Math 152 – Python Lab 9

April 25, 2023

0.1 MATH 152 Lab 9

MATH 152 Lab 9 Section Number: 571

Members:

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

[1]:
```
from sympy import *
from sympy.plotting import plot, plot_parametric
import matplotlib.pyplot as plt
import numpy as np
```

0.1.1 Question 1

1a

[2]:
```
n, x = symbols("n x")
f = ((-1) ** (n + 1) * (-2 + 4 * x ** 2) ** n) / (4 ** n * n)
# | a_{n+1} / a_n |
a_np1_an = abs(f.subs(n, n + 1) / f)
a_np1_an = simplify(a_np1_an)
lim = limit(a_np1_an, n, oo)
print("lim = ", lim)
```

```
lim =  Abs(2*x**2 - 1)/2
```

1b

[3]:
```
# ROC and IOC
IOC = solve(lim < 1, x)
print(f"IOC/endpoints: {IOC}")
print(f"ROC: {sqrt(6) / 2}")

# check if IOC are in the interval
print(f"Lower endpoint: {-sqrt(6) / 2}")
print(f"Is it in the IOC? {-sqrt(6) / 2 < sqrt(6) / 2 and -sqrt(6) / 2 >␣
↪-sqrt(6) / 2}")
```

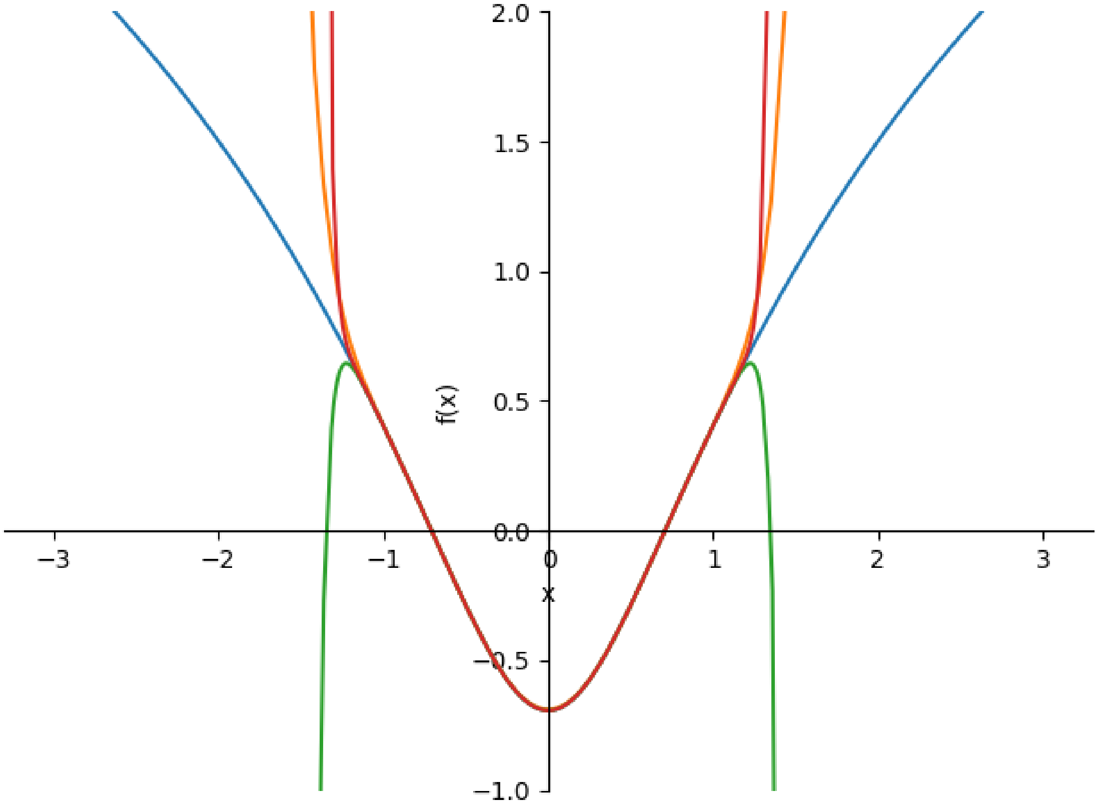
```
print(f"Upper endpoint: {sqrt(6) / 2}")
print(f"Is it in the IOC? {sqrt(6) / 2 < sqrt(6) / 2 and sqrt(6) / 2 > -sqrt(6)␣
↪/ 2}")
```

```
IOC/endpoints: (-sqrt(6)/2 < x) & (x < sqrt(6)/2)
ROC: sqrt(6)/2
Lower endpoint: -sqrt(6)/2
Is it in the IOC? False
Upper endpoint: sqrt(6)/2
Is it in the IOC? False
```

