Assignment No 4

EE18B120; Sarvani Cheruvu

February 26, 2020

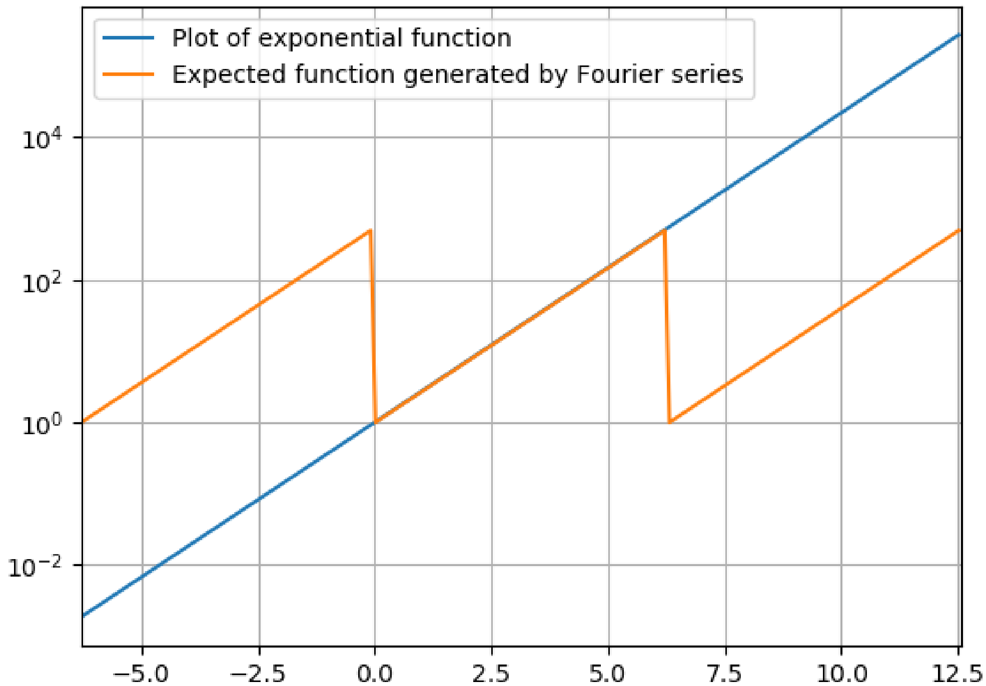
1 Abstract

This experiment fits two functions e^x and $cos(cosx)$ over the interval $[0,2*\pi)$ using the Fourier series.

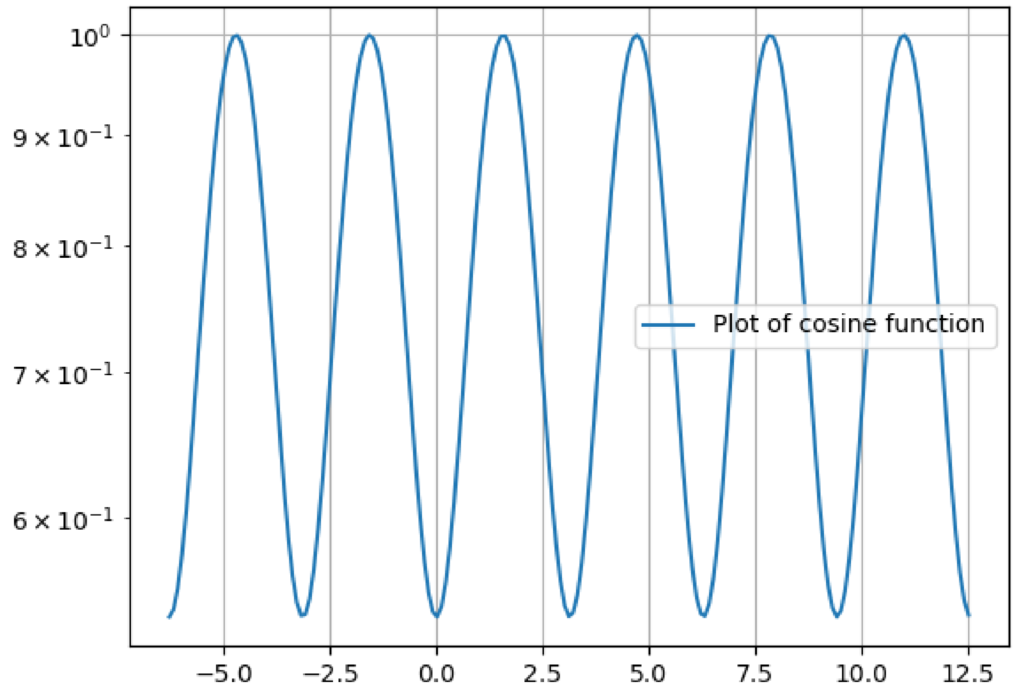
1.1 Function plotting

The function generated by Fourier series is periodic. The expected function for $cos(cos(x))$ will be the same, but for e^x it is a periodic repetition of the function in $[0,2*\pi)$.

```
data=np.arange(-2*pi,4*pi,0.1)
data1=np.arange(0,2*pi,0.1)
figure(1)
xlim(-2*pi,4*pi)
semilogy(data,e(data),label='Plot of exponential
    function')
legend()
semilogy(data,np.concatenate((e(data1),e(data1),e(data1))),label='Expected
    function generated by Fourier series')
legend()
grid(True)

figure(2)
semilogy(data,cosine(data),label='Plot of cosine
    function')
legend()
grid(True)
```



Plot of exponential function
Expected function generated by Fourier series
10^4
10^2
10^0
10^{-2}
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5



1.2 Coefficients: computation and plotting

The coefficients of the two functions are computed.

a. The integral is an odd function. In theory, we must obtain 0 in all cases. However, the computer implementation of integrals is not equivalent to a manual method, which is why we only get a value approximately equal to 0.

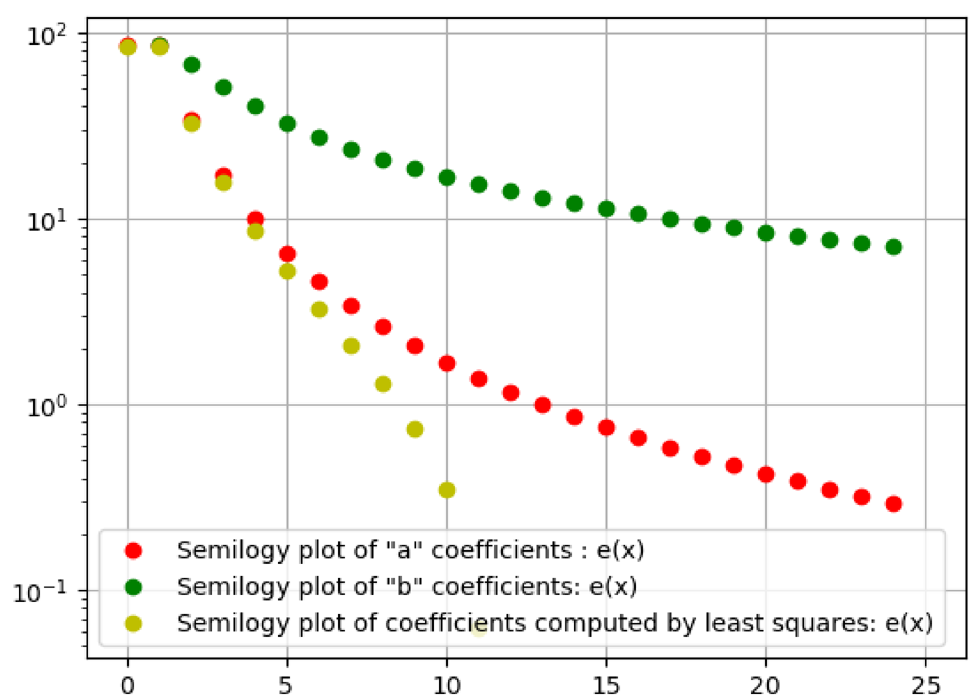
b. Coefficients in the case of e^x vary as $1/(1+n^2)$, which is why the decay is slow. In the second case, the integral cannot be written as a derivative of any function, which is why it is computed by numerical integration.

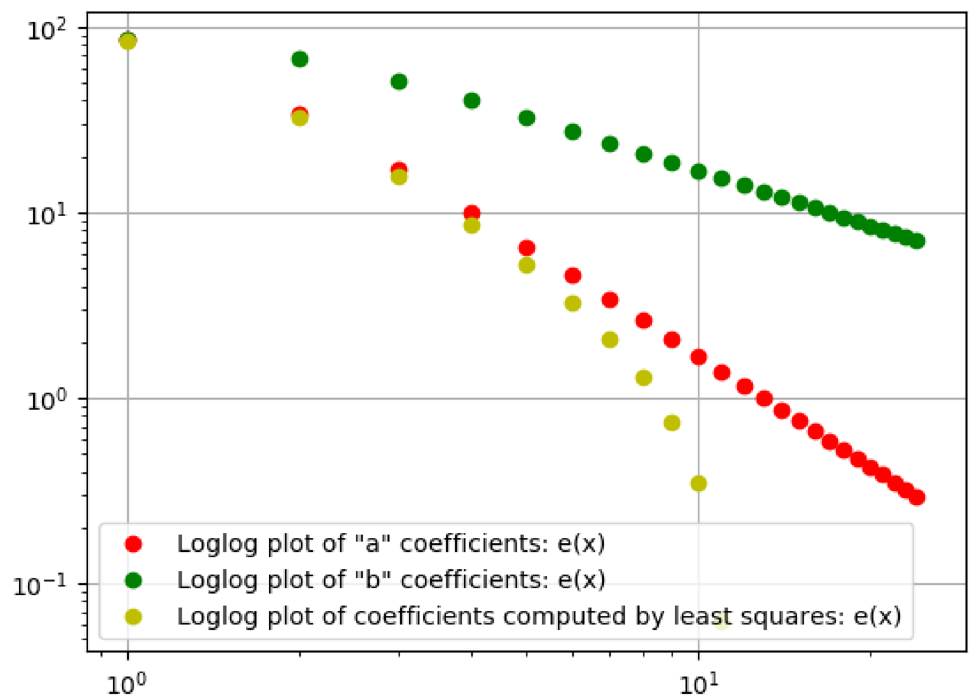
c. *log* $1/(1+n^2)$ varies approximately linearly on a logarithmic scale.

```
def u(x,k,f):
  return f(x)*cos(k*x)
def v(x,k,f):
  return f(x)*sin(k*x)
coeffea=np.zeros((26,1))
coeffeb=np.zeros((25,1))
coeffca=np.zeros((26,1))
coeffcb=np.zeros((25,1))
```

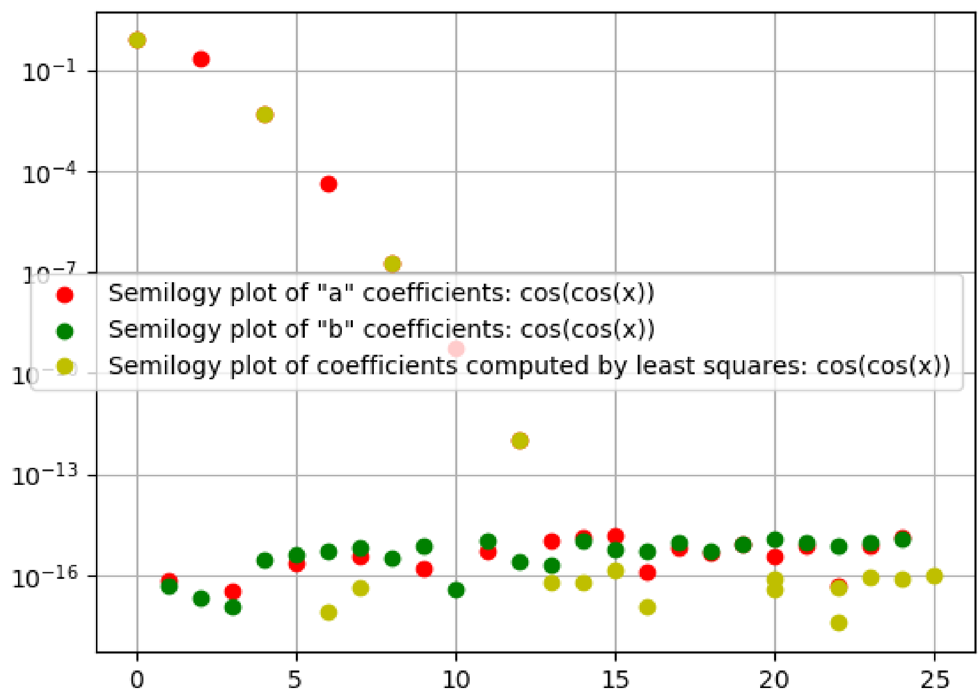
```python
coeffe=np.zeros((51,1,))
coeffc=np.zeros((51,1))
na=np.zeros((26,1))
na[0]=0
nb=np.zeros((25,1))
n=np.zeros((51,1))
coeffea[0]=(integrate.quad(u,0,2*pi,args=(0,e)))[0]/(2*pi)
coeffca[0]=(integrate.quad(u,0,2*pi,args=(0,cosine)))[0]/(2*pi)
coeffe[0]=coeffea[0]
coeffc[0]=coeffca[0]
for i in range(1,25):
    coeffea[i]=abs((integrate.quad(u,0,2*pi,args=(i,e)))[0]/pi)
    coeffeb[i-1]=abs((integrate.quad(v,0,2*pi,args=(i,e)))[0]/pi)
    coeffca[i]=abs((integrate.quad(u,0,2*pi,args=(i,cosine)))[0]/pi)
    coeffcb[i-1]=abs((integrate.quad(v,0,2*pi,args=(i,cosine)))[0]/pi)
    na[i]=i
    nb[i-1]=i
    coeffe[2*i-1]=coeffea[i]
    coeffe[2*i+1]=coeffeb[i-1]
    coeffc[2*i-1]=coeffca[i]
    coeffc[2*i+1]=coeffcb[i-1]
    n[2*i-1]=i
    n[2*i]=i
na[25]=25
nb[24]=25
n[50]=25
n[49]=25

figure(3)
semilogy(na,coeffea,linewidth=0.0,marker='o',color='r',label='Semilogy
    plot of "a" coefficients : e(x)')
legend()
semilogy(nb,coeffeb,linewidth=0.0,marker='o',color='g',label='Semilogy
    plot of "b" coefficients: e(x)')
legend()
grid(True)

figure(4)
loglog(na,coeffea,linewidth=0.0,marker='o',color='r',label='Loglog
    plot of "a" coefficients: e(x)')
legend()
loglog(nb,coeffeb,linewidth=0.0,marker='o',color='g',label='Loglog
    plot of "b" coefficients: e(x)')
legend()
grid(True)
```

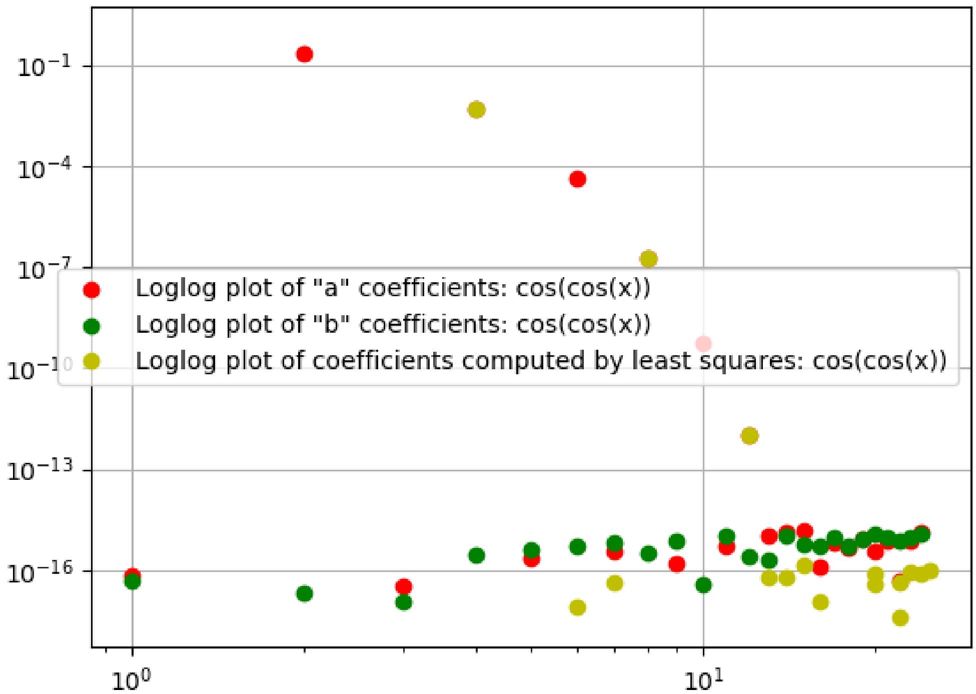
```
figure(5)
semilogy(na,coeffca,linewidth=0.0,marker='o',color='r',label='Semilogy
    plot of "a" coefficients: cos(cos(x))')
legend()
semilogy(nb,coeffcb,linewidth=0.0,marker='o',color='g',label='Semilogy
    plot of "b" coefficients: cos(cos(x))')
legend()
grid(True)

figure(6)
loglog(na,coeffca,linewidth=0.0,marker='o',color='r',label='Loglog
    plot of "a" coefficients: cos(cos(x))')
loglog(nb,coeffcb,linewidth=0.0,marker='o',color='g',label='Loglog
    plot of "b" coefficients: cos(cos(x))')
legend()
grid(True)
```





10^2
10^1
10^0
10^{-1}
10^0
10^1
Loglog plot of "a" coefficients: e(x)
Loglog plot of "b" coefficients: e(x)
Loglog plot of coefficients computed by least squares: e(x)

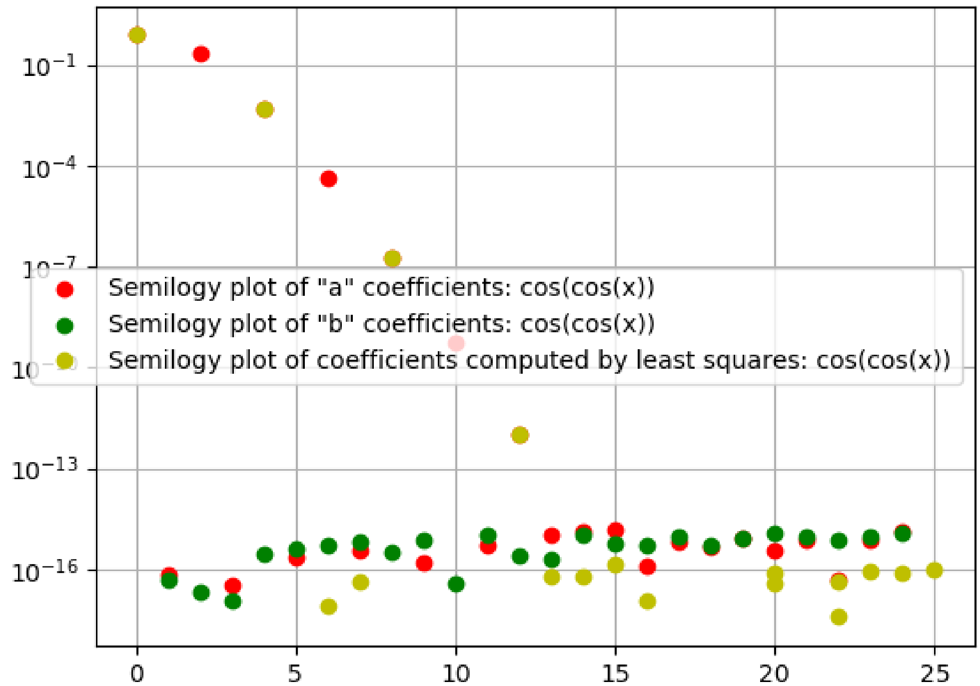

Semilogy plot of "a" coefficients: cos(cos(x))
Semilogy plot of "b" coefficients: cos(cos(x))
Semilogy plot of coefficients computed by least squares: cos(cos(x))
10^{-1}
10^{-4}
10^{-7}
10^{-13}
10^{-16}
0
5
10
15
20
25



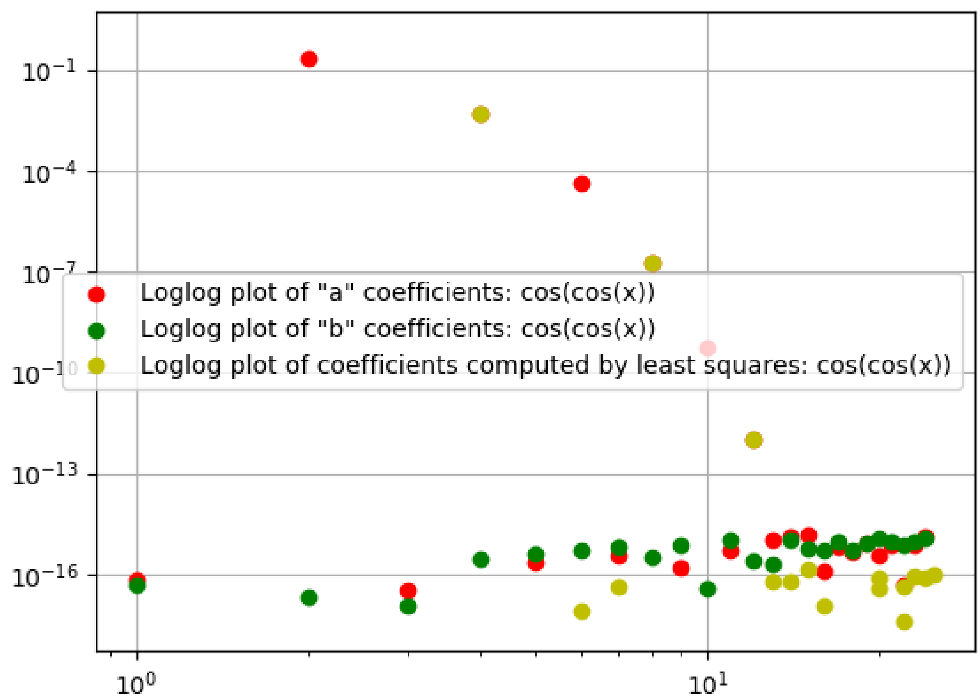
1.3 Least Squares Approach

```
x=linspace(0,2*pi,401)
x=x[:-1]
A=zeros((400,51))
A[:,0]=1
for k in range(1,26):
  A[:,2*k-1]=cos(k*x)
  A[:,2*k]=sin(k*x)

#5
c1=lstsq(A,e(x),rcond=-1)[0]
c2=lstsq(A,cos(cos(x)),rcond=-1)[0]

figure(3)
semilogy(n,c1,linewidth=0.0,marker='o',color='y',label='Semilogy
    plot of coefficients computed by least squares: e(x)')
legend()
```

```
grid(True)

figure(4)
loglog(n,c1,linewidth=0.0,marker='o',color='y',label='Loglog
    plot of coefficients computed by least squares: e(x)')
legend()
grid(True)

figure(5)
semilogy(n,c2,linewidth=0.0,marker='o',color='y',label='Semilogy
    plot of coefficients computed by least squares:
    cos(cos(x))')
legend()
grid(True)

figure(6)
loglog(n,c2,linewidth=0.0,marker='o',color='y',label='Loglog
    plot of coefficients computed by least squares:
    cos(cos(x))')
legend()
grid(True)
diffe=abs(c1-coeffe)
diffc=abs(c2-coeffc)
maxe=np.max(diffe)
maxc=np.max(diffc)

print("Maximum error between the two methods:")
print("e(x)",maxe)
print("cos(cos(x))",maxc)
```



Semilogy plot of "a" coefficients: cos(cos(x))
Semilogy plot of "b" coefficients: cos(cos(x))
Semilogy plot of coefficients computed by least squares: cos(cos(x))
10^{-1}
10^{-4}
10^{-7}
10^{-13}
10^{-16}
0
5
10
15
20
25

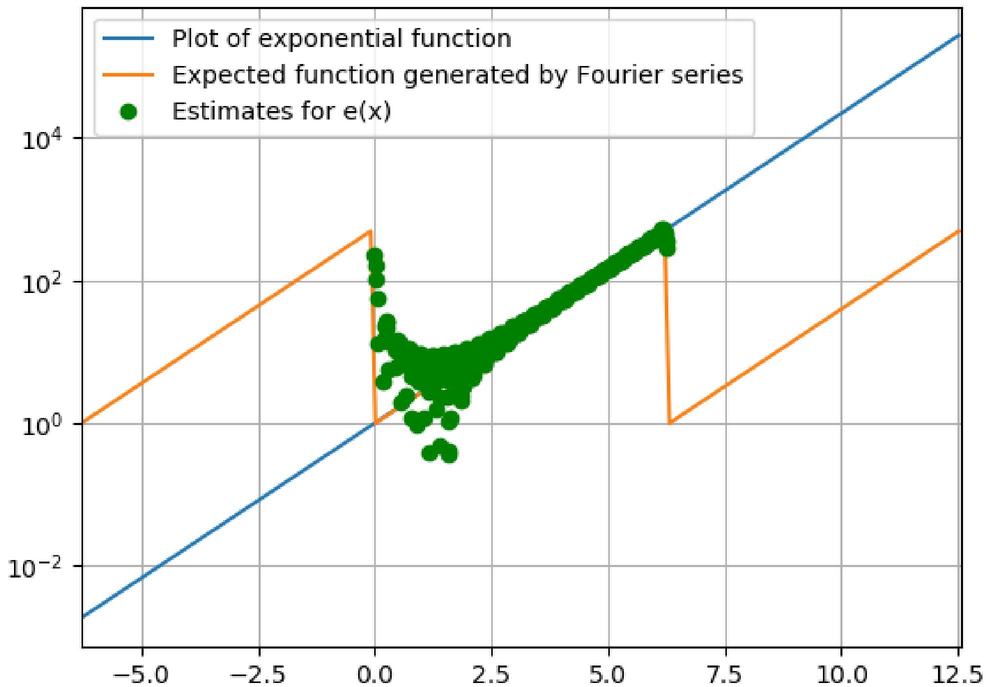


1.4 Comparison of the two methods

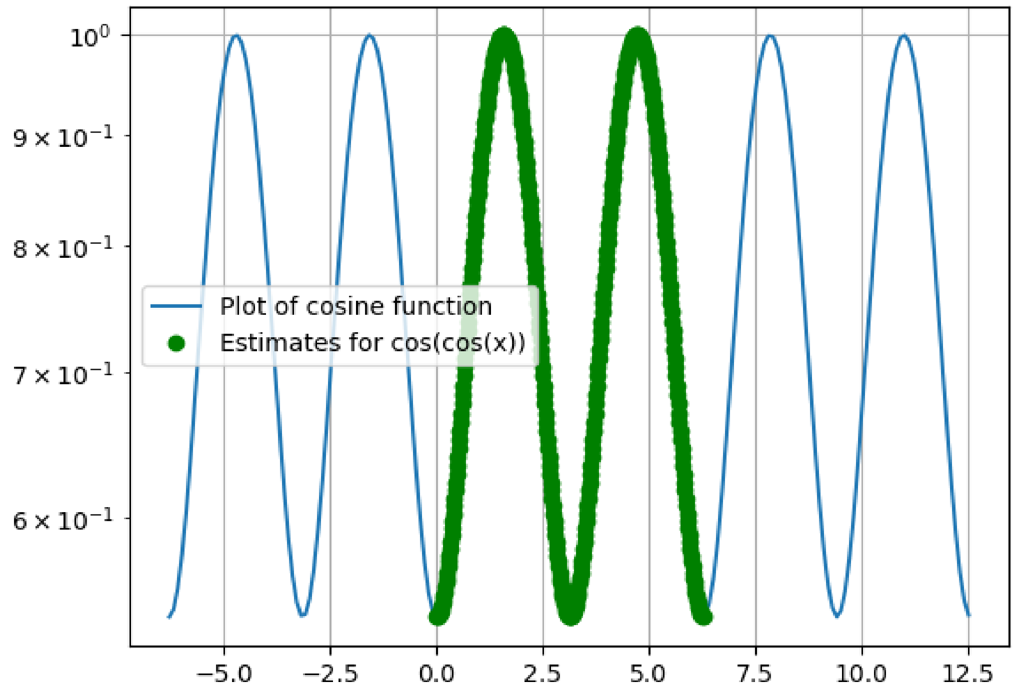
Fourier series computations work best for periodic functions, specifically those that are sinusoidal. e^x, being neither, shows heavy deviation.It is also discontinuous at multiples of $2*\pi$. Moreover, as discussed in a previous question, the coefficients are not zero where they are ideally supposed to be. On the other hand, $cos(cos(x))$ is sinusoidal and periodic. This is why the expected function matches the computed one.

```
figure(1)
semilogy(x,np.dot(A,c1),linewidth=0.0,marker='o',color='g',label='Estimates
    for e(x)')
legend()
grid(True)

figure(2)
semilogy(x,np.dot(A,c2),linewidth=0.0,marker='o',color='g',label='Estimates
    for cos(cos(x))')
legend()
grid(True)
```

```
show()
```



Plot of exponential function
Expected function generated by Fourier series
Estimates for e(x)
10^4
10^2
10^0
10^{-2}
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5



2 Conclusion

The Fourier series coefficients were computed by direct integration and by least squares methods for two different functions.