MATH 152 Lab 4

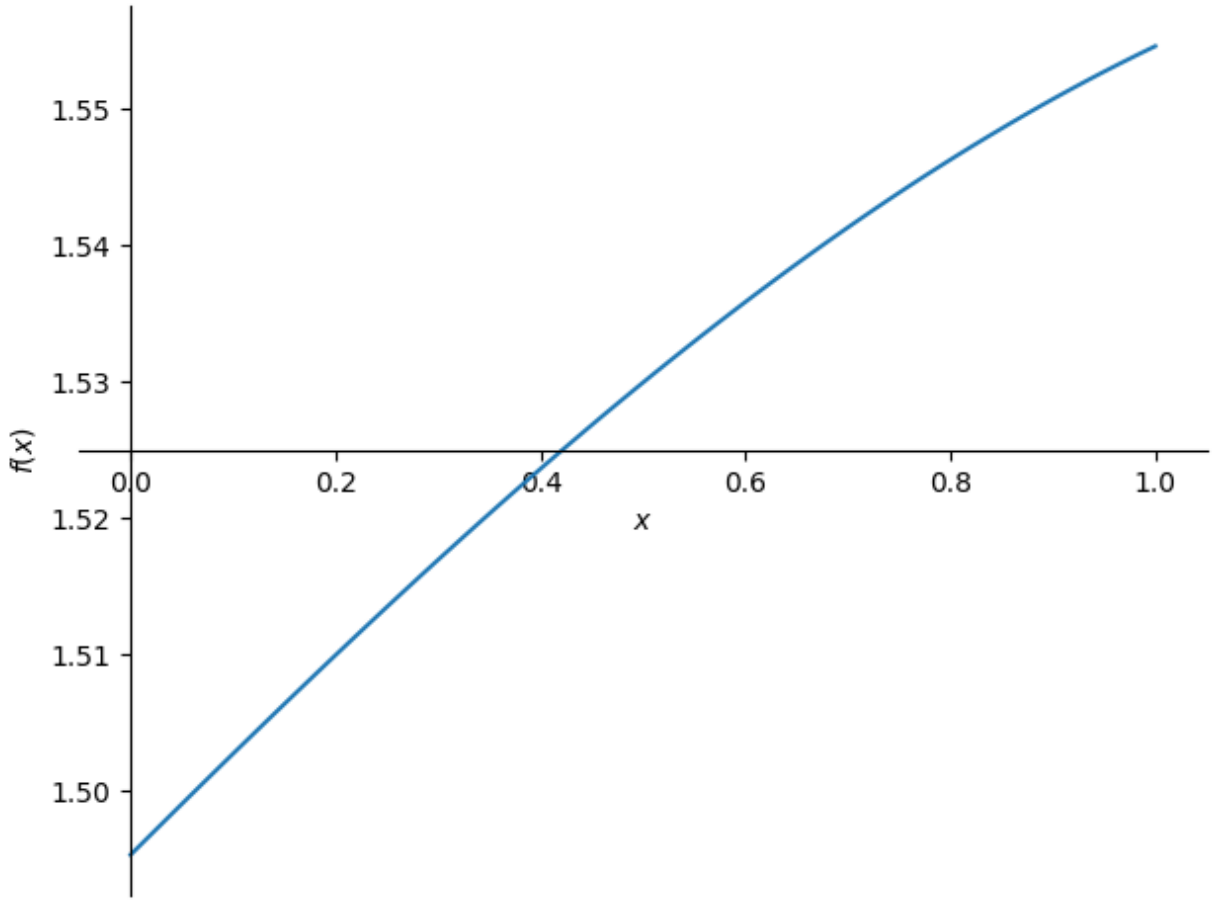
Kevin Lei Grant Smith Alex Smolen Kevin Lee Section 542

In [2]:
```
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
import scipy
```

Question 1

1a

In [3]:
```
x = symbols("x")
fx = (5 + sin(x))**(1/4)
plot(fx, (x, 0, 1))
```



Out[3]:
```
<sympy.plotting.plot.Plot at 0x22823fe1810>
```

1b Left Endpoint Approximation

In [4]:
```
a = 0
b = pi/2
n = 200
```

```
dx = (b - a) / n
xVals = [i for i in np.arange(a, b, float(dx))]
yVals = [fx.subs(x, i) for i in xVals]

left = sum(yVals) * dx
print(f"The left endpoint riemann sum using n = 200 subintervals is {N(left)}")
```

```
The left endpoint riemann sum using n = 200 subintervals is 2.41936685199696
```