1c

[4]:
```
g = ln((2 * (x) ** 2 + 1) / 2)
g = simplify(g)
s5 = sum(f.subs(n, i) for i in range(1, 6))
s10 = sum(f.subs(n, i) for i in range(1, 11))
s15 = sum(f.subs(n, i) for i in range(1, 16))

print(f"s5 = {s5.evalf()}")
print(f"s10 = {s10.evalf()}")
print(f"s15 = {s15.evalf()}")

# plot on range x = [-3, 3], y = [-1, 2]
# plot both f and g on same graph
plot(g, s5, s10, s15, (x, -3, 3), ylim=[-1, 2])
```

```
s5 = x**2 + 0.2*(x**2 - 0.5)**5 - 0.25*(x**2 - 0.5)**4 + 0.333333333333333*(x**2
- 0.5)**3 - 0.5*(x**2 - 0.5)**2 - 0.5
s10 = x**2 - 0.1*(x**2 - 0.5)**10 + 0.111111111111111*(x**2 - 0.5)**9 -
0.125*(x**2 - 0.5)**8 + 0.142857142857143*(x**2 - 0.5)**7 -
0.166666666666667*(x**2 - 0.5)**6 + 0.2*(x**2 - 0.5)**5 - 0.25*(x**2 - 0.5)**4 +
0.333333333333333*(x**2 - 0.5)**3 - 0.5*(x**2 - 0.5)**2 - 0.5
s15 = x**2 + 0.0666666666666667*(x**2 - 0.5)**15 - 0.0714285714285714*(x**2 -
0.5)**14 + 0.0769230769230769*(x**2 - 0.5)**13 - 0.0833333333333333*(x**2 -
0.5)**12 + 0.0909090909090909*(x**2 - 0.5)**11 - 0.1*(x**2 - 0.5)**10 +
0.111111111111111*(x**2 - 0.5)**9 - 0.125*(x**2 - 0.5)**8 +
0.142857142857143*(x**2 - 0.5)**7 - 0.166666666666667*(x**2 - 0.5)**6 +
0.2*(x**2 - 0.5)**5 - 0.25*(x**2 - 0.5)**4 + 0.333333333333333*(x**2 - 0.5)**3 -
0.5*(x**2 - 0.5)**2 - 0.5
```



[4]: <sympy.plotting.plot.Plot at 0x7f1c67446ec0>

0.1.2 Question 2

2a

[5]:
```
x, n, t = symbols("x n t")
f = (((-1) ** n) * pi ** (1 / 2) * x ** (2 * n + 1)) / ((2 * n + 1) *␣
↪factorial(n))

ans1 = (f.subs(n, n + 1) / f).simplify()
print(abs(ans1))
print(limit(abs(ans1), n, oo))
```

```
Abs(x**2*(2*n + 1)/((n + 1)*(2*n + 3)))
0
```

