Exp:- 3 Numerical Integration using Trapezoidal and Simpson's $\frac{1}{3}$ Rule

Aim :- To implement the Trapezoidal and Simpson's 1/3 Rule in python for a function given. Visualize the integration process by plotting the function and the areas under the curve. We are taking Function = Force.

Principle :- Numerical integration is a method to approximate the value of a definite integral when an exact solution is difficult or impossible to find analytically.

Trapezoidal rule :- It approximates the area under the curve by dividing it into trapezoids.

For a function $F(x)$ over interval $[a, b]$ with n subintervals

$$\int_a^b F(x)\,dx = \frac{h}{2}\left[F(x_0) + 2\left[F(x_1) + F(x_2) + \cdots F(x_{n-1})\right] + F(x_n)\right]$$

Simpson's $\frac{1}{3}$ rule:- It approximates the area under the curve by dividing it into parabolas. It estimates the area of the curve more accurately. It requires even number of interval (n must be even).

$$\int_a^b F(x)\,dx = \frac{h}{3}\left[F(x_0) + 4\sum_{\text{odd } i} F(x_i) + 2\sum_{\text{even } i} F(x_i) + F(x_n)\right]$$

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

b = upper limit a = lower limit n = Intervals

$F(x_0)$ and $F(x_n)$ = Value of function at lower limit and upper limit respectively.

$4 \sum_{odd\ i} F(x_i)$ = Sum of function values at odd indexed interio points

$2 \sum_{even\ i} F(x_i)$ = Sum of function values at even indexed interior points.

Theory :-

Simpson's rule generally provides better accuracy for the sa number of intervals $\boxed{\text{Even}}$.

Physics application

Work done by a Variable force. When a Variable force F(x acts on an object moving along the x-axis from position a to b the work done is:

$$W = \int_a^b F(x)\,dx$$

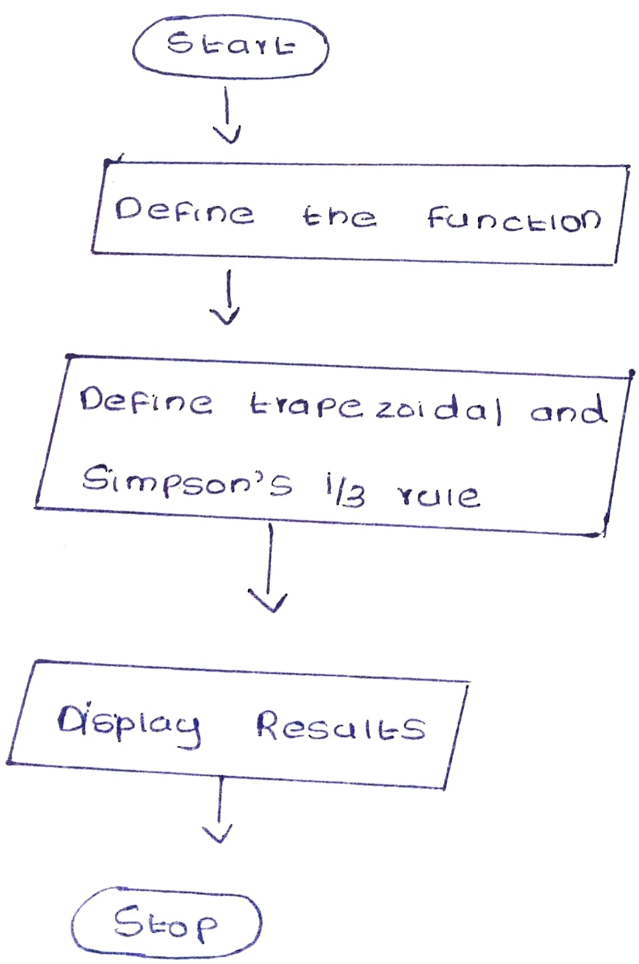
This is ideal for numerical integration when F(x) cannot be integrated analytically. In this experiment we are taking the concept of work done by variable force.

Aug 07, 2025, 00:46

Algorithm

1. Import Matplotlib and numpy
2. Define the Function
3. Define Trapezoidal rule and Simpson's 1/3 rule
4. Print the result
5. Plot the function and shaded area.

Flow chart



Code for Trapezoidal rule

```
import matplotlib.pyplot as plt

import numpy as np

# force function ; F(x) = 5x^2+2

def F(x) :
    return 5* x**2 +2

# Trapezoidal rule for work

def trapez (F, a, b, n) :
    h=(b-a)/n
    Sum = 0
    x = 0.5 *h
    For i in range (1,n) :
        Sum = Sum + h * F(x)
        x = x+h
    return Sum  # This is work in joules

# work calculation

a, b, n = 0.0, 2.0, 1000 # work from x=0 to x=2 in meters

work = trapez (F, a, b, n)

Print (f" work done by F(x) = 5x²+2 from x={a} to x={b} is
    : {work :.6f } J")

# plotting

x_val = np. linspace (a, b, 400)

Force_val = [F(x) For x in x_val]

plt.plot (x_val, Force_val, label = 'F(x) = 5x²+2', color ='darkorange')
```

Show trapezoids

```
n = 20
h = (b-a)/n
for i in range(n):
    x0 = a+i*h
    x1 = x0+h
    plt.fill([x0, x0, x1, x1], [0, f(x0), f(x1), 0], 'orange',
edgecolor = 'black', alpha = 0.4)
# Labels
plt.title('work done by variable force using Trapezoidal rule')
plt.xlabel('Displacement x (m)')
plt.ylabel('force F(x) (N)')
plt.grid(True)
plt.legend()
plt.show()
```

Code for Simpson's $\frac{1}{3}$ Rule

```
import matplotlib.pyplot as plt
import numpy as np
def f(x):
    return 5*x**2
def simpson_work(f, a, b, n):
    h = (b-a)/n
    result = f(a)
    for i in range(1, n, 2):
```

Aug 07, 2025, 9:46

```
    result = result + 4 * f(a+i*h)

  for i in range (2, n, 2):
    result = result + 2 * f(a+i*h)
    result += f(b)
    return (h/3) * result

a, b, n = 0.0, 2.0, 100
work = simpson_work (f, a, b, n)
Print (f"work done by force F(x) = 5x^2+2 From x= {a} to x= {b}
    is: {work_done:.6f} J")

x_vals = np.linspace (a, b, 400)
Force_val = [F(xi) for xi in x_val]
Plt.plot = (x_val, Force_val, label = 'F(x) = 5x^2+2', color = 'purple')
Plt.fill_between (x_val, Force_val, alpha = 0.3, color = 'purple', label =
'work done area')
Plt.title ("work done by variable force F(x) = 5x^2+2")
Plt.xlabel ("Displacement x (m)")
Plt.ylabel ("Force F(x)(N)")
Plt.grid (True)
Plt.legend ()
Plt.show ()
```

Aug 07, 2025, 09:46