

Transmisja Danych – Lab 03

Krystian Bartosik 213A, nr 44266

Kod źródłowy:

```
// Krystian Bartosik
// bk44266@zut.edu.pl
// FEDCBA
#define _USE_MATH_DEFINES
#include <iostream>
#include <fstream>
#include "math.h"
#include <complex>

using namespace std;

complex<double>* DFT(double* Tab, int n)
{
    complex<double>* c = new complex<double>[n];
    complex<double> i = 0.0 + 1.0i;
    for (int k = 0; k < n; k++)
    {
        c[k] = 0.0 + 0.0i;

        for (int j = 0; j < n; j++)
        {
            c[k] = c[k] + (Tab[j] * exp(-2 * M_PI * i * (double)k * (double)j / double(n)));
        }
    }
    return c;
}

double * IDFT(complex<double>* Tab, int n)
{
    double* c = new double[n];
    complex<double> i = 0.0 + 1.0i;
    for (int k = 0; k < n; k++)
    {
        c[k] = 0;

        for (int j = 0; j < n; j++)
        {
            c[k] = c[k] + (Tab[j] * exp(2 * M_PI * i * (double)k * (double)j / double(n))).real();
        }
        c[k] = (1.0 / n) * c[k];
    }
    return c;
}

double TonProsty(double A, double f, double Fi, double t)
{
    return A * sin(2 * M_PI * f * t + Fi);
}

double x(double t)
{
    return pow(6 * t, 2) + 6 * t + 2;
}
```

```

double y(double t)
{
    return 2 * pow(x(t), 2) + 12 * cos(t);
}

double z(double t)
{
    return sin(2 * M_PI * 7 * t) * x(t) - 0.2 * log10(abs(y(t)) + M_PI);
}

double u(double t)
{
    return sqrt(abs(y(t) * y(t) * z(t))) - 1.8 * sin(0.4 * t * z(t) * x(t));
}

double v(double t)
{
    if ((0.22 > t) && (t >= 0))
    {
        return (1 - 7 * t) * sin((2 * M_PI * t * 10) / (t + 0.04));
    }

    if ((0.22 <= t) && (t < 0.7))
    {
        return 0.63 * t * sin((2 * M_PI * t * 10) / (t + 0.04));
    }

    if ((1 >= t) && (t >= 0.7))
    {
        return pow(t, -0.662) + 0.77 * sin(8 * t);
    }
}

double p(double t, int N)
{
    double SUM = 0;

    for (int n = 1; n <= N; n++)
    {
        SUM = SUM + ((cos(12 * t * pow(n, 2)) + cos(16 * t * n)) / (pow(n, 2)));
    }

    return SUM;
}

int main()
{
    fstream File;
    File.open("C:/Users/Qrystian/Desktop/results.txt", ios::out);
    double* Tab;
    double* M;
    double* Y;
    complex<double>* X;

    int size = (unsigned int)(662.0 / 0.1);
    Tab = new double[size];
    M = new double[size];

