NUMERICAL ANALYSIS

6

조 혜 성

1. f(x) 적분

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

```
def Simpson(a, b, n, dx):
    s = 0
    if n%2 == 0:
        for i in range(0, n-1, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
    else:
        for i in range(0, n-3, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
        s += (3*dx)/8*(f(b-3*dx)+3*f(b-2*dx)+3*f(b-dx)+f(b))
    err = s-ans
def Trapezoidal(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += dx*(f(a+dx*i)+f(a+dx*(i+1)))/2
```

```
def Midpoint(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += f(a+dx*(i+0.5))*dx
```

```
def Romberg(a, b):
    n = int(input("Enter the divisions for Romberg: "))
    k = []
    R= ((b-a)/2)*(f(a)+f(b))
    H = [[0 for x in range(0)] for y in range(n)]
    for i in range (0, n, 1):
        hi = (b-a)/2**i
        for j in range (1, 2**i+1, 1):
            H[i].append(hi/2*f((a+hi*(2*j-1)/2)))
    for i in range (0, n, 1):
        s = 0.0
        for j in range (0, 2**i, 1):
            s += H[i][j]
        k.append(s)
    for i in range (0, n, 1):
       R = R/2 + k[i]
    print ("%.8f" %(R))
    err = R-ans
```

1. f(x) 적분

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

```
def Simpson(a, b, n, dx):
    s = 0
    if n%2 == 0:
        for i in range(0, n-1, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
    else:
        for i in range(0, n-3, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
        s += (3*dx)/8*(f(b-3*dx)+3*f(b-2*dx)+3*f(b-dx)+f(b))
    err = s-ans
def Trapezoidal(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += dx*(f(a+dx*i)+f(a+dx*(i+1)))/2
```

$$R_{1,1} = \frac{\pi}{2}[\sin 0 + \sin \pi] = 0,$$

$$R_{2,1} = \frac{1}{2}\left[R_{1,1} + \pi \sin \frac{\pi}{2}\right] = 1.57079633,$$

$$R_{3,1} = \frac{1}{2}\left[R_{2,1} + \frac{\pi}{2}\left(\sin \frac{\pi}{4} + \sin \frac{3\pi}{4}\right)\right] = 1.89611890,$$

$$R_{4,1} = \frac{1}{2}\left[R_{3,1} + \frac{\pi}{4}\left(\sin \frac{\pi}{8} + \sin \frac{3\pi}{8} + \sin \frac{5\pi}{8} + \sin \frac{7\pi}{8}\right)\right] = 1.97423160,$$

```
def Midpoint(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += f(a+dx*(i+0.5))*dx
```

```
def Romberg(a, b):
    n = int(input("Enter the divisions for Romberg: "))
    k = []
    R= ((b-a)/2)*(f(a)+f(b))
    H = [[0 for x in range(0)] for y in range(n)]
    for i in range (0, n, 1):
        hi = (b-a)/2**i
        for j in range (1, 2**i+1, 1):
            H[i].append(hi/2*f((a+hi*(2*j-1)/2)))
    for i in range (0, n, 1):
        s = 0.0
        for j in range (0, 2**i, 1):
            s += H[i][j]
        k.append(s)
    for i in range (0, n, 1):
       R = R/2 + k[i]
       print ("%.8f" %(R))
    err = R-ans
```

1. f(x) 적분

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

```
def Simpson(a, b, n, dx):
    s = 0
    if n%2 == 0:
        for i in range(0, n-1, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
    else:
        for i in range(0, n-3, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
        s += (3*dx)/8*(f(b-3*dx)+3*f(b-2*dx)+3*f(b-dx)+f(b))
    err = s-ans
def Trapezoidal(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += dx*(f(a+dx*i)+f(a+dx*(i+1)))/2
```

$$R_{1,1} = \frac{\pi}{2}[\sin 0 + \sin \pi] = 0,$$

$$R_{2,1} = \frac{1}{2}\left[R_{1,1} + \pi \sin \frac{\pi}{2}\right] = 1.57079633,$$

$$R_{3,1} = \frac{1}{2}\left[R_{2,1} + \frac{\pi}{2}\left(\sin \frac{\pi}{4} + \sin \frac{3\pi}{4}\right)\right] = 1.89611890,$$

