Fourier Approximations

Mohammed Muqeeth
EE16B026
Department of Electrical Engineering
IIT Madras

February 24, 2018

Abstract

This report explains fourier approximations taking a periodic and a non periodic function as examples.The approach is made by least squares method and direct integration method.

1 Introduction

We will compute fourier series of two functions,exp(x) and cos(cos(x)) over the interval [0,2π) using Eq. (1) and Eq. (2)

$$a_0 + \sum_{n=1}^{\infty} \{a_n \cos(nx) + b_n \sin(nx)\} \tag{1}$$

where the coefficients a_n and $b_n$are given as

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

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

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

for least square approach we take 25 coefficients

$$a_0 + \sum_{n=1}^{25} a_n cos(nx_i) + \sum_{n=1}^{25} b_n sin(nx_i) \approx f(x_i) \tag{2}$$

This turns into matrix problem Ac = b :

$$\begin{pmatrix} 1 & cos(x_1) & sin(x_1) & \dots & cos(25x_1) & sin(25x_1) \\ 1 & cos(x_2) & sin(x_2) & \dots & cos(25x_2) & sin(25x_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & cos(x_{400}) & sin(x_{400}) & \dots & cos(25x_{400}) & sin(25x_{400}) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ \dots \\ a_{25} \\ b_{25} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \dots \\ f(x_{400}) \end{pmatrix} \tag{3}$$

where c are fourier coefficents.

2 Using integral approach

2.1 Define functions $\exp(x)$ and $\cos(\cos(x))$ and plot them:

1. follow below code for defining two functions and plotting them:

```
def f1(x):
    return exp(x)

def f2(x):
    return cos(cos(x))

def f3(x):
    l = []
    for i in x:
        if(i<0):
            i = i + 2*pi
        if(i>2*pi):
            i = i - 2*pi
        l.append(i)
    return exp(l)

def f4(x):
    return cos(cos(x))

x=linspace(0,2*pi,401)
x=x[:-1] # drop last term to have a proper periodic integral
x1 = arange(-2*pi,4*pi,0.1)
y1 = f1(x1)
y2 = f2(x1)
y3 = f3(x1)
y4 = f4(x1)
plt.semilogy(x1,y1,'r')
plt.semilogy(x1,y3,'k')
plt.semilogy(x,fn_values,'go')
plt.grid(color='k',linestyle ='-',linewidth = 1)
plt.xlabel('x')
plt.ylabel('$f(x) = e^x$')
plt.legend(['semilog plot of f(x) vs x','expected plot from fourier','
   plot from leastsquares'],loc='best')
plt.show()

plt.plot(x1,y2,'r')
plt.plot(x1,y4,'k')
plt.plot(x,gn_values,'go')
plt.grid(color='k',linestyle ='-',linewidth = 1)
plt.xlabel('x')
plt.ylabel('$f(x) = cos(cos(x))$')
plt.legend(['linear plot of f(x) vs x','expected plot from fourier','
   plot from leastsquares'],loc='best')
plt.show()
```

2. The corresponding plots are in Figure 1 and 2 respectively.

2.2 Obtain first 51 coefficients for $\exp(x)$ and $\cos(\cos(x))$:

1. create two new functions to be integrated, namely u(x, k) = f (x) cos (kx) and v(x, k) = f (x) sin (kx)

2. To integrate these, use the option in quad to pass extra arguments to the function being integrated.Append index zero of return values into a list

$$coeff = quad(u, 0, 2 * pi, args = (k))[0]$$

3. Follow below code to get coefficients:

```
from pylab import *
import matplotlib.pyplot as plt
from numpy import *
from scipy.integrate import quad
def f(x):
    return exp(x)

def g(x):
    return cos(cos(x))

def u(x,k):
    return f(x)*cos(k*x)

def v(x,k):
    return f(x)*sin(k*x)

a = []
b = []
#get fourier coefficients of exp(x) into a and b lists
a.append((quad(u,0,2*pi,args=(0)))[0]/(2*pi))
for k in range(1,26):
    a.append(quad(u,0,2*pi,args=(k))[0]/pi)
    b.append(quad(v,0,2*pi,args=(k))[0]/pi)

def w(x,k):
    return g(x)*cos(k*x)

def z(x,k):
    return g(x)*sin(k*x)

c = []
d = []
#get fourier coefficients of cos(cos(x)) into  c and d lists
c.append((quad(w,0,2*pi,args=(0)))[0]/(2*pi))
for k in range(1,26):
    c.append(quad(w,0,2*pi,args=(k))[0]/pi)
    d.append(quad(z,0,2*pi,args=(k))[0]/pi)
```

4. Take above coefficients in this vector form

$$\begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ \dots \\ a_{25} \\ b_{25} \end{pmatrix}$$

5. To take coefficients in given vector form

```
#To take coefficients in given vector form of integral approach
c1_intg = [0]*(len(a)+len(b))
c1_intg[0] = a[0]
c1_intg[1::2] = a[1:]
c1_intg[2::2] = b# Now c1_intg has exp(x) fourier coefficients

c2_intg = [0]*(len(c)+len(d))
c2_intg[0] = c[0]
c2_intg[1::2] = c[1:]
c2_intg[2::2] = d#Now c2_intg has cos(cos(x)) fourier
```

2.3 Plot magnitude of coefficients:

1. Follow below code for plotting magnitude of coefficients:

```
x1 = arange(0,51,1)#we take 26 a_n and 25 b_n coefficients
#semilog plot of coefficients of exp(x)
plt.title('semilog plot of coefficients of exp(x)')
plt.semilogy(x1,absolute(c1_intg),'ro')
plt.semilogy(x1,absolute(c1_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best
    ')
plt.show()
#loglog plot of coefficients of exp(x)
plt.title('loglog plot of coefficients of exp(x)')
plt.loglog(x1,absolute(c1_intg),'ro')
plt.loglog(x1,absolute(c1_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best
    ')
plt.show()
#semilog plot of coefficients of cos(cos(x))
plt.title('semilog plot of coefficients of cos(cos(x))')
plt.semilogy(x1,absolute(c2_intg),'ro')
plt.semilogy(x1,absolute(c2_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best
    ')
```

```
plt.show()
#loglog plot of coefficients of exp(x)
plt.title('loglog plot of coefficients of cos(cos(x))')
plt.loglog(x1,absolute(c2_intg),'ro')
plt.loglog(x1,absolute(c2_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best
    ')
plt.show()
```

2. Magnitude of coefficients for $f(x) = \exp(x)$ are in Figures 3 and 4, and the coefficients for $f(x) = \cos(\cos x)$ are in Figures 5 and 6

2.4 Results and Discussion

1. From Figure 1 clearly $\exp(x)$ is not periodic.The expected plot in Figure 1 does not match with $\exp(x)$ semilog plot because it is not periodic and discontinuity.

2. From Figure 2 $\cos(\cos(x))$ is periodic.The expected plot in Figure 2 matches exactly with $\cos(\cos(x))$.

Figure 1: exp(x) vs x

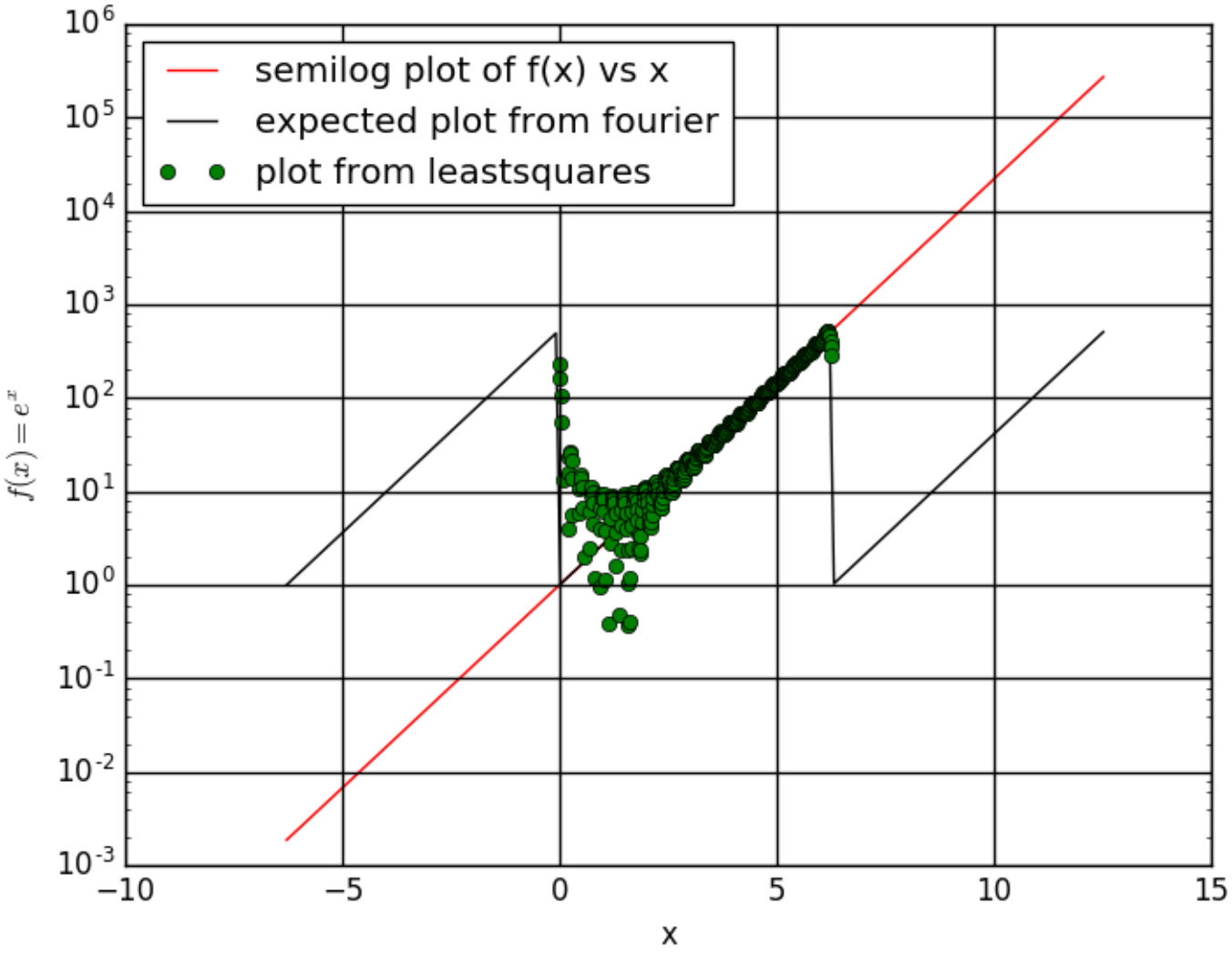
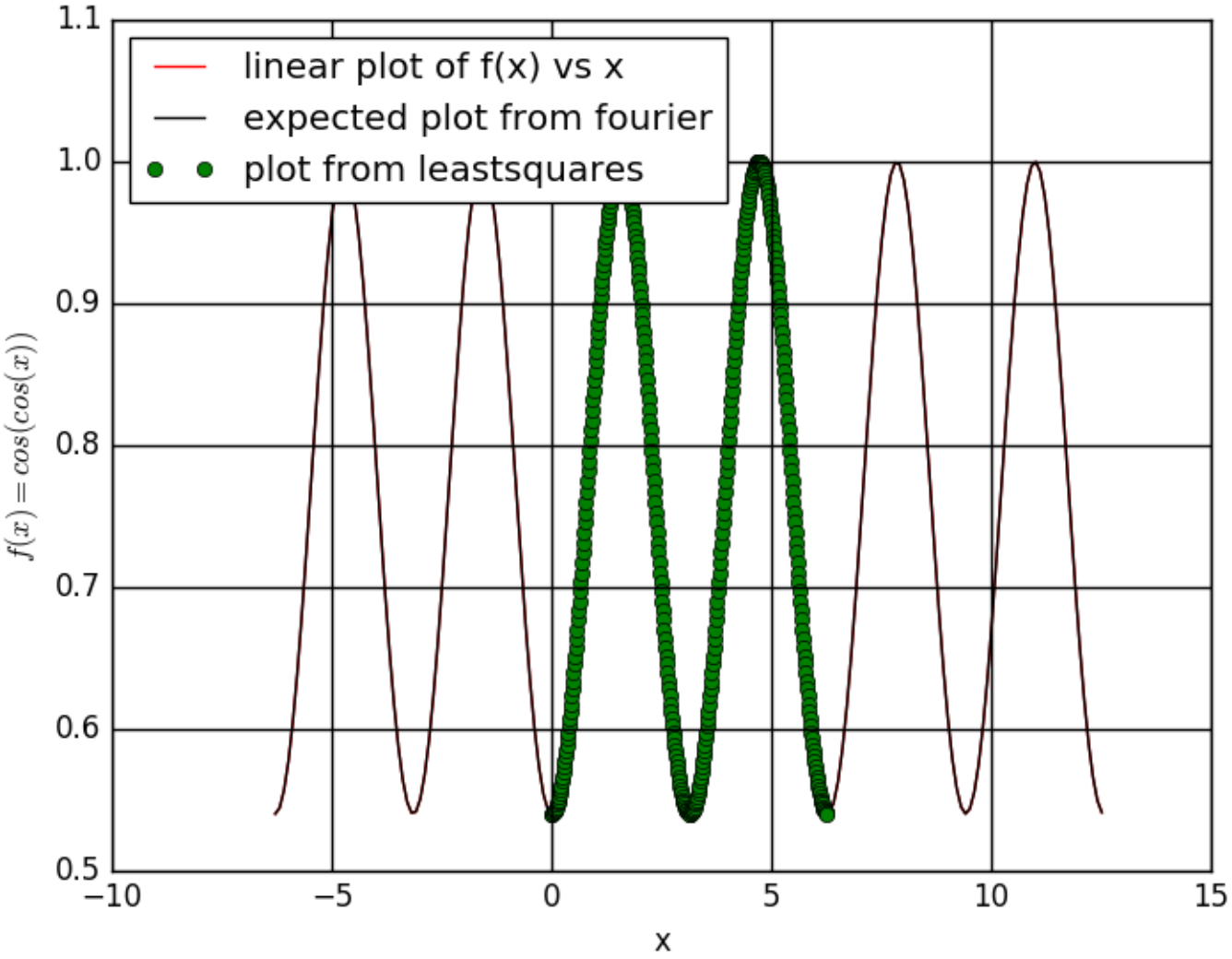


Figure 2: cos(cos(x)) vs x



3 Using leastsquares approach:

3.1 Solve Eq . 3 to get fourier coefficients

1. Define a vector x going from 0 to 2π in 400 steps using linspace.
2. Solve Ac = b using $c = lstsq(A, b)[0]$.It retuns a list of coefficients in the from mentioned in section 2.2.
3. The code for it is below:

```
#least square approach
x=linspace(0,2*pi,401)
x=x[:-1] # drop last term to have a proper periodic integral
b=f(x)
c=g(x)
# f has been written to take a vector
A=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]=cos(k*x) # cos(kx) column
    A[:,2*k]=sin(k*x)
# sin(kx) column
#endfor
```

4. The plot of magnitudes of coefficients of $f(x) = \exp(x)$ are in Figures 3 and 4, and the coefficients for$f(x) = \cos(\cos x)$ are in Figures 5 and 6 respectively.

3.2 Compare the answers got by least squares and by the direct integration. Calculate deviation

1. Now we have coefficients obtained by both approaches. c1_intg,c2_intg under section2.2, c1_lsa,c2_lsa under section 3.1 have coefficients $a_n b_n$ of $\exp(x)$ and $\cos(\cos(x))$ respectively.

2. The deviation between coefficients is calculated as follows:

```
c1_lsa=lstsq(A,b)[0]#coefficients of exp(x) using leastsquares
   approach
c1_intg = asarray(c1_intg)
deviation = max(abs(c1_lsa-c1_intg))#get max deviation
print deviation

c2_lsa=lstsq(A,c)[0]#coefficients using leastsquares approach
c2_intg = asarray(c2_intg)
deviation = max(abs(c2_lsa-c2_intg))#get max deviation
print deviation

#to print coefficients of exp(x) and cos(cos(x)) with both approaches
print c1_intg
print c2_intg
print c1_lsa
print c2_lsa
```

3. From estimated values of coefficients of exp(x) and cos(cos(x)) in c1_lsa and c2_lsa respectively. we can get function values by computing A.c1_lsa and A.c2_lsa

```
#get values of functions by performing matrix multiplication on A and
   coefficients from least squares
fn_values = dot(A,c1_lsa)
gn_values = dot(A,c2_lsa)
```

4 Results and Discussion:

1. From Figure 5 and Figure 6 , the $b_n$coefficients of $\cos(\cos(x))$ is nearly zero since $\cos(\cos(x))$ is even function and $\sin(nx)$ is odd function.The integral over 0 to 2π of $\cos(\cos(x))\sin(nx)$ is zero.

2. Since $\cos(\cos(x))$ is continuous it can be constructed using lesser frequency components i.e k ranging from 0 to 25.But $\exp(x)$ is discontinuous at $2m\pi$ so to accomodate discontinuity high frequency components are required i.e k values should range more than 25.

3. The Riemann-Lebesgue lemma says that the Fourier coefficients converge to zero as |k| $\to \infty$.The more smoother the function is ,the fast the fourier coefficients converges to zero.Since cos(cos(x)) is infinitely differentiable(hence smoother) its coefficients decay faster when compared to exp(x) which is discontinuous at multiples of 2π(hence not smoother). These rate of decay can be seen in Figures 3,4,5,6

4. The integral which computes a_n,$b_n$of exp(x) will be of the form $\frac{const}{1+n^2}$ which on loglog plot looks linear.Hence the Figure 4 is linear.

$$\int_0^{2\pi} e^x \cos(nx)dx = \frac{e^{2\pi}-1}{1+n^2}$$

5. $\int_0^{2\pi} cos(cos(x))cos(kx)dx$ takes the form of r^k which on semilog log plot looks linear and as $k \to largevalue$ integral goes to zero from Riemann-Lebesgue lemma.Hence Figure 5 is linear.

6. The deviation of coefficients obtained from integral and leastsquares are 1.33273087034 and $2.63744981915e-15$ for exp(x) and cos(cos(x)) respectively.The values do not agree because integral approach assumes infinite coefficients to construct the functions whereas leastsquares tries to fit it to curves with 25 coefficients only(as we have taken).

7. For cos(cos(x)) since 25 coefficients make the function closer enough, the least squares fits properly to expected one. whereas for exp(x) because of its discontinuity it requires higher frequency coefficients, therefore plot from 25 coefficients approach closer to function except at discontinuity where disagreement is clearly seen from Figure 1.

Figure 3: semilog plot of coefficients of exp(x)

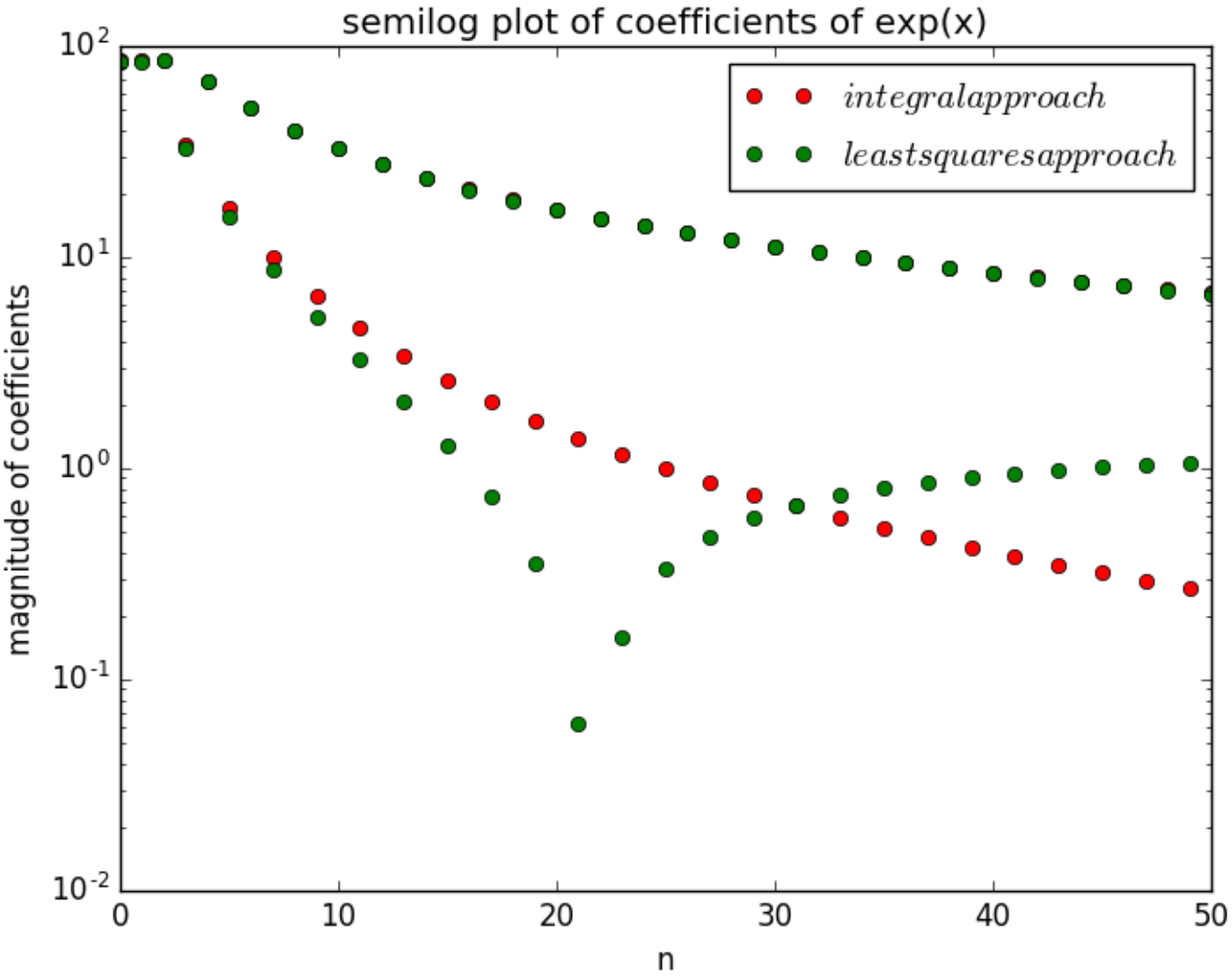


Figure 4: loglog plot of coefficients of exp(x)

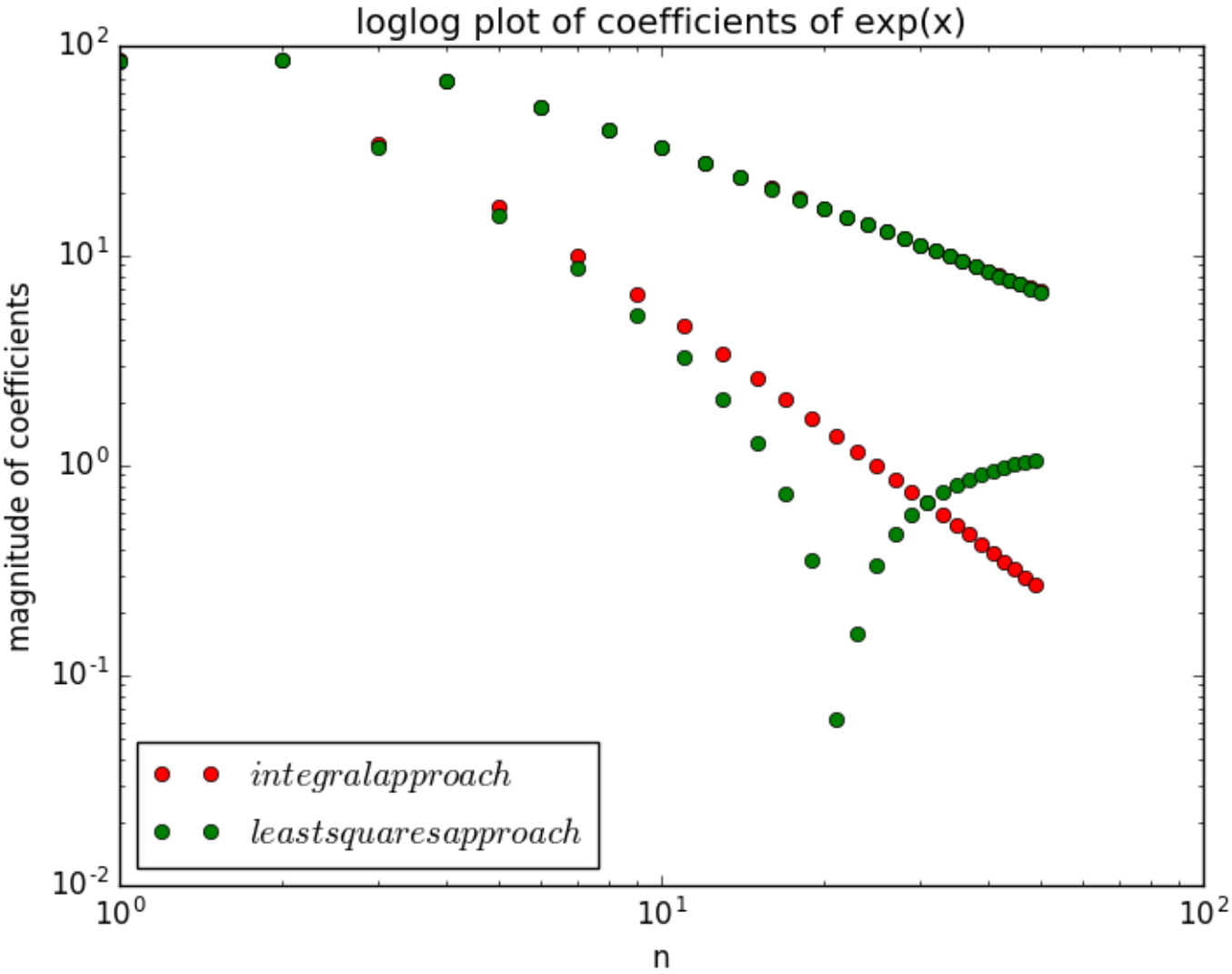


Figure 5: semilog plot of coefficients of cos(cos(x))

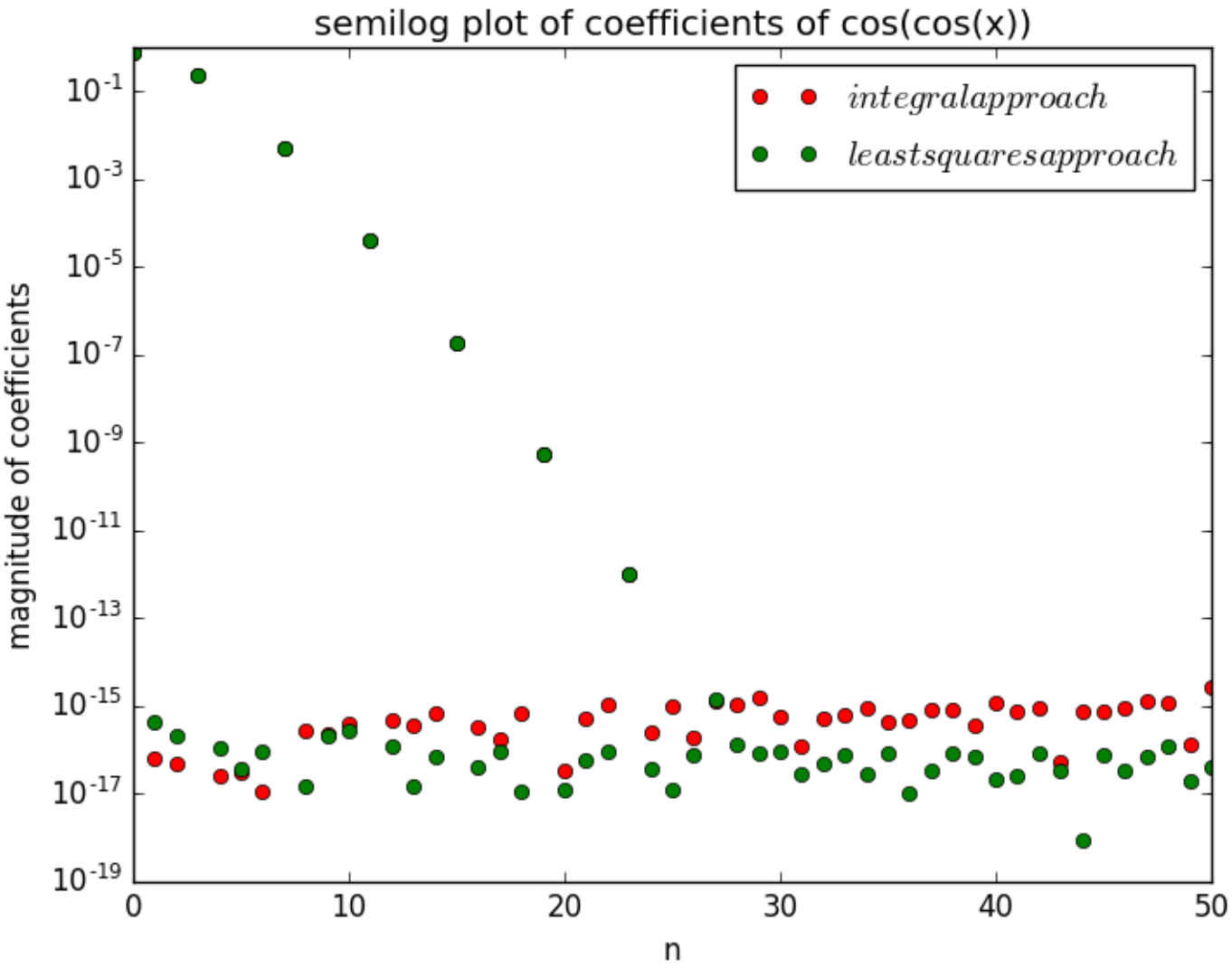
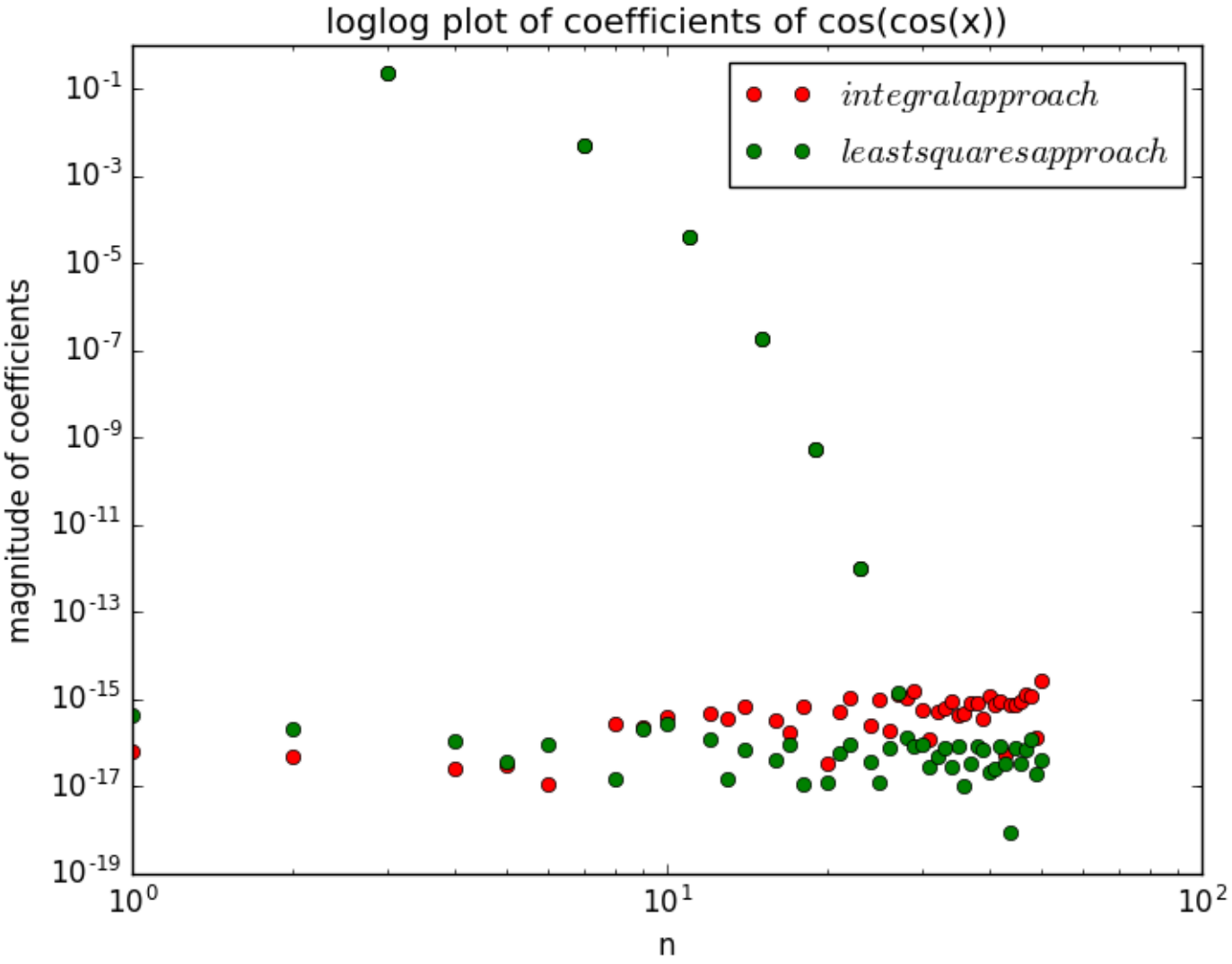


Figure 6: loglog plot of coefficients of cos(cos(x))



5 Python code:

```
from pylab import *
import matplotlib.pyplot as plt
from numpy import *
from scipy.integrate import quad
def f(x):
    return exp(x)
def g(x):
    return cos(cos(x))
def u(x,k):
    return f(x)*cos(k*x)
def v(x,k):
    return f(x)*sin(k*x)
a = []
b = []
#get fourier coefficients of exp(x) into a and b lists
a.append((quad(u,0,2*pi,args=(0)))[0]/(2*pi))
for k in range(1,26):
    a.append(quad(u,0,2*pi,args=(k))[0]/pi)
```

```
    b.append(quad(v,0,2*pi,args=(k))[0]/pi)
def w(x,k):
    return g(x)*cos(k*x)
def z(x,k):
    return g(x)*sin(k*x)
c = []
d = []
#get fourier coefficients of cos(cos(x)) into  c and d lists
c.append((quad(w,0,2*pi,args=(0)))[0]/(2*pi))
for k in range(1,26):
    c.append(quad(w,0,2*pi,args=(k))[0]/pi)
    d.append(quad(z,0,2*pi,args=(k))[0]/pi)
#To take coefficients in given vector form of integral approach
c1_intg = [0]*(len(a)+len(b))
c1_intg[0] = a[0]
c1_intg[1::2] = a[1:]
c1_intg[2::2] = b# Now c1_intg has exp(x) fourier coefficients
c2_intg = [0]*(len(c)+len(d))
c2_intg[0] = c[0]
c2_intg[1::2] = c[1:]
c2_intg[2::2] = d#Now c2_intg has cos(cos(x)) fourier
#least square approach
x=linspace(0,2*pi,401)
x=x[:-1] # drop last term to have a proper periodic integral
b=f(x)
c=g(x)
# f has been written to take a vector
A=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]=cos(k*x) # cos(kx) column
    A[:,2*k]=sin(k*x)
# sin(kx) column
#endfor
c1_lsa=lstsq(A,b)[0]#coefficients of exp(x) using leastsquares approach
c1_intg = asarray(c1_intg)
deviation = max(abs(c1_lsa-c1_intg))#get max deviation
print deviation
c2_lsa=lstsq(A,c)[0]#coefficients using leastsquares approach
c2_intg = asarray(c2_intg)
deviation = max(abs(c2_lsa-c2_intg))#get max deviation
print deviation
#to print coefficients of exp(x) and cos(cos(x)) with both approaches
print c1_intg
print c2_intg
print c1_lsa
print c2_lsa
x1 = arange(0,51,1)#we take 26 a_n and 25 b_n coefficients
#semilog plot of coefficients of exp(x)
plt.title('semilog plot of coefficients of exp(x)')
plt.semilogy(x1,absolute(c1_intg),'ro')
```

```
plt.semilogy(x1,absolute(c1_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best')
plt.show()
#loglog plot of coefficients of exp(x)
plt.title('loglog plot of coefficients of exp(x)')
plt.loglog(x1,absolute(c1_intg),'ro')
plt.loglog(x1,absolute(c1_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best')
plt.show()
#semilog plot of coefficients of cos(cos(x))
plt.title('semilog plot of coefficients of cos(cos(x))')
plt.semilogy(x1,absolute(c2_intg),'ro')
plt.semilogy(x1,absolute(c2_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best')
plt.show()
#loglog plot of coefficients of exp(x)
plt.title('loglog plot of coefficients of cos(cos(x))')
plt.loglog(x1,absolute(c2_intg),'ro')
plt.loglog(x1,absolute(c2_lsa),'go')
plt.xlabel('n')
plt.ylabel('magnitude of coefficients')
plt.legend(['$integral approach$','$leastsquares approach$'],loc='best')
plt.show()
#get values of functions by performing matrix multiplication on A and coefficients fro
fn_values = dot(A,c1_lsa)
gn_values = dot(A,c2_lsa)
def f1(x):
    return exp(x)
def f2(x):
    return cos(cos(x))
def f3(x):
    l = []
    for i in x:
        if(i<0):
            i = i + 2*pi
        if(i>2*pi):
            i = i - 2*pi
        l.append(i)
    return exp(l)
def f4(x):
    return cos(cos(x))
x=linspace(0,2*pi,401)
x=x[:-1] # drop last term to have a proper periodic integral
x1 = arange(-2*pi,4*pi,0.1)
y1 = f1(x1)
y2 = f2(x1)
y3 = f3(x1)
```

```
y4 = f4(x1)
plt.semilogy(x1,y1,'r')
plt.semilogy(x1,y3,'k')
plt.semilogy(x,fn_values,'go')
plt.grid(color='k',linestyle ='-',linewidth = 1)
plt.xlabel('x')
plt.ylabel('$f(x) = e^x$')
plt.legend(['semilog plot of f(x) vs x','expected plot from fourier','plot from leastsq
plt.show()
plt.plot(x1,y2,'r')
plt.plot(x1,y4,'k')
plt.plot(x,gn_values,'go')
plt.grid(color='k',linestyle ='-',linewidth = 1)
plt.xlabel('x')
plt.ylabel('$f(x) = cos(cos(x))$')
plt.legend(['linear plot of f(x) vs x','expected plot from fourier','plot from leastsqu
plt.show()
```