a^b f(x)dx$ (remember, in MC we cannot calculate but estimate) and for this we define an *estimator*. Given a random variable X drawn from a PDF $p(x)$, then the estimator is defined as

$$\mathcal{F}_N \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \tag{30}$$

Then the average of the estimator $\mathcal{F}_N$ is

$$\langle \mathcal{F}_N \rangle = \int_a^b \mathcal{F}_N\, p(x)\, dx = \int_a^b f(x)\, dx \tag{31}$$

with the variance as in (28). For our particular case of uniform PDF in $[a, b]$,

$$\mathcal{F}_N = \frac{b-a}{N} \sum_{i=1}^{N} f(X_i) \tag{32}$$

The above expression (32) looks very similar to Midpoint expression (7). Therefore, the steps involve in Monte Carlo integration methods are,

1. Choose a N, say 10 or 20 or 50 or whatever.
2. Draw N number of random variables X_i from its domain $[a, b]$. Usually the in-built random numbers in any language return uniform random numbers in the range $[0, 1]$. To convert it to $[a, b]$ one may use

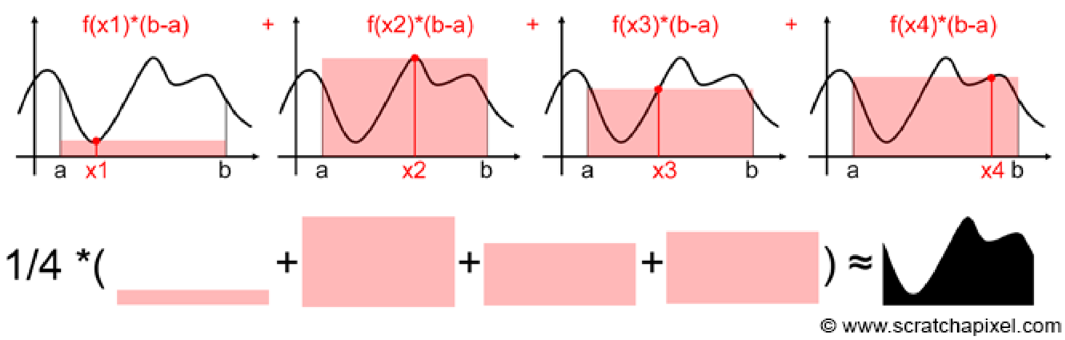
$$X = a + (b - a)\xi \quad \text{where,} \quad \xi \in [0, 1]$$

3. For each X_i calculate $f(X_i)$ and determine $\mathcal{F}_N$ using eqn. (32) and σ_f using

$$\sigma_f^2 = \frac{1}{N}\sum_{i=1}^{N} f(X_i)^2 - \left(\frac{1}{N}\sum_{i=1}^{N} f(X_i)\right)^2 \tag{33}$$

4. Either tabulate or plot $\mathcal{F}_N$ versus N. Also keep track of σ_f for each N.
5. Go to step (1), increase N by 10 or whatever times and repeat the above cycle.

Schematically the monte carlo integration looks very much like the following figure

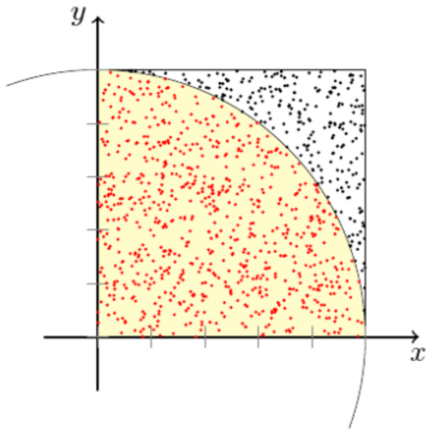


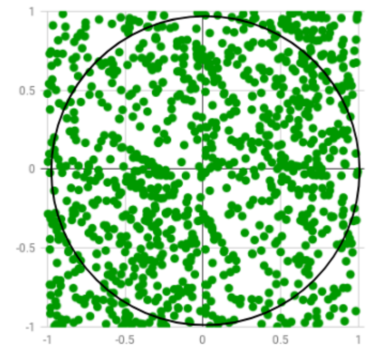
where we chose $N = 4$ random numbers $\{x_1, x_2, x_3, x_4\}$ and calculated the area $(b - a) * f(x_i)$ to estimate $\int_a^b f(x)\,dx$. As N becomes larger and larger, you will find $\mathcal{F}_N$ converging to a value but decrease in error or σ_f will be rather slow ($\sim 1/\sqrt{N}$).

A popular way to demonstrate the application of monte carlo is to evaluate the integral

$$\int_0^1 \sqrt{1 - x^2}\,dx = \frac{\pi}{4} \tag{34}$$

which return an estimate for π. The method is straight forward – generate uniformly distributed random number $\in [0, 1]$ *i.e.* within a unit square and count how many of them falls within quarter circle. The situation is shown in the left-hand figure below. From the four times the ratio of those inside and outside gives the estimate of π.





One can also use a full unit circle in a unit square, then the radius of the circle is 1/2 and area is $\pi/4$. But in this case we need random numbers for x and y coordinates $\in [-0.5, 0.5]$ and check for $x^2 + y^2 \leq 1$. This situation is depicted in the right-hand figure above.

Pseudo-random number generation