

# Chapter-07 Numerical Integration

Date	@November 1, 2024
tag	

## Midpoint Rule

interval  $[a,b] = [x_0, x_1]$

$$Z_0 = (a + b)/2$$

### 1. Divide the Interval:

- The interval  $[a,b]$  is divided into  $N$  equal subintervals.
- The width  $h$  of each subinterval is given by:

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

### 2. The endpoints of the subintervals are:

$$x_i = a + i \cdot h \quad \text{for } i = 0, 1, \dots, N.$$

### 3. Identify the Midpoints:

- The midpoint  $z_i$  of each subinterval  $[x_i, x_{i+1}]$  is:

$$z_i = \frac{x_i + x_{i+1}}{2}$$

- Since  $x_{i+1} = x_i + h$ , we can write:

$$z_i = \frac{x_i + (x_i + h)}{2} = x_i + \frac{h}{2}$$

### 4. Apply the Midpoint Rule:

- The midpoint rule approximation for each subinterval is:

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx h \cdot f(z_i)$$

- Therefore, the integral over the entire interval  $[a,b]$  is approximated by:

$$I_N[f] \approx h \sum_{i=0}^{N-1} f(z_i)$$

N is directly prop to approximation

```
f = lambda x: np.sin(x)
N = 10 # intervals
a = 0.0; b= 2.0*np.pi; h=(b-a)/N
x = np.linspace(a, b, N+1)
z = (x[1:] + x[:-1]) / 2.0
w = h*np.ones_like(z)
quad = np.inner(w, f(z))
print('Result =', quad)
```

Result = -1.3139323596589723e-16

Let  $N = 10$  (no of subintervals)

$$a = 0, b = 2\pi$$

$$1) h = \frac{b-a}{N} = \frac{2\pi-0}{10} = 0.2\pi$$

2) Compute the Endpoints  $x_i^e$  of subintervals,

$$x_i^e = a + i \cdot h \quad \text{for } i = 0, 1, \dots, N$$

for  $i = 0$  to  $10$ :

$$x_0 = 0 + 0 \cdot 0.2\pi = 0 \quad x_6 = 1.2\pi$$

$$x_1 = 0.2\pi \quad x_7 = 1.4\pi$$

$$x_2 = 0.4\pi \quad x_8 = 1.6\pi$$

$$x_3 = 0.6\pi \quad x_9 = 1.8\pi$$

$$x_4 = 0.8\pi \quad x_{10} = 2\pi$$

$$x_5 = \pi$$

3) Compute the Midpoints  $z_i^m$

$$z_i^m = \frac{x_i^e + x_{i+1}^e}{2}$$

$$z_0 = \frac{x_0 + x_1}{2} = \frac{0 + 0.2\pi}{2} = 0.1\pi$$

$$z_1 = \frac{x_1 + x_2}{2} = \frac{0.2\pi + 0.4\pi}{2} = 0.3\pi$$

$$z_2 = \frac{0.4\pi + 0.6\pi}{2} = 0.5\pi$$

$$z_3 = \frac{0.6\pi + 0.8\pi}{2} = 0.7\pi$$

$$z_4 = \frac{0.8\pi + \pi}{2} = 0.9\pi$$

$$z_5 = \frac{\pi + 1.2\pi}{2} = 1.1\pi$$

$$z_6 = \frac{1.2\pi + 1.4\pi}{2} = 1.3\pi$$

$$z_7 = \frac{1.4\pi + 1.6\pi}{2} = 1.5\pi$$



$$z_8 = \frac{1.6\pi + 1.8\pi}{2} = 1.7\pi$$

$$z_9 = \frac{1.8\pi + 2\pi}{2} = 1.9\pi$$

4) Evaluate  $f(z_i) = \sin(z_i)$  for each midpoint:

$$f(z_0) = \sin(0 \cdot 1\pi)$$

$$f(z_5) = \sin(1 \cdot 1\pi)$$

$$f(z_1) = \sin(0 \cdot 3\pi)$$

$$f(z_6) = \sin(1 \cdot 3\pi)$$

$$f(z_2) = \sin(0 \cdot 5\pi)$$

$$f(z_7) = \sin(1 \cdot 5\pi)$$

$$f(z_3) = \sin(0 \cdot 7\pi)$$

$$f(z_8) = \sin(1 \cdot 7\pi)$$

$$f(z_4) = \sin(0 \cdot 9\pi)$$

$$f(z_9) = \sin(1 \cdot 9\pi)$$

5) Calculate the Approximate Integral.

