Przykładowe wykonania funkcji:

1. w projekcie:

tests/test\_projektu.py

oraz :

projekt/test\_projektu.py

1. czysty kod:

import sys

sys.path.append("build/Release")

import \_core

import time

from cmath import sin, pi

# Parametry sygnału

freq = 1

t\_start = 0

t\_end = 3

samples = 100

# 3. Sygnał sinusoidalny

sinus = \_core.sin\_signal(freq, t\_start, t\_end, samples)

print("sin\_signal:", sinus[:5])

\_core.plot\_signal(sinus)

time.sleep(1)

# 4. Cosinusoidalny

cosinus = \_core.cos\_signal(freq, t\_start, t\_end, samples)

print("cos\_signal:", cosinus[:5])

\_core.plot\_signal(cosinus)

time.sleep(1)

# 5. Sygnał prostokątny

square = \_core.square\_signal(freq, t\_start, t\_end, samples)

print("square\_signal:", square[:5])

\_core.plot\_signal(square)

time.sleep(1)

# 6. Piłozębny

saw = \_core.sawtooth\_signal(freq, t\_start, t\_end, samples)

print("sawtooth\_signal:", saw[:5])

\_core.plot\_signal(saw)

time.sleep(1)

# 7. DFT

# użyjmy sinusa z 4 próbek jako przykład

simple\_signal = [complex(s, 0) for s in \_core.sin\_signal(freq, 0, 1, 4)]

dft\_result = \_core.DFT(simple\_signal)

print("DFT:", dft\_result)

# 8. IDFT

idft\_result = \_core.IDFT(dft\_result)

print("IDFT:", idft\_result)

# 9. Wykres IDFT

\_core.plot\_signal([x.real for x in idft\_result])

# 10 filtr 1D

filtered1Dsin=\_core.apply\_filter(sinus, [0.1, 2, 4])

print("\n","1DFiltered sin: ",filtered1Dsin)

\_core.plot\_signal(filtered1Dsin)

# 11. filtr 2D

kernel = [[0.25, 0.5, 0.25]]

filtered2D = \_core.apply\_filter\_2D(sinus, kernel) # wektor sinusa jako 1xN

\_core.plot\_signal(filtered2D[0])

#12 Usuwanie niskich częstotliwości z sygnału (IDFT)

filtered = \_core.remove\_low\_f(cosinus, 4)

print("Oryginalny sygnał:")

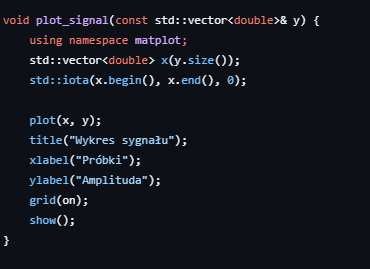
\_core.plot\_signal([round(s.real, 3) for s in cosinus])

time.sleep(1)

print("\nPo usunięciu niskich częstotliwości:")

\_core.plot\_signal([round(s.real, 3) for s in filtered])

1. Wizualizacja za pomocą matplotplusplus(gnuplot):



**2. DFT I IDFT**

Biblioteki:

#define \_USE\_MATH\_DEFINES

#include <vector>

#include <complex>

#include <cmath>

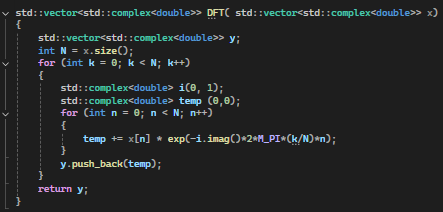
#include <iostream>

1. DFT

Wzór:



Kod:

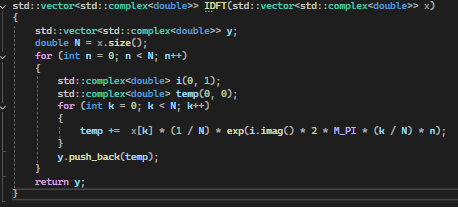


1. IDFT

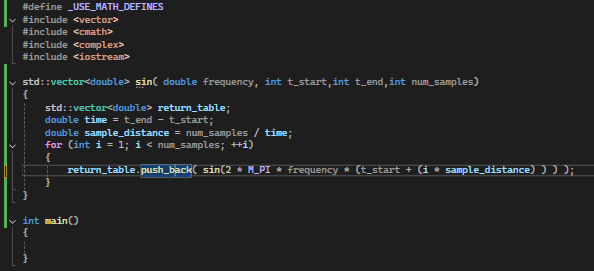
Wzór:



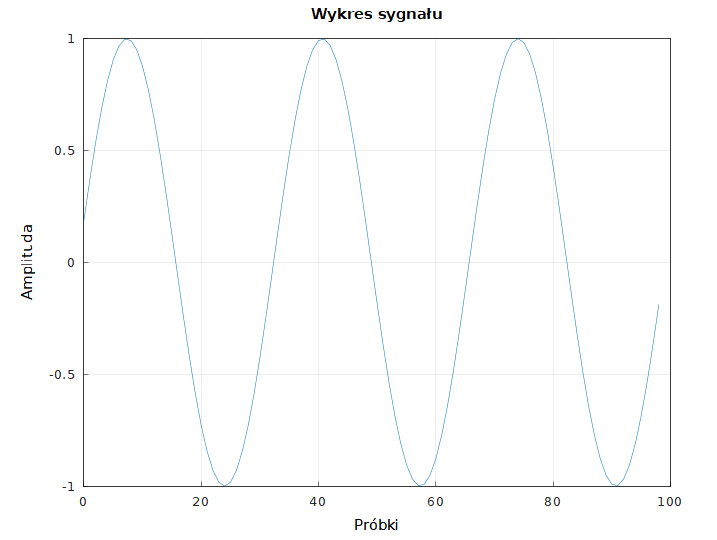
Kod:

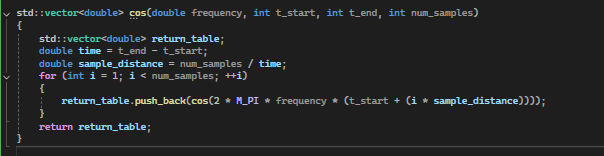


4. Funkcje zintegrowane

1. Funkcja sinus

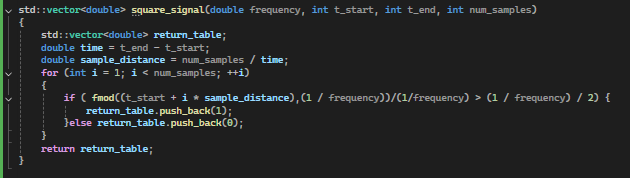
funkcja zwraca vector zawierający wartości sinusa o zadanej częstotliwości w zadanych granicach czasu o zadanej ilości próbek

Wizualizacja:  


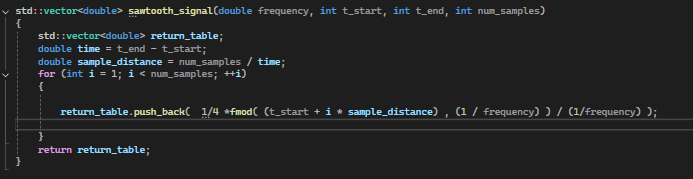
1. Sygnał Cosinus

funkcja zwraca vector zawierający wartości cosinusa o zadanej częstotliwości w zadanych granicach czasu o zadanej ilości próbek

1. Sygnał prostokątny

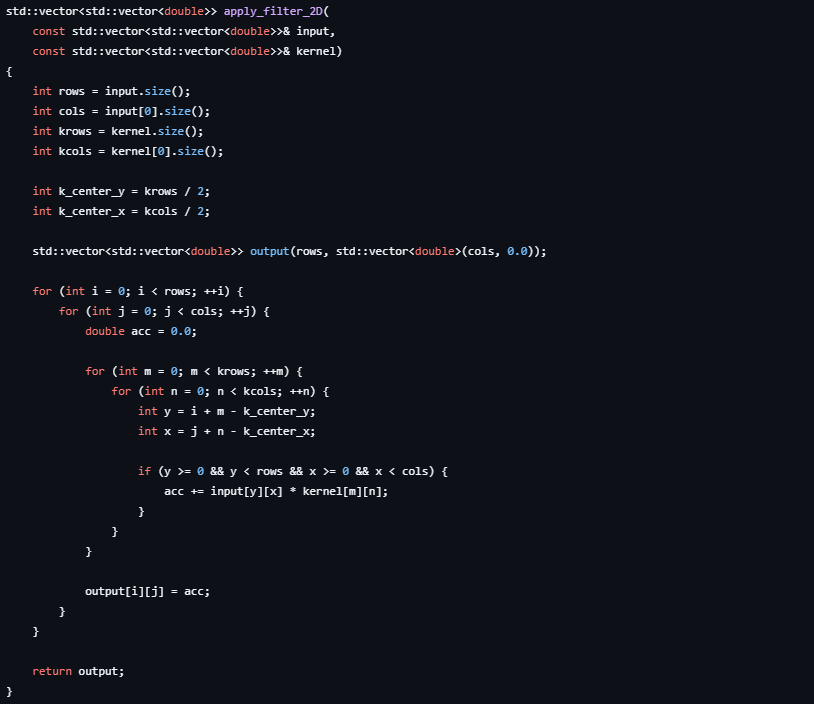


1. PIŁO zębny



1.Filtracja 1D 

2.Filtracja 2D



3.(dodatkowe)usuwanie wartości niskich częstotliwości przy użyciu DFT i IDFT

