Math 152 – Python Lab 2

February 3, 2023

0.1 MATH 152 Lab 2

MATH 152 Lab 2 Section Number: 571

Members:

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

[1]:
```
from sympy import *
from sympy.plotting import (plot,plot_implicit)
```

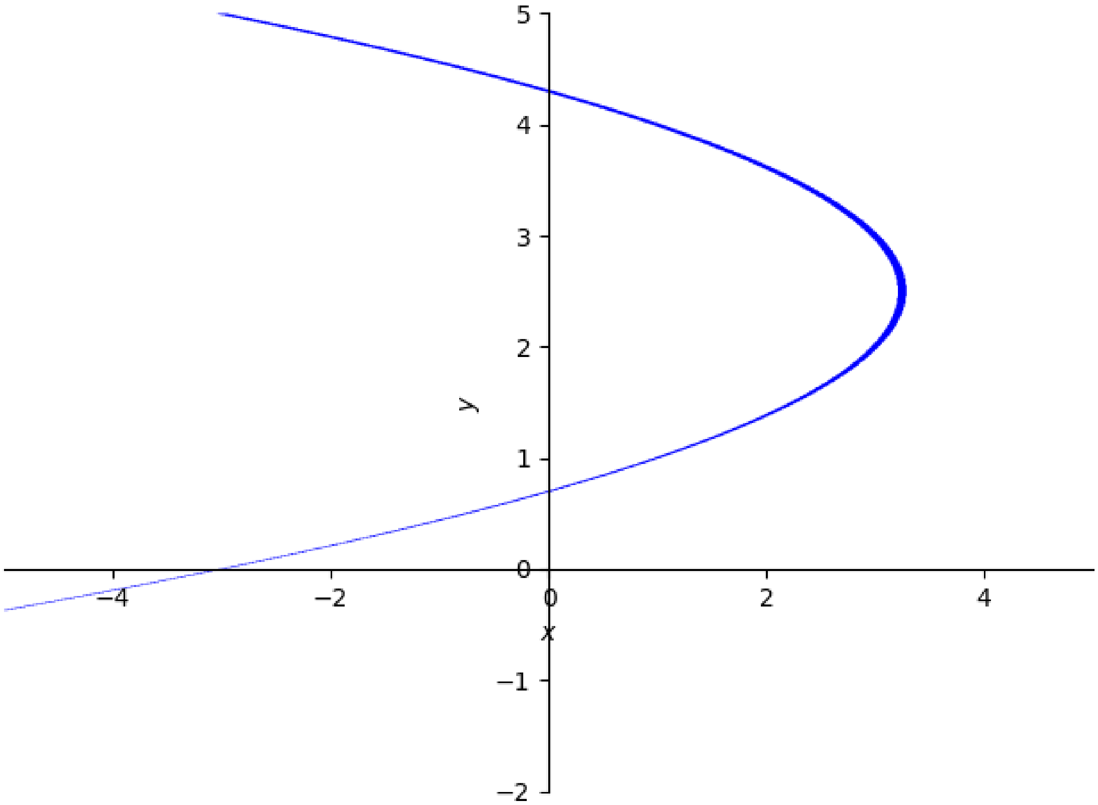
0.1.1 Question 1

1a

[2]:
```
x, y = symbols('x y')

fx = -x - y ** 2 + 5 * y - 3

# plot on y [-2, 5]
plot_implicit(fx, x, (y, -2, 5))
print("The graph of x = -y^2 + 5y - 3 is shown above.")
```



The graph of x = -y^2 + 5y - 3 is shown above.

1b

[3]:
```
fx = -y ** 2 + 5 * y - 3
y_int = solve(fx, y)
print(f"The y-intercept's for the function x = -y^2 + 5y - 3 are at {y_int}")
```

```
The y-intercept's for the function x = -y^2 + 5y - 3 are at [5/2 - sqrt(13)/2,
sqrt(13)/2 + 5/2]
```

1c

[4]:
```
vol = pi * integrate(fx ** 2, (y, *y_int))
print(f"The volume of the function {fx} is {vol.simplify()} which is about {vol.
 ↪evalf()} when rotated about the y-axis")
```

```
The volume of the function -y**2 + 5*y - 3 is 169*sqrt(13)*pi/30 which is about
63.8097434818162 when rotated about the y-axis
```

