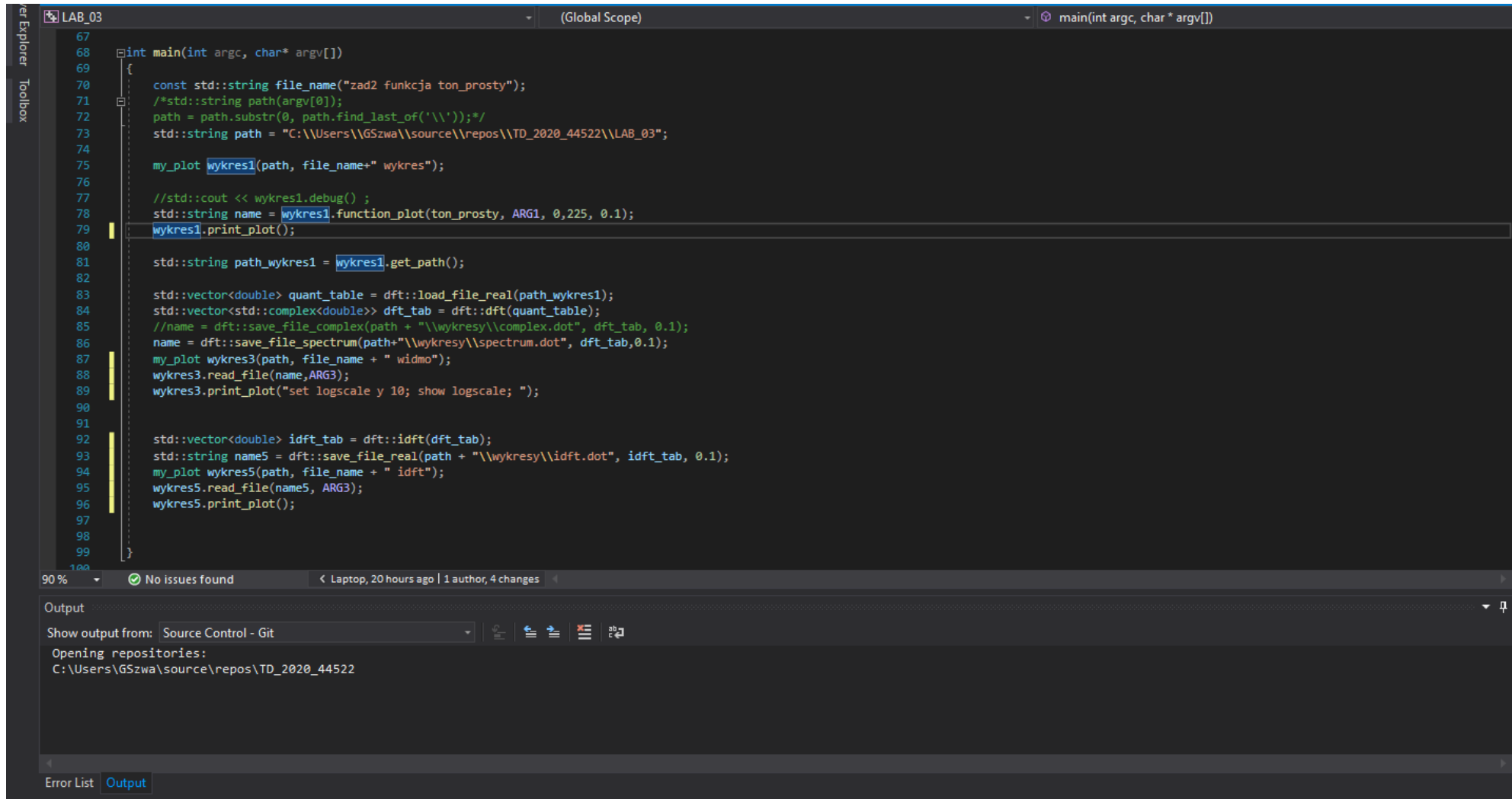


## Sprawozdanie z LAB 01

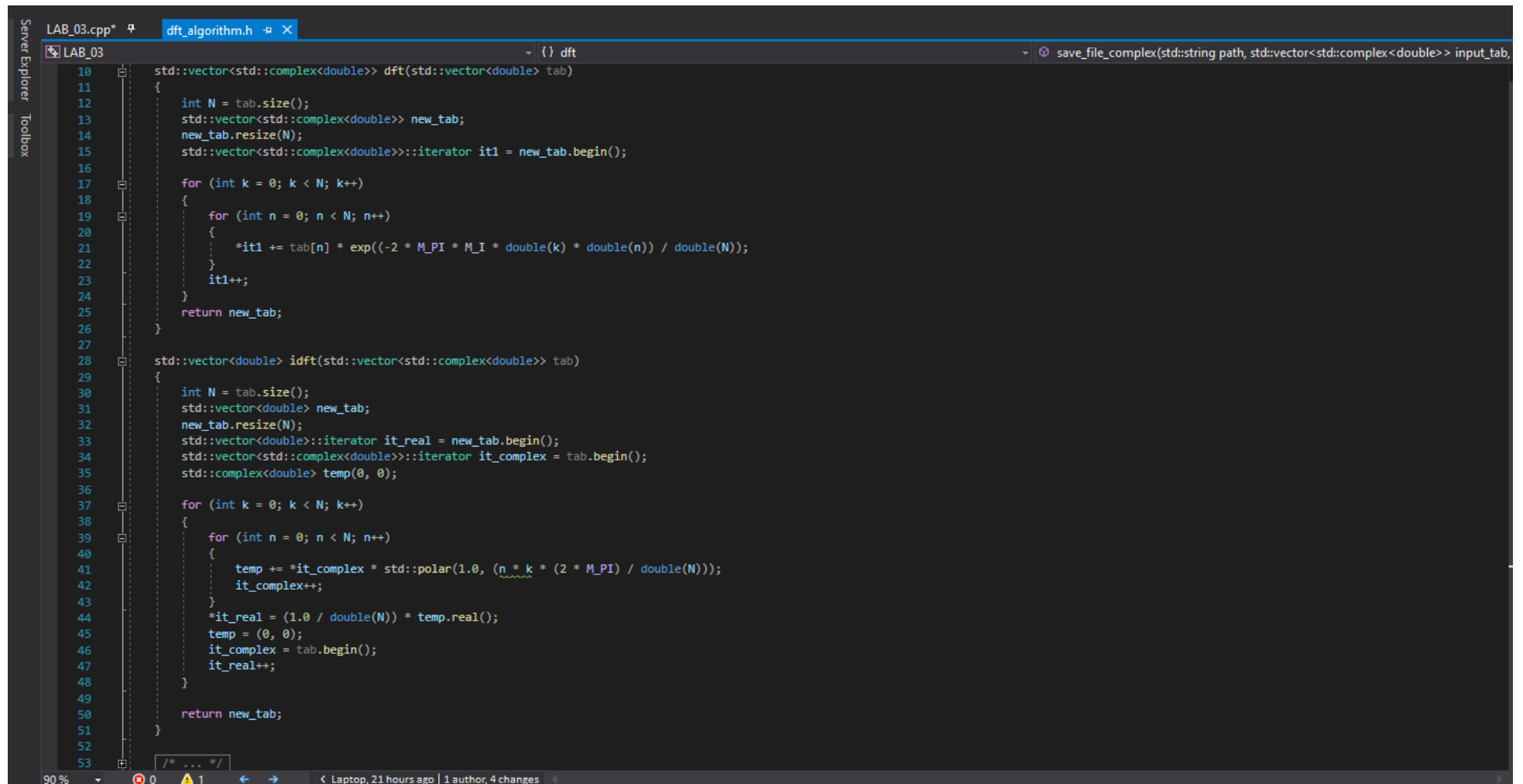
Zadaniem zajęć było wykonanie Dyskretnej Transformaty Fouriera.

