NUMERICAL ANALYSIS AND METHODS: TRAPEZOIDAL RULE (SAGEMATH)

Compiled by owolabi odunayo joseph

TRAPEZOIDAL RULE AS A NUMERICAL METHOD

- The Trapezoidal rule is a fundamental numerical integration technique used in numerical analysis to approximate the definite integral of a function.
- It is particularly useful when dealing with functions that are difficult or impossible to integrate analytically.
- The trapezoidal rule is used to evaluate the area under a curve by dividing the total area into smaller trapezoids.
- This method approximates the integral of a function by dividing the area under the curve into a series of trapezoids with equal intervals and summing their areas.
- The more trapezoids you use, the closer the approximation gets to the actual integral value.
- This rule works by using linear approximations of the function.

Given a function f(x) whose graph is shown at the right side with interval (a, b)

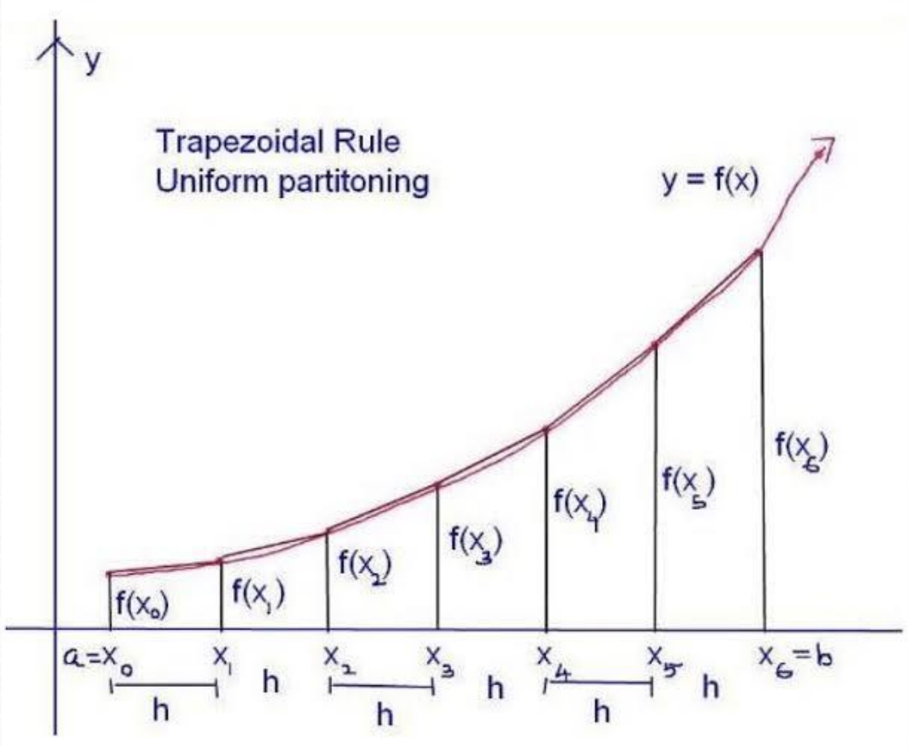
$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + f(x_6) + 2(f(x_1) + f(x_2) + f(x_3) + f(x_4) + f(x_5)]$$

$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + f(x_6)] + h\sum_{i=1}^{5} f(x_i)$$

General solution:

$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + f(x_n)] + h\sum_{i=1}^{n-1} f(x_i)$$

Where $h = \frac{b-a}{n}$; n is the number of intervals



GRAPHICAL VIEW OF THE TRAPEZOIDAL RULE

COMPUTING THE TRAPEZOIDAL RULE IN SAGEMATH

- Certainly! Implementing the trapezoidal rule in sagemath involves creating a python program to approximate definite integrals using the trapezoidal approximation.
- The following below contains an explicit explanation of the sample codes line by line

```python
In [1]: fx = input("Enter f(x): ")
        n = int(input("No of sub-intervals(stripe): "))
        a = eval(input("Enter lower limit: "))
        b = eval(input("Enter upper limit: "))

        Enter f(x): x^3 - 2x^2 + 6x + 10
        No of sub-intervals(stripe): 10
        Enter lower limit: 2
        Enter upper limit: 5
```

The code above gets the

(i) function of x from the user

(ii) Number of intervals the upper and lower limit should be divided into

(iii) The upper limit and the lower limit

```
In [14]: h = (b - a)/n
         xis = [round(a + i*h, 1) for i in range(n+1)]
         print("h(interval between two point) = ", h)
         print()
         print("number of x-coordinates = ", xis)
```

```
h(interval between two point) =  0.3

number of x-coordinates =  [2.0, 2.3, 2.6, 2.9, 3.2, 3.5, 3.8, 4.1,
4.4, 4.7, 5.0]
```

The code above computes

(i) h, which is the interval between each x values

(ii) xis, which is list of x values having an interval h between each values i.e $x_0, x_1, x_2, x_3 \dots . x_n$.

