

MPHYCC-09: M.Sc. Physics (Practical)

By

Dr. Santosh Prasad Gupta

Assistant Professor

Department of Physics

Patna University, Patna

Methods for definite Integration

Riemann Sums:

Let $f(x)$ be a continuous and non-negative function defined on the closed interval $[a,b]$. How to find the area of the region S bounded by the curve $y=f(x)$, the x -axis, and the vertical lines $x=a$ and $x=b$.

We can approximate this area by dividing the region into rectangles.

Suppose that the interval $[a,b]$ is divided into n subintervals defined by the points

$$a=x_0 < x_1 < x_2 < \dots < x_i < \dots < x_n = b$$

Then the n subintervals are $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$.

This subdivision is called the partition of $[a,b]$ and is denoted by P . The width of the i th subinterval $[x_{i-1}, x_i]$ is given by $\Delta x_i = x_i - x_{i-1}$, so the subintervals generally may have different widths. The width of the largest subinterval is called the norm of the partition P and is denoted $\|P\|$. Thus

$$\|P\| = \max\{\Delta x_1, \Delta x_2, \dots, \Delta x_n\}.$$

We use the partition P to divide the region S into strips S_1, S_2, \dots, S_n . We then approximate the strips S_i using rectangles R_i and choosing a sample point ξ_i in each subinterval $[x_{i-1}, x_i]$. The point ξ_i can be anywhere in its subinterval. The area of the i th rectangle R_i is given by

$$A_i = f(\xi_i) \Delta x_i,$$

so the area of the region S is approximated by the sum of the areas of the so the area of the region S is approximated by the sum of the areas of the rectangles R_i :

$$A \approx \sum A_i = \sum f(\xi_i) \Delta x_i = f(\xi_1) \Delta x_1 + f(\xi_2) \Delta x_2 + \dots + f(\xi_n) \Delta x_n.$$

The sum $\sum f(\xi_i) \Delta x_i$ is called the Riemann Sum, which was introduced by Bernhard Riemann (1826–1866), a German mathematician. There are several types of Riemann Sums. The Left Riemann Sum uses the left endpoints of the subintervals. The Right Riemann Sum uses the right endpoints, and the Midpoint Riemann Sum is calculated using the midpoints of the subintervals.

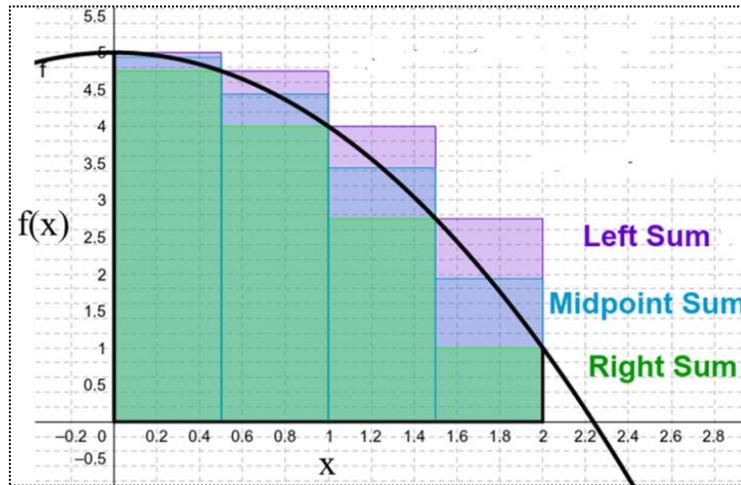


Figure: Riemann Sum, left, right and midpoint sum

The Definite Integral

If we take the limit of the Riemann Sum as the norm of the partition $\|p\|$ approaches zero, we get the exact value of the area A :

$$A = \lim_{\|P\| \rightarrow 0} (\sum f(\xi_i) \Delta x_i)$$

This limit is called the definite integral of the uncton $f(x)$ from a to b and is denoted by

$$\int_a^b f(x) dx.$$

Left Riemann Sums: $\int_a^b f(x) dx \sim h[f(a) + f(a + h) + f(a + 2h) + \dots + f(a + (n - 1)h)]$

Where h is width of the rectangle strip, n is the number of strip such as

$$h = \frac{|b - a|}{n}$$

/*Left Riemann sum to calculate the integral*/

```
#include<stdio.h>
```

```
#include<math.h>
```

