

Taylor Series Expansion

Lab Report for Assignment No. 1

College Roll No : 2020PHY1097

University Roll No : 20068567031

Name : Kabir Sethi

Unique Paper Code : 32221401

Paper Title : Mathematical physics III Lab

Course and Semester : B.Sc.(H) Physics Sem IV

Due Date : January 15 , 2022

Date of Submission : January 15 , 2022

Lab Report File Name :
2020PHY1097_A1.pdf,2020PHY1097_A1.py

Partner's Name : Pawanpreet Kaur

Partner's College Roll No. : 2020PHY1092

1 Theory

1.1 Taylor Series

Taylor series of a function is an infinite sum of terms that are expressed in terms of the function's derivatives at a single point. The Taylor Series expansion of a function $f(x)$ about a point $x = a$ is given by

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots + \frac{f^n(a)}{n!}(x-a)^n$$

The above expansion in summation notation is

$$f(x) = \sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n$$

The Taylor series can also be written for two variables. The following series is the 2nd order Taylor Series for two variables is

$$f(x,y) \approx Q(x,y) = f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b) + f_{xx}(a,b)^2(x-a)^2 + f_{xy}(a,b)(x-a)(y-b) + f_{yy}(a,b)^2(y-b)^2$$

1.2 Radius of Convergence

The **Radius of convergence** of a power series is the radius of the largest disk at the center of the series in which the series converges. It is either a non-negative real number or ∞ . When it is positive, the power series converges absolutely and uniformly on compact sets inside the open disk of radius equal to the radius of convergence.

The Radius of Convergence for a series can be found out by ratio test. The ratio test can be performed by dividing the $n + 1^{th}$ and n^{th} terms of the series and applying limit on it. The following expression is for finding the radius of convergence by ratio test:

$$\lim_{n \rightarrow \infty} \frac{n+1^{th} Term}{n^{th} Term}$$

1.3 Maclaurin series

A Maclaurin series is a Taylor series expansion of a function about 0, i.e. $a=0$

$$f(x) = f(0) + f'(0)(x-0) + \frac{f''(0)}{2!}(x-0)^2 + \frac{f'''(0)}{3!}(x-a)^3 + \dots + \frac{f^n(0)}{n!}(x-0)^n$$

The above expansion also can be written as

$$f(x) = \sum_{n=0}^{\infty} \frac{f^n(0)}{n!}(x-0)^n$$

The Maclaurin Series for sin x can be derived as

So $f(x) = \sin(x)$ and as for a Maclaurin Series the reference point is 0 i.e, $a=0$, then

$$\begin{aligned} f(0) &= \sin(0) = 0 \\ f'(0) &= \cos(0) = 1 \\ f''(0) &= -\sin(0) = 0 \end{aligned}$$

and so on.

Therefore the Maclaurin Series for sin x is :

$$\sin x = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

or

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

The **Radius of Convergence** for the series can be found out by the ratio test.

So the ratio test is

$$\begin{aligned} \lim_{n \rightarrow \infty} \left[\frac{(-1)^k}{2(k+1)+1!} x^{2(k+1)+1} / \frac{(-1)^k}{2k+1!} x^{2k+1} \right] &= \lim_{n \rightarrow \infty} \frac{2k+1!}{2k+3!} |x|^2 \\ &= \lim_{n \rightarrow \infty} \frac{1}{(2k+3)(2k+2)} |x|^2 = 0 \end{aligned}$$

Because this limit is zero for all real values of x, the radius of convergence of the expansion is the set of all real numbers.

Similarly, **The Maclaurin Series for cos(x) can be derived as**

$$f(x) = \cos(x), a = 0 \text{ then}$$

$$f(0) = \cos(0) = 1$$

$$f'(0) = -\sin(0) = 0$$

$$f''(0) = -\cos(0) = -1, \text{ so on.}$$

Therefore the Maclaurin Series for cos x is :

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

or

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} (x)^{2n}$$

The **Radius of Convergence** for the series can be found out by the ratio test.