$$R_{4,1} = \frac{1}{2}\left[R_{3,1} + \frac{\pi}{4}\left(\sin \frac{\pi}{8} + \sin \frac{3\pi}{8} + \sin \frac{5\pi}{8} + \sin \frac{7\pi}{8}\right)\right] = 1.97423160,$$

```
def Midpoint(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += f(a+dx*(i+0.5))*dx
```

```
def Romberg(a, b):
    n = int(input("Enter the divisions for Romberg: "))
    k = []
    R= ((b-a)/2)*(f(a)+f(b))
    H = [[0 for x in range(0)] for y in range(n)]
    for i in range (0, n, 1):
        hi = (b-a)/2**i
        for j in range (1, 2**i+1, 1):
            H[i].append(hi/2*f((a+hi*(2*j-1)/2)))
    for i in range (0, n, 1):
        s = 0.0
        for j in range (0, 2**i, 1):
            s += H[i][j]
        k.append(s)
    for i in range (0, n, 1):
       R = R/2 + k[i]
       print ("%.8f" %(R))
    err = R-ans
```

1. f(x) 적분

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

```
def Simpson(a, b, n, dx):
    s = 0
    if n%2 == 0:
        for i in range(0, n-1, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
    else:
        for i in range(0, n-3, 2):
            s += dx/3*(f(a+dx*i) + 4*f(a+dx*(i+1)) + f(a+dx*(i+2)))
        s += (3*dx)/8*(f(b-3*dx)+3*f(b-2*dx)+3*f(b-dx)+f(b))
    err = s-ans
def Trapezoidal(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += dx*(f(a+dx*i)+f(a+dx*(i+1)))/2
```

$$R_{1,1} = \frac{\pi}{2}[\sin 0 + \sin \pi] = 0,$$

$$R_{2,1} = \frac{1}{2}\left[R_{1,1} + \pi \sin \frac{\pi}{2}\right] = 1.57079633,$$

$$R_{3,1} = \frac{1}{2}\left[R_{2,1} + \frac{\pi}{2}\left(\sin \frac{\pi}{4} + \sin \frac{3\pi}{4}\right)\right] = 1.89611890,$$

$$R_{4,1} = \frac{1}{2}\left[R_{3,1} + \frac{\pi}{4}\left(\sin \frac{\pi}{8} + \sin \frac{3\pi}{8} + \sin \frac{5\pi}{8} + \sin \frac{7\pi}{8}\right)\right] = 1.97423160,$$

```
def Midpoint(a, b, n, dx):
    s = 0
    for i in range (0, n, 1):
        s += f(a+dx*(i+0.5))*dx
```

```
def Romberg(a, b):
    n = int(input("Enter the divisions for Romberg: "))
    k = []
    R= ((b-a)/2)*(f(a)+f(b))
    H = [[0 for x in range(0)] for y in range(n)]
    for i in range (0, n, 1):
        hi = (b-a)/2**i
        for j in range (1, 2**i+1, 1):
            H[i].append(hi/2*f((a+hi*(2*j-1)/2)))
    for i in range (0, n, 1):
        s = 0.0
        for j in range (0, 2**i, 1):
            s += H[i][j]
        k.append(s)
    for i in range (0, n, 1):
       R = R/2 + k[i]
       print ("%.8f" %(R))
    err = R-ans
```

2. Result

$$-\frac{f^{(4)}(\xi)}{2880}(b-a)^5.$$

$$-\frac{f''(\xi)}{12}(b-a)^3.$$

$$+\frac{f''(\xi)}{24}(b-a)^3.$$

```
Type the start point for the integral: 0
Type the end point for the integral: 0.8
How many Divisions do you want?: 400
<Simpson>
Result: 1.64053333, error : -0.00000001

<Trapezoidal>
Result: 1.64051733, error : -0.00001601

<Midpoint>
Result: 1.64054133, error : 0.00000799
How many Divisions do you want?: 4000
<Simpson>
Result: 1.64053333, error : -0.00000001

<Trapezoidal>
Result: 1.64053317, error : -0.00000017

<Midpoint>
Result: 1.64053341, error : 0.00000007
```

```
How many Divisions do you want?: 40
<Simpson>
Result: 1.64053163, error : -0.00000171

<Trapezoidal>
Result: 1.63893376, error : -0.00159958

<Midpoint>
Result: 1.64133296, error : 0.00079962
How many Divisions do you want?: 40000
<Simpson>
Result: 1.64053333, error : -0.00000001

<Trapezoidal>
Result: 1.64053333, error : -0.00000001

<Midpoint>
Result: 1.64053333, error : -0.00000001
```

