Algorytm Cooleya-Tukeya

December 21, 2020

Wydział Informatyki Politechniki Białostockiej Przetwarzanie Sygnałów i Obrazów – Pracownia specjalistyczna

Ćwiczenie Nr.3 Dyskretne przekształcenie Fouriera

Imię i nazwisko studenta: Sylwia Mościcka.

Grupa: PS 5

Data realizacji ćwiczenia 21.12.2020r.

Zadanie 3.5

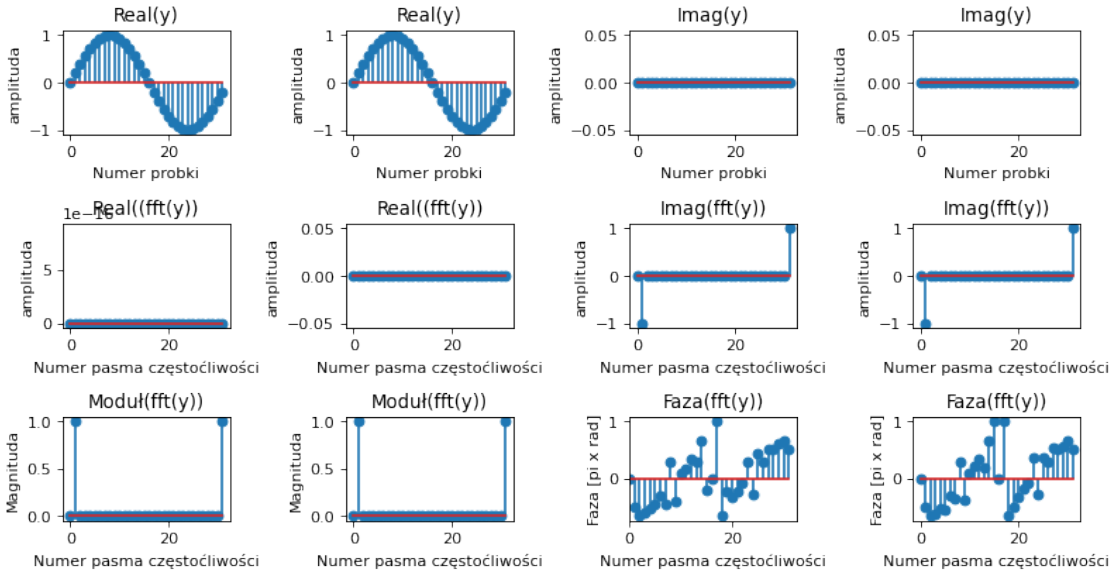
Napisz własną funkcję realizującą dyskretną transformatę Fouriera – DFT oraz jej transformatę odwrotną - IDFT. Wykorzystaj postać macierzową przekształcenia oraz funkcję generującą macierz typu DFT (ang. DFT matrix), np. from scipy.linalg import dft. Porównaj wyniki z zadania 3.1 i 3.2 z wynikami własnej funkcji transformaty Fouriera.

Algorytm Cooleya-Tukeya FFT - funkcje programowe oraz własne przedstawiające zadanie 3.1. oraz 3.2.

[13]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
from cmath import exp,pi

#Algorytm Cooleya-Tukeya FFT
def DFT(x):

    N = len(x)
    if N == 2:
        return np.array([x[0] + x[1], x[0] - x[1]], dtype=np.complex)
    Xp = DFT(x[::2])
    Xn = DFT(x[1::2])
    w = np.exp(-2 * np.pi * 1j * np.arange(N // 2) / N)
```

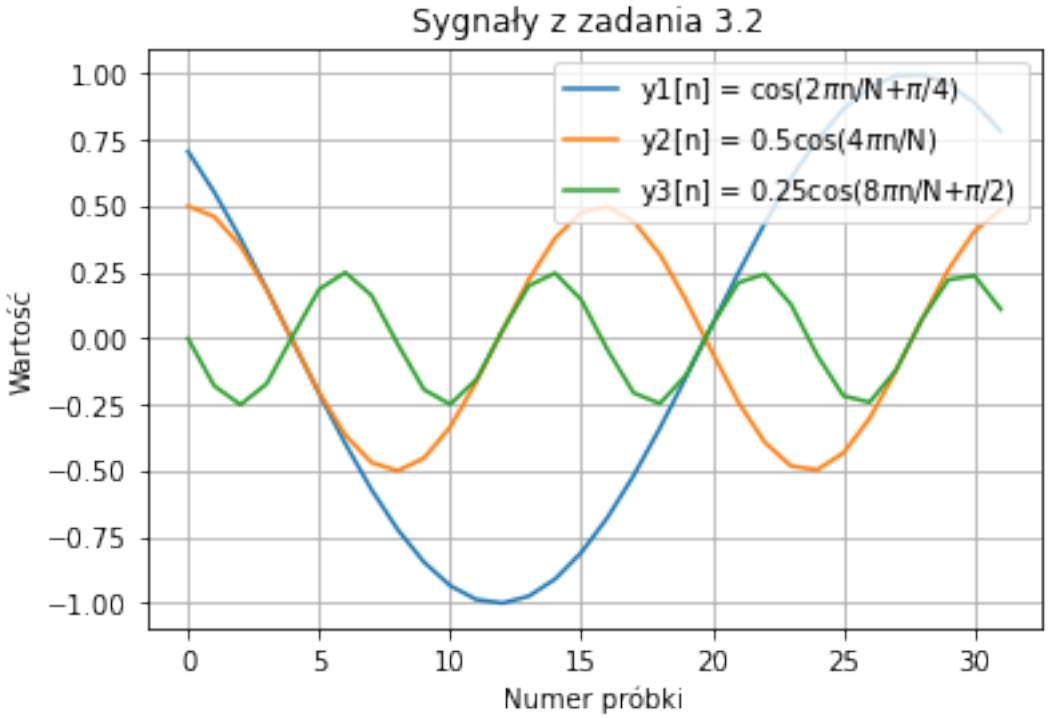
```
    X = np.hstack((Xp + w * Xn, Xp - w * Xn))
    return X
plt.figure(figsize=(10,10),dpi=80)
N= 32
x= np.linspace(0,2*np.pi,N,endpoint=False)
y= np.sin(x)
f1= np.array([[]],dtype=complex)
f1=2*DFT(y)/N
plt.subplot(6,4,1)
plt.stem(y,use_line_collection=True)
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,3)
plt.stem(np.imag(y),use_line_collection=True)
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,5)
plt.stem(np.round(np.real(f1),10),use_line_collection=True)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,7)
plt.stem(np.imag(f1),use_line_collection=True)
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,9)
plt.stem(np.abs(f1),use_line_collection=True)
plt.title('Moduł(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.subplot(6,4,11)
plt.stem(np.angle(f1)/np.pi,use_line_collection=True)
plt.title('Faza(fft(y))')
plt.xlim(0)
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.tight_layout()
#3.1 funkcja programowa
N = 32
x = np.linspace(0,2*np.pi,N,endpoint=False)
y = np.sin(x)
f1= 2*np.fft.fft(y)/N
plt.subplot(6,4,2)
plt.stem(y,use_line_collection=True)
```

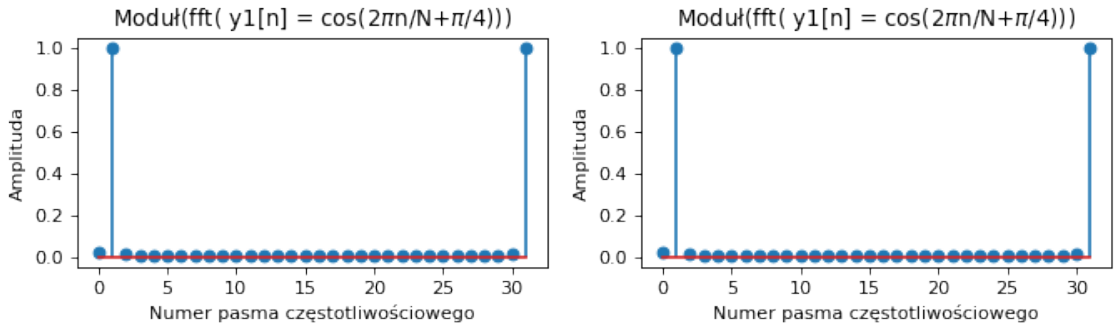
```
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,4)
plt.stem(np.imag(y),use_line_collection=True)
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,6)
plt.stem(np.round(np.real(f1),10),use_line_collection=True)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,8)
plt.stem(np.imag(f1),use_line_collection=True)
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,10)
plt.stem(np.abs(f1),use_line_collection=True)
plt.title('Moduł(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.subplot(6,4,12)
plt.stem(np.angle(f1)/np.pi,use_line_collection=True)
plt.title('Faza(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.tight_layout()

plt.show()
```



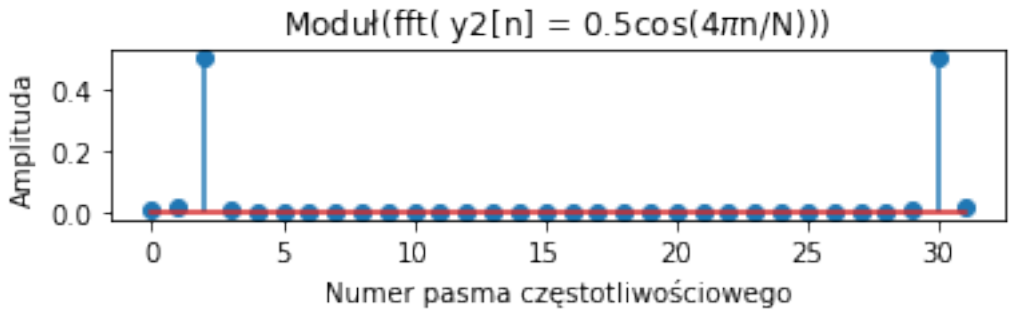
[27]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')

def DFT(x):
    '''Oblicza FFT dla sygnału x o długości będącej potęgą 2.'''
    N = len(x)
    if N == 2:
        return np.array([x[0] + x[1], x[0] - x[1]], dtype=np.complex)
    Xp = DFT(x[::2])
    Xn = DFT(x[1::2])
    w = np.exp(-2 * np.pi * 1j * np.arange(N // 2) / N)
    X = np.hstack((Xp + w * Xn, Xp - w * Xn))
    return X
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
plt.title('Sygnały z zadania 3.2 ')
plt.plot(y1,label = "y1[n] = cos(2$\pi$n/N+$\pi$/4)")
plt.plot(y2,label = "y2[n] = 0.5cos(4$\pi$n/N)")
plt.plot(y3,label = "y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)")
plt.xlabel('Numer próbki')
plt.ylabel('Wartość')
```

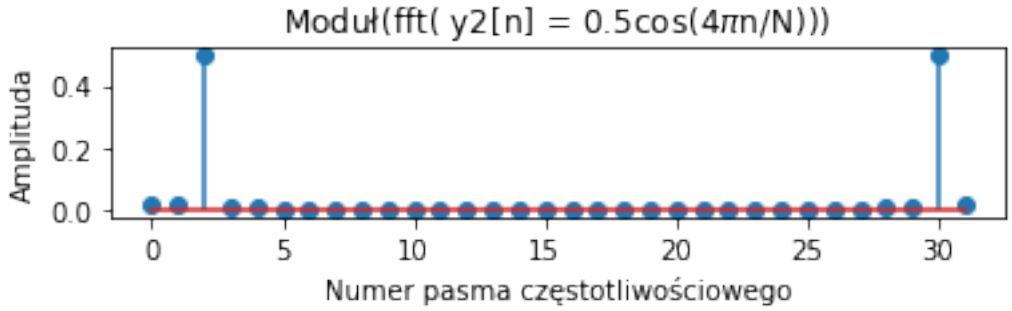
```
plt.legend(loc='upper right')
plt.grid()
plt.show()
# Zadanie3.2
ffty1=2*np.fft.fft(y1)/N
plt.figure(figsize=(10,5),dpi=80)
plt.subplot(2,2,1)
plt.stem(np.abs(ffty1),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.subplot(2,2,2)
ffty1=2*DFT(y1)/N
plt.stem(np.abs(ffty1),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.show()
```



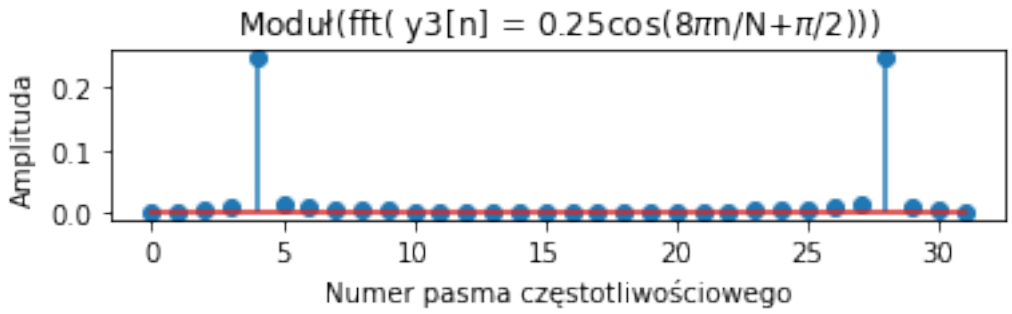


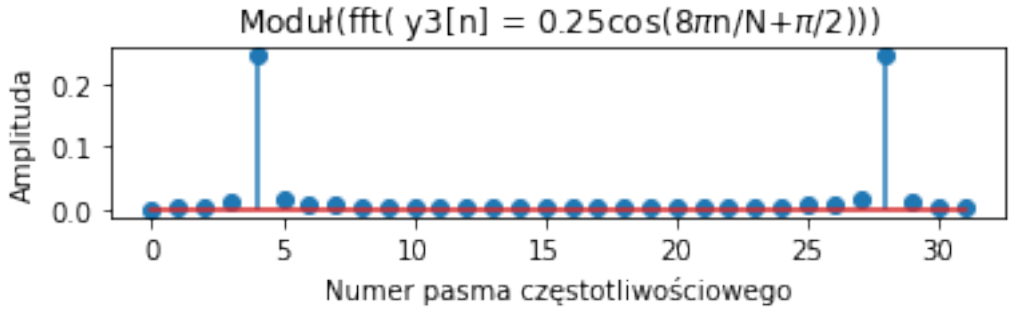
[28]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
plt.subplot(3,1,1)
ffty2=2*np.fft.fft(y2)/N
plt.stem(np.abs(ffty2),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.subplot(3,1,3)
ffty2=2*DFT(y2)/N
plt.stem(np.abs(ffty2),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.show()
```





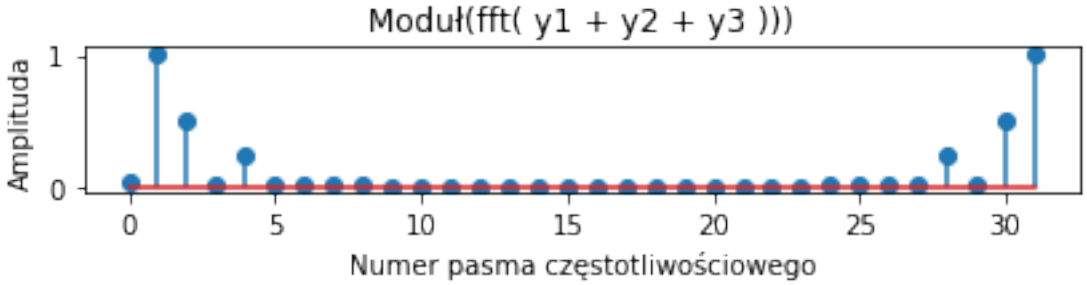
[29]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
plt.subplot(3,1,1)
ffty3=2*np.fft.fft(y3)/N
plt.stem(np.abs(ffty3),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.subplot(3,1,3)
ffty3=2*DFT(y3)/N
plt.stem(np.abs(ffty3),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.show()
```

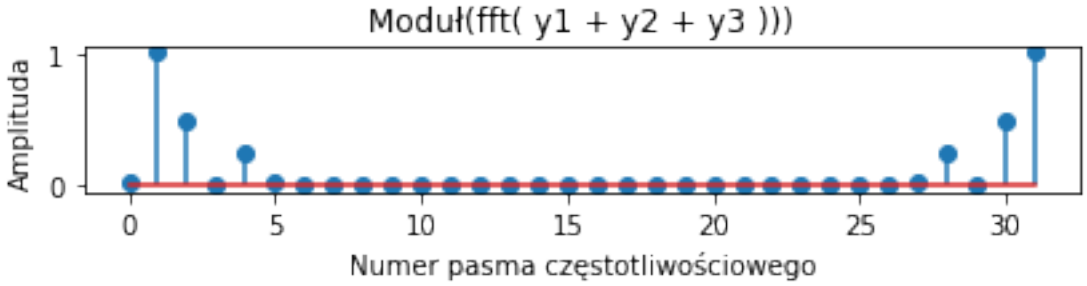




[30]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
```

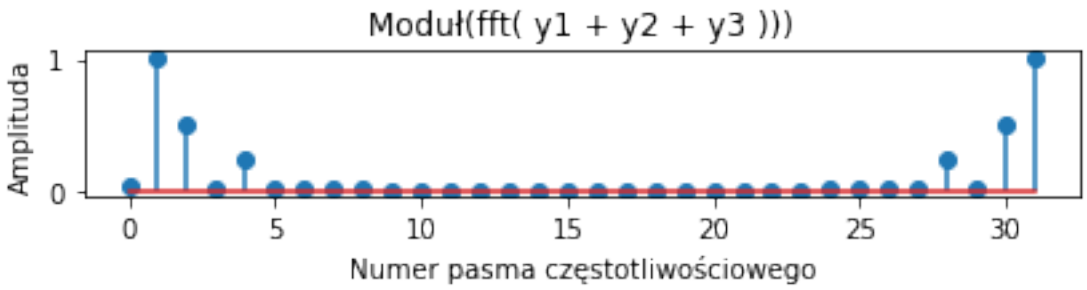
```
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
ffty4=2*DFT(y1+y2+y3)/N
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')

plt.show()
```

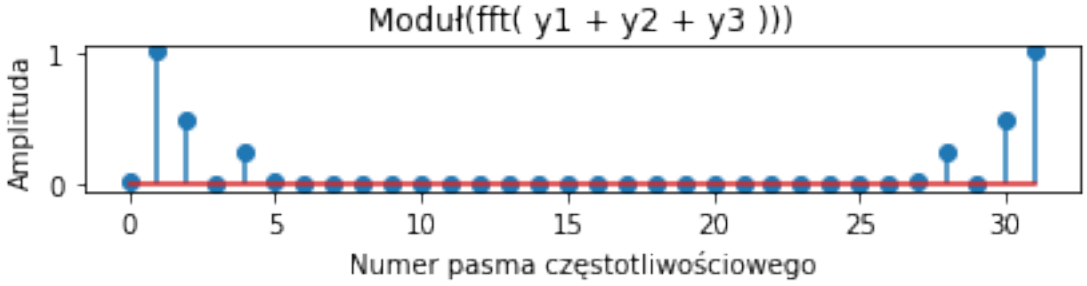




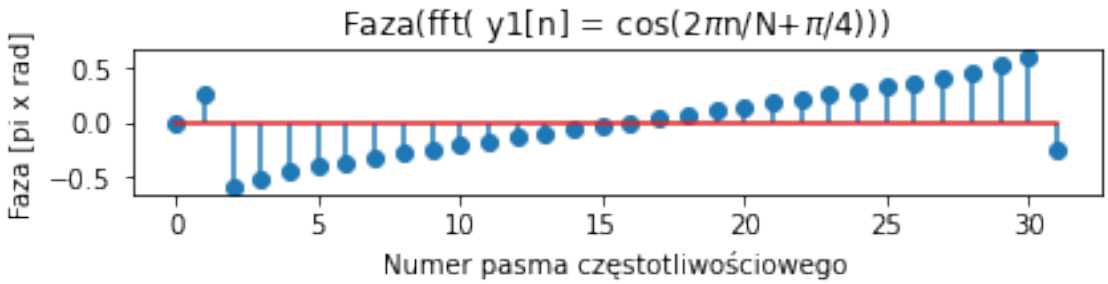
[31]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
```

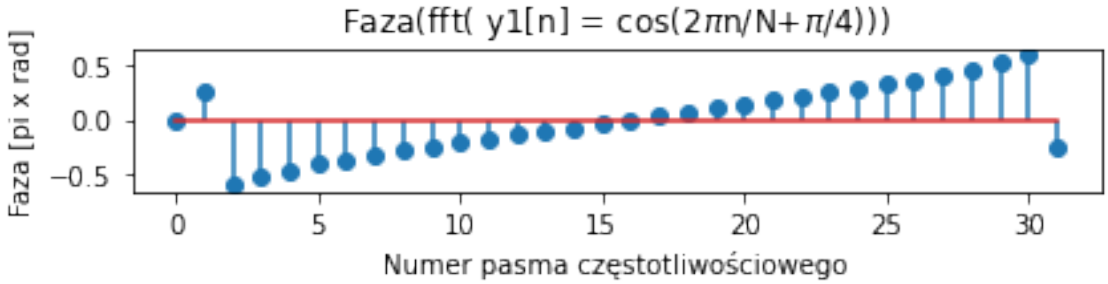
```
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
ffty4=2*DFT(y1+y2+y3)/N
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')

plt.show()
```





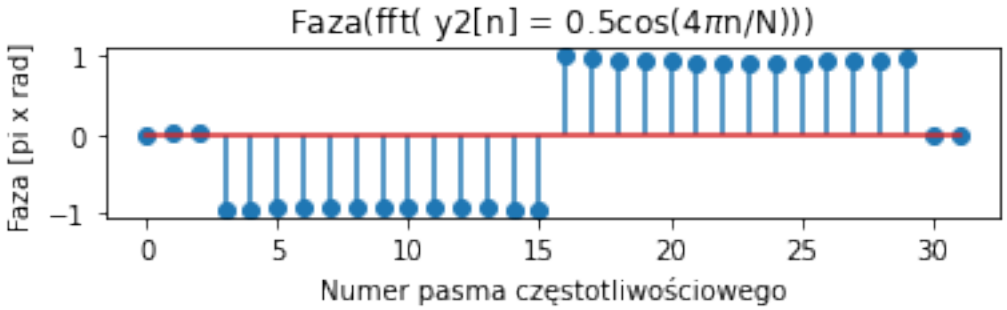
[32]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty1)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
ffty4=2*DFT(y1+y2+y3)/N
plt.stem(np.angle(ffty1)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')

plt.show()
```

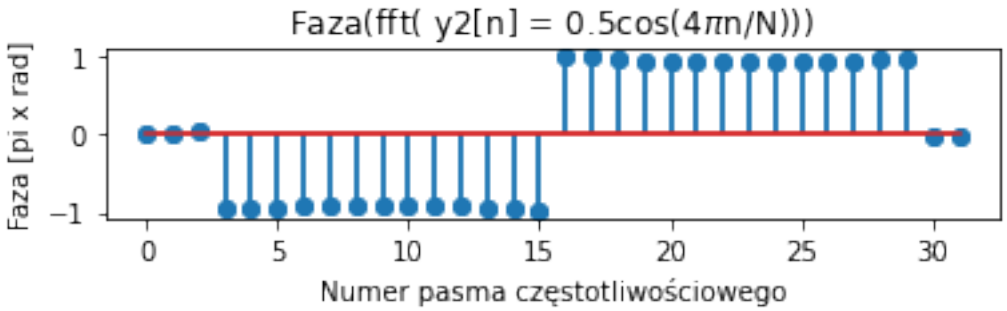




[33]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty2)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
```

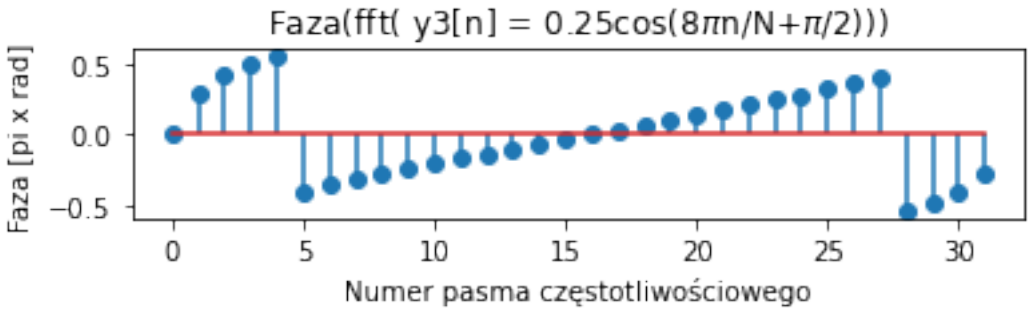
```
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
plt.stem(np.angle(ffty2)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.stem(np.angle(ffty2)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y2[n] = 0.5cos(4$\pi$n/N)))')


plt.show()
```

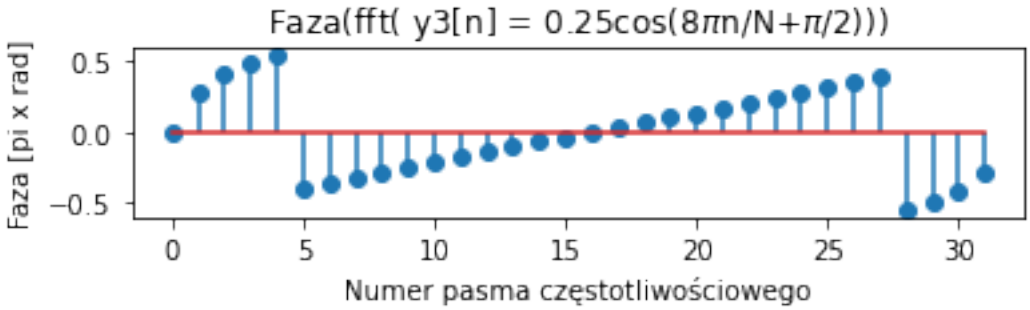




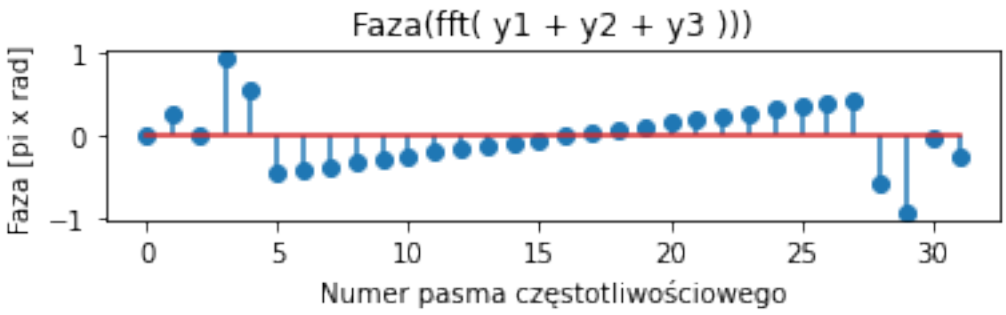
[34]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
```

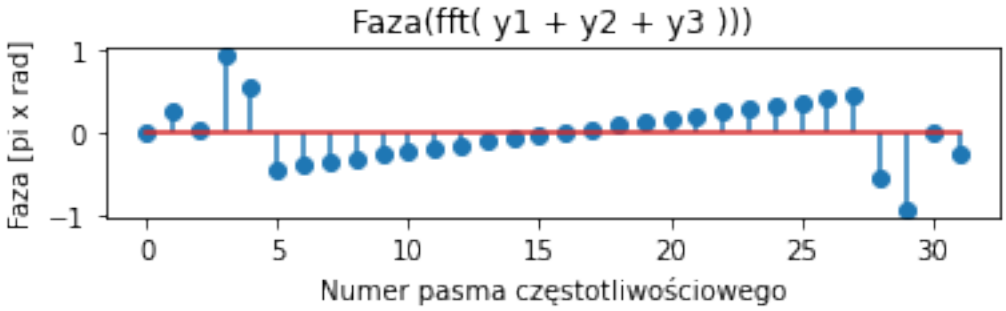
```
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty3)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
plt.stem(np.angle(ffty3)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.show()
```





[35]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty4)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1 + y2 + y3 ))')
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
plt.stem(np.angle(ffty4)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1 + y2 + y3 ))')
plt.show()
```



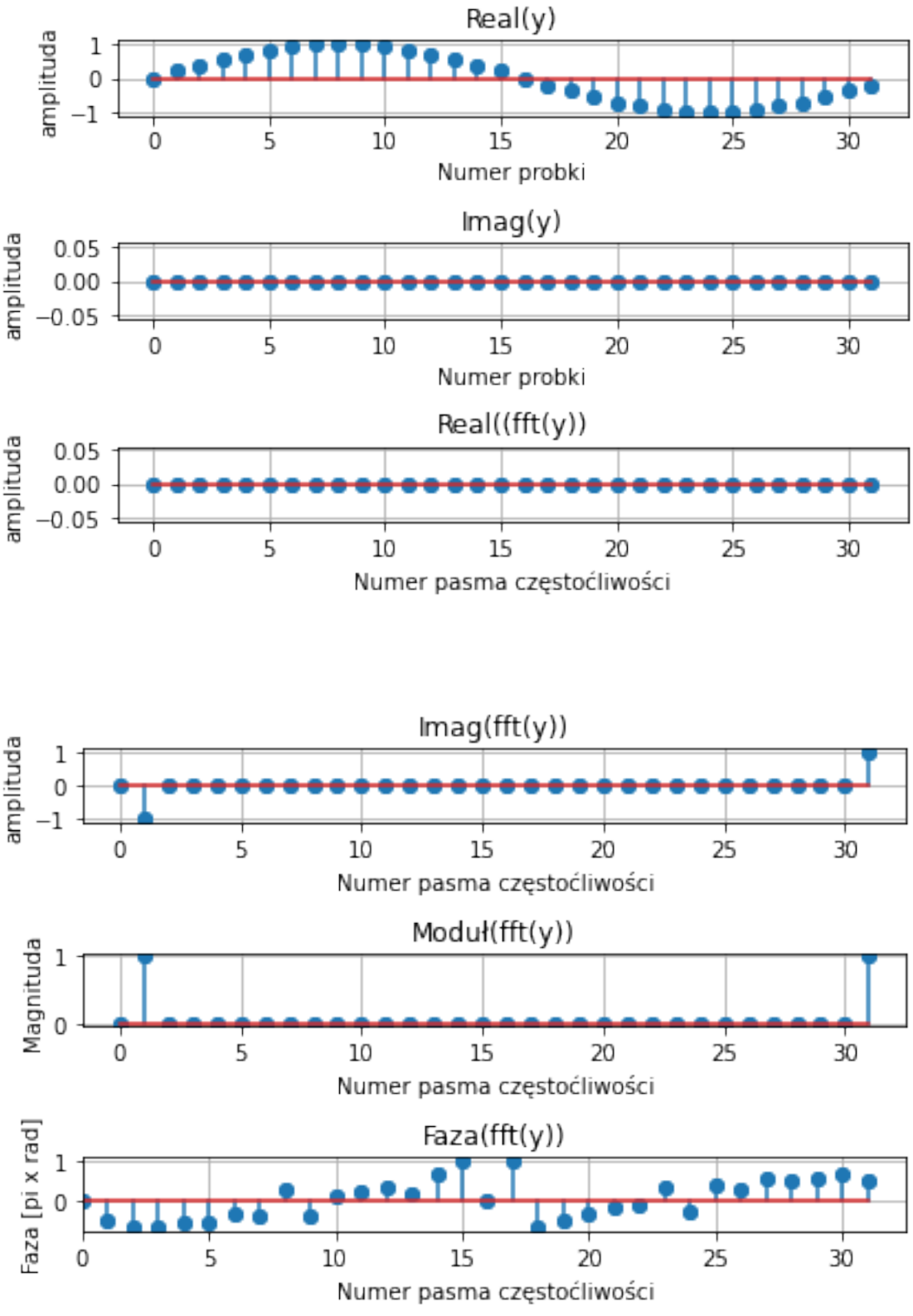


Funkcje DFT do zadania 3.1

[36]:
```
import simpleaudio as sa
import numpy as np
import matplotlib.pyplot as plt


frequency = 200      # Our played note will be 440 Hz
samplePerSec = 44100          # 44100 samples per second

# Generate a 440 Hz sine wave
t=np.arange(0 , 1 , 1/samplePerSec)
note = np.sin(frequency * t * 2 * np.pi)

# Start playback for 2 or 3 seconds
play_obj = sa.play_buffer(note, 1, 2, samplePerSec)

# Wait for playback to finish
play_obj.wait_done()
N = 32
t=np.arange(0 , 0.01 , 1/samplePerSec)
x = np.linspace(0,2*np.pi,N,endpoint=False)
```

```
y = np.sin(x)
f1= 2*np.fft.fft(y)/N


# (S show the DFT points)
S = [0 for _ in range(N)]   # Initialization the S with 0

# DFT calculation
for i in range(N):
    for j in range(N):
        tmp = [((0-1j)*(2*np.pi*i*j)) / N]
        S[i] += s[j] * np.exp(tmp)


# DFT graphs dziala1
plt.figure("Descrete fourier transform")
plt.subplot(3, 1, 1)
plt.scatter([i for i in range(N)], [j for j in y])
plt.stem([i for i in range(N)], [j for j in y])
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.grid(True)

plt.subplot(3, 1, 2)
plt.stem([i for i in range(N)], [j for j in np.imag(y)])
plt.scatter([i for i in range(N)], [j for j in np.imag(y)])
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.grid(True)

plt.subplot(3, 1, 3)
phase_S = np.round(np.real(f1),10)
plt.scatter([i for i in range(N)], phase_S)
plt.stem([i for i in range(N)], phase_S)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.grid(True)

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

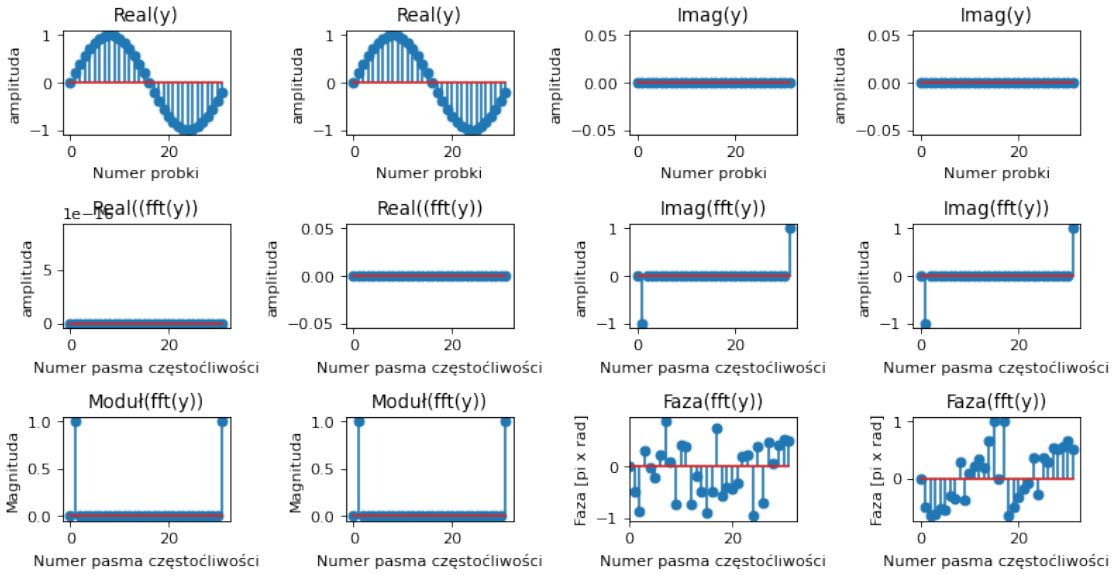
```
plt.subplot(3, 1, 1)
plt.scatter([i for i in range(N)], [j for j in np.imag(f1)])
plt.stem([i for i in range(N)], [j for j in np.imag(f1)])
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.grid(True)

plt.subplot(3, 1, 2)
plt.scatter([i for i in range(N)], np.abs(2*np.fft.fft(y)/N))
plt.stem([i for i in range(N)], np.abs(2*np.fft.fft(y)/N))
plt.title('Moduł(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.grid(True)

plt.subplot(3, 1, 3)
phase_S = np.angle(2*np.fft.fft(y)/N)
plt.scatter([i for i in range(N)], phase_S/np.pi)
plt.stem([i for i in range(N)], phase_S/np.pi)
plt.title('Faza(fft(y))')
plt.xlim(0)
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.grid(True)

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


Real(y)
amplituda
Numer probki
Imag(y)
amplituda
Numer probki
Real((fft(y))
amplituda
Numer pasma częstoćliwości
Imag(fft(y))
amplituda
Numer pasma częstoćliwości
Moduł(fft(y))
Magnituda
Numer pasma częstoćliwości
Faza(fft(y))
Faza [pi x rad]
Numer pasma częstoćliwości

Funkcje DFT do zadania 3.1 oraz 3.2 (porównanie własnej z programową

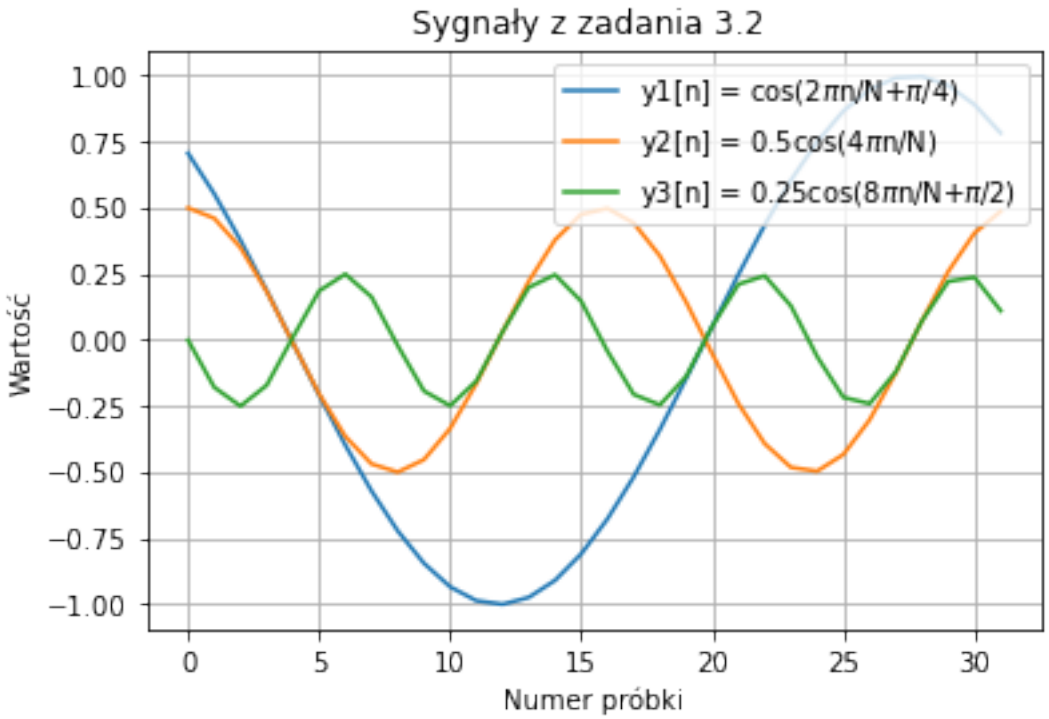
[37]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
from cmath import exp,pi


def DFT(x):
    N = numpy.size(x)
    X = numpy.zeros((N,),dtype=numpy.complex128)
    for m in range(0,N):
        for n in range(0,N):
            X[m] += x[n]*numpy.exp(-numpy.pi*2j*m*n/N)
    return X
plt.figure(figsize=(10,10),dpi=80)
N= 32
x= np.linspace(0,2*np.pi,N,endpoint=False)
y= np.sin(x)
f1= np.array([[]],dtype=complex)
f1=2*DFT(y)/N
plt.subplot(6,4,1)
plt.stem(y,use_line_collection=True)
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,3)
plt.stem(np.imag(y),use_line_collection=True)
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,5)
plt.stem(np.round(np.real(f1),10),use_line_collection=True)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,7)
plt.stem(np.imag(f1),use_line_collection=True)
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,9)
plt.stem(np.abs(f1),use_line_collection=True)
plt.title('Moduł(fft(y))')
```

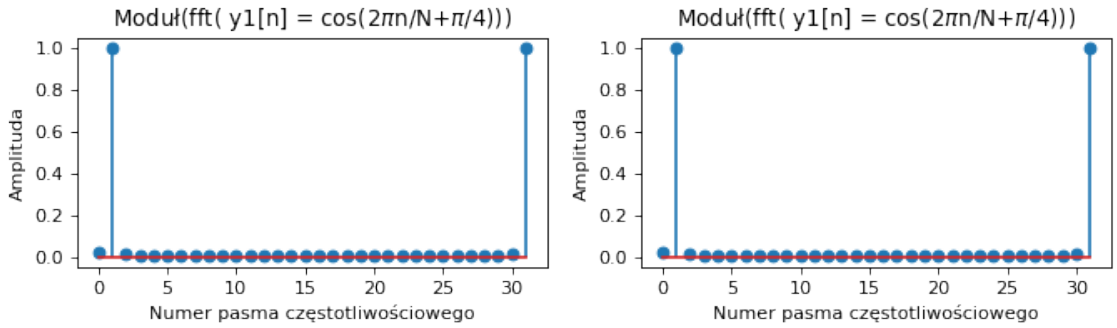
```
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.subplot(6,4,11)
plt.stem(np.angle(f1)/np.pi,use_line_collection=True)
plt.title('Faza(fft(y))')
plt.xlim(0)
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.tight_layout()
#3.1 funkcja programowa
N = 32
x = np.linspace(0,2*np.pi,N,endpoint=False)
y = np.sin(x)
f1= 2*np.fft.fft(y)/N
plt.subplot(6,4,2)
plt.stem(y,use_line_collection=True)
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,4)
plt.stem(np.imag(y),use_line_collection=True)
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,6)
plt.stem(np.round(np.real(f1),10),use_line_collection=True)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,8)
plt.stem(np.imag(f1),use_line_collection=True)
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,10)
plt.stem(np.abs(f1),use_line_collection=True)
plt.title('Moduł(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.subplot(6,4,12)
plt.stem(np.angle(f1)/np.pi,use_line_collection=True)
plt.title('Faza(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.tight_layout()

plt.show()
```



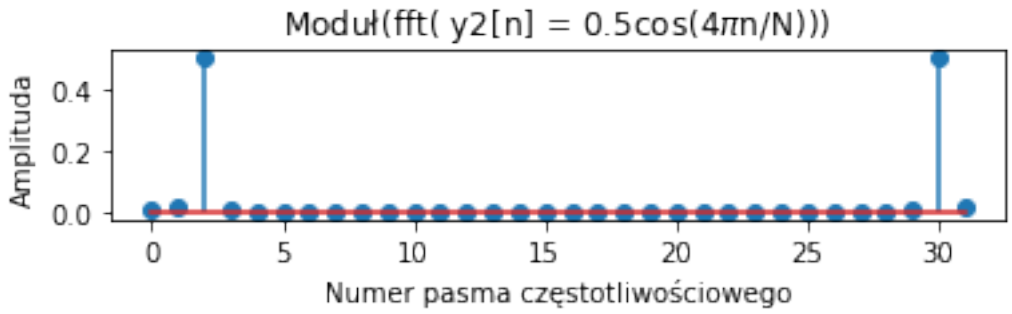
[38]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')


def DFT(x):
    N = numpy.size(x)
    X = numpy.zeros((N,),dtype=numpy.complex128)
    for m in range(0,N):
        for n in range(0,N):
            X[m] += x[n]*numpy.exp(-numpy.pi*2j*m*n/N)
    return X
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
plt.title('Sygnały z zadania 3.2 ')
plt.plot(y1,label = "y1[n] = cos(2$\pi$n/N+$\pi$/4)")
plt.plot(y2,label = "y2[n] = 0.5cos(4$\pi$n/N)")
plt.plot(y3,label = "y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)")
plt.xlabel('Numer próbki')
plt.ylabel('Wartość')
plt.legend(loc='upper right')
plt.grid()
```

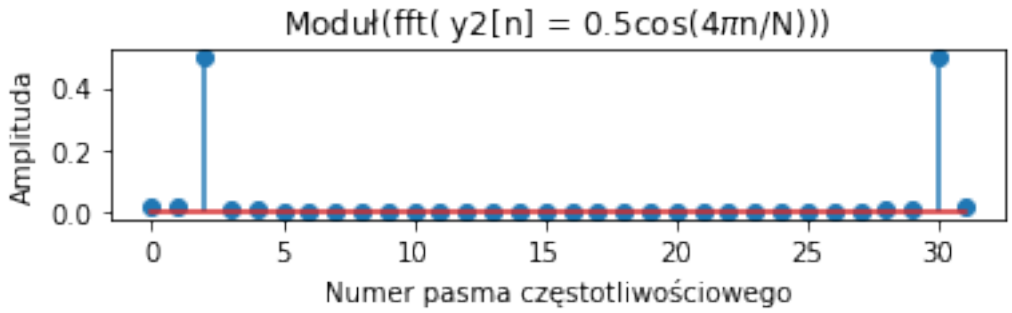
```
plt.show()
# Zadanie3.2
ffty1=2*np.fft.fft(y1)/N
plt.figure(figsize=(10,5),dpi=80)
plt.subplot(2,2,1)
plt.stem(np.abs(ffty1),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.subplot(2,2,2)
ffty1=2*DFT(y1)/N
plt.stem(np.abs(ffty1),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.show()
```



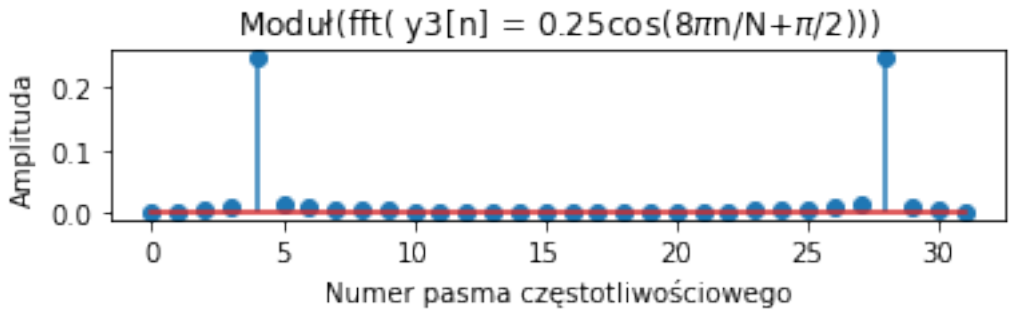


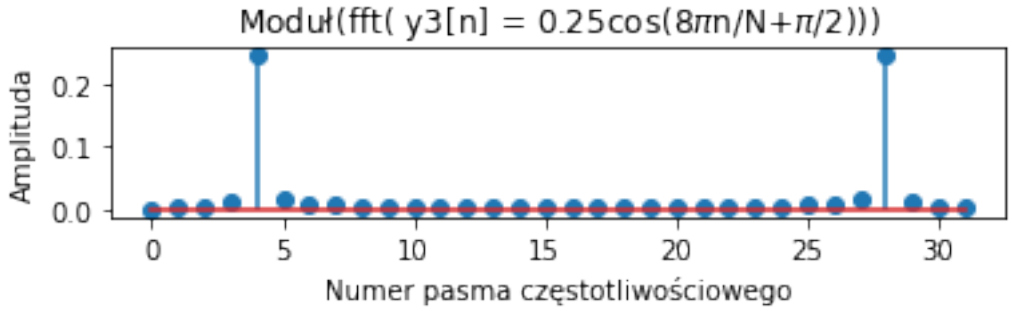
[39]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
plt.subplot(3,1,1)
ffty2=2*np.fft.fft(y2)/N
plt.stem(np.abs(ffty2),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.subplot(3,1,3)
ffty2=2*DFT(y2)/N
plt.stem(np.abs(ffty2),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.show()
```





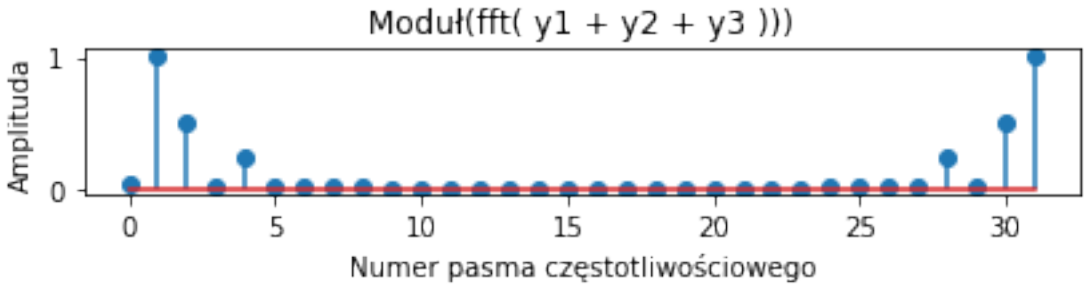
[40]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
plt.subplot(3,1,1)
ffty3=2*np.fft.fft(y3)/N
plt.stem(np.abs(ffty3),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.subplot(3,1,3)
ffty3=2*DFT(y3)/N
plt.stem(np.abs(ffty3),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.show()
```

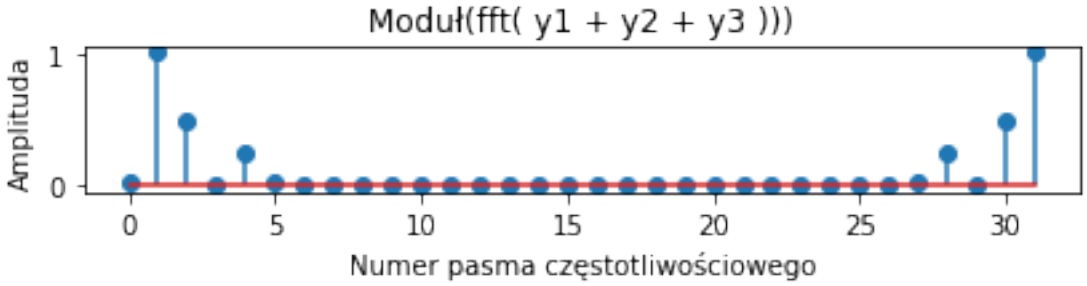




[41]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
```

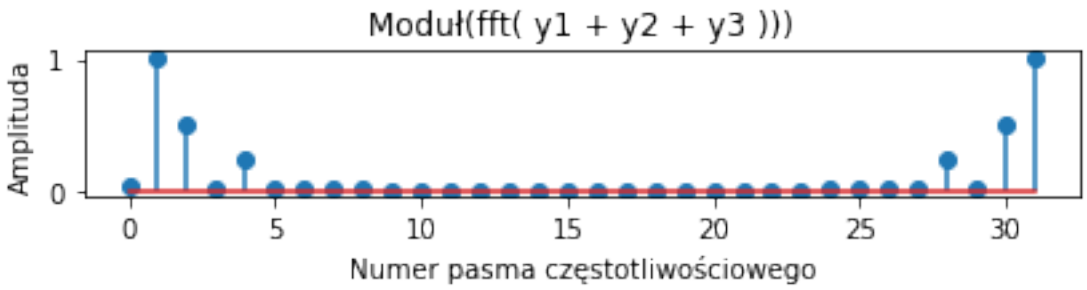
```
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
ffty4=2*DFT(y1+y2+y3)/N
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')

plt.show()
```

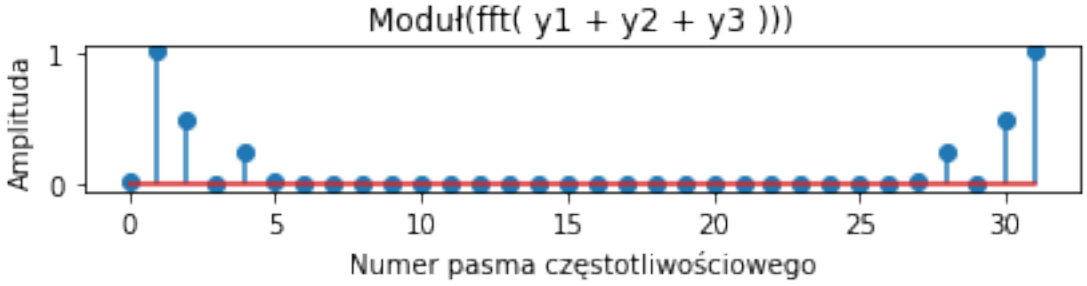




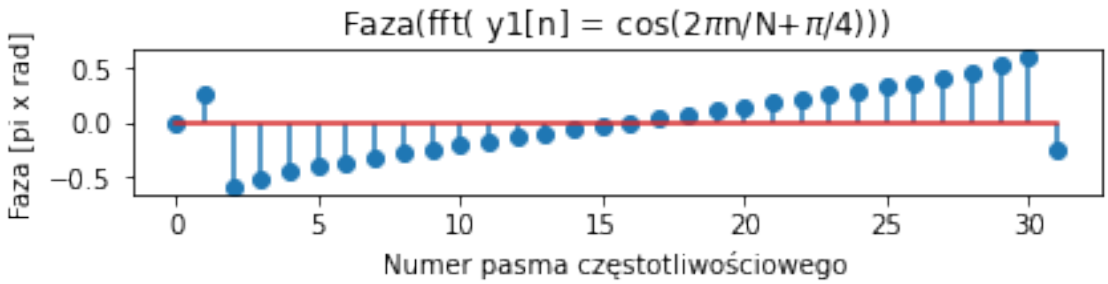
[43]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
```

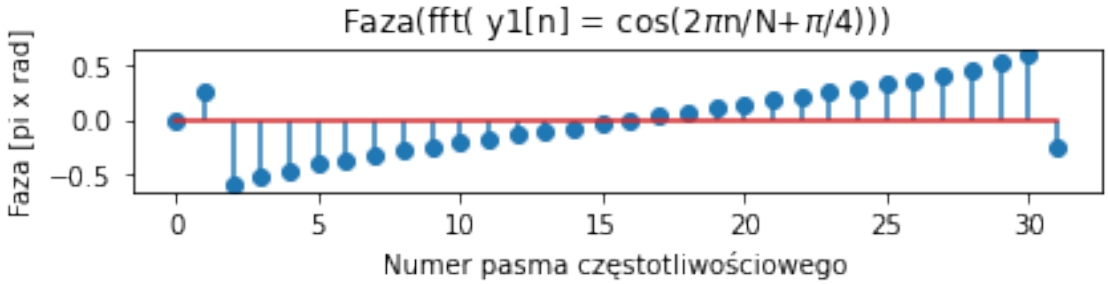
```
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
ffty4=2*DFT(y1+y2+y3)/N
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')

plt.show()
```





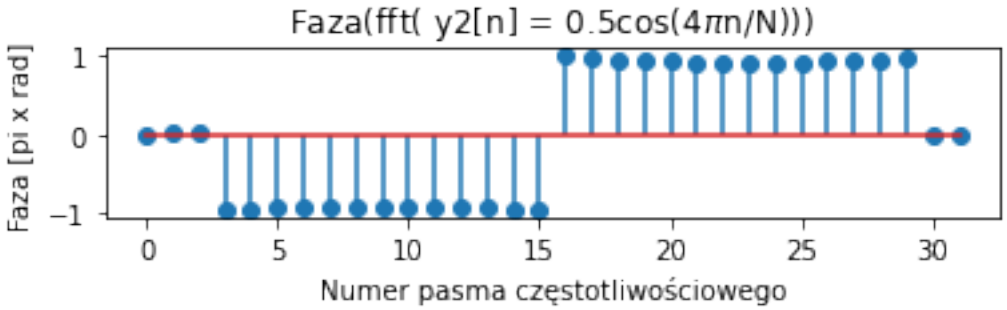
[44]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty1)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
ffty4=2*DFT(y1+y2+y3)/N
plt.stem(np.angle(ffty1)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')

plt.show()
```

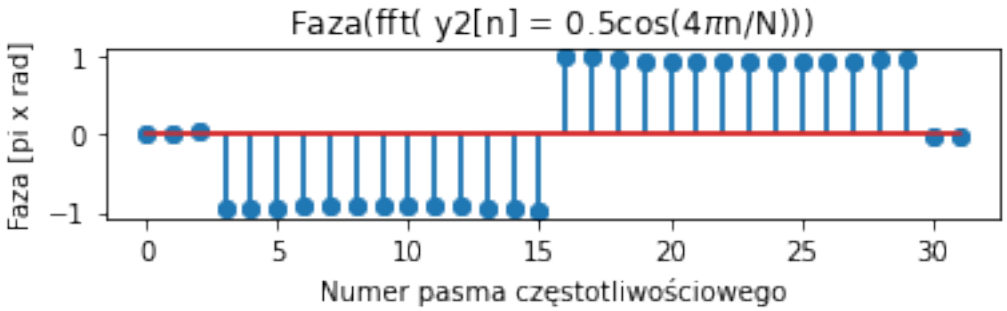




[45]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty2)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
```

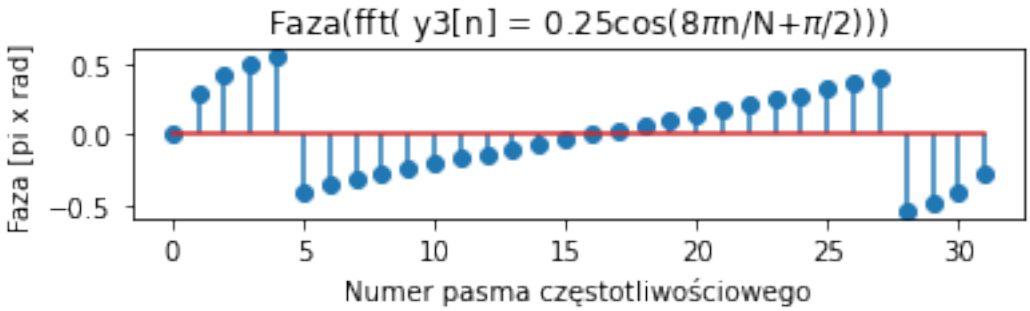
```
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
plt.stem(np.angle(ffty2)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.stem(np.angle(ffty2)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y2[n] = 0.5cos(4$\pi$n/N)))')


plt.show()
```

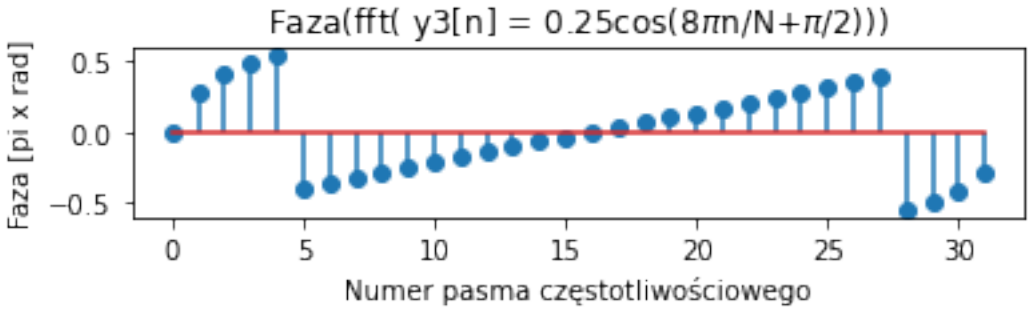




[46]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
```

```
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.angle(ffty3)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*DFT(y1)/N
ffty2=2*DFT(y2)/N
ffty3=2*DFT(y3)/N
plt.stem(np.angle(ffty3)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.show()
```

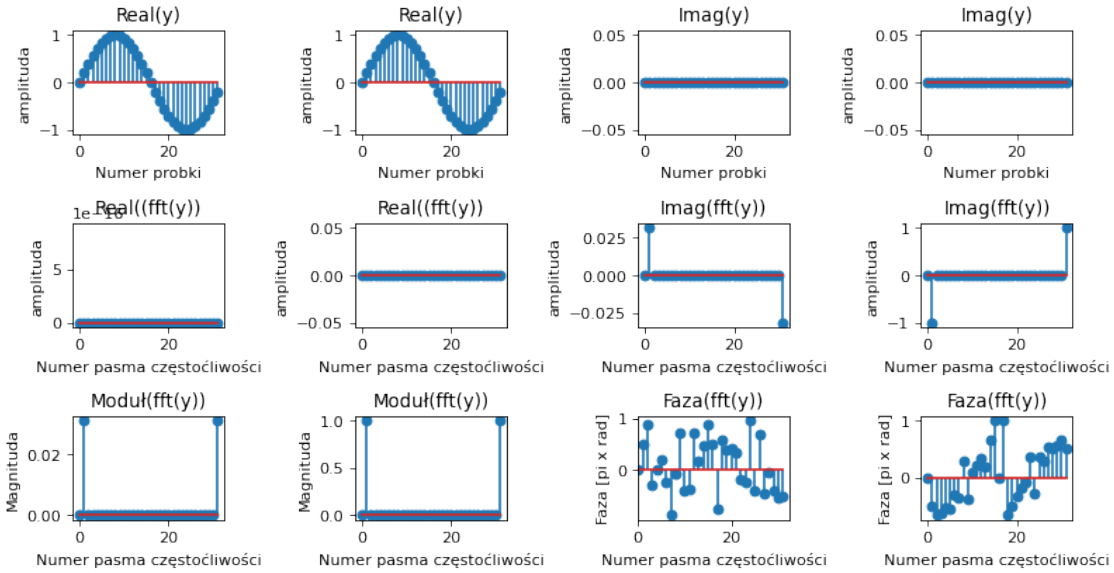




1.0.2 Funkcje IDFT do zadania 3.1 oraz 3.2 (porównanie własnej z programową

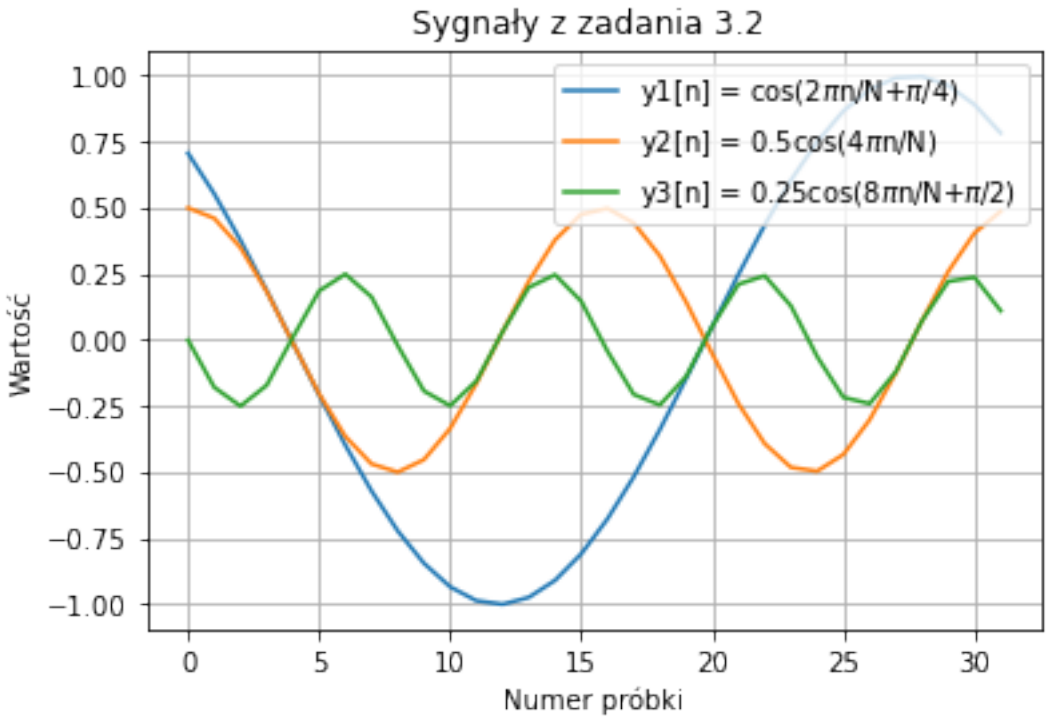
[47]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
from cmath import exp,pi


def IDFT(x):
    N = numpy.size(x)
    X = numpy.zeros((N,),dtype=numpy.complex128)
    for m in range(0,N):
        for n in range(0,N):
            X[m] += x[n]*numpy.exp(numpy.pi*2j*m*n/N)
    return X/N
plt.figure(figsize=(10,10),dpi=80)
N= 32
x= np.linspace(0,2*np.pi,N,endpoint=False)
y= np.sin(x)
f1= np.array([[]],dtype=complex)
f1=2*IDFT(y)/N
plt.subplot(6,4,1)
plt.stem(y,use_line_collection=True)
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,3)
plt.stem(np.imag(y),use_line_collection=True)
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,5)
plt.stem(np.round(np.real(f1),10),use_line_collection=True)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,7)
plt.stem(np.imag(f1),use_line_collection=True)
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,9)
plt.stem(np.abs(f1),use_line_collection=True)
plt.title('Moduł(fft(y))')
```

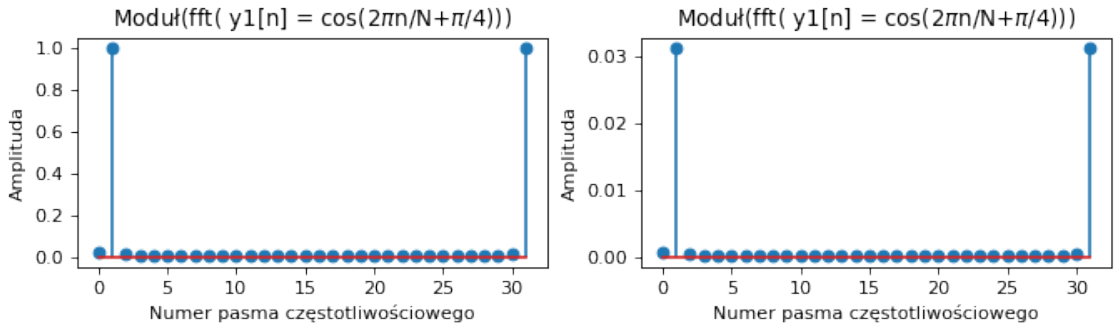
```
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.subplot(6,4,11)
plt.stem(np.angle(f1)/np.pi,use_line_collection=True)
plt.title('Faza(fft(y))')
plt.xlim(0)
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.tight_layout()
#3.1 funkcja programowa
N = 32
x = np.linspace(0,2*np.pi,N,endpoint=False)
y = np.sin(x)
f1= 2*np.fft.fft(y)/N
plt.subplot(6,4,2)
plt.stem(y,use_line_collection=True)
plt.title('Real(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,4)
plt.stem(np.imag(y),use_line_collection=True)
plt.title('Imag(y)')
plt.xlabel('Numer probki')
plt.ylabel('amplituda')
plt.subplot(6,4,6)
plt.stem(np.round(np.real(f1),10),use_line_collection=True)
plt.title('Real((fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,8)
plt.stem(np.imag(f1),use_line_collection=True)
plt.title('Imag(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('amplituda')
plt.subplot(6,4,10)
plt.stem(np.abs(f1),use_line_collection=True)
plt.title('Moduł(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Magnituda')
plt.subplot(6,4,12)
plt.stem(np.angle(f1)/np.pi,use_line_collection=True)
plt.title('Faza(fft(y))')
plt.xlabel('Numer pasma częstoćliwości')
plt.ylabel('Faza [pi x rad]')
plt.tight_layout()

plt.show()
```



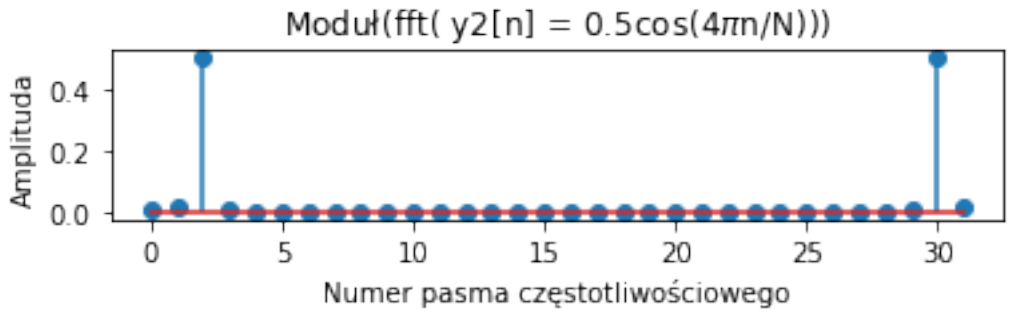
[48]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')


def IDFT(x):
    N = numpy.size(x)
    X = numpy.zeros((N,),dtype=numpy.complex128)
    for m in range(0,N):
        for n in range(0,N):
            X[m] += x[n]*numpy.exp(numpy.pi*2j*m*n/N)
    return X/N
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
plt.title('Sygnały z zadania 3.2 ')
plt.plot(y1,label = "y1[n] = cos(2$\pi$n/N+$\pi$/4)")
plt.plot(y2,label = "y2[n] = 0.5cos(4$\pi$n/N)")
plt.plot(y3,label = "y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)")
plt.xlabel('Numer próbki')
plt.ylabel('Wartość')
plt.legend(loc='upper right')
plt.grid()
```

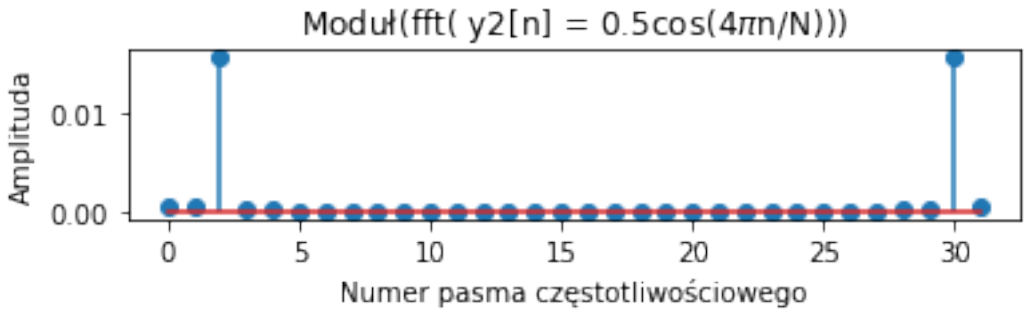
```
plt.show()
# Zadanie3.2
ffty1=2*np.fft.fft(y1)/N
plt.figure(figsize=(10,5),dpi=80)
plt.subplot(2,2,1)
plt.stem(np.abs(ffty1),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.subplot(2,2,2)
ffty1=2*IDFT(y1)/N
plt.stem(np.abs(ffty1),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')
plt.show()
```



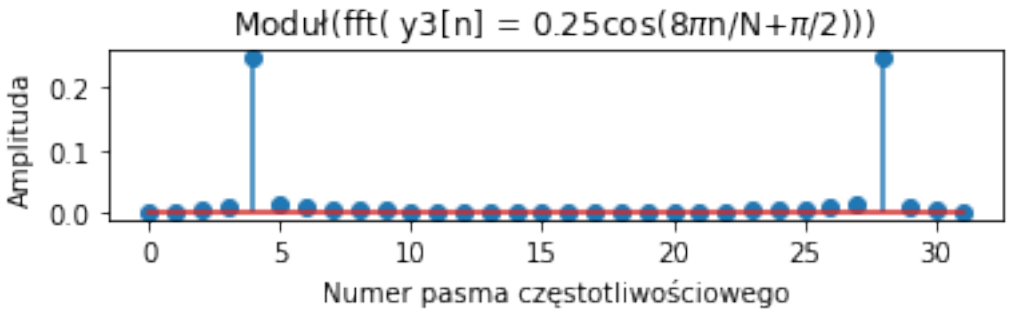


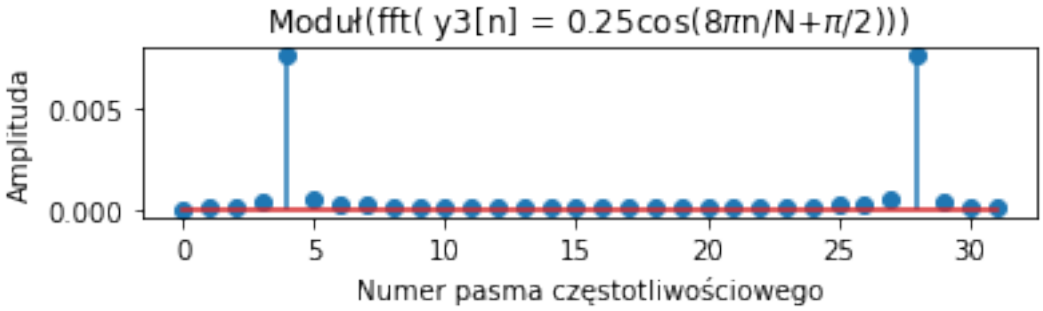
[49]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
plt.subplot(3,1,1)
ffty2=2*np.fft.fft(y2)/N
plt.stem(np.abs(ffty2),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.subplot(3,1,3)
ffty2=2*IDFT(y2)/N
plt.stem(np.abs(ffty2),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y2[n] = 0.5cos(4$\pi$n/N)))')
plt.show()
```





[50]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
plt.subplot(3,1,1)
ffty3=2*np.fft.fft(y3)/N
plt.stem(np.abs(ffty3),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.subplot(3,1,3)
ffty3=2*IDFT(y3)/N
plt.stem(np.abs(ffty3),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y3[n] = 0.25cos(8$\pi$n/N+$\pi$/2)))')
plt.show()
```

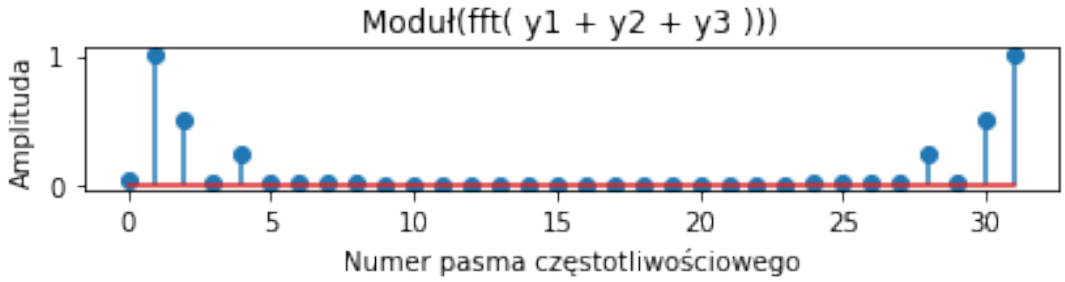


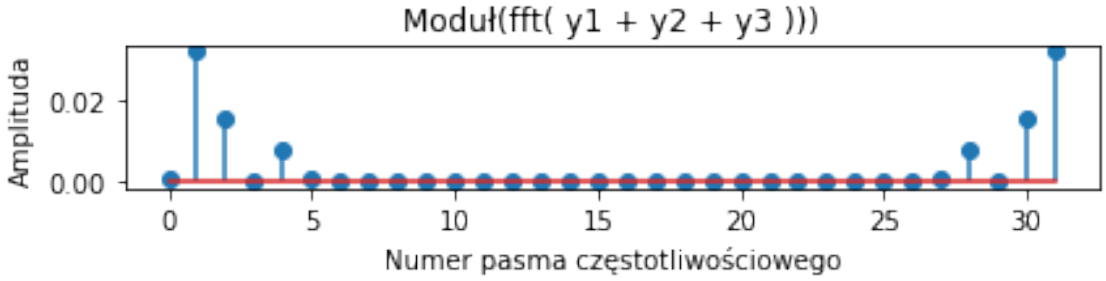


[51]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*IDFT(y1)/N
```

```
ffty2=2*IDFT(y2)/N
ffty3=2*IDFT(y3)/N
ffty4=2*IDFT(y1+y2+y3)/N
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
```

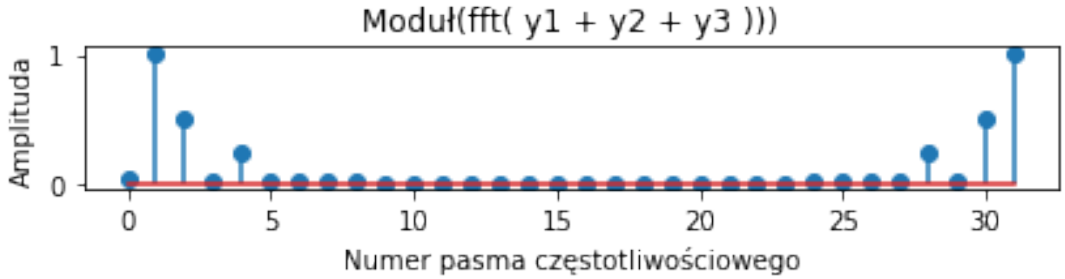
[51]: Text(0.5, 1.0, 'Moduł(fft(y1 + y2 + y3)))')

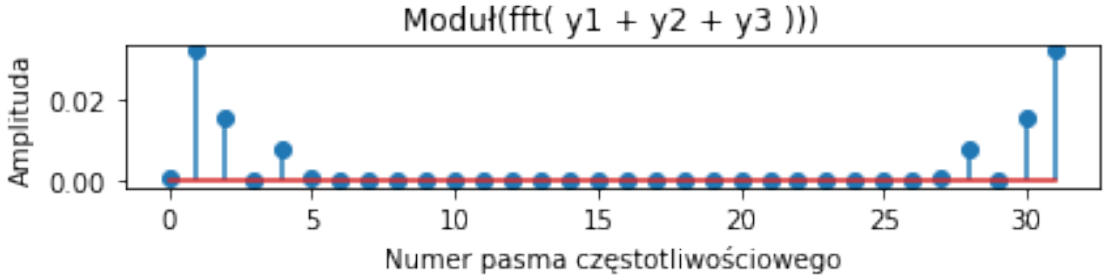




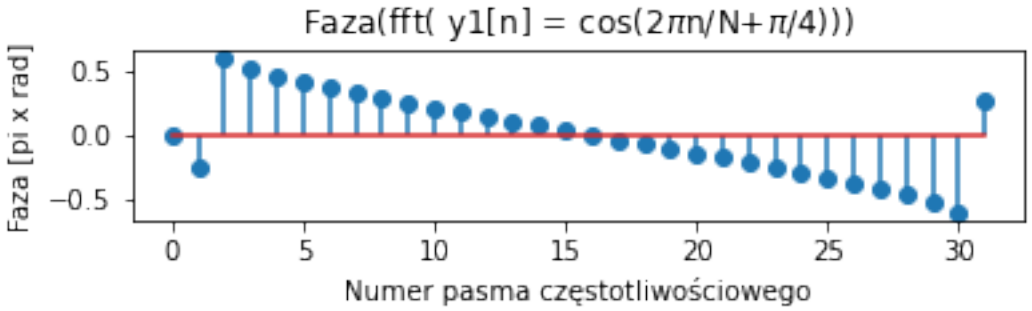
[52]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)
```

```
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')
plt.tight_layout()
plt.subplot(3,1,3)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*IDFT(y1)/N
ffty2=2*IDFT(y2)/N
ffty3=2*IDFT(y3)/N
ffty4=2*IDFT(y1+y2+y3)/N
plt.stem(np.abs(ffty4),use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Amplituda')
plt.title('Moduł(fft( y1 + y2 + y3 )))')

plt.show()
```



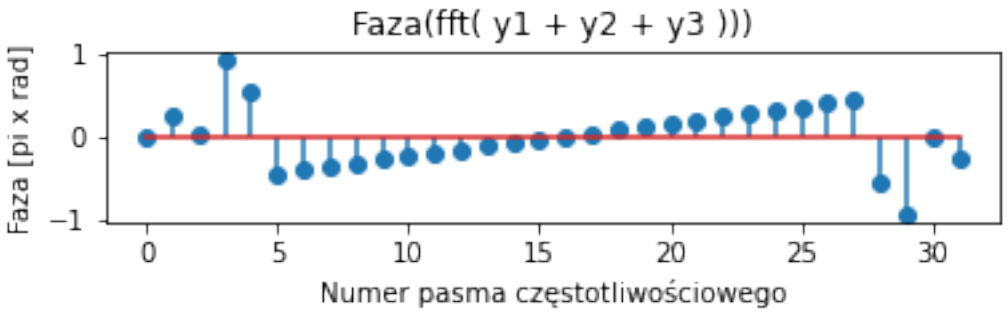


[57]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N


plt.tight_layout()
plt.subplot(3,1,1)
N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*IDFT(y1)/N
ffty2=2*IDFT(y2)/N
ffty3=2*IDFT(y3)/N
ffty4=2*IDFT(y1+y2+y3)/N
plt.stem(np.angle(ffty1)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1[n] = cos(2$\pi$n/N+$\pi$/4)))')

plt.show()
```



[58]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import dft
import scipy as sp
```

```
import warnings
warnings.filterwarnings('ignore')
ffty1=2*np.fft.fft(y1)/N
ffty2=2*np.fft.fft(y2)/N
ffty3=2*np.fft.fft(y3)/N
ffty4=2*np.fft.fft(y1+y2+y3)/N
plt.subplot(3,1,1)

N=32
n=np.linspace(0,np.pi*10,N)
y1=np.cos(2*np.pi*n/N+np.pi/4)
y2=0.5*np.cos(4*np.pi*n/N)
y3=0.25*np.cos(8*np.pi*n/N+np.pi/2)
ffty1=2*IDFT(y1)/N
ffty2=2*IDFT(y2)/N
ffty3=2*IDFT(y3)/N
plt.stem(np.angle(ffty4)/np.pi,use_line_collection=True)
plt.xlabel('Numer pasma częstotliwościowego')
plt.ylabel('Faza [pi x rad]')
plt.title('Faza(fft( y1 + y2 + y3 )))')
plt.show()
```



Wnioski:

Porównując wyniki z wcześniej otrzymanymi można dostrzec, że algorytm Algorytm Cooleya-Tukeya został zrealizowany poprawnie.