2b

[6]:
```
print("this equation converges over all values of x")
```

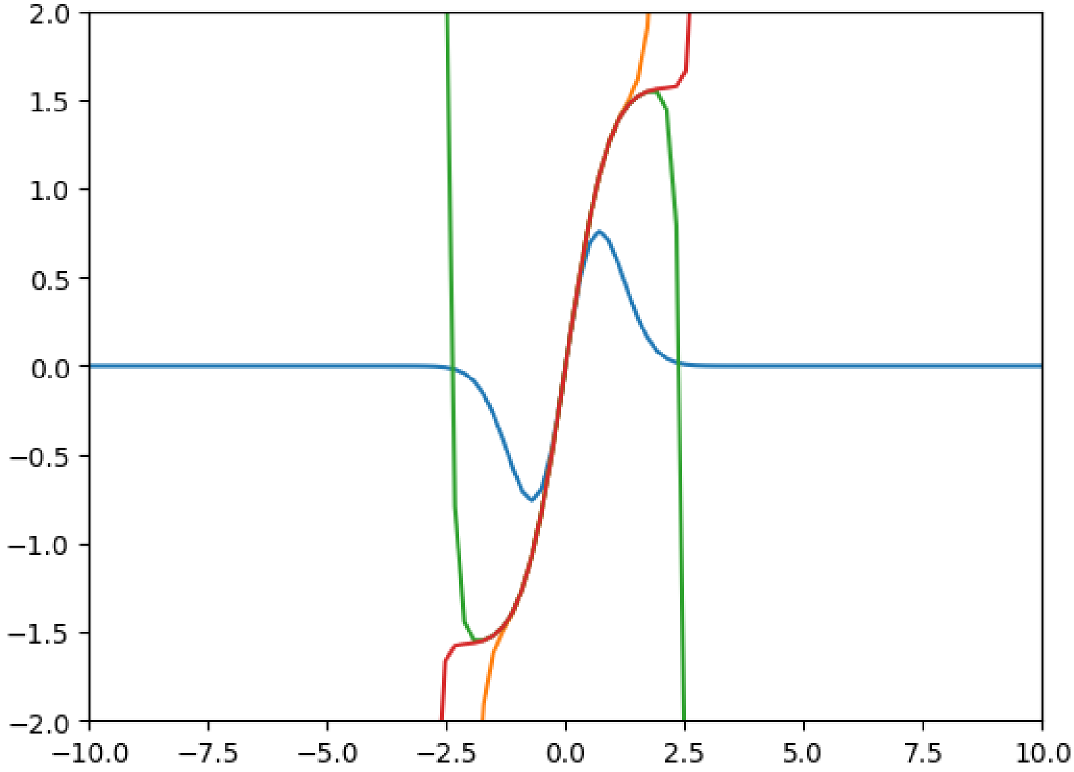
```
this equation converges over all values of x
```

2c

[7]:
```
x, n, t = symbols("x n t")
list = []
ft = pi ** (1 / 2) * (integrate(exp(-(t ** 2)), (t)))

fx = np.linspace(-10, 10, 100)
for x in fx:
    y = pi ** (1 / 2) * (integrate(exp(-(x ** 2)), (t, 0, x)))
    list.append(y)
plt.plot(fx, list)
plt.xlim(-10, 10)
plt.ylim(-2, 2)
# plotting s5,s10,s15
def sum_of_series(x):
    s = 0
    for n in range(0, 5):
        a_n = ((-1) ** n * sqrt(pi) * x ** (2 * n + 1)) / ((2 * n + 1) *␣
 ↪factorial(n))
        s += a_n
    return s


sum_of_series(np.linspace(-10, 10, 100))
plt.plot(fx, sum_of_series(np.linspace(-10, 10, 100)))


def sum_of_series(x):
    s = 0
    for n in range(0, 10):
        a_n = ((-1) ** n * sqrt(pi) * x ** (2 * n + 1)) / ((2 * n + 1) *␣
 ↪factorial(n))
        s += a_n
    return s


plt.plot(fx, sum_of_series(np.linspace(-10, 10, 100)))


def sum_of_series(x):
    s = 0
    for n in range(0, 15):
        a_n = ((-1) ** n * sqrt(pi) * x ** (2 * n + 1)) / ((2 * n + 1) *␣
 ↪factorial(n))
        s += a_n
    return s
```

```
plt.plot(fx, sum_of_series(np.linspace(-10, 10, 100)))
```