0.1.2 Question 2

2a

[5]:
```
x = Symbol("x")
y1 = sin(x)
y2 = cos(x)
sol = solve(y1 - y2)[0]
R = pi * (integrate(y2 ** 2, (x, 0, sol)) - integrate(y1 ** 2, (x, 0, sol)))
print(f"The volume of the functions {y1} and {y2} is {R.simplify()} which is␣
↪about {R.evalf()} when rotated about the x-axis")
```

```
The volume of the functions sin(x) and cos(x) is pi/2 which is about
1.57079632679490 when rotated about the x-axis
```

2b

[6]:
```
y = Symbol("y")
y1 = asin(y)
y2 = acos(y)
R = pi * (integrate(y2 ** 2, (y, cos(sol), 1)) + integrate(y1 ** 2, (y, 0,␣
↪cos(sol))))
print(f"The volume of the function is {R.simplify()} which is about {R.evalf()}␣
↪when rotated about the y-axis")
```

```
The volume of the function is pi*(-4 + sqrt(2)*pi)/2 which is about
0.695678892459293 when rotated about the y-axis
```

2c

[7]:
```
R = 2 * pi * (integrate(x * (cos(x) - sin(x)), (x, 0, pi / 4)))

print(f"Using culindrical shells we get the volume to be {R.simplify()} which␣
↪is about {R.evalf()} when rotated about the y-axis")
```

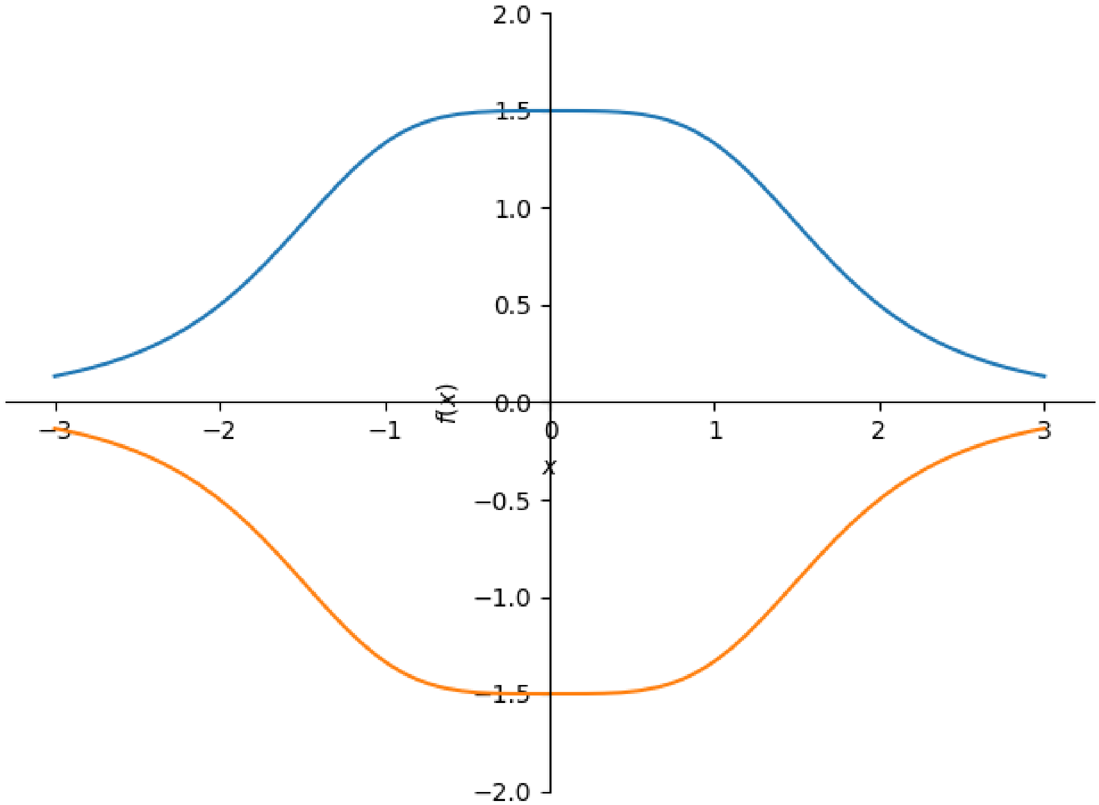
```
Using culindrical shells we get the volume to be pi*(-4 + sqrt(2)*pi)/2 which is
about 0.695678892459293 when rotated about the y-axis
```

0.1.3 Question 3

3a

[8]:
```
f = 12 / (8 + x ** 4)
lemon = plot(f, (x, -3, 3), ylim=(-2, 2), show=False)
lemon.extend(plot(-f, (x, -3, 3), ylim=(-2, 2), show=False))
lemon.show()

print(f"The graph of {f} and its reflection is shown above.")
```



The graph of 12/(x**4 + 8) and its reflection is shown above.

