Assignment 4

Adithya

February 28, 2019

1 Fourier Approximations

1.1 Plotting exp(x) and cos(cos(x))

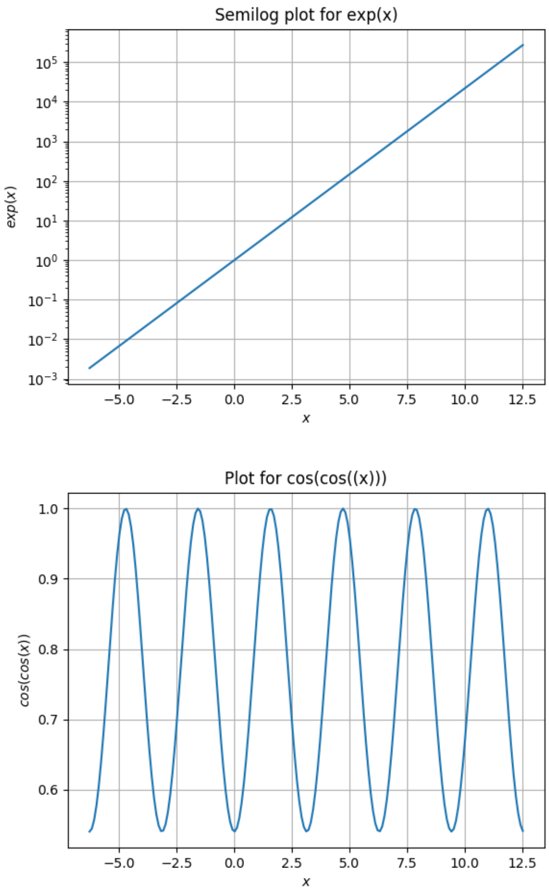


Figure 1: exp() and cos(cos()) vs x

```
def EXP(x):
     return np.exp(x)
def coscos(x):
     return (np.cos(np.cos(x)))

x = np.arange(-2*np.pi,4*np.pi,0.1)
y1 = EXP(x)
z=coscos(x)

plt.figure(0)
plt.semilogy(x,y1)
plt.xlabel('$x$')
plt.ylabel('$exp(x)$')
plt.title('$Semilog plot for exp(x)$')
plt.grid(True)
plt.show()

plt.figure(1)
plt.plot(x,z)
plt.xlabel('$x$')
plt.ylabel('$cos(cos(x))$')
plt.title('$Plot for cos(cos((x)))$')
plt.grid(True)
plt.show()
```

cos(cos(x)) is periodic but exp(x) is non-periodic.

1.2 Obtaining Fourier Coefficients

Code for getting Fourier Coefficients:

```
def cosm(x,k,f):
     return np.cos(k*x)*f(x)
def sinm(x,k,f):
     return np.sin(k*x)*f(x)
exp_a0=(integral.quad(cosm,0,2*np.pi,args=(0,EXP))[0])/(2*np.pi)
exp_an=np.array([integral.quad(cosm,0,2*np.pi,args=(k,EXP))[0] for k in range(1,26)]
exp_bn=np.array([integral.quad(sinm,0,2*np.pi,args=(k,EXP))[0] for k in range(1,26)]

coscos_a0=(integral.quad(cosm,0,2*np.pi,args=(0,coscos))[0])/(2*np.pi)
coscos_an=np.array([integral.quad(cosm,0,2*np.pi,args=(k,coscos))[0] for k in range(
coscos_bn=np.array([integral.quad(sinm,0,2*np.pi,args=(k,coscos))[0] for k in range(
```

