

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

POWER SERIES IN PYTHON

A power series is an infinite series whose terms include powers of $x-a$ (think of a power series as an infinite-degree polynomial). We are interested in the values of x for which the series converges (the domain of the function, also called the **interval of convergence**). This week in lecture, you will learn that the interval of convergence of a power series is always one of three things:

- 1) all real numbers: $(-\infty, \infty)$
- 2) only at $x=a$
- 3) an interval $(a-r, a+r)$ for some nonzero number r (endpoints may be included or not). In this case, we call r the **radius of convergence**.

Here are the strategies for finding the radius and interval of convergence of a power series:

- 1) Apply the Ratio Test:
 - a) find $|a_{n+1}/a_n|$ (Key Python commands: **subs** and possibly **simplify**)
 - b) take the limit (Key Python command: **limit**)
- 2) If there are endpoints:
 - a) Determine what they are (if not obvious, solve the inequality $LIMIT < 1$ for x . Key Python command: **reduce_inequalities**)
 - b) Substitute each endpoint into the power series (Key Python command: **subs...for x!**)
 - c) Simplify the expression (Key Python command: **simplify**) and see if the series converges or not (can usually tell by observation. If not, apply an appropriate convergence test from the list to determine it)

EXAMPLES:

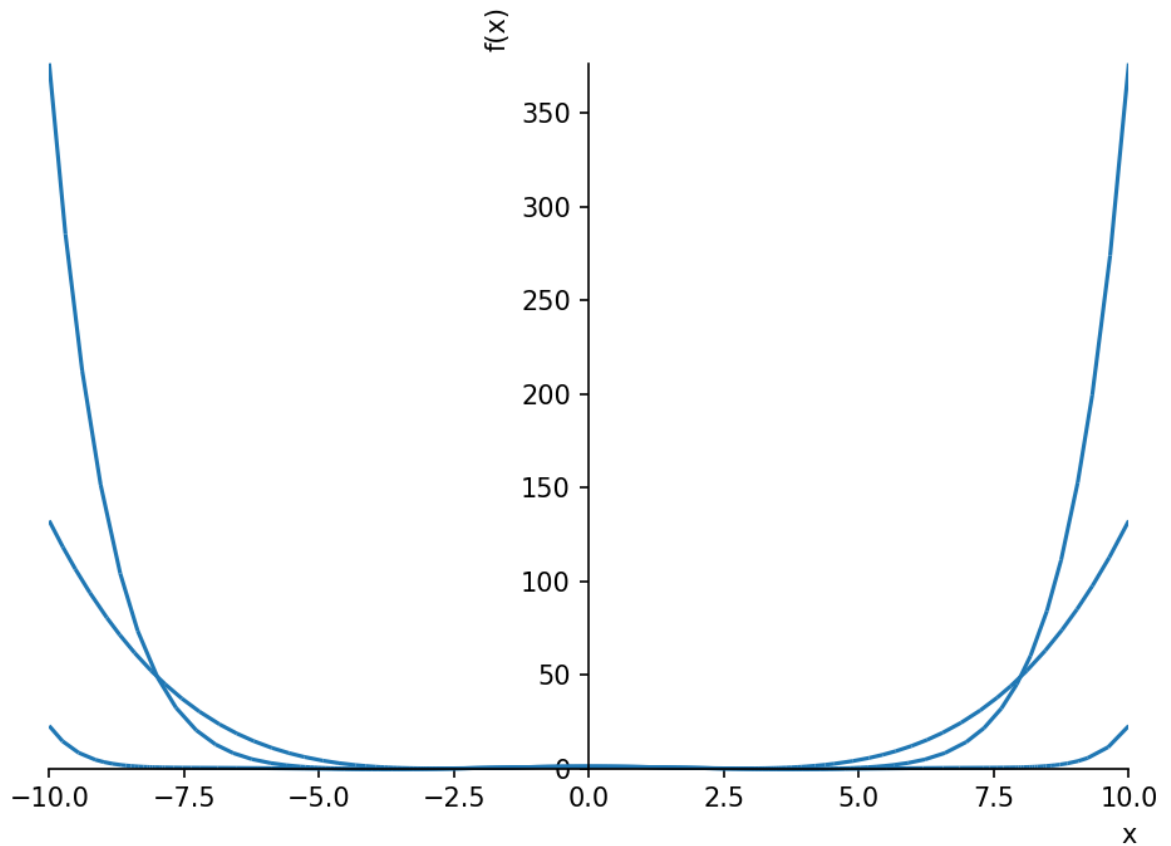
- 1) Determine the radius and interval of convergence of the following series. Then plot the stated partial sums of the series on the same graph (use your interval of convergence to determine a plot domain).
 - a) $\sum_{n=0}^{\infty} ((-1)^n (x^{2n})) / (2^{2n} (n!)^2)$; plot S_2 , S_4 , and S_8 and also plot the Bessel-0 function $\text{besselj}(0,x)$
 - b) $\sum_{n=0}^{\infty} ((-1)^n (x-2)^{2n}) / 2^{2n}$; plot S_2 , S_4 , and S_8 and also plot $f(x)=4/(x^2-4x+8)$

```
In [2]: matplotlib notebook
```

```
In [3]: # Step 1: Apply the Ratio Test
n=symbols('n',integer=True)
x=symbols('x',real=True)
a=(-1)**n*(x**(2*n))/(2**(2*n)*(factorial(n)**2)
# Step 1a
RatioTest=abs(a.subs(n,n+1)/a)
print('The Ratio Test is',RatioTest,'which simplifies to',RatioTest.simplify())
# Step 1b
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)')
# Since there are no endpoints, we can skip Step 2
```

