Math 152 – Python Lab 4

February 27, 2023

0.1 MATH 152 Lab 4

MATH 152 Lab 3 Section Number: 571

Members:

- Brighton Sikarskie
- Alex Krakora
- Joseph Pham
- Diego Mendez

[1]:
```
from numpy import *
import sympy as sp
```

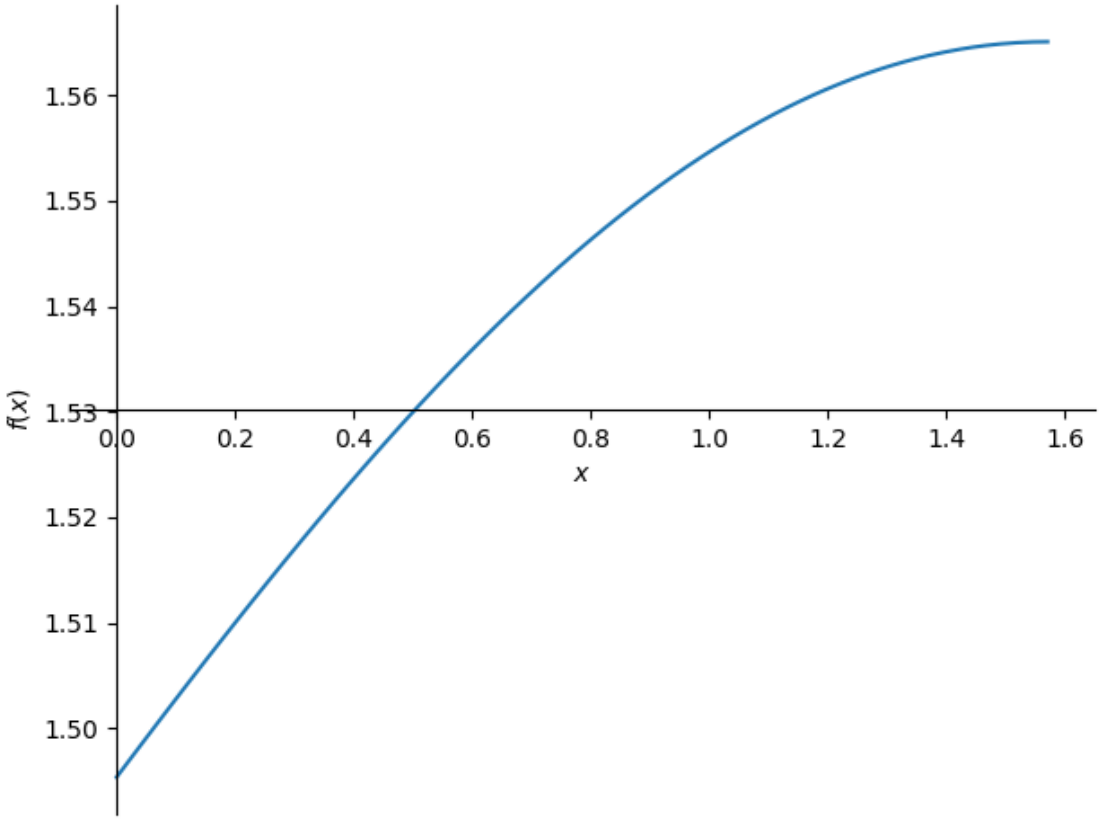
0.1.1 Question 1

1a

[2]:
```
x = sp.symbols("x")
f = (5 + sp.sin(x)) ** (1 / 4)

lower = 0
upper = pi / 2

# Plot the function
sp.plot(f, (x, lower, upper))
print(f"The graph of f(x) = {f} is shown above.")
```



The graph of f(x) = (sin(x) + 5)**0.25 is shown above.

1b Left Endpoint Approximation

[3]:
```
n = 200
delta_x = (upper - lower) / n
# make a list from inclusive lower to exclusive upper
# with step size delta_x
x_values = arange(lower, upper, delta_x)
y_values = [f.subs(x, x_value) for x_value in x_values]
# sum the list y_values and multiply by delta_x
# to get the area under the curve
area_left = sum(y_values) * delta_x
print(f"The area under the curve is {area_left}.")
```

The area under the curve is 2.41936685199696.