[7]: [<matplotlib.lines.Line2D at 0x7f1c64c20490>]



2d

[8]:
```
x = symbols("x")
s = 0
for i in range(0, 100):
    s += f.subs([(x, 5), (n, i)])
print(float(s))
print(float(pi / 2))
print("the answer is nearly identical to pi/2")
```

```
1.5707963267924816
1.5707963267948966
the answer is nearly identical to pi/2
```

0.1.3 Question 3

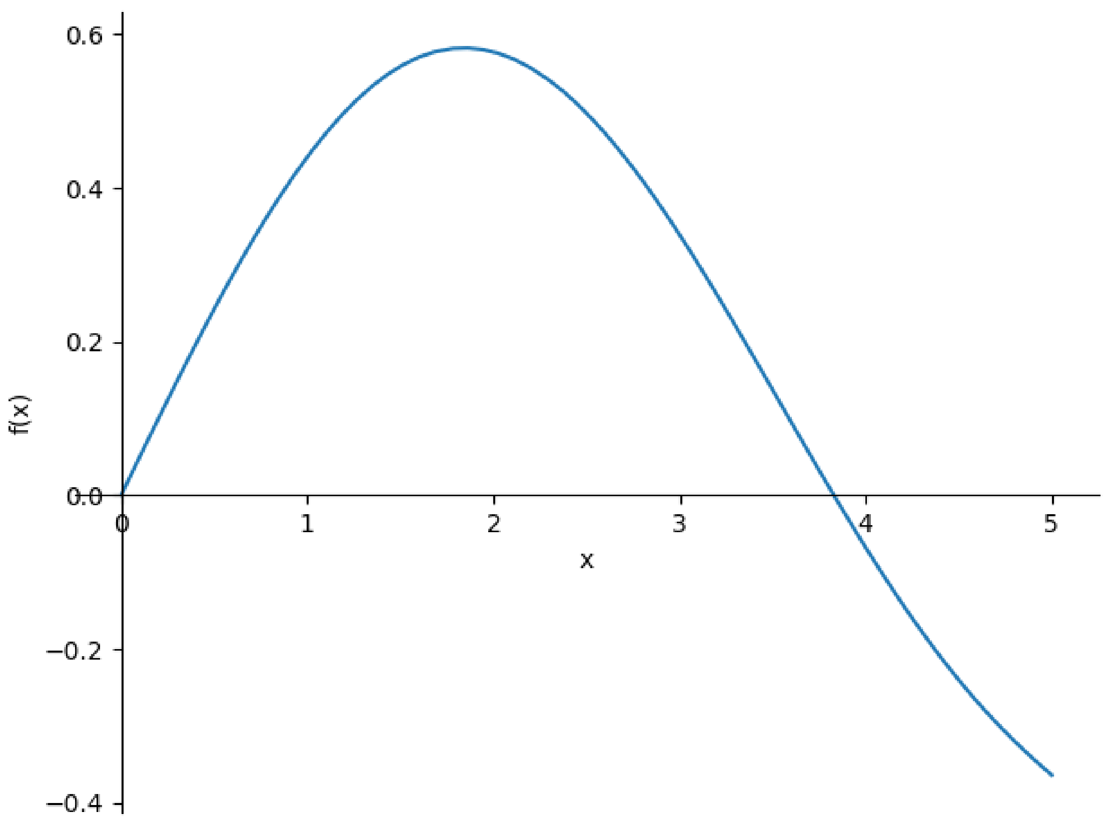
3a

[9]:
```
n = symbols("n", integer=True)
x = symbols("x", real=True)
a = (((-1) ** n) * (x ** (2 * n + 1))) / (factorial(n) * factorial(n + 1) * (2
 ↪** (2 * n + 1)))
RatioTest = abs(a.subs(n, n + 1) / a)
print("The Ratio Test is", RatioTest, "which simplifies to", RatioTest.
