Numerical analysis of Taylor Series

Harsh Saxena
(2020PHY1162)(20068567023)

January 15, 2022

Lab Report for Assignment No. 1

College Roll No :	2020PHY1162
University Roll NoName:	20068567023
Unique Paper Code:	32221401
Paper Title:	Mathematical physics III Lab
Course and Semester :	B.Sc.(H) Physics Sem IV
Due Date:	Jan 15,2022
Date of Submission:	Jan 15,2022
Lab Report File Name:	MP3L1_2020PHY1162.pdf
Partner's Name:	Shashvat Jain
Partner's College Roll No.:	2020PHY1114

1 Theory

Any one-variable infinitely differentiable real-valued function $f(x) : A \to B$ where $A, B \subseteq \mathbb{R}$ might be expanded as an infinite power series function with parameter $x_0 \in A$, This series function is also termed as **Taylor series** representation of f because of its procurement from the **Taylor's Theorem.**

$$\begin{aligned} f(x) = T(x, x_0) &= f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \cdots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \end{aligned} \tag{1}$$

Taylor series representation of a function with the parameter $x_0 = 0$ is called the **Maclaurin series**.

$$f(x) = T(x, 0) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!}(x)^n \tag{2}$$

The point on the line $x = x_0$ is called the center of taylor series. The value of the function and its derivatives must be known at the center and the radius of convergence of the series is determined about this point.

Radius of Convergence of a power series: Every power series has a radius of convergence R which is the distance of its center from the nearest singularity(point of divergence). If $R > 0$, then the power series $\sum_{n=0}^{\infty} c_n(x - x_0)^n$ converges for all $|x - a| \leq R$ and diverges for $|x - a| > R$. If the series converges for all x, then we write $R = \infty$.

Taylor series representation for a **function of two variables** $f(x, y) : \mathbb{R}^2 \to \mathbb{R}$ about $(x, y) = (x_0, y_0)$ is given by the Taylor theorem as follows,

$$\begin{aligned} f(x, y) = T((x, y), (x{,}y_0)) = f(x_0, y_0) + f_x|_{x_0,y_0}(x - x_0) + f_y|_{x_0,y_0}(y - y_0) + +f_xx|_{x_0,y_0}(x - x_0)^2 \\ +f_{yy}|_{x_0,y_0}(y - y_0)^2 + f_{xy}|_{x_0,y_0}(x - x_0)(y - y_0) + \cdots \end{aligned} \tag{3}$$

The functions $\exp(x), \sin(x), \cos(x) : \mathbb{R} \to \mathbb{R}$ are defined by the following Maclaurin series expansions.

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} \qquad \text{for all } x \tag{4}$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \qquad \text{for all } x \tag{5}$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \qquad \text{for all } x \tag{6}$$

2 Algorithm

Algorithm 1 Euler expansion

procedure EULER(x, a, n)

Input: x points , a about which we calculate series,n as no. of terms to include in partial temrs
Output: Returns y the approx value of e^{x-a}
$exp = 0$ ▷ Initialize sum
for k $-$ 1 to n **do**
  $exp = exp + \frac{(x-a)^k}{factorial(k)}$
end for
return exp
end procedure

Algorithm 2 Sin Series Expansion

procedure SINSERIES(x, a, n)

Input: x points , a about which we calculate series,n as no. of terms to include in partial terms
Output: Returns y the approx value of $sin(x-a)$
$sin = 0$ ▷ Initialize sum
for k $-$ 1 to n **do**
  $sin = sin + \frac{(-1)^k(x-a)^{2k+1}}{factorial(2k+1)}$
end for
return sin
end procedure

Algorithm 3 Cos Series Expansion

procedure COSSERIES(x, a, n)

Input: x points , a about which we calculate series,n as no. of terms to include in partial terms
Output: Returns y the approx value of $cos(x-a)$
$cos = 0$ ▷ Initialize sum
for k $-$ 1 to n **do**
  $cos = cos + \frac{(-1)^k(x-a)^{2k}}{factorial(2k)}$
end for
return cos
end procedure