So the ratio test is

$$\begin{aligned}\lim_{n \rightarrow \infty} \left[\frac{(-1)^{k+1}}{2(k+1)!} x^{2(k+1)} / \frac{(-1)^k}{2k!} x^{2k} \right] &= \lim_{n \rightarrow \infty} \frac{2k!}{2k+2!} |x|^2 \\ &= \lim_{n \rightarrow \infty} \frac{1}{(2k+1)(2k+2)} |x|^2 = 0\end{aligned}$$

Because this limit is zero for all real values of x , the radius of convergence of the expansion is the set of all real numbers.

The Maclaurin Series for e^x can be derived as

So $f(x) = e^x$ and as for a Maclaurin Series the reference point is 0 i.e, $a=0$, then

$$\begin{aligned}f(0) &= e^0 = 1 \\ f'(0) &= e^0 = 1 \\ f''(0) &= e^0 = 1\end{aligned}$$

and so on.

Therefore the Maclaurin Series for e^x is :

$$\begin{aligned}e^x &= 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \\ &\text{or} \\ e^x &= \sum_{n=0}^{\infty} \frac{(x)^n}{(n)!}\end{aligned}$$

The **Radius of Convergence** for the series can be found out by the ratio test.

So the ratio test gives us

$$\lim_{n \rightarrow \infty} \left[\frac{(x)^{k+1}}{(k+1)!} x / \frac{(x)^k}{k!} \right] = \lim_{n \rightarrow \infty} \frac{|x|}{k+1} = 0$$

Because this limit is zero for all real values of x , the radius of convergence of the expansion is the set of all real numbers.

2 Algorithm

Algorithm 1 Exponential Function(e^x)

```

function SERIESEXP( $xvector, n$ )
    ▷  $n$  is the number of terms for iterations and  $xvector$  is a vector containing values of  $x$ 
    for which  $e^x$  has to be calculated ◁
     $y = []$  ▷ Vector to store values of  $e^x$ 
    for all  $x \in \{xvector\}$  do ▷ Create a loop to calculate for each value of  $x$ 
         $e = 0$  ▷ Declaring a variable to which summation of terms will be added
        for all  $i \in \{0, \dots, n-1\}$  do
             $e += \frac{x^i}{i!}$  ▷ Approximation is recursively added to  $e$ 
        append  $e$  in  $y$  ▷ Appending value of  $e^x$  in  $y$ 
    return  $y$  ▷ Returns vector  $y$ 

```

Algorithm 2 Sine Function ($\sin(x)$)

```

function MYSIN SERIES( $xvector, n$ )
    ▷  $n$  is the number of terms for iterations and  $xvector$  is a vector containing values of  $x$ 
    for which  $\sin(x)$  has to be calculated ◁
     $y = []$  ▷ Vector to store values of  $\sin(x)$ 
    for all  $x \in \{xvector\}$  do ▷ Create a loop to calculate for each value of  $x$ 
         $\sin = 0$  ▷ Declaring a variable to which summation of terms will be added
        for all  $i \in \{0, \dots, n-1\}$  do
             $\sin += \frac{(-1)^i x^{2i+1}}{(2i+1)!}$  ▷ Approximation is recursively added to  $\sin$ 
        append  $\sin$  in  $y$  ▷ Appending value of  $\sin(x)$  in  $y$ 
    return  $y$  ▷ Returns vector  $y$ 

```

Algorithm 3 Cosine Function ($\cos(x)$)

function MYCOSERIES($xvector, n$)

▷ n is the number of terms for iterations and $xvector$ is a vector containing values of x for which $\cos(x)$ has to be calculated ◁

$y = []$ ▷ Vector to store values of $\cos(x)$

for all $x \in \{xvector\}$ **do** ▷ Create a loop to calculate for each value of x