/* Define the function */

```
double f(double x){
```

```
    double s;
```

```
    s=x*x*x;
```

```
    return s; /* function is x^3*/
```

```
}
```

```

/*Program begins*/
main(){
    int n,i;
    double a,b,h,x,sum=0,integral;
    /*The parameter which is required to put initially */
    printf("\nEnter the no. of sub-intervals:");
    scanf("%d",&n);
    printf("\nEnter the initial limit:");
    scanf("%lf",&a);
    printf("\nEnter the final limit:");
    scanf("%lf",&b);
    /*begin Riemann rule: */
    h=fabs(b-a)/n;
    for(i=1;i<n;i++){
        x=a+i*h;
        sum=sum+f(x);
    }
    integral=h*(f(a)+sum);
    /*Display the answer */
    printf("\nThe integral is: %lf\n",integral);
}

```

Right Riemann Sums: $\int_a^b f(x)dx \sim h[f(a+h) + f(a+2h) + \dots + f(a+(n-1)h) + f(b)]$

Where h is the width of the rectangle strip, n is the number of strip such as

$$h = \frac{|b-a|}{n}$$

```

/*Right Riemann sum to calculate the integral*/
#include<stdio.h>
#include<math.h>
/* Define the function */
double f(double x){

```

```

double s;
s=x*x*x;
return s; /* function is x^3*/
}

/*Program begins*/
main(){
int n,i;
double a,b,h,x,sum=0,integral;
/*The parameter which is required to put initially */
printf("\nEnter the no. of sub-intervals:");
scanf("%d",&n);
printf("\nEnter the initial limit:");
scanf("%lf",&a);
printf("\nEnter the final limit:");
scanf("%lf",&b);
/*begin Riemann rule: */
h=fabs(b-a)/n;
for(i=1;i<n;i++){
x=a+i*h;
sum=sum+f(x);
}
integral=h*(f(b)+sum);
/*Display the answer */
printf("\nThe integral is: %lf\n",integral);
}

```

Mid-point Riemann Sums: $\int_a^b f(x)dx \sim \frac{h}{2} [f(a) + 2f(a+h) + \dots + 2f(a+(n-1)h) + f(b)]$

Where h is the width of the rectangle strip, n is the number of strip such as

$$h = \frac{|b - a|}{n}$$

```

/*mid-point sum to calculate the integral*/
#include<stdio.h>
#include<math.h>

/* Define the function */
double f(double x){ /* function is defined here*/
    double s;
    s=1+x+x*x+log(x);
    return s;
}

/*Program begins*/
main(){
    int n,i;
    double a,b,h,x,sum=0,integral;
    /*The paramater which is required to put initially */
    printf("\nEnter the no. of sub-intervals:");
    scanf("%d",&n);
    printf("\nEnter the initial limit:");
    scanf("%lf",&a);
    printf("\nEnter the final limit:");
    scanf("%lf",&b);

    /*begin mid point value rule: */
    h=fabs(b-a)/n;
    for(i=1;i<n;i++){
        x=a+i*h;
        sum=sum+f(x);
    }
    integral=h/2*(f(a)+f(b)+2*sum);

    /*Display the answer */
    printf("\nThe integral is: %lf\n",integral);
}

```

Trapezoidal rule: is a technique for approximating the definite integral.

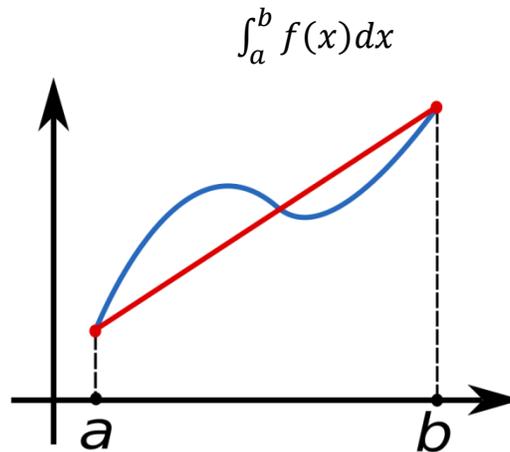


Figure: The function $f(x)$ (in blue) is approximated by a linear function (in red).