midpoint formula:

$$I_N \approx h \sum_{i=0}^{N-1} f(z_i)$$

$$h = 0.2\pi$$

$$I_N = (0.2\pi)(\sin 0 \cdot 1\pi + \sin 0 \cdot 3\pi + \sin 0 \cdot 5\pi + \sin 0 \cdot 7\pi + \sin 0 \cdot 9\pi + \sin 1 \cdot 1\pi + \sin 1 \cdot 3\pi + \sin 1 \cdot 5\pi + \sin 1 \cdot 7\pi + \sin 1 \cdot 9\pi)$$

$$N = 1$$

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

$$f(x) = x^2 \quad [2, 5]$$

$$1) h = \frac{5-2}{1} = 3$$

$$2) x_i = a + i \cdot h \quad i = 0, 1$$

$$x_0 = 2 + 0(3) = 2$$

$$x_1 = 2 + 1(3) = 5$$

$$3) z_0 = \frac{x_0 + x_1}{2} = \frac{2+5}{2} = 3.5$$

$$4) f(z_0) = \sin(3.5) \quad (3.5)^2 = 12.25$$

$$5) h(12.25) = 3(12.25) = 36.75$$

$$N = 2$$

$$1) h = \frac{b-a}{N} = \frac{5-2}{2} = \frac{3}{2}$$

$$2) x_i = a + i \cdot h \quad \text{for } i=0, 1, 2$$

$$x_0 = 2 + 0(3/2) = 2$$

$$x_1 = 2 + 1(3/2) = 7/2$$

$$x_2 = 2 + 2(3/2) = 5$$

$$3) z_0 = \frac{2+\frac{7}{2}}{2} = 11/4$$

$$z_1 = \frac{\frac{7}{2}+5}{2} = 17/4$$

$$4) f(z_0) = (11/4)^2 = 121/16$$

$$f(z_1) = (17/4)^2 = 289/16$$

$$5) (3/2)(121/16 + 289/16)$$
$$= 38.4375$$

Actual Value

$$\int_2^5 x^2 = \frac{x^3}{3} \Big|_2^5 = \frac{5^3}{3} - \frac{2^3}{3} = 39.$$

Error ( $N=1$ )

$$139 - 38.751 = 2.25.$$

Error ( $N=2$ )

$$139 - 38.43751 = 0.5625$$

$N \uparrow \rightarrow$  Accuracy  $\uparrow$

### Application in Classical Mechanics

force  $f$  at a location  $x$

$x_i$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	
$f(x_i)$	0.0	0.45	1.45	2.3	3.1	3.1	3.1	2.5	1.1	1.1	1.1	1.1	0.8	0.6	0.3	0.0

compute work done by the particular force using the trapezoidal rule

```
from scipy.integrate import trapezoid
import numpy as np

x = np.arange(0.0, 1.5, 0.1)
y = np.array([0.0, 0.45, 1.45, 2.3, 3.1, 3.1, 3.1,
              2.5, 1.1, 1.1, 1.1, 0.8, 0.6, 0.3, 0.0])
print( trapezoid(y,x) )
#2.1
```

# Newton-Cotes Quadrature Rules

## Trapezoidal Rule

$$\begin{aligned} I_1[f] &= [f(a) + f(b)] \frac{h}{2} \\ f(x) &= x^2 \\ [2, 5] \\ f(a) &= f(2) = (2)^2 = 4 \\ f(b) &= f(5) = (5)^2 = 25 \\ I_1[f] &= (4 + 25) \frac{h}{2} \\ h &= \frac{b-a}{N} = \frac{5-2}{1} = 3 \\ I_1[f] &= \frac{29 \times 3}{2} = 43.5 \end{aligned}$$