3b

[9]:
```
A = integrate(f, (x, -3, 3)) * 2
print(f"The surface area of the largest slice is {A.simplify()} which is about␣
↪{A.evalf()}")
```

The surface area of the largest slice is 3*2**(1/4)*(log((2*sqrt(2) + 6*2**(1/4) + 9)/(-6*2**(1/4) + 2*sqrt(2) + 9)) - 2*atan(1 - 3*2**(3/4)/2) + 2*atan(1 + 3*2**(3/4)/2))/2 which is about 10.6390331471780

3c

[10]:
```
V = pi * integrate(f ** 2, (x, -3, 3))

print(f"The volume of the lemon from part (a) is {V.simplify()} which is about␣
↪{V.evalf()}")
```

The volume of the lemon from part (a) is 27*pi*(32 + log((2*sqrt(2) + 6*2**(1/4) + 9)**(89*2**(1/4))/(-6*2**(1/4) + 2*sqrt(2) + 9)**(89*2**(1/4))) - 178*2**(1/4)*atan(1 - 3*2**(3/4)/2) + 178*2**(1/4)*atan(1 + 3*2**(3/4)/2))/2848 which is about 19.7537908960011

0.1.4 Question 4

4ai

[11]:
```
f1 = exp(-sqrt(x))
A = integrate(f1, (x, 0, 1))
print(f"The integral of {f1} from 0 to 1 is {A.simplify()} which is about {A.
 ↪evalf()}")
```

```
The integral of exp(-sqrt(x)) from 0 to 1 is 2 - 4*exp(-1) which is about
0.528482235314231
```

4aii

[12]:
```
f2 = exp(-cos(x)) * sin(2 * x)
A = integrate(f2, (x, 0, pi / 2))
print(f"The integral of {f2} from 0 to pi/2 is {A.simplify()} which is about {A.
 ↪evalf()}")
```

```
The integral of exp(-cos(x))*sin(2*x) from 0 to pi/2 is 2 - 4*exp(-1) which is
about 0.528482235314231
```

4aiii

[13]:
```
f3 = 2 * x * exp(-x)
A = integrate(f3, (x, 0, 1))
print(f"The integral of {f3} from 0 to 1 is {A.simplify()} which is about {A.
 ↪evalf()}")
```

```
The integral of 2*x*exp(-x) from 0 to 1 is 2 - 4*exp(-1) which is about
0.528482235314231
```

4b

[14]:
```
print(f"For the first integral ({f1}) you can let u = sqrt(x) and du = 1/
 ↪(2*sqrt(x))dx which would give dx = 2*u*du. When you plug those values into␣
 ↪the first integral you get 2*u*e^(-u)*du which is the same as the third␣
 ↪integral, but in terms of u not x. You do need to change the bounds, but in␣
 ↪this case the bounds end up being the same.")
print(f"For the second integral ({f2}) you can let u = cos(x) and du =␣
 ↪-sin(x)*dx which would give dx = du/(-sin(x)). When you plug those values␣
 ↪into the second integral you get e^(-u)*(sin(2x)/(-sin(x)))*du. Then you can␣
 ↪apply the double angle identity sin(2x)=2*sin(x)*cos(x) for the numerator␣
 ↪and get e^(-u)*2*cos(x)*du, then substitute cos(x) for u to get the same␣
 ↪inside of the third integral. To change the bounds plug in the lower bound␣
 ↪(0) into cos(x) to get 0 and the upper bound (pi/2) into cos(x) to get 1.")
```

```
For the first integral (exp(-sqrt(x))) you can let u = sqrt(x) and du =
1/(2*sqrt(x))dx which would give dx = 2*u*du. When you plug those values into
the first integral you get 2*u*e^(-u)*du which is the same as the third
integral, but in terms of u not x. You do need to change the bounds, but in this
case the bounds end up being the same.
```

For the second integral (exp(-cos(x))*sin(2*x)) you can let u = cos(x) and du = -sin(x)*dx which would give dx = du/(-sin(x)). When you plug those values into the second integral you get e^(-u)*(sin(2x)/(-sin(x)))*du. Then you can apply the double angle identity sin(2x)=2*sin(x)*cos(x) for the numerator and get e^(-u)*2*cos(x)*du, then substitute cos(x) for u to get the same inside of the third integral. To change the bounds plug in the lower bound (0) into cos(x) to get 0 and the upper bound (pi/2) into cos(x) to get 1.