



Sprawozdanie z laboratorium
Przetwarzanie Sygnałów i Obrazów

Ćwiczenie numer: 3

Temat: Dyskretne przekształcenie Fouriera

Wykonujący ćwiczenie:

- Zaborowska Magda
- Wójtowicz Patryk

Studia dzienne I stopnia

Kierunek: Informatyka

Semestr: III

Grupa zajęciowa: PS 12

Prowadzący ćwiczenie:

mgr inż. Dawid Najda

.....

OCENA

Data wykonania ćwiczenia

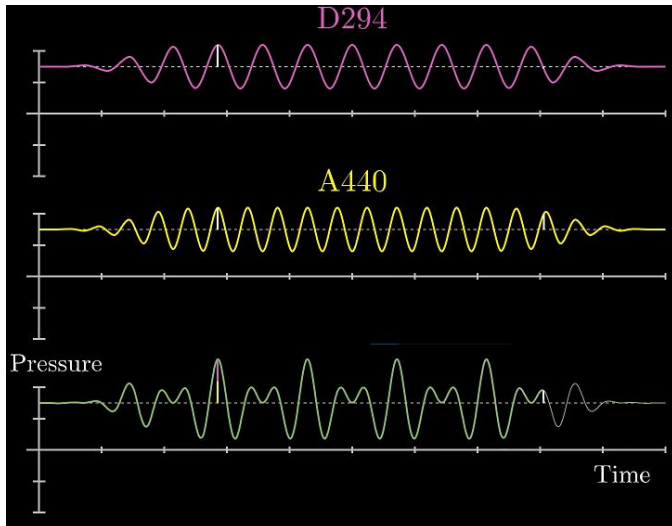
05.11.2021r.

.....

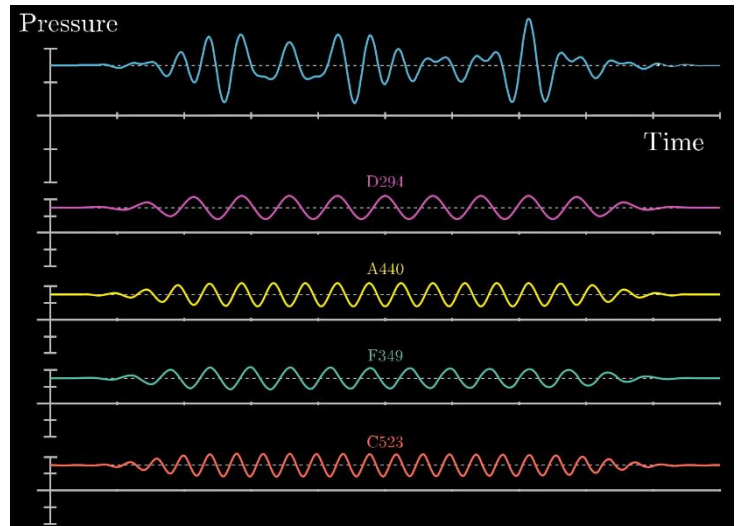
Data i podpis prowadzącego

Teoria

Mając dwa sygnały jesteśmy w stanie je zsumować tworząc nowy sygnał. Co jednak gdybyśmy chcieli się dowiedzieć z jakich sygnałów stworzony jest nasz sygnał pierwotny?



Rysunek 1 Sumowanie sygnałów okresowych.



Rysunek 2 Wydobycie z sygnału nieokresowego jego składowych sygnałów okresowych.

Służy do tego szereg Fouriera. Rozkłada on funkcję okresową na szereg funkcji trygonometrycznych. Zakładając, że funkcja $f(x)$ jest okresowa, a jej dziedziną jest zbiór liczb rzeczywistych, okres funkcji wynosi T oraz jest całkowalna w przedziale $-\frac{T}{2}; \frac{T}{2}$, szereg ma postać:

$$S(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2n\pi}{T}x\right) + b_n \sin\left(\frac{2n\pi}{T}x\right) \right)$$

Równanie 1 Wzór na szereg Fouriera.

Natomiast współczynniki a_n i b_n można obliczyć w sposób:

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos\left(\frac{2n\pi}{T}x\right) dx, \quad b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin\left(\frac{2n\pi}{T}x\right) dx,$$

$$a_0 = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) dx$$

Równanie 2 Wzory na współczynniki a_n , b_n .

Sumując ze sobą kolejne wartości obliczone w nawiasie w drugiej części wzoru $S(x)$, otrzymamy coraz dokładniejszy wykres zadanej funkcji. Natomiast zapisując je, otrzymamy rozkład na składowe naszej funkcji.

Transformacja Fouriera służy do przekształcenia funkcji ciągłej o dziedzinie czasu w funkcję ciągłą o dziedzinie częstotliwości. Jej wynikiem jest transformata Fouriera, która pokazuje z jakich częstotliwości składa się pierwotna funkcja. Umożliwia ona opisanie danej funkcji za pomocą ciągłej superpozycji częstotliwości. Dla funkcji $f(x)$ transformację Fouriera określamy wzorem:

$$F(s) = \int_{-\infty}^{+\infty} f(x) \cdot e^{-2\pi i x s} dx$$

Równanie 3. i to jednostka urojona, s to częstotliwość, x to czas.

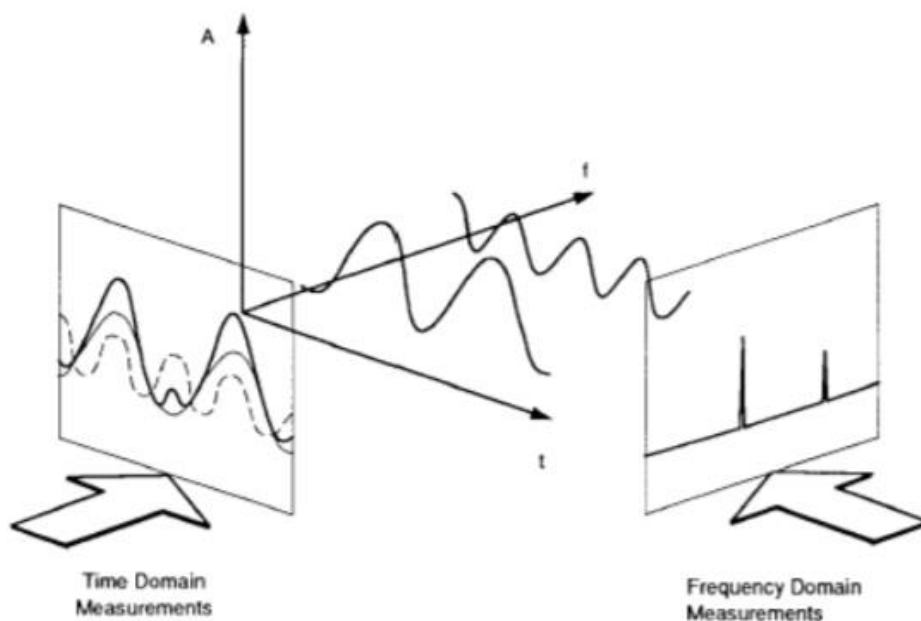
Dyskretna transformata Fouriera jest transformatą Fouriera, ale wyznaczoną dla sygnału dyskretnego, czyli próbkowanego. Jest ona określona następującym wzorem:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk} \quad k = 0, \dots, N-1$$

Równanie 4. N to ilość próbek, k to indeks próbki, x_n to wartość próbki, i to jednostka urojona.

Złożoność obliczania DFT z powyższego wzoru wynosi $O(N^2)$. Wynika stąd, że nie jest to wydajny sposób, dlatego powstała szybka transformata Fouriera FFT. Złożoność A algorytmu FFT jest liniowo-logarytmiczna ($O(N \log_2 N)$). Jest to znacząca różnica. Oryginalna funkcja FFT liczy dla ilości próbek równej $N=2^k$ czyli dla potęg dwójki.

Za pomocą transformacji Fouriera otrzymamy widmo sygnału. Jest to reprezentacja sygnału w dziedzinie częstotliwości. Jest to zarówno wykres transformaty Fouriera jak i sama transformata. Widmo sygnału jest reprezentowane jako prążki o wysokości odpowiadającej amplitudzie poszczególnych składowych sygnału.



Rysunek 3 Po lewej stronie sygnały składowe, po prawej widmo.

Zadanie 1

Treść zadania:

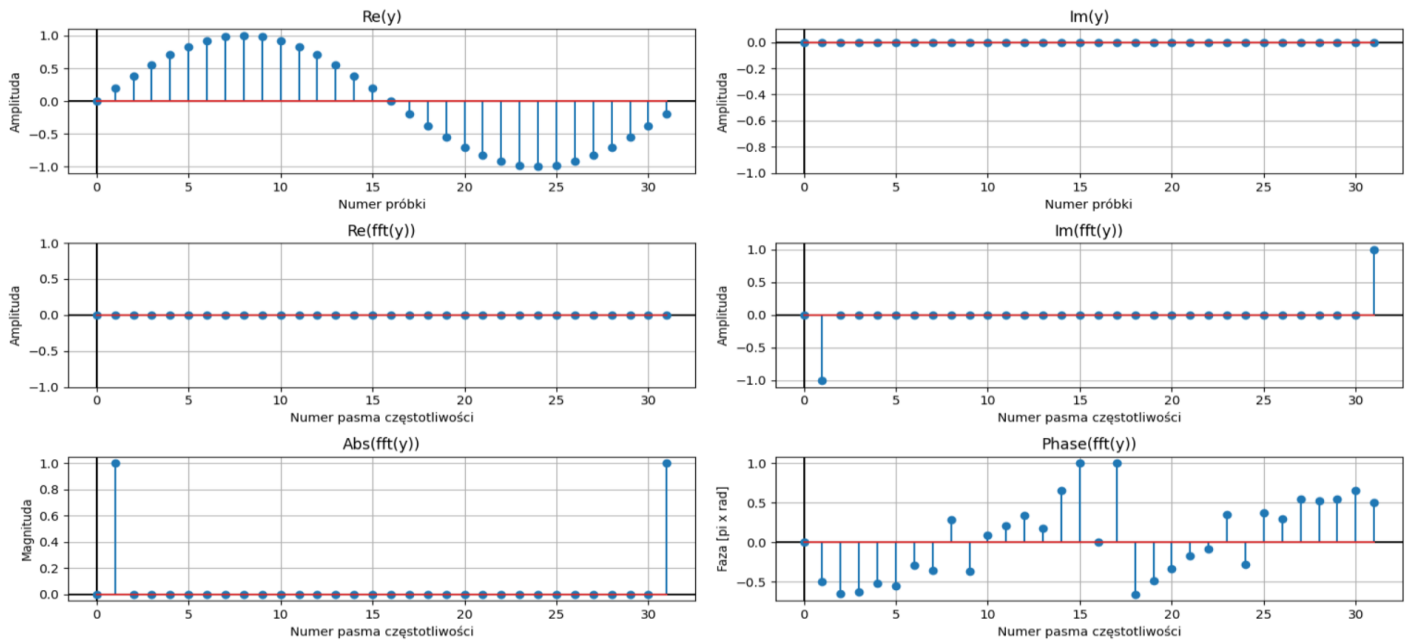
- Wygeneruj dokładnie 1 okres fali sinusoidalnej (32 lub 64 próbek).
- Sporządź wykresy sinusoidy i jej transformaty Fouriera (funkcja `fft`) w jednym oknie (część rzeczywistą - real, urojoną - imag, moduł - absolute i kąt - angle).
- Co możesz powiedzieć o symetrii widma zespolonego?

A) Realizacja w kodzie:

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  n = 32 #ilość próbek
5  x = np.arange(n) # tablica w n komurkami
6  y = np.sin(2*np.pi*x/n) # wzór funkcji bazowej
7  z = 2*np.fft.fft(y)/n #transformata Fouriera dla funkcji y
8
9  plt.figure(figsize=(8,6),tight_layout=1) # tworzenie figury na której wyświetlą się wykresy
10
11 plt.subplot(3,2,1) #oznaczenie położenie wykresu
12 plt.title('Re(y)') #tytuł
13 plt.xlabel('Numer próbek') #nazwa osi OX
14 plt.ylabel('Amplituda') #nazwa osi OY
15 plt.axhline(y=0,color = "k") #pogrubienie osi OX
16 plt.axvline(x=0,color = "k") #pogrubienie osi OY
17 plt.stem(np.real(y),use_line_collection=10) #stworzenie wykresy typu stem
18 plt.grid(True,which='both') #włączenie siatki
```