## Composite Trapezoidal Rule

1. Calculate  $h = \frac{b-a}{N}$

2. List the  $x_i$  Values:

- Compute each  $x_i$  using  $x_i = a + i \cdot h$

$$i = 0, 1, \dots, N$$

3. Evaluate  $f(x_i)$  at Each Point.

4. Plug Into the Composite Formula:

- Compute:

$$I_N[f] \approx \frac{h}{2} [f(x_0) + 2f(x_1) + \dots + 2f(x_{N-1}) + f(x_N)].$$

5. Calculate the Final Result.

$$N = 2$$

$$1) h = \frac{b-a}{N} = \frac{5-2}{2} = \frac{3}{2}$$

2) Calc  $x_i$  values

$$x_i = a + i \cdot h \quad i = 0, 1, 2.$$

$$x_0 = 2 + 0(3/2) = 2$$

$$x_1 = 2 + 1(3/2) = 7/2$$

$$x_2 = 2 + 2(3/2) = 5$$

3) Evaluate  $f(x_i)$

$$f(x_0) = f(2) = (2)^2 = 4$$

$$f(7/2) = (7/2)^2 = 49/4 \approx 12.25$$

$$f(5) = (5)^2 = 25$$

$$4) I_N[f] = \frac{h}{2} [ f(x_0) + 2f(x_1) + \dots + f(x_2) ].$$

$$= \frac{3}{4} [ 4 + 2(\frac{49}{4}) + 25 ]$$

$$= \frac{3}{4} [ 4 + 12.25 + 25 ]$$

$$= \frac{3}{4} [ 36.25 ] =$$

$$= 40.125.$$

$$\text{Error} = |39 - 40.125| = 1.125.$$

$N \uparrow \rightarrow \text{approx} \uparrow$

```
def trapezoidal(f, a, b, N):
    # Composite trapezoidal rule with N subintervals
    x = np.linspace(a, b, N+1)
    y = f(x)
    h = (b - a)/N
    # First add the internal nodes
```

```

sum = 0.0
for i in range(1,N):
    sum += 2.0*y[i]
# Add the two boundary nodes
sum = 0.5*h*(f(a) + sum + f(b))
return sum

f = lambda x: np.sin(x)

trapezoidal(f, 0.0, np.pi/2.0, 10)

```

```

from scipy.integrate import trapezoid
f = lambda x: np.sin(x)
x = np.linspace(0.0, np.pi/2.0, 11)
y = f(x)
print( trapezoid(y,x) )
#0.9979429863543573

```

## Simpson's Rule

Simpson's 1/3 rule uses N=2 and thus quadratic polynomials:

$$I_2 = \sum_{i=0}^2 w_i f(x_i) = \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \frac{h}{3}$$

$$\begin{aligned}
 h &= \frac{b-a}{N} = \frac{5-2}{2} = \frac{3}{2} \quad a=2, b=5 \\
 I_2 &= (f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)) \frac{h}{3} \\
 &= (4 + 4f(2.25) + 25) \frac{3}{6} \\
 &= 39. \\
 E_{max} &= |39 - 39| = 0 \Rightarrow
 \end{aligned}$$

## Composite Simpson's Rule

1. Calculate  $h = \frac{b-a}{N}$ .
2. List the Points  $x_i$ :
  - Compute  $x_i = a + i \cdot h$ $i = 0, 1, \dots, N$
3. Evaluate  $f(x_i)$ 
  - Compute  $f(x_i)$  for each  $x_i$ .
4. Apply the Composite Simpson's Rule Formula:
 
$$I_N[f] \approx \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + \dots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)]$$
5. Compute the Final Approximation:
  - Multiply each  $f(x_i)$  by its respective coefficient.
  - Sum up all the terms.
  - Multiply by  $h/3$  to get the final approximation.

## Coefficients Explanation

1. Endpoints
  - The function values at the endpoints  $f(x_0)$  and  $f(x_N)$  are multiplied by 1. These are the first and last terms.