 ↪simplify())
L = limit(RatioTest, n, oo)  # NOTE that since there are TWO symbolic
 ↪variables, you HAVE to specify which -> oo
print("The limit of the Ratio Test is", L, "which is always < 1 so the ROC is
 ↪oo and the interval is (-oo,oo)")
```

```
The Ratio Test is 2**(-2*n - 3)*2**(2*n + 1)*Abs(x)**(-2*n - 1)*Abs(x)**(2*n +
3)*Abs(factorial(n)/factorial(n + 2)) which simplifies to x**2/(4*Abs((n + 1)*(n
+ 2)))
The limit of the Ratio Test is 0 which is always < 1 so the ROC is oo and the
interval is (-oo,oo)
```

3b

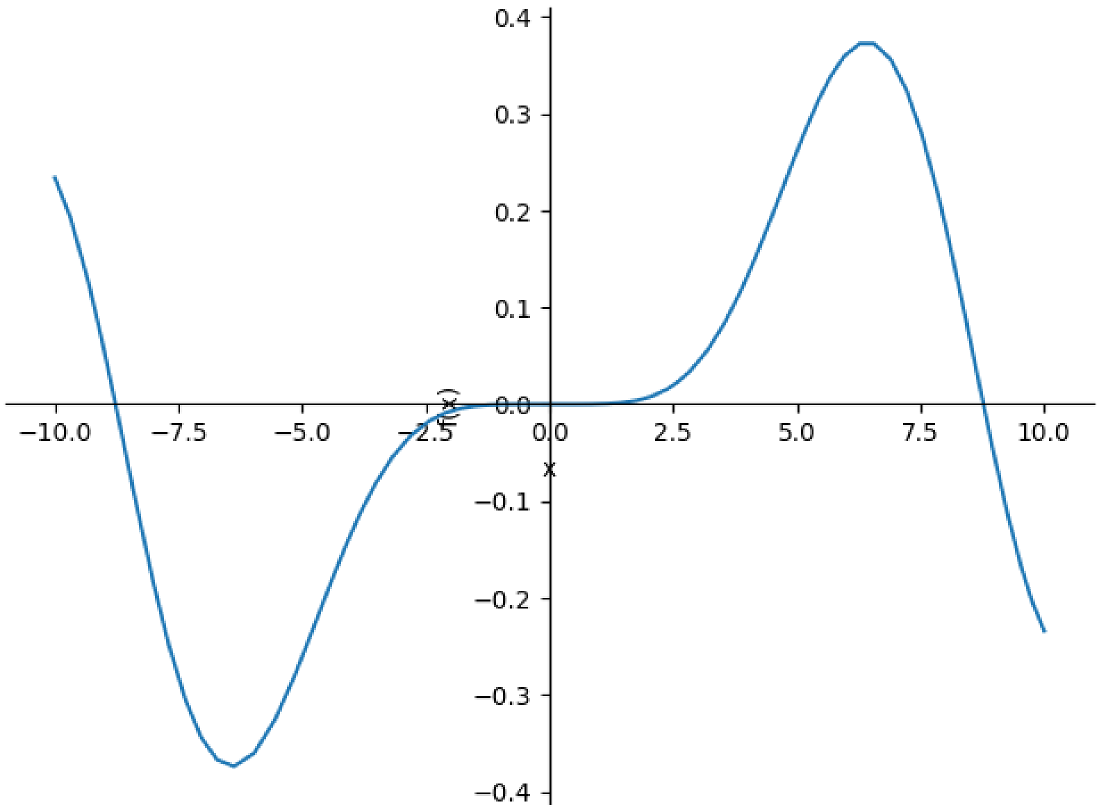
[10]:
```
a0to5 = [a.subs({n: i}) for i in range(6)]
S5 = sum(a0to5)
plot(S5, (x, 0, 5))
```



