

MSMS 106

Ananda Biswas

November 28, 2024

Practical 06



Composite Integration



Question 1 :

Write an R program to approximate the integral of $f(x)$ over the interval $[a, b]$ using Simpson's $\frac{1}{3}$ rule with n sub-intervals of equal length.

⊕ Simpson's $\frac{1}{3}$ rule is given by

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[\frac{1}{3}f(a) + \frac{4}{3}f\left(\frac{a+b}{2}\right) + \frac{1}{3}f(b) \right]; \text{ where } h = \frac{b-a}{2}.$$

For composite integration, we divide the interval $[a, b]$ into $2N$ sub-intervals each of length $h = \frac{b-a}{2N}$; then we get $2N + 1$ abscissas $x_0, x_1, \dots, x_{2N-1}, x_{2N}$ with $x_0 = a$, $x_{2N} = b$ and $x_i = x_0 + ih \forall i = 1(1)2N-1$.

We write

$$\int_a^b f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \dots + \int_{x_{2N-2}}^{x_{2N}} f(x)dx$$

and individually apply Simpson's $\frac{1}{3}$ rule to each of the integrals in RHS.

```
simpson_one_third <- function(func, a, b){  
  h <- (b - a) / 2  
  
  x <- seq(from = a, to = b, by = h)  
  
  s <- (1/3) * func(x[1]) + (4/3) * func(x[2]) + (1/3) * func(x[3])  
  
  return(h * s)  
}
```

```
composite_simpson_one_third <- function(func, a, b, n){  
  
  if(n %% 2 != 0) stop("number of intervals must be even")  

```

```

h <- (b - a) / n

x <- seq(from = a, to = b, by = h)

result <- 0

# x_indices is required to access the lower limits
x_indices <- seq(from = 1, to = n - 1, by = 2)

for (i in x_indices) {
  result <- result + simpson_one_third(func, x[i], x[i+2])
}

return(result)
}

```

- $f(x) = x^2$, $[a, b] = [0, 2]$ and number of intervals = 4

```

f1 <- function(x) x^2

composite_simpson_one_third(f1, 0, 2, 4)

## [1] 2.666667

```

- $f(x) = \frac{1}{1+x^2}$, $[a, b] = [0, 1]$ and number of intervals = 4, 6

```

f2 <- function(x) 1 / (1 + x^2)

composite_simpson_one_third(f2, 0, 1, 4)

## [1] 0.7853922

```

```

composite_simpson_one_third(f2, 0, 1, 6)

## [1] 0.7853979

```

- $f(x) = \frac{1}{1+x}$, $[a, b] = [0, 1]$, number of intervals = 2, 4, 8

```

f3 <- function(x) 1 / (1 + x)

composite_simpson_one_third(f3, 0, 1, 2)

## [1] 0.6944444

```

```

composite_simpson_one_third(f3, 0, 1, 4)

## [1] 0.693254

```

```
composite_simpson_one_third(f3, 0, 1, 8)
```

```
## [1] 0.6931545
```



Question 2 :

Write an R program to approximate the integral of $f(x)$ over the interval $[a, b]$ using Simpson's $\frac{3}{8}$ rule with n sub-intervals of equal length.

⊕ Simpson's $\frac{3}{8}$ rule is given by

$$\int_a^b f(x)dx \approx \frac{3h}{8} [f(a) + 3f(a+h) + 3f(a+2h) + f(b)]; \text{ where } h = \frac{b-a}{3}.$$

For composite integration, we divide the interval $[a, b]$ into $3N$ sub-intervals each of length $h = \frac{b-a}{3N}$; then we get $3N + 1$ abscissas $x_0, x_1, \dots, x_{3N-3}, x_{3N-2}, x_{3N-1}, x_{3N}$ with $x_0 = a$, $x_{3N} = b$ and $x_i = x_0 + ih \forall i = 1(1)3N-1$.

We write

$$\int_a^b f(x)dx = \int_{x_0}^{x_3} f(x)dx + \int_{x_3}^{x_6} f(x)dx + \dots + \int_{x_{3N-3}}^{x_{3N}} f(x)dx$$

and individually apply Simpson's $\frac{3}{8}$ rule to each of the integrals in RHS.

```
simpson_three_eight <- function(func, a, b){
  h <- (b - a) / 3

  x <- seq(from = a, to = b, by = h)

  s <- (3/8) * func(x[1]) + (9/8) * func(x[2]) + (9/8) * func(x[3]) + (3/8) * func(x[4])

  return(h * s)
}
```

```
composite_simpson_three_eight <- function(func, a, b, n){

  if(n %% 3 != 0) stop("number of intervals must be a multiple of 3")

  h <- (b - a) / n

  x <- seq(from = a, to = b, by = h)

  result <- 0

  # x_indices is required to access the lower limits
  x_indices <- seq(from = 1, to = n - 1, by = 3)

  for (i in x_indices) {
```

```

    result <- result + simpson_three_eight(func, x[i], x[i+3])
  }

  return(result)
}

```

- $f(x) = x^2$, $[a, b] = [0, 1]$ and number of intervals = 6

```

composite_simpson_three_eight(f1, 0, 1, 6)

## [1] 0.3333333

```

- $f(x) = e^x$, $[a, b] = [0, 1]$ and number of intervals = 6

```

f4 <- function(x) exp(x)

composite_simpson_three_eight(f4, 0, 1, 6)

## [1] 1.718298

```



Question 3 :

Write an R program to approximate the integral of $f(x)$ over the interval $[a, b]$ using Trapezoidal rule with n sub-intervals of equal length.

- ⊕ Trapezoidal Rule is given by

$$\int_a^b f(x)dx = \frac{b-a}{2} [f(a) + f(b)].$$

For composite integration, we divide the interval $[a, b]$ into N sub-intervals each of length $h = \frac{b-a}{N}$; then we get $N + 1$ abscissas $x_0, x_1, \dots, x_{N-1}, x_N$ with $x_0 = a$, $x_N = b$ and $x_i = x_0 + ih \forall i = 1(1)\overline{N-1}$.

We write

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{N-1}}^{x_N} f(x)dx$$

and individually apply Trapezoidal rule to each of the integrals in RHS.

```

trapezoidal_rule <- function(func, a, b){
  h <- (b - a) / 2

  result <- h * ( func(a) + func(b))

  return(result)
}

```

```

composite_trapezoidal_rule <- function(func, a, b, n){

  h <- (b - a) / n

  x <- seq(from = a, to = b, by = h)

  result <- 0

  for (i in 1:n) {
    result <- result + trapezoidal_rule(func, x[i], x[i+1])
  }

  return(result)
}

```

- $f(x) = x^2$, $[a, b] = [0, 1]$ and number of intervals = 10

```
composite_trapezoidal_rule(f1, 0, 1, 10)
```

```
## [1] 0.335
```

- $f(x) = \frac{1}{1+x}$, $[a, b] = [0, 1]$ and number of intervals = 2, 4, 8

```
composite_trapezoidal_rule(f3, 0, 1, 2)
```

```
## [1] 0.7083333
```

```
composite_trapezoidal_rule(f3, 0, 1, 4)
```

```
## [1] 0.6970238
```

```
composite_trapezoidal_rule(f3, 0, 1, 8)
```

```
## [1] 0.6941219
```