Where $a = x_0$ and $b = x_n$

```
In [15]: def eval_poly(poly, val):
             xs = [x.strip().replace('^', '**') for x in poly.split('+')]
             return sum([eval(n.replace('x', str(val))) for n in xs])
```

The code above evaluates the function of x for every value of xis

i.e

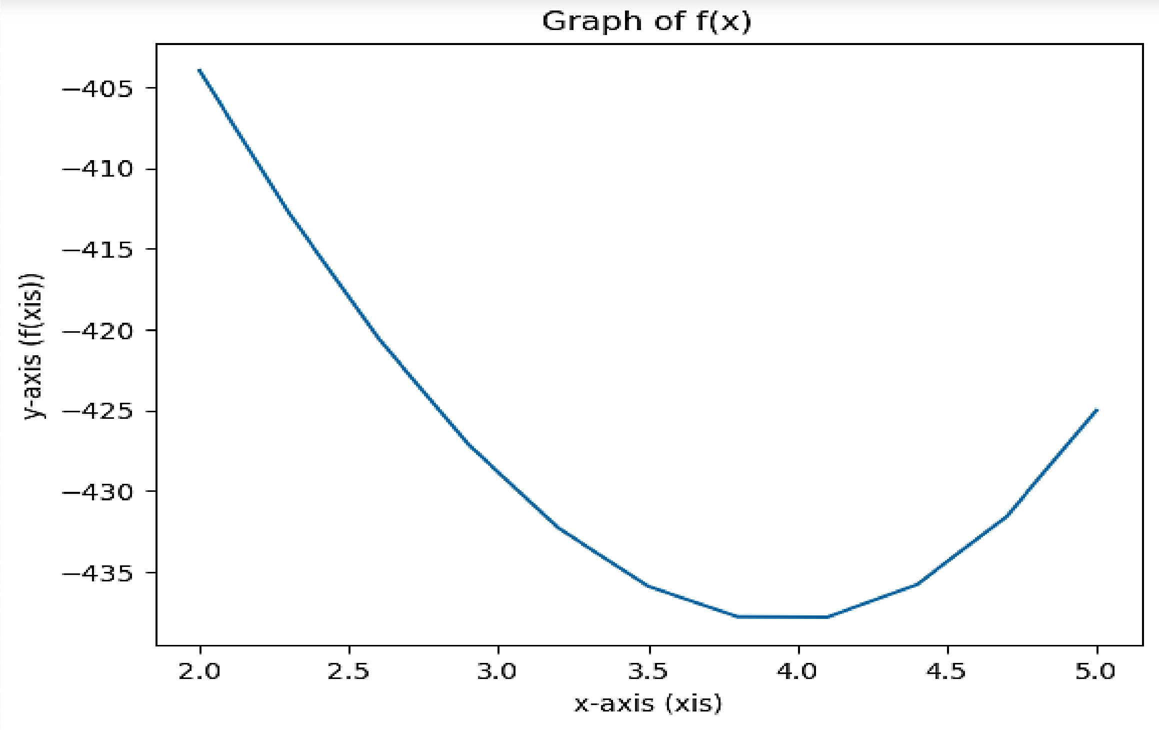
$f(x_0), f(x_1), f(x_2) \dots \dots . f(x_n)$

```
In [31]: def integral_solution():
             xis_left = xis.copy()
             xis_left.remove(a)
             xis_left.remove(b)


             xis_sum = 0
             for i in range(len(xis_left)):
                 xis_sum = 2*(eval_poly(fx, xis_left[i])) + xis_sum

             sol = (h/2)*(eval_poly(fx, xis[0]) + eval_poly(fx, xis[n]) + xis_sum)

             print("The solution of the integral is ", round(sol, 2))

         integral_solution()

         The solution of the integral is  -1285.82
```

The code above solve the trapezoidal formula with the value of h and $f(xis)$

$$i.e\ \frac{h}{2}[f(x_0) + f(x_n)] + h\sum_{i=1}^{n-1} f(x_i)$$

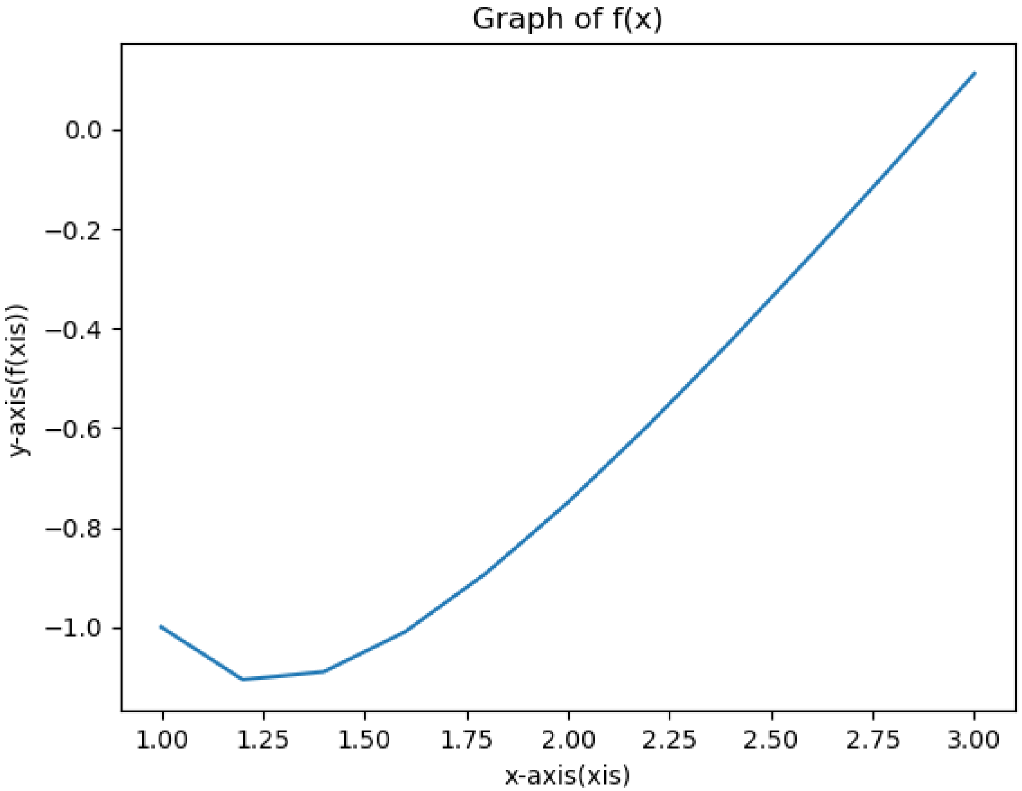
```
In [35]: import matplotlib.pyplot as plt
         y = [eval_poly(fx, x) for x in xis]
         plt.plot(xis, y);
         plt.title("Graph of f(x)");
         plt.xlabel("x-axis (xis)");
         plt.ylabel("y-axis (f(xis))");
         plt.figure(figsize = (3,3));
```



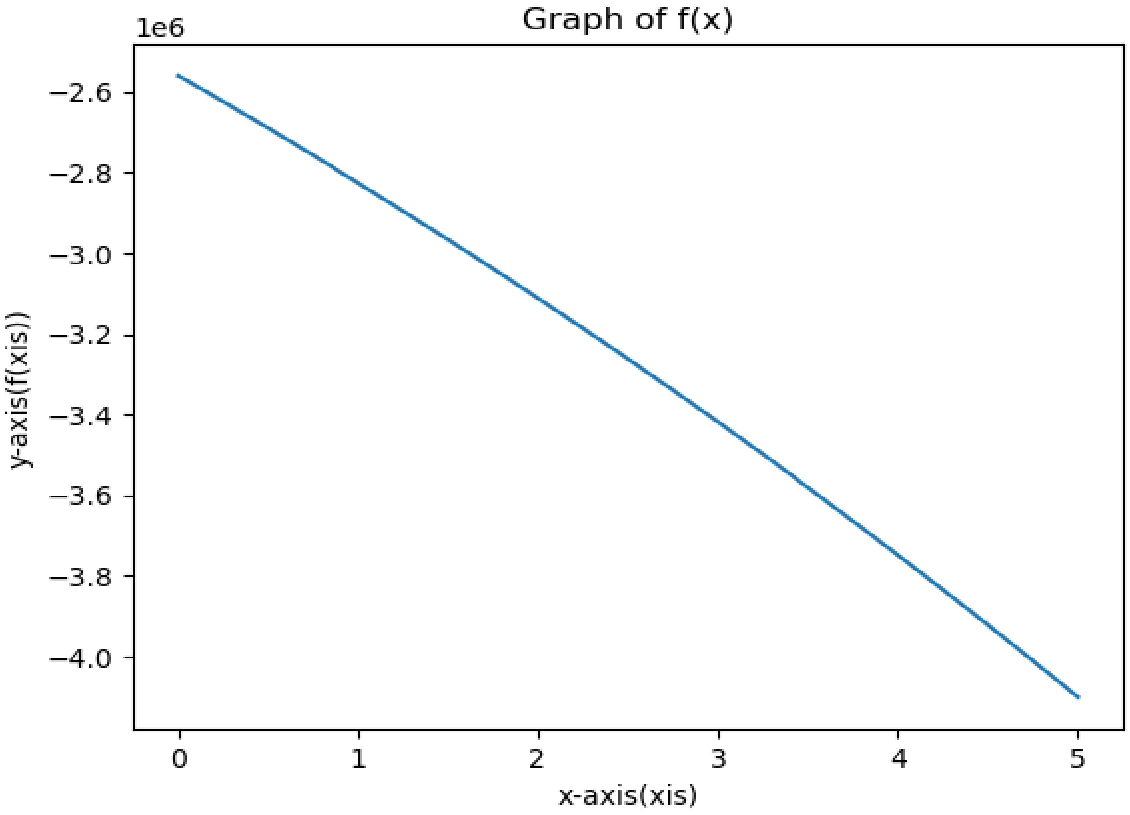
The code above plots the graph of $f(Xis)$ against Xis of the function $f(x)$

