## 7.7 Exercises

1. Consider the following code which approximates a Riemann sum for $f(x) = \sin(x)$ on the interval $[0, 2\pi]$ with $h = dx$ spacing.

```
import numpy as np

def f(x):
    return np.sin(x)

dx = 0.1
x = np.arange(0,2*np.pi,dx)
I = sum(f(x))*dx
```

   Use the code to generate a table of the error between $I$ and the truel value of the integral against the size of $dx$ as it decreases from $1 \to 10^{-6}$. What value of $n$ does the algorithm seem to scale with in terms of $\mathcal{O}(h^n)$? (You may alter the code to generate the table)

2. Let $f$ be a real valued function that is continuous on the interval $[a, b]$. Then the fundamental theorem of calculus says that for any $x \in [a, b]$:

$$\frac{d}{dx}\left(\int_a^x f(t)dt\right) = f(x)$$

   We're going to numerically verify this for $f(x) = e^{-x^2}$ by estimating the derivative of an integral.

   (A) Program the function $f(x) = e^{-x^2}$

   (B) Program the function $F(x) = \int_0^x f(x)dx$ using any method of numerical integration you like.

   (C) Using a finite difference formula to estimate the derivative, verify that $f(2) \approx F'(2)$

3. Let $B(0; 1)$ be the unit ball centered on the origin. Use a Monte Carlo method to calculate:

$$\int_{B(0;1)} e^{-(x^2+y^2)}dxdy$$

   You may do a coordinate transformation.