The trapezoidal rule works by approximating the region under the graph of the function $f(x)$ as a trapezoid and calculating its area. It follows that

$$\int_a^b f(x) dx \sim \frac{(b-a)}{2} [f(a) + f(b)]$$

The trapezoidal rule may be viewed as the result obtained by averaging the left and right Riemann sums, and is sometimes defined this way. The integral can be even better approximated by partitioning the integration interval, applying the trapezoidal rule to each subinterval, and summing the results. In practice, this "chained" (or "composite") trapezoidal rule is usually what is meant by "integrating with the trapezoidal rule". Suppose that the interval $[a, b]$ is divided into n subintervals defined by the points $a=x_0 < x_1 < x_2 < \dots < x_i < \dots < x_n = b$. Then the n subintervals are $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$. The step size or the size of strip will be

$$h = \frac{|b-a|}{n}$$

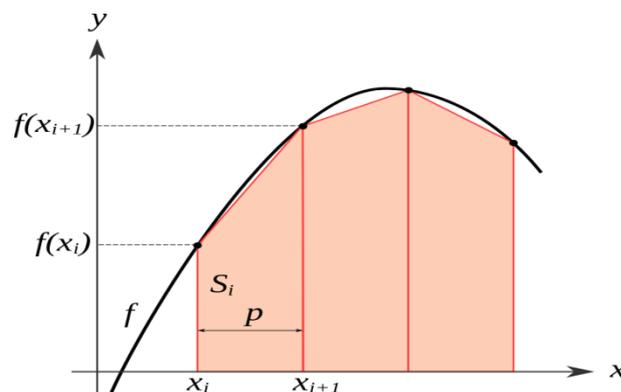


Figure: Illustration of "chained trapezoidal rule" used on an irregularly-spaced partition of $[a, b]$

And the integral $\int_a^b f(x)dx$ can be approximated as follows:

$$\int_a^b f(x)dx \sim \frac{h}{2} [f(a) + 2f(a+h) + \dots + 2f(a+(n-1)h) + f(b)]$$

The approximation becomes more accurate as the resolution of the partition increases (that is, for larger n and step size decreases). When the partition has a regular spacing, as is often the case, the formula can be simplified for calculation efficiency.

```
/*Trapezoidal rule to calculate the integral*/
#include<stdio.h>
#include<math.h>
/* Define the function */
double f(double x){
    double s;
    s=x*x*x;
    return s; /* function is x^3*/
}
/*Program begins*/
main(){
    int n,i;
    double a,b,h,x,sum=0,integral;
    /*The paramater which is required to put initially */
    printf("\nEnter the no. of sub-intervals:");
    scanf("%d",&n);
    printf("\nEnter the initial limit:");
    scanf("%lf",&a);
    printf("\nEnter the final limit:");
    scanf("%lf",&b);
    /*begin trapezoidal rule: */
    h=fabs(b-a)/n;
    for(i=1;i<n;i++){
        x=a+i*h;
        sum=sum+f(x);
    }
    integral=(h/2)*(f(a)+f(b)+2*sum);
    /*Display the answer */
    printf("\nThe integral is: %lf\n",integral);
}
```

Simpson's 1/3 rule: Function is approximated with second order polynomial.

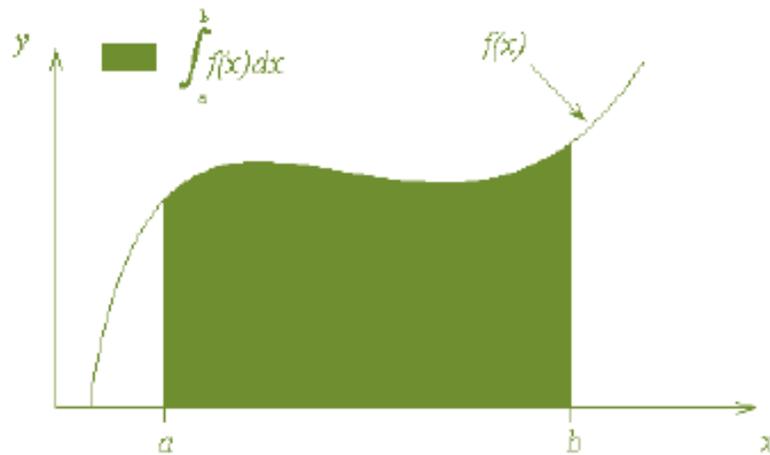


Figure 1 Integration of a function

$$I = \int_a^b f(x) dx \approx \int_a^b f_2(x) dx$$

where $f_2(x)$ is a second order polynomial given by

$$f_2(x) = a_0 + a_1x + a_2x^2$$

Choose

$$(a, f(a)), \left(\frac{a+b}{2}, f\left(\frac{a+b}{2}\right) \right), \text{ and } (b, f(b))$$

as the three points of the function to evaluate a_0 , a_1 and a_2 .