```
20 # Tworzenie części zespolonej wykresu y
21 plt.subplot(3,2,2)
22 plt.title('Im(y)')
23 plt.xlabel('Numer próbek')
24 plt.ylabel('Amplituda')
25 plt.axhline(y=0,color = "k")
26 plt.axvline(x=0,color = "k")
27 plt.ylim(-1,0.1)
28 plt.stem(np.imag(y),use_line_collection=1)
29 plt.grid(True,which='both')
30
31 # Tworzenie wykresu części rzeczywistej FFT(y)
32 plt.subplot(3,2,3)
33 plt.title('Re(fft(y))')
34 plt.xlabel('Numer pasma częstotliwości')
35 plt.ylabel('Amplituda')
36 plt.axhline(y=0,color = "k")
37 plt.axvline(x=0,color = "k")
38 plt.ylim(-1,1)
39 plt.stem(np.real(z),use_line_collection=1)
40 plt.grid(True,which='both')
41
42 # Tworzenie wykresu części zespolonej FFT(y)
43 plt.subplot(3,2,4)
44 plt.title('Im(fft(y))')
45 plt.xlabel('Numer pasma częstotliwości')
46 plt.ylabel('Amplituda')
47 plt.axhline(y=0,color = "k")
48 plt.axvline(x=0,color = "k")
49 plt.stem(np.imag(z),use_line_collection=1)
50 plt.grid(True,which='both')
51
52 # Tworzenie wykresu modułu FFT(y)
53 plt.subplot(3,2,5)
54 plt.title('Abs(fft(y))')
55 plt.xlabel('Numer pasma częstotliwości')
56 plt.ylabel('Magnituda')
57 plt.axhline(y=0,color = "k")
58 plt.axvline(x=0,color = "k")
59 plt.stem(np.abs(z),use_line_collection=1)
60 plt.grid(True,which='both')
61
62 # Tworzenie wykresu fazy FFT(y)
63 plt.subplot(3,2,6)
64 plt.title('Phase(fft(y))')
65 plt.xlabel('Numer pasma częstotliwości')
66 plt.ylabel('Faza [pi x rad]')
67 plt.axhline(y=0,color = "k")
68 plt.axvline(x=0,color = "k")
69 plt.stem(np.angle(z)/np.pi,use_line_collection=1)
70 plt.grid(True,which='both')
71
72 plt.show()
```

Wynik:



Dla transformaty Fouriera funkcji $\sin(x)$ w części urojonej zauważalna jest symetria względem punktu $(16, 0)$, natomiast dla modułu funkcji $\text{FFT}(\sin x)$ widoczna jest symetria względem osi X w punkcie $x=16$.

Zadanie 2

Treść zadania:

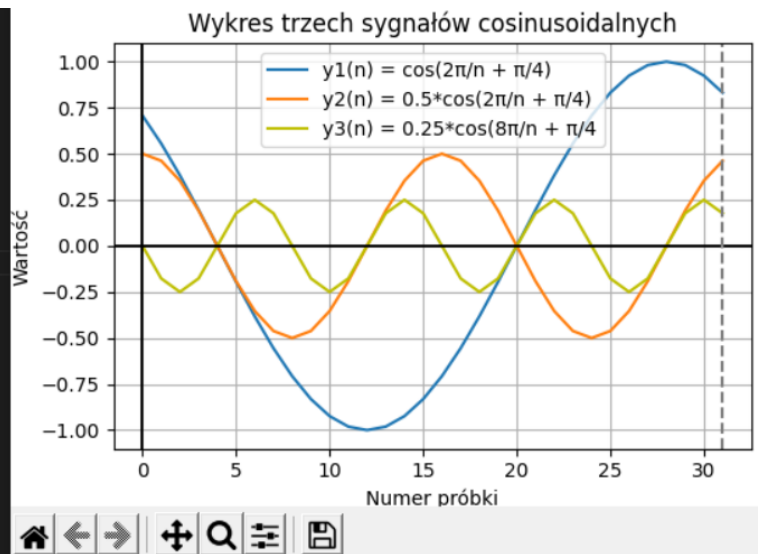
Wygeneruj następujące sygnały:

$$y_1[n] = \cos(2\pi n/N + \pi/4), \quad y_2[n] = 0.50 \cos(4\pi n/N), \quad y_3[n] = 0.25 \cos(8\pi n/N + \pi/2)$$

1. Wyznacz ich transformaty Fouriera.
2. Obliczenia powtórz dla sygnału $y_4 = y_1 + y_2 + y_3$.
3. Jaki jest związek pomiędzy amplitudą, fazą, liczbą okresów poszczególnych sygnałów a wartościami widma zespolonego?
4. Jak zachowuje się funkcja fft w stosunku do sumy sygnałów?

Realizacja założeń:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 n = 32
5 x = np.arange(n)
6 y1 = np.cos(2*np.pi*x/n+(np.pi/4))
7 y2 = 0.5*np.cos(4*np.pi*x/n)
8 y3 = 0.25*np.cos(8*np.pi*x/n+(np.pi/2))
9
10 plt.figure(figsize=(8,6))
11
12 plt.plot(x,y1)
13 plt.plot(x,y2)
14 plt.plot(x,y3,'y')
15 plt.grid(True,which='both')
16 plt.title('Wykres trzech sygnałów cosinusoidalnych')
17 plt.xlabel('Numer próbki')
18 plt.ylabel('Wartość')
19 plt.axhline(y=0,color = "k")
20 plt.axvline(x=0,color = "k")
21 plt.axvline(x=n-1,color = "grey",linestyle = "--")
22 plt.legend(['y1(n) = cos(2π/n + π/4)', 'y2(n) = 0.5*cos(2π/n + π/4)', 'y3(n) = 0.25*cos(8π/n + π/4)'])
23
24 plt.show()
```

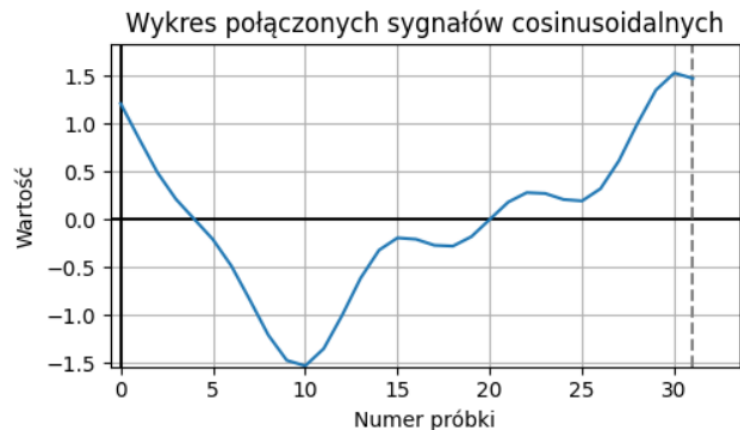


Stworzenie zsumowanego sygnału:

```

25
26 # Podpunkt 2 , obliczenia powtórz dla sygnału y4 = y1 + y2 + y3
27 y4 = y1 + y2 + y3
28
29 plt.figure(figsize=(8,6))
30 plt.grid(True,which='both')
31 plt.title('Wykres połączonych sygnałów cosinusoidalnych')
32 plt.xlabel('Numer próbki')
33 plt.ylabel('Wartość')
34 plt.axhline(y=0,color = "k")
35 plt.axvline(x=0,color = "k")
36 plt.axvline(x=n-1,color = "grey",linestyle = "--")
37 plt.plot(x,y4)
38 plt.show()
39
40 # Podpunkt 4 - obliczenia transformacji Fouriera dla sygnałów y1, y2, y3, y4

```

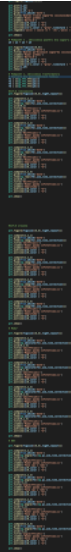


Obliczenie transformat Fouriera oraz tworzenie wykresów części rzeczywistej widma sygnałów:

```

40 # Podpunkt 1, obliczenia transformatory
41 # Fouriera dla poszczególnych funkcji
42 f1 = 2*np.fft.fft(y1)/n
43 f2 = 2*np.fft.fft(y2)/n
44 f3 = 2*np.fft.fft(y3)/n
45 f4 = 2*np.fft.fft(y4)/n
46
47 #Część rzeczywista
48 plt.figure(figsize=(8,6),tight_layout=1)
49
50 plt.subplot(2,2,1)
51 plt.grid(True,which='both')
52 plt.stem(np.real(f1),use_line_collection=1)
53 plt.ylim(0,1)
54 plt.title('Re(fft(y1))')
55 plt.xlabel('Numer pasma częstotliwości')
56 plt.ylabel('Amplituda')
57 plt.axhline(y=0,color = "k")
58 plt.axvline(x=0,color = "k")
59
60 plt.subplot(2,2,2)
61 plt.grid(True,which='both')
62 plt.stem(np.real(f2),use_line_collection=1)
63 plt.ylim(0,1)
64 plt.title('Re(fft(y2))')
65 plt.xlabel('Numer pasma częstotliwości')
66 plt.ylabel('Amplituda')
67 plt.axhline(y=0,color = "k")
68 plt.axvline(x=0,color = "k")

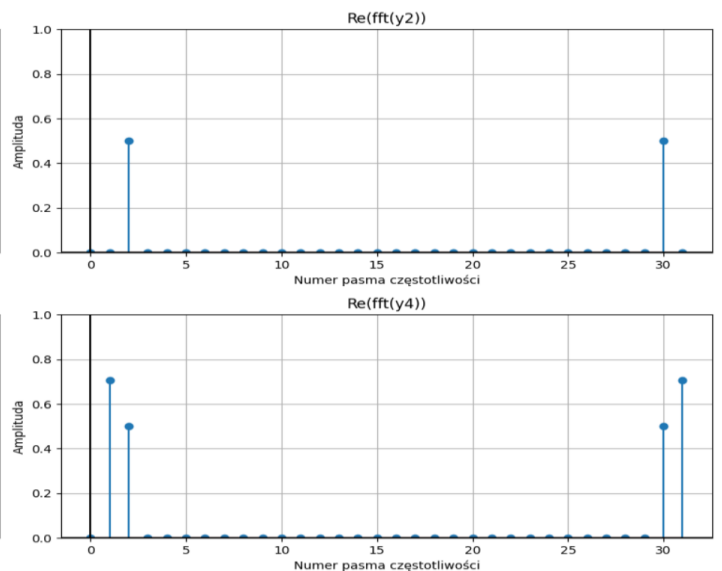
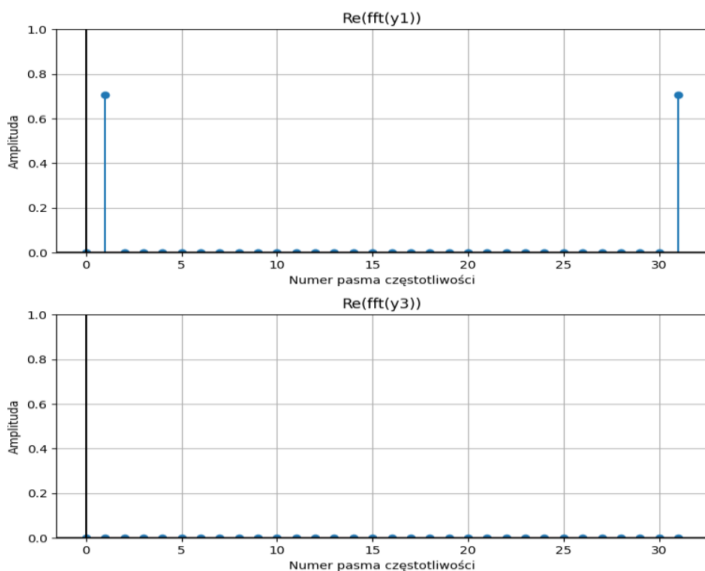
```



```

70 plt.subplot(2,2,3)
71 plt.grid(True,which='both')
72 plt.stem(np.real(f3),use_line_collection=1)
73 plt.ylim(0,1)
74 plt.title('Re(fft(y3))')
75 plt.xlabel('Numer pasma częstotliwości')
76 plt.ylabel('Amplituda')
77 plt.axhline(y=0,color = "k")
78 plt.axvline(x=0,color = "k")
79
80 plt.subplot(2,2,4)
81 plt.grid(True,which='both')
82 plt.stem(np.real(f4),use_line_collection=1)
83 plt.ylim(0,1)
84 plt.title('Re(fft(y4))')
85 plt.xlabel('Numer pasma częstotliwości')
86 plt.ylabel('Amplituda')
87 plt.axhline(y=0,color = "k")
88 plt.axvline(x=0,color = "k")
89
90 plt.show()
91
92
93
94
95
96
97
98

```

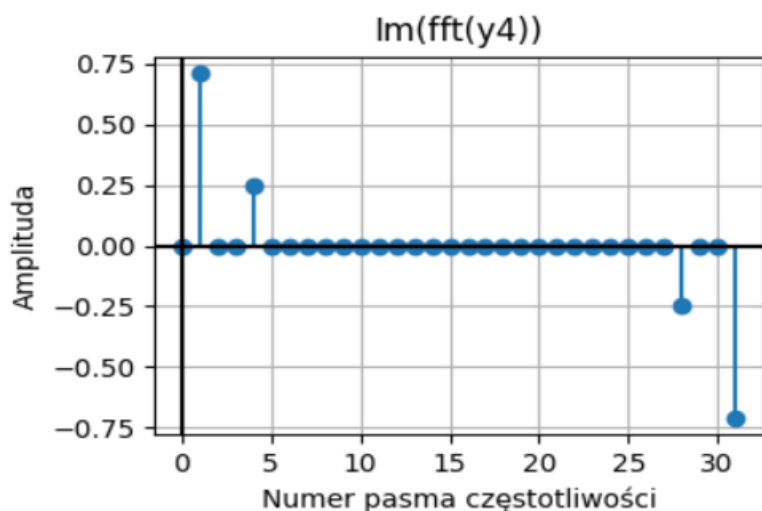
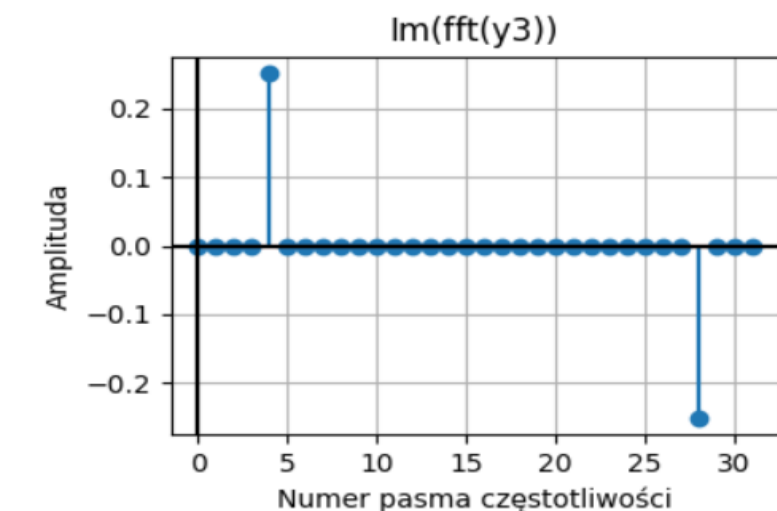
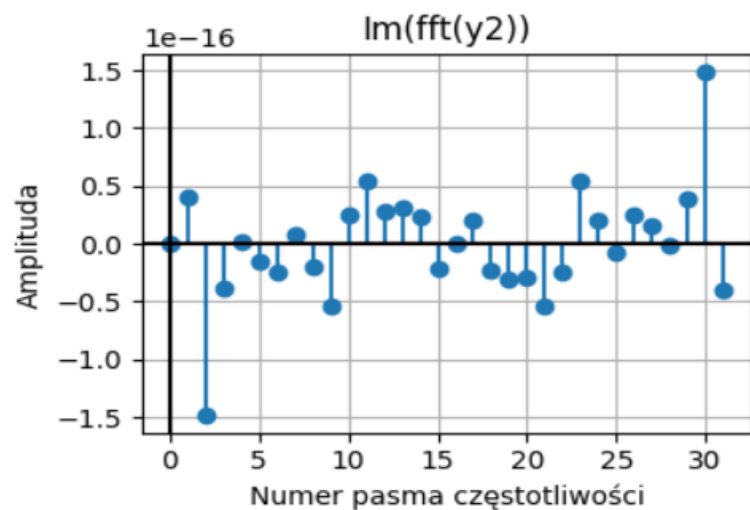
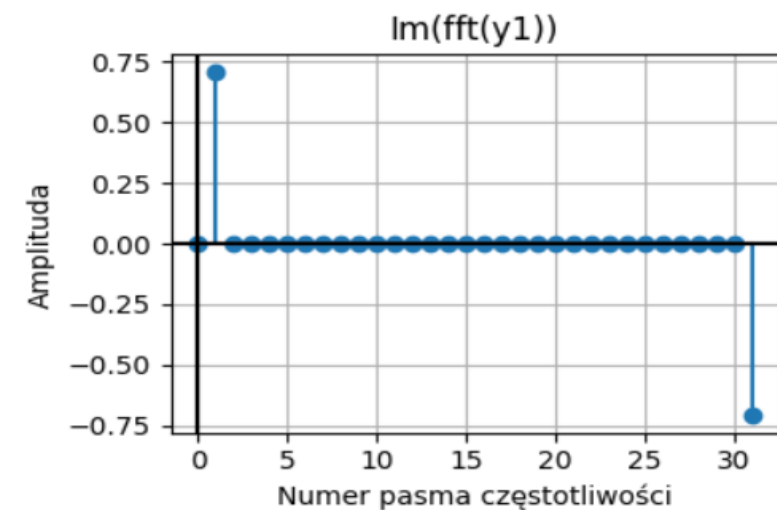


