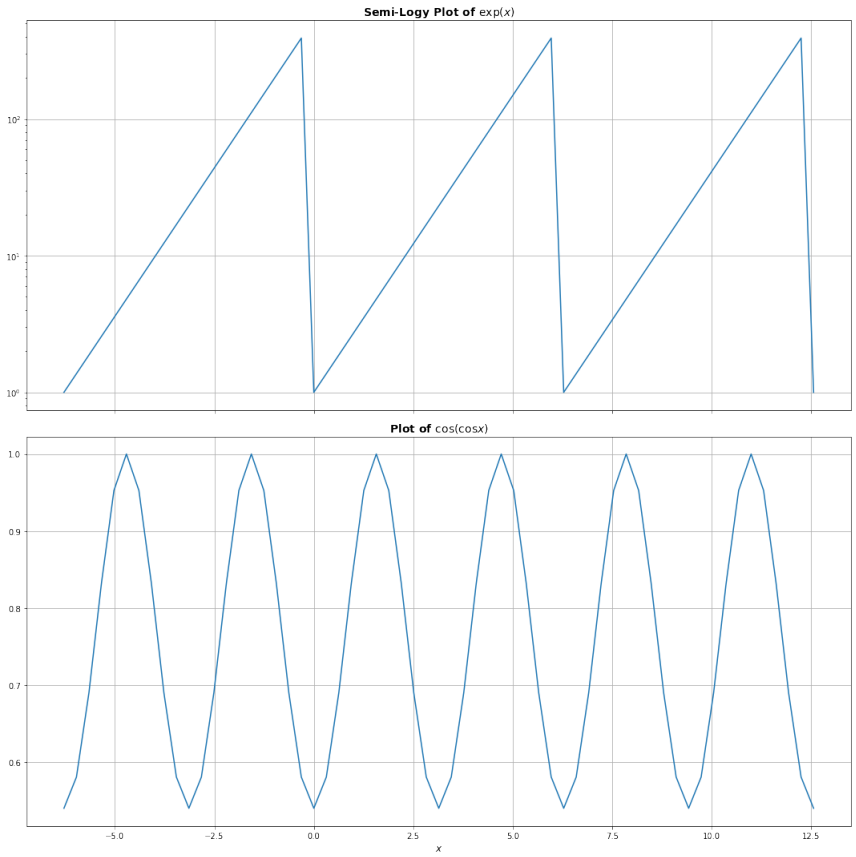
Assignment 4

Pruthvi Raj R G, EE17B114

February 28, 2019

1 Fourier Approximations

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




```
import numpy as np

def make_periodic(oldfunc):
    def newfunc(x):
        return oldfunc(np.remainder(x,2*np.pi))
```

```
    return newfunc

@make_periodic
def exp(x):
    return np.exp(x)

exp=make_periodic(exp)
def cos_cos(x):
    return np.cos(np.cos(x))

import matplotlib.pyplot as plt


import pylab
pylab.rcParams['figure.figsize'] = (7, 7)

figure, axarr = plt.subplots(2, sharex=True)


axarr[0].set_title(r'Semi-Logy Plot of $\exp(x)$', fontsize=10)
axarr[1].set_title(r'Plot of $\cos{(\cos{x})}$', fontsize=10)

plt.xlabel('$x$',fontsize=12)

x=np.linspace(-2,4,6/0.1+1)
x=x*np.pi
axarr[0].semilogy(x,make_periodic(exp)(x))
axarr[1].plot(x,cos_cos(x))


figure.tight_layout()
plt.show()
```

1.2 Finding and Plotting Fourier Coefficients

```
[fontsize=small]
from scipy import integrate as integrate

def cos_u(x,k,f):
    # Compute cosine Fourier Components
    return f(x)*np.cos(k*x)
```