1.3 Plotting Coefficients

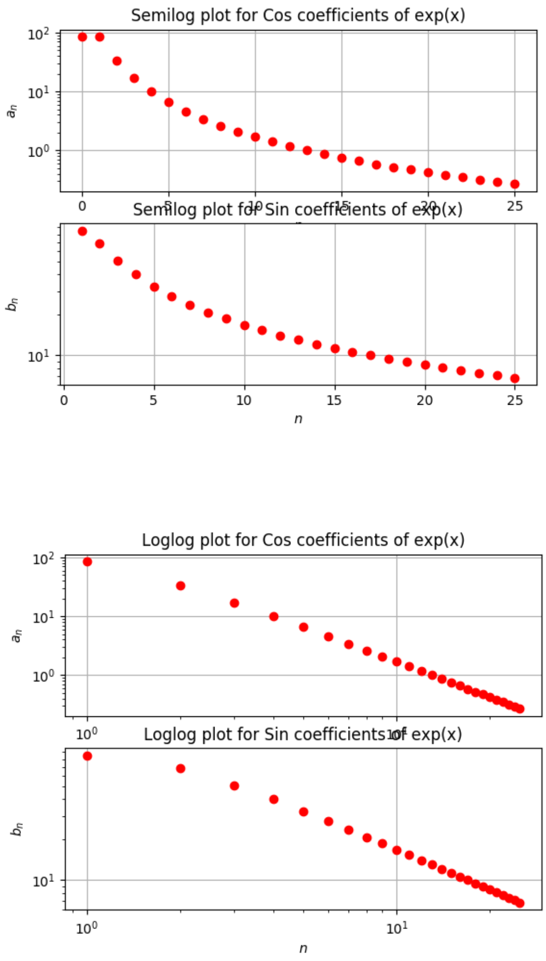


Figure 2: Semilog and loglog plots of exp(x)

1.4 Creating Matrices

This is how we get A,b matrices.

```
y=np.linspace(0,2*np.pi,401)
y=y[:-1]
b_exp=EXP(y)
b_coscos=coscos(y)
A=np.zeros((400,51))
A[:,0]=1
```

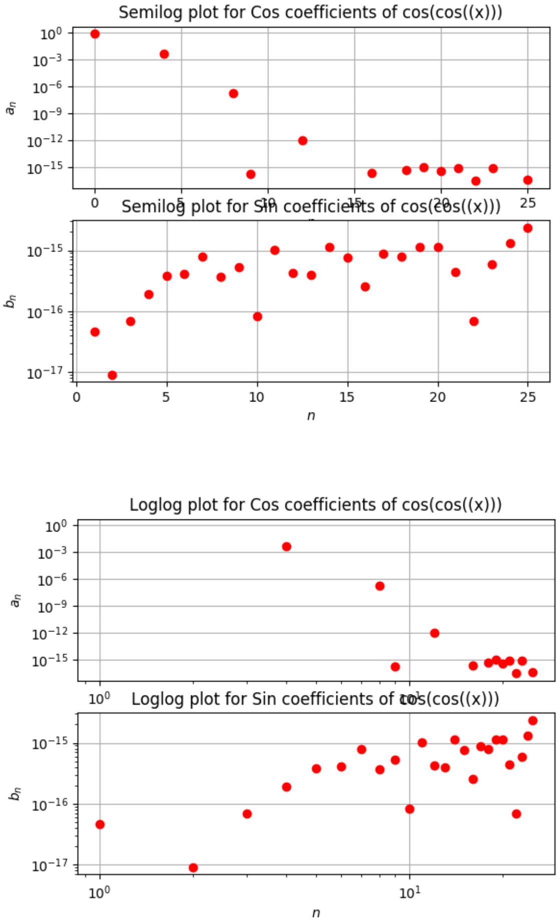


Figure 3: Semilog and loglog plots of cos(cos((x))

```
for k in range(1,26):
    A[:,k]=np.cos(k*y)
    A[:,k+25]=np.sin(k*y)
```

1.5 Using lstsq

What this does is to find the best fit numbers that will satisfy Equation at exactly the points we have evaluated f (x).

```
c1=lstsq(A,b_exp)[0]
c2=lstsq(A,b_coscos)[0]
```