[10]: <sympy.plotting.plot.Plot at 0x7f1c64854700>

3c

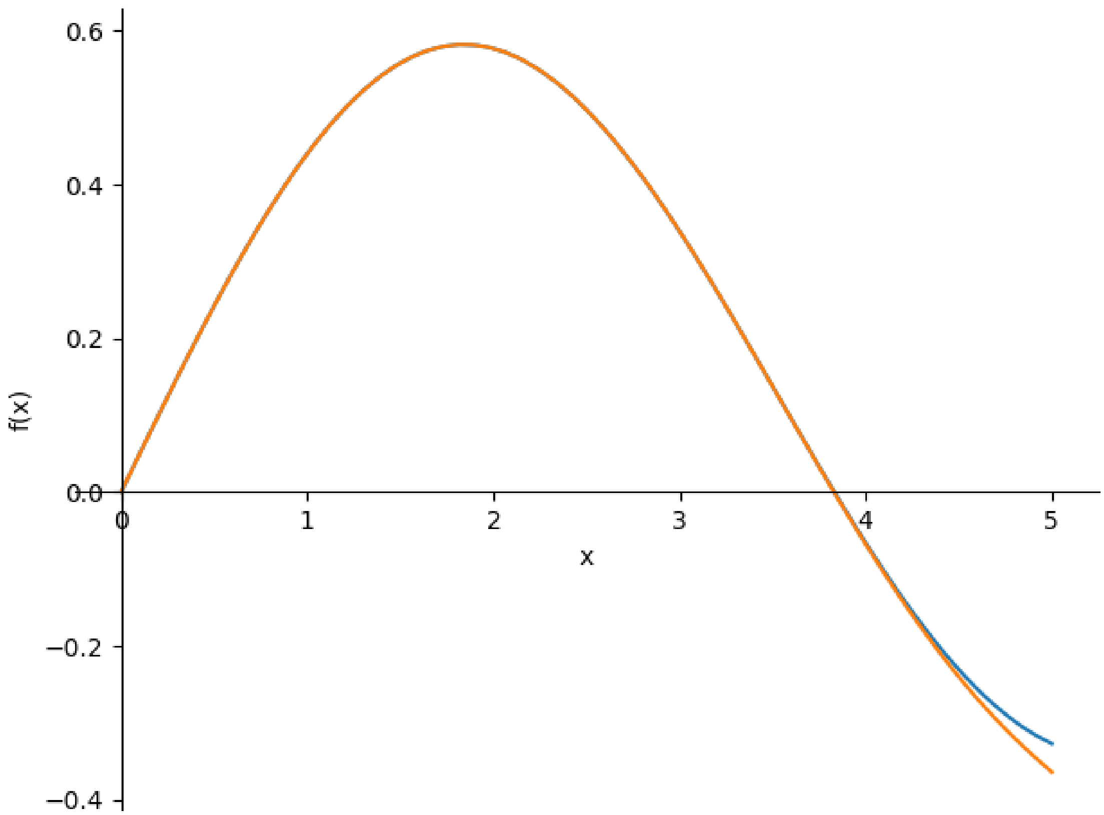
[11]:
```
J5 = besselj(5, x)
plot(J5)
```



[11]: <sympy.plotting.plot.Plot at 0x7f1c648425c0>

3d

[12]:
```
J1 = besselj(1, x)
a0to5 = [a.subs({n: i}) for i in range(6)]
S5 = sum(a0to5)
plot(J1, S5, (x, 0, 5))
```



[12]: <sympy.plotting.plot.Plot at 0x7f1c672371c0>