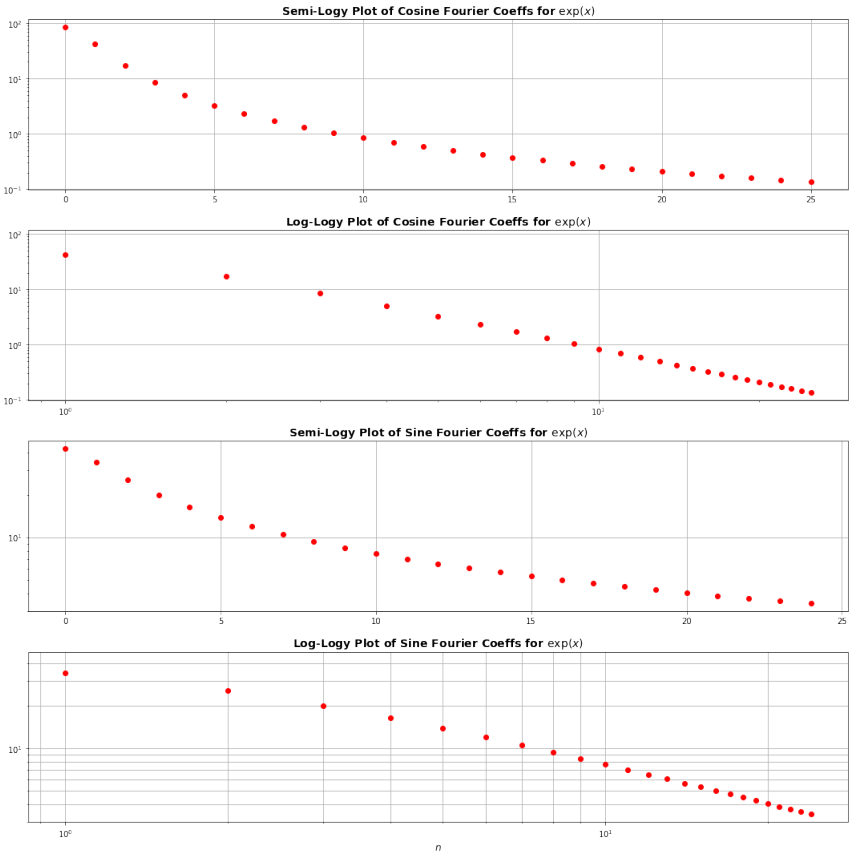
```
def sin_u(x,k,f):
    # Compute sine Fourier Components
    return f(x)*np.sin(k*x)

# For exp (x)
exp_a0=np.array(integrate.quad(cos_u,0,2*np.pi,args=(0,exp))[0])/(2*np.pi)
exp_a=np.array([integrate.quad(cos_u,0,2*np.pi,args=(k,exp))[0] for k in range(1,26)])/(2*
exp_b=np.array([integrate.quad(sin_u,0,2*np.pi,args=(k,exp))[0] for k in range(1,26)])/(2*

# For cos(cos (x))
cos_cos_a0 = np.array(integrate.quad(cos_u,0,2*np.pi,args=(0,cos_cos))[0])/(2*np.pi)
cos_cos_a = np.array([integrate.quad(cos_u,0,2*np.pi,args=(k,cos_cos))[0] for k in range(1
cos_cos_b = np.array([integrate.quad(sin_u,0,2*np.pi,args=(k,cos_cos))[0] for k in range(1

# b_n for cos_cos_b is nearly zero
print(cos_cos_b)

figure, axarr = plt.subplots(4)
pylab.rcParams['figure.figsize'] = (7, 7)
axarr[0].set_title(r'Semi-Logy Plot of Cosine Fourier Coeffs for $\exp(x)$', fontsize=14,f
axarr[1].set_title(r'Log-Logy Plot of Cosine Fourier Coeffs for $\exp(x)$', fontsize=14,fo
axarr[2].set_title(r'Semi-Logy Plot of Sine Fourier Coeffs for $\exp(x)$', fontsize=14,fon
axarr[3].set_title(r'Log-Logy Plot of Sine Fourier Coeffs for $\exp(x)$', fontsize=14,font


plt.xlabel('$n$',fontsize=12)

n=list(range(51))

exp_list=[exp_a0]
exp_list[1:26]=exp_a
exp_list[26:52]=exp_b

axarr[0].semilogy(list(range(26)),np.array(exp_list[:26]),'ro')
axarr[1].loglog(list(range(26)),np.array(exp_list[:26]),'ro')
axarr[2].semilogy(list(range(25)),np.array(np.abs(exp_list[26:])),'ro')
axarr[3].loglog(list(range(25)),np.array(np.abs(exp_list[26:])),'ro')
figure.tight_layout()
plt.show()


figure, axarr = plt.subplots(4)
axarr[0].set_title(r'Semi-Logy Plot of Fourier Coeffs for $\cos{(\cos{x})}$', fontsize=14,
axarr[1].set_title(r'Log-Log Plot of Fourier Coeffs for $\cos{(\cos{x})}$', fontsize=14,fo
axarr[2].set_title(r'Semi-Logy Plot of Sine Fourier Coeffs for $\cos{(\cos{x})}$', fontsiz
axarr[3].set_title(r'Log-Logy Plot of Sine Fourier Coeffs for $\cos{(\cos{x})}$', fontsize

plt.xlabel('$n$',fontsize=12)

n=list(range(51))
```

