Assignment No 4 - Fourier Approximations

Raghu Dinesh T

February 27, 2020

1 Abstract

The basic goal and outcome of this assignment could be understood as follows:

a) We choose two functions f(x)=exp(x) and g(x) = cos(cos(x))

b) The functions' values are approximated for x in the range 0 to 2pi by calculating the first 51 fourier coefficients

c) The fourier coefficients are calculated using two different methods and are then plotted in semilog and loglog axes for both the functions:
1)direct integration
2)least squares method

d)We compare the coefficients obtained from both the methods are plot the error between them for both the functions

-We finally find the functions's values at each point using the fourier coefficients obtained so that we get an estimate of how good the approximation of taking the first 50 fourier coefficents holds good

2 SubTasks

2.1 SubTask 1

Python functions are written for exp(x) and cos(cos(x)) and plotted from x= -4pi to 2pi in fig(0) and fig(1) respectively

-While cos(cos(x)) turns out to be periodic with a period of 2pi, exp(x) is not periodic.

-Hence we define f(x) as exp(x%2pi).It now becomes a function of period 2pi and it's equal to exp(x) from 0 to 2pi.This is the expected function using fourier approximation.This is also plotted in fig(0)

Code:

```
def exp(x):
return pl.exp(x)
def f(x):
return pl.cos(pl.cos(x))

x=pl.linspace(-2*pl.pi,4*pl.pi,1000)

pl.figure(1)
pl.title("Actual")
pl.semilogy(x,np.absolute(exp(x)))
pl.xlabel("x")
pl.ylabel("e^x")
pl.grid()
pl.title("Fourier")
pl.semilogy(x,np.absolute(exp(x%(2*pl.pi))))
pl.xlabel("x")
pl.ylabel("e^x")
pl.grid()


pl.figure(2)
pl.title("Actual")
pl.plot(x,f(x))
pl.xlabel("x")
pl.ylabel("cos(cos(x))")
pl.grid()
pl.title("Fourier")
pl.plot(x,f(x%(2*pl.pi)))
pl.xlabel("x")
pl.ylabel("cos(cos(x))")
pl.grid()
```

2.2 SubTask 2

Finding the fourier coefficients using direct integration
Code:

```
NOC=26

e=[0];c=[0]

e[0]=(0.5/pl.pi)*si.quad(exp,0,2*pl.pi)[0]
c[0]=(0.5/pl.pi)*si.quad(f,0,2*pl.pi)[0]
```

```
for k in range(1,NOC):
e.append((1/pl.pi)*si.quad(ue,0,2*pl.pi,args=(k))[0])
e.append((1/pl.pi)*si.quad(ve,0,2*pl.pi,args=(k))[0])
c.append((1/pl.pi)*si.quad(uc,0,2*pl.pi,args=(k))[0])
c.append((1/pl.pi)*si.quad(vc,0,2*pl.pi,args=(k))[0])

n=range(1,NOC)
```

2.3 SubTask 3

Fourier coefficients are found out for both the funcions using the least square method
Code:

```
x=np.linspace(0,2*pl.pi,401)
x=x[:-1] # drop last term to have a proper periodic integral
# f has been written to take a vector
A=np.zeros((400,51))
# allocate space for A
A[:,0]=1
# col 1 is all ones
for k in range(1,26):
A[:,2*k-1]=pl.cos(k*x) # cos(kx) column
A[:,2*k]=pl.sin(k*x)
# sin(kx) column
#endfor
c1=sl.lstsq(A,exp(x))[0]
c2=sl.lstsq(A,f(x))[0]
# the [0] is to pull out the
# best fit vector. lstsq returns a list.
```

2.4 SubTask 4

The coefficients obtained from both the methods are plotted in both semilog and loglog scales for both the functions
Code:

```
pl.figure(3)
pl.semilogy(0,abs(e[0]),"ro")
pl.semilogy(n,np.absolute(e[1:51:2]),"ro",label="fourier coefficients")
pl.semilogy(n,np.absolute(e[2:51:2]),"ro")
pl.title("Fourier coefficients of e^x vs n on semilog scale")
```

```
pl.ylabel("Fourier coefficients")
pl.xlabel("n")

pl.figure(4)
pl.loglog(n,np.absolute(e[1:51:2]),"ro",label="fourier coefficients")
pl.loglog(n,np.absolute(e[2:51:2]),"ro")
pl.title("Fourier coefficients of e^x vs n on loglog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")

pl.figure(5)
pl.semilogy(0,abs(c[0]),"ro")
pl.semilogy(n,np.absolute(c[1:51:2]),"ro",label="fourier coefficients")
pl.semilogy(n,np.absolute(c[2:51:2]),"ro")
pl.title("Fourier coefficients of cos(cos(x)) vs n on semilog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")

pl.figure(6)
pl.loglog(n,np.absolute(c[1:51:2]),"ro",label="fourier coefficients")
pl.loglog(n,np.absolute(c[2:51:2]),"ro")
pl.title("Fourier coefficients of cos(cos(x)) vs n on loglog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")

pl.figure(3)
pl.semilogy(0,abs(c1[0]),"go")
pl.semilogy(n,np.absolute(c1[1:51:2]),"go",label="least mean square coefficients")
pl.semilogy(n,np.absolute(c1[2:51:2]),"go")
pl.title("Fourier coefficients of e^x vs n on semilog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")
pl.legend()

pl.figure(4)
pl.loglog(n,np.absolute(c1[1:51:2]),"go",label="least mean square coefficients")
pl.loglog(n,np.absolute(c1[2:51:2]),"go")
pl.title("Fourier coefficients of e^x vs n on loglog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")
pl.legend()

pl.figure(5)
pl.semilogy(0,abs(c2[0]),"go")
```

```
pl.semilogy(n,np.absolute(c2[1:51:2]),"go",label="least mean square coefficients")
pl.semilogy(n,np.absolute(c2[2:51:2]),"go")
pl.title("Fourier coefficients of cos(cos(x)) vs n on semilog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")
pl.legend()

pl.figure(6)
pl.loglog(n,np.absolute(c2[1:51:2]),"go",label="least mean square coefficients")
pl.loglog(n,np.absolute(c2[2:51:2]),"go")
pl.title("Fourier coefficients of cos(cos(x)) vs n on loglog scale")
pl.ylabel("Fourier coefficients")
pl.xlabel("n")
pl.legend()
```