$cos = 0$ ▷ Declaring a variable to which summation of terms will be added

for all $i \in \{0, \dots, n-1\}$ **do**

$cos += \frac{(-1)^i x^{2i}}{(2i)!}$ ▷ Approximation is recursively added to cos

append cos in y

return y ▷ Appending value of $\cos(x)$ in y
▷ Returns vector y

3 Discussion

fig(a) in figure [1] and [2] shows the $\sin(x)$ vs x plot and $\cos(x)$ vs x plot respectively for x in the range $[-2\pi, 2\pi]$. As we increase the number of terms in the approximation the value of our function reach almost equal to the values of inbuilt functions. .Numpy's inbuilt \sin and \cos function have been used as a reference for comparing the results given by series to the actual values. Uptill a certain number of terms the difference in values from our function and the inbuilt function decreases and after that even if we increase the number of terms the values are almost same. It can be understood from these plots that for x in the given range , the plots for $n = 10$ and $n = 20$ are overlapping with numpy's inbuilt functions.

fig(b) in figure [1] and [2] shows the $\sin(x)$ vs n plot and $\cos(x)$ vs n plot respectively for $x_0 = \frac{\pi}{4}$. The dashed line shows the exact value of $\sin(\frac{\pi}{4})$ and $\cos(\frac{\pi}{4})$ in figure [1] and [2] respectively. As the number of terms considered for series approximation increase, the series approximation tends to give the value close to actual value.

Ques 1 (Sine Series)

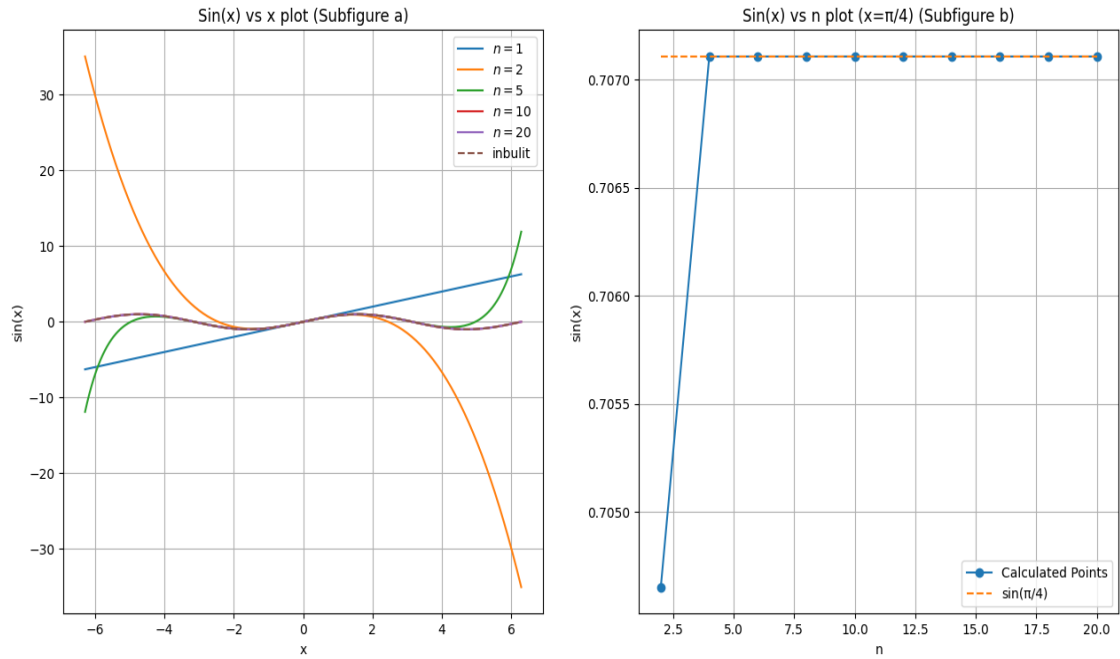


Figure 1: **(a):** $\sin(x)$ vs x plot for x in the range $[-2\pi, 2\pi]$, **(b):** $\sin(\frac{\pi}{4})$ vs n plot

Ques 1 (Cosine Series)