## 2. Odd Index Points

- The function values at these points are multiplied by **4**. These are the terms with odd indices, starting from  $x_1$  and ending at  $x_{N-1}$ .

## 3. Even Index Points

- The function values at these points are multiplied by **2**. These are the terms with even indices, starting from  $x_2$  and ending at  $x_{N-2}$ .

- Simpson's rule is more accurate** than the trapezoidal rule for functions that are well approximated by parabolas.
- N must be even**, so if you are given an odd number of subintervals, you will need to adjust N accordingly.

$$\begin{aligned}
 N &= 4. & a &= 2 & b &= 5 \\
 \Rightarrow h &= \frac{5-2}{4} = \frac{3}{4} \\
 2) \quad x_i &= a + hi \quad \text{for } i=0, 1, 2, 3, 4 \\
 x_0 &= 2 + 0(3/4) = 2 \\
 x_1 &= 2 + 1(3/4) = 2.75 = \frac{11}{4} \\
 x_2 &= 2 + 2(3/4) = 3.5 = \frac{7}{2} \\
 x_3 &= 2 + 3(3/4) = 4.25 = \frac{17}{4} \\
 x_4 &= 2 + 4(3/4) = 5 \\
 3) \quad f(2) &= 4 \\
 f(2.75) &= 7.5625 \\
 f(3.5) &= 12.25 \\
 f(4.25) &= 18.0625 \\
 f(5) &= 25 \\
 4) \quad \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + f(x_4)] & \\
 &= \frac{h}{3} [4 + 4(7.5625) + 2(12.25) + 4(18.0625) + 25] \\
 &= \frac{3}{12} (150.75) = 37.5 \\
 \text{Error} &\approx 0.
 \end{aligned}$$

```

def Simpson(f, a, b, N):
    # Composite Simpson rule with N subintervals
    h = (b - a)/float(N)
    x=np.linspace(a, b, N + 1)
    s = f(x)
    s[1:N]=2*s[1:N]
    s[1:N:2]=2*s[1:N:2]
    return h/3.0*np.sum(s)

f = lambda x: np.sin(x)

```

```
Simpson(f, 0.0, np.pi/2.0, 10)
```

1.0000033922209004

```
from scipy.integrate import simpson
f = lambda x: np.sin(x)
x = np.linspace(0.0, np.pi/2.0, 11)
y = f(x)
print( simpson(y,x) )
#1.0000033922209006
```

---

Degree of Exactness

## Gauss-Legendre Quadrature Rules

$$f(x) = x^2$$

interval  $[2, 5]$

- 1) Transform the integral from the interval  $[2, 5]$  to  $[-1, 1]$  over Gaussian Legendre is defined over  $[-1, 1]$ .

$$\int_2^5 x^2 dx = \int_{-1}^1 f(3)$$

$$x = \frac{b-a}{2}t + \frac{a+b}{2}$$

$$a=2, b=5$$

$$x = \frac{5-2}{2}t + \frac{2+5}{2} = \frac{3}{2}t + \frac{7}{2}$$

Change Integral Bounds

$$\frac{dx}{dt} = \frac{3}{2} + 0 = \frac{3}{2} dt$$

$$dx = \frac{3}{2} dt$$

$$\int_2^5 x^2 dx = \int_{-1}^1 \left( \frac{3}{2}t + \frac{7}{2} \right) \cdot \frac{3}{2} dt$$
$$= \frac{3}{2} \int_{-1}^1 \left( \frac{3}{2}t + \frac{7}{2} \right) dt$$

- 2) For a 2 point Gaussian-Legendre over  $[-1, 1]$ , the nodes and weights are:

$$t_1 = -\frac{1}{\sqrt{3}}$$

$$t_2 = \frac{1}{\sqrt{3}}$$

$$w_1 = 1$$

$$w_2 = 1$$

3) Evaluate  $f(x)$  at the transformed nodes for  $t_1$  and  $t_2$ .