Question 2 Right Endpoint Approximation

In [5]:
```
xVals = [i for i in np.arange(a + dx, b + dx, float(dx))]
yVals = [fx.subs(x, i) for i in xVals]
right = sum(yVals) * dx
print(f"The right endpoint riemann sum using n = 200 subintervals is {N(right)}")
```

```
The right endpoint riemann sum using n = 200 subintervals is 2.41991455568038
```

Question 3 Midpoint Approximation

3a

In [6]:
```
xVals = [i for i in np.arange(a + dx/2, b - dx/2 + dx, float(dx))]
yVals = [fx.subs(x, i) for i in xVals]
mid = sum(yVals) * dx
print(f"The midpoint riemann sum using n = 200 subintervals is {N(mid)}")
```

```
The midpoint riemann sum using n = 200 subintervals is 2.41964128034331
```

3b

In [7]:
```
avg = N((left + right) / 2)
print(f"Average of left and right riemann sums: {avg}")
print(f"The average of the left and right riemann sums are approximately equal to the
```

```
Average of left and right riemann sums: 2.41964070383867
The average of the left and right riemann sums are approximately equal to the midpoin
t sum.
```

Question 4 Trapezoid Approximation

4a

In [8]:
```
xVals = [i for i in np.arange(a, b + dx, float(dx))]
yVals = [fx.subs(x, i) for i in xVals]
trapezoid = np.trapz(yVals, xVals)
print(f"The trapezoid approximation is {N(trapezoid)}")
```

```
The trapezoid approximation is 2.41964070383866
```

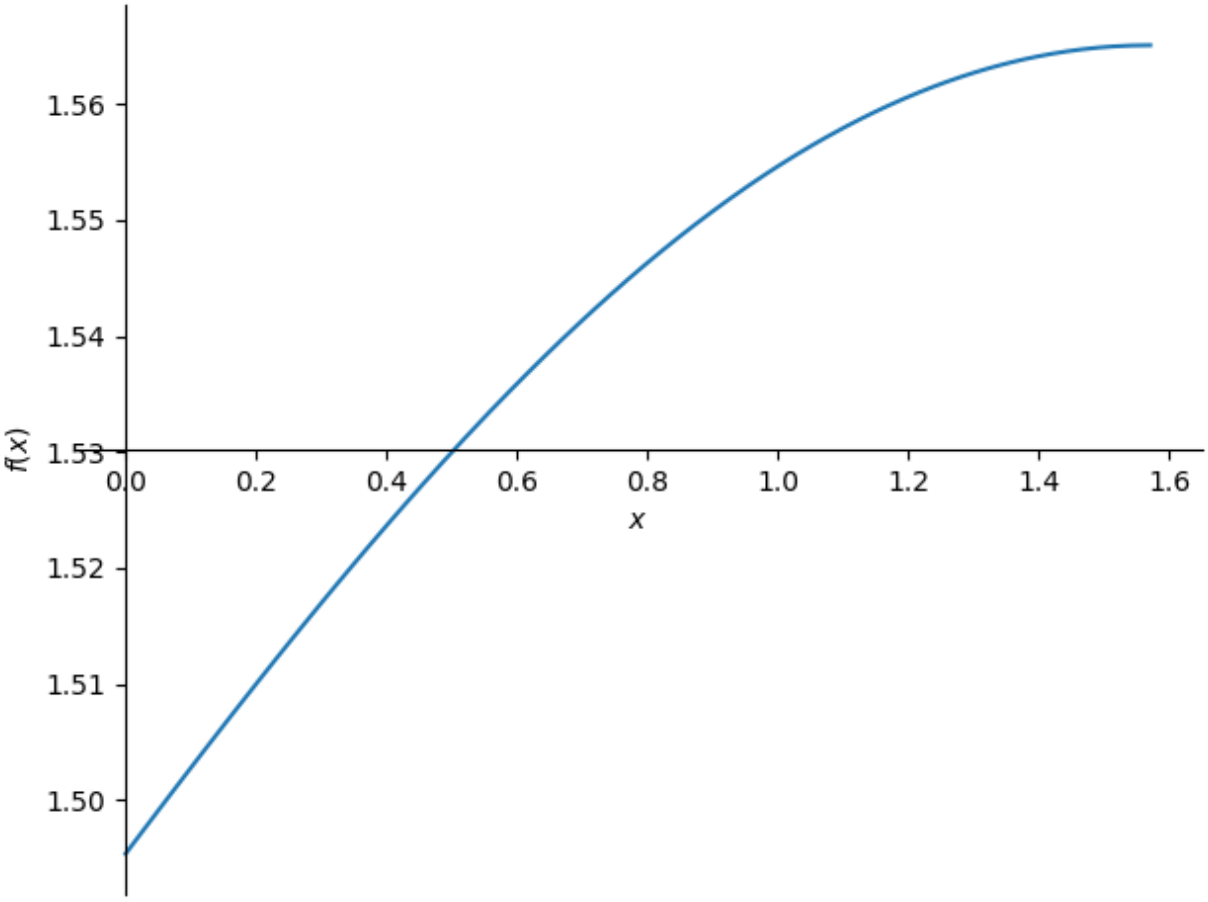
4b

In [9]:
```
print(f"The trapezoid approximation is approximately equal to the average of the left
```