The Ratio Test is $2^{2n}(-2n - 2)x^{2n}x^{2n+2} \frac{\text{Abs}(\text{factorial}(n)^2/\text{factorial}(n+1)^2)}$ which simplifies to $x^2/(4(n+1)^2)$
 The limit of the Ratio Test is θ which is always < 1 so the ROC is ∞ and the interval is $(-\infty, \infty)$

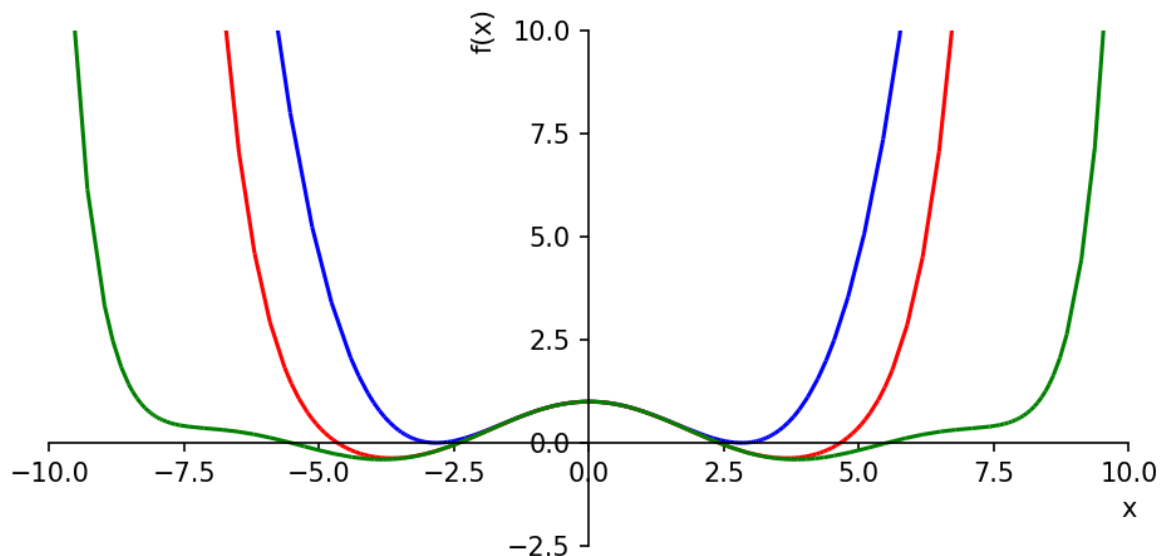
```
In [4]: a0to2=[a.subs(n,i) for i in range(3)] #Remember range(b) is from 0 to b-1
        !!!!
        S2=sum(a0to2)
        a0to4=[a.subs(n,i) for i in range(5)]
        S4=sum(a0to4)
        a0to8=[a.subs(n,i) for i in range(9)]
        S8=sum(a0to8)
        #Now plot them together. Let's use [-10 to 10] since domain is all real nu
        mbers
        plot((S2,(x,-10,10)),(S4,(x,-10,10)),(S8,(x,-10,10)))
```