0.1.2 Question 2 Right Endpoint Approximation

[4]:
```
n = 200
delta_x = (upper - lower) / n
# make a list from exclusive lower to inclusive upper
# with step size delta_x
x_values = arange(lower + delta_x, upper + delta_x, delta_x)
y_values = [f.subs(x, x_value) for x_value in x_values]
# sum the list y_values and multiply by delta_x
# to get the area under the curve
area_right = sum(y_values) * delta_x
print(f"The area under the curve is {area_right}.")
```

```
The area under the curve is 2.41991455568038.
```

0.1.3 Question 3 Midpoint Approximation

3a

[5]:
```
# start at a + delta_x and go to b - delta_x inclusive
# with step size delta_x
x_values = arange(lower + delta_x / 2, upper + delta_x / 2, delta_x)
y_values = [f.subs(x, x_value) for x_value in x_values]
# sum the list y_values and multiply by delta_x
# to get the area under the curve
area_mid = sum(y_values) * delta_x
print(f"The area under the curve is {area_mid}.")
```

```
The area under the curve is 2.41964128034331.
```

3b

[6]:
```
left_right_average = (area_left + area_right) / 2
print(f"The average of the left and right areas is {left_right_average}.")

print(f"Is the average of the left and right areas equal to the area under the␣
  ↪curve using midpoint? {left_right_average == area_mid}.")
```

```
The average of the left and right areas is 2.41964070383867.
Is the average of the left and right areas equal to the area under the curve
using midpoint? False.
```

0.1.4 Question 4 Trapezoid Approximation

4a

[7]:
```
from scipy.integrate import trapz

x_values = linspace(lower, upper, n + 1)
y_values = [f.subs(x, x_value) for x_value in x_values]
```

```
result = trapz(y_values, x_values).evalf()

print(f"The area under the curve is {result}.")
```

```
The area under the curve is 2.41964070383867.
```

4b

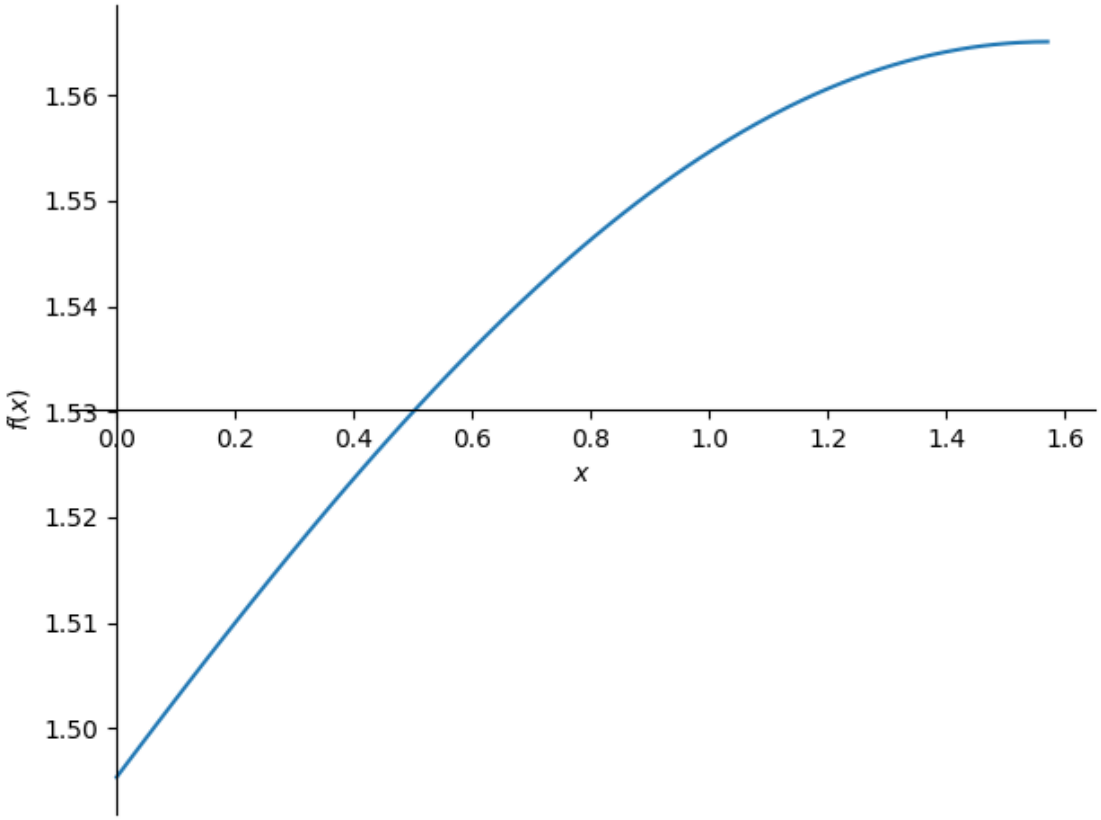
[8]:
```
print(f"Is the trapezoidal rule result equal to the area under the curve using␣
↪the average of left and right Riemann sums? {result == left_right_average}.")
```

```
Is the trapezoidal rule result equal to the area under the curve using the
average of left and right Riemann sums? False.
```