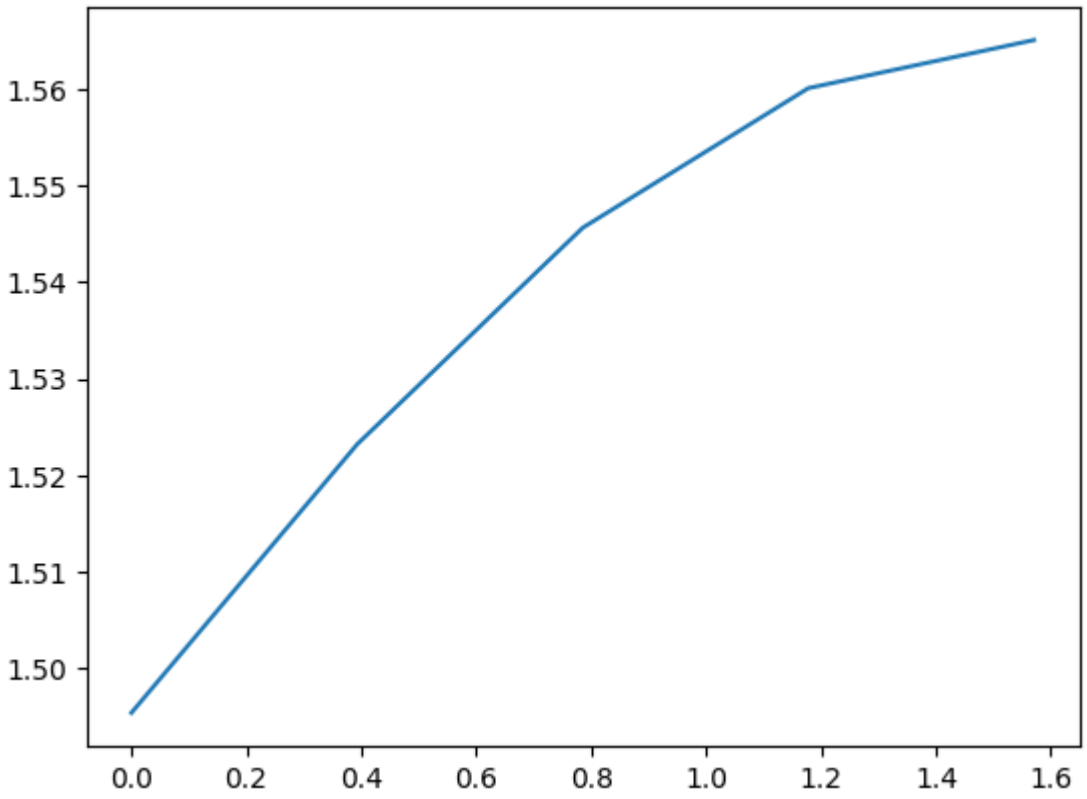
The trapezoid approximation is approximately equal to the average of the left and rig
ht endpoint approximations.

4c

In [10]:
```
x=symbols('x')
f=(5+sin(x))**Rational(1,4)
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]
plt.plot(xp,yp)
```



Out[10]: [<matplotlib.lines.Line2D at 0x2282679d5d0>]



Question 5 Simpson's Rule

In [11]:
```
xVals = [i for i in np.arange(a, b + dx, float(dx))]
yVals = [fx.subs(x, i) for i in xVals]
simp = scipy.integrate.simps(yVals, xVals)
print(f"The simpson approximation with n=200 subintervals is {N(simp)}")
```

```
The simpson approximation with n=200 subintervals is 2.41964108817649
```

Question 6 Errors

In [12]:
```
actual = 2.4196410881
print(f"Error of left endpoint approximation: {abs(actual - N(left))}")
print(f"Error of right endpoint approximation: {abs(actual - N(right))}")
print(f"Error of midpoint approximation: {abs(actual - N(mid))}")
print(f"Error of trapezoid approximation: {abs(actual - N(trapezoid))}")
print(f"Error of Simpson's method: {abs(actual - N(simp))}")
```

```
Error of left endpoint approximation: 0.000274236103036785
Error of right endpoint approximation: 0.000273467580377496
Error of midpoint approximation: 1.92243313712481E-7
Error of trapezoid approximation: 3.84261339192449E-7
Error of Simpson's method: 7.64948104858831E-11
```