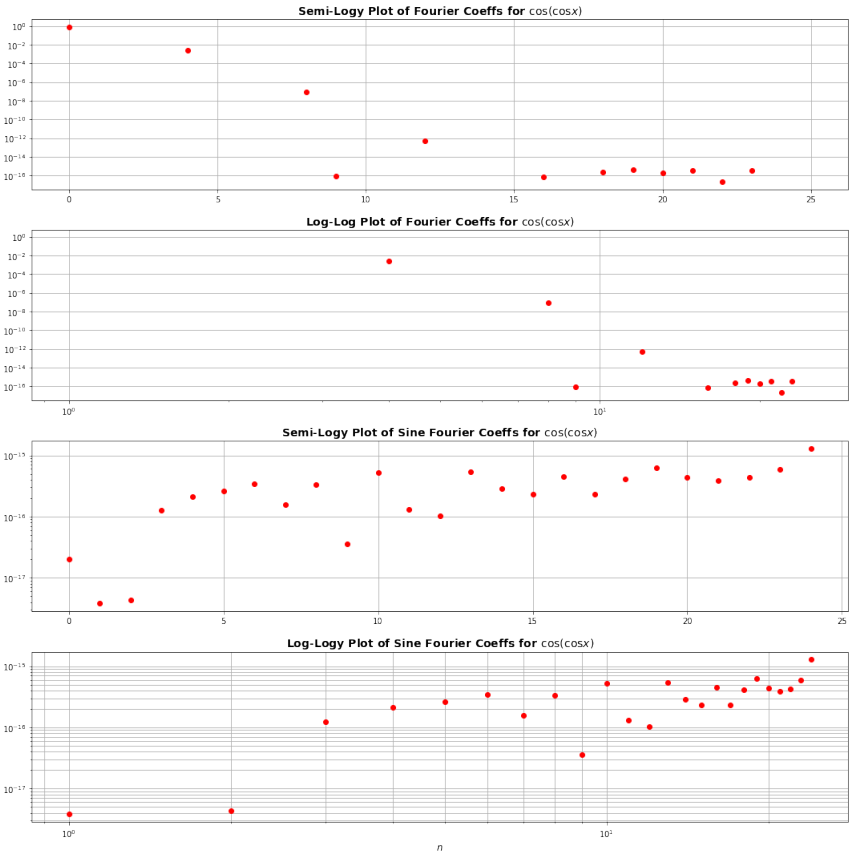
```
cos_cos_list=[cos_cos_a0]
cos_cos_list[1:26]=cos_cos_a
cos_cos_list[26:52]=cos_cos_b

axarr[0].semilogy(list(range(26)),np.array(cos_cos_list[:26]),'ro')
axarr[1].loglog(list(range(26)),np.array(cos_cos_list[:26]),'ro')
axarr[2].semilogy(list(range(25)),np.array(np.abs(cos_cos_list[26:])),'ro')
axarr[3].loglog(list(range(25)),np.array(np.abs(cos_cos_list[26:])),'ro')
figure.tight_layout()
plt.show()

figure, axarr = plt.subplots(2)
axarr[0].set_title(r'Semi-Logy Plot of sine Fourier Coeffs for $\cos{(\cos{x})}$', fontsiz
axarr[1].set_title(r'Log-Log Plot of sine Fourier Coeffs for $\cos{(\cos{x})}$', fontsize=


plt.xlabel('$n$',fontsize=12)

n=list(range(25))

cos_cos_list=[cos_cos_a0]
cos_cos_list[1:26]=cos_cos_a
cos_cos_list[26:52]=cos_cos_b

axarr[0].semilogy(n,np.array(cos_cos_list)[26:],'ro')
axarr[1].loglog(n,cos_cos_list[26:],'ro')
figure.tight_layout()
plt.show()
```