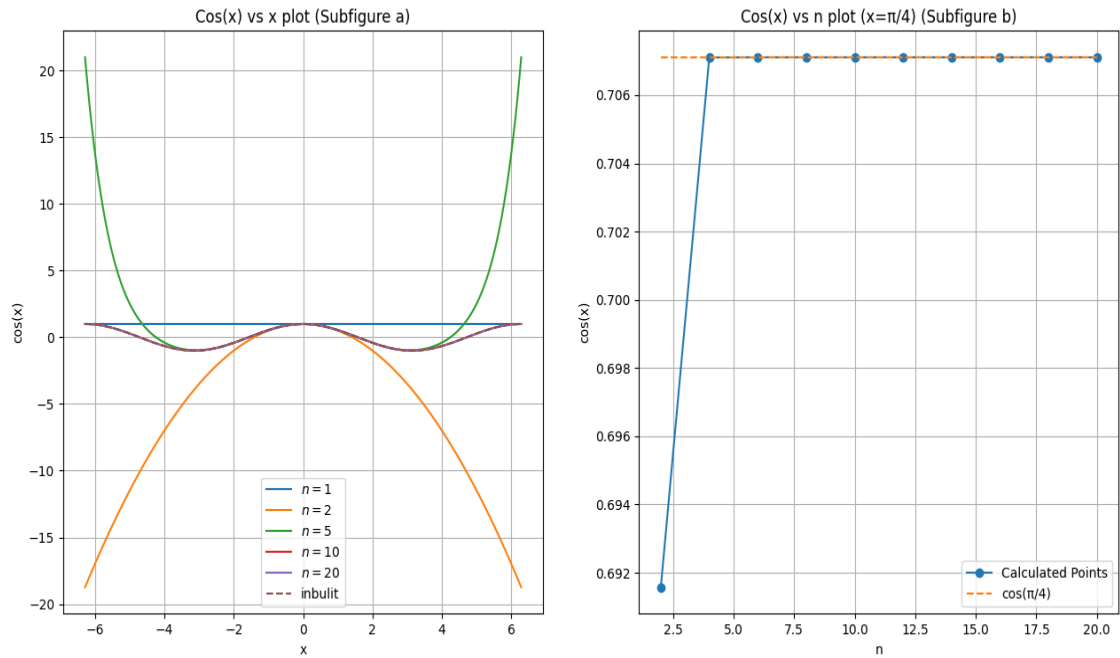


Figure 2: **(a):** $\cos(x)$ vs x plot for x in the range $[-2\pi, 2\pi]$, **(b):** $\cos(\frac{\pi}{4})$ vs n plot