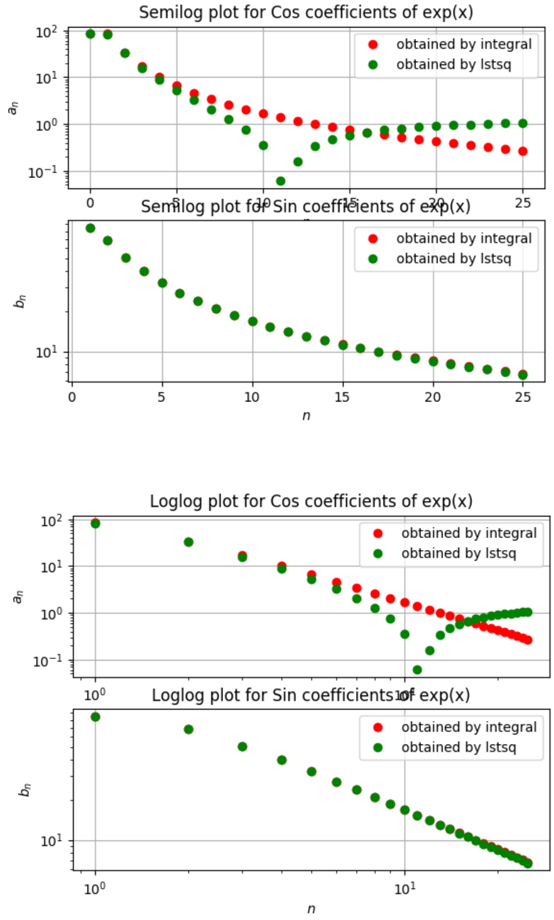
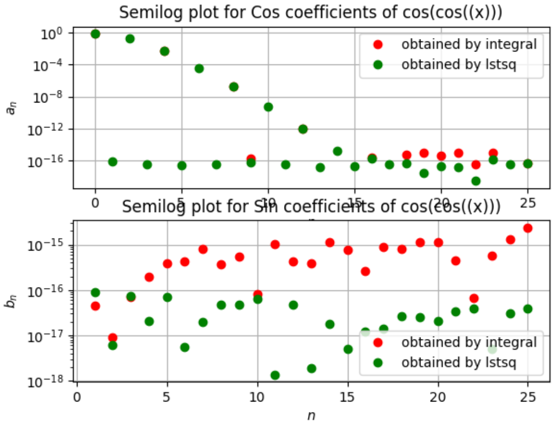


Figure 4: Semilog and loglog plots of exp(x)



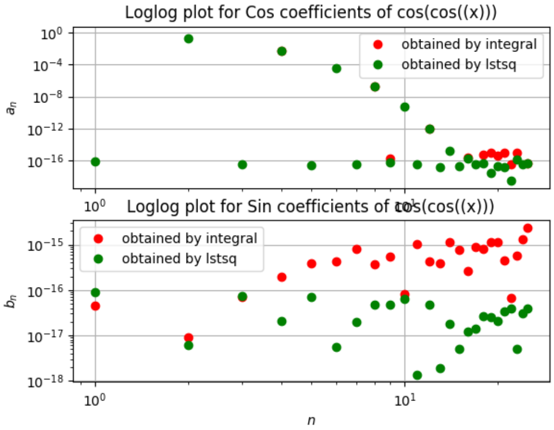


Figure 5: Semilog and loglog plots of cos(cos((x))

1.6 Finding the maximum deviation

We can directly get it from vectors by this.

```
 np.amax(np.abs(c1-np.asarray(exp_coeff)))
np.amax(np.abs(c2-np.asarray(coscos_coeff)))
```

Output:
Maximum deviation in case of exp(x) is 1.3327308703353253
Maximum deviation in case of cos(cos(x)) is 2.646921459797403e-15

1.7 Comparing Original and Obtained Functions:

```
b_est_exp=np.dot(A,c1)
b_est_coscos=np.dot(A,c2)
```

Because of the unperiodic nature of exponential function, we can not properly fit the values. But, it is not the case with cos(cos(x)) as it is periodic.

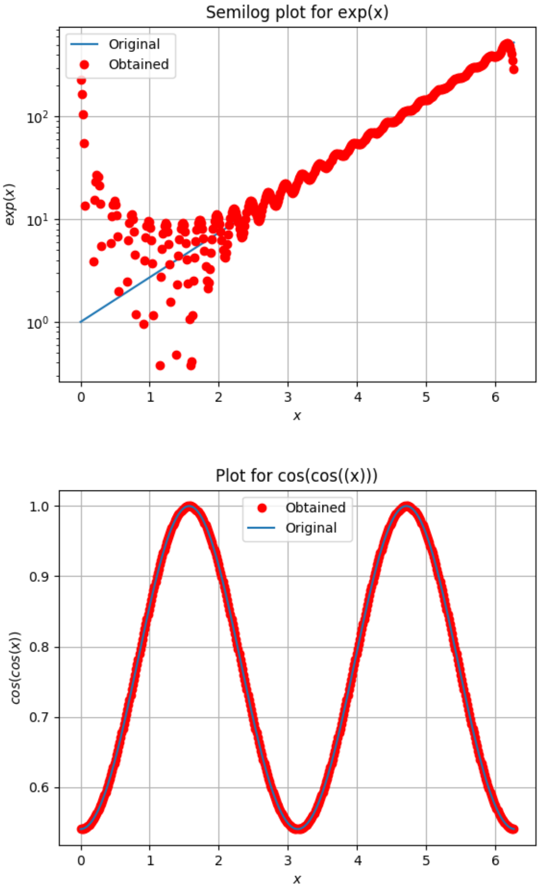


Figure 6: Obtained and Original functions