Below are some other examples

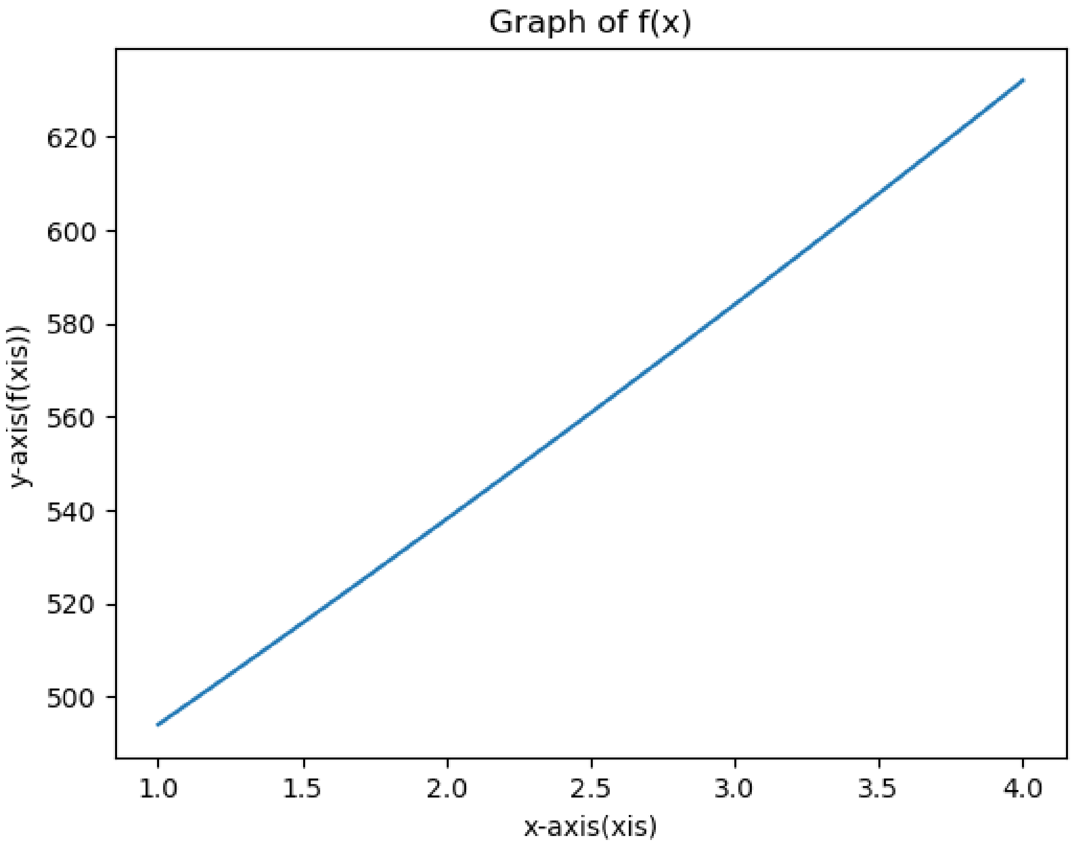
```
Enter f(x): x-3 + x^-2
No of sub-intervals(stripe): 10
Enter lower limit: 1
Enter upper linit: 3

The solution to the integral f(x) is  -1.33
```