3 Programming

First we defined the following functions for MySinSeries,MyCosSeries which take in **x**: the point where the series is to be calculated, **a**: the center of the Taylor series and **n**: the number of terms of the series to calculate.

```
import matplotlib.pyplot as plt
  import math
  import numpy as np
  import numba as nb
  import pandas as pd
  plt.style.use("seaborn-dark-palette")

  @nb.vectorize
  def exp(x,a,n):
      sum_ = 1
      for i in np.arange(1,n):
          sum_ += (x-a)**(i)/math.gamma(i+1)
      return(sum_)

  @nb.vectorize
  def MySinSeries(x,a,n):  #returns y as vector b/c of vectorize function
      sum_ = 0
      for i in np.arange(n):
          sum_ += (-1)**i*(x-a)**(2*i+1)/math.gamma(2*i+2)
      return(sum_)

  @nb.vectorize
  def MyCosSeries(x,a,n):
      sum_ = 0
      for i in np.arange(n):
          sum_ += (-1)**i*(x-a)**(2*i)/math.gamma(2*i+1)
      return(sum_)

  '''this function will take input as function of taylor series
  expansion, the vector x and tolerence of y and outputs the no. of terms required in
  series expansion along with the result for given accuracy'''

  def Acc_expansion(f,x,a,tol):
      n = 200
      N = []
      func_acc = []
      for k in range(len(x)):
         for i in range(2,n):
             new_f = f(x[k],a,i)
             old_f = f(x[k],a,i-1)
             if new_f == old_f :
                 abs1 = 0
             else:
                 abs1 = abs(new_f - old_f)/abs(new_f)

             if abs1 == 0 :
                n1 =  0
                break
             elif abs1 <= tol :
                n1 = i
                break
         func_acc.append(new_f)
         N.append(n1)
      return N,func_acc
```

Next we proceeded obtain the data and get the necessary plots using the following script.

```
xs= np.linspace(-2*np.pi,2*np.pi,50)
m = [1,2,5,10,20]
plttyp = ['--1','--<','--x','--','--']

x0 = np.pi/4
m2 = np.arange(2,22,2)

# for Sine series
yj_sin = np.array([MySinSeries(xs,0,i) for i in m],dtype=float)

y0_sin = np.array(MySinSeries(x0,0,m2))

fig,(ax1,ax2) = plt.subplots(1,2)
for j in range(len(m)) :
    ax1.plot(xs,yj_sin[j],plttyp[j],label=f"m={m[j]}")
ax1.plot(xs,np.sin(xs),label = 'Inbuilt sine function')
ax1.set_ylim(-10,10)
ax1.set_xlabel('x')
ax1.set_ylabel('sin(x)')
ax2.plot(m2,y0_sin,'--*',label = 'MySinSeries(\u03C0/4,n)')
ax2.plot(m2,[np.sin(x0)]*len(m2),label = 'sin(\u03C0/4)')
ax2.set_xlabel('No. of terms')
ax2.set_ylabel('value of sin at \u03C0/4 ')
fig.suptitle('Sin Taylor expansion analysis')
ax2.legend()
ax1.legend()

# for cos series
yj_cos = np.array([MyCosSeries(xs,0,i) for i in m],dtype=float)

y0_cos = np.array(MyCosSeries(x0,0,m2))

fig,(ax1,ax2) = plt.subplots(1,2)
for j in range(len(m)) :
    ax1.plot(xs,yj_cos[j],plttyp[j],label=f"m={m[j]}")
ax1.plot(xs,np.cos(xs),label = 'Inbuilt Cos function')
ax1.set_ylim(-10,10)
ax1.set_xlabel('x')
ax1.set_ylabel('Cos(x)')
ax2.plot(m2,y0_cos,'--*',label = 'MyCosSeries(\u03C0/4,n)')
ax2.plot(m2,[np.cos(x0)]*len(m2),label = 'cos(\u03C0/4)')
ax2.set_xlabel('No. of terms')
ax2.set_ylabel('value of Cos at \u03C0/4 ')
fig.suptitle('Cos Taylor expansion analysis')
ax2.legend()
ax1.legend()


# Question 2

x_ac = np.arange(0,np.pi + (np.pi/8),np.pi/8) # x in [0 , pi ]

#for a accuracy of n significant digits we require a rel tol of 0.5*10**(-n)

tol1 = 0.5*10**(-3)      #for plotting
tol2 = 0.5*10**(-6)      #for tabulation

N_6,Sin_ap6 = Acc_expansion(MySinSeries, x_ac, 0, tol2)

sin_ap6tb = ["%.7f" % k for k in Sin_ap6]

sin_tb2 = ["%.7f" % k for k in np.sin(x_ac)]

d1 = {'X':x_ac,'Sin(x)(calc)':sin_ap6tb,'No. of terms':N_6,'Sin(x)(inbuilt)':sin_tb2}
```

```
df1 = pd.DataFrame(data=d1)

print('Sin calcualted at these points for accuracy upto 6 significant digits => \n',
    df1)

N_3 , Sin_ap3 = Acc_expansion(MySinSeries, xs, 0, tol1)

fig,ax = plt.subplots()
ax.plot(xs,Sin_ap3,'o',label = 'Calculated sin for accuracy upto 3 significant digits
    ')
ax.plot(xs,np.sin(xs),label = 'Continuous sin curve')
ax.set_title('Sin(x) calculated and analytic')
ax.set_xlabel('x')
ax.set_ylabel('Sin(x)')
plt.legend()
```

4 Discussion

1. We compare taylor expansion of Sin for different no. of terms,We see that in the interval $[-2\pi, 2\pi]$ The taylor expansion very nicely approximate the Sin function for n = 10,20.
   Further we see the convergence of taylor expansion of Sin at $X0 = \frac{\pi}{4}$ with increasing n,see the figure 1.
   We have also tabulated the difference between approx Y0 at n and n-2,We observe that as we increase n the error reduces with each increasing n,and after some n the convergence with true value is reached.

S.No.	Y's	Error
1.	Y0(4)-Y0(2)	0.002453818
2.	Y0(6)-Y0(4)	0.000000312
3.	Y0(8)-Y0(6)	0.00000000000693
4.	Y0(10)-Y0(8)	0
5.	Y0(12)-Y0(10)	0
6.	Y0(14)-Y0(12)	0
7.	Y0(16)-Y0(14)	0
8.	Y0(18)-Y0(16)	0
9.	Y0(20)-Y0(18)	0

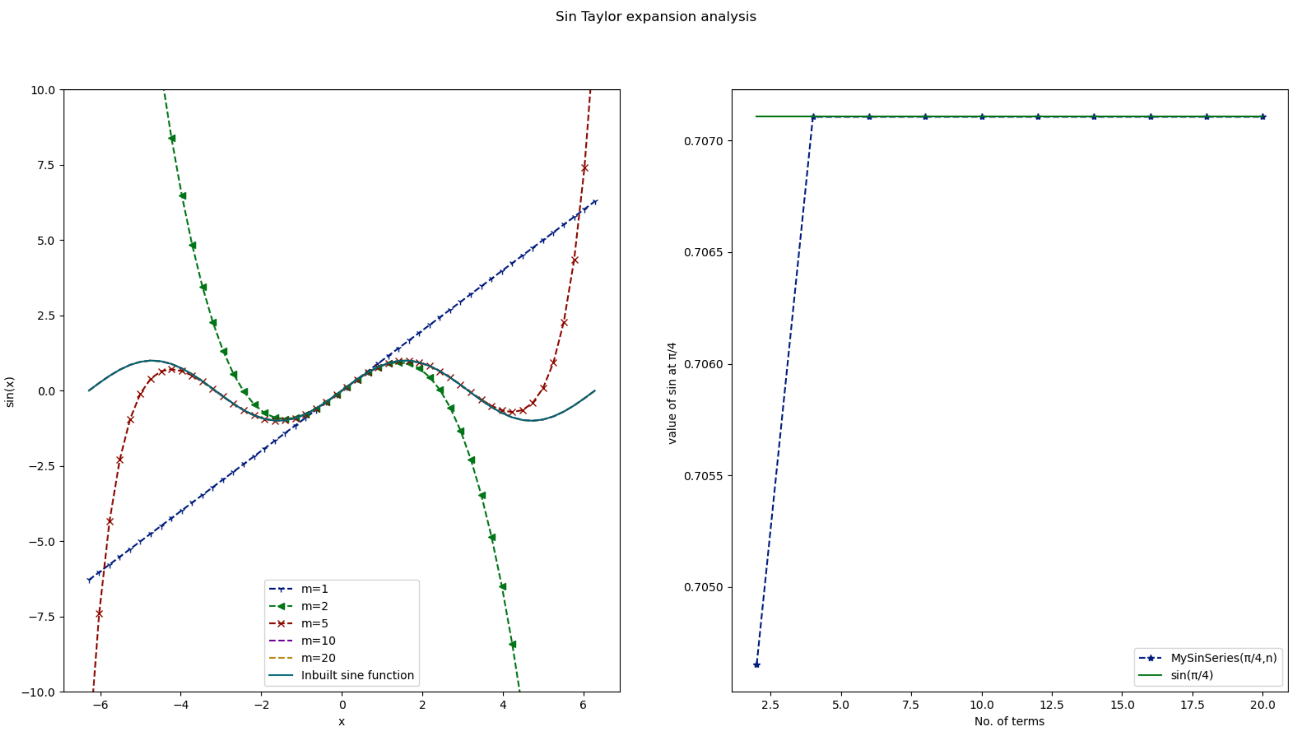

Figure 1: Graphs generated for MySinSeries: The graph on the left shows different taylor series approximations corresponding to different values of m=1,2,5,10,20. The graph on the right shows how the value of sin(π/4) varies with different values of m = 2,4,...,20.

2. We compare taylor expansion of Cos for different no. of terms,We see that in the interval $[-2\pi, 2\pi]$ The taylor expansion very nicely approximate the Cos function for n = 10,20.
   Further we see the convergence of taylor expansion of Cos at $X0 = \frac{\pi}{4}$ with increasing n,see the figure 2.
   We provide the same argument as for Sin,by Seeing the below table.

S.No.	Y's	Error
1.	Y0(4)-Y0(2)	0.015528352356888
2.	Y0(6)-Y0(4)	3.56624907904556E-06
3.	Y0(8)-Y0(6)	1.14621756530652E-10
4.	Y0(10)-Y0(8)	9.9920072216264E-16
5.	Y0(12)-Y0(10)	0
6.	Y0(14)-Y0(12)	0
7.	Y0(16)-Y0(14)	0
8.	Y0(18)-Y0(16)	0
9.	Y0(20)-Y0(18)	0

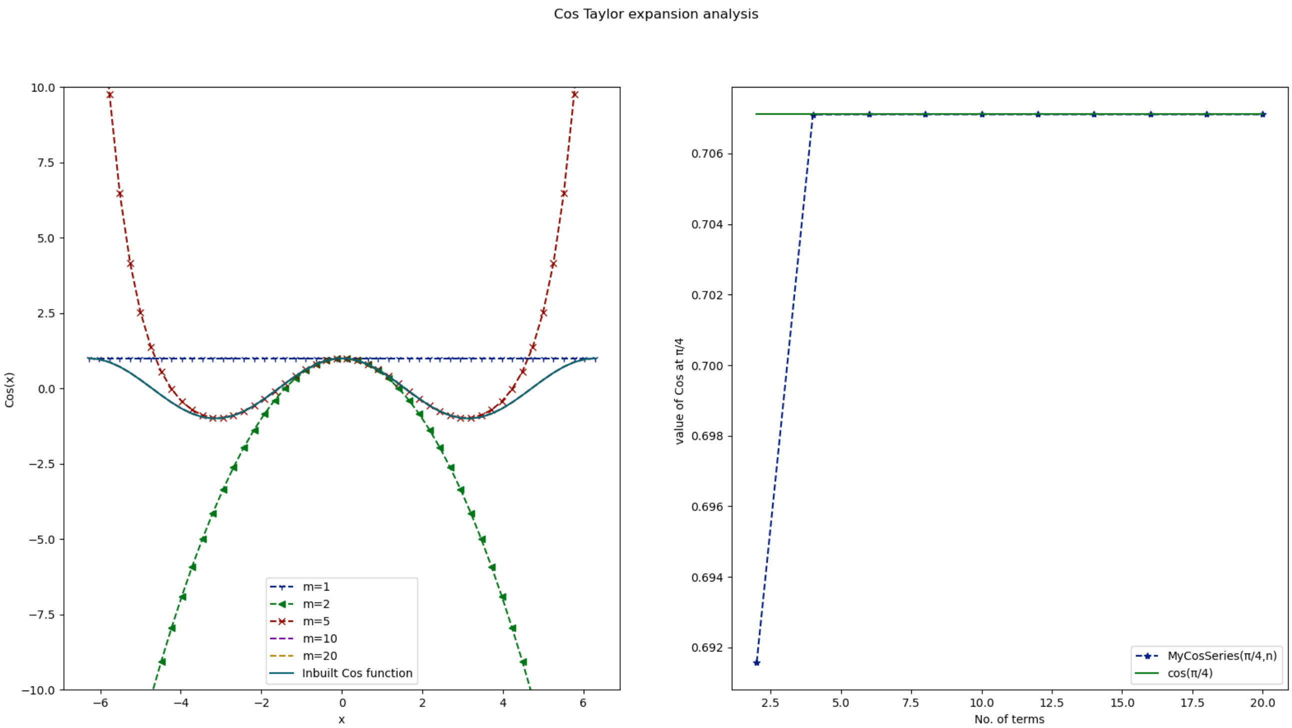

Figure 2: Graphs generated for MyCosSeries: The graph on the left shows the different taylor series approximations corresponding to different values of m=1,2,5,10,20. The graph on the right shows how the value of $\cos(\pi/4)$ varies with different values of m=2,4,...,20.

3. We Observe that the points that signify the Calculated Sin with accuracy upto 3 significant digits,are very close to original points with a absolute error of order $10^{-5} - 10^{-7}$.

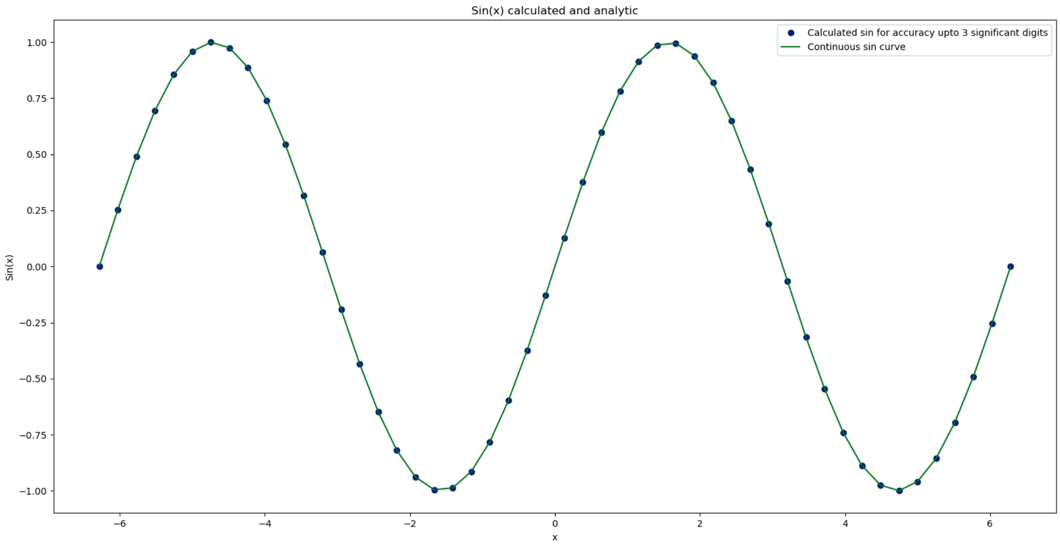

Figure 3: Graph for approx sin upto 3 s.d: The continuous curve on the graph is Inbuilt sin function and the scattered points are Sin calculated using Taylor series with accuracy of 3 significant digits.

4. Here we have tabulated the Sin calculated using taylor series expansion with an accuracy of 6 significant digits,Note that upto 5 decimal places both the Sin_{Calc} and $Sin_{inbuilt}$ are equal.
Further we see that in the interval $[0, \pi]$ as we increase x with as step of $\frac{\pi}{8}$ the No. of terms required to bring the accuracy also increases.
We observe that when we take relative error as measure for accuracy, when the function approaches 0 the relative tolerence shoots up even before the accuracy is reached

Sin calcualted at these points for accuracy upto 6 significant digits =>

	X	Sin(x)(calc)	No. of terms	Sin(x)(inbuilt)
0	0.000000	0	0	0
1	0.392699	0.3826834	5	0.3826834
2	0.785398	0.7071068	5	0.7071068
3	1.178097	0.9238795	6	0.9238795
4	1.570796	1	7	1
5	1.963495	0.9238795	8	0.9238795
6	2.356194	0.7071068	8	0.7071068
7	2.748894	0.3826834	9	0.3826834
8	3.141593	4.431805e-16	18	1.224647e-16

Figure 4: Tabulated Sin calculated upto 6 s.d.: This data shows the tabulated data for Sin calculated upto Accuracy of 6 Signicant digits and also tells us the no. of terms in Taylor expansion is required to reach this accuracy