Adaptive Quadrature
CPL project

Date of submission: 21. 11. 2021

Ritadip Bharati

Roll no: 1911129

Brief Note on Numerical methods of Integration:

In numerical integration,we try to get approximate solution of a definite integral up to a desired precision.

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

where if f(x) is a piece-wise continuous and smooth function over the interval given. the solution of the integration is the area under the curve for the given range.
The numerical integration technique is often called as numerical quadrature or simply quadrature when we do integration over a function of one variable.
To do this we we divide the bounded range or [a,b] and multiply them with some weight function and sum them up.

$$I_N = \int_a^b p(x)dx = \int_a^b \sum_{n=0}^{N} f(x_n) \cdot l_n(x)dx = \sum_{n=0}^{N} \left(f(x_n) \cdot \int_a^b L_n(x)dx \right) \equiv \sum_{n=0}^{N} f(x_n) \cdot w(x_n) \tag{2}$$

The choice of number of divisions will depend on the maximum error we Desire.
The methods of numerical integration are:
a) Midpoint rule
b) Trapezoidal rule and,
c) Simpson rule

Midpoint rule:

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

$$h = (b-a)/N \tag{3}$$

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

$$x_1 = \frac{(a)+(a+h)}{2}, x_2 = \frac{(a+h)+(a+2h)}{2}, x_1 = \frac{(a+2h)+(a+3h)}{2}, \dots \tag{4}$$

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

$$M_N = \sum_{n=1}^{N} hf(x_n) \Rightarrow \lim_{N\to\infty} M_N = \int_a^b f(x)dx \tag{5}$$

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

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, where x_n's are the boundaries of the intervals.

1 As before in (3), 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 $x_0, x_1, x_2, \dots x_N$ where

$$x_0 = a, x_1 = x_0 + h, x_2 = x_0 + 2h, \dots x_n = x_0 + Nh = b \tag{6}$$

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

$$T_n = \frac{h}{2}\left(f(x_{n-1}) + f(x_n)\right) \tag{7}$$

4 Sum over all the T_n 's to approximate the integral

$$\begin{aligned} T_N = \sum_{n=1}^{N} T_n &= \frac{h}{2}\left([f(x_0)+f(x_1)] + [f(x_1)+f(x_2)] + \cdots + [f(x_{N-1})+f(x_N)]\right) \\ &= \frac{h}{2}\left(f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{N-1}) + f(x_N)\right) \\ &= \sum_{n=0}^{N} w(x_n) f(x_n) \Rightarrow \lim_{N\to\infty} T_N = \int_a^b f(x)dx \end{aligned} \tag{8}$$

So, $w(x_n) = \frac{h}{2}$ for $n = 0, N$ and $w(x_n) = h$ for all other values of n.

Simpson's rule:

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. Let they be $x_0, x_1 and x_2$, where $x_1 = (x_0 + x_2)/2$.

1 Here we divide [a,b] range in N/2 divisions and get $x_0, x_2, x_4 \dots$ and find $x_1, x_3, \dots$ with previous relation. we take step as, $h = (x_2 - x_0)/2$.

$$\begin{aligned} \int_{x_0}^{x_2} f(x)dx \approx &\frac{h}{3}\left(\left(a_2x_2^2 + a_1x_2 + a_0\right) + \left(a_2x_0^2 + a_1x_0 + a_0\right) + a_2\left(x_2^2 + 2x_1x_0 + x_0^2\right)\right. \\ &\left. + 2a_1(x_2 + x_0) + 4a_0\right) \\ = &\frac{h}{3}\Big(f(x_2) + f(x_0) + a_2(2x_1)^2 + 2a_1(2x_1) + 4a_0\Big) \\ = &\frac{h}{3}\left(f(x_2) + f(x_0) + 4\left(a_2x_1^2 + a_1x_1 + a_0\right)\right) \\ = &\frac{h}{3}\left(f(x_0) + 4f(x_1) + f(x_2)\right) \end{aligned} \tag{9}$$

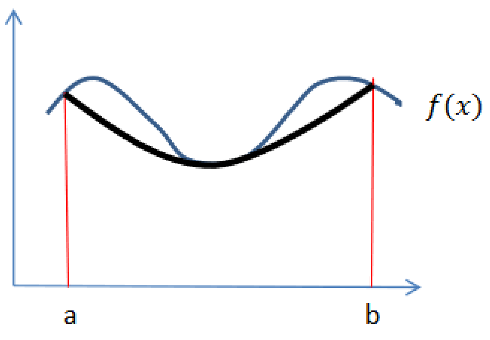
2 Summing up the results of integration in the ranges $(x_0, x_2), (x_2, x_4), (x_4, x_6) \dots$ will give,

$$\begin{aligned} S_N &= \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)\right) \\ &= \sum_{n=1}^{N} w(x_n) f(x_n) \Rightarrow \lim_{N\to\infty} S_N = \int_a^b f(x)dx \end{aligned} \tag{10}$$

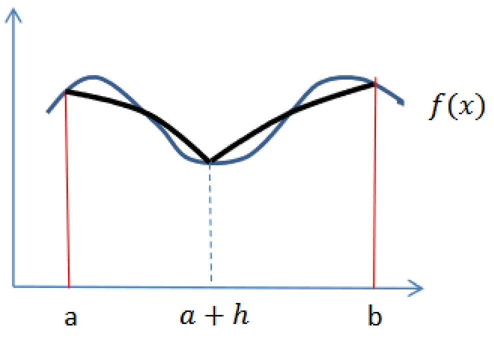
3 Thus, $w(x_n) = h/3$ for $n = 1 and N$, $w(x_n) = 2h/3$ for $n =$ odd numbers except 1 and N, and $w(x_n) = 4h/3$ for $n =$ even numbers.