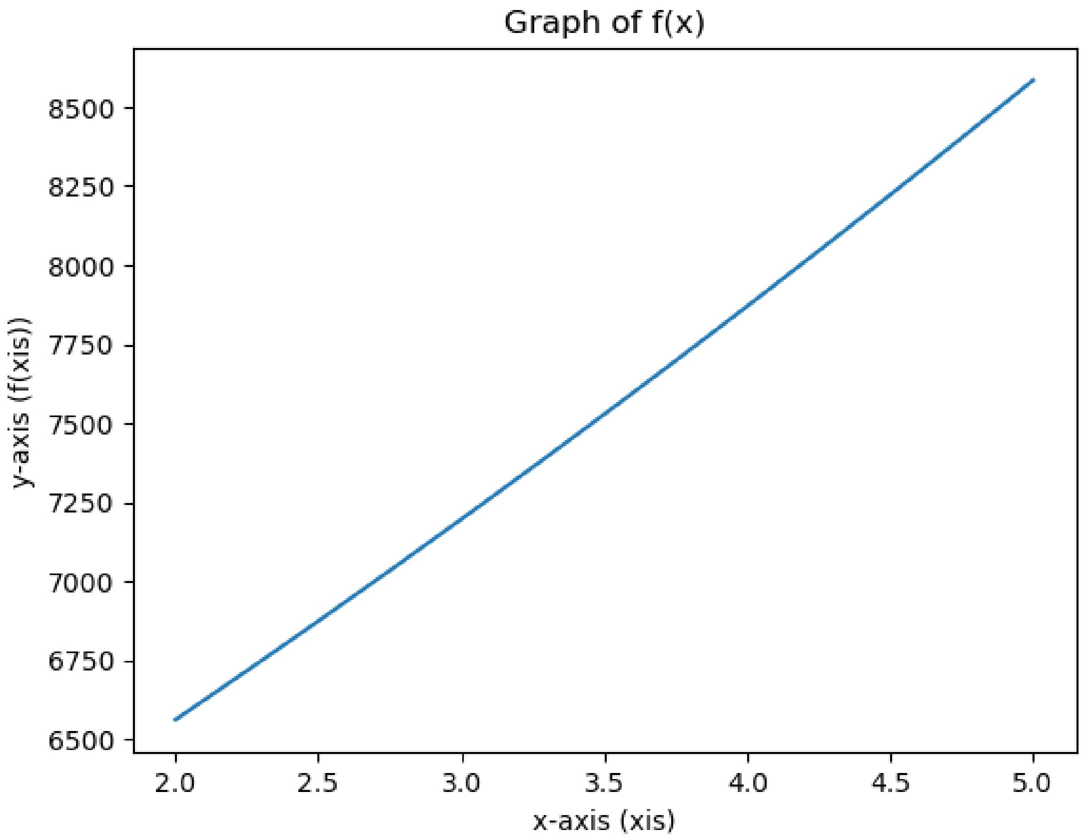

```
Enter f(x): 25  + x^2 - 4x^4
No of sub-intervals(stripe): 20
Enter lower limit: 0
Enter upper linit: 5

The solution to the integral f(x) is  -16426023.38
```



```
Enter f(x): 2x^2 + 4x + 12
No of sub-intervals(stripe): 15
Enter lower limit: 1
Enter upper linit: 4

The solution to the integral f(x) is  1684.52
```




```
Enter f(x): 3x^3/5 + 4x/5
No of sub-intervals(stripe): 12
Enter lower limit: 2
Enter upper linit: 5

The solution to the integral f(x) is  22629.18
```



Real Life Application Of Trapezoidal Rule

Below are some real-life scenarios where the trapezoidal rule is applied;

- Engineering: trapezoidal rule is often used in engineering for like calculating the area under curves in stress-strain diagrams, estimating fluid flow rates in pipelines and determining of a vibrating object over time.
- Economics: in economics, the trapezoidal rule can be employed to estimate consumer and producer surplus by approximating the area under supply and demand curves.
- Finance: Trapezoidal rule can be used to calculate the net present value (NPV) of cash flows in financial analysis. It is also useful in estimating bond-prices and yield to maturity.
- Environmental science: Ecologists and environmental scientists use the trapezoidal rule to estimate changes in environmental parameters over time, such as population growth, pollution levels, or habitat size.
- Computer Graphics: In computer graphics and 3D modeling, trapezoidal rule can be used to approximate shading and rendering effects, such as calculating the illuminations of pixels in an image.
- Numerical Simulations: Trapezoidal rule is part of numerical methods used in various simulations, like computational fluid dynamics, where it helps in approximating the behavior of fluids in complex systems.
- And so on.