Sprawozdanie PSIwSUM

STFT i Falki

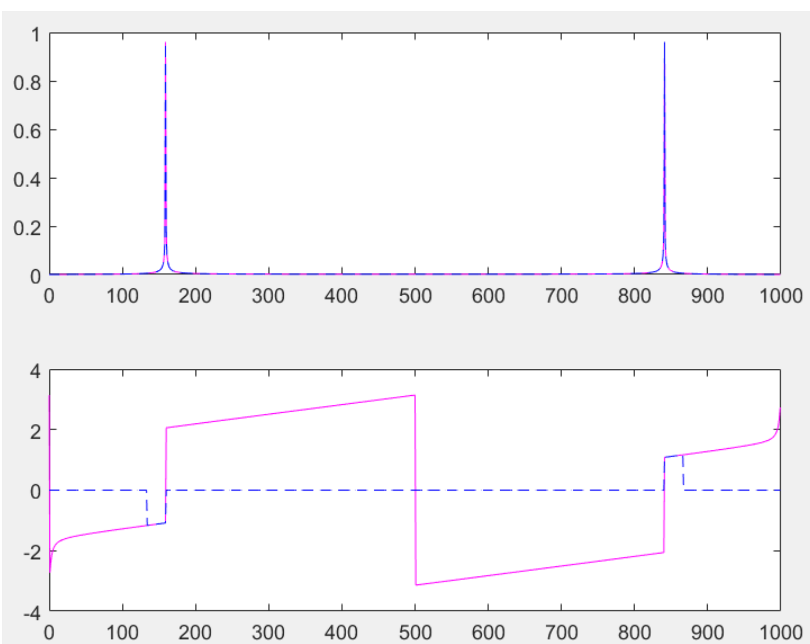
Urszula Starowicz

407177

Zadanie 1

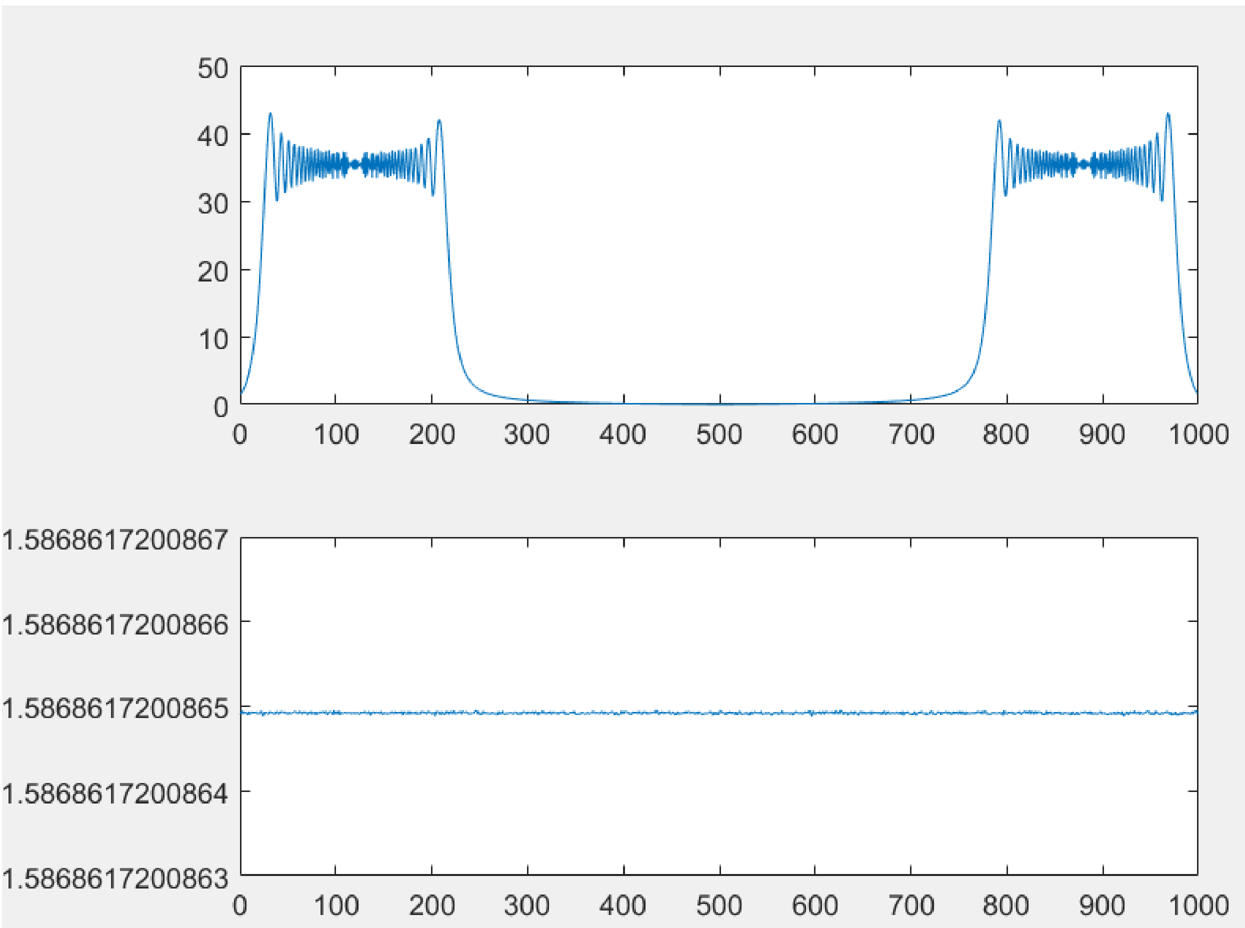
Własna implementacja Dyskretnej Transformaty Fourier'a.

```
clear all;
close all;
clc;

T=1;
Fs=1000;
df=1/T;
N=T*Fs;
t=0:N-1;
F=[0:(N-1)]/N*Fs;
h=1e0;
s=sin(t);
N=length(s);
n=0:N-1;

S_fft=fft(s);
A_fft=2*abs(S_fft)/N;
P_fft=angle(S_fft);
ind=S_fft>h;
P_fft(~ind)=0;

for k=0:1:N-1

    e_cos=cos(-
2*pi*k*n/N);
    e_sin=sin(-
2*pi*k*n/N);

e=complex(e_cos,e_sin);
    S_dft(k+1)=sum(s.*e);
end

A_dft=2*abs(S_dft)/N;
P_dft=angle(S_fft);
ind=S_dft>h;

subplot(2,1,1)
plot(F,A_dft,'m')
hold on
plot(F,A_fft,'b--')
hold off

subplot(2,1,2)
plot(F,P_dft,'m')
hold on
plot(F,P_fft,'b--')
hold off
```



Zadanie 2

```
clear all
close all

f1=20;
f2=220;
t1=1;
fs=1000;
T=1;
t=0:1/fs:T;
n=length(t);
f=(0:(n-1))/n*fs;

chch=chirp(t,f1,t1,f2);

N=100;
A=0;
for k=0:1:fs;
    c=cos(-2*pi*k*n/N);
    s=sin(-2*pi*k*n/N);
    e=complex(c,s);
    Sdft(k+1)=sum(chch.*e);
end

X=fft(chch);

subplot(2,1,1)
plot(f,abs(fft(chch)))
subplot(2,1,2)
plot(f,abs(Sdft))
```

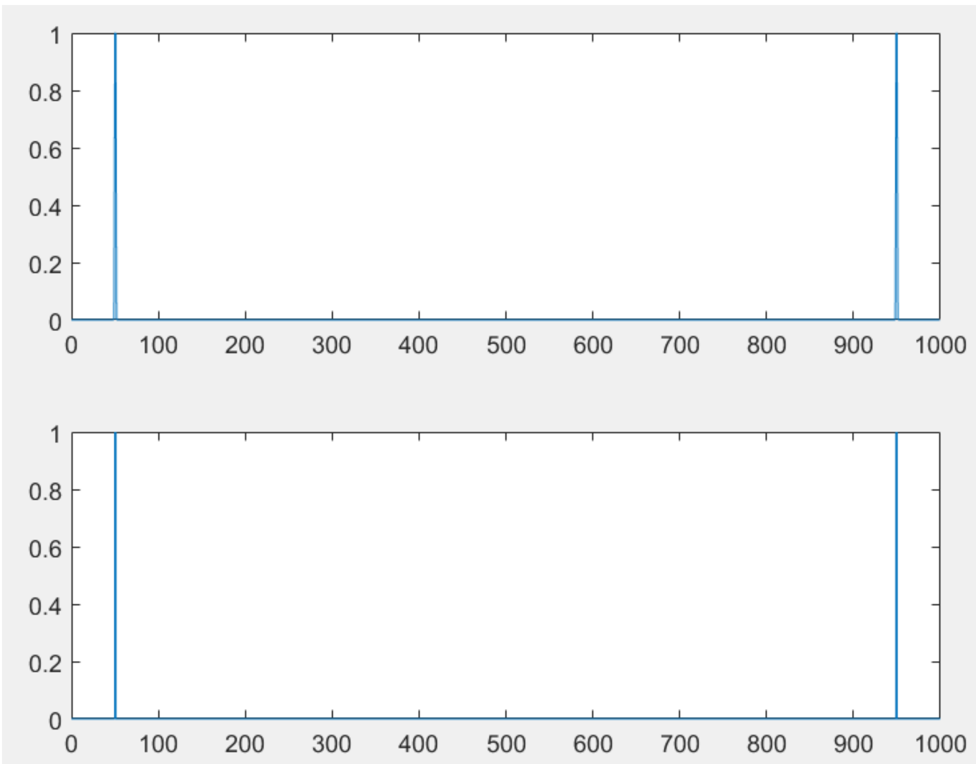


Wnioski:

Funkcja wbudowana oraz dft powinna pokrywać się, co oznaczałoby, że funkcje działają na podobnych założeniach.

Zadanie 3 – Porównanie widma dwóch sinusów o różnym czasie i częstotliwości różnej o jeden.

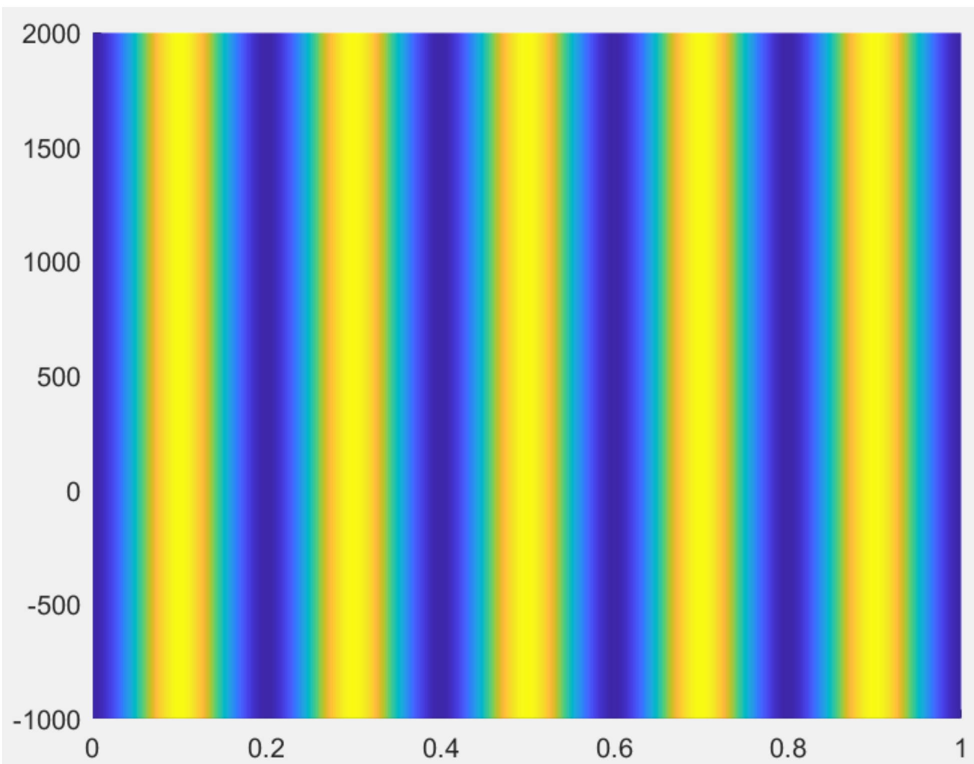
```
clear all;
close all;
clc;

T1=1;
T2=10;
Fs=1000;
N1=T1*Fs;
N2=T2*Fs;
t1=[0:N1-1]/Fs;
t2=[0:N2-1]/Fs;

F1=[0:N1-1]/N1*Fs;
F2=[0:N2-1]/N2*Fs;

s1=sin(2*pi*50*t1);
s2=sin(2*pi*51*t1);
s=s1+s2;

s3=sin(2*pi*50*t2);
s4=sin(2*pi*51*t2);
ss=s3+s4;

S1=fft(s);
S2=fft(ss);

a=2*abs(S1)/N1;
b=2*abs(S2)/N2;

subplot(2,1,1)
plot(F1,a)
subplot(2,1,2)
plot(F2,b)
```



Dłuższy czas daje nam więcej próbek – co oznacza, że wynik (odwzorowanie) będzie dokładniejszy. Można zaobserwować, że dokładniejsze odwzorowanie daje węższy pik.

Zadanie 4

```
clear all;
close all;
clc;

T=1;
Fs=1000;
df=1/T;
dt=1/Fs;
N=T*Fs;
t=(0:N-1)/Fs;
F=[0:(N-1)]/N*Fs;
a=0.25;
b=0;
Na=a*Fs;
chir=chirp(t,20,1,220);
j=1;
for i=1:1:N
 if j+Na>N+1
 break;
 end
c(i)=chir(j);
s=chir(1,j:1:i*Na);
l=length(s);
h=hann(l)';
S(1,j:1:i*Na)=fft(c(i)).*h;
j=Na+j;
end
%axis ydirection
imagesc('XData',t,'YData',F,'CData',S)
```

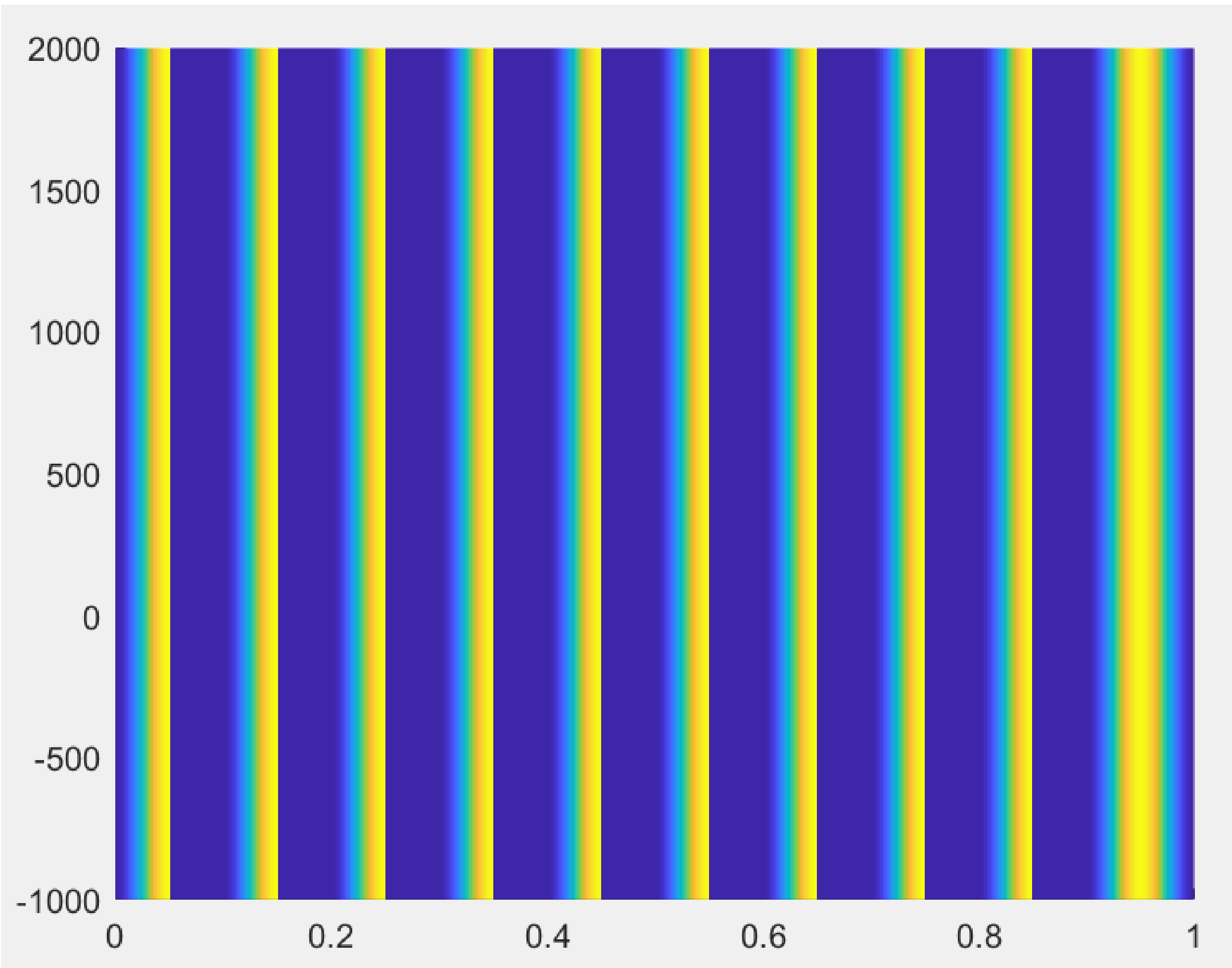


Próbowałam obrócić wykres z pomocą poleconej funkcji, jednak żadna z wersji znalezionej w help nie zadziałała.

```
clear all;
close all;
clc;

T=1;
Fs=1000;
df=1/T;
for i=1:1:N
 c(i)=chir(j);
 if j+Na>N+1
 break;
 end
s=chir(1,j:1:j+Na-1);
l=length(s);
```

```
dt=1/Fs;
N=T*Fs;
t=(0:N-1)/Fs;
F=[0:(N-1)]/N*Fs;
a=0.1;
b=0.5;
Na=a*Fs;
chir=chirp(t,20,1,220);
j=1;
```

```
h=hann(l)';
S(1,j:1:j+Na-1)=fft(c(i)).*h;
j=b*Na+j;
end
imagesc('XData',t,'YData',F,'CData',S)
```



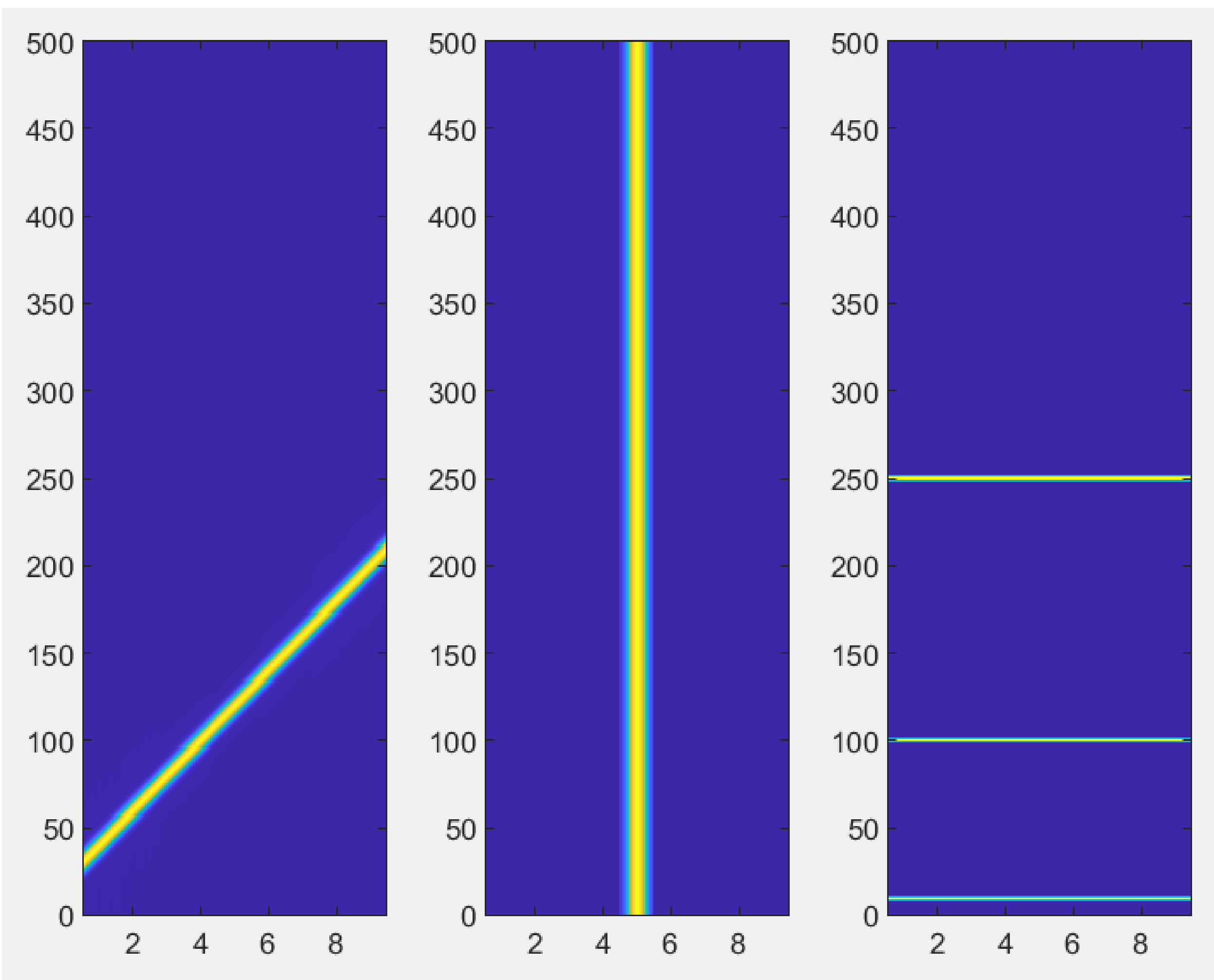
Zadanie 6

Jako, że potrzeba do jego wykonania poprawnie działającego zadania poprzedniego, nie dałam rady go wykonać. Ze względu, że nie wiedziałam do jakiego wyniku dążę i co powinnam poprawić.

Zadanie 7

```
clear all; clc;
Fs=1000;
t=0:1/Fs:10;
s1=chirp(t,20,10,220);
s2=zeros(size(t));
s2(5000)=1;
s3=sin(10*2*pi*t);
s4=sin(100*2*pi*t);
s5=sin(250*2*pi*t);
s=s3+s4+s5;
W1=1000;
W2=1000;
W3=1000;
N1=990;
N2=990;
N3=990;
n1=1000;
n2=1000;
n3=1000;
```

```
[S1,F1,T1] =
spectrogram(s1,W1,N1,n1,Fs);
[S2,F2,T2] =
spectrogram(s2,W2,N2,n2,Fs);
[S3,F3,T3] = spectrogram(s,W3,N3,n3,Fs);
subplot(1,3,1)
imagesc(T1,F1,abs(S1))
axis xy
subplot(1,3,2)
imagesc(T2,F2,abs(S2))

axis xy
subplot(1,3,3)
imagesc(T3,F3,abs(S3))
```



Funkcje:

Spectrogram – podaje krótkookienkowego Fouriera, której parametry definiujemy ręcznie (takie jak: sygnał, wielkość okna, ilość próbek i częstotliwości).

Cwt – daje nam ciągłą transformatę falkową. Wynik zależy od podanych parametrów (sygnału, metody użytej do wykonania obliczeń, czasu oraz częstotliwości próbkowania).

Chirp – sygnał typu chirp (cosinus o częstotliwości zmieniającej się w czasie), który zależy od wprowadzonych parametrów (macierzy chwil czasowych, częstotliwości, czasu do którego się odwołujemy, metodzie obliczeń, fazie sygnału, kształtu funkcji – wklęsła lub wypukła - oraz zawartości liczb zespolonych).