2. Result- Romberg

```
Enter the divisions for Romberg: 6
1.06880000
1.48480000
1.60080000
1.63055000
1.63803438
1.63990840
<Romberg>
Result: 1.63990840, error : -0.00062494
```

$$-\frac{(b-a)}{12}h_k^2 f''(\xi_k)$$

```
Enter the divisions for Romberg 8
<Romberg>
Result: 1.64049427, error : -0.00003907
Enter the divisions for Romberg: 10
<Romberg>
Result: 1.64053089, error : -0.00000245
Enter the divisions for Romberg: 12
<Romberg>
Result: 1.64053318, error : -0.00000016
Enter the divisions for Romberg: 14
<Romberg>
Result: 1.64053332, error : -0.00000002
Enter the divisions for Romberg: 15
<Romberg>
Result: 1.64053333, error : -0.00000001
```

3. 질문

```
C:₩Users₩Administrator₩.spyder-py3>python 0.py
Type the start point for the integral: 0
Type the end point for the integral: 0.8
Answer : 1.640533333      err of the Answer : 1.8213578793317207e-14
```

```
a = float(input("Type the start point for the integral: "))
b = float(input("Type the end point for the integral: "))
ans, err = integrate.quad(f, a, b)
```

Mid point, Trapezoidal, Simpson 법을 이용하되, 각 방법에서 구간의 크기를 줄여 가면서, 오차가 줄어드는 정도를 비교하시오. (정답: 1.64053334)

4. 다른 식 적용

```
def f(x):
     f= np.exp(-3*x**4+2*x+3)
     return f
```

```
Type the start point for the integral: 0
Type the end point for the integral: 1
Answer : 32.234000518    err of the Answer : 4.71941822140998e-10
How many Divisions do you want?: 4000
<Simpson>
Result: 32.23400052, error : 0.00000000

<Trapezoidal>
Result: 32.23399992, error : -0.00000059

<Midpoint>
Result: 32.23400081, error : 0.00000030
Enter the divisions for Romberg: 20
<Romberg>
Result: 32.23400052, error : -0.00000000
```

```
def f(x):
     f= np.log(3*x**4+2*x+3)+np.exp((0.8)*x**2)
     return f
```

```
Type the start point for the integral: 0
Type the end point for the integral: 1
Answer : 2.832541869    err of the Answer : 3.1447532009622277e-14
How many Divisions do you want?: 4000
<Simpson>
Result: 2.83254187, error : 0.00000000

<Trapezoidal>
Result: 2.83254189, error : 0.00000002

<Midpoint>
Result: 2.83254186, error : -0.00000001

Enter the divisions for Romberg: 15
<Romberg>
Result: 2.83254187, error : 0.00000000
```

4. 다른 식 적용

음의 방향으로 적분 가능?

```
def f(x):
    f= np.log(3*x**4+2*x+3)
    return f
```

```
Type the start point for the integral: 0
Type the end point for the integral: 1
Answer : 1.487509141     err of the Answer : 1.6514668972769243e-14
How many Divisions do you want?: 5000
<Simpson>
Result: 1.48750914, error : -0.00000000

<Trapezoidal>
Result: 1.48750914, error : 0.00000000

<Midpoint>
Result: 1.48750914, error : -0.00000000

Enter the divisions for Romberg: 16
<Romberg>
Result: 1.48750914, error : 0.00000000
```

```
Type the start point for the integral: 1
Type the end point for the integral: 0
Answer : -1.487509141     err of the Answer : 1.6514668972769243e-14
How many Divisions do you want?: 4000
<Simpson>
Result: -1.48750914, error : -0.00000000

<Trapezoidal>
Result: -1.48750915, error : -0.00000001

<Midpoint>
Result: -1.48750914, error : 0.00000000

Enter the divisions for Romberg: 16
<Romberg>
Result: -1.48750914, error : -0.00000000
```

You're Welcome ☺