Math 152 – Python Lab 8

April 11, 2023

0.1 MATH 152 Lab 8

MATH 152 Lab 8 Section Number: 571

Members:

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

[1]:

```python
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]:

```python
# series from 1 to inf
# of (n^50*50^n)/(n!)

# the first 5 terms
n = symbols("n")
running_sum = 0
func = (n ** 50 * 50 ** n) / (factorial(n))

for i in range(1, 6):
    x = func.subs(n, i).evalf()
    running_sum += x
    print(f"For n={i} the value is {x} and the running sum is {running_sum}")

print("The series looks like its going to some very value 2E41")
```

```
For n=1 the value is 50.0000000000000 and the running sum is 50.0000000000000
For n=2 the value is 1.40737488355328E+18 and the running sum is
1.40737488355328E+18
For n=3 the value is 1.49562080769136E+28 and the running sum is
1.49562080783210E+28
For n=4 the value is 3.30117343809435E+35 and the running sum is
```

```
3.30117358765643E+35
For n=5 the value is 2.31296463463574E+41 and the running sum is
2.31296793580933E+41
The series looks like its going to some very value 2E41
```

1b

[3]:

```python
n = symbols("n")
an = (n ** 50 * 50 ** n) / factorial(n)
an_plus1 = (n + 1) ** 50 * 50 ** (n + 1) / factorial(n + 1)

print(f"an={an}")
print(f"an+1={an_plus1}")

func = (an_plus1 / an).simplify()

print(f"an+1/an={func}")

lim = limit(func, n, oo).doit().evalf()

print(f"The limit is {lim}")

lim = abs(lim)

print(f"The absolute value of the limit is {lim}")

print(f"Since the limit is 0 it converges absolutely : if L < 1 then the series converges absolutely")
```

```
an=50**n*n**50/factorial(n)
an+1=50**(n + 1)*(n + 1)**50/factorial(n + 1)
an+1/an=50*(n + 1)**49/n**50
The limit is 0
The absolute value of the limit is 0
Since the limit is 0 it converges absolutely : if L < 1 then the series
converges absolutely
```

1c

[4]:

```python
print("THe answer to part b tells us that the series converges absolutely")
```

```
THe answer to part b tells us that the series converges absolutely
```

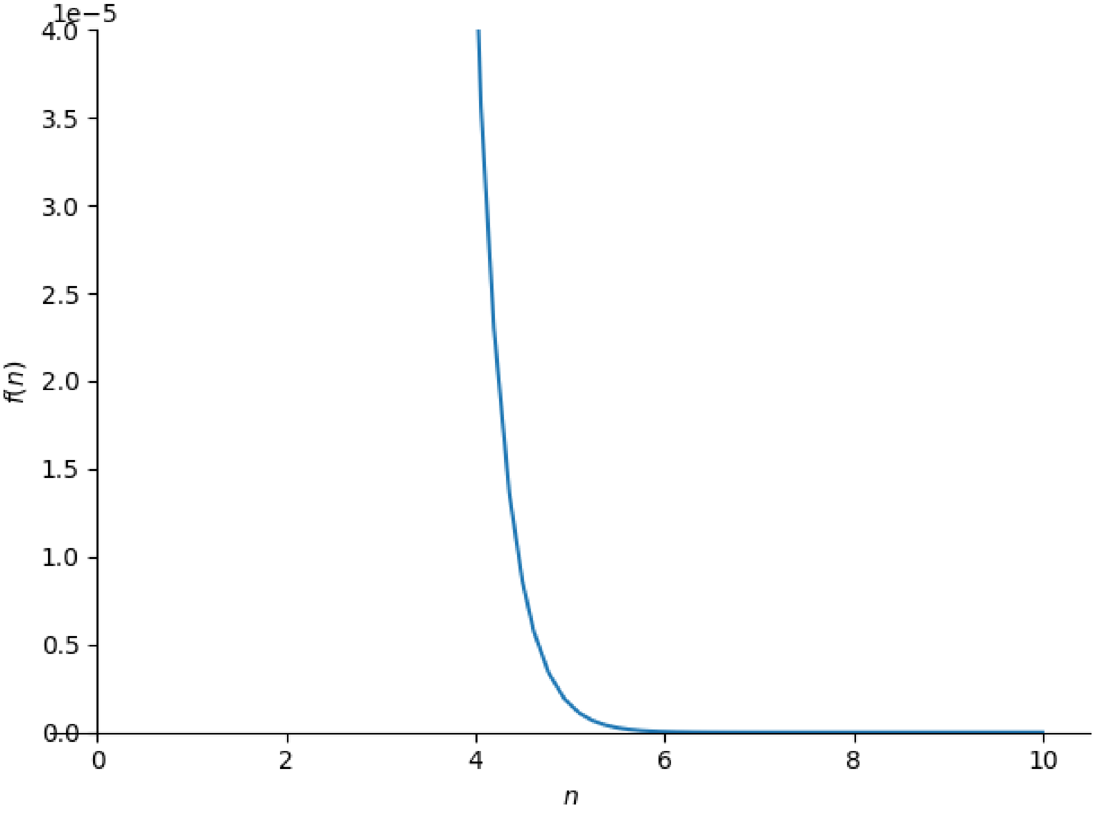
0.1.2 Question 2

2a

[5]:

```python
n = symbols("n")
an = n ** 8 * exp(-5 * n)
sn = integrate(an, (n, n, oo))
```

```
plot(sn, 0.0001, (n, 0, 10), ylim=(0, 0.00004))
print("To be within 0.00001 there needs to be 4 terms")
```



```
To be within 0.00001 there needs to be 4 terms
```

2b

[6]:
```
a = nsolve(sn - 0.0001, 3)
print(f"Using nsolve the answer is {a}")
```

```
Using nsolve the answer is 3.72617427704785
```

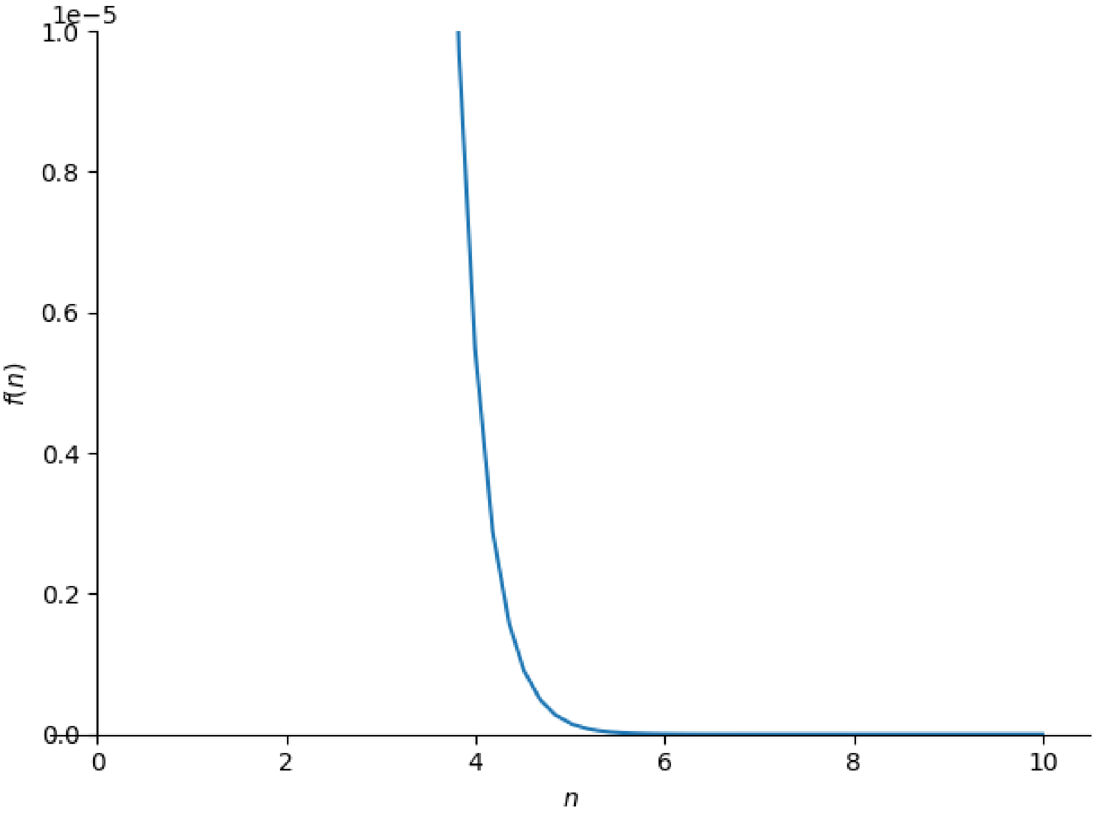
2c

[7]:
```
sum_ = 0
for i in range(1, 5):
    sum_ += an.subs(n, i).evalf()

print(f"The sum of the series is {sum_}")
```

```
The sum of the series is 0.0205024339068942
```

2d

```
[8]: a1 = (-1) ** n * an
     as_ = an.subs(n, n + 1)
     plot(as_, (n, 0, 10), ylim=(0, 0.00001))
     print("It requires 3 terms to reach 0.001")
```



```
It requires 3 terms to reach 0.001
```

2e

```
[9]: min_ = nsolve(as_ - 0.0001, 3)
     print(f"With nsolve the answer is {min_}")
```