$$f(a) = f_2(a) = a_0 + a_1a + a_2a^2$$

$$f\left(\frac{a+b}{2}\right) = f_2\left(\frac{a+b}{2}\right) = a_0 + a_1\left(\frac{a+b}{2}\right) + a_2\left(\frac{a+b}{2}\right)^2$$

$$f(b) = f_2(b) = a_0 + a_1b + a_2b^2$$

Solving the above three equations for unknowns, a_0 , a_1 and a_2 give

$$a_0 = \frac{a^2 f(b) + abf(b) - 4abf\left(\frac{a+b}{2}\right) + abf(a) + b^2 f(a)}{a^2 - 2ab + b^2}$$

$$a_1 = -\frac{af(a) - 4af\left(\frac{a+b}{2}\right) + 3af(b) + 3bf(a) - 4bf\left(\frac{a+b}{2}\right) + bf(b)}{a^2 - 2ab + b^2}$$

$$a_2 = \frac{2\left(f(a) - 2f\left(\frac{a+b}{2}\right) + f(b)\right)}{a^2 - 2ab + b^2}$$

Then

$$\begin{aligned} I &\approx \int_a^b f_2(x) dx \\ &= \int_a^b (a_0 + a_1x + a_2x^2) dx \\ &= \left[a_0x + a_1 \frac{x^2}{2} + a_2 \frac{x^3}{3} \right]_a^b \\ &= a_0(b-a) + a_1 \frac{b^2 - a^2}{2} + a_2 \frac{b^3 - a^3}{3} \end{aligned}$$

Substituting values of a_0 , a_1 and a_2 give

$$\int_a^b f_2(x) dx = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Since for Simpson 1/3 rule, the interval $[a, b]$ is broken into 2 segments, the segment width

$$h = \frac{b-a}{2}$$

Hence the Simpson's 1/3 rule is given by

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Since the above form has 1/3 in its formula, it is called Simpson's 1/3 rule.

A very good approximation to a definite integral can be found with Simpson's rule. As before, the area under the curve is divided into a number of strips of equal width. However, this time, there must be an even number of strips as they are taken in pairs

In general the integral $\int_a^b f(x) dx$ can be approximated as follows:

$$\begin{aligned} \int_a^b f(x) dx &\sim \frac{h}{3} [f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) \dots \\ &\quad + 4f(a+(n-1)h) + f(b)] \end{aligned}$$

```

/*Simpson's 1/3 rule to calculate the integral*/
#include<stdio.h>
#include<math.h>

/* Define the function to be integrated here: */
double f(double x)
{ double s;
s=log(x)+x*x*x+(1/x);
return s;
}
/*Program begins*/
main(){
int n,i;
double a,b,h,x,sum=0,integral;
/*Ask the user for necessary input */
printf("\nEnter the no. of sub-intervals(EVEN): ");
scanf("%d",&n);
printf("\nEnter the initial limit: ");
scanf("%lf",&a);
printf("\nEnter the final limit: ");
scanf("%lf",&b);
/*Begin Simpson's Procedure: */
h=fabs(b-a)/n;
for(i=1;i<n;i++){
x=a+i*h;
if(i%2==0){
sum=sum+2*f(x);
}
else{
sum=sum+4*f(x);
}
}
integral=(h/3)*(f(a)+f(b)+sum);
/*Print the answer */
printf("\nThe integral is: %lf\n",integral);
}

```

Simpson's 3/8 rule: Simpson's 3/8 rule is another method for numerical integration proposed by Thomas Simpson. It is based upon a cubic interpolation rather than a quadratic interpolation as in case of Simpson's 1/3 rule. Simpson's 3/8 rule is as follows:

$$\int_a^b f(x)dx \sim \frac{3h}{8} [f(a) + 3f(a+h) + 3f(a+2h) + 2f(a+3h) \dots + 3f(a+(n-1)h) + f(b)]$$

```
#include<stdio.h>
#include <math.h>
float y(float x){
float s;
s=1/(1+x*x);
return s;
//function of which integration is to be calculated
}
int main(){
float a,b,h,sum;
int i,n,j;
sum=0;
printf("Enter lower limit a: \n upper limit b:\n number of subintervals n:\n ");
scanf("%f%f%d",&a,&b,&n);
h = (b-a)/n;
sum = y(a)+y(b);
for(i=1;i<n;i++)
{
if(i%3==0){
sum=sum+2*y(a+i*h);
}
else{
sum=sum+3*y(a+i*h);
}
}
printf("Value of integral is %f\n", (3*h/8)*sum);
}
```