For  $t_1 = -\frac{1}{\sqrt{3}}$

$$x_1 = \frac{3}{2} \left( -\frac{1}{\sqrt{3}} \right) + \frac{7}{2} = \frac{7 - \sqrt{3}}{2} = 2.63347$$

For  $t_2 = \frac{1}{\sqrt{3}}$

$$x_2 = \frac{3}{2} \left( \frac{1}{\sqrt{3}} \right) + \frac{7}{2} = \frac{7 + \sqrt{3}}{2} = 4.366$$

4) Compute  $f(x_1)$  and  $f(x_2)$

$$f(x_1) = (x_1)^2 = 6.938$$

$$f(x_2) = (x_2)^2 = 19.0662$$

5) Final Approximation

$$\int_2^5 f(x) dx = \frac{3}{2} [w_1 \cdot f(x_1) + w_2 \cdot f(x_2)]$$

$$= \frac{3}{2} [(1) f(x_1) + (1) f(x_2)]$$

$$= 38.9995$$

$$\approx 39$$

## Computing nodes and weights

$$\begin{aligned} (n+1)P_{n+1}(x) &= (2n+1)xP_n(x) - nP_{n-1}(x) \\ P_0(x) &= 1 \Rightarrow P_1(x) = x \\ \Rightarrow n=1 \quad (P_0(x)) & \\ 2P_2(x) &= 3xP_1(x) - P_0(x) \\ 2P_2(x) &= 3x(x) - 1 = 3x^2 - 1 \\ P_2(x) &= (3x^2 - 1) \cdot \frac{1}{2} \\ \Rightarrow n=2 \quad P_3(x) & \\ 3P_3(x) &= 5xP_2(x) - 2P_1(x) \\ &= 5x\left(\frac{3x^2 - 1}{2}\right) - 2x \\ 3P_3(x) &= 15x^3 - 5x - 2x \\ 3P_3(x) &= 15x^3 - 5x - 4x \\ P_3(x) &= \frac{15x^3 - 5x - 4x}{6} = \frac{15x^3 - 9x}{6} = \frac{5x^3 - 3x}{2} \\ \text{For } n=3 \quad (3\text{-point Quadrature}), \quad P_3(x) &= 0 \\ 5x^3 - 3x &= 0 \Rightarrow 5x^3 - 3x = 0 \Rightarrow x(5x^2 - 3) = 0 \\ x=0 \quad \text{or} \quad 5x^2 - 3 &= 0 \Rightarrow x^2 = \frac{3}{5} \Rightarrow x = \pm \sqrt{\frac{3}{5}} \end{aligned}$$

## Method of computing weights:

### Method 1

Nodes.

$$x_1 = 0 \quad x_2 = \sqrt{\frac{3}{5}} \quad x_3 = -\sqrt{\frac{3}{5}}$$

Weights  $w_i$

$$w_i = \int_{-1}^1 t_i(x) dx$$

$$t_1(x) = \prod_{j \neq 1} \frac{x - x_j}{x_i - x_j}$$

$$\begin{aligned} t_1(x) &= \frac{x - x_2}{x_1 - x_2} \cdot \frac{x - x_3}{x_1 - x_3} = \frac{x - \sqrt{\frac{3}{5}}}{-\sqrt{\frac{3}{5}}} \cdot \frac{x + \sqrt{\frac{3}{5}}}{\sqrt{\frac{3}{5}}} \\ &= \frac{x^2 - \frac{3}{5}}{-\frac{3}{5}} \end{aligned}$$

$$t_2(x) = \frac{x - x_1}{x_2 - x_1} \cdot \frac{x - x_3}{x_2 - x_3} = \frac{x}{\sqrt{\frac{3}{5}}} \cdot \frac{x + \sqrt{\frac{3}{5}}}{\sqrt{\frac{2}{5}} + \sqrt{\frac{3}{5}}} = \frac{x(x + \sqrt{\frac{3}{5}})}{6/5}$$

$$t_3(x) = \frac{x - x_1}{x_3 - x_1} \cdot \frac{x - x_2}{x_3 - x_2} = \frac{x}{-\sqrt{\frac{3}{5}}} \cdot \frac{x - \sqrt{\frac{3}{5}}}{\sqrt{\frac{3}{5}} - \sqrt{\frac{3}{5}}} = \frac{x(x - \sqrt{\frac{3}{5}})}{6/5}$$

