EE2703 Assignment 4

Bandi Yaswanth, EE20B020

10^{th} March 2022

1 Abstract

We will fit two functions, e^x and $cos(cos(x))$ over the interval $[0, 2\pi)$ using their computed Fourier series coefficients.

2 Introduction

The Fourier Series of a function $f(x)$ with period 2π is computed as follows:

$$f(x) = a_0 + \sum_{n=1}^{+\infty}\{a_n cos(nx) + b_n sin(nx)\} \tag{1}$$

where ,

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(x)dx$$
$$a_n = \frac{1}{2\pi}\int_0^{2\pi} f(x) * \cos(nx)dx$$
$$b_n = \frac{1}{2\pi}\int_0^{2\pi} f(x) * \sin(nx)dx$$

For the sake of this assignment, Since e^x doesn't have a period of 2π,We choose to change its definition in a piece-wise manner to satisfy periodicity.

3 Assignment Questions

3.1 Defining the functions

Defining the functions e^x and $cos(cos(x))$ that can take both the scalar input and vector input and returns the appropriate output and plotting them over the interval $[-4\pi, 2\pi]$. $cos(cos(x))$ is periodic with period 2π while the function e^x is not periodic. Therefore, the function generated by the fourier series will be $cos(cos(x))$ and $exp(x\%2\pi)$.

```
def exp(x):
    return np.exp(x)

def cos(x):
    return np.cos(np.cos(x))

t = linspace(-2*pi,4*pi,401)
Exp = exp(t)
Cos = cos(t)
Exp2 = exp(t%(2*pi))

figure(1)
plt.semilogy(t,Exp,label = 'original exp(t)')
plt.semilogy(t,Exp2,label = 'periodic exp(t) with period 2*pi')
plt.grid(True, color = "grey", linewidth = "1.4", linestyle = "--")
xlabel(r'$t$'     ,size = 20)
ylabel(r'exp($t$)',size = 20)
legend(loc='upper right')
title(r'exponential of $t$ in semilog axis')

figure(2)
plot(t,Cos)
plt.grid(True, color = "grey", linewidth = "1.4", linestyle = "--")
xlabel(r'$t$'          ,size = 20)
ylabel(r'cos(cos($t$))',size = 20)
title(r'cosine of cos(cos($t$)) in linear axis')
```

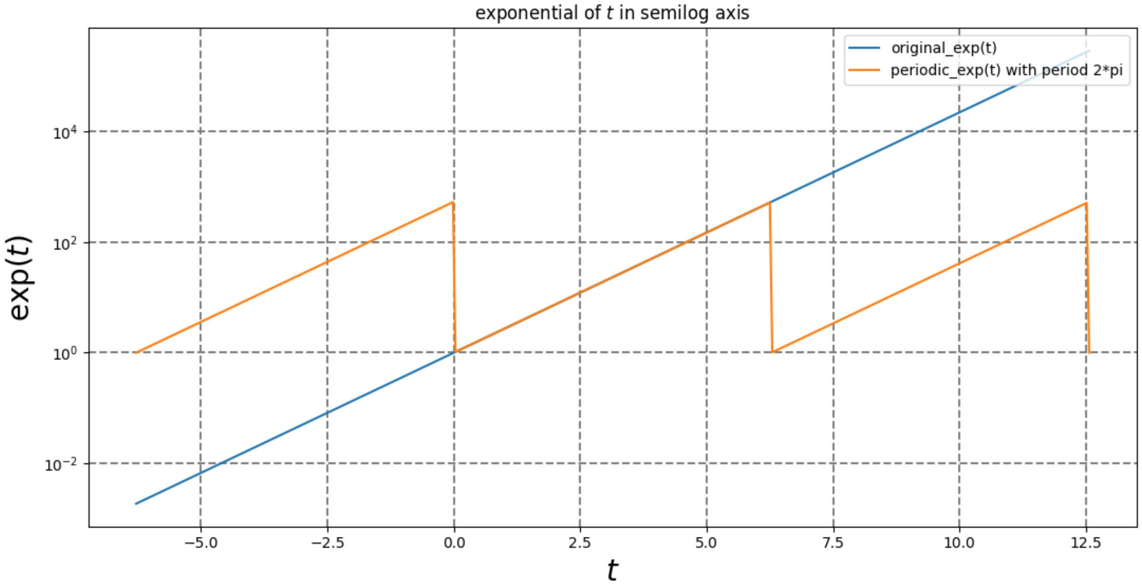

Figure 1: $exp(x)$ actual function vs periodic function

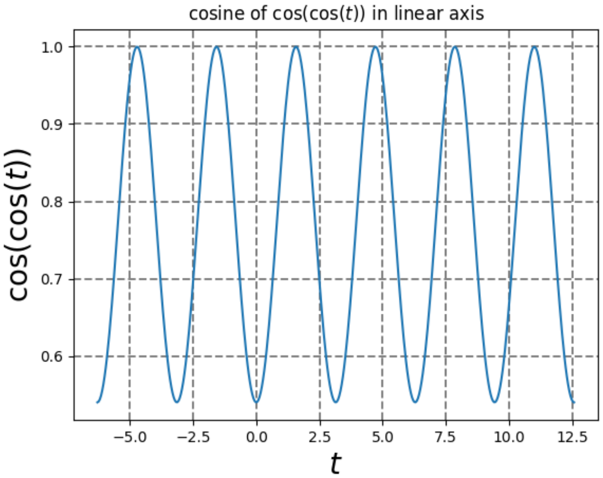

Figure 2: $cos(cos(x))$

3.2 Generating Fourier Coefficients

Obtaining the first 51 fourier coefficients namely first 26 fourier cosine and first 25 sine coefficients in the following order.

$$\begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \dots \\ a_{25} \\ b_{25} \end{bmatrix}$$

- Creating two new functions to be integrated, namely $u(x,k) = f(x)cos(kx)andv(x,k) = f(x)sin(kx)$. To integrate these, use the option in quad to pass extra arguments to the function being integrated.

```
def u(x,k):
      return np.cos(k*x)*f(x)
def v(x,k):
      return np.sin(k*x)*f(x)

coef1 = np.zeros(51)
coef2 = np.zeros(51)
def f(x):
      return exp(x)
for i in range(51):
      if(i == 0):
           coef1[i] =quad(u,0,2*pi,args=(i))[0]/(2*pi)
      elif(i%2 != 0):
           coef1[i] =quad(u,0,2*pi,args=((i+1)/2))[0]/pi
      elif(i%2 == 0):
           coef1[i] =quad(v,0,2*pi,args=(i/2))[0]/pi


def f(x):
      return cos(x)
for i in range(51):
      if(i == 0):
           coef2[i] =quad(u,0,2*pi,args=(i))[0]/(2*pi)
      elif(i%2 != 0):
           coef2[i] =quad(u,0,2*pi,args=((i+1)/2))[0]/pi
      elif(i%2 == 0):
           coef2[i] =quad(v,0,2*pi,args=(i/2))[0]/pi
```

3.3 Visualizing Fourier Coefficients

As expected, $\sum \|b_n\|$ for cos(cos(x)) = 2.09e-14 which is almost 0. The coefficients for e^x decay faster than that of The Log-log plot for Fourier coefficients of e^x is nearly linear because :

$$\int_0^{2\pi} e^x cos(kx)dx = \frac{(e^{2\pi} - 1)}{(k^2 + 1)} \tag{2}$$

and

$$\int_0^{2\pi} e^x sin(kx)dx = \frac{(-ke^{2\pi} + k)}{(k^2 + 1)} \tag{3}$$

the log-log plots of these functions are linear

The semi-log plot for Fourier Coefficients of $cos(cos(x))$ is linear as the integral converges to a Linear Combination of Bessel functions which are proportional to e^x.

```
n = linspace(0,50,51)
figure(3)
plt.semilogy((n),abs(coef1),’ro’,label = ’fourier␣coeff’)
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r’$n$␣’)
ylabel(r’coefficient[$n$]␣’)
legend(loc=’upper␣right’)
title(r’fourier␣coefficients␣of␣$exp(t)$␣in␣the␣semilog␣scale’)


figure(4)
plt.loglog((n),abs(coef1),’ro’,label = ’fourier␣coeff’)
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r’$n$␣’)
ylabel(r’coefficient[$n$]␣’)
legend(loc=’upper␣right’)
title(r’fourier␣coefficients␣of␣$exp(t)$␣in␣the␣loglog␣scale’)


figure(5)
plt.semilogy((n),abs(coef2),’ro’,label = ’fourier␣coeff’)
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r’$n$␣’)
ylabel(r’coefficient[$n$]␣’)
legend(loc=’upper␣right’)
title(r’fourier␣coefficients␣of␣$cos(cos(t))$␣in␣the␣semilog␣scale’)


figure(6)
plt.loglog((n),abs(coef2),’ro’,label = ’fourier␣coeff’)
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r’$n$␣’)
ylabel(r’coefficient[$n$]␣’)
legend(loc=’upper␣right’)
title(r’fourier␣coefficients␣of␣$cos(cos(t))$␣in␣the␣loglog␣scale’)
```

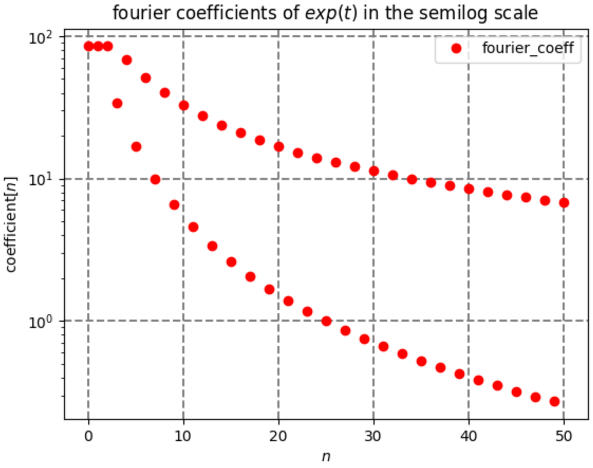

Figure 3: plot of fourier coefficients of $exp(x)$ in semi-log scale

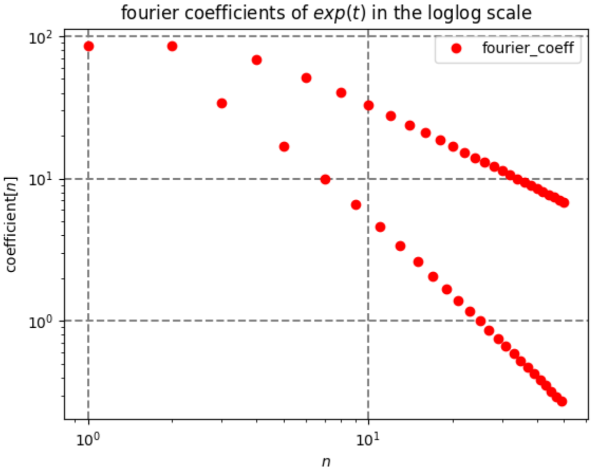

Figure 4: plot of fourier coefficients of $exp(x)$ in log-log scale

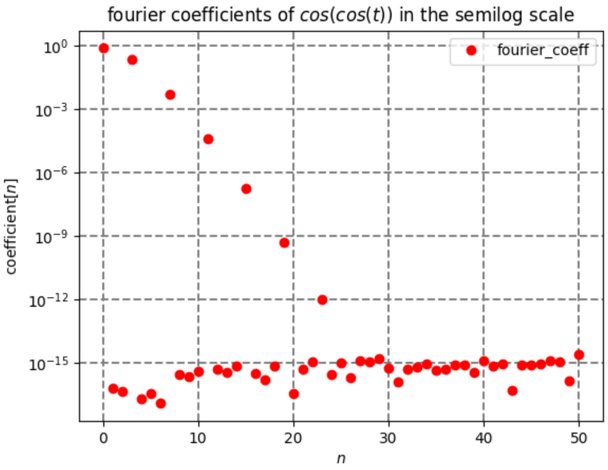

Figure 5: plot of fourier coefficients of $cos(cos(x))$ in semi-log scale

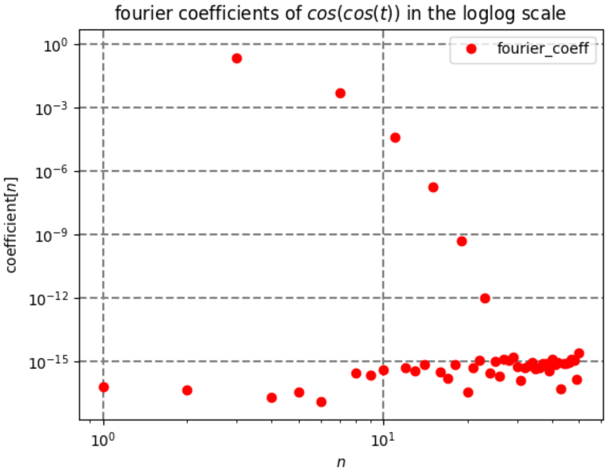

Figure 6: plot of fourier coefficients of $cos(cos(x))$ in log-log scale

3.4 A Least Squares Approach

Building on last weeks work we now try a Least Squares Approach to this problem. We linearly choose 400 values of x in the range [0,2π).It can be Noted that far better approximations were achieved when a larger value(10000) was used instead of 400. We try to solve Equation (1) By using regression on these 400 values

$$\begin{pmatrix} 1 & \cos(x_1) & \sin(x_1) & .... & \cos(25x_1) & \sin(25x_1) \\ 1 & \cos(x_2) & \sin(x_2) & .... & \cos(25x_2) & \sin(25x_2) \\ ... & ... & ... & .... & ... & ... \\ 1 & \cos(x_{400}) & \sin(x_{400}) & .... & \cos(25x_{400}) & \sin(25x_{400}) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ ... \\ a_{25} \\ b_{25} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ ... \\ f(x_{400}) \end{pmatrix}$$

We create the matrix on the left side and call it A . We want to solve $Ac = b$ where c are the fourier coefficients.

```
x = linspace(0,2*pi,401)
x = x[:-1]
b1 = np.exp(x)
b2 = np.cos(x)
A = np.zeros((400,51))
A[:,0] = 1

for k in range(1,26):
    A[:,2*k] = np.sin(k*x)
c1=lstsq(A,b1,rcond = 1)[0]
c2=lstsq(A,b2,rcond = 1)[0]
```

3.5 Visualizing output of the Least Squares Approach

```
n = linspace(0,50,51)
figure(7)
plt.semilogy((n),abs(coef1),'ro',label = 'fourier␣coeff')
plt.semilogy((n),abs(c1),'go',label = 'least␣sq␣coeff')
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r'$n$␣')
ylabel(r'coefficient[$n$]␣')
legend(loc='upper␣right')
title(r'fourier␣coefficients␣of␣$exp(t)$␣in␣the␣semilog␣scale')

figure(8)
plt.loglog((n),abs(coef1),'ro',label = 'fourier␣coeff')
plt.loglog((n),abs(c1),'go',label = 'least␣sq␣coeff')
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r'$n$␣')
ylabel(r'coefficient[$n$]␣')
legend(loc='upper␣right')
title(r'fourier␣coefficients␣of␣$exp(t)$␣in␣the␣loglog␣scale')

figure(9)
```

```
plt.semilogy((n),abs(coef2),'ro',label = 'fourier_coeff')
plt.semilogy((n),abs(c2),'go',label = 'least_sq_coeff')
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r'$n$ ')
ylabel(r'coefficient[$n$] ')
legend(loc='upper right')
title(r'fourier coefficients of $cos(cos(t))$ in the semilog scale')

figure(10)
plt.loglog((n),abs(coef2),'ro',label = 'fourier_coeff')
plt.loglog((n),abs(c2),'go',label = 'least_sq_coeff')
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”—”)
xlabel(r'$n$ ')
ylabel(r'coefficient[$n$] ')
legend(loc='upper right')
title(r'fourier coefficients of $cos(cos(t))$ in the loglog scale')
```

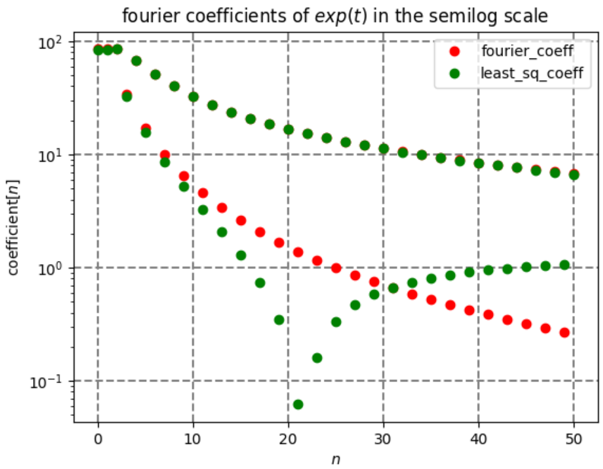

Figure 7: plot of fourier coefficients of $exp(x)$ in semi-log scale

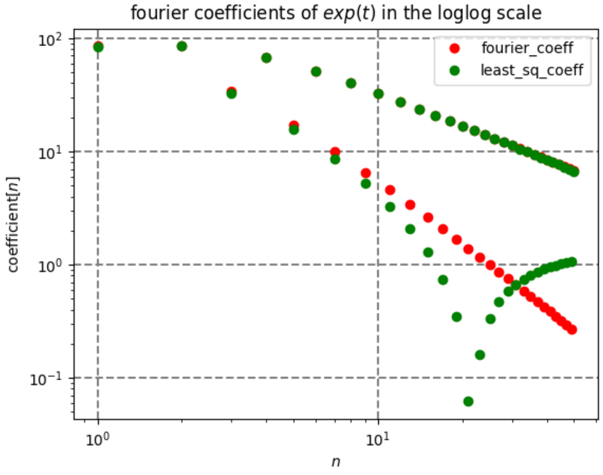

Figure 8: plot of fourier coefficients of $exp(x)$ in log-log scale

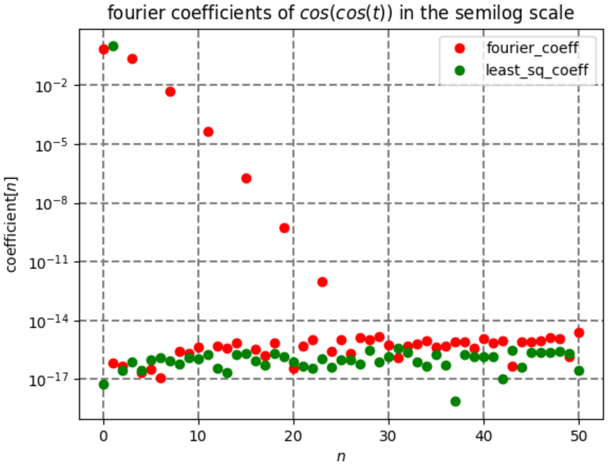

Figure 9: plot of fourier coefficients of $cos(cos(x))$ in semi-log scale

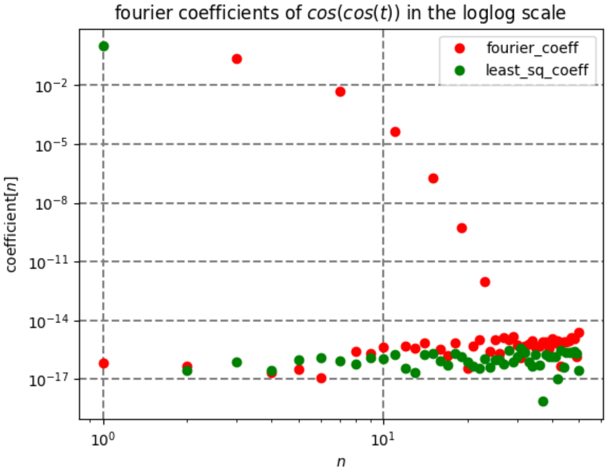

Figure 10: plot of fourier coefficients of $cos(cos(x))$ in log-log scale

3.6 Comparing Predictions

The maximum absolute error for e^x =1.3327308703353822
Whereas the same metric for cos(cos(x)) = 2.5869790664721666e-15
Our Predictions for e^x are very poor compared to that of cos(cos(x)). This can be fixed by sampling at a larger number of points.
For instance if we sample at 1e5 points :
The maximum absolute error for e^x =0.005344860583583633
Whereas the same metric for cos(cos(x)) = 2.6342502811925345e-15
This is a small change in accuracy at a very large computational cost. Hence it isn't worth it as far as this assignment is concerned.

```
experr = np.zeros(51)
coserr = np.zeros(51)
for i in range(51):
    experr[i] = abs(coef1[i] - c1[i])
for i in range(51):
    coserr[i] = abs(coef2[i] - c2[i])
figure(11)
plot(n,experr,'bo',label = 'error␣magnitude␣for␣the␣exp($t$)')
plot(n,coserr,'mo',label = 'error␣magnitude␣for␣the␣cos(cos($t$))')
plt.grid(True, color = "grey", linewidth = "1.4", linestyle = "--")
legend(loc='upper␣right')
xlabel(r'$n$',size = 20)
ylabel(r'error␣coeff[$n$]',size = 20)
title(r'magnitude␣error␣in␣the␣coefficients')

print("the␣largest␣error␣in␣the␣coefficients␣of␣the␣exp($x$)␣is␣%f" %(max(experr)))
print("the␣largest␣error␣in␣the␣coefficients␣of␣the␣cos(cos($x$))␣is␣%f" %(max(coserr)))
```

3.7 Plotting Results

```
Exp2 = np.dot(A,coef1)
Cos2 = np.dot(A,coef2)

t = linspace(0,2*pi,401)
t = t[:-1]
Exp = exp(t)
Cos = cos(t)

figure(12)
plt.semilogy(t,Exp2,'go',label = 'exp(t)␣from␣fourier')
plt.semilogy(t,Exp,label = 'actual␣exp(t)',color = 'black')
plt.grid(True, color = "grey", linewidth = "1.4", linestyle = "--")
xlabel(r'$t$',size = 20)
ylabel(r'exp($t$)␣',size = 20)
legend(loc='upper␣right')
title(r'variation␣of␣the␣exp($t$)')

figure(13)
```

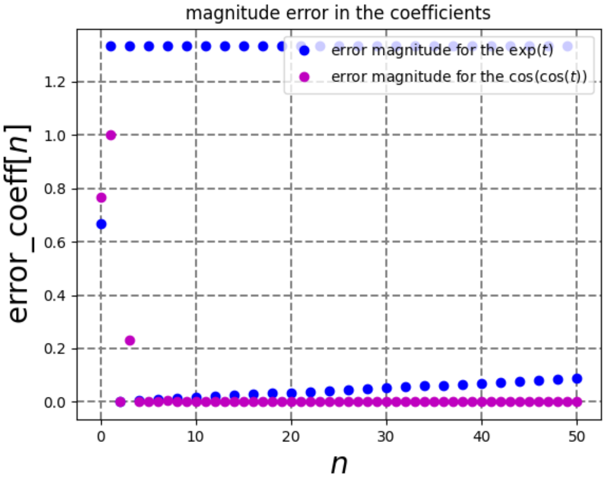

Figure 11: error in the magnitude of the fourier coefficients and those by the least square method.

```
plot(t,Cos2,’go’,label = ’cos(cos(t))␣from␣fourier’)
plot(t,Cos,label = ’actual␣cos(cos(t))’,color = ’black’)
plt.grid(True, color = ”grey”, linewidth = ”1.4”, linestyle = ”--”)
xlabel(r’$t$’,size = 20)
ylabel(r’cos(cos($t$))’,size = 20)
legend(loc=’upper␣right’)
title(r’variation␣of␣cos(cos($t$))’)

show()
```

It should be noted that e^x is a non periodic function and Fourier series' exists only for periodic functions. Hence we have considered a variation of e^x with period 2pi that has the actual value of e^x only in the range [0,2π). Hence it is acceptable that there is a large discrepancy in the predicted value of e^x at these boundaries

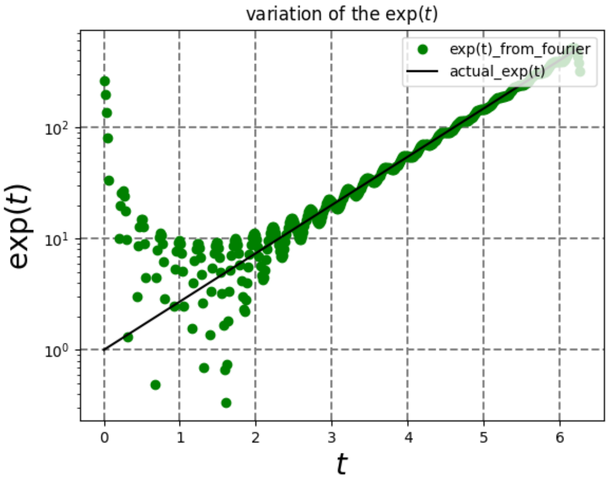

Figure 12: variation of the actual $exp(t)$ from the estimated function with fourier coefficients.

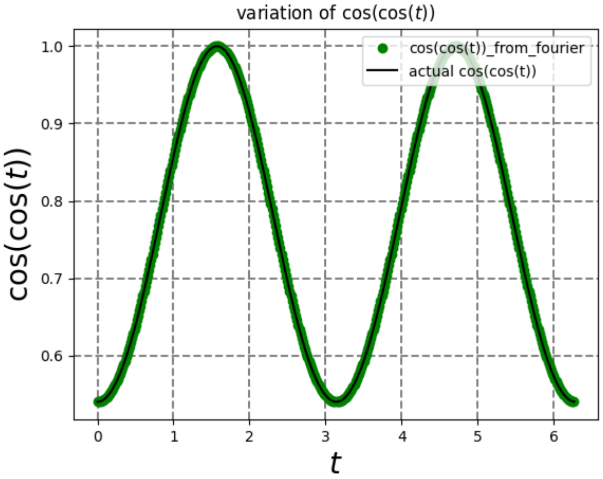

Figure 13: variation of actual $cos(cos(x))$ from the estimated function with fourier coefficients.

4 Conclusion

We have examined the case of approximating functions using their Fourier coefficients upto a threshold. Whilst doing so, we perform the same for two cases, one a continuous function, and the other a function with finite discontinuities.

The methods adopted in finding the respective Fourier coefficients have been direct evaluation of the Fourier series formula, as well an Least Square best fit. We notice close matching of the two methods in case of $\cos(\cos(x))$ while, there is a larger discrepancy in $\exp(x)$.

Besides this, we also highlight the fact of non-uniform convergence of the fourier series in case of finitely discontinuous functions.