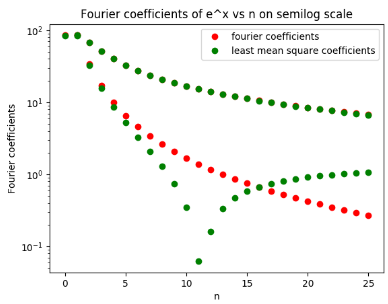

Figure 1: Fourier coefficients of f(x) in semilogy axis

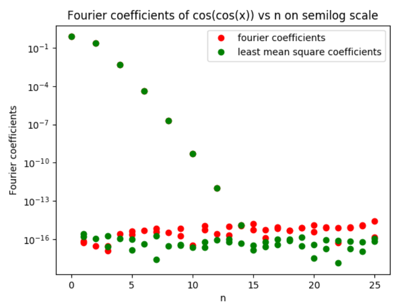

Figure 2: Fourier coefficients of g(x) in semilogy axis

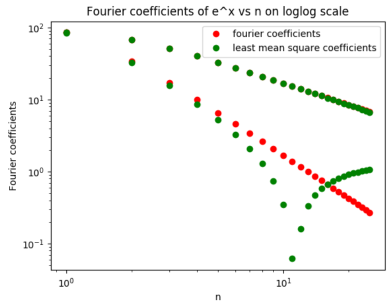

Figure 3: Fourier coefficients of f(x) in loglog axis

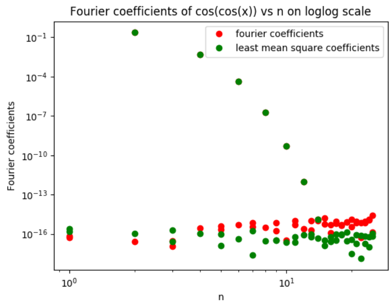

Figure 4: Fourier coefficients of g(x) in loglog axis

2.5 SubTask 5

The absolute errors between the coefficients obtained by both the methods are found out and the maximum error is printed. **Code:**

```
erre=np.absolute(c1-e)
errf=np.absolute(c2-c)
mde=np.amax(erre)
mdf=np.amax(errf)
```

2.6 SubTask 6

Using the coefficents obtained from the least squares method,the value of f(x) and g(x) is found out for each value of x and plotted in the original figures:fig(0) and fig(1) where the functions were initially plotted
Code:

```
f1=A.dot(c1)
f2 =A.dot(c2)
figure(0)
semilogy(x,f1,'go')
```

```
figure(1)
semilogy(x,f2,'go')


show()
```

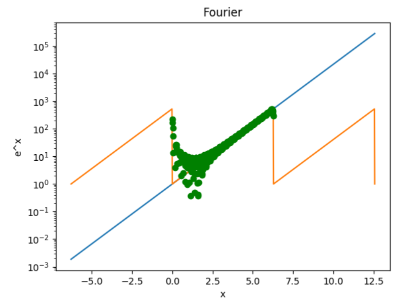

Figure 5: Plot of exp(x) along with approximated function using fourier series approximation

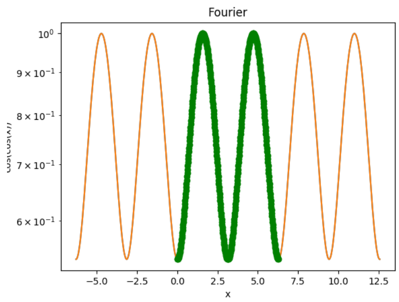

Figure 6: Plot of cos(cos(x)) along with approximated function using fourier series approximation

3 Conclusion

a) The bn coefficients obtained for g(x) are approximately 0 as it is an even function

b) The coefficients for exp(x) decrease less quickly then for cos(cos(x)) because higher harmonics also have significant value for exp(x) while lower harmonics have the most significant values for cos(cos(x))

c) The loglog plot for exp(x) looks linear, wheras the semilog plot in Figure 5 for cos(cos(x))linear.This is due to the corresponding behaviour of the fourier coefficients obtained analytically

d) The max error between the fourier coefficients calculated using integration and least squares method for exp(x) is : 1.332730870335368

e) The max error between the fourier coefficients calculated using integration and least squares method for cos(cosx) is : 2.5394379427569514e-15

f) We observe that the values obtained using fourier series closely agree for cos(cos(x)) but not for exp(x).This is because higher fourier coefficients have significant magnitudes for exp(x) but not for cos(cos(x))

g) For cos(cos(x)) the coefficients greater than 50 do not have significant magnitudes.