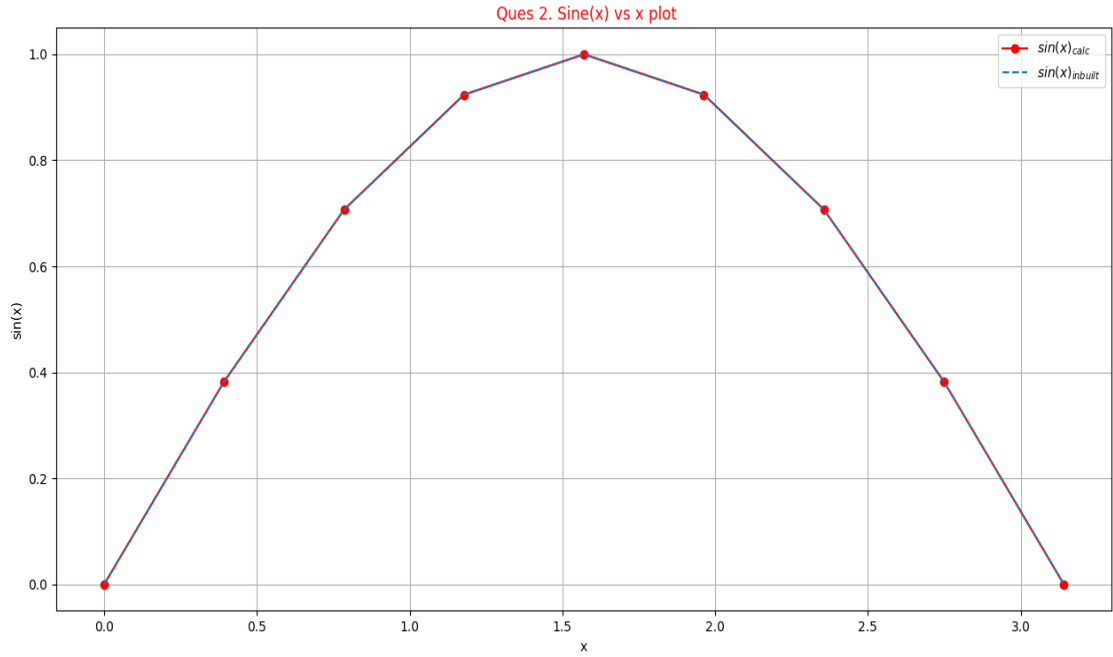


Figure 3: $\sin(x)$ vs x plot for x in the range $[0, \pi]$

Figure [3] shows the value of $\sin(x)$ obtained using a function which takes the value of tolerance from the user and returns the value obtained using series approximation accurate upto entered significant digits of the tolerance. The plot contains the values of $\sin(x)$ correct to three significant digits.

Figure [4] displays the table made using pandas dataframe containing Values of $\sin(x)$ for x in the range $[0, \pi]$ accurate to 6 significant digits evaluated using series representation along with n , the number of terms required to get this accuracy. It can be understood from the plot that as the value of x increases, more terms are required to get a value of $\sin(x)$ with a given precision.

```

Enter the tolerance value :0.1e-6
      x    sin(x)_calc    n    sin(x)_inbuilt
0  0.000000  0.000000e+00  1    0.000000e+00
1  0.392699  3.826834e-01  5    3.826834e-01
2  0.785398  7.071068e-01  6    7.071068e-01
3  1.178097  9.238795e-01  7    9.238795e-01
4  1.570796  1.000000e+00  7    1.000000e+00
5  1.963495  9.238795e-01  8    9.238795e-01
6  2.356194  7.071068e-01  9    7.071068e-01
7  2.748894  3.826834e-01  10   3.826834e-01
8  3.141593  3.328057e-16  18   1.224647e-16

```

Figure 4: Values of $\sin(x)$ accurate up to 6 significant digits evaluated using series representation along with n , the number of terms required to get this accuracy.

4 Programs

```

1 #Name= Kabir Sethi
2 #College Roll No. = 2020PHY1097
3 #University Roll No. = 20068567031
4
5 import math
6 import numpy as np
7 import matplotlib.pyplot as plt
8 import pandas as pd
9
10 #Ques 3 (a)
11 #Sine Function
12 def MySinSeries(xvector,n):
13     y=[]
14     for x in xvector:
15         sin_app=0
16         for i in range(n):
17             sin_app+=(-1)**i*x**((2*i)+1)/math.factorial((2*i)+1)
18         y.append(sin_app)
19     return y
20
21 #Ques 3 (a)
22 #Cosine Function
23 def MyCosSeries(xvector,n):
24     y=[]
25     for x in xvector:
26         cos_app=0
27         for i in range(n):
28             cos_app+=(-1)**i*x**(2*i)/math.factorial(2*i)