Tworzenie wykresów części urojonej widma sygnałów:

```

92 #Część urojona
93
94 plt.figure(figsize=(8,6),tight_layout=1)
95
96 plt.subplot(2,2,1)
97 plt.grid(True,which='both')
98 plt.stem(np.imag(f1),use_line_collection=1)
99 plt.title('Im(fft(y1))')
100 plt.xlabel('Numer pasma częstotliwości')
101 plt.ylabel('Amplituda')
102 plt.axhline(y=0,color = "k")
103 plt.axvline(x=0,color = "k")
104
105 plt.subplot(2,2,2)
106 plt.grid(True,which='both')
107 plt.stem(np.imag(f2),use_line_collection=1)
108 plt.title('Im(fft(y2))')
109 plt.xlabel('Numer pasma częstotliwości')
110 plt.ylabel('Amplituda')
111 plt.axhline(y=0,color = "k")
112 plt.axvline(x=0,color = "k")
113
114
115
116 plt.subplot(2,2,3)
117 plt.grid(True,which='both')
118 plt.stem(np.imag(f3),use_line_collection=1)
119 plt.title('Im(fft(y3))')
120 plt.xlabel('Numer pasma częstotliwości')
121 plt.ylabel('Amplituda')
122 plt.axhline(y=0,color = "k")
123 plt.axvline(x=0,color = "k")
124
125 plt.subplot(2,2,4)
126 plt.grid(True,which='both')
127 plt.stem(np.imag(f4),use_line_collection=1)
128 plt.title('Im(fft(y4))')
129 plt.xlabel('Numer pasma częstotliwości')
130 plt.ylabel('Amplituda')
131 plt.axhline(y=0,color = "k")
132 plt.axvline(x=0,color = "k")
133
134 plt.show()

```



Tworzenie wykresów modułu widma sygnałów:

```

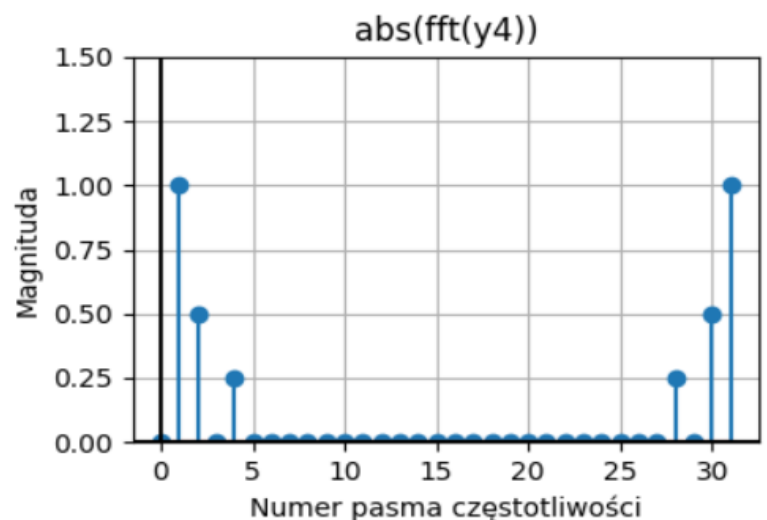
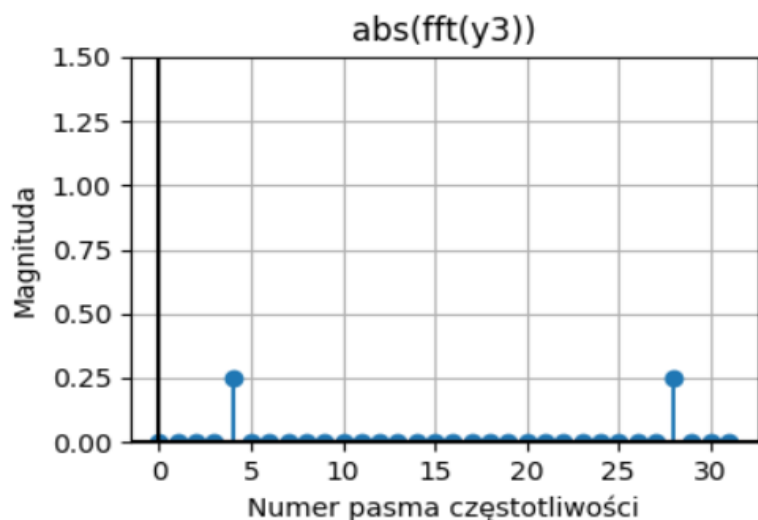
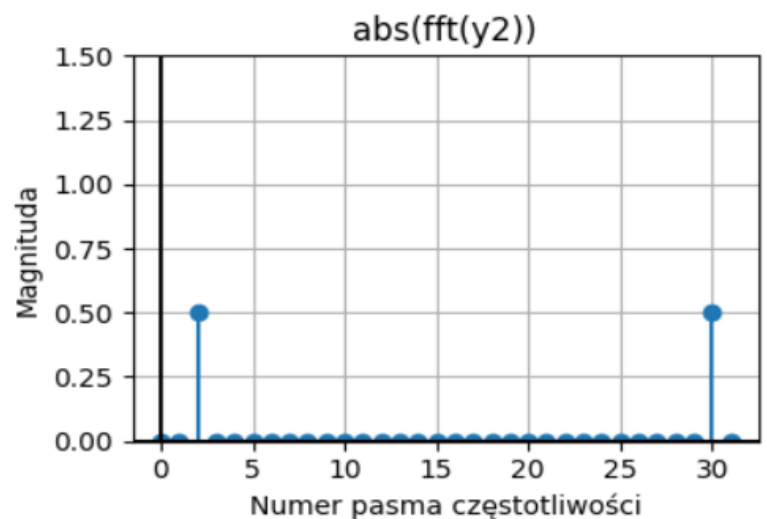
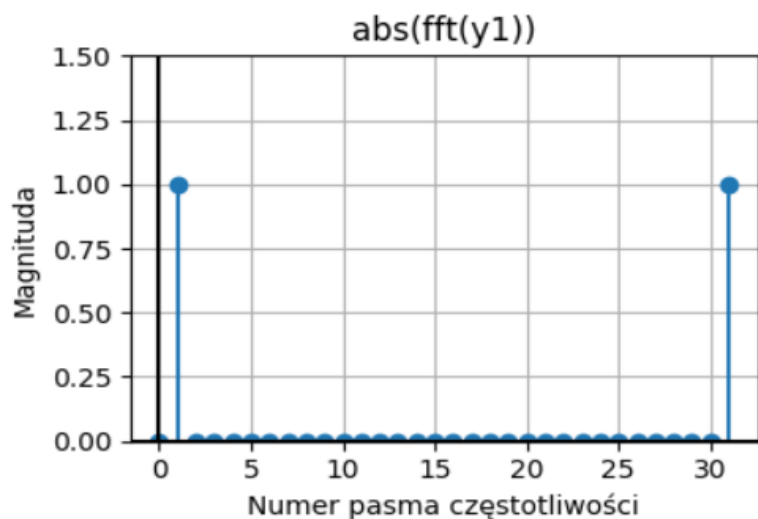
140 # Moduł
141
142 plt.figure(figsize=(8,6),tight_layout=1)
143
144 plt.subplot(2,2,1)
145 plt.grid(True,which='both')
146 plt.stem(np.absolute(f1),use_line_collection=1)
147 plt.ylim(0,1.5)
148 plt.title('abs(fft(y1))')
149 plt.xlabel('Numer pasma częstotliwości')
150 plt.ylabel('Magnituda')
151 plt.axhline(y=0,color = "k")
152 plt.axvline(x=0,color = "k")
153
154 plt.subplot(2,2,2)
155 plt.grid(True,which='both')
156 plt.stem(np.absolute(f2),use_line_collection=1)
157 plt.ylim(0,1.5)
158 plt.title('abs(fft(y2))')
159 plt.xlabel('Numer pasma częstotliwości')
160 plt.ylabel('Magnituda')
161 plt.axhline(y=0,color = "k")
162 plt.axvline(x=0,color = "k")
163

```

```

164
165
166 plt.subplot(2,2,3)
167 plt.grid(True,which='both')
168 plt.stem(np.absolute(f3),use_line_collection=1)
169 plt.ylim(0,1.5)
170 plt.title('abs(fft(y3))')
171 plt.xlabel('Numer pasma częstotliwości')
172 plt.ylabel('Magnituda')
173 plt.axhline(y=0,color = "k")
174 plt.axvline(x=0,color = "k")
175
176 plt.subplot(2,2,4)
177 plt.grid(True,which='both')
178 plt.stem(np.absolute(f4),use_line_collection=1)
179 plt.ylim(0,1.5)
180 plt.title('abs(fft(y4))')
181 plt.xlabel('Numer pasma częstotliwości')
182 plt.ylabel('Magnituda')
183 plt.axhline(y=0,color = "k")
184 plt.axvline(x=0,color = "k")
185
186 plt.show()
187

```



Tworzenie wykresów fazy widma sygnałów:

```

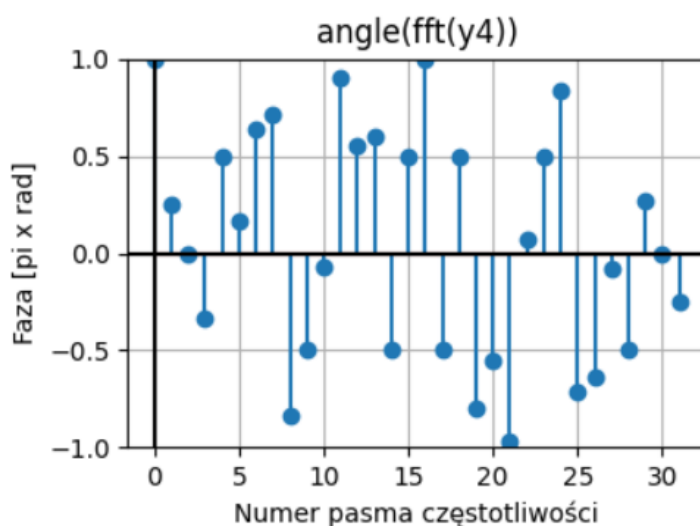
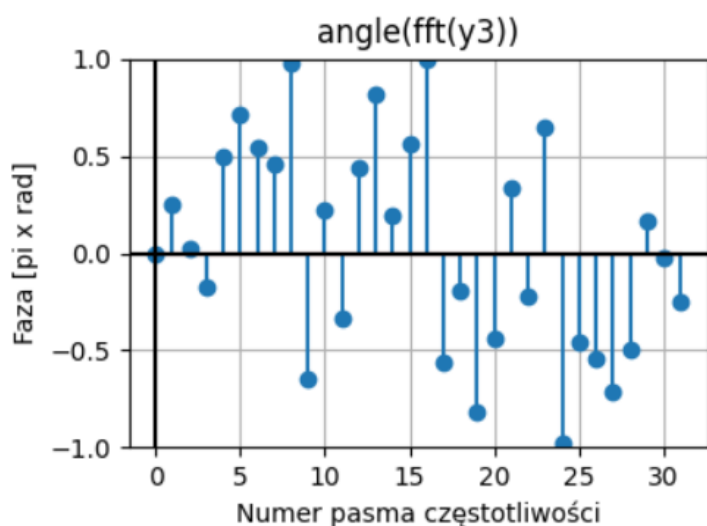
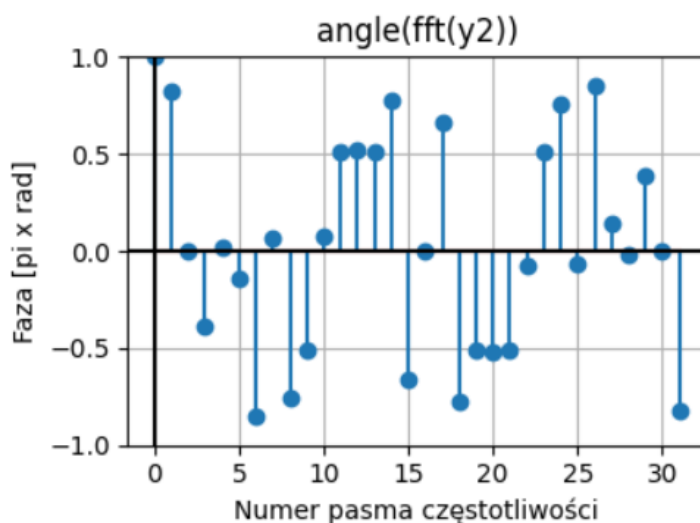
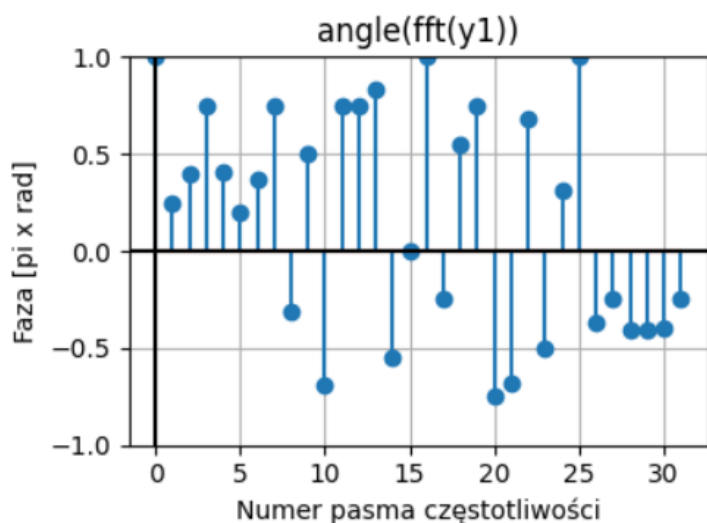
191 # Kąt
192
193 plt.figure(figsize=(8,6),tight_layout=1)
194
195 plt.subplot(2,2,1)
196 plt.grid(True,which='both')
197 plt.stem(np.angle(f1)/np.pi,use_line_collection=1)
198 plt.ylim(-1,1)
199 plt.title('angle(fft(y1))')
200 plt.xlabel('Numer pasma częstotliwości')
201 plt.ylabel('Faza [pi x rad]')
202 plt.axhline(y=0,color = "k")
203 plt.axvline(x=0,color = "k")
204
205 plt.subplot(2,2,2)
206 plt.grid(True,which='both')
207 plt.stem(np.angle(f2)/np.pi,use_line_collection=1)
208 plt.ylim(-1,1)
209 plt.title('angle(fft(y2))')
210 plt.xlabel('Numer pasma częstotliwości')
211 plt.ylabel('Faza [pi x rad]')
212 plt.axhline(y=0,color = "k")
213 plt.axvline(x=0,color = "k")
214

```

```

215
216 |
217 plt.subplot(2,2,3)
218 plt.grid(True,which='both')
219 plt.stem(np.angle(f3)/np.pi,use_line_collection=1)
220 plt.ylim(-1,1)
221 plt.title('angle(fft(y3))')
222 plt.xlabel('Numer pasma częstotliwości')
223 plt.ylabel('Faza [pi x rad]')
224 plt.axhline(y=0,color = "k")
225 plt.axvline(x=0,color = "k")
226
227 plt.subplot(2,2,4)
228 plt.grid(True,which='both')
229 plt.stem(np.angle(f4)/np.pi,use_line_collection=1)
230 plt.ylim(-1,1)
231 plt.title('angle(fft(y4))')
232 plt.xlabel('Numer pasma częstotliwości')
233 plt.ylabel('Faza [pi x rad]')
234 plt.axhline(y=0,color = "k")
235 plt.axvline(x=0,color = "k")
236
237 plt.show()

```



Zadanie 3

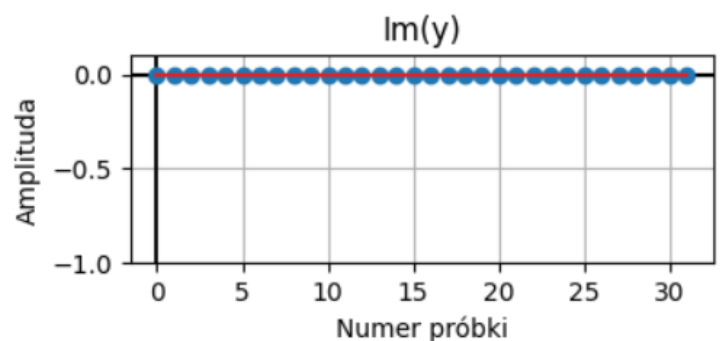
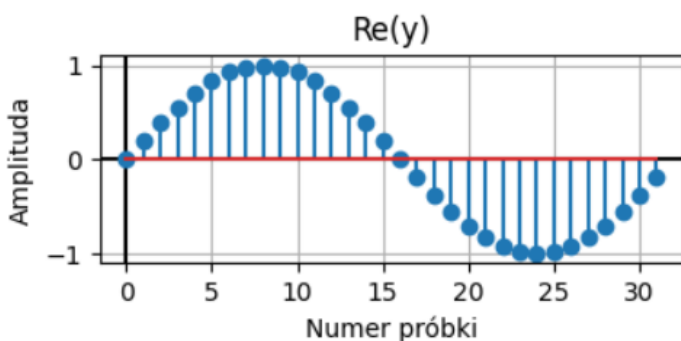
Treść zadania:

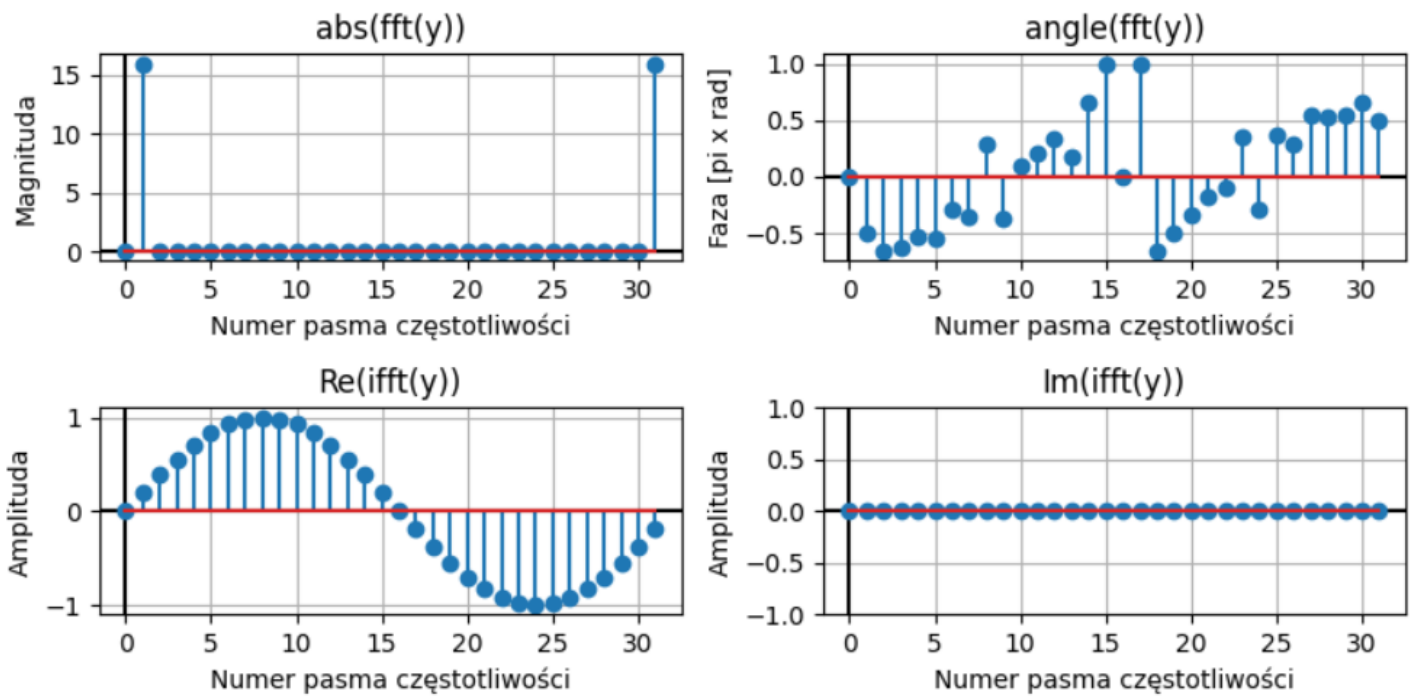
- Wyznacz odwrotną transformatę Fouriera (funkcja `ifft`) dla widm zespolonych FFT otrzymanych w zadaniach 3.1 i 3.2.
- Jak zinterpretujesz otrzymane wyniki?

Realizacja odwrotnej transformaty dla zadanie 3.1:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 #zadanie 1
5 n = 32
6 x = np.arange(n)
7 y = np.sin(2*np.pi*x/n)
8 yfft = np.fft.fft(y) # fft - szybka transformata
9 yifft = np.fft.ifft(yfft) # ifft - odwrotna transformata
10 plt.figure(figsize=(8,6),tight_layout=1)
11 plt.subplot(3,2,1)
12 plt.title('Re(y)')
13 plt.xlabel('Numer próbek')
14 plt.ylabel('Amplituda')
15 plt.axhline(y=0,color = "k")
16 plt.axvline(x=0,color = "k")
17 plt.stem(np.real(y),use_line_collection=1)
18 plt.grid(True,which='both')
19
20 plt.subplot(3,2,2)
21 plt.title('Im(y)')
22 plt.xlabel('Numer próbek')
23 plt.ylabel('Amplituda')
24 plt.axhline(y=0,color = "k")
25 plt.axvline(x=0,color = "k")
26 plt.ylim(-1,0.1)
27 plt.stem(np.imag(y),use_line_collection=1)
28 plt.grid(True,which='both')
29
30 plt.subplot(3,2,3)
31 plt.title('abs(fft(y))')
32 plt.xlabel('Numer pasma częstotliwości')
33 plt.ylabel('Magnituda')
34 plt.axhline(y=0,color = "k")
35 plt.axvline(x=0,color = "k")
36 plt.stem(np.abs(yfft),use_line_collection=1)
37 plt.grid(True,which='both')
38
39
40
41
42 plt.subplot(3,2,4)
43 plt.title('angle(fft(y))')
44 plt.xlabel('Numer pasma częstotliwości')
45 plt.ylabel('Faza [pi x rad]')
46 plt.axhline(y=0,color = "k")
47 plt.axvline(x=0,color = "k")
48 plt.stem(np.angle(yfft)/np.pi,use_line_collection=1)
49 plt.grid(True,which='both')
50
51 plt.subplot(3,2,5)
52 plt.title('Re(ifft(y))')
53 plt.xlabel('Numer pasma częstotliwości')
54 plt.ylabel('Amplituda')
55 plt.axhline(y=0,color = "k")
56 plt.axvline(x=0,color = "k")
57 plt.ylim(-1.1,1.1)
58 plt.stem(np.real(yifft),use_line_collection=1)
59 plt.grid(True,which='both')
60
61 plt.subplot(3,2,6)
62 plt.title('Im(ifft(y))')
63 plt.xlabel('Numer pasma częstotliwości')
64 plt.ylabel('Amplituda')
65 plt.axhline(y=0,color = "k")
66 plt.axvline(x=0,color = "k")
67 plt.ylim(-1,1)
68 plt.stem(np.imag(yifft),use_line_collection=1)
69 plt.grid(True,which='both')
70
71 plt.show()
72
73
74
75
```

Wynik:





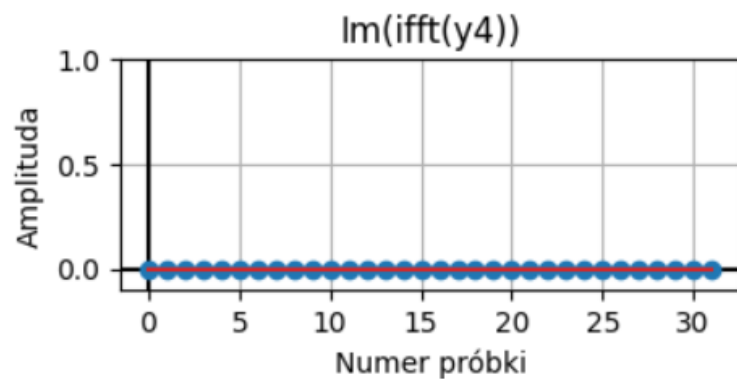
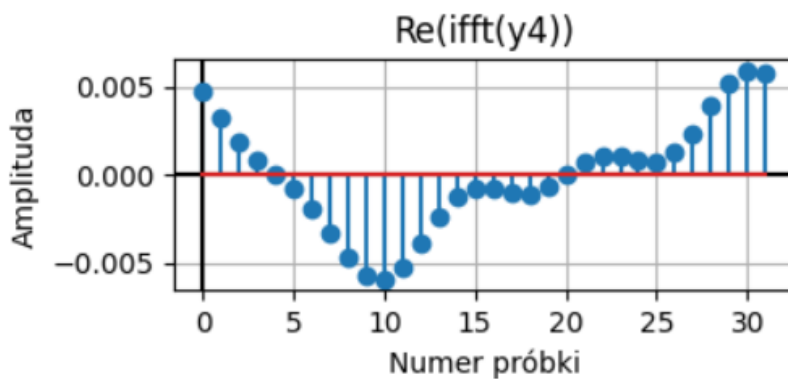
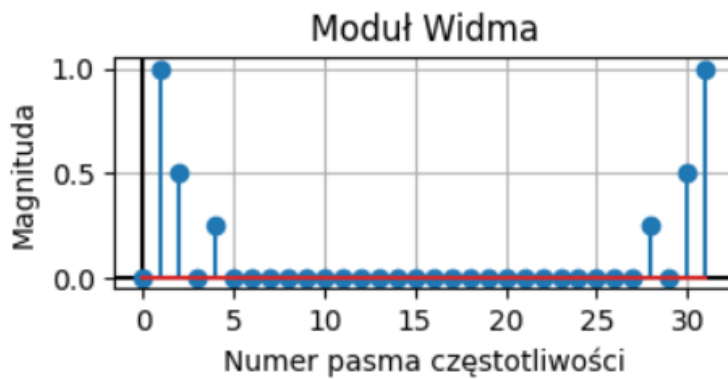
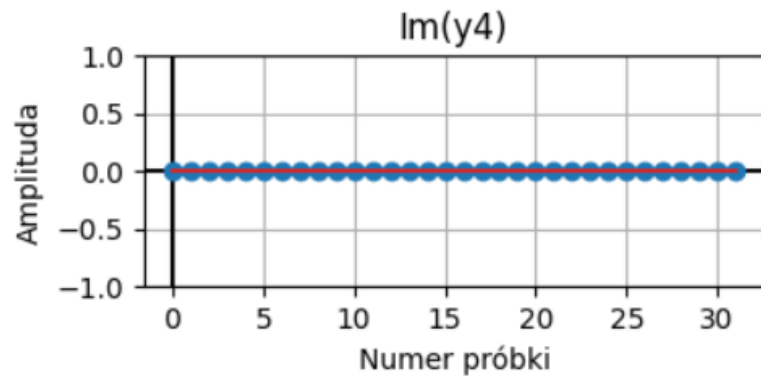
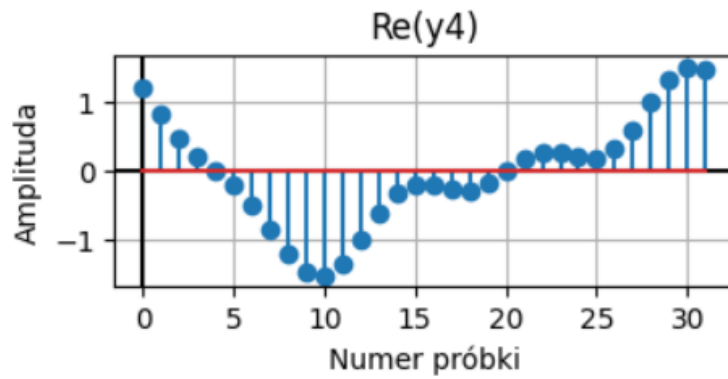
Realizacja odwrotnej transformaty dla zadanie 3.2:

```

78 #zadanie 2
79 y1 = np.cos(2*np.pi*x/n+(np.pi/4))
80 y2 = 0.5*np.cos(4*np.pi*x/n)
81 y3 = 0.25*np.cos(8*np.pi*x/n+(np.pi/2))
82 y4=y1+y2+y3
83
84 f4 = 2*np.fft.fft(y4)/n
85 if4 = 2*np.fft.ifft(f4)/n
86
87 plt.figure(figsize=(8,6),tight_layout=1)
88 plt.subplot(3,2,1)
89 plt.title('Re(y4)')
90 plt.xlabel('Numer próbek')
91 plt.ylabel('Amplituda')
92 plt.axhline(y=0,color = "k")
93 plt.axvline(x=0,color = "k")
94 plt.stem(np.real(y4),use_line_collection=1)
95 plt.grid(True,which='both')
96
97 plt.subplot(3,2,2)
98 plt.title('Im(y4)')
99 plt.xlabel('Numer próbek')
100 plt.ylabel('Amplituda')
101 plt.axhline(y=0,color = "k")
102 plt.axvline(x=0,color = "k")
103 plt.ylim(-1,1)
104 plt.stem(np.imag(y4),use_line_collection=1)
105 plt.grid(True,which='both')
106
107 plt.subplot(3,2,3)
108 plt.title('Moduł Widma')
109 plt.xlabel('Numer pasma częstotliwości')
110 plt.ylabel('Magnituda')
111 plt.axhline(y=0,color = "k")
112 plt.axvline(x=0,color = "k")
113 plt.stem(np.abs(f4),use_line_collection=1)
114 plt.grid(True,which='both')
115
116 plt.subplot(3,2,4)
117 plt.title('Faza Widma')
118 plt.xlabel('Numer pasma częstotliwości')
119 plt.ylabel('Faza [pi x rad]')
120 plt.axhline(y=0,color = "k")
121 plt.axvline(x=0,color = "k")
122 plt.stem(np.angle(f4)/np.pi,use_line_collection=1)
123 plt.grid(True,which='both')
124
125 plt.subplot(3,2,5)
126 plt.title('Re(ifft(y4))')
127 plt.xlabel('Numer próbek')
128 plt.ylabel('Amplituda')
129 plt.axhline(y=0,color = "k")
130 plt.axvline(x=0,color = "k")
131 plt.stem(np.real(if4),use_line_collection=1)
132 plt.grid(True,which='both')
133
134 plt.subplot(3,2,6)
135 plt.title('Im(ifft(y4))')
136 plt.xlabel('Numer próbek')
137 plt.ylabel('Amplituda')
138 plt.axhline(y=0,color = "k")
139 plt.axvline(x=0,color = "k")
140 plt.ylim(-0.1,1)
141 plt.stem(np.imag(if4),use_line_collection=1)
142 plt.grid(True,which='both')
143
144 plt.show()

```

Wynik:



Zadanie 4

Treść zadania:

- Powtórz zadanie 3.1 dla sinusoidy zespolonej $y[n] = \exp(j(\omega n + \phi))$, gdzie ω – pulsacja unormowana oraz ϕ – przesunięcie fazowe.
- Jakie są różnice (w symetrii) w stosunku do widma sygnału rzeczywistego?

Realizacja w kodzie:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 n = 32
5 x = np.arange(n)
6 y = np.exp(1j*(2*np.pi*1/n*x+np.pi/2)) #wzór funkcji
7 z = 2*np.fft.fft(y)/n #FFT dla funkcji
8
9 plt.figure(figsize=(8,6),tight_layout=1)
10 #cz. rzeczywista funkcji y
11 plt.subplot(3,2,1)
12 plt.title('Re(y)')
13 plt.xlabel('Numer próbki')
14 plt.ylabel('Amplituda')
15 plt.axhline(y=0,color = "k")
16 plt.axvline(x=0,color = "k")
17 plt.stem(np.real(y),use_line_collection=1)
18 plt.grid(True,which='both')
19 #cz. urojona funkcji y
20 plt.subplot(3,2,2)
21 plt.title('Im(y)')
22 plt.xlabel('Numer próbki')
23 plt.ylabel('Amplituda')
24 plt.axhline(y=0,color = "k")
25 plt.axvline(x=0,color = "k")
26 plt.stem(np.imag(y),use_line_collection=1)
27 plt.grid(True,which='both')
28 #cz. rzeczywista funkcji FFT
29 plt.subplot(3,2,3)
30 plt.title('Re(fft(y))')
31 plt.xlabel('Numer pasma częstotliwości')
32 plt.ylabel('Amplituda')
33 plt.axhline(y=0,color = "k")
34 plt.axvline(x=0,color = "k")
35 plt.ylim(-1,1)
36 plt.stem(np.real(z),use_line_collection=1)
37 plt.grid(True,which='both')

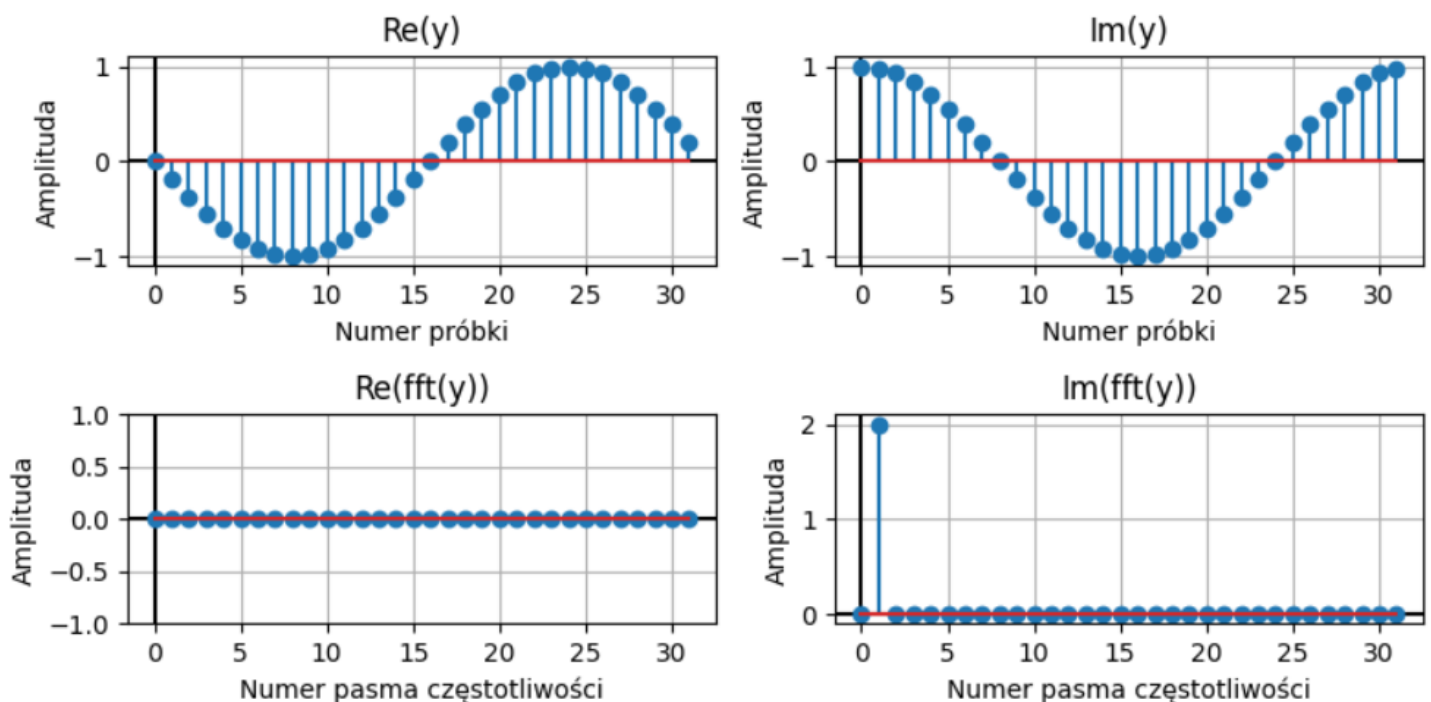
```

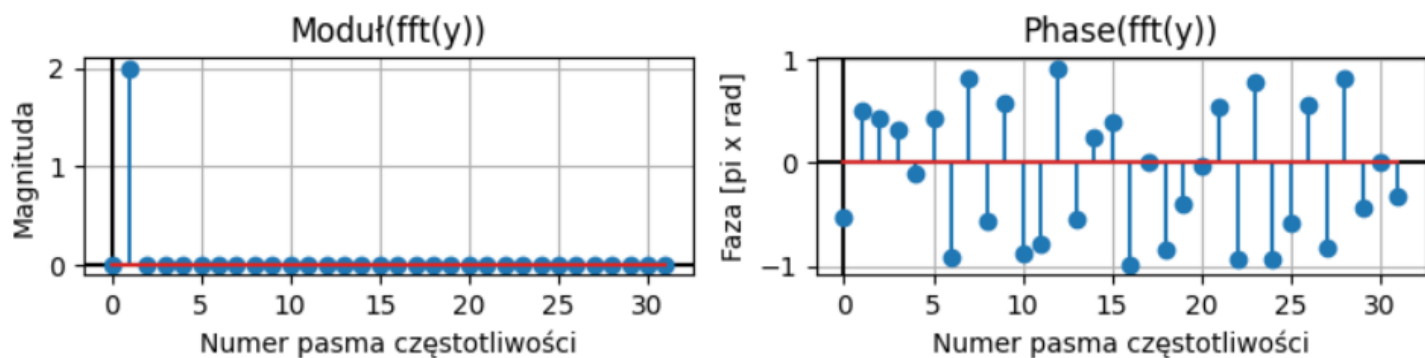
```

38
39 #cz. urojona funkcji FFT(y)
40 plt.subplot(3,2,4)
41 plt.title('Im(fft(y))')
42 plt.xlabel('Numer pasma częstotliwości')
43 plt.ylabel('Amplituda')
44 plt.axhline(y=0,color = "k")
45 plt.axvline(x=0,color = "k")
46 plt.stem(np.imag(z),use_line_collection=1)
47 plt.grid(True,which='both')
48 #moduł funkcji FFT(y)
49 plt.subplot(3,2,5)
50 plt.title('Moduł(fft(y))')
51 plt.xlabel('Numer pasma częstotliwości')
52 plt.ylabel('Magnituda')
53 plt.axhline(y=0,color = "k")
54 plt.axvline(x=0,color = "k")
55 plt.stem(np.abs(z),use_line_collection=1)
56 plt.grid(True,which='both')
57 #kąt funkcji FFT(y)
58 plt.subplot(3,2,6)
59 plt.title('Phase(fft(y))')
60 plt.xlabel('Numer pasma częstotliwości')
61 plt.ylabel('Faza [pi x rad]')
62 plt.axhline(y=0,color = "k")
63 plt.axvline(x=0,color = "k")
64 plt.stem(np.angle(z)/np.pi,use_line_collection=1)
65 plt.grid(True,which='both')
66
67 plt.show()

```

Wynik:





Zadanie 5

Treść zadania:

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

Stworzenie własnych funkcji DFT IDFT:

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.linalg import dft as dft
4
5  #stworzenie własnych funkcji DFT oraz IDFT
6  def DFT(y,n):
7      m = dft(n)
8      return y@m
9
10 def IDFT(f,n):
11     m = dft(n)
12     odw = np.conj(m)/n
13     return f@odw
```

Tworzenie zmiennych:

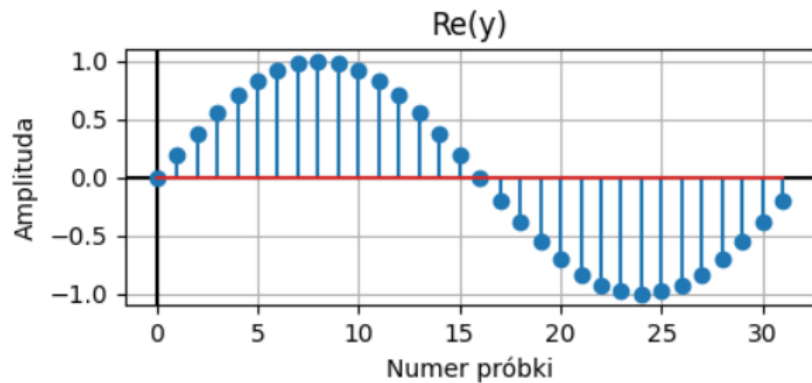
```
17  n = 32
18  x = np.arange(n) #zakres 32 próbek
19  y = np.sin(2*np.pi*x/n) #funkcja sinusa
20  z = 2*np.fft.fft(y)/n #funkcja FFT z sinusa
21  ydft = 2*DFT(y,n)/n #funkcja własna DFT z sinusa
22
```


Wykres funkcji bazowej:

```

25
26 plt.title('Re(y)')
27 plt.xlabel('Numer próbek')
28 plt.ylabel('Amplituda')
29 plt.axhline(y=0,color = "k")
30 plt.axvline(x=0,color = "k")
31 plt.stem(np.real(y),use_line_collection=1)
32 plt.grid(True,which='both')
33
34 plt.show()
35
36

```



Porównanie własnej funkcji z funkcją z zadania 3.1:

```

37 plt.subplot(4,2,1)
38 plt.title('Re(z)')
39 plt.xlabel('Numer pasma częstotliwości')
40 plt.ylabel('Amplituda')
41 plt.axhline(y=0,color = "k")
42 plt.axvline(x=0,color = "k")
43 plt.stem(np.real(z),use_line_collection=1)
44 plt.ylim(-1,1)
45 plt.grid(True,which='both')
46
47 plt.subplot(4,2,2)
48 plt.title('Re(ydft) - funkcja własna')
49 plt.xlabel('Numer pasma częstotliwości')
50 plt.ylabel('Amplituda')
51 plt.axhline(y=0,color = "k")
52 plt.axvline(x=0,color = "k")
53 plt.ylim(-1,1)
54 plt.stem(np.real(ydft),use_line_collection=1)
55 plt.grid(True,which='both')
56
57 plt.subplot(4,2,3)
58 plt.title('Im(z)')
59 plt.xlabel('Numer pasma częstotliwości')
60 plt.ylabel('Amplituda')
61 plt.axhline(y=0,color = "k")
62 plt.axvline(x=0,color = "k")
63 plt.stem(np.imag(z),use_line_collection=1)
64 plt.grid(True,which='both')
65
66 plt.subplot(4,2,4)
67 plt.title('Im(ydft) - funkcja własna')
68 plt.xlabel('Numer pasma częstotliwości')
69 plt.ylabel('Amplituda')
70 plt.axhline(y=0,color = "k")
71 plt.axvline(x=0,color = "k")
72 plt.stem(np.imag(ydft),use_line_collection=1)
73 plt.grid(True,which='both')

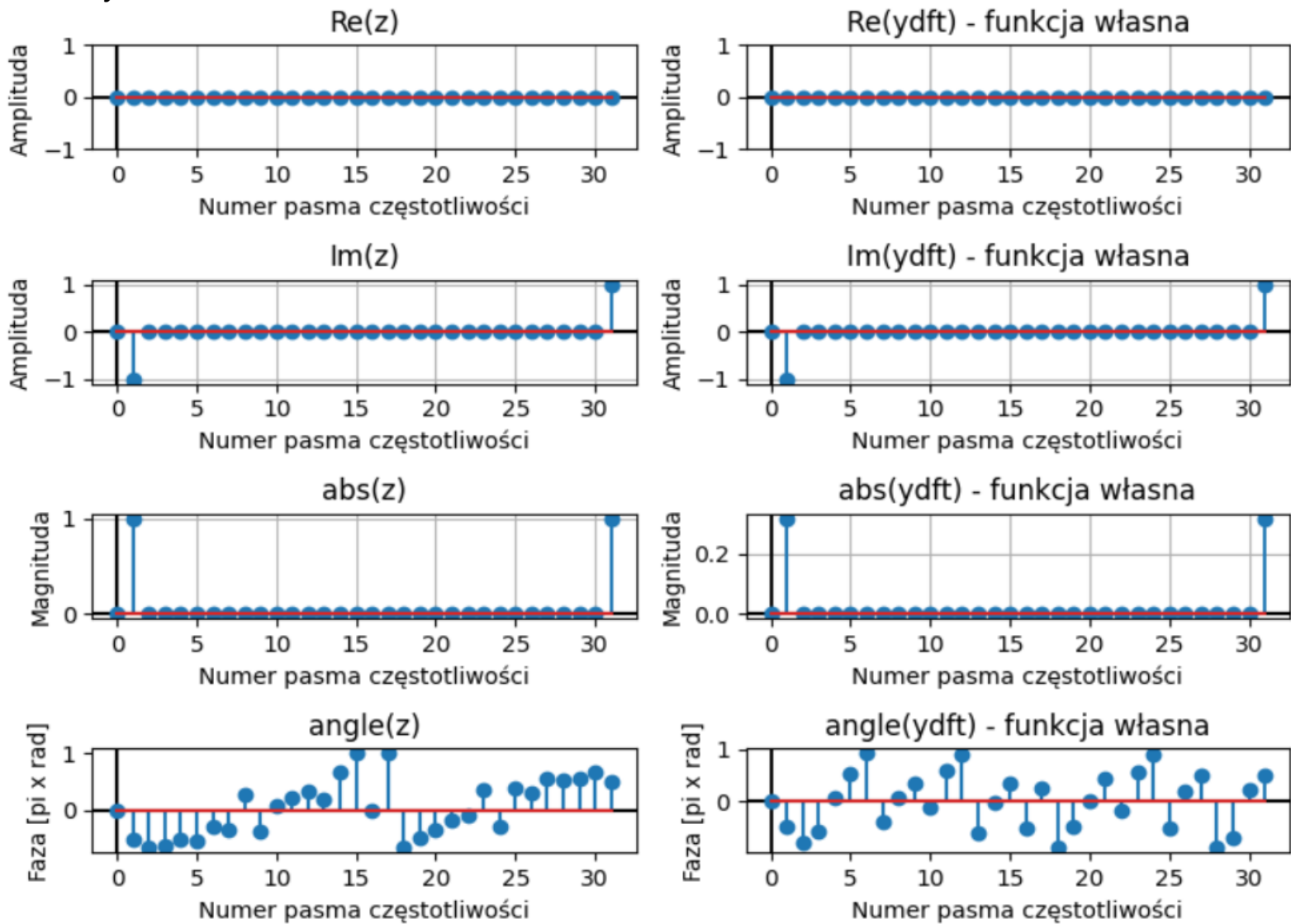
```

```

74
75 plt.subplot(4,2,5)
76 plt.title('abs(z)')
77 plt.xlabel('Numer pasma częstotliwości')
78 plt.ylabel('Magnituda')
79 plt.axhline(y=0,color = "k")
80 plt.axvline(x=0,color = "k")
81 plt.stem(np.abs(z),use_line_collection=1)
82 plt.grid(True,which='both')
83
84 plt.subplot(4,2,6)
85 plt.title('abs(ydft) - funkcja własna')
86 plt.xlabel('Numer pasma częstotliwości')
87 plt.ylabel('Magnituda')
88 plt.axhline(y=0,color = "k")
89 plt.axvline(x=0,color = "k")
90 plt.stem(np.abs(ydft)/np.pi,use_line_collection=1)
91 plt.grid(True,which='both')
92
93 plt.subplot(4,2,7)
94 plt.title('angle(z)')
95 plt.xlabel('Numer pasma częstotliwości')
96 plt.ylabel('Faza [pi x rad]')
97 plt.axhline(y=0,color = "k")
98 plt.axvline(x=0,color = "k")
99 plt.stem(np.angle(z)/np.pi,use_line_collection=1)
100
101 plt.subplot(4,2,8)
102 plt.title('angle(ydft) - funkcja własna')
103 plt.xlabel('Numer pasma częstotliwości')
104 plt.ylabel('Faza [pi x rad]')
105 plt.axhline(y=0,color = "k")
106 plt.axvline(x=0,color = "k")
107 plt.stem(np.angle(ydft)/np.pi,use_line_collection=1)
108
109 plt.show()
110
111

```

Wynik:



Deklaracja zmiennych z zadania 3.2 oraz stworzenie wykresy sumy sygnałów:

```

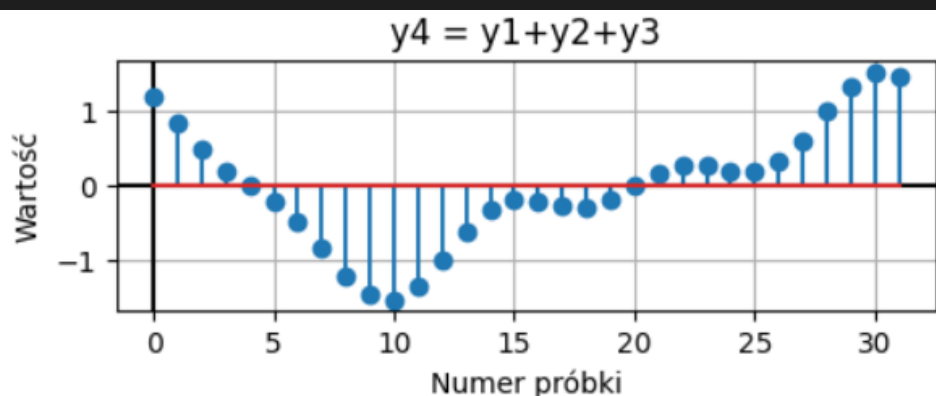
114 y1 = np.cos(2*np.pi*x/n+(np.pi/4)) #pierwsza funkcja z zadania 3.2
115 y2 = 0.5*np.cos(4*np.pi*x/n) #druga funkcja z zadania 3.2
116 y3 = 0.25*np.cos(8*np.pi*x/n+(np.pi/2)) #trzecia funkcja z zadania 3.2
117 y4 = y1 + y2 + y3 #suma funkcji
118 f4 = 2*np.fft.fft(y4)/n #FFT z sumy funkcji
119 ydft = 2*DFT(y4,n)/n #własne DFT z sumy funkcji

```

```

123 plt.title('y4 = y1+y2+y3')
124 plt.xlabel('Numer próbki')
125 plt.ylabel('Wartość')
126 plt.axhline(y=0,color = "k")
127 plt.axvline(x=0,color = "k")
128 plt.stem(y4,use_line_collection=1)
129 plt.grid(True,which='both')
130
131 plt.show()
132

```



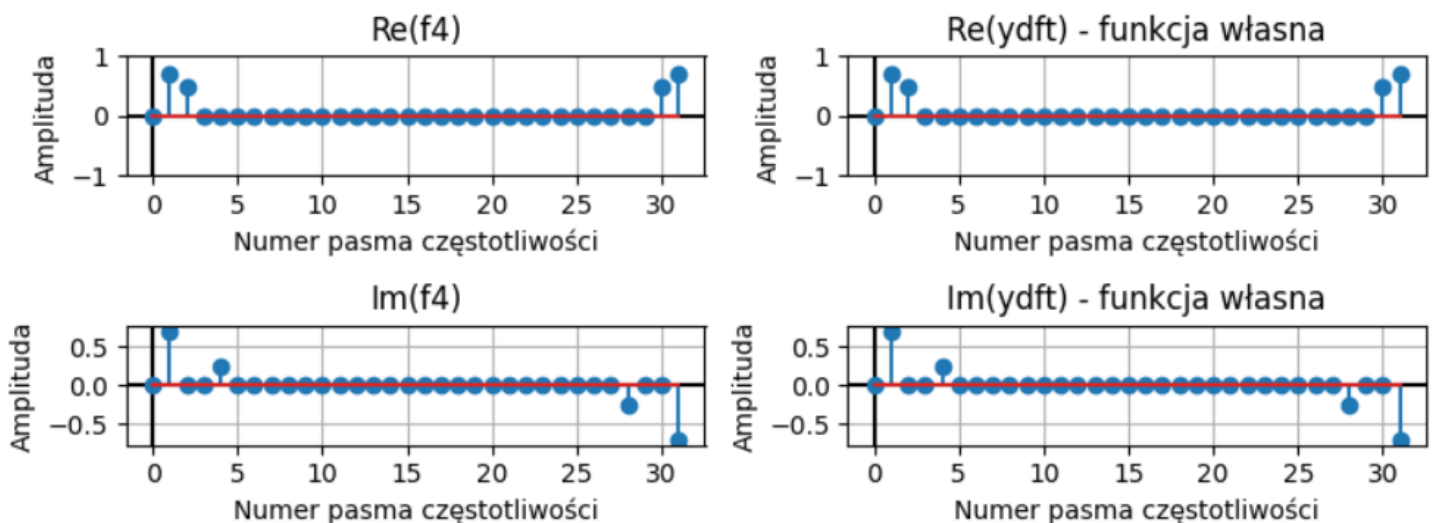
Porównanie własnej funkcji z funkcją z zadania 3.2:

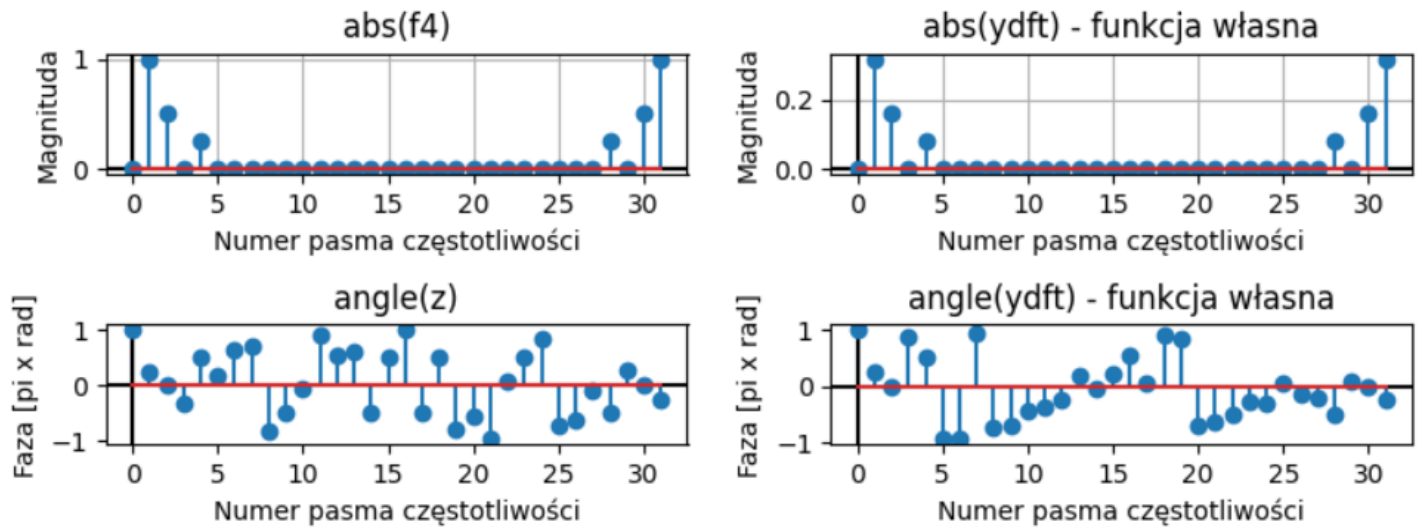
```

140 plt.subplot(4,2,1)
141 plt.title('Re(f4)')
142 plt.xlabel('Numer pasma częstotliwości')
143 plt.ylabel('Amplituda')
144 plt.axhline(y=0,color = "k")
145 plt.axvline(x=0,color = "k")
146 plt.stem(np.real(f4),use_line_collection=1)
147 plt.ylim(-1,1)
148 plt.grid(True,which='both')
149
150 plt.subplot(4,2,2)
151 plt.title('Re(ydft) - funkcja własna')
152 plt.xlabel('Numer pasma częstotliwości')
153 plt.ylabel('Amplituda')
154 plt.axhline(y=0,color = "k")
155 plt.axvline(x=0,color = "k")
156 plt.ylim(-1,1)
157 plt.stem(np.real(ydft),use_line_collection=1)
158 plt.grid(True,which='both')
159
160 plt.subplot(4,2,3)
161 plt.title('Im(f4)')
162 plt.xlabel('Numer pasma częstotliwości')
163 plt.ylabel('Amplituda')
164 plt.axhline(y=0,color = "k")
165 plt.axvline(x=0,color = "k")
166 plt.stem(np.imag(f4),use_line_collection=1)
167 plt.grid(True,which='both')
168
169 plt.subplot(4,2,4)
170 plt.title('Im(ydft) - funkcja własna')
171 plt.xlabel('Numer pasma częstotliwości')
172 plt.ylabel('Amplituda')
173 plt.axhline(y=0,color = "k")
174 plt.axvline(x=0,color = "k")
175 plt.stem(np.imag(ydft),use_line_collection=1)
176 plt.grid(True,which='both')
177
178 plt.subplot(4,2,5)
179 plt.title('abs(f4)')
180 plt.xlabel('Numer pasma częstotliwości')
181 plt.ylabel('Magnituda')
182 plt.axhline(y=0,color = "k")
183 plt.axvline(x=0,color = "k")
184 plt.stem(np.abs(f4),use_line_collection=1)
185 plt.grid(True,which='both')
186
187 plt.subplot(4,2,6)
188 plt.title('abs(ydft) - funkcja własna')
189 plt.xlabel('Numer pasma częstotliwości')
190 plt.ylabel('Magnituda')
191 plt.axhline(y=0,color = "k")
192 plt.axvline(x=0,color = "k")
193 plt.stem(np.abs(ydft)/np.pi,use_line_collection=1)
194 plt.grid(True,which='both')
195
196 plt.subplot(4,2,7)
197 plt.title('angle(z)')
198 plt.xlabel('Numer pasma częstotliwości')
199 plt.ylabel('Faza [pi x rad]')
200 plt.axhline(y=0,color = "k")
201 plt.axvline(x=0,color = "k")
202 plt.stem(np.angle(f4)/np.pi,use_line_collection=1)
203
204 plt.subplot(4,2,8)
205 plt.title('angle(ydft) - funkcja własna')
206 plt.xlabel('Numer pasma częstotliwości')
207 plt.ylabel('Faza [pi x rad]')
208 plt.axhline(y=0,color = "k")
209 plt.axvline(x=0,color = "k")
210 plt.stem(np.angle(ydft)/np.pi,use_line_collection=1)
211
212 plt.show()

```

Wyniki





Zadanie 6

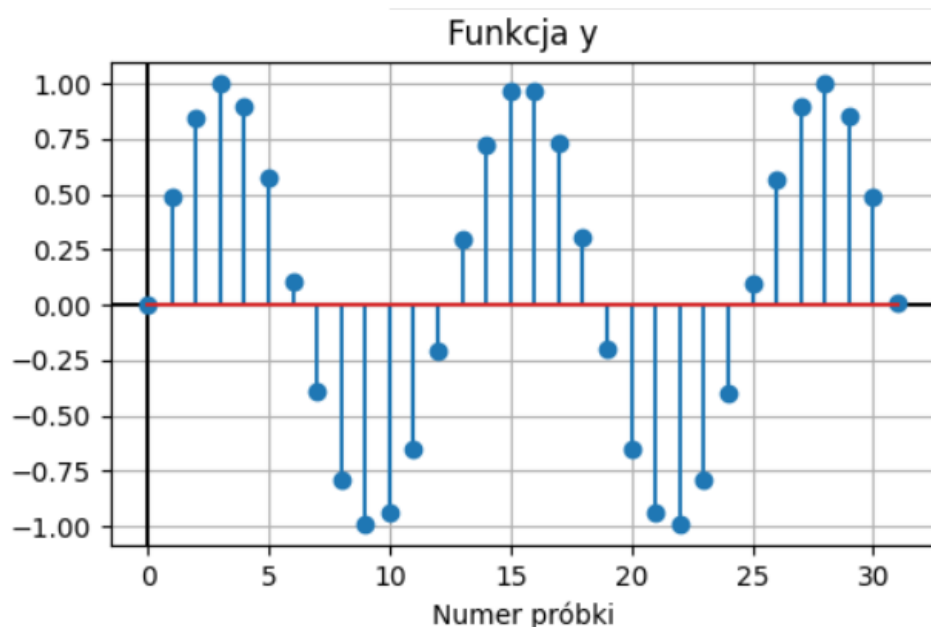
Treść zadania:

Wygeneruj 2.5 okresu sinusoidy. Sprawdź jak wygląda moduł widma zespolonego FFT sygnału.

- Skąd biorą się zniekształcenia?
- Porównaj z FFT sygnału wymnożonego przez funkcję okna. Zastosuj funkcję okna typu: bartlett, blackman, hamming, hann, kaiser, rectangular (okno prostokątne).
- Porównaj parametry okien.

Wygenerowanie 2,5 okresu sinusoidy:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import scipy.signal as sig
4
5 n = 32
6 x = np.arange(n)
7 y = np.sin(((5*np.pi)+0.5)*x/n)
8
9 plt.title('Funkcja y')
10 plt.xlabel('Numer próbki')
11 plt.ylabel('Amplituda')
12 plt.axhline(y=0,color = "k")
13 plt.axvline(x=0,color = "k")
14 plt.stem(y,use_line_collection=1)
15 plt.grid(True,which='both')
16
17 plt.show()
```

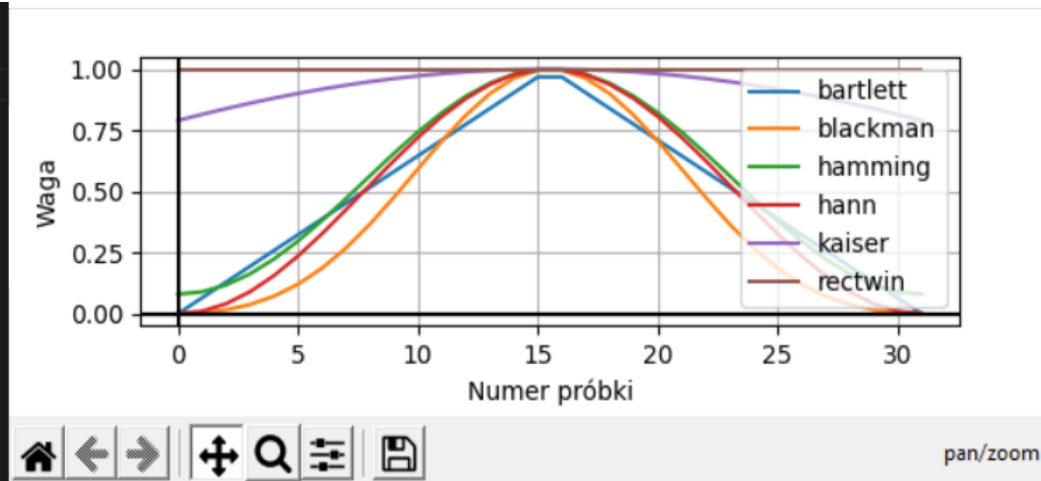


Funkcje okna:

```

19
20
21
22
23 plt.xlabel('Numer próbki')
24 plt.ylabel('Waga')
25 plt.plot(np.bartlett(n))
26 plt.plot(np.blackman(n))
27 plt.plot(np.hamming(n))
28 plt.plot(sig.hann(n))
29 plt.plot(np.kaiser(n,1))
30 plt.plot(np.kaiser(n,0))
31 plt.axhline(y=0,color = "k")
32 plt.axvline(x=0,color = "k")
33 plt.grid(True,which='both')
34 plt.legend(["bartlett","blackman","hamming","hann","kaiser","rectwin"],loc="upper right")
35 plt.show()

```

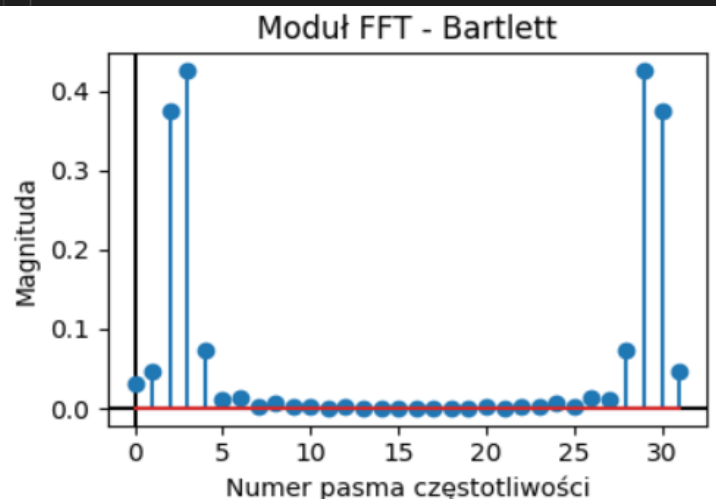
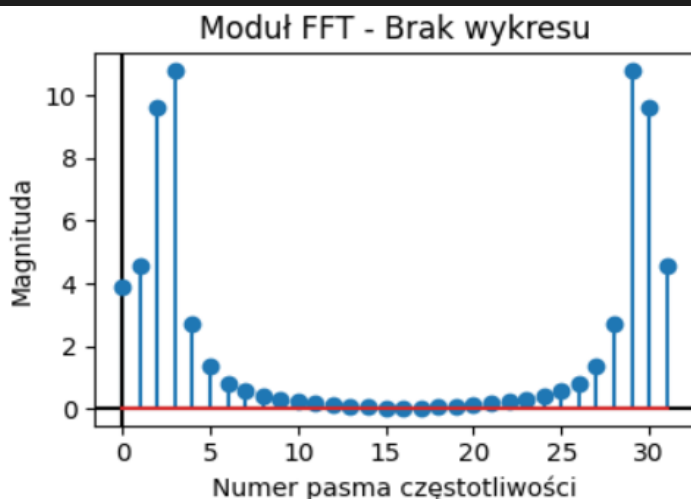


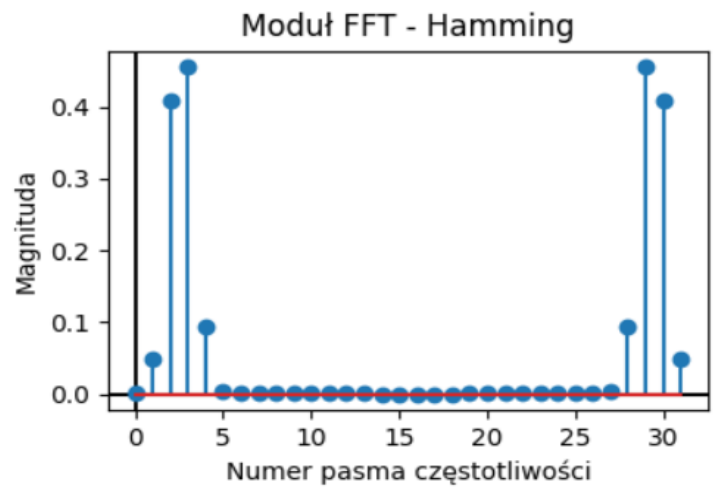
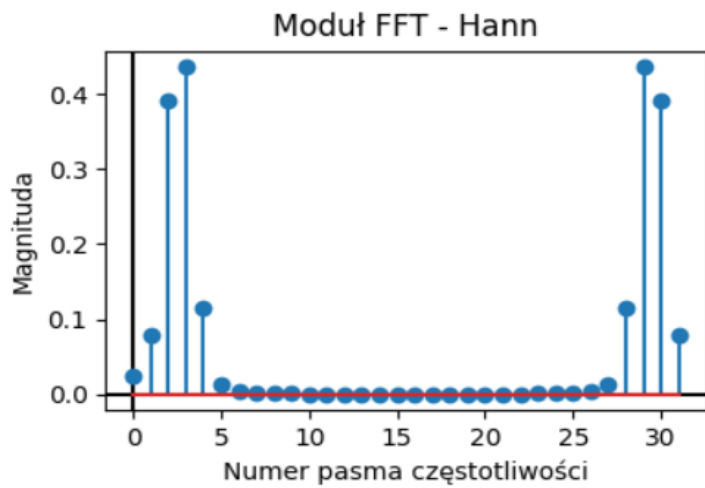
Porównanie FFT sygnału wymnożonego przez funkcję okna:

```

35 z = np.fft.fft(y)
36 plt.subplot(2,2,1)
37 plt.title('Moduł FFT - Brak wykresu')
38 plt.xlabel('Numer pasma częstotliwości')
39 plt.ylabel('Magnituda')
40 plt.axhline(y=0,color = "k")
41 plt.axvline(x=0,color = "k")
42 plt.stem(abs(z),use_line_collection=1)
43
44 z1 = np.bartlett(n)*y
45 z12 = 2*np.fft.fft(z1)/n
46 plt.subplot(2,2,2)
47 plt.title('Moduł FFT - Bartlett')
48 plt.xlabel('Numer pasma częstotliwości')
49 plt.ylabel('Magnituda')
50 plt.axhline(y=0,color = "k")
51 plt.axvline(x=0,color = "k")
52 plt.stem(abs(z12),use_line_collection=1)
53
54
55
56 z2 = sig.hann(n)*y
57 z22 = 2*np.fft.fft(z2)/n
58 plt.subplot(2,2,3)
59 plt.title('Moduł FFT - Hann')
60 plt.xlabel('Numer pasma częstotliwości')
61 plt.ylabel('Magnituda')
62 plt.axhline(y=0,color = "k")
63 plt.axvline(x=0,color = "k")
64 plt.stem(abs(z22),use_line_collection=1)
65
66 z3 = np.hamming(n)*y
67 z32 = 2*np.fft.fft(z3)/n
68 plt.subplot(2,2,4)
69 plt.title('Moduł FFT - Hamming')
70 plt.xlabel('Numer pasma częstotliwości')
71 plt.ylabel('Magnituda')
72 plt.axhline(y=0,color = "k")
73 plt.axvline(x=0,color = "k")
74 plt.stem(abs(z32),use_line_collection=1)
75
76 plt.show()

