Łukasz Turowski, TD_2020_20A, 45136

LAB_03

Zadanie 1.

Funkcja Dyskretna Transformata Fouriera (DFT):

$$X(k) = \sum_{n=0}^{N-1} x(n) w_N^{-kn},\ 0 \leqslant k \leqslant N-1$$

$$w_N = e^{i\frac{2\pi}{N}}$$

```
def DFT(n):
    xk = []
    for i in range(len(n)):
        temp = 0
        for j in range(len(n)):
            temp += n[j] * np.exp(1j * 2 * np.pi / len(n)) ** (-i * j)
        xk.append(temp)
    return xk
```

Zadanie 2.

Funkcja Widmo amplitudowe:

$$M(k) = \sqrt{Re[X(k)]^2 + Im[X(k)]^2}$$

```
def M(x):
    temp = []
    for i in range(0, len(np.real(x))):
        temp.append(np.sqrt(np.real(x[i]) ** 2 + np.imag(x[i]) ** 2))
    return temp
```

Wartość amplitudy w skali decybelowej:

$$M'(k) = 10 \cdot \log_{10} M(k)$$

```
def M_pri(x):
    return 10 * np.log10(x)
```

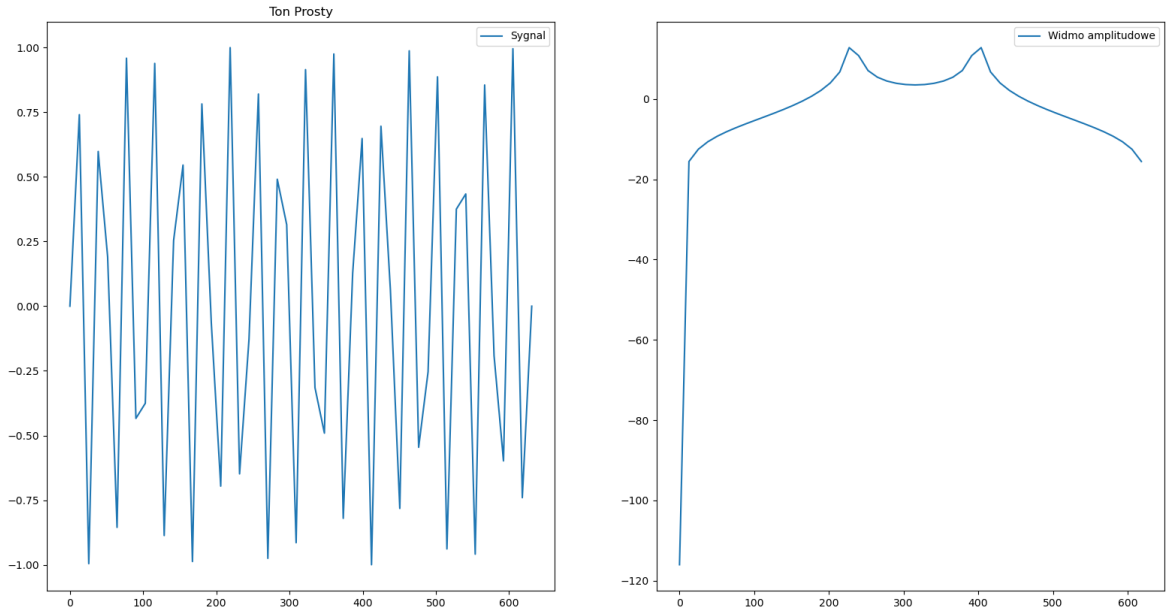
Skala częstotliwości:

$$f_k = k \cdot \frac{f_s}{N}.$$

```
def skala(x, fs):
    temp = []
    for k in range(len(x)):
        temp.append(k * (fs / len(x)))
    return temp
```

Wykresy dla Tonu Prostego:

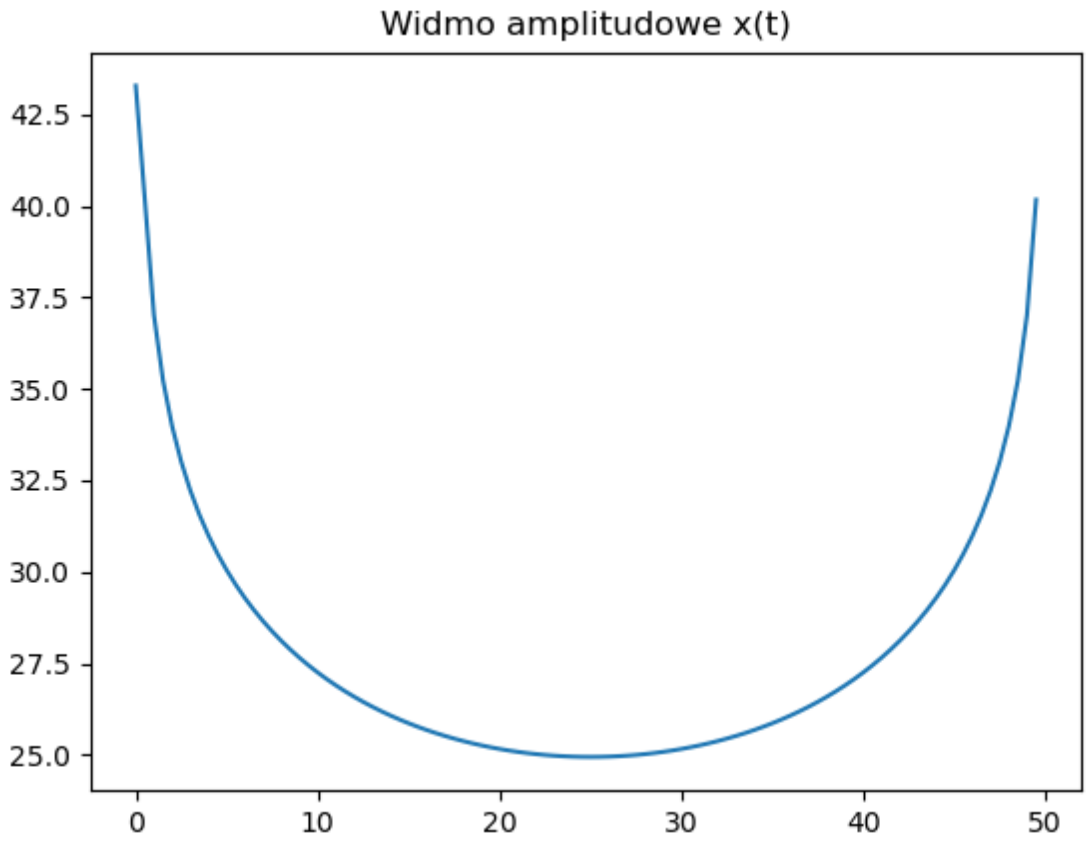
```
n = np.linspace(0, 631)
plt.subplot(1, 2, 1)
plt.plot(n, Ton(a, f, fi, n), label='Sygnal')
plt.legend()
plt.title('Ton Prosty')
plt.subplot(1, 2, 2)
plt.plot(skala(DFT(Ton(a, f, fi, n)), 631), M_pri(M(DFT(Ton(a, f, fi, n)))),
label='Widmo amplitudowe')
plt.legend()
plt.show()
```



Zadanie 3.

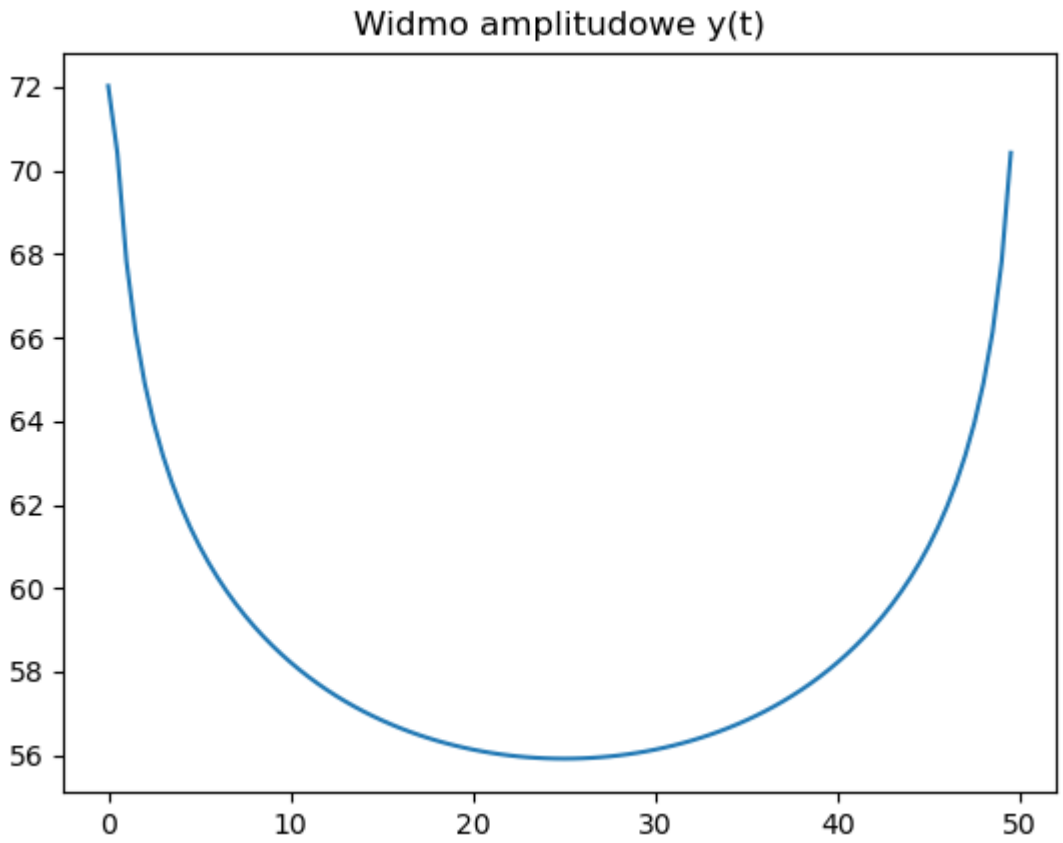
Widmo amplitudowe x(t):

```
def x(t):
    return A * t ** 2 + B * t + C

plt.plot(skala(DFT(x(t)), 50), M_pri(M(DFT(x(t)))))
plt.title('Widmo amplitudowe x(t)')
plt.show()
```



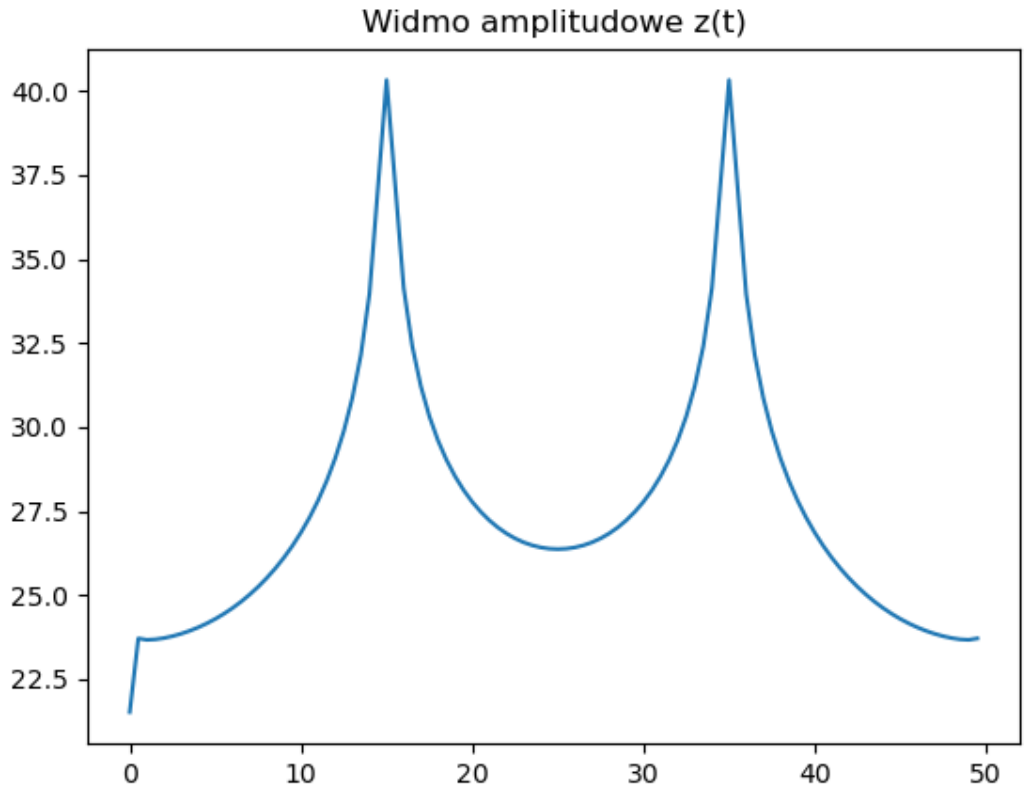
Widmo amplitudowe y(t):

```
def y(t):
    return 2 * x(t) ** 2 + 12 * np.cos(t)

plt.plot(skala(DFT(y(t)), 50), M_pri(M(DFT(y(t)))))
plt.title('Widmo amplitudowe y(t)')
plt.show()
```



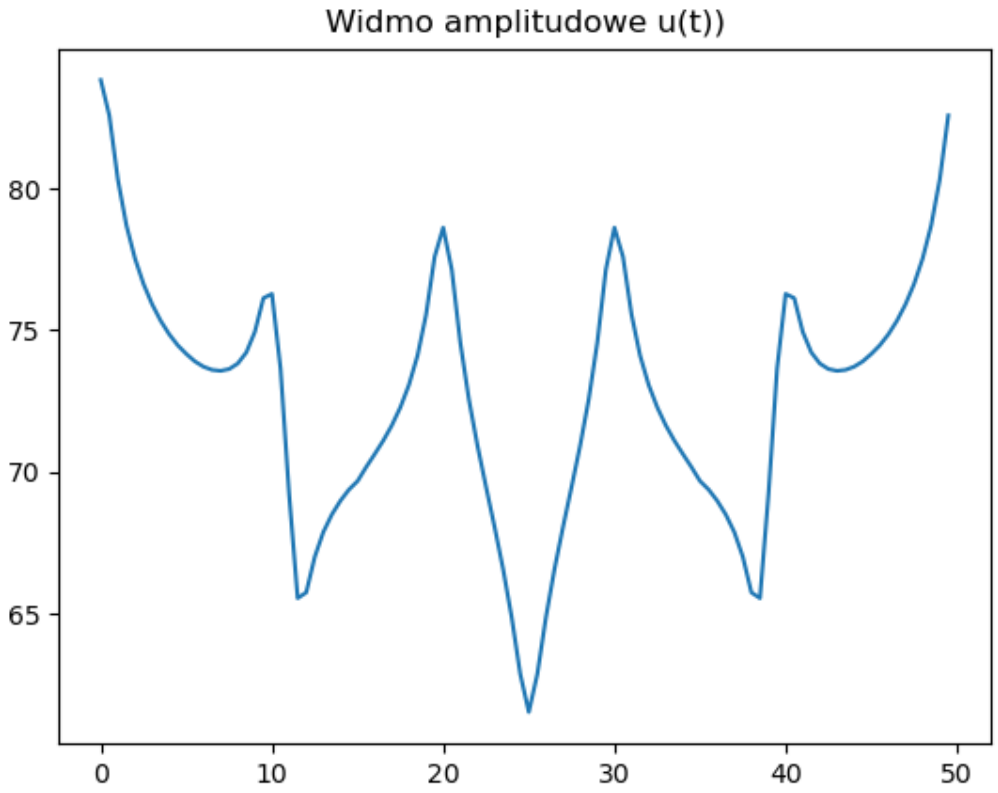
Widmo amplitudowe z(t):

```
def z(t):
    return np.sin(2 * np.pi * 7 * t) * x(t) - 0.2 * np.log10(np.abs(y(t)) + np.pi)

plt.plot(skala(DFT(z(t)), 50), M_pri(M(DFT(z(t)))))
plt.title('Widmo amplitudowe z(t)')
plt.show()
```



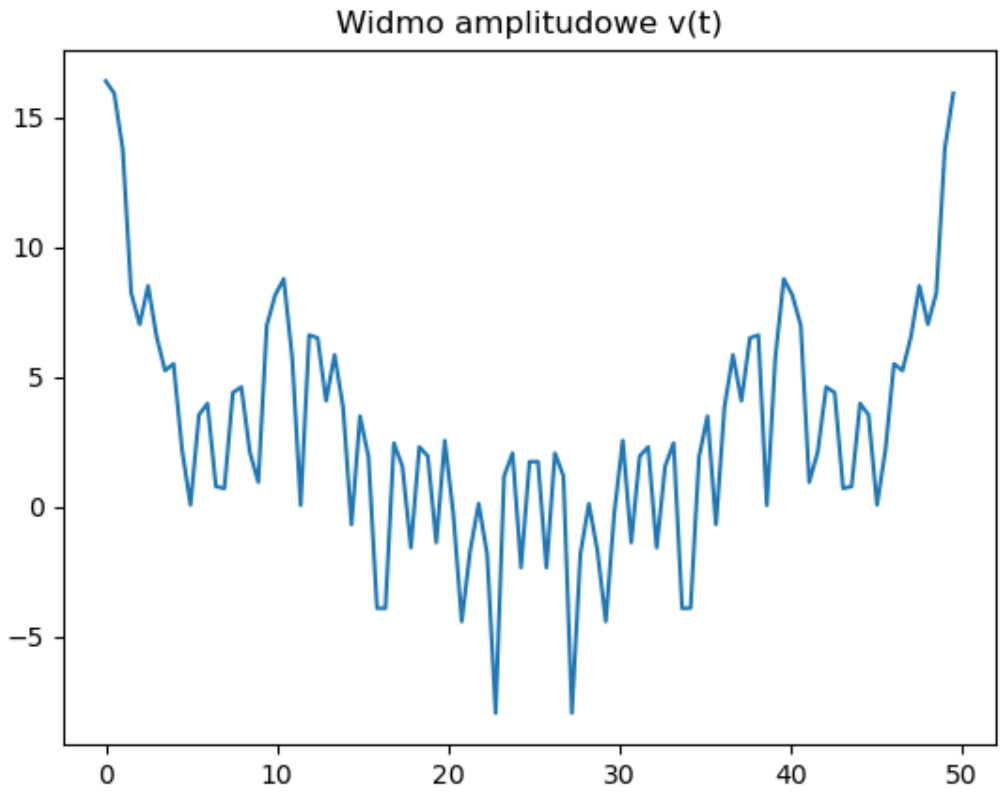
Widmo amplitudowe u(t):

```
def u(t):
    return np.sqrt(np.abs(y(t) * y(t) * z(t))) - 1.8 * np.sin(0.4 * t * z(t) *
x(t))

plt.plot(skala(DFT(u(t)), 50), M_pri(M(DFT(u(t)))))
plt.title('Widmo amplitudowe u(t))')
plt.show()
```



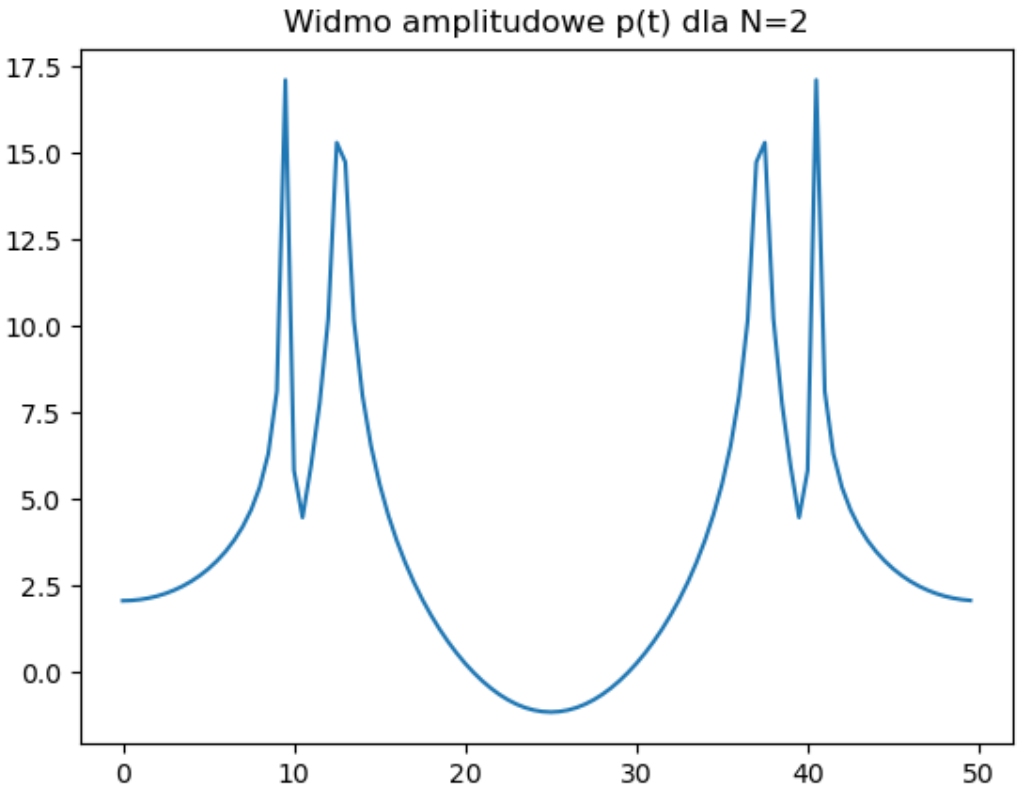
Widmo amplitudowe v(t):

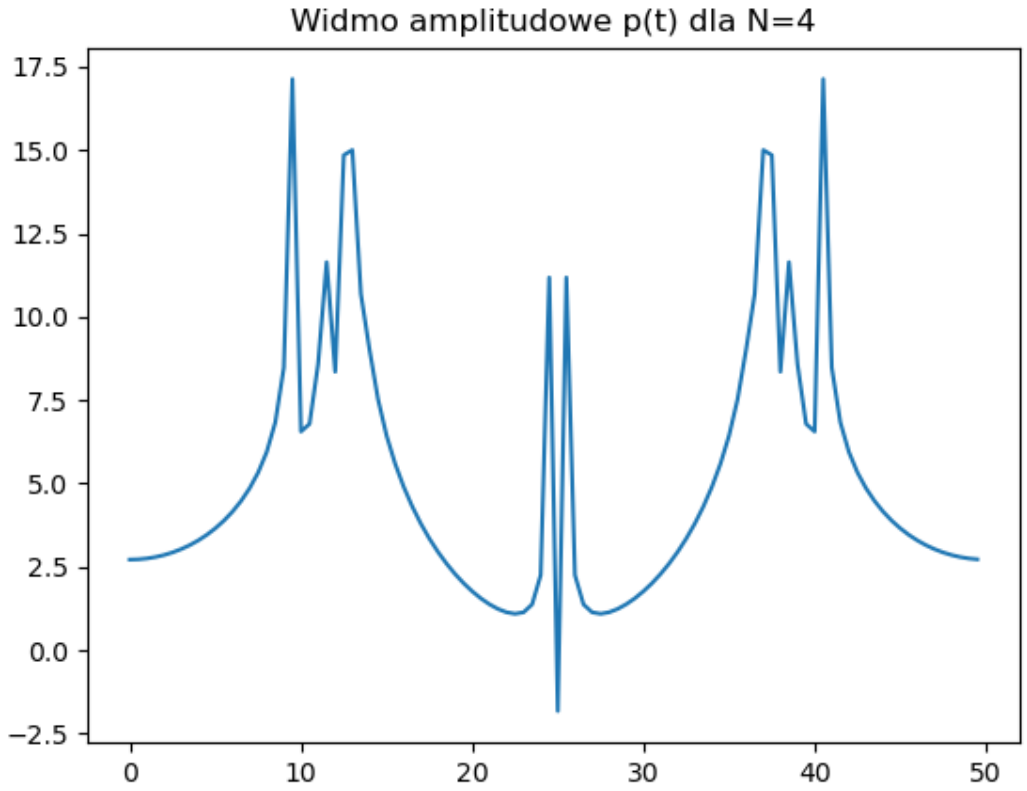
```
def v(x):
    vt = []
    for t in x:
        if (t < 0.22) and (t >= 0):
            vt.append((1 - 7 * t) * np.sin((2 * np.pi * t * 10) / (t + 0.04)))
        if (t >= 0.22) and (t < 0.7):
            vt.append(0.63 * t * np.sin(125 * t))
        if (t <= 1) and (t >= 0.7):
            vt.append((t ** (-0.662)) + 0.77 * np.sin(8 * t))
    return vt

plt.plot(skala(DFT(v(t1)), 50), M_pri(M(DFT(v(t1)))))
plt.title('Widmo amplitudowe v(t)')
plt.show()
```

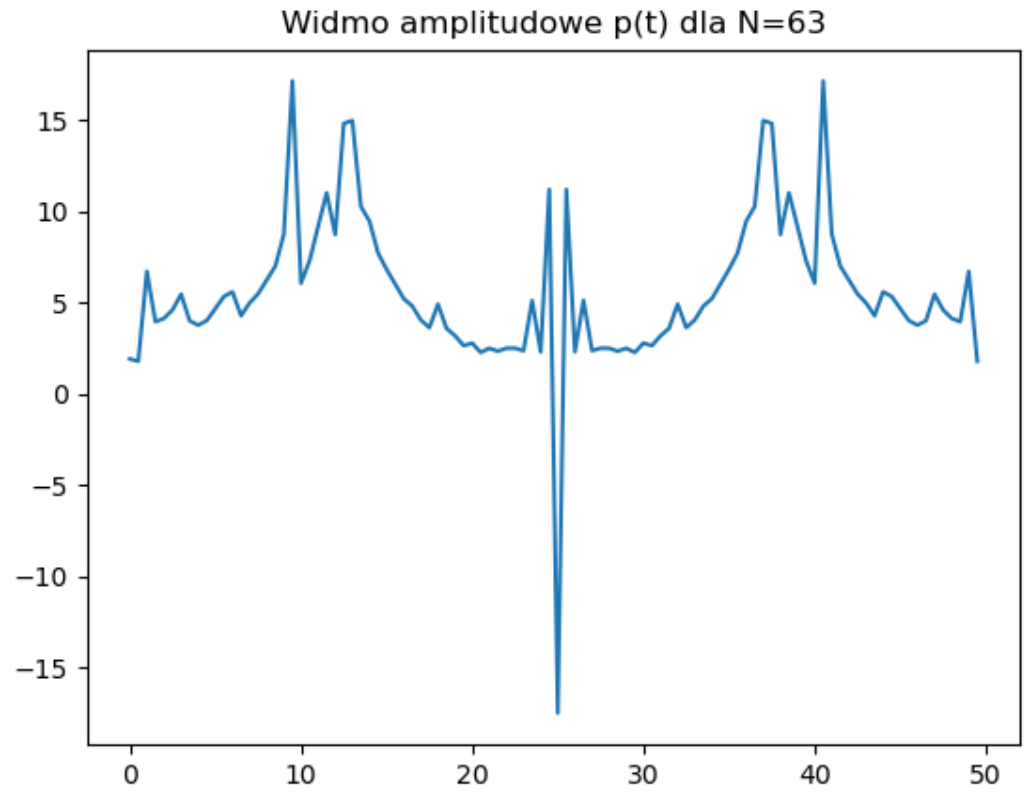


Widmo amplitudowe p(t) dla N= 2, 4, 63

```
def p(t, N):
    pt = 0
    for n in np.arange(1, N):
        pt += ((np.cos(12 * t * n ** 2) + np.cos(16 * t * n)) / n ** 2)
    return pt

plt.plot(skala(DFT(p(t, N)), 50), M_pri(M(DFT(p(t, N)))))
plt.title('Widmo amplitudowe p(t) dla N=63')
plt.show()
```




Widmo amplitudowe p(t) dla N=4
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
0
10
20
30
40
50


Widmo amplitudowe p(t) dla N=63
15
10
5
0
−5
−10
−15
0
10
20
30
40
50

Zadanie 4.

Funkcja Odwrotna Dyskretna Transformata Fouriera

```
def IDFT(n):
    xk = []
    for i in range(len(n)):
        temp = 0
        for j in range(len(n)):
            temp += np.real(n[j] * np.exp(1j * 2 * np.pi / len(n)) ** (i * j))
        xk.append(temp / len(n))
    return xk

plt.plot(n, IDFT(DFT(Ton(a, f, fi, n))))
plt.title('Odwrotna Transformata dla Tonu Prostego')
plt.show()
```