Numerical Integration

G V V Sharma*

***Abstract*—Through examples, this manual introduces numerical integration by the Trapezoidal rule, Simpsons 1/3rd and 3/8 Rule and Generalized Quadrature. Python codes are provided for all these methods.**

1 Trapezoidal Rule

Problem 1. Use Fig. 1 to find the integral

$$\int_a^b f(x)\, dx \tag{1.1}$$

by summing up the areas of the trapeziums in Fig. 1.

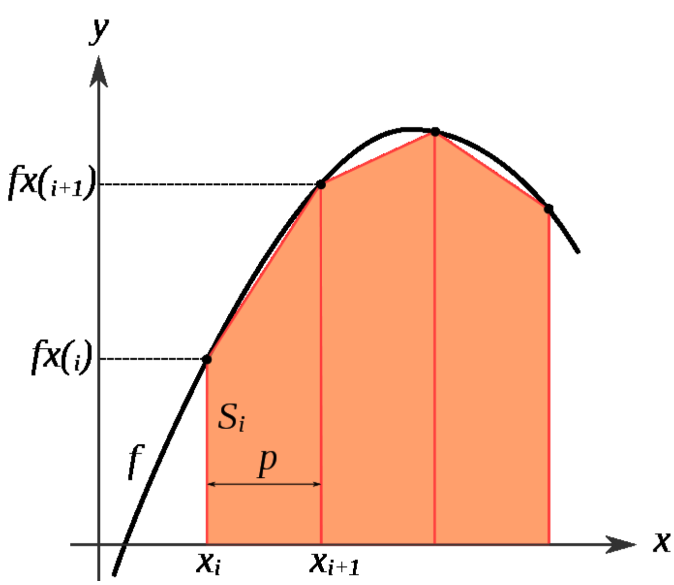

Fig. 1: Trapezoidal Rule.

Solution: The integral can be computed as [1]

$$\int_a^b f(x)\, dx \approx h\left[\tfrac{1}{2} f(a) + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + \tfrac{1}{2} f(b)\right] \tag{1.2}$$

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

Problem 2. Solve

$$\int_0^1 e^{-x^2}\, dx \tag{2.1}$$

using the trapezoidal rule with $n = 10$.

Solution: The following script computes the integral in (2.1) resulting in

$$J = 0.701724989509 \tag{2.2}$$

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
        return np.exp(-x**2)

a = 0
b = 1
n = 11
x = np.linspace(a,b,n)
h = (b-a)/(n-1)

J = h/2*(f(x[0])+2*np.sum(f(x[1:(n
   -2)]))+f(x[n-1]))
print(J)
```

2 Simpson's Rule

2.1 Simpson's 1/3 Rule

The Simpson's 1/3 rule for (2.1) can be expressed as [2]

$$\int_a^b f(x)\, dx \approx \frac{h}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{2m-2} + 4f_{2m-1} + f_{2m}\right] \tag{2.3}$$

*The author is with the Department of Electrical Engineering, IIT, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All material in the manuscript is released under GNU GPL. Free to use for all.

where

$$h = \frac{b-a}{2m}, f_j = f\left(x_j\right), \quad (2.4)$$

$$x_j = x_{j-1} + h, x_0 = a, x_{2m} = b \quad (2.5)$$

Problem 3. Adapt (2.3) into a difference equation

Solution: The desired equation is

$$s_0 = f_0 + f_{2m} \quad (3.1)$$

$$s_1 = f_1 + f_3 + \cdots + f_{2m-1} \quad (3.2)$$

$$s_2 = f_2 + f_4 + \cdots + f_{2m-2} \quad (3.3)$$

$$h = \frac{b-a}{2m} \quad (3.4)$$

$$J = \frac{h}{3}(s_0 + 4s_1 + 2s_2) \quad (3.5)$$

Problem 4. Solve (2.1) using the Simpson's $\frac{1}{3}$rd rule with $n = 10$.

Solution: The following script computes the integral in (2.1) using (2.3) resulting in

$$J = 0.746824948254 \quad (4.1)$$

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
        return np.exp(-x**2)

a = 0
b = 1
n = 11
m = int((n-1)/2)
x = np.linspace(a,b,n)
h = (b-a)/(n-1)
odd_n = list(range(1,2*m+1,2))
even_n = list(range(2,2*m,2))

s0 = f(a) + f(b)
s1 = np.sum(f(x[odd_n]))
s2 = np.sum(f(x[even_n]))
J = h/3*(s0+4*s1+2*s2)
print(J)
```

2.2 Simpson's 3/8 Rule

The Simpson's 3/8 rule can be expressed as [3]

$$\int_a^b f(x)\,dx \approx \tfrac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + 3f(x_5) + 2f(x_6) + ... + f(x_n)\right]. \quad (4.2)$$

Problem 5. Adapt (4.2) into a difference equation

Solution: The desired equation is

$$s_0 = f_0 + f_{3m} \quad (5.1)$$

$$s_1 = f_1 + f_4 + \cdots + f_{3m-2} \quad (5.2)$$

$$s_2 = f_2 + f_5 + \cdots + f_{3m-1} \quad (5.3)$$

$$s_3 = f_3 + f_6 + \cdots + f_{3m-3} \quad (5.4)$$

$$h = \frac{b-a}{3m} \quad (5.5)$$

$$J = \frac{3h}{8}(s_0 + 3s_1 + 3s_2 + 2s_3) \quad (5.6)$$

Problem 6. Solve (2.1) using (4.2) with $n = 30$.

Solution: The following script computes the integral in (2.1) using (4.2) resulting in

$$J = 0.746824155509 \quad (6.1)$$

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
        return np.exp(-x**2)

a = 0
b = 1
n = 31
m = int((n-1)/3)
x = np.linspace(a,b,n)
h = (b-a)/(n-1)
one_m = list(range(1,3*m+1,3))
two_m = list(range(2,3*m+2,3))
three_m = list(range(3,3*m,3))

s0 = f(a) + f(b)
s1 = np.sum(f(x[one_m]))
s2 = np.sum(f(x[two_m]))
s3 = np.sum(f(x[three_m]))
J = 3*h/8*(s0+3*s1+3*s2 +2*s3)
print(J)
```

3 Generalized Quadrature

The Gauss quadrature formula is given by [2]

$$\int_{-1}^{1} f(t)\,dt \approx \sum_{j=1}^{n} w_j f\left(t_j\right) \tag{6.2}$$

where w_j and t_j are obtained from the Legendre polynomial of order n.

Problem 7. Find an expression for (2.1) from (6.2).

Solution: From (6.2), substituting

$$x = \frac{1}{2}\left[a\left(1-t\right) + b\left(t+1\right)\right], \tag{7.1}$$

$$\int_{a}^{b} f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f\left\{(t+1)\frac{(b-a)}{2} + a\right\} dt \tag{7.2}$$

Problem 8. Solve (2.1) using (7.2) with $n = 3$ in (6.2).

Solution: The following script computes the integral in (2.1) using (6.2) resulting in

$$J = 0.746814584191 \tag{8.1}$$

```
import numpy as np

def f(x):
        return np.exp(-x**2)

a = 0
b = 1

deg = 3
t, w = np.polynomial.legendre.
   leggauss(deg)
x = 0.5*(t + 1)*(b - a) + a
J = np.sum(w * f(x))* 0.5*(b - a)
print(J)
```

References

[1] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Trapezoidal_rule

[2] E. Kreyszig, *Advanced engineering mathematics*, 8th ed. New Delhi,: Wiley,, c2007.

[3] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Simpson%27s_rule