Out[4]: <sympy.plotting.plot.Plot at 0x7719af0>

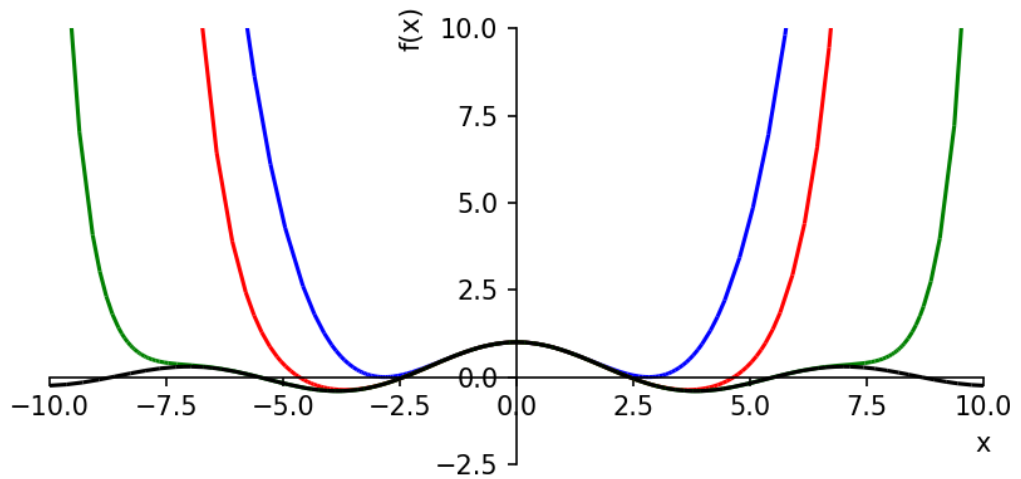
In [5]: matplotlib notebook

```
In [6]: # Maybe Limit the y-range to see what's going on
# I'm also changing colors, but that is not required for you. In that case,
# drop the "show=False" and all remaining lines
fplots=plot((S2,(x,-10,10)),(S4,(x,-10,10)),(S8,(x,-10,10)),ylim=[-10,10],s
how=False)
fplots[0].line_color='b'
fplots[1].line_color='r'
fplots[2].line_color='g'
fplots.show()
```



```
In [12]: matplotlib notebook
```

```
In [14]: # And now we add the Bessel- $\theta$  function, shown here in black
fTplots=plot((S2,(x,-10,10)),(S4,(x,-10,10)),(S8,(x,-10,10)),(besselj( $\theta$ ,x),
(x,-10,10)),ylim=[-10,10],show=False)
fTplots[0].line_color='b'
fTplots[1].line_color='r'
fTplots[2].line_color='g'
fTplots[3].line_color='k'
fTplots.show()
```



Now for example 2. Same idea, but we have a bit more to consider on this one...

```
In [15]: # Step 1: Apply the Ratio Test
n=symbols('n',integer=True)
x=symbols('x',real=True)
# Step 1a
a=(-1)**n*((x-2)**(2*n))/2**(2*n)
RatioTest=abs(a.subs(n,n+1)/a)
print('The Ratio Test is',RatioTest,'which simplifies to',RatioTest.simplify())
# Step 1b
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) #This doesn't look too helpful, so
    try expand, simplify, or factor
print('The limit of the Ratio Test is',L.factor()) #The others didn't work,
    but this one did
```