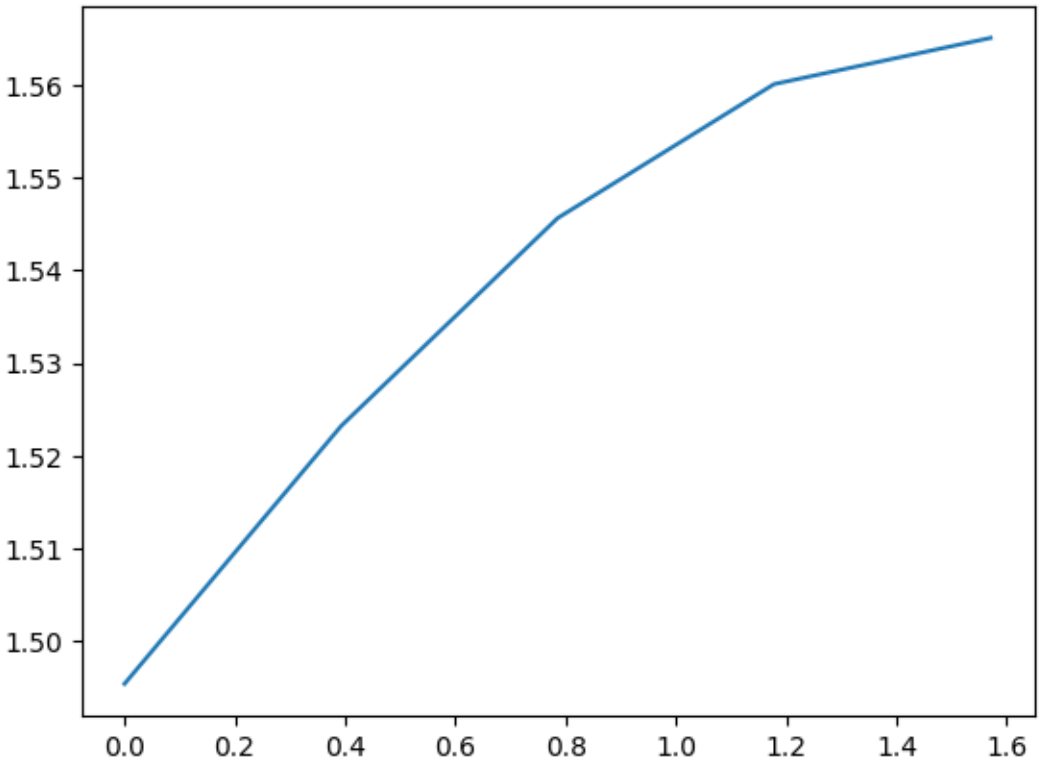
4c

[9]:
```
x = sp.symbols("x")
f = (5 + sp.sin(x)) ** sp.Rational(1, 4)
sp.plot(f, (x, 0, pi / 2))
xp = [0, pi / 8, pi / 4, 3 * pi / 8, pi / 2]
yp = [f.subs(x, i) for i in xp]
import matplotlib.pyplot as plt

plt.plot(xp, yp)
```



[9]: [<matplotlib.lines.Line2D at 0x7fd690add540>]



0.1.5 Question 5 Simpson's Rule

[10]:
```
import numpy as np
from scipy.integrate import simps

a = 0
b = np.pi / 2
n = 200

x = np.linspace(a, b, n)
y = (5 + np.sin(x)) ** 1 / 4
sim = simps(y, x)
print(sim)
```

2.213495403355763

0.1.6 Question 6 Errors

[11]:
```
# The value of the integral to 10 decimal places is 2.4196410881.
# Use this value to estimate the error |actual - estimate| in
# each of the five approximations.
actual = 2.4196410881
# 5 estimate
five = abs(actual - sim)
print("for 5 the error is", five)
```

```
for 5 the error is 0.2061456847442371
```