$$w_1 = \int_{-1}^1 t_1(x) dx$$

$$= \int_{-1}^1 \frac{x^2 - \frac{3}{5}}{-\frac{3}{5}} dx = -\frac{5}{3} \int_{-1}^1 \left( x^2 - \frac{3}{5} \right) dx$$

$$= -\frac{5}{3} \left( \frac{x^3}{3} \Big|_{-1}^1 - \frac{3}{5} x \Big|_{-1}^1 \right) = -\frac{5}{3} \left( \frac{1}{3} - \frac{(-1)}{3} - \left( \frac{3}{5} - \frac{3(-1)}{5} \right) \right)$$

$$= -\frac{5}{3} \left( \frac{2}{3} - \frac{6}{5} \right) = -\frac{8}{3} \left( \frac{-8}{15} \right) = \frac{8}{9}$$

$$w_2 = \int_{-1}^1 t_2(x) dx$$

- - -

**Method 2(Easier):**

Nodes :-

$$x_1 = 0$$

$$x_2 = \sqrt{\frac{3}{5}}$$

$$x_3 = -\sqrt{\frac{3}{5}}$$

Weights :-

$$w_1 = \frac{2}{3}$$

$$[1 - (x_1)^2][P'_n(x_1)]^2$$

$$P_3(x) = \frac{5x^3 - 3x}{2}$$

$$P'_3(x) = \frac{1}{2} (15x^2 - 3) = \frac{3}{2} (5x^2 - 1)$$

$$\Rightarrow \text{At } x_3 = -\sqrt{\frac{3}{5}}$$

$$P'_3(x_3) = \frac{3}{2} \left[ 5 \left( \frac{3}{5} \right) - 1 \right] = \frac{3}{2} (2) = 3.$$

$$\Rightarrow \text{At } x_2 = \sqrt{\frac{3}{5}}$$

$$P'_3(x_2) = 3.$$

$$\Rightarrow \text{At } x_1 = 0$$

$$P'_3(0) = \frac{3}{2} (-1) = -\frac{3}{2}.$$

$$w_1 = \frac{2}{[1 - 0] [(-3/2)^2]} = \frac{2}{9/4} = \frac{8}{9}.$$

$$w_2 = \frac{2}{\left[1 - \left(\frac{3}{\sqrt{5}}\right)^2\right] (3)^2} = \frac{2}{\frac{2}{5} \cdot 9} = \frac{50}{18} = \frac{10}{18} = \frac{5}{9}$$

$$w_3 = \frac{2}{\left[1 - \left(\frac{3}{\sqrt{5}}\right)^2\right] (3)^2} = \frac{2}{\frac{2}{5} \cdot 9} = \frac{5}{9}.$$

```

def GaussLegendre3(f, a, b):
    # Implementation of Gauss-Legendre quadrature with
    # 3 nodes in general interval $[a,b]$
    N = 2
    # define quadrature weights and nodes
    w = np.array([5.0/9.0, 8.0/9.0, 5.0/9.0])
    z = np.array([-np.sqrt(3.0/5.0), 0.0, np.sqrt(3.0/5.0)])
    # implement formula (\ref{eq:gengauleg})
    c1 = (b-a)/2.0
    c2 = (a+b)/2.0
    s = c1*np.inner(w, f( c1*z + c2 ))
    return s

f = lambda x: 0.5*np.exp(x/2.0)*np.cos(x/2.0)
GaussLegendre3(f, -2.0, 2.0)

```

```

from scipy.integrate import fixed_quad
f = lambda x: 0.5*np.exp(x/2.0)*np.cos(x/2.0)
print( fixed_quad(f, -2.0, 2.0, n=3) )
#(1.9333904692642976, None)

```

```

from scipy.integrate import quadrature
f = lambda x: 0.5*np.exp(x/2.0)*np.cos(x/2.0)
print( quadrature(f, -2.0, 2.0, tol=1.e-15, maxiter=100) )
#(1.9334214962992706, 9.692340263711685e-10)

```