```





```

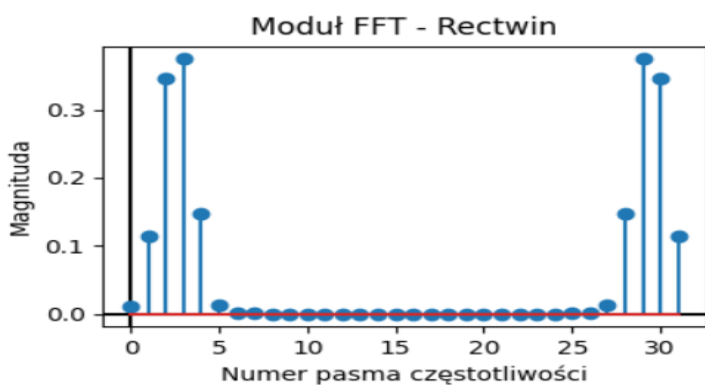
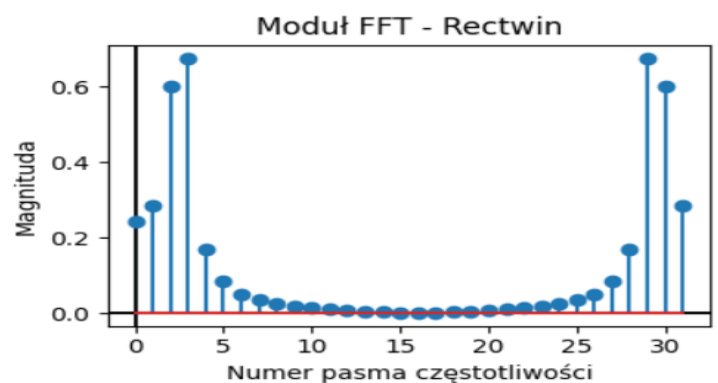
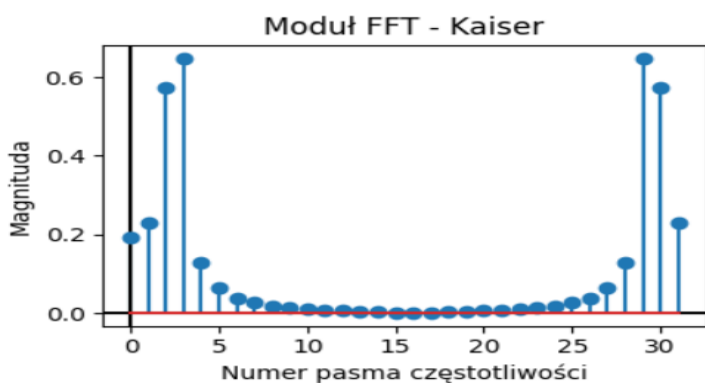
80 z4 = np.kaiser(n,1)*y
81 z42 = 2*np.fft.fft(z4)/n
82 plt.subplot(2,2,1)
83 plt.title('Moduł FFT - Kaiser')
84 plt.xlabel('Numer pasma częstotliwości')
85 plt.ylabel('Magnituda')
86 plt.axhline(y=0,color = "k")
87 plt.axvline(x=0,color = "k")
88 plt.stem(abs(z42),use_line_collection=1)
89
90 z5 = np.kaiser(n,0)*y
91 z52 = 2*np.fft.fft(z5)/n
92 plt.subplot(2,2,2)
93 plt.title('Moduł FFT - Rectwin')
94 plt.xlabel('Numer pasma częstotliwości')
95 plt.ylabel('Magnituda')
96 plt.axhline(y=0,color = "k")
97 plt.axvline(x=0,color = "k")
98 plt.stem(abs(z52),use_line_collection=1)
99

```

```

101
102
103 z6 = np.blackman(n)*y
104 z62 = 2*np.fft.fft(z6)/n
105 plt.subplot(2,2,3)
106 plt.title('Moduł FFT - Rectwin')
107 plt.xlabel('Numer pasma częstotliwości')
108 plt.ylabel('Magnituda')
109 plt.axhline(y=0,color = "k")
110 plt.axvline(x=0,color = "k")
111 plt.stem(abs(z62),use_line_collection=1)
112
113 plt.show()

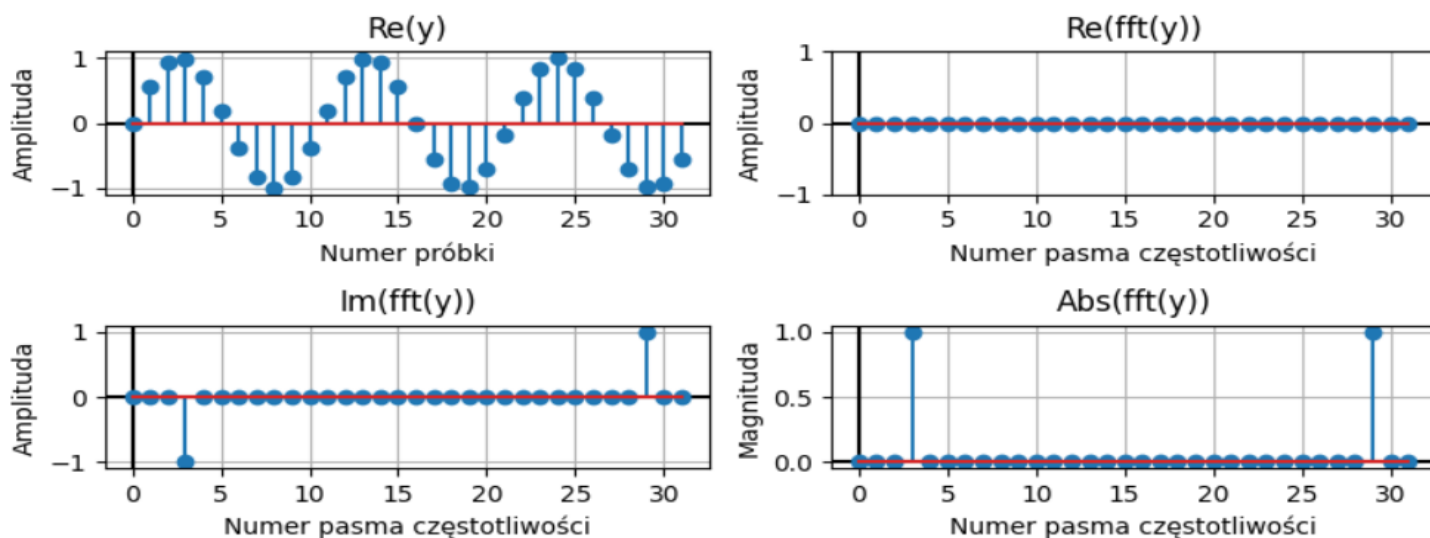
```



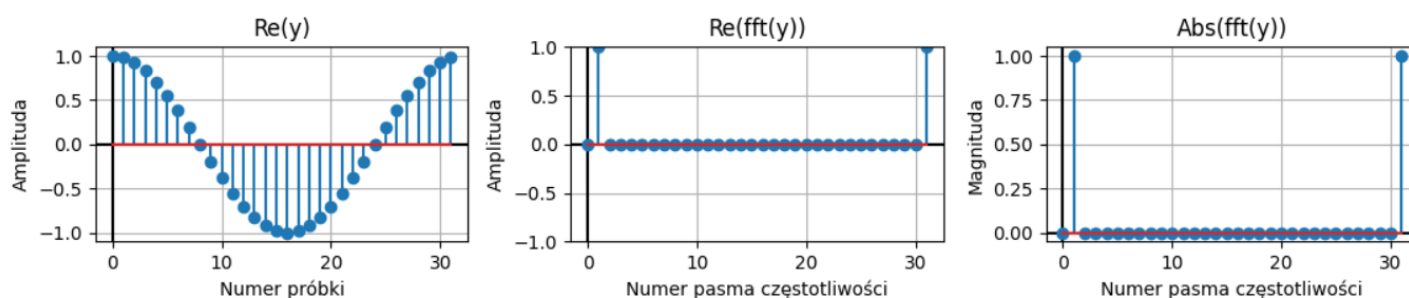
Interpretacja wyników i wnioski

Wynalezienie szeregu Fouriera jest niezwykle istotnym odkryciem. Pozwala rozłożyć dowolną okresową funkcję na szereg jej składowych. Dzięki temu jesteśmy w stanie przeanalizować ogromną ilość sygnałów, które na pierwszy rzut oka wydają się być bardzo skomplikowane. Transformacja Fouriera również ułatwia analizowania sygnałów. Za jej pomocą jesteśmy w stanie przekształcić funkcję z dziedziną czasu w funkcję z dziedziną częstotliwości. Za pomocą graficznego przedstawienia widma jesteśmy w stanie określić na przykład z ilu częstotliwości składa się nasz początkowy sygnał. Używając języka Python korzystamy z szybkiej transformaty Fouriera. Dzieje się tak, ponieważ ma ona dużo lepszą złożoność obliczeniową ($O(n \log n)$).

W pierwszym zadaniu badano transformatę Fouriera funkcji sinus. Zauważono, że dla częstotliwości $f=1$ na wykresie części urojonej oraz modułu powstały dwa „piki”. Dla części urojonej zachodzi symetria względem punktu $(n/2, 0)$, natomiast dla modułu widoczna jest symetria względem prostej $x=n/2$. Dodatkowo sprawdzono co się dzieje w przypadku zwiększenia częstotliwości. Okazało się, że „piki” przesuwają się do punktu $(n/2, 0)$.



Zbadano również jak zachowuje się widmo w przypadku funkcji cosinus. Otóż w części rzeczywistej jak i modułu pojawiają się dwa „piki”, w obu przypadkach są one symetryczne względem prostej $x=n/2$.



W zadaniu drugim porównano widma sygnałów składowych z widmem ich sumy. Zauważono, że widmo sumy sygnałów jest połączeniem widm składowych.

W kolejnym zadaniu należało wyznaczyć odwrotną transformatę Fouriera dla widm FFT z poprzednich zadań. Skorzystano z funkcji `ifft` z pakietu NumPy. Działa ona poprawnie. Widma powstałe w wyniku odwrotnej transformaty pokrywają się z wykresami początkowymi sygnału. Oznacza to, że mając transformatę Fouriera, jesteśmy w stanie powrócić do sygnału początkowego.

W czwartym zadaniu porównywano widma sinusoidy zespolonej z tą z zadania 3.1. Różnią się one znacząco. W części urojonej w zadaniu 4 otrzymano tylko jeden „pik”, podobnie w przypadku modułu. Natomiast w zadaniu 1 pojawiają się dwa „piki”. Nie jesteśmy zatem w stanie mówić tu o symetrii. Zauważono również różnice w części urojonej obu funkcji y . W przypadku zadania 1 jest ona dla każdej

próbki równa zero, natomiast w sygnale sinusoidy zespolonej częścią urojoną jest cosinus.

W zadaniu piątym porównując wykresy otrzymane w zadaniu 3.1 oraz 3.2 i wykresów z użyciem funkcji własnej okazuje się, że wykresy są takie same z wyjątkiem fazy widma, które nieznacznie się różni. Oznacza to, że własna funkcja dobrze zastępuje polecenie `np.fft.fft()` i można ją poprawnie używać do generowania wykresów.

W kolejnym zadaniu badano zniekształcenia modułu widma zespolonego FFT. Zniekształcenia te powstają w wyniku zastosowania różnych funkcji typów okna przy tworzeniu wykresu. Każda funkcja typu okna różni się od drugiej w swojej składni, dlatego powstają też drobne zniekształcenia. Nie różnią się one bardzo od modułu FFT bez okna, lecz zwracają nieco inne amplitudy oraz częstotliwość w wykresach modułu z oknem różni się nieco od tego bez okna. Parametry w wykresach modułu z różnymi oknami są do siebie podobne.

Źródła

- [1] [http://wikidyd.iem.pw.edu.pl/attachments/LAPPD\(2f\)Cw2/c2_aws_instr.pdf](http://wikidyd.iem.pw.edu.pl/attachments/LAPPD(2f)Cw2/c2_aws_instr.pdf)
- [2] <http://aragorn.pb.bialystok.pl/~boldak/DIP/CPO-W06-v05-50pr.pdf>
- [3] <https://pl.wikipedia.org>
- [4] https://home.agh.edu.pl/~zobmat/2020/II_mich_mar/wstep.html
- [5] http://prac.im.pwr.edu.pl/~agniesz/rachunek_prawd_MAEW104/wyklady/R_Pr_MAEW104_wyklad2_szeregi_Fouriera.pdf
- [6] [http://wikidyd.iem.pw.edu.pl/attachments/LAPPD\(2f\)Cw2/c2_aws_instr.pdf](http://wikidyd.iem.pw.edu.pl/attachments/LAPPD(2f)Cw2/c2_aws_instr.pdf)
- [7] https://eti.pg.edu.pl/documents/176593/26763380/Wykl_AlgorOblicz_10.pdf
- [8] <https://www.youtube.com/watch?v=spUNpyF58BY>