The Ratio Test is $2^{2n} \cdot 2^{2n} \cdot (-2n - 2) \cdot (x - 2)^{-2n} \cdot (x - 2)^{2n} \cdot (2n + 2)$
 which simplifies to $(x - 2)^{2/4}$
 The limit of the Ratio Test is $x^{2/4} - x + 1$
 The limit of the Ratio Test is $(x - 2)^{2/4}$

```
In [16]: # It is not obvious which values of x make this ratio < 1, so proceed with
    Step 2a
interval=reduce_inequalities(L<1)
print('The interval of convergence so far is',interval,'which means the radius
    of convergence is 2')
```

The interval of convergence so far is $(0 < x) \ \& \ (x < 4)$ which means the radius of convergence is 2

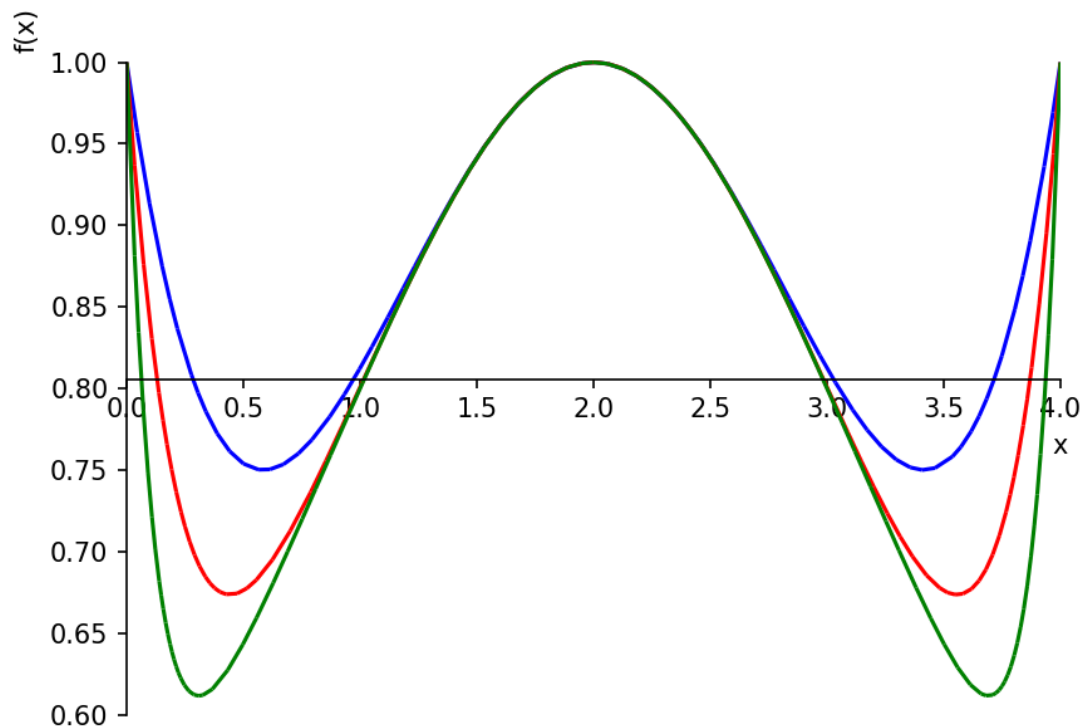
Notice that because of the $(x-2)^2$ in the power series, the center of the series is 2, which is also the center of the interval of convergence!

We now proceed with Step 2b and 2c for each endpoint:

```
In [17]: # Step 2b with x=0
a_left=a.subs(x,0)
print('The terms of the series are',a_left,', so this series diverges when
    x=0') #Wasn't a need for Step 2c
# Step 2b with x=4
a_right=a.subs(x,4)
print('The terms of the series are',a_right,', so this series diverges when
    x=4') #Again no need for Step 2c
```

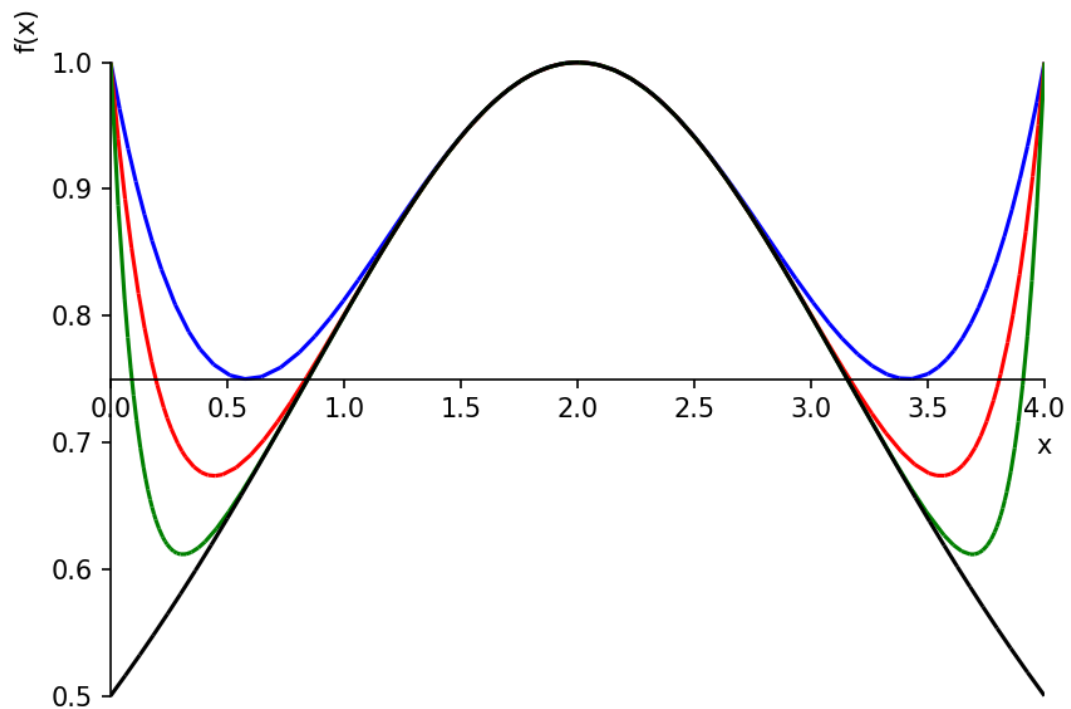
The terms of the series are $(-1)^n$, so this series diverges when $x=0$
 The terms of the series are $(-1)^n$, so this series diverges when $x=4$

```
In [19]: # Now we plot the partial sums and the given function
a0to2=[a.subs({n:i}) for i in range(3)] #Remember range(b) is from 0 to b-1 !!!!
S2=sum(a0to2)
a0to4=[a.subs({n:i}) for i in range(5)]
S4=sum(a0to4)
a0to8=[a.subs({n:i}) for i in range(9)]
S8=sum(a0to8)
#Now plot them together. Let's use [0 to 4] as determined by the Interval
of Convergence
fplots=plot((S2,(x,0,4)),(S4,(x,0,4)),(S8,(x,0,4)),show=False)
fplots[0].line_color='b'
fplots[1].line_color='r'
fplots[2].line_color='g'
fplots.show()
```



```
In [18]: matplotlib notebook
```

```
In [20]: # And now we add the given function
fTplots=plot((S2,(x,0,4)),(S4,(x,0,4)),(S8,(x,0,4)),(4/(x**2-4*x+8),(x,0,4)),show=False)
fTplots[0].line_color='b'
fTplots[1].line_color='r'
fTplots[2].line_color='g'
fTplots[3].line_color='k'
fTplots.show()
```



Notice in both sets of graphs the higher the number of terms lead to a longer "interval of accuracy" (the partial sums stay close to the actual function graph over a larger domain).

In []: