Task2

Ronak Mehta (MHTRON001)

These tasks involve writing code, or modifying existing code, to meet the objectives described.

1. Find and plot the Fourier series frequency-domain representation for the signal $x_1(t)$ below over the range $k = -8, \ldots, 8$:

   Do this using both symbolic processing and by evaluating the integral for the coefficients by hand. Compare the result with that displayed earlier for $x(t)$. Note that the signals are related in time by $x_1(t) = x(t-2)$.

   You should observe that shifting a signal only changes the phase in the frequency domain, while the magnitude remains unchanged.

2. Use symbolic processing to find and plot the frequency-domain representation of $x_2(t)$ below over the range $k = -8, \ldots, 8$:

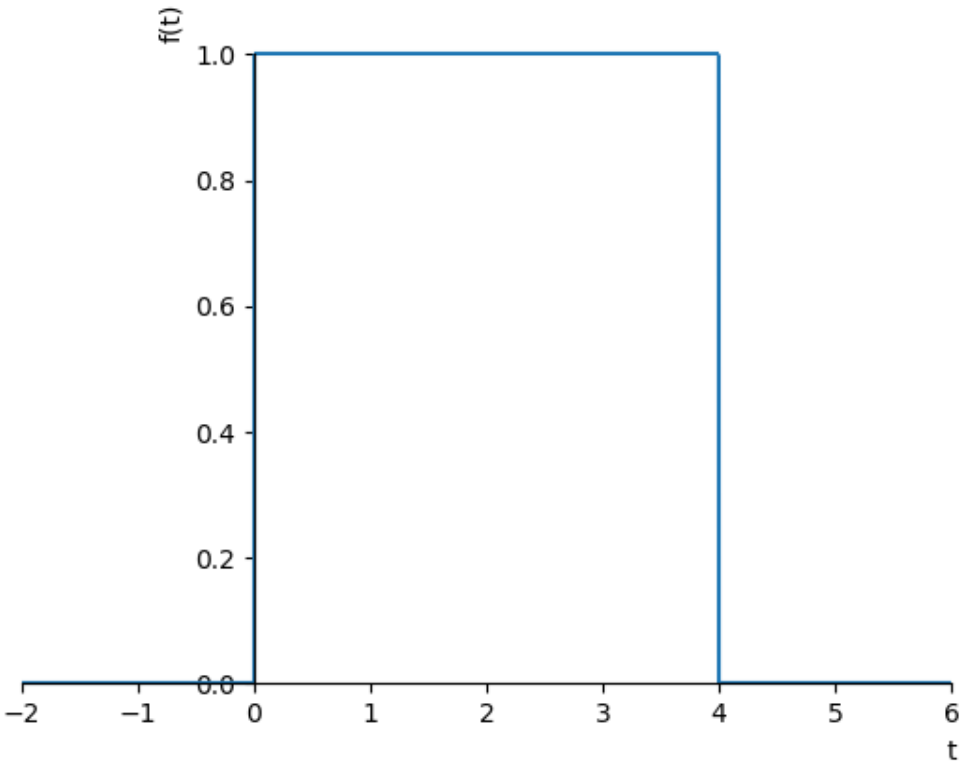
   Also plot the reconstruction over the range $t = -4$ to $t = 4$ using only components up to and including the 5th harmonic.

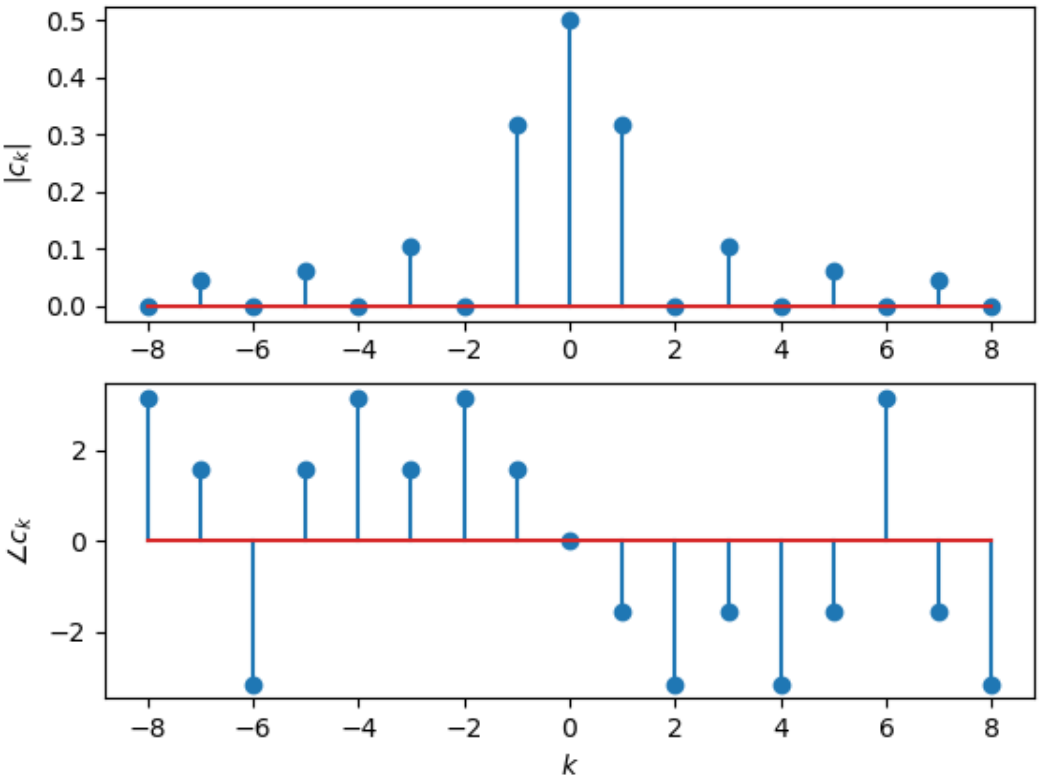
   You should find that as k increases the magnitude of the coefficients in this case falls off much faster than those of $x(t)$. This is because $x_2(t)$ is smoother (it is at least continuous, while $x(t)$ is discontinuous). The reconstruction is therefore also more accurate with a smaller number of terms.

3. Find and plot the frequency-domain representation of $x_3(t)$ below over the range $k = -8, \ldots, 8$.

In [8]:
```
#Question1:

import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook
import sympy as sp

def fsrrec(ckv,omega0,tv):

    xv = ckv[0]*np.ones(tv.shape);
    #tv.shape returns the dimensions of the matrix (or, in this case, array) t
v.
    #np.ones creates an array of the input size populated with 1's
    for k in range(1,len(ckv)):
        kh = 2*np.abs(ckv[k])*np.cos(k*omega0*tv + np.angle(ckv[k]));  #create
 kth harmonic
        xv = xv + kh; #add kth harmonic to x
    return(np.real(xv));

def fsrrec_plots(ckv,omega0,tv):

    xv = ckv[0]*np.ones(tv.shape);
    plt.figure(1)
    plt.plot(tv,xv)
    for k in range(1,len(ckv)):
        kh = 2*np.abs(ckv[k])*np.cos(k*omega0*tv + np.angle(ckv[k]));  #create
 kth harmonic
        plt.plot(tv,kh);

        xv = xv + kh; #add kth harmonic to x
    plt.show()
    return(np.real(xv));

t = sp.symbols('t')
x = sp.Piecewise( (0, t<=0), (1, t<4), (0, True));
sp.plot(x.subs(t,sp.re(t)), (t,-2,6));

Ts, k, w0 = sp.symbols('Ts k w0');
w0 = 2*sp.pi/Ts;
expt = sp.exp(-1j*k*w0*t);
cke = 1/Ts*sp.integrate(x*expt, (t, -Ts/2, Ts/2));
ck = cke.subs(Ts,8).doit();  # set value for period and evaluate
ck

kv = np.arange(-8,9);  # coefficients to calculate
#by default, np.arange returns the integers between the given start and end po
ints

ckvs = np.zeros(kv.shape, dtype=np.complex64);  # corresponding coefficient va
lues
for i in range(len(kv)):
    ki = kv[i];
    ckvs[i] = ck.subs({k:ki}).evalf();
ckvs

fh, ax = plt.subplots(2);
```

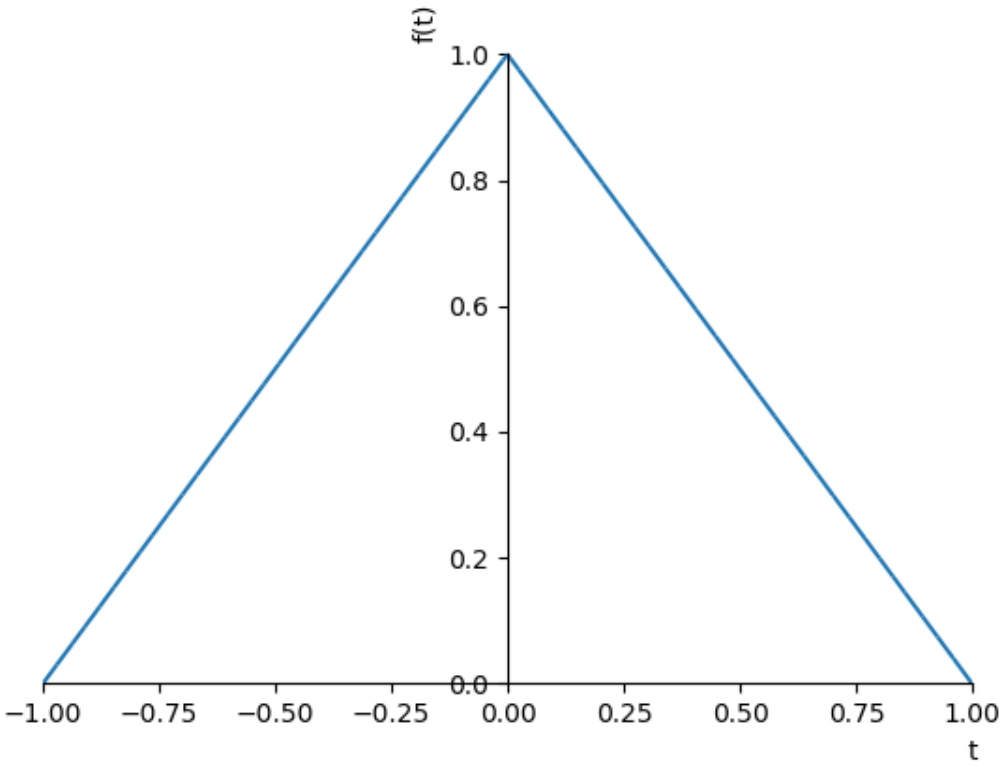
```
ax[0].stem(kv, np.abs(ckvs), c='g');  ax[0].set_ylabel(r'$|c_k|$');
ax[1].stem(kv, np.angle(ckvs), c='g');  ax[1].set_ylabel(r'$\angle c_k$');
plt.xlabel('$k$');
```

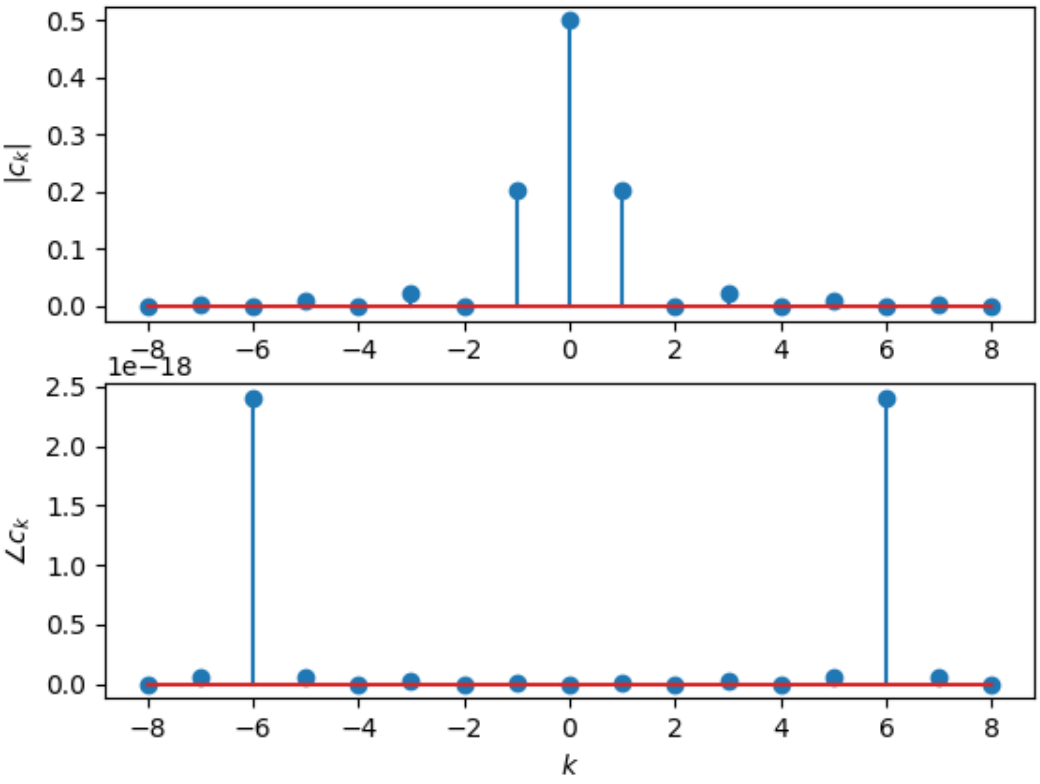

f(t)
1.0
0.8
0.6
0.4
0.2
0.0
−2
−1
0
1
2
3
4
5
6
t


$|c_k|$
0.5
0.4
0.3
0.2
0.1
0.0
−8
−6
−4
−2
0
2
4
6
8
$\angle c_k$
2
0
−2
−8
−6
−4
−2
0
2
4
6
8
k

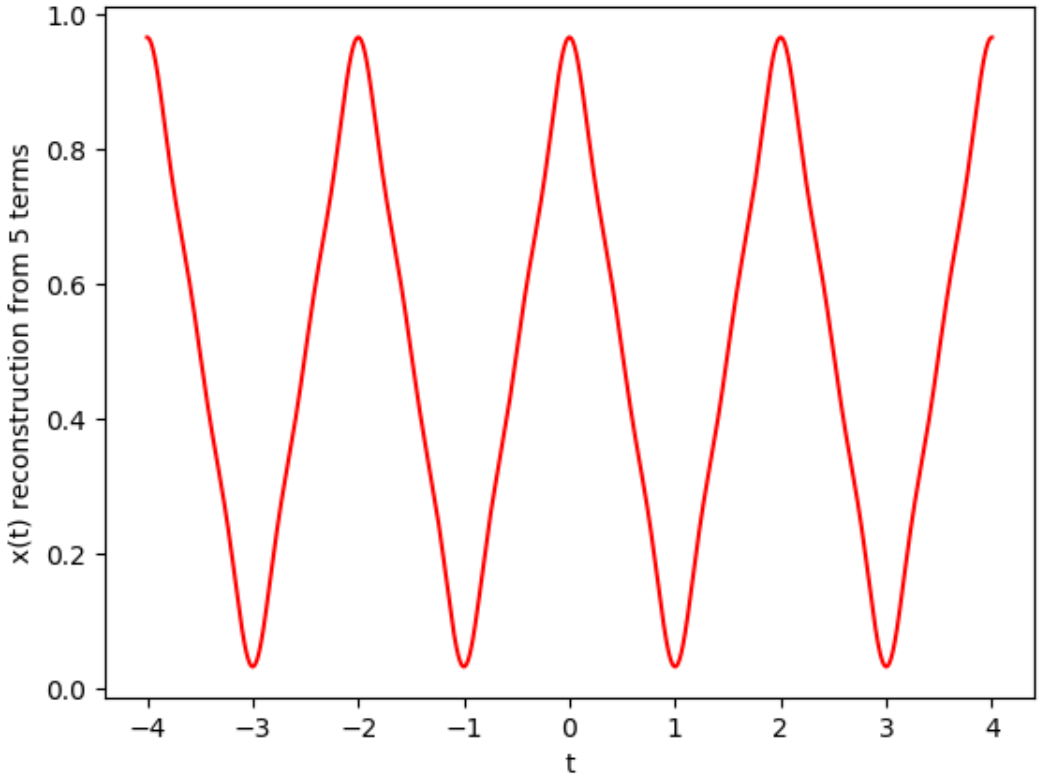
```
C:\Users\acer\Anaconda3\Anaconda New\lib\site-packages\matplotlib\cbook\depre
cation.py:107: MatplotlibDeprecationWarning: stem() got an unexpected keyword
argument 'c'. This will raise a TypeError in future versions.
  warnings.warn(message, mplDeprecation, stacklevel=1)
```

In [9]:
```
#Question2:

import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook
import sympy as sp

T=2
t = sp.symbols('t')
x = sp.Piecewise( (t+1, t<=0), (1-t, t<1), (0,True));
sp.plot(x.subs(t,sp.re(t)), (t,-1,1));

Ts, k, w0 = sp.symbols('Ts k w0');
w0 = 2*sp.pi/Ts;
expt = sp.exp(-1j*k*w0*t);
cke = 1/Ts*sp.integrate(x*expt, (t, -Ts/2, Ts/2));
ck = cke.subs(Ts,2).doit();  # set value for period and evaluate
ck

kv = np.arange(-8,9);  # coefficients to calculate
#by default, np.arange returns the integers between the given start and end po
ints

ckvs = np.zeros(kv.shape, dtype=np.complex64);  # corresponding coefficient va
lues
for i in range(len(kv)):
    ki = kv[i];
    ckvs[i] = ck.subs({k:ki}).evalf();
ckvs

fh, ax = plt.subplots(2);
ax[0].stem(kv, np.abs(ckvs), c='g');  ax[0].set_ylabel(r'$|c_k|$');
ax[1].stem(kv, np.angle(ckvs), c='g');  ax[1].set_ylabel(r'$\angle c_k$');
plt.xlabel('$k$');

#Recontruction=================================================================
=================
kv = np.arange(-5,6);
ckvs = np.zeros(kv.shape, dtype=np.complex64);  # corresponding coefficient va
lues
for i in range(len(kv)):
    ki = kv[i];
    ckvs[i] = ck.subs({k:ki}).evalf();
ckvs
kzi = np.where(kv==0)[0][0];  # index for zero element
ckvsp = ckvs[kzi:];
tv = np.linspace(-4,4,10000);
xv = fsrrec(ckvsp,2*np.pi/T,tv);
fh = plt.figure();
plt.plot(tv,xv,'r');
plt.xlabel('t');  plt.ylabel('x(t) reconstruction from ' + str(len(ckvsp)-1) +
 ' terms');
```

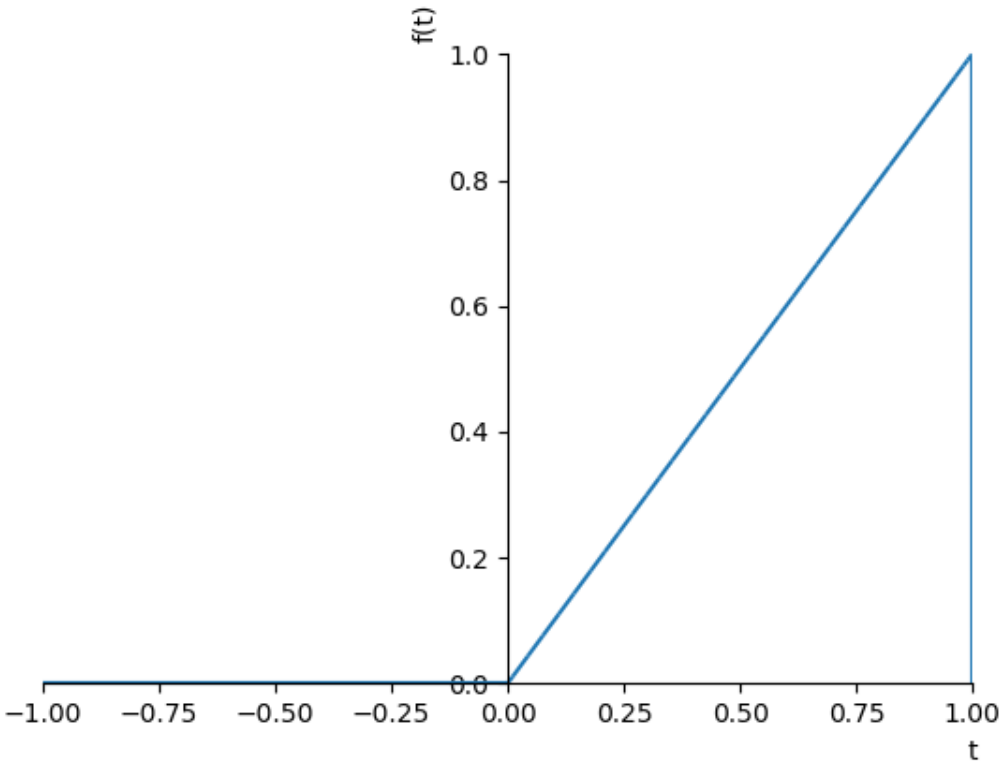

f(t)
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t

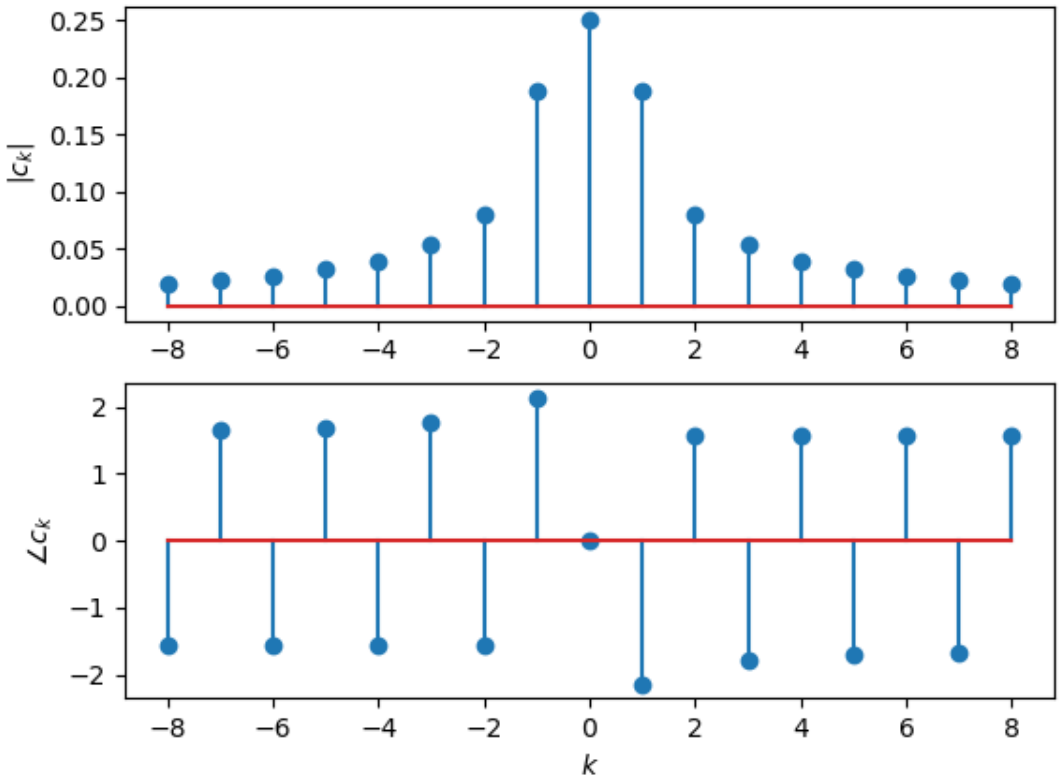

$|c_k|$
0.5
0.4
0.3
0.2
0.1
0.0
−8
−6
−4
−2
0
2
4
6
8
1e−18
$\angle c_k$
2.5
2.0
1.5
1.0
0.5
0.0
k

```
C:\Users\acer\Anaconda3\Anaconda New\lib\site-packages\matplotlib\cbook\depre
cation.py:107: MatplotlibDeprecationWarning: stem() got an unexpected keyword
argument 'c'. This will raise a TypeError in future versions.
  warnings.warn(message, mplDeprecation, stacklevel=1)
```



In [10]:
```
#Question3:

import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook
import sympy as sp

T=2
t = sp.symbols('t')
x = sp.Piecewise( (0, t<=0), (t, t<1), (0,True));
sp.plot(x.subs(t,sp.re(t)), (t,-1,1));

Ts, k, w0 = sp.symbols('Ts k w0');
w0 = 2*sp.pi/Ts;
expt = sp.exp(-1j*k*w0*t);
cke = 1/Ts*sp.integrate(x*expt, (t, -Ts/2, Ts/2));
ck = cke.subs(Ts,2).doit();  # set value for period and evaluate
ck

kv = np.arange(-8,9);  # coefficients to calculate
#by default, np.arange returns the integers between the given start and end po
ints

ckvs = np.zeros(kv.shape, dtype=np.complex64);  # corresponding coefficient va
lues
for i in range(len(kv)):
    ki = kv[i];
    ckvs[i] = ck.subs({k:ki}).evalf();
ckvs

fh, ax = plt.subplots(2);
ax[0].stem(kv, np.abs(ckvs), c='g');  ax[0].set_ylabel(r'$|c_k|$');
ax[1].stem(kv, np.angle(ckvs), c='g');  ax[1].set_ylabel(r'$\angle c_k$');
plt.xlabel('$k$');

#kzi = np.where(kv==0)[0][0];  # index for zero element
#ckvsp = ckvs[kzi:];
#tv = np.linspace(-4,4,10000);
#xv = fsrrec(ckvsp,2*np.pi/T,tv);
#fh = plt.figure();
#plt.plot(tv,xv,'r');
#plt.xlabel('t');  plt.ylabel('x(t) reconstruction from ' + str(len(ckvsp)-1)
 + ' terms');
```


f(t)
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t


$|c_k|$
0.25
0.20
0.15
0.10
0.05
0.00
−8
−6
−4
−2
0
2
4
6
8
$\angle c_k$
2
1
0
−1
−2
−8
−6
−4
−2
0
2
4
6
8
k

```
C:\Users\acer\Anaconda3\Anaconda New\lib\site-packages\matplotlib\cbook\depre
cation.py:107: MatplotlibDeprecationWarning: stem() got an unexpected keyword
argument 'c'. This will raise a TypeError in future versions.
  warnings.warn(message, mplDeprecation, stacklevel=1)
```