

Raport – projekt 3 TP

Tomasz Kupski, 203579

Stanisław Oryl, 203702

Wszystkie generowane funkcje miały parametry:

częstotliwość 5Hz

ilość próbek 100

czas początkowy 0s

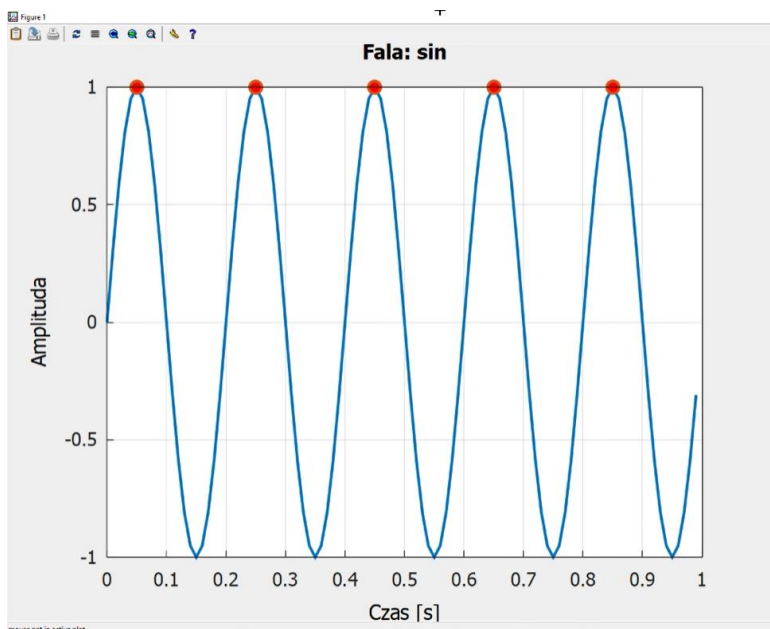
czas końcowy 1s

```
void generate_and_plot_matplotlib(...)
```

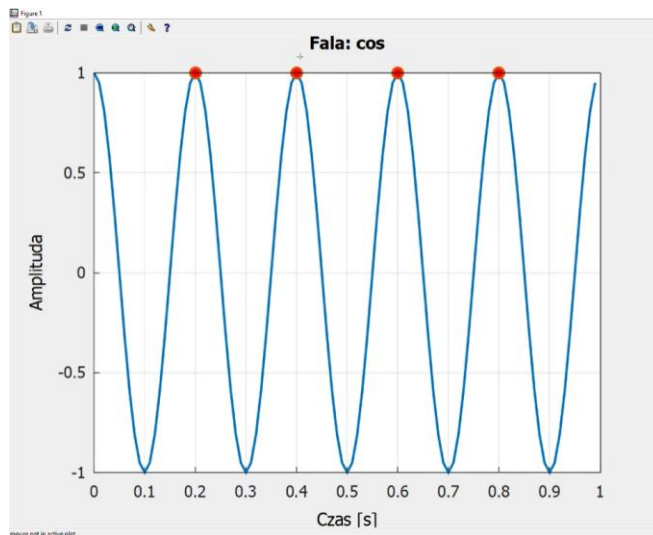
Główna funkcja sterująca, integrująca generowanie sygnału, jego analizę (DFT/IDFT), filtrację lub tylko wizualizację. Na podstawie parametrów wejściowych decyduje, jakie operacje przeprowadzić. Dla zwykłego sygnału rysuje wykres i zaznacza wykryte piki. W przypadku transformacji rysuje widmo (DFT), sygnał zrekonstruowany (IDFT) lub przefiltrowane dane.

```
std::vector<std::pair<int, double>> find_peaks(const std::vector<double>& signal)
```

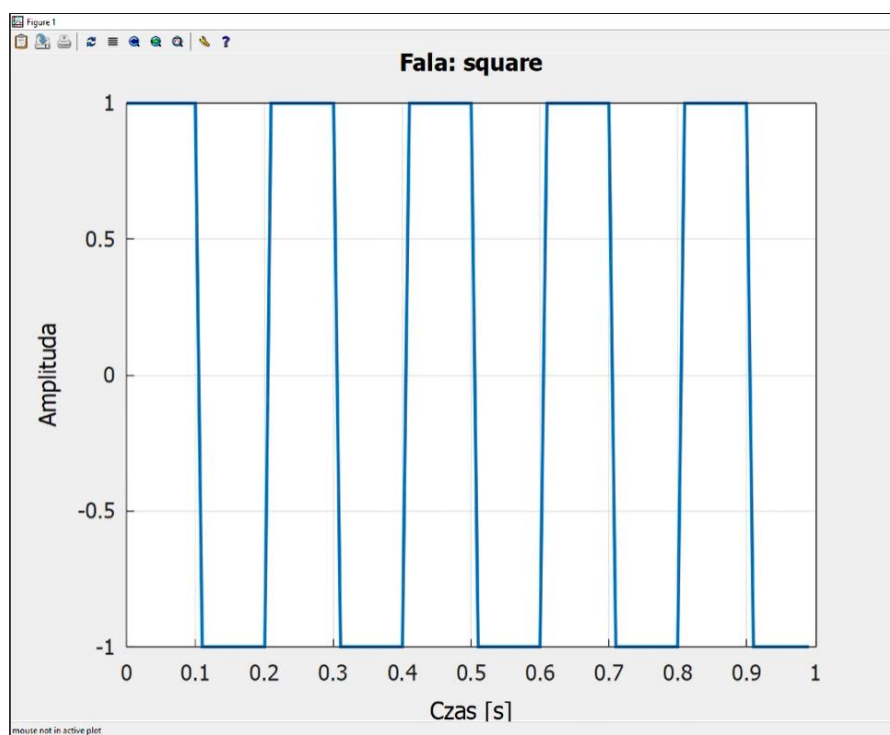
Wyszukuje lokalne maksima (piki) w jednowymiarowym sygnale. Porównuje każdą próbkę z jej sąsiadami i zwraca indeksy oraz wartości punktów, które są większe od obu sąsiednich próbek. Jest to przydatne w analizie sygnałów do identyfikacji zdarzeń, cykli lub punktów charakterystycznych.

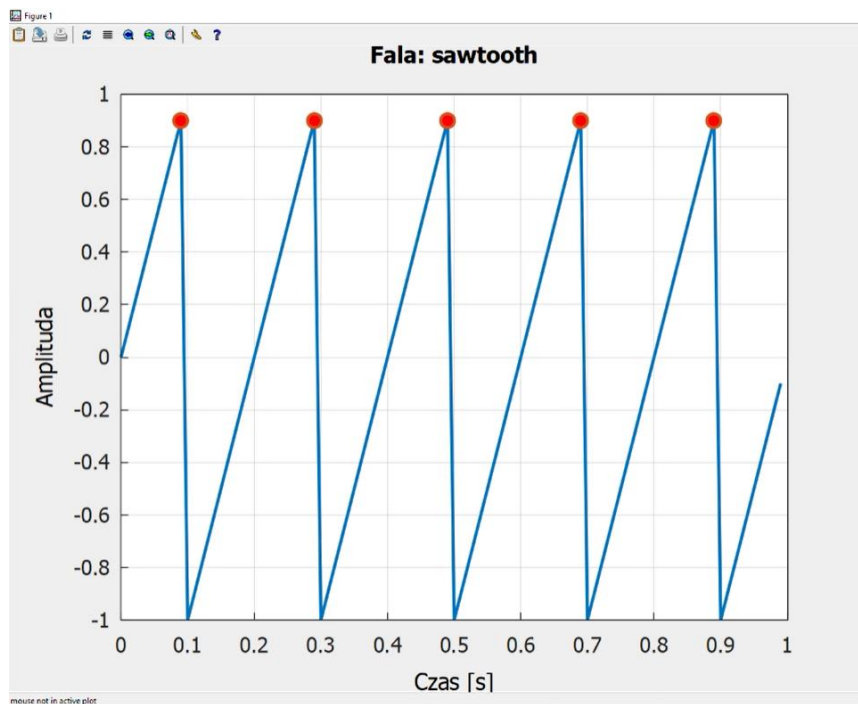


```
Wykryto 5 pików:  
czas: 0.05, wartosc: 1  
czas: 0.25, wartosc: 1  
czas: 0.45, wartosc: 1  
czas: 0.65, wartosc: 1  
czas: 0.85, wartosc: 1
```



```
Wykryto 4 pik.: w:  
czas: 0.2, wartosc: 1  
czas: 0.4, wartosc: 1  
czas: 0.6, wartosc: 1  
czas: 0.8, wartosc: 1
```





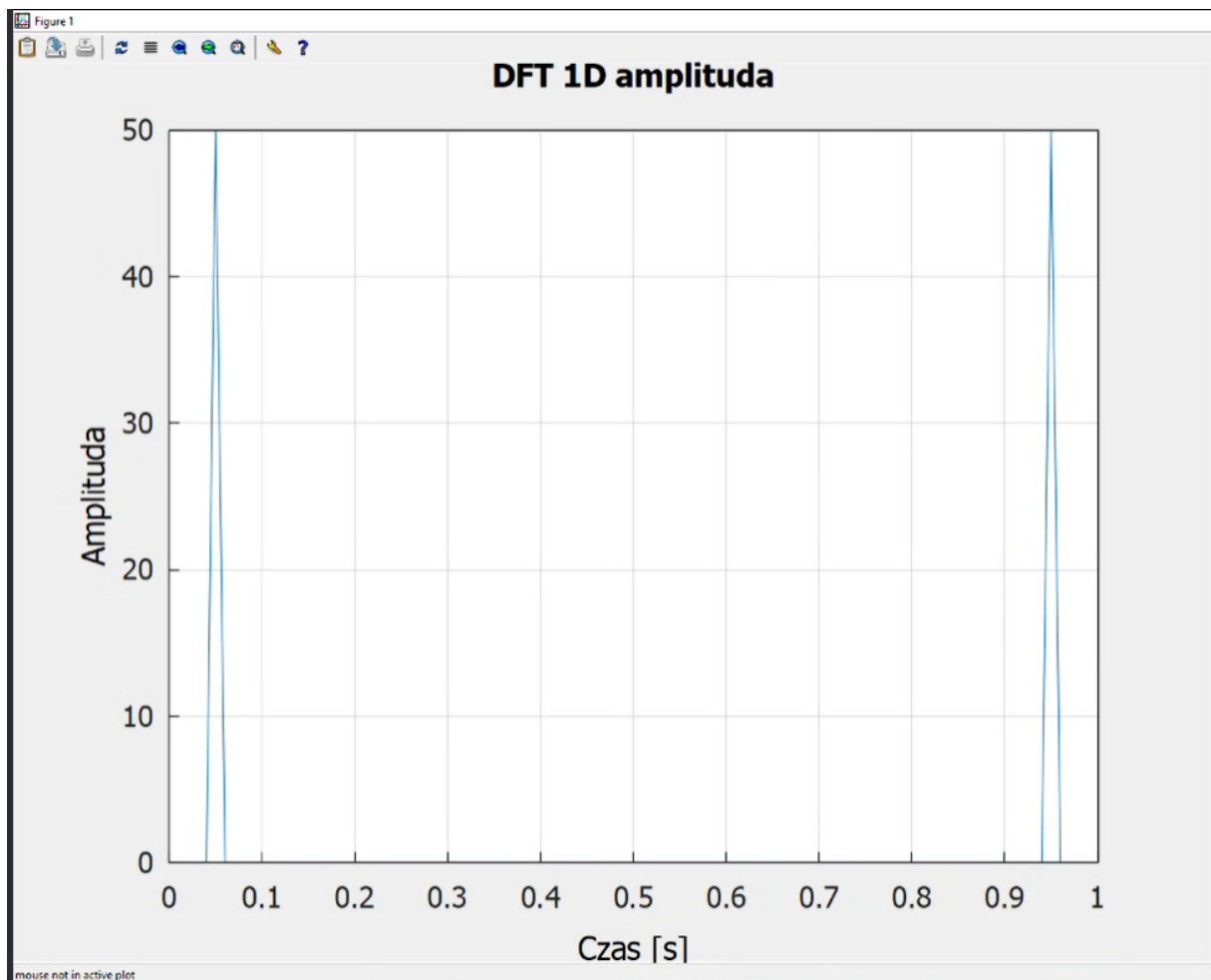
Wykryto 5 pik.: w:

czas: 0.09,	wartosc: 0.9
czas: 0.29,	wartosc: 0.9
czas: 0.49,	wartosc: 0.9
czas: 0.69,	wartosc: 0.9
czas: 0.89,	wartosc: 0.9

```
std::vector<Complex> dft(const std::vector<double>& in)
```

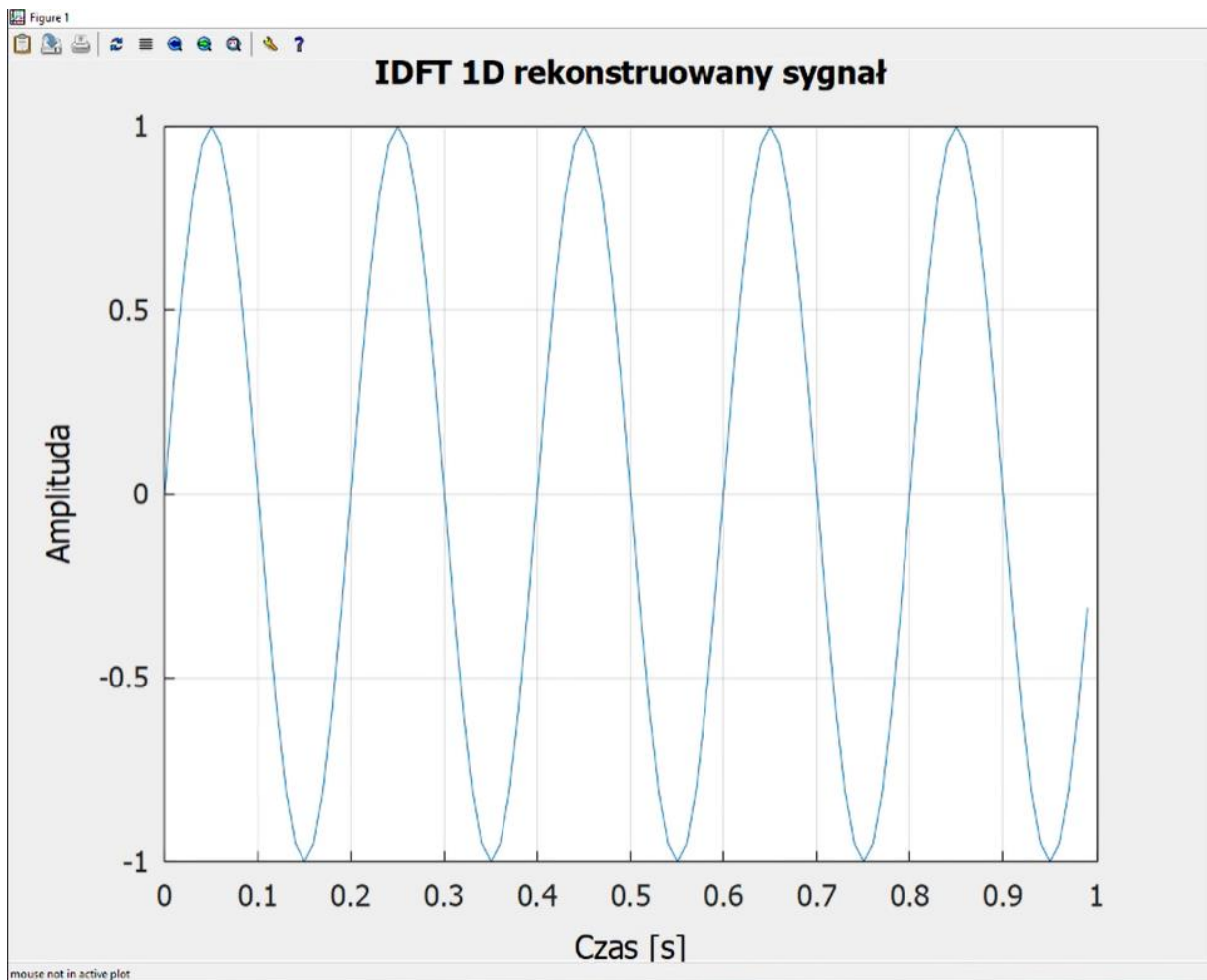
Funkcja ta realizuje dyskretną transformację Fouriera (DFT) dla jednowymiarowego sygnału rzeczywistego. Przekształca sygnał z dziedziny czasu do dziedziny częstotliwości, zwracając wektor liczb zespolonych reprezentujących amplitudy i fazy poszczególnych częstotliwości.

Transformacja ta pozwala analizować widmo częstotliwościowe sygnału, co jest użyteczne np. w filtracji lub kompresji danych.



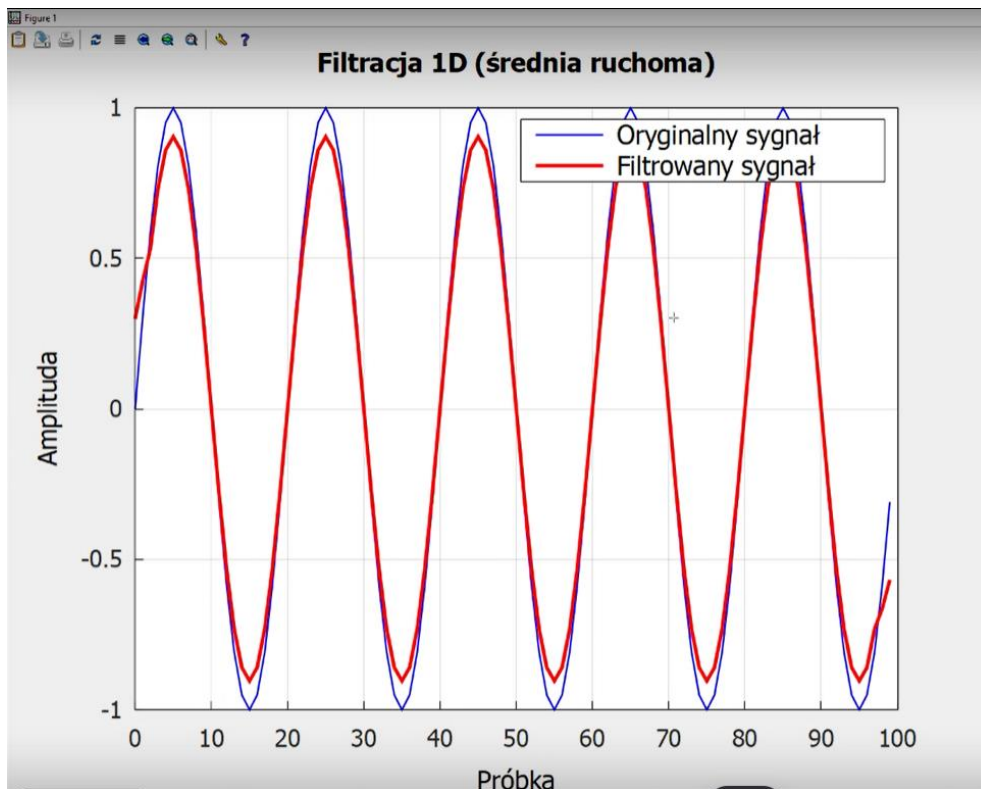
`std::vector<double> idft(const std::vector<Complex>& in)`

Ta funkcja oblicza odwrotną transformację Fouriera (IDFT) dla sygnału zespolonego. Służy do przekształcenia sygnału z powrotem z dziedziny częstotliwości do dziedziny czasu. Wynikowy sygnał jest rzeczywisty, zakładając, że dane wejściowe były wynikiem transformacji sygnału rzeczywistego. Algorytm wykorzystuje wzór odwrotnej DFT i dzieli wynik przez N, czyli liczbę próbek.



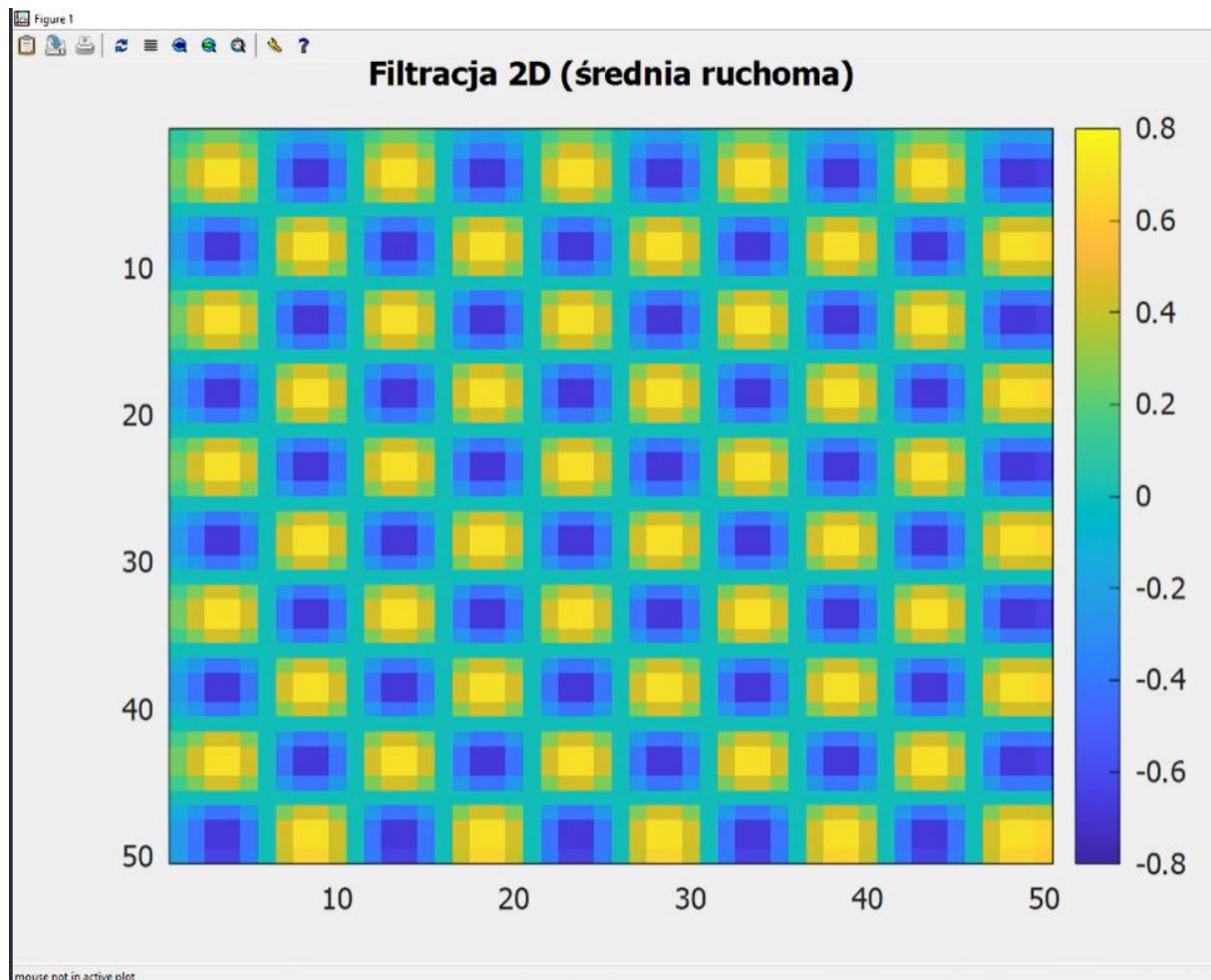
```
std::vector<double> filter_1d(const std::vector<double>& signal, int  
window_size = 5)
```

Funkcja realizuje prosty filtr dolnoprzepustowy w postaci średniej ruchomej (ang. moving average) dla jednowymiarowego sygnału. Dla każdej próbki sygnału obliczana jest średnia wartość w oknie o zadanym rozmiarze, co pozwala na wygładzenie danych i eliminację wysokoczęstotliwościowego szumu. Funkcja dodatkowo rysuje wykres porównujący oryginalny sygnał z przefiltrowanym.



```
std::vector<std::vector<double>> filter_2d(const  
std::vector<std::vector<double>>& signal, int window_size = 3)
```

Funkcja wykonuje filtrację 2D za pomocą średniej ruchomej na sygnale reprezentowanym jako macierz (np. obraz lub mapa danych). Dla każdego punktu obliczana jest średnia z jego sąsiedztwa w kwadratowym oknie. Jest to analog filtracji 1D, ale rozszerzony na dwuwymiarowe dane. Na końcu funkcja rysuje obraz w postaci mapy kolorów (heatmapy), co ułatwia analizę efektu wygładzania.



Plik CMakeLists.txt zawierający załączenie bibliotek pybind11 oraz matplotlibplus do projektu.

```
CMakeLists.txt
1  cmake_minimum_required(VERSION 3.14)
2  project(hello_pybind)
3  add_subdirectory(pybind11)
4  add_subdirectory(matplotlibplus)
5  set(CMAKE_CXX_STANDARD 17)
6  include_directories("C:/Users/kutom/Desktop/prjk3/pybind11/include")
7  include_directories("C:/Users/kutom/AppData/Local/Programs/Python/Python311/include")
8  include_directories("C:/Users/kutom/Desktop/prjk3/matplotlibplus/source/matplot")
9  include_directories("C:/Users/kutom/Desktop/prjk3/pybind11/include")
10 pybind11_add_module(example example.cpp)
11
12 target_link_libraries(example PUBLIC matplot)
```