```
With nsolve the answer is 3.09942143489560
```

2f

```
[10]: sum_=0
      for i in range(1,5):
          sum_+=a1.subs(n,i).evalf()

      print(f"The sum of the series is {sum_}")
```

```
The sum of the series is 0.00301248965909831
```

0.1.3 Question 3

3a

[11]:
```
n = symbols("n")
x = symbols("x")
an = factorial(n) ** 2 / (factorial(2 * n)) * x ** n
an1 = factorial(n + 1) ** 2 / (factorial(2 * (n + 1))) * x ** n
value = abs(an1 / an).simplify()
print(value)
print(limit(value, n, oo))
```

```
Abs((n + 1)/(2*n + 1))/2
1/4
```

3b

[12]:
```
print("the radius of convergence is 4 and the endpoints are -4 and 4")
```

```
the radius of convergence is 4 and the endpoints are -4 and 4
```

3c

[13]:
```
print(Sum(an.subs(x, 4), (n, 0, oo)).doit())
```

```
oo
```

3d

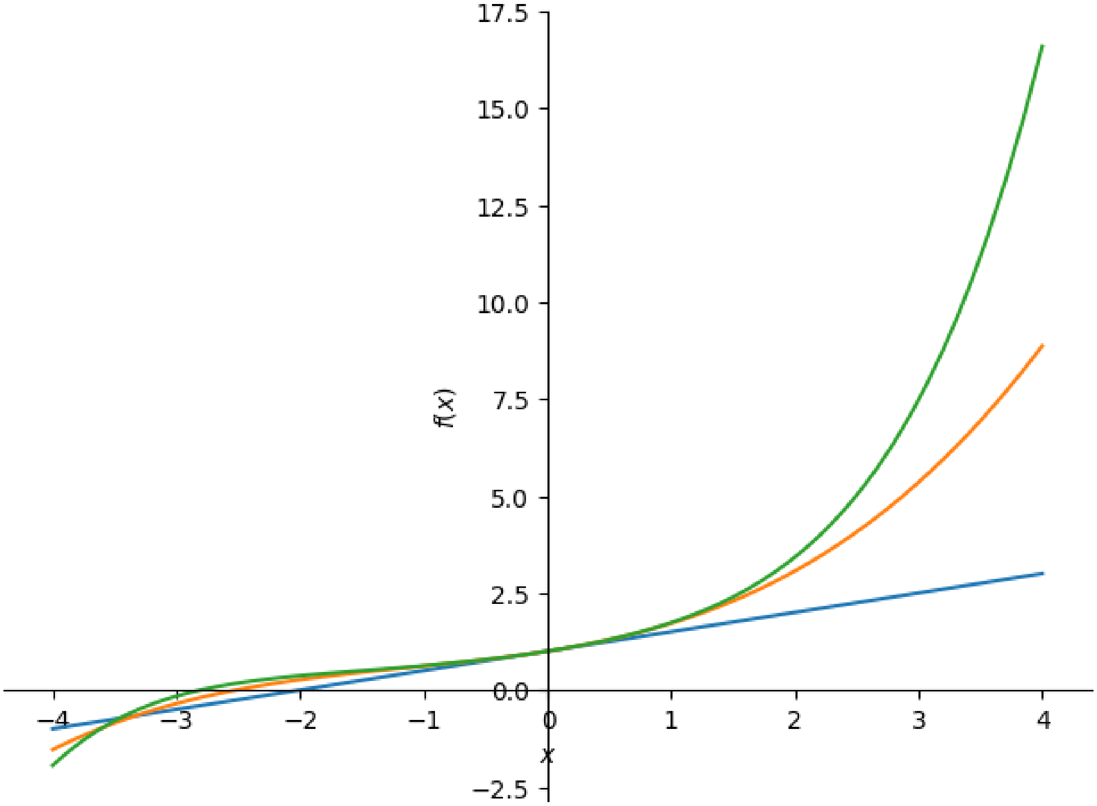
[14]:
```
print(an.subs([(x, 4), (n, 10)]).evalf())
print()
print(an.subs([(x, 4), (n, 100)]).evalf())
print()
print(an.subs([(x, 4), (n, 1000)]).evalf())
print()
print(an.subs([(x, 4), (n, 10000)]).evalf())

print("this series is not converging")
```

```
5.67546385503042

17.7467079428307

56.0569188406160

177.247600671712
this series is not converging
```

3e

[15]:
```
p1 = plot(Sum(an, (n, 0, 1)), (x, -4, 4), show=False)
p1.extend(plot(Sum(an, (n, 0, 3)), (x, -4, 4), show=False))
```

```
p1.extend(plot(Sum(an, (n, 0, 5)), (x, -4, 4), show=False))
p1.show()
```


17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
f(x)
−4
−3
−2
−1
0
1
2
3
4
x