

In-Service Course
(Spell I)

Venue:
IIT Gandhinagar

Study Material 

Solving Taylor Series With Python

f^n

Prepared by:
Navneet Sadh
PGT CS, KV KOKRAJHAR






A little math in the beginning.

(A lot actually)

Let's learn about the problem first. 

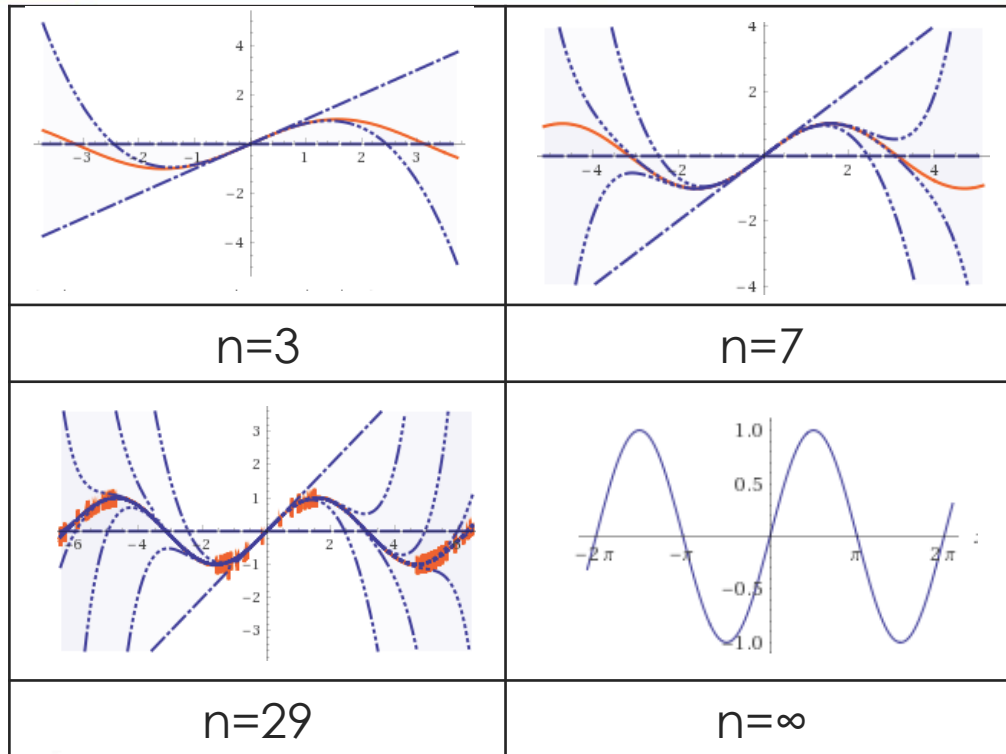


r

James Gregory	Brook Taylor	Colin Maclaurin
Did the Formulation	Formally Introduced	Used a special case
 A portrait of James Gregory, a Scottish astronomer, mathematician, and physicist. He is shown from the chest up, wearing a dark coat with a white clerical collar. He is holding a globe in his left hand.	 A portrait of Brook Taylor, an English mathematician. He is shown from the chest up, wearing a dark coat with a white collar. He has long, curly hair.	 A portrait of Colin Maclaurin, a Scottish mathematician. He is shown from the chest up, wearing a light-colored coat. He has long, curly hair. The background of the portrait appears to be a page of handwritten text.

Faces of Taylor Series

Plotting the Graph of Sin(x): With variation in no. of terms



On your left you can see various graphs. Each one of them are plotted with increasing order of approximation.
i.e. when we increase the number of terms.

By this we can conclude that graph becomes accurate when n approaches infinite.

That's why it is also called '**infinite series**'

The Equation.

Too much information?

No problem now we'll see the basic **equation**.

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$$

On the R.H.S the equation was solved up to 7th term.

..and the answer will be: 7.3809



Implementation

Let's Code. 



Calling the function sin(x): Only two inputs are needed x,n

```
sin(x,n)  
x=Phase  
n=no. of terms
```

Definition of sin(x):

```
def sin(x,n):  
    sine = 0  
    for i in range(n):  
        sign = (-1)**i  
        sine = sine +  
        ((x**(2.0*i+1))/  
        factorial(2*i+1))*sign  
    return sine
```

Definition of factorial (x):

```
def factorial(n):  
    if n > 1:  
        return n *  
        factorial(n-1)  
    return 1
```

Observations:

1. For loop will run n no. of times
2. Power of x will increase by 2 i.e. 1,3,5,7.....
3. Each new term will have opposite sign



Practice

Your Turn. 

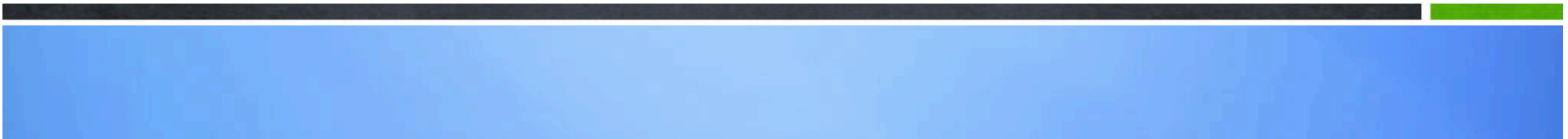


Solve the following series using Python Function

1. e^x

2. $\tan^{-1}x$

3. $\cos x$





Thank You.