Oto kod źródłowy rysujący funkcję:



```
LAB_03 (Global Scope) main(int argc, char * argv[])
67
68 int main(int argc, char* argv[])
69 {
70     const std::string file_name("zad2 funkcja ton_prosty");
71     /*std::string path(argv[0]);
72     path = path.substr(0, path.find_last_of('\\'));*/
73     std::string path = "C:\\Users\\GSzwa\\source\\repos\\TD_2020_44522\\LAB_03";
74
75     my_plot wykres1(path, file_name+" wykres");
76
77     //std::cout << wykres1.debug() ;
78     std::string name = wykres1.function_plot(ton_prosty, ARG1, 0,225, 0.1);
79     wykres1.print_plot();
80
81     std::string path_wykres1 = wykres1.get_path();
82
83     std::vector<double> quant_table = dft::load_file_real(path_wykres1);
84     std::vector<std::complex<double>> dft_tab = dft::dft(quant_table);
85     //name = dft::save_file_complex(path + "\\wykresy\\complex.dot", dft_tab, 0.1);
86     name = dft::save_file_spectrum(path+"\\wykresy\\spectrum.dot", dft_tab,0.1);
87     my_plot wykres3(path, file_name + " widmo");
88     wykres3.read_file(name,ARG3);
89     wykres3.print_plot("set logscale y 10; show logscale; ");
90
91
92     std::vector<double> idft_tab = dft::idft(dft_tab);
93     std::string name5 = dft::save_file_real(path + "\\wykresy\\idft.dot", idft_tab, 0.1);
94     my_plot wykres5(path, file_name + " idft");
95     wykres5.read_file(name5, ARG3);
96     wykres5.print_plot();
97
98
99
100
90 % No issues found < Laptop, 20 hours ago | 1 author, 4 changes
Output
Show output from: Source Control - Git
Opening repositories:
C:\\Users\\GSzwa\\source\\repos\\TD_2020_44522
Error List Output
```

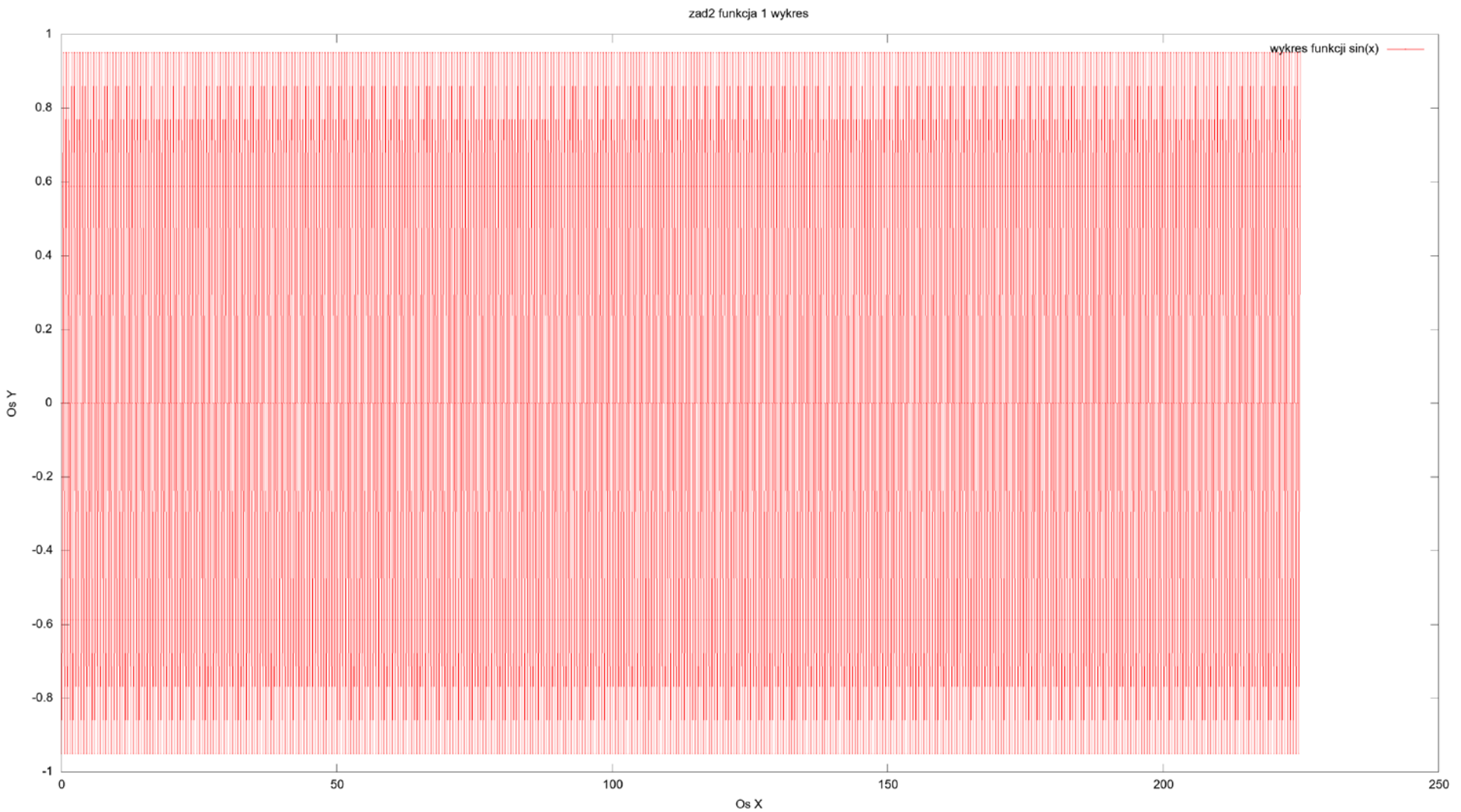
Oraz kod źródłowy funkcji dft oraz idft:



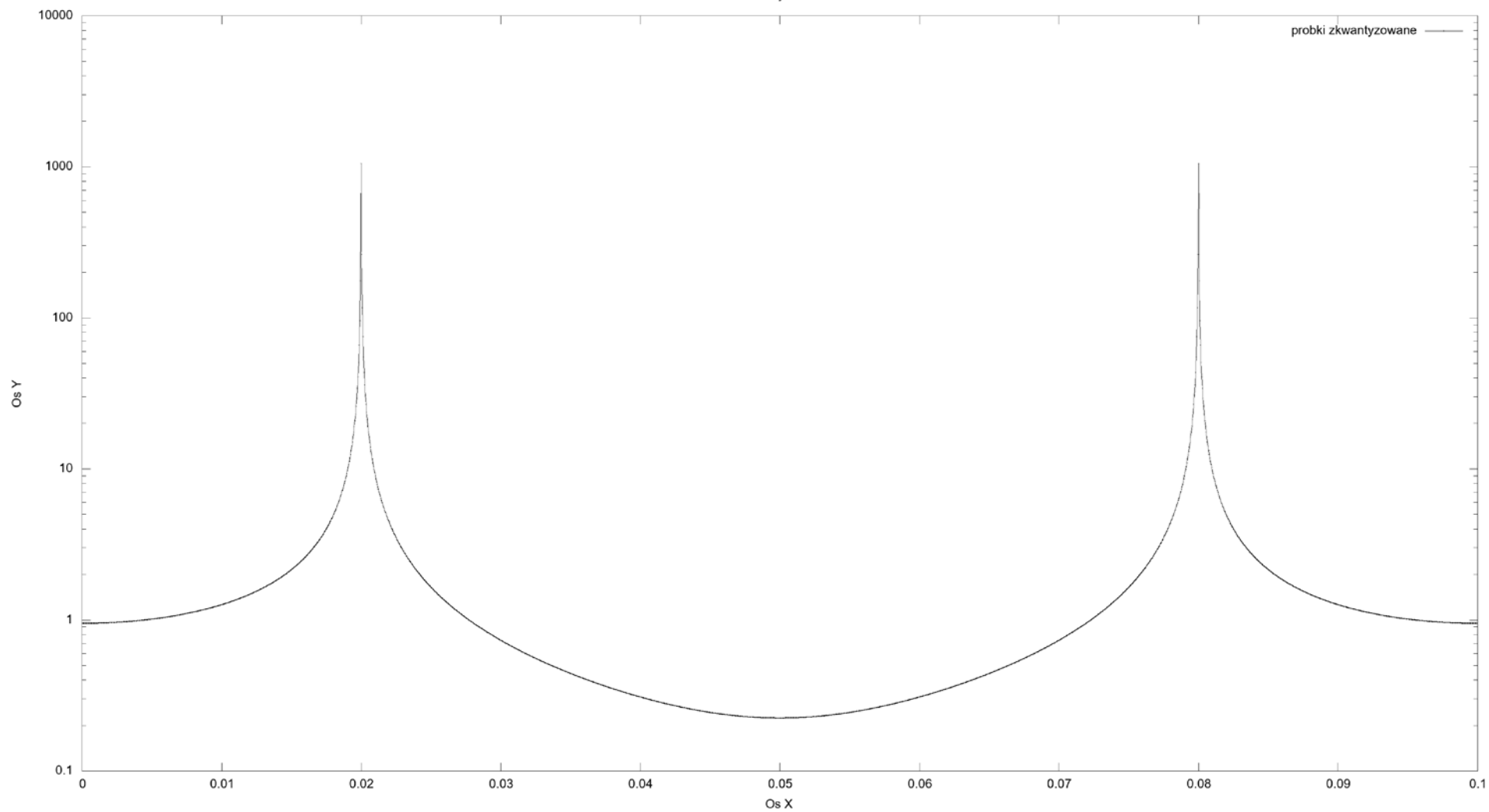
The screenshot shows a C++ IDE with two tabs: 'LAB\_03.cpp\*' and 'dft\_algorithm.h'. The 'dft\_algorithm.h' tab is active, displaying the source code for two functions: `dft` and `idft`. The `dft` function (lines 10-26) takes a `std::vector<double>` and returns a `std::vector<std::complex<double>>`. It uses a nested loop to calculate the Discrete Fourier Transform. The `idft` function (lines 28-53) takes a `std::vector<std::complex<double>>` and returns a `std::vector<double>`. It uses a nested loop to calculate the Inverse Discrete Fourier Transform, including a normalization factor of `1.0 / double(N)`. The IDE interface includes a 'Server Explorer' and 'Toolbox' on the left, a status bar at the bottom showing '90%' zoom and '1' warning, and a file explorer on the right showing the project structure.

```
10 std::vector<std::complex<double>> dft(std::vector<double> tab)
11 {
12     int N = tab.size();
13     std::vector<std::complex<double>> new_tab;
14     new_tab.resize(N);
15     std::vector<std::complex<double>>::iterator it1 = new_tab.begin();
16
17     for (int k = 0; k < N; k++)
18     {
19         for (int n = 0; n < N; n++)
20         {
21             *it1 += tab[n] * exp((-2 * M_PI * M_I * double(k) * double(n)) / double(N));
22         }
23         it1++;
24     }
25     return new_tab;
26 }
27
28 std::vector<double> idft(std::vector<std::complex<double>> tab)
29 {
30     int N = tab.size();
31     std::vector<double> new_tab;
32     new_tab.resize(N);
33     std::vector<double>::iterator it_real = new_tab.begin();
34     std::vector<std::complex<double>>::iterator it_complex = tab.begin();
35     std::complex<double> temp(0, 0);
36
37     for (int k = 0; k < N; k++)
38     {
39         for (int n = 0; n < N; n++)
40         {
41             temp += *it_complex * std::polar(1.0, (n * k * (2 * M_PI) / double(N)));
42             it_complex++;
43         }
44         *it_real = (1.0 / double(N)) * temp.real();
45         temp = (0, 0);
46         it_complex = tab.begin();
47         it_real++;
48     }
49
50     return new_tab;
51 }
52
53 /* ... */
```

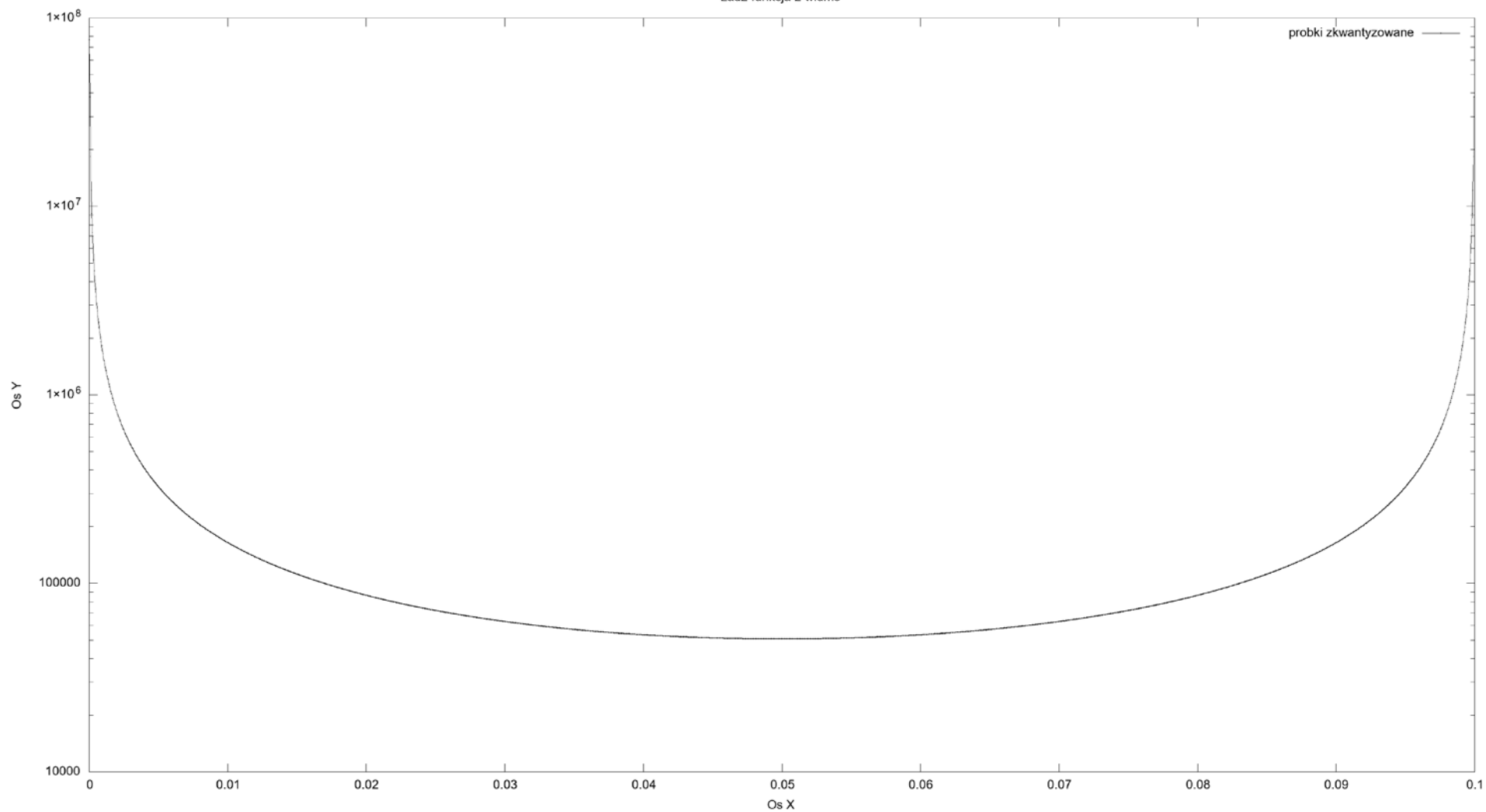
Podczas zajęć sporządziłem 11 wykresów.

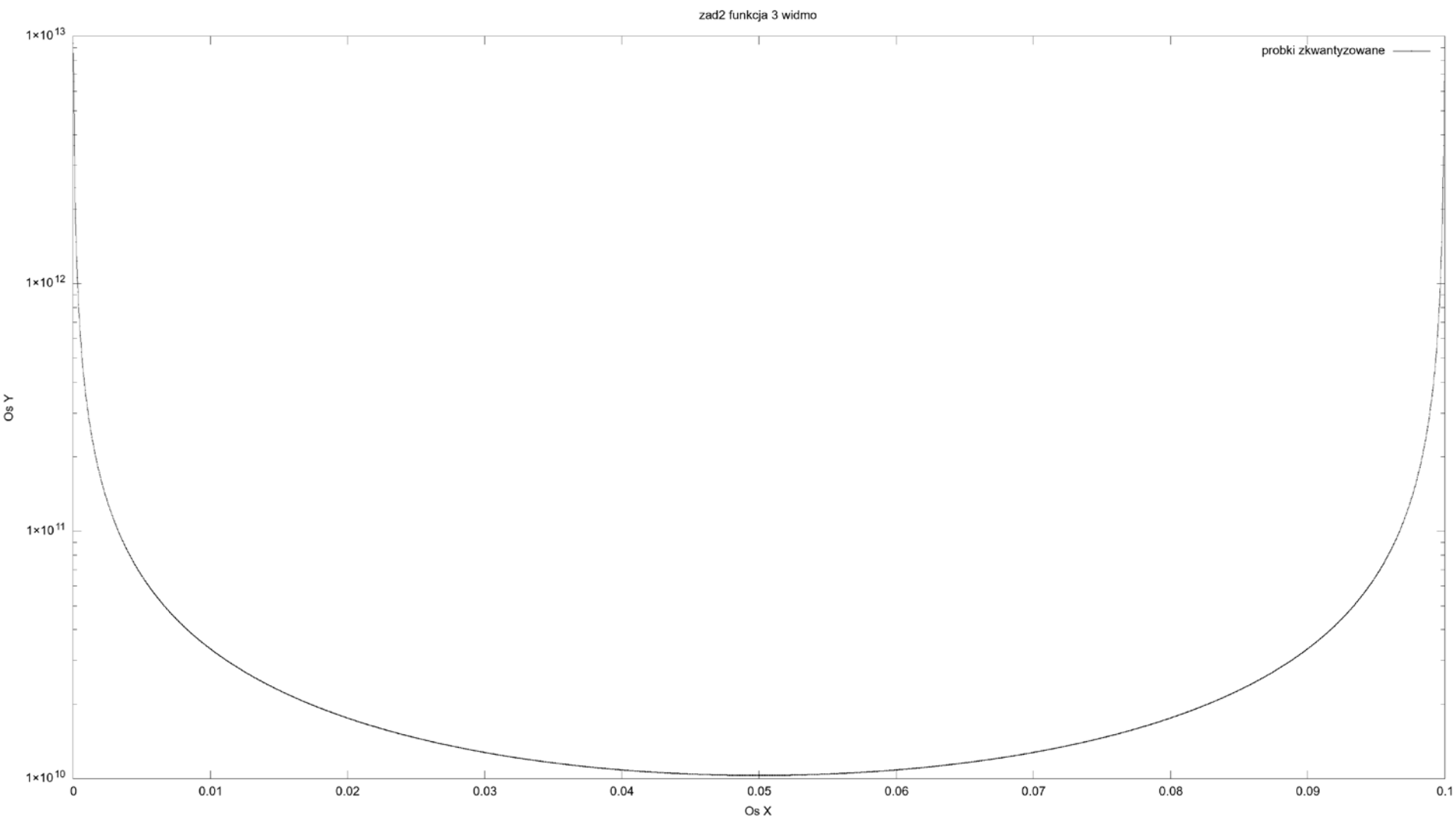


zad2 funkcja 1 widmo

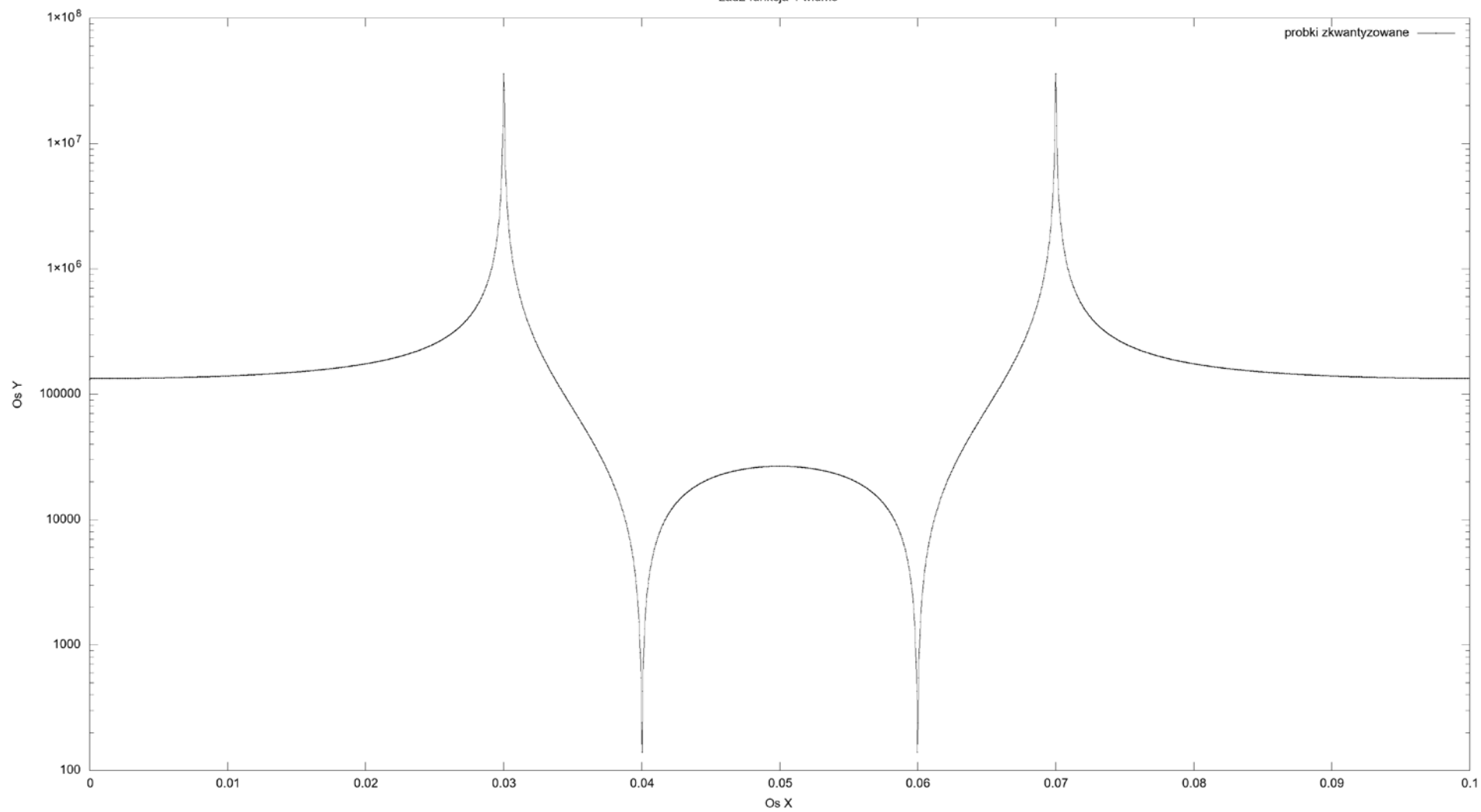


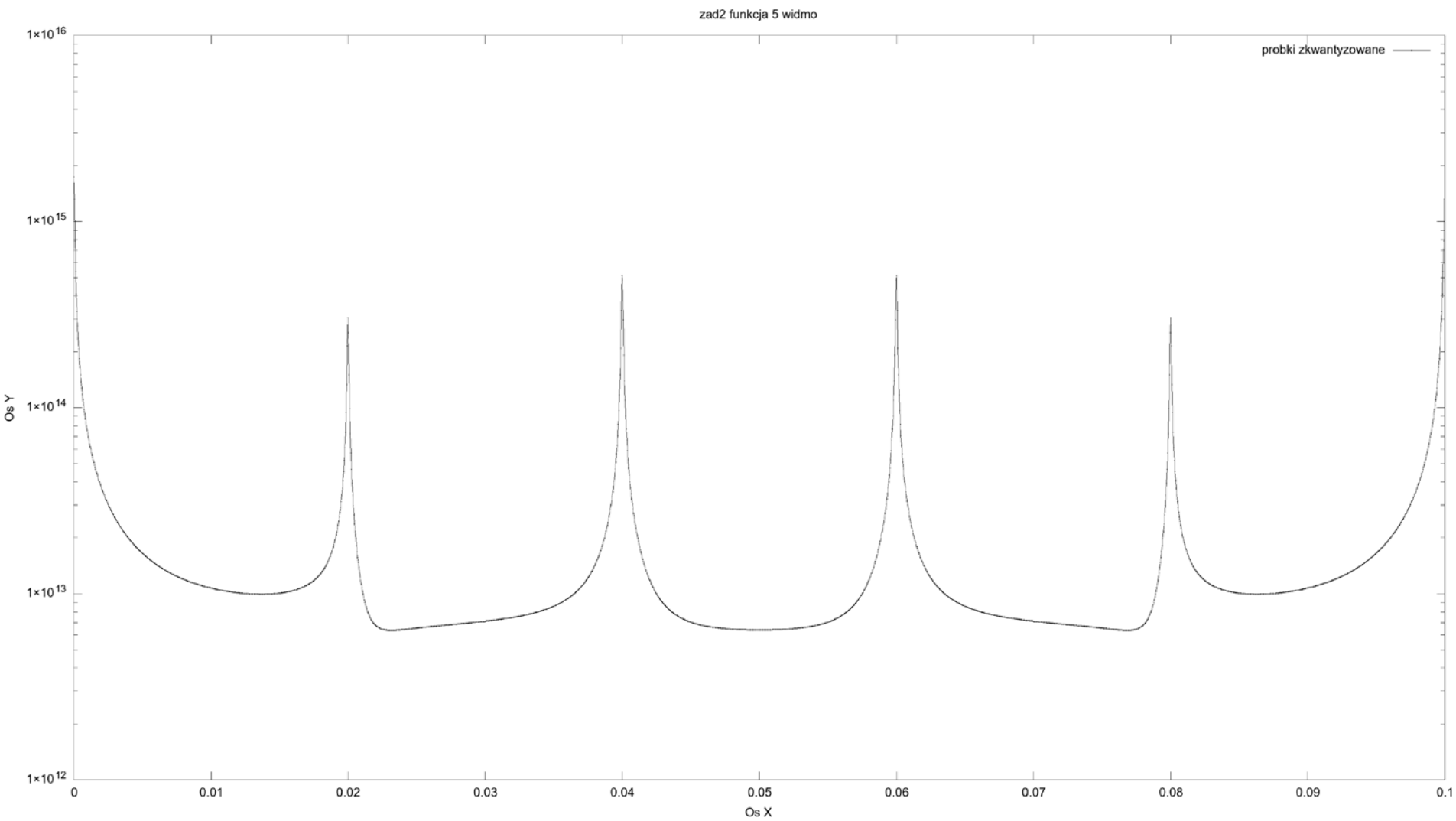
zad2 funkcja 2 widmo





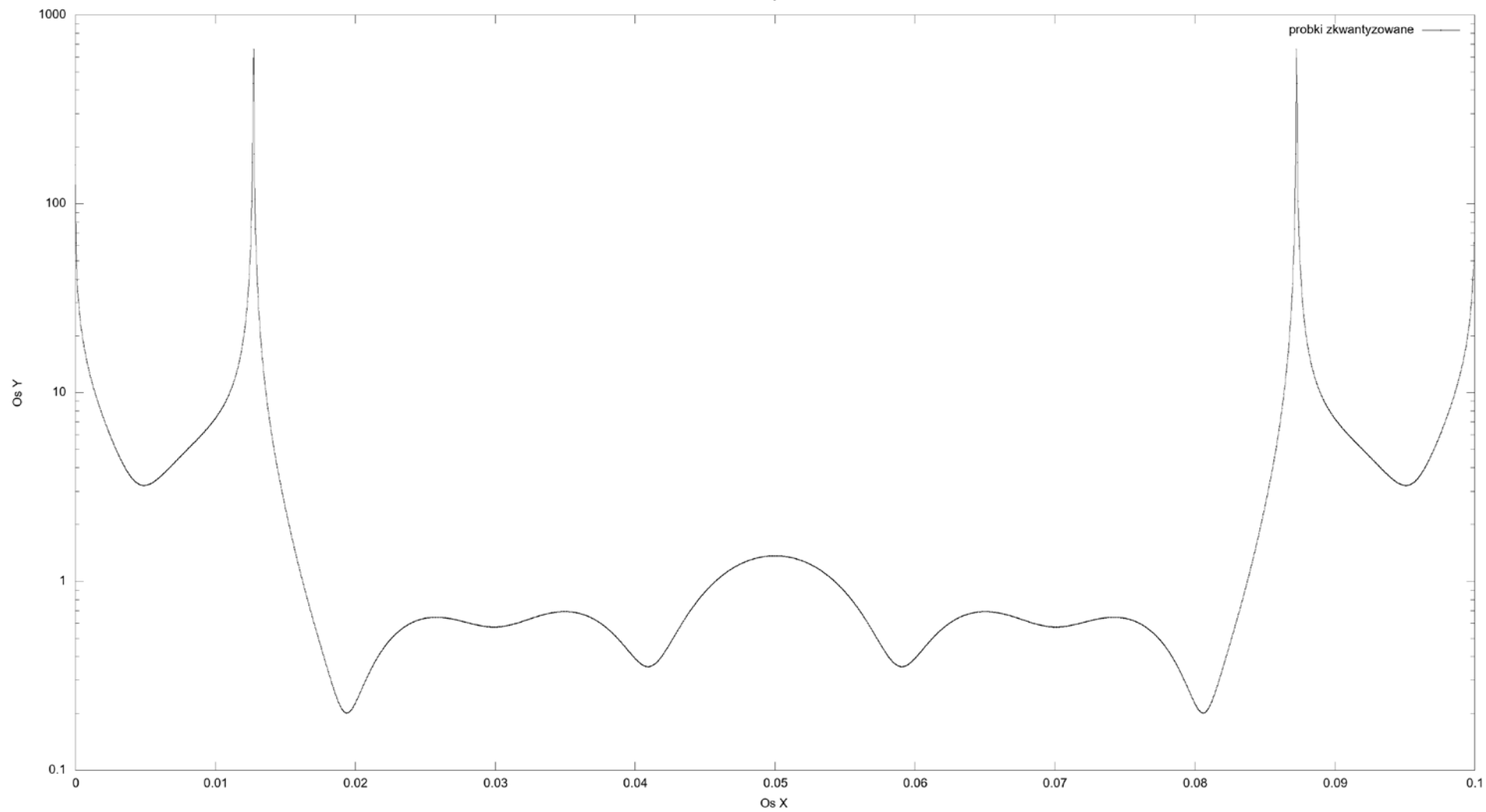
zad2 funkcja 4 widmo



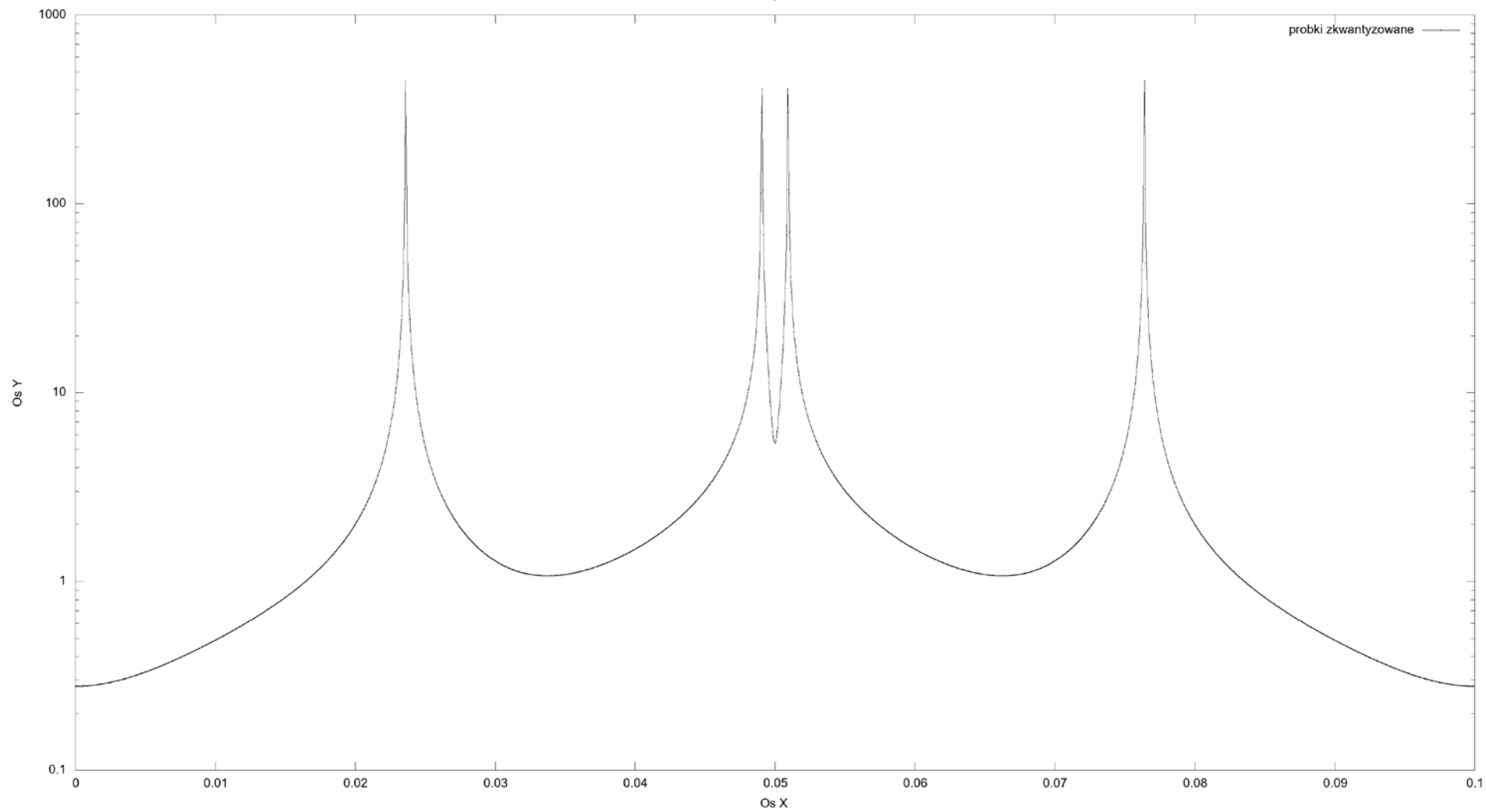




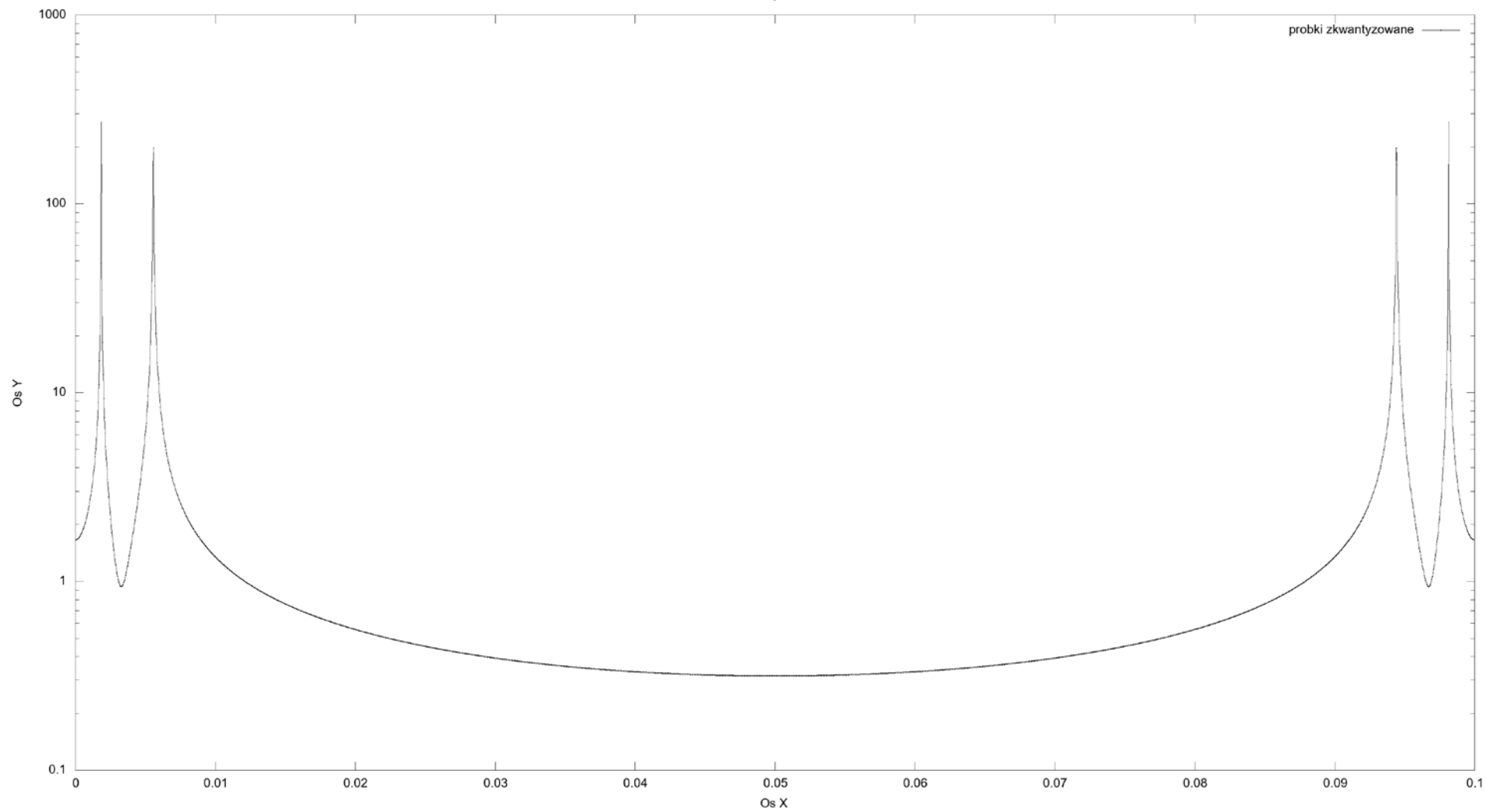
zad2 funkcja 6 widmo



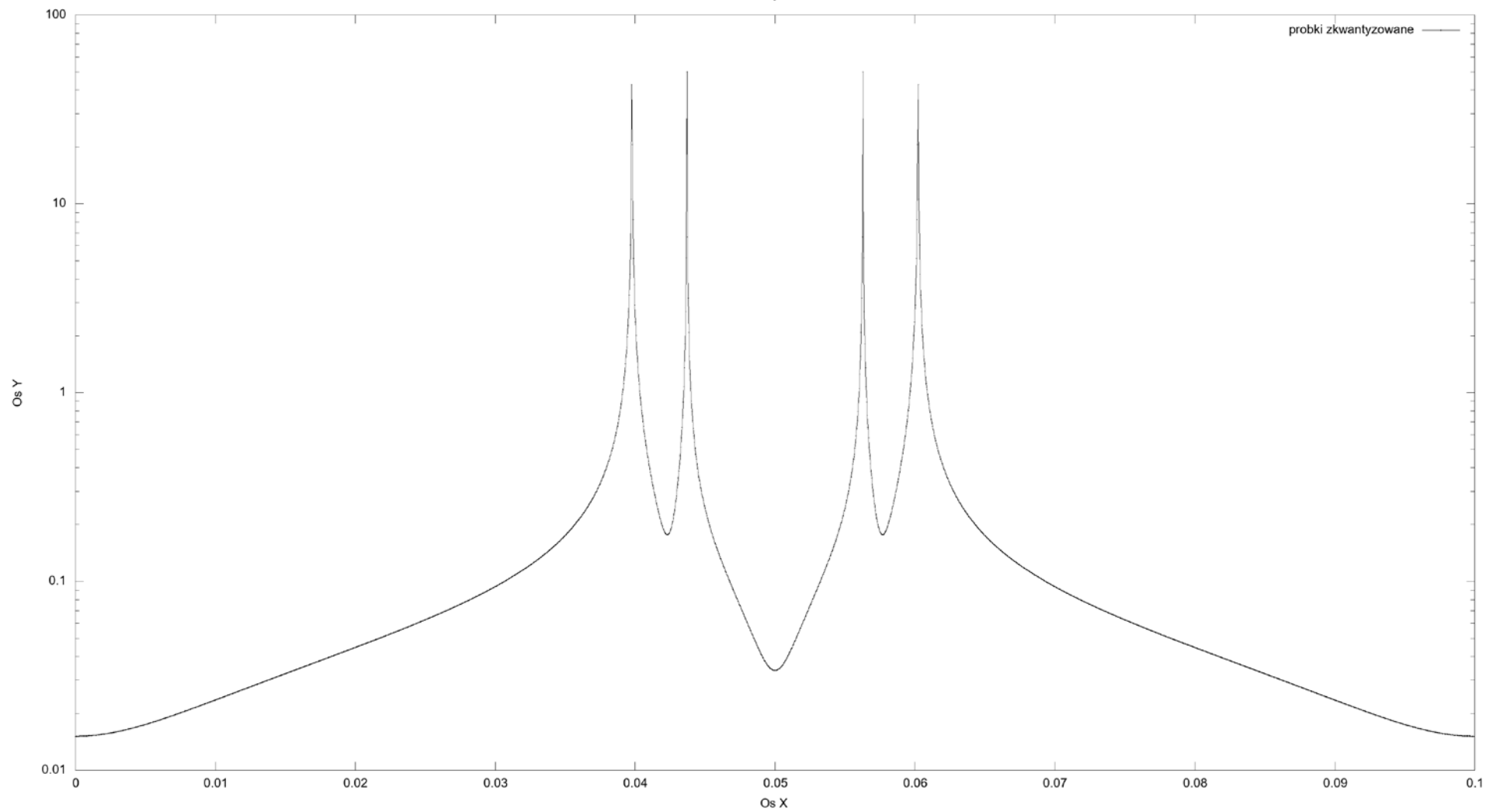
zad2 funkcja 7 widmo

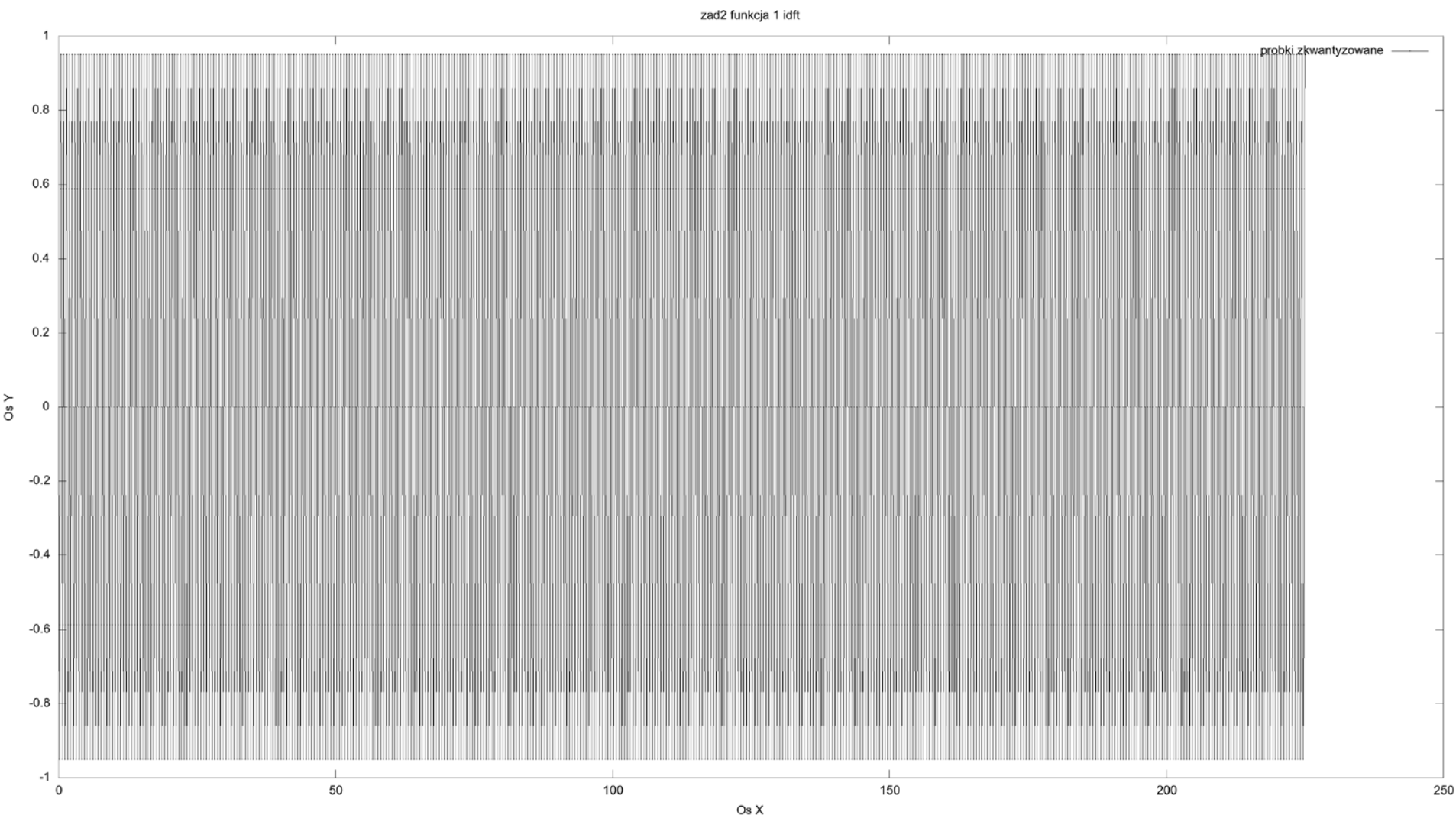


zad2 funkcja 8 widmo



zad2 funkcja 9 widmo





## **Podsumowanie**

Dzięki tym laboratoriom nauczyłem się algorytmu DFT oraz IDFT, poznałem jak zamienić funkcję czasu na funkcję częstotliwości , co pozwala na zrozumienie w jaki sposób jest wysyłany sygnał za pomocą analogowego ośrodka..

Wykonał Szwarz Grzegorz