Note on Adaptive quadrature:

Adaptive quadrature is the one-dimensional integration technique where the step size is adaptive or determined according to the curve. We know that as $N \to \infty$ in equation 2, the summation converges to the integral. So, we divide the section we are integrating into further 2 parts, so, N doubled. We do this until the difference between N and 2N division converges to a tolerance limit.



(a) Step 1



(b) Step 2

Figure 1: Adaptive quadrature

Here, we do the following:

1 Here, we start the number of divisions from 2,4 or 8 basically a small number, say s. Now, we numerically integrate the whole range [a,b] by dividing it into s section.Then we divide the range [a,b] into two sections, lets say [a,m] and [m,b] and integrate them separately, and add them. In this process we actually divide the range [a,b] in 2s sections.

$$\int_a^b f(x)dx = \int_m^b f(x)dx + \int_a^m f(x)dx \tag{11}$$

2 Then we compare the results of integration with s sections and 2s sections. If the difference is more than tolerance, we further divide the first one of the pairs, i.e, [a,m] into two parts, and the procedure continues.

3 When the difference of them is less than our desired tolerance, we add that part in our integral.

4 As an example, at one step we compare the results of integration in the range [c,d] and sum of the integrations in the range [c,(c+d)/2] and [(c+d)/2,d] and the difference is less than tolerance.Then we add one of these in the integration variable and go to similar treatment to next range, like [d,e].

So, pseudo-code of the algorithm is:

```
def adaptive_quadrature(function,a,b,tolerance,Iteration_limit):
    stack=[a,b]
    Integral=0
    Iteration=0
    while length of the stack is not 0:
        new_b=stack.pop()
        new_a=stack.pop()
        Integral1=Integration(function,new_a,new_b,4)
        m=(new_a+new_b)/2
        Integral2=Integration(function,new_a,m,4)+Integration(function,m,new_b,4)
        Iteration+=1
```

```
                if abs(Integral1-Integral2)/15<(new_b-new_a)*tolerance:
                    Integral+=Integral2
                else:
                    stack.append(m)
                    stack.append(new_b)
                    stack.append(new_a)
                    stack.append(m)
                end if
            if Iteration>Iteration_limit:
                break
            end if
        end while loop
        return(Iteration)
```

First of all, the list "stack" contains the reverse series of boundaries over which the integration will be done in a order of lower bound, upper bound. So, at first we have [a,b] as the stack. So, as long as there is any element in the stack, we have to integrate and compare, i.e. run the while loop.
During the while loop, we store the last two elements of "stack" in new_a and new_b variable, which are basically lower and upper bound of the section to be integrated.
Now, if the difference between the integral is less than our tolerance limit, then we add one of the integral to our output integral variable. along with that, the length of the stack decreases by 2 due to the pop statements and not any new append.
But if the difference is greater than tolerance, then we append the new boundaries over which further integrations will occur. Here, after the next step, the stack will look like [m,b,a,m]. So, the next loop will run to integrate the function in the range [a,m], and similar treatment will continue.

Problem statement:

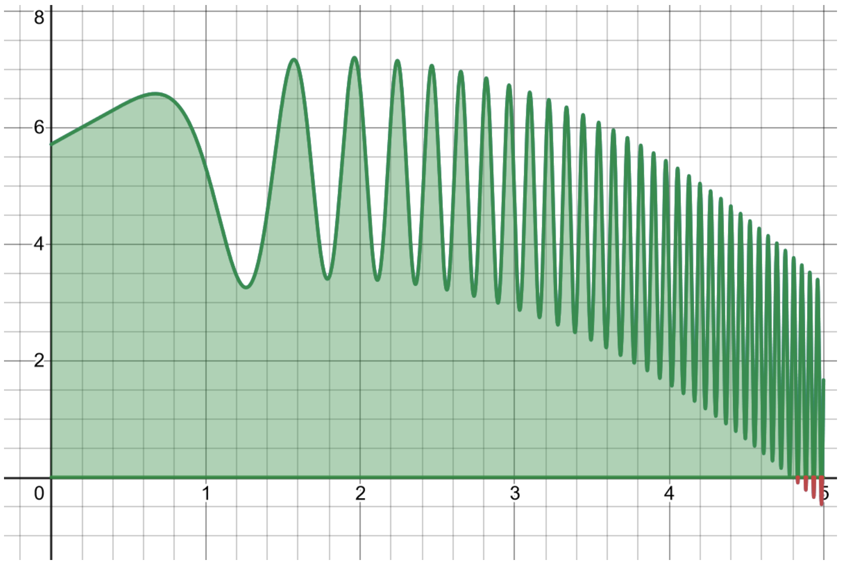


Figure 2: y vs x; the integral is the area under the curve i.e. (green area-red area)

Let, $y = 1.9\cos(1.7x^3 - 0.3) - 0.4x^2 + 1.5x + 3.9$. This function is quite smooth in the range of 0 to 1.2, however, it changes very frequently in the range 1.2 to 5.
Calculate, $I = \int_0^5 ydx$
Answer: The Numerical Quadrature methods gives:

Midpoint: 22.954788427168403
Trapezoidal: 22.95524359275345
Simpson: 22.94949978251816

Adaptive Quadrature methods gives:
Midpoint: 22.954792848503683
Trapezoidal: 22.954592003256842
Simpson: 22.954790934972888
We can see, the results from adaptive quadrature are more accurate, i.e. closer to 22.95479088366101. Notably, Here, the number of divisions in numerical quadrature method is kept 1000.