

To evaluate the integrand $1/(x^2+4)$ in the range 0 to 3 Correct upto 4 decimal places.

Method 1: Gauss Quadrature method

```
1 import numpy as np
2 from scipy.integrate import quad

1 # Define the function to integrate
2 def f(x):
3     return 1/(x**2 + 4)
4
5 # Perform definite integration from 0 to 1
6 result, error = quad(f, 0, 3)
7 print("Result:", f"{result:.3f}")
8 #print("Estimated error:", error)
```

Result: 0.491

Method 2: Using Trapezoidal method

```
1 #!pip install --upgrade scipy
2
3 import numpy as np
4 #from scipy.integrate import trapz

1 x = np.linspace(0, 3, 100)
2 y = 1/(x**2 + 4) # Corresponding function values
3
4 # Perform definite integration using trapezoidal rule
5 result = np.trapz(y, x)
6 print("Result:", f"{result:.3f}")
```

Result: 0.491
<ipython-input-3-eea7a54dce01>:5: DeprecationWarning: `trapz` is deprecated. Use `trapezoid` instead, or one of the numerical integrators in `scipy.integrate`.
result = np.trapz(y, x)

Method 3: Using Simpsons method

```
1 #!pip install --upgrade scipy
2
3 import numpy as np
4 from scipy import integrate

1 # Sample data points
2 x = np.linspace(0, 3, 100)
3 y = 1/(x**2 + 4) # Corresponding function values
4
5 # Perform definite integration using Simpson's rule
6 result = integrate.simps(y, x)
7 print("Result:", result)
```

AttributeError Traceback (most recent call last)
<ipython-input-4-405b0ea80349> in <cell line: 0>()
4
5 # Perform definite integration using Simpson's rule
----> 6 result = integrate.simps(y, x)
7 print("Result:", result)

AttributeError: module 'scipy.integrate' has no attribute 'simps'

Method 4: Symbolic integration.

```
1 import sympy as sp
2
3 # Define the variable and function
4 x = sp.symbols('x')
5 f = 1/(x**2 + 4)
```

```
6
7 # Perform definite integration from 0 to 1
8 result = sp.integrate(f, (x, 0, 3))
9 print("Result:", result)
10 print('Value of the Definite Integral :', f"{result.evalf():.4f}")
```

↩ Result: $\text{atan}(3/2)/2$
Value of the Definite Integral : 0.4914