Section 7.7

Alvin Lin

Calculus II: August 2016 - December 2016

Note

The following calculations were done with a Python program I wrote, available here:
`https://github.com/omgimanerd/experimental/blob/master/calc.py`

Source code:

```
#!/usr/bin/python

from numpy import arange, linspace

def delta_x(a, b, n):
    return (b - a) / n

def left_riemann_sum(f, a, b, n):
    interval = list(linspace(a, b, n, endpoint=False))
    return (interval, delta_x(a, b, n) * sum([f(x) for x in interval]))

def right_riemann_sum(f, a, b, n):
    interval = list(linspace(a, b, n + 1)[1:])
    return (interval, delta_x(a, b, n) * sum([f(x) for x in interval]))

def midpoint_approximation(f, a, b, n):
    delta = delta_x(a, b, n)
    interval = list(arange(a + (delta / 2), b, delta))
    return (interval, delta * sum([f(x) for x in interval]))

def trapezoidal_approximation(f, a, b, n):
    interval = list(linspace(a, b, n + 1))
    return (interval,
            (left_riemann_sum(f, a, b, n)[1] + right_riemann_sum(f, a, b, n)[

def simpsons_rule_approximation(f, a, b, n):
```

```
    series = linspace(a, b, n + 1)
    interval = list(linspace(a, b, n + 1))
    for index, term in enumerate(series):
        if index == 0 or index == len(series) - 1:
            series[index] = f(term)
        elif not(index % 2): # even
            series[index] = 2 * f(term)
        else:
            series[index] = 4 * f(term)
    return (interval, (delta_x(a, b, n) / 3) * sum(series))

if __name__ == '__main__':
    from math import *
    exec("f = lambda x: {}".format(input("Input your function: lambda x: ")))
    a = int(input("Lower limit: "))
    b = int(input("Upper limit: "))
    n = int(input("n: "))
    print("-----")
    print("Interval: {0}\nLeft Riemann Sum Approximation: {1}\n".format(
        *left_riemann_sum(f, a, b, n)))
    print("Interval: {0}\nRight Riemann Sum Approximation: {1}\n".format(
        *right_riemann_sum(f, a, b, n)))
    print("Interval: {0}\nMidpoint Sum Approximation: {1}\n".format(
        *midpoint_approximation(f, a, b, n)))
    print("Interval: {0}\nTrapezoidal Sum Approximation: {1}\n".format(
        *trapezoidal_approximation(f, a, b, n)))
    print("Interval: {0}\nSimpson's Rule Approximation: {1}\n".format(
        *simpsons_rule_approximation(f, a, b, n)))
```

Exercise 7c

$$\int_1^2 \sqrt{x^3 - 1}\,\mathrm{d}x \quad n = 10$$

$$\Delta x = \frac{b-a}{n} = \frac{2-1}{10} = 0.1$$

$$S_n \approx 1.511519$$

Exercise 11c

$$\int_0^4 x^3 \sin x\,\mathrm{d}x \quad n = 8$$

$$\Delta x = \frac{b-a}{n} = \frac{4-0}{8} = 0.5$$

$$S_n \approx -5.605350$$

Exercise 15c

$$\int_0^1 \frac{x^2}{1+x^4}\,\mathrm{d}x \quad n = 10$$

$$\Delta x = \frac{b-a}{n} = \frac{1-0}{10} = 0.1$$

$$S_n \approx 0.243751$$

If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech