EE2703 Assignment 4

Gali Ushasri ee20b033

March 8, 2022

1 Abstract

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

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x)dx$$

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

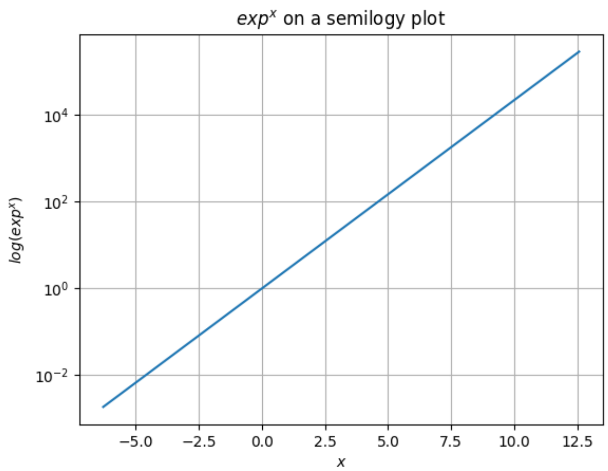
$$b_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) * \sin(nx)dx$$

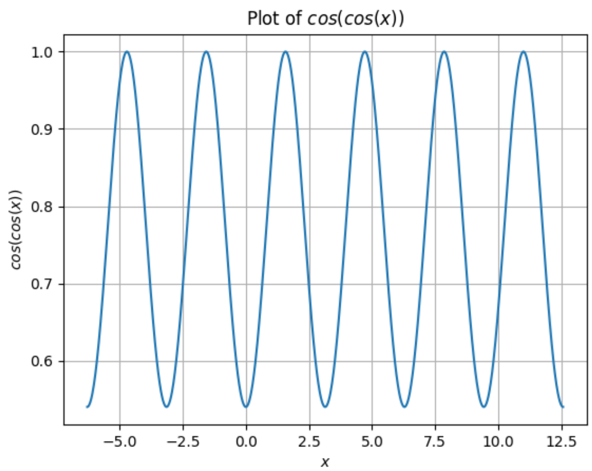
3 Assignment

3.1 Creating the functions

$cos(cos(x))$ is a periodic function with period 2π whereas e^x is not. The functions that generated from the Fourier series are $cos(cos(x))$ and $e^{x\%(2\pi)}$

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

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

def exp_plot(lowerlimit,upperlimit, n):
    x = np.linspace(lowerlimit,upperlimit,n)
    plt.grid()
    plt.semilogy(x,exp(x))
    plt.title(r'$exp^x$ on a semilogy plot')
    plt.xlabel('$x$')
    plt.ylabel(r'$log(exp^x)$')
    plt.savefig('exp_plot.png')
    plt.close()

def coscos_plot(lowerlimit,upperlimit, n):
    x = np.linspace(lowerlimit,upperlimit,n)
    plt.grid()
    plt.title(r'Plot of $cos(cos(x))$')
    plt.xlabel('$x$')
    plt.ylabel(r'$cos(cos(x))$')
    plt.plot(x,coscos(x))
    plt.savefig('coscos_plot.png')
    plt.close()
```





3.2 Generating Fourier Coefficients

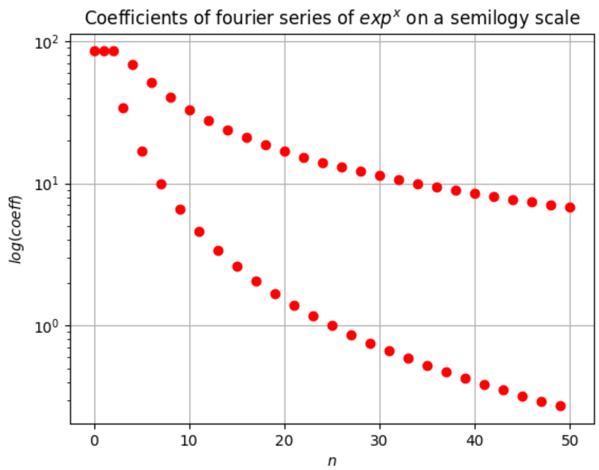
The first 51 coefficients are generated using the scipy.integrate.quad and the equations mentioned in the introduction function.They are saved in the following form as required by part 3:

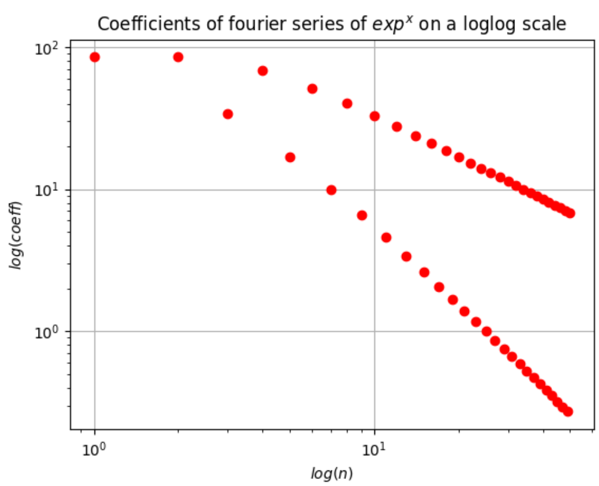
$$\begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \ldots \\ a_{25} \\ b_{25} \end{bmatrix}$$

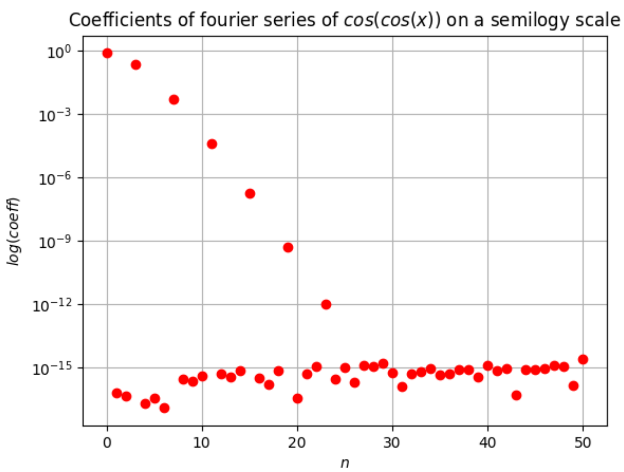
```
def FT(n,function):
 a = np.zeros(n)

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

 a[0] = integrate.quad(function,0,2*np.pi)[0]/(2*np.pi)

 for i in range(1,n):
     if(i%2==1):
         a[i] = integrate.quad(u,0,2*np.pi,args=(int(i/2)+1,function))[0]
     else:
         a[i] = integrate.quad(v,0,2*np.pi,args=(int(i/2),function))[0]
 return a
```

3.3 Visualizing Fourier Coefficients

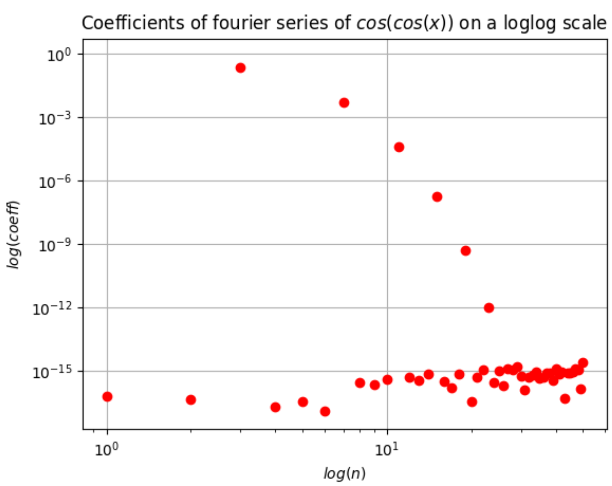
```
def plot_semilog_loglog(eFt_coeff, coscosFt_coeff, color = 'ro'):
    eFt_coeff = np.abs(eFt_coeff)
    coscosFt_coeff = np.abs(coscosFt_coeff)
    plt.grid()
    plt.title(r"Coefficients of fourier series of $exp^x$ on a semilogy scale")
    plt.xlabel(r'$n$')
    plt.ylabel(r'$log(coeff)$')
    plt.semilogy(eFt_coeff, color)
    plt.savefig('semilog_eFt_coeff.png')
    plt.close()

    plt.grid()
    plt.title(r"Coefficients of fourier series of $exp^x$ on a loglog scale")
    plt.xlabel(r'$log(n)$')
    plt.ylabel(r'$log(coeff)$')
    plt.loglog(eFt_coeff, color)
    plt.savefig('loglog_eFt_coeff.png')
    plt.close()

    plt.grid()
    plt.title(r"Coefficients of fourier series of $cos(cos(x))$ on a semilogy scale")
    plt.xlabel(r'$n$')
    plt.ylabel(r'$log(coeff)$')
    plt.semilogy(coscosFt_coeff, color)
    plt.savefig('semilog_coscos_coeff.png')
    plt.close()

    plt.grid()
    plt.title(r"Coefficients of fourier series of $cos(cos(x))$ on a loglog scale")
    plt.xlabel(r'$log(n)$')
    plt.ylabel(r'$log(coeff)$')
    plt.loglog(coscosFt_coeff, color)
    plt.savefig('loglog_coscos_coeff.png')
    plt.close()
```


Coefficients of fourier series of exp^x on a semilogy scale
10^2
10^1
10^0
log(coeff)
0
10
20
30
40
50
n


Coefficients of fourier series of exp^x on a loglog scale
10^2
10^1
10^0
log(coeff)
10^0
10^1
log(n)



Coefficients of fourier series of $cos(cos(x))$ on a semilogy scale
10^{0}
10^{-3}
10^{-6}
10^{-9}
10^{-12}
10^{-15}
$log(coeff)$
0
10
20
30
40
50
n



Coefficients of fourier series of $cos(cos(x))$ on a loglog scale
10^{0}
10^{-3}
10^{-6}
10^{-9}
10^{-12}
10^{-15}
$log(coeff)$
10^{0}
10^{1}
$log(n)$

3.4 A Least Squares Approach

We linearly choose 400 values of x in the range $[0,2\pi)$. We try to solve Equation (1) By using regression on these 400 values

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

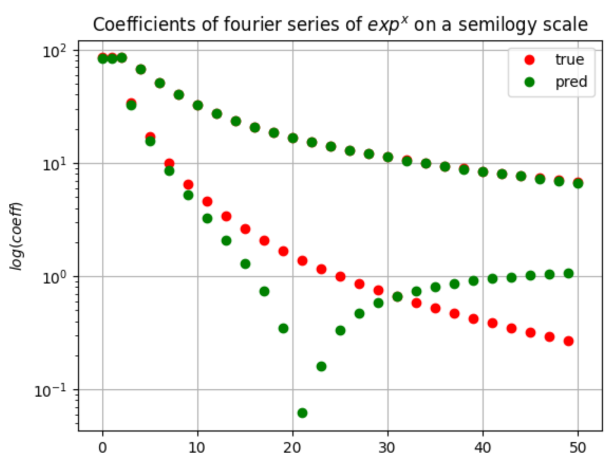
We create the matrix on the left side and call it A . We want to solve $Ac = b$ where c are the fourier coefficients.

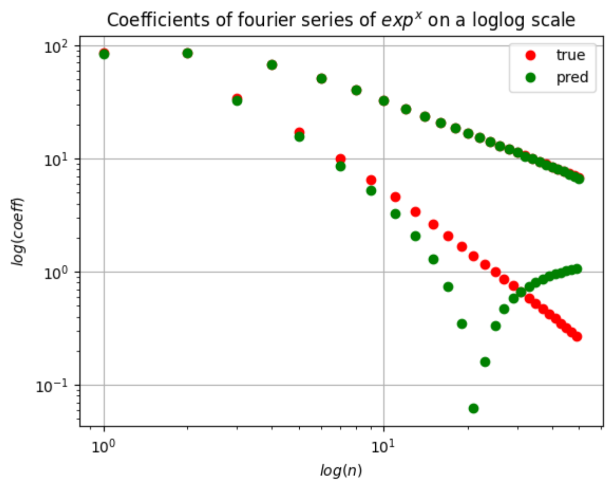
```
def generateA_b(f,x):
    A = np.zeros((400,51))
    A[:,0] = 1
    for k in range(1,26):
        A[:,2*k-1]=np.cos(k*x)
        A[:,2*k]=np.sin(k*x)
    return A,f(x)
x = np.linspace(0,2*np.pi,401)
x=x[:-1]
A_coscos,b_coscos = generateA_b(coscos,x)
A_exp,b_exp = generateA_b(exp,x)
```

3.5 Visualizing output of the Least Squares Approach

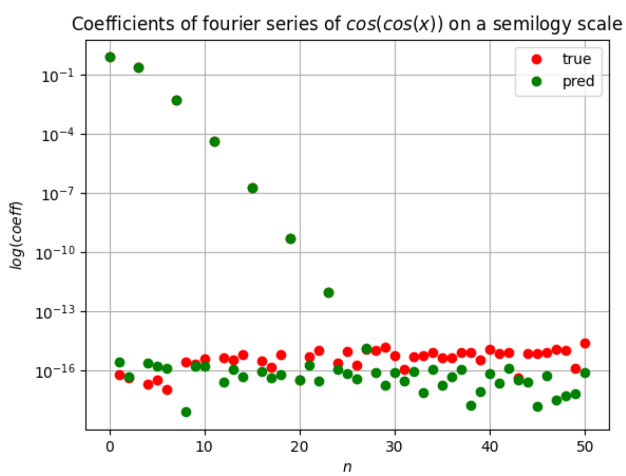
```
c_coscos = scipy.linalg.lstsq(A_coscos,b_coscos)[0]
c_exp = scipy.linalg.lstsq(A_exp,b_exp)[0]

def plot(eFt_coeff,coscosFt_coeff,c_exp,c_coscos,color):
    eFt_coeff = np.abs(eFt_coeff)
    coscosFt_coeff = np.abs(coscosFt_coeff)

    plt.grid()
    plt.title(r"Coefficients of fourier series of $exp^x$ on a semilogy scale")
    plt.xlabel(r'$n$')
    plt.ylabel(r'$log(coeff)$')
    plt.semilogy(eFt_coeff,'ro')
    plt.semilogy(c_exp,color)
    plt.legend(["true","prediction"])
    plt.savefig('fig1.png')
    plt.close()

    plt.grid()
    plt.title(r"Coefficients of fourier series of $exp^x$ on a loglog scale")
    plt.xlabel(r'$log(n)$')
    plt.ylabel(r'$log(coeff)$')
    plt.loglog(eFt_coeff,'ro')
    plt.loglog(c_exp,color)
```

```
plt.legend(["true","prediction"])
plt.savefig('fig2.png')
plt.close()

c_exp = np.abs(c_exp)
c_coscos = np.abs(c_coscos)

plt.grid()
plt.xlabel(r'$n$')
plt.ylabel(r'$log(coeff)$')
plt.semilogy(coscosFt_coeff,'ro')
plt.semilogy(c_coscos,color)
plt.legend(["true","prediction"])
plt.title(r"Coefficients of fourier series of $cos(cos(x))$ on a semilogy scale")
plt.savefig('fig3.png')
plt.close()

plt.grid()
plt.xlabel(r'$log(n)$')
plt.ylabel(r'$log(coeff)$')
plt.loglog(coscosFt_coeff,'ro')
plt.loglog(c_coscos,color)
plt.legend(["true","prediction"])
plt.title(r"Coefficients of fourier series of $cos(cos(x))$ on a loglog scale")
plt.savefig('fig4.png')
plt.close()
```

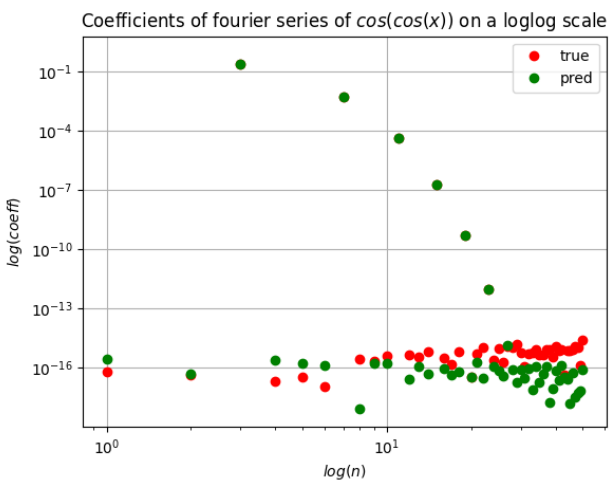




Coefficients of fourier series of exp^x on a loglog scale
true
pred
10^2
10^1
10^0
10^{-1}
log(coeff)
10^0
10^1
log(n)



Coefficients of fourier series of $cos(cos(x))$ on a semilogy scale
true
pred
10^{-1}
10^{-4}
10^{-7}
10^{-10}
10^{-13}
10^{-16}
log(coeff)
0
10
20
30
40
50
n


Coefficients of fourier series of $cos(cos(x))$ on a loglog scale
true
pred
10^{-1}
10^{-4}
10^{-7}
10^{-10}
10^{-13}
10^{-16}
$log(coeff)$
10^{0}
10^{1}
$log(n)$

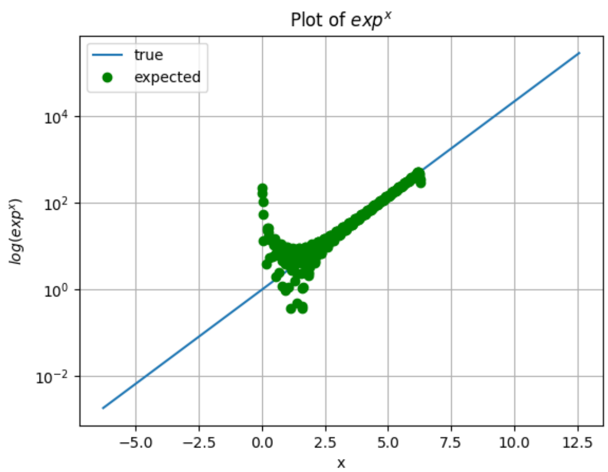
3.6 Comparing Predictions

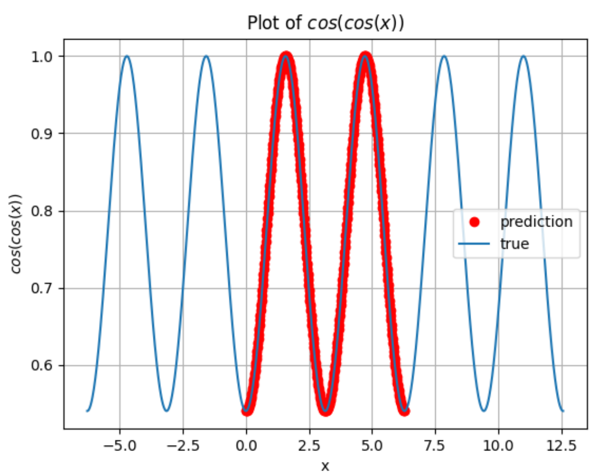
The maximum absolute error for e^x =1.3327308703353395
Whereas the same metric for cos(cos(x)) = 2.694184945384304e-15

3.7 Plotting Results

```
c_exp = np.reshape(c_exp,(51,1))

f_values = np.matmul(A_exp,c_exp)

x = np.linspace(0,2*np.pi,400)
plt.grid()
plt.title(r"Plot of $exp^x$")
plt.xlabel('x')
plt.ylabel(r'$log(exp^x)$')
t = np.linspace(-2*np.pi,4*np.pi,1000)
plt.semilogy(t,exp(t))
plt.semilogy(x,f_values,'go')
plt.legend(["true","prediction"])
plt.savefig('fig5.png')
plt.close()

c_coscos = np.reshape(c_coscos,(51,1))

f_values = np.matmul(A_coscos,c_coscos)

x = np.linspace(0,2*np.pi,400)
plt.grid()
plt.title(r"Plot of $cos(cos(x))$")
plt.xlabel('x')
plt.ylabel(r'$cos(cos(x))$')
t = np.linspace(-2*np.pi,4*np.pi,1000)
plt.plot(x,f_values,'ro')
plt.plot(t,coscos(t))
plt.legend(["true","prediction"])
plt.savefig('fig6.png')
plt.close()
```

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 $[0,2\pi)$. Hence it is acceptable that there is a large discrepancy in the predicted value of e^x at these boundaries


Plot of exp^x
true
expected
10^4
10^2
10^0
10^{-2}
$log(exp^x)$
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
x


Plot of $cos(cos(x))$
1.0
0.9
0.8
0.7
0.6
$cos(cos(x))$
prediction
true
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
x

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.