```

```

int j = 0;
for (double t = 0.0; t <= 662.0; t = t + 0.1)
{
    Tab[j] = TonProsty(1.0, 6, 2 * M_PI, t);
    //Tab[j] = x(t);
    //Tab[j] = y(t);
    //Tab[j] = z(t);
    //Tab[j] = u(t);
    //Tab[j] = v(t);
    //Tab[j] = p(t, 2);
    //Tab[j] = p(t, 4);
    //Tab[j] = p(t, 66);
    j++;
}
size = (unsigned int)(662.0 / 0.1);
X = DFT(Tab, size);
Y = IDFT(X, size);

for (j = 0; j < size; j++)
{
    M[j] = sqrt(pow(X[j].real(), 2) + pow(X[j].imag(), 2));
    M[j] = 10 * log10(M[j]);
}

j = 0;
for (double t = 0.0; t <= 662.0; t = t + 0.1)
{
    //File << t << " " << TonProsty(1.0, 6, 2 * M_PI, t) << endl;
    //File << t << " " << M[j] << endl;
    File << t << " " << Y[j] << endl;
    j++;
}

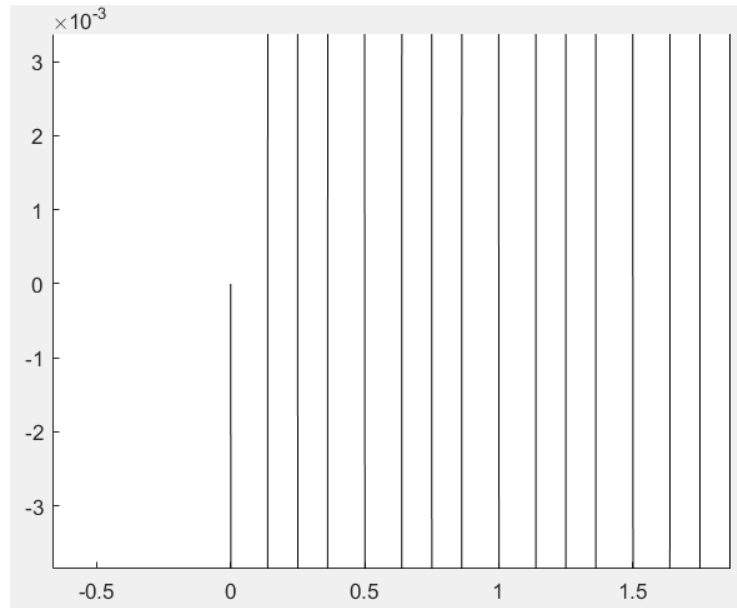
File.close();
}

```

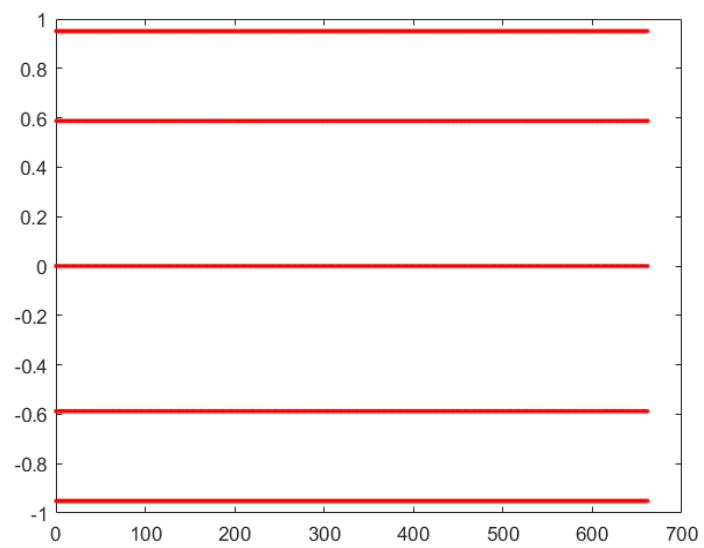
Legenda:

- Zadanie 1 (Obliczenia)
- Zadanie 2 (Obliczenia)
- Zadanie 2.1 (Wykres Tonu Prostego dla n należącego <0;ABC>)
- Zadanie 2.2 (Wykres widma amplitudowego dla Tonu Prostego)
- Zadanie 3.1 (Wykres widma amplitudowego dla x())
- Zadanie 3.2 (Wykres widma amplitudowego dla y())
- Zadanie 3.3 (Wykres widma amplitudowego dla z())
- Zadanie 3.4 (Wykres widma amplitudowego dla u())
- Zadanie 3.5 (Wykres widma amplitudowego dla v())
- Zadanie 3.6 (Wykres widma amplitudowego dla p() dla N=2)
- Zadanie 3.7 (Wykres widma amplitudowego dla p() dla N=4)
- Zadanie 3.8 (Wykres widma amplitudowego dla y() dla N=66)
- Zadanie 4 (Obliczenia)
- Zadanie 4.1 (Wykres odwracający)