```

```

29         y.append(cos_app)
30     return y
31
32 #Ques 3(b)
33 def TolFunction(x_a,key):
34     x_a=np.array(x_a)
35     r=[];n_a=[]
36     tol=float(input("Enter the tolerance value :"))
37     for x in x_a:
38         n=0;sin_app=0;lis=[]
39         while True:
40             if key==0:
41                 sin_app=sin_app+(-1)**n*x**((2*n)+1)/math.factorial
42                 ((2*n)+1) #sin approximation
43             elif key==1:
44                 sin_app=sin_app+(-1)**n*x**(2*n)/math.factorial(2*n) #
45                 cos approximation
46             else:
47                 return("Wrong key entered")
48             n+=1
49             lis.append(sin_app)
50             if sin_app==0:
51                 r.append(sin_app)
52                 n_a.append(n)
53                 break
54             if len(lis)>=2:
55                 err= abs(lis[-1]-lis[-2])/abs(lis[-1])
56                 if err<tol:
57                     r.append(sin_app)
58                     n_a.append(n)
59                     break
60         return r,n_a
61
62 #Graph Function
63 def gr1(x,Tit,f,n_a,lis,inb,yl_1,t_1,s,d,p1_l,y0,yl_2,t_2):
64     #fig, (ax1, ax2) = plt.subplots(2)
65     fig, (ax1, ax2) = plt.subplots(1, 2)
66     fig.suptitle(Tit,c="r")
67     for u,v in zip(f,n_a):
68         ax1.plot(x,lis[u],label= "n=0".format(v))
69         ax1.plot(x,inb,label="inbulit",linestyle="--")
70         ax1.set(xlabel="x",ylabel=yl_1,title=t_1)
71         ax1.legend()
72         ax1.grid()
73         ax2.plot(s,y0,marker="o")
74         ax2.plot(s,d,label=p1_l,linestyle="--")
75         ax2.set(xlabel="n",ylabel=yl_2,title=t_2)
76         ax2.legend()
77         ax2.grid()
78     plt.show()
79
80 def gr2(x,y1,l1,y2,l2,xl,yl,tit):
81     plt.plot(x,y1,label=l1,c="r",marker="o")
82     plt.plot(x,y2,label=l2,linestyle="--")
83     plt.xlabel(xl)

```

```

82     plt.ylabel(y1)
83     plt.title(tit,c="r")
84     plt.legend()
85     plt.grid()
86     plt.show()
87
88 #IMPLEMENTATION
89 #Ques (a)
90 x=np.linspace(-2*np.pi,2*np.pi,200)
91 sin_in=np.sin(x)                                #Inbuilt Sin
92 cos_in=np.cos(x)                                #Inbuilt Cos
93 n_a=[1,2,5,10,20]
94 lis=[]
95 lis1=[]
96 for n in n_a:
97     lis.append(MySinSeries(x,n))
98     lis1.append(MyCosSeries(x,n))
99 f=np.arange(0,5)
100 x0=[np.pi/4]
101 s=np.arange(2,21,2)
102 y0_sin=[]
103 y0_cos=[]
104 for n in s:
105     y0_sin.append(MySinSeries(x0,n))
106     y0_cos.append(MyCosSeries(x0,n))
107 d1=np.array([np.sin(x0)]*len(s))
108 d2=np.array([np.cos(x0)]*len(s))
109
110 gr1(x,'Ques 1 (Sine Series)',f,n_a,lis,sin_in,"sin(x)","Sin(x) vs x
    plot",s,d1,"sin(\u03C0/4)",y0_sin,"sin(x)","Sin(x) vs n plot (x=\
    \u03C0/4)")
111 gr1(x,'Ques 1 (Cosine Series)',f,n_a,lis1,cos_in,"cos(x)","Cos(x) vs x
    plot",s,d2,"cos(\u03C0/4)",y0_cos,"cos(x)","Cos(x) vs n plot (x=\
    \u03C0/4)")
112
113 #Ques (b)
114 x=np.linspace(0,np.pi,9)
115 sin_in=np.sin(x)
116 e=TolFunction(x,0)
117 gr2(x,e[0],"sin(x)calc",sin_in,"sin(x)inbuilt","x","sin(x)"," Ques 2. Sine(x) vs
    x plot") #Plot
118
119
120 calc=TolFunction(x,0)
121 data={"x":x,"sin(x)_calc":calc[0],"n":calc[1],"sin(x)_inbuilt":sin_in}
    #Table
122 print(pd.DataFrame(data))

```

Source Code 1: Python Program