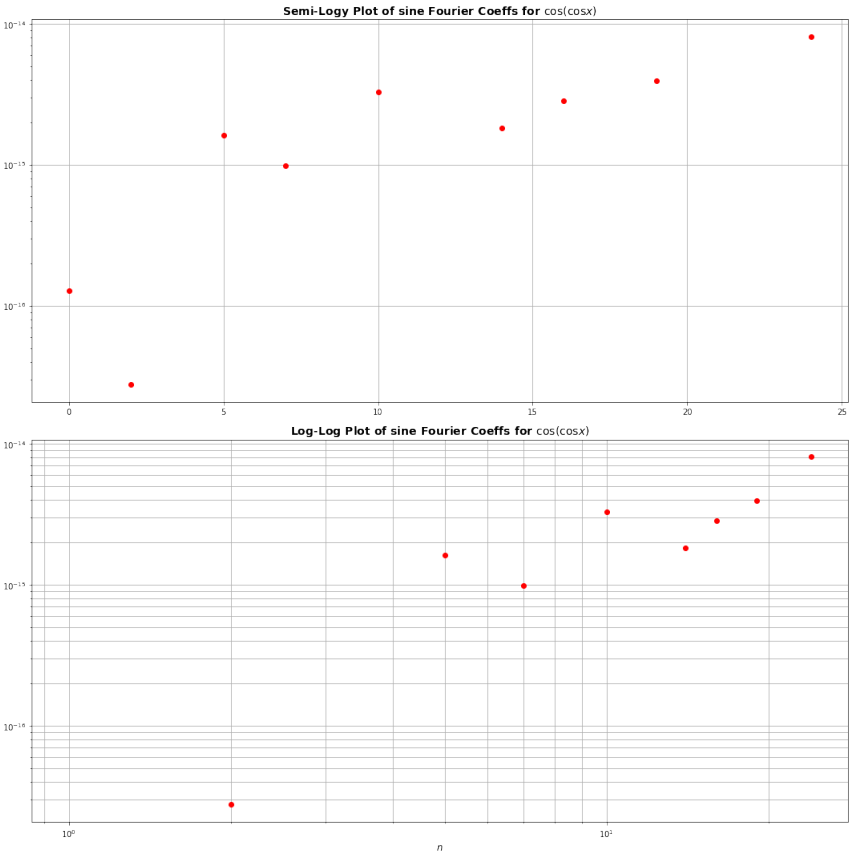
1.3 Creating Matrices

This is how we get A,b matrices.

```
def create_matrices(f,x,n,c):
    b=f(x)
    A=np.zeros((n,c))
    A[:,0]=1

    for i in range(1,26):
        A[:,i]=np.cos(i*x)

    for i in range(1,26):
        A[:,i+25]=np.sin(i*x)

    return (A,b)

x=np.linspace(0,2*np.pi,num=400)
```


Semi-Logy Plot of Cosine Fourier Coeffs for exp(x)
Log-Logy Plot of Cosine Fourier Coeffs for exp(x)
Semi-Logy Plot of Sine Fourier Coeffs for exp(x)
Log-Logy Plot of Sine Fourier Coeffs for exp(x)
n


Semi-Logy Plot of Fourier Coeffs for cos(cosx)
Log-Log Plot of Fourier Coeffs for cos(cosx)
Semi-Logy Plot of Sine Fourier Coeffs for cos(cosx)
Log-Logy Plot of Sine Fourier Coeffs for cos(cosx)
n

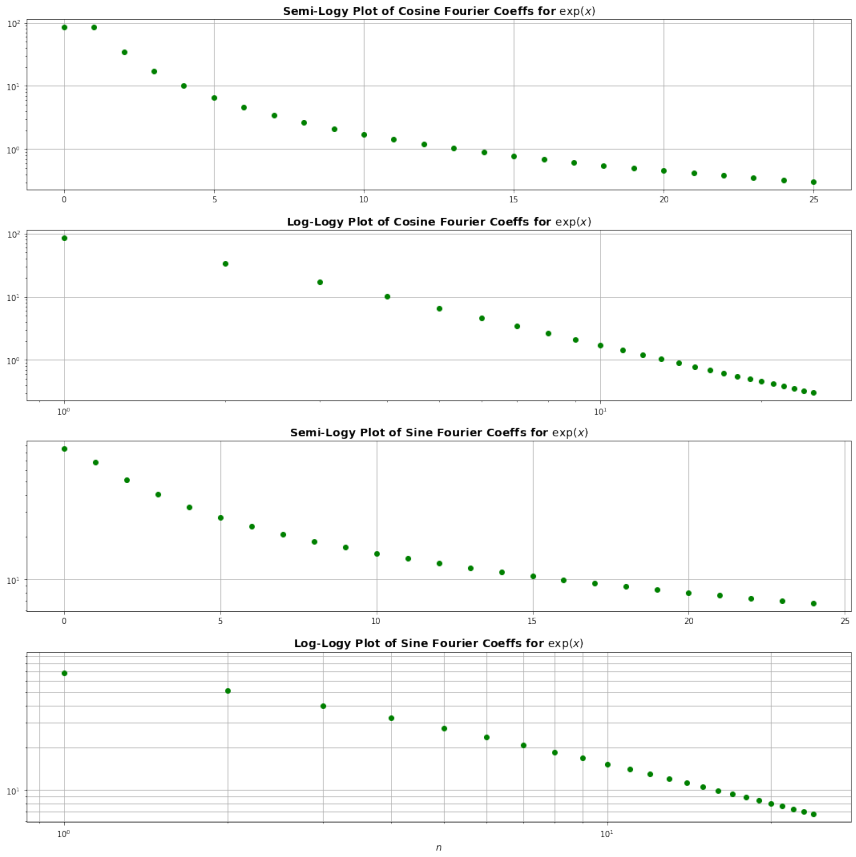

Semi-Logy Plot of sine Fourier Coeffs for cos(cosx)
10^{-14}
10^{-15}
10^{-16}
0
5
10
15
20
25
Log-Log Plot of sine Fourier Coeffs for cos(cosx)
10^{-14}
10^{-15}
10^{-16}
10^{0}
10^{1}
n

```
A_cos_cos,b_cos_cos=(create_matrices(cos_cos,x,len(x),51))
A_exp,b_exp=(create_matrices(exp,x,len(x),51))
```

1.4 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).

```
from scipy.linalg import lstsq

c_exp=lstsq(A_exp,b_exp)[0]
c_cos_cos=lstsq(A_cos_cos,b_cos_cos)[0]
figure, axarr = plt.subplots(4)
axarr[0].set_title(r'Semi-Logy Plot of Cosine Fourier Coeffs for $\exp(x)$', fontsize=14,f
axarr[1].set_title(r'Log-Logy Plot of Cosine Fourier Coeffs for $\exp(x)$', fontsize=14,fo
axarr[2].set_title(r'Semi-Logy Plot of Sine Fourier Coeffs for $\exp(x)$', fontsize=14,fon
axarr[3].set_title(r'Log-Logy Plot of Sine Fourier Coeffs for $\exp(x)$', fontsize=14,font

plt.xlabel('$n$',fontsize=12)

n=list(range(51))

axarr[0].semilogy(list(range(26)),np.array(c_exp[:26]),'go')
axarr[1].loglog(list(range(26)),np.array(c_exp[:26]),'go')
axarr[2].semilogy(list(range(25)),np.array(np.abs(c_exp[26:])),'go')
axarr[3].loglog(list(range(25)),np.array(np.abs(c_exp[26:])),'go')
figure.tight_layout()
plt.show()

figure, axarr = plt.subplots(4)
axarr[0].set_title(r'Semi-Logy Plot of Fourier Coeffs for $\cos{(\cos{x})}$', fontsize=14,
axarr[1].set_title(r'Log-Log Plot of Fourier Coeffs for $\cos{(\cos{x})}$', fontsize=14,fo
axarr[2].set_title(r'Semi-Logy Plot of Sine Fourier Coeffs for $\cos{(\cos{x})}$', fontsiz
axarr[3].set_title(r'Log-Logy Plot of Sine Fourier Coeffs for $\cos{(\cos{x})}$', fontsize


plt.xlabel('$n$',fontsize=12)

n=list(range(51))

axarr[0].semilogy(list(range(26)),np.array(cos_cos_list[:26]),'go')
axarr[1].loglog(list(range(26)),np.array(cos_cos_list[:26]),'go')
axarr[2].semilogy(list(range(25)),np.array(np.abs(cos_cos_list[26:])),'go')
axarr[3].loglog(list(range(25)),np.array(np.abs(cos_cos_list[26:])),'go')
figure.tight_layout()
plt.show()
```



1.5 Finding the maximum deviation

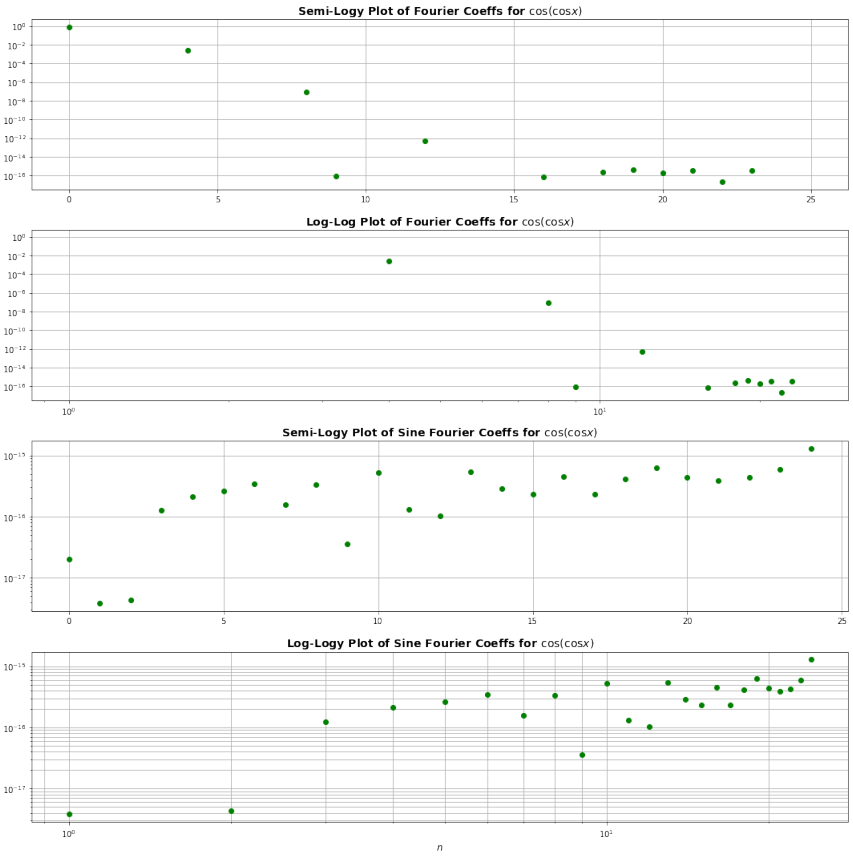
We can directly get it from vectors by this.

```
def error (u,v):
    return np.amax(np.abs(u-v))

error(1,2)
print ("Argmax Error on fourier coefficients for $\cos{\cos{(x)}}$ \t",error(c_cos_cos,cos
print(c_exp)
print(exp_list)
print ("Argmax Error on fourier coefficients for $\exp{(x)}$ \t",error(c_exp,exp_list))
```

Output:
Maximum deviation in case of exp(x) is 42.566244812
Maximum deviation in case of cos(cos(x)) is 0.114903484932

1.6 Comparing Original and Obtained Functions:

```
b_eval_exp=np.matmul(A_exp,c_exp.T)
b_eval_cos_cos=np.matmul(A_cos_cos,c_cos_cos.T)
print(len(b_eval_exp))
```




```
print(len(b_eval_cos_cos))

figure, axarr = plt.subplots(2, sharex=True)



axarr[0].set_title(r'SPlot of $\exp(x)$', fontsize=14,fontweight="bold")
axarr[1].set_title(r'Plot of $\cos{(\cos({x}))}$', fontsize=14,fontweight="bold")

plt.xlabel('$x$',fontsize=12)

axarr[0].semilogy(x,make_periodic(exp)(x))
axarr[0].semilogy(x,b_eval_exp,'ro')
axarr[1].plot(x,cos_cos(x))
axarr[1].semilogy(x,b_eval_cos_cos,'ro')

figure.tight_layout()
axarr[0].legend(('Actual function','Estimated by Regression'))
axarr[1].legend(('Actual function','Estimated by Regression'))
plt.show()
```

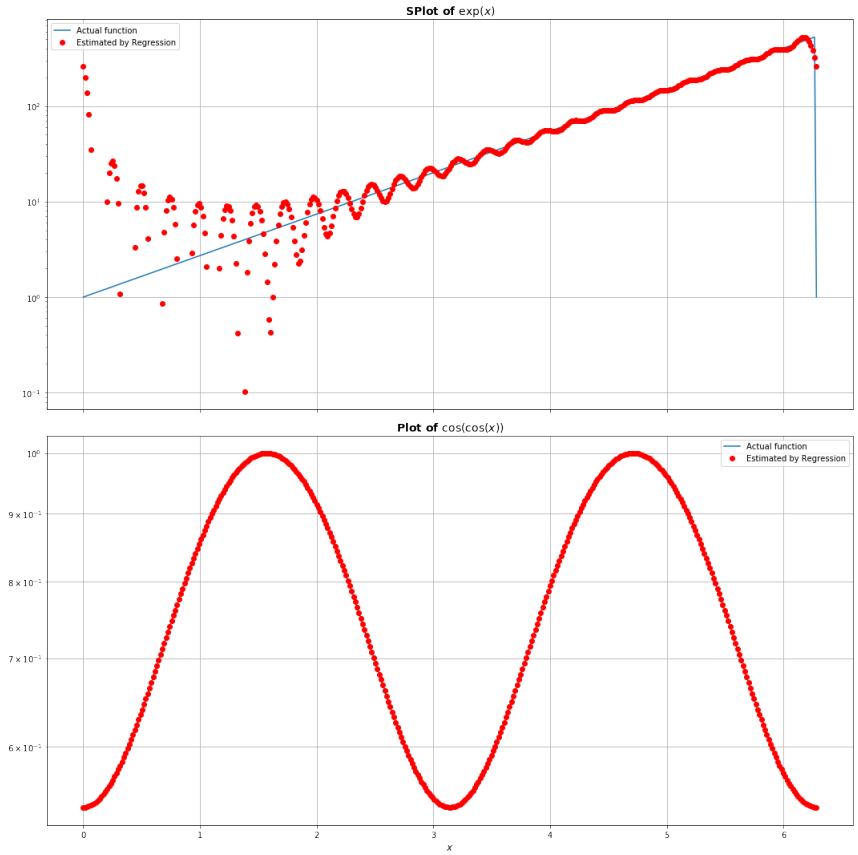


Figure 1: Obtained and Original functions

Since the function exponential of x is non periodic we will not be able to find the exact fourier coefficients
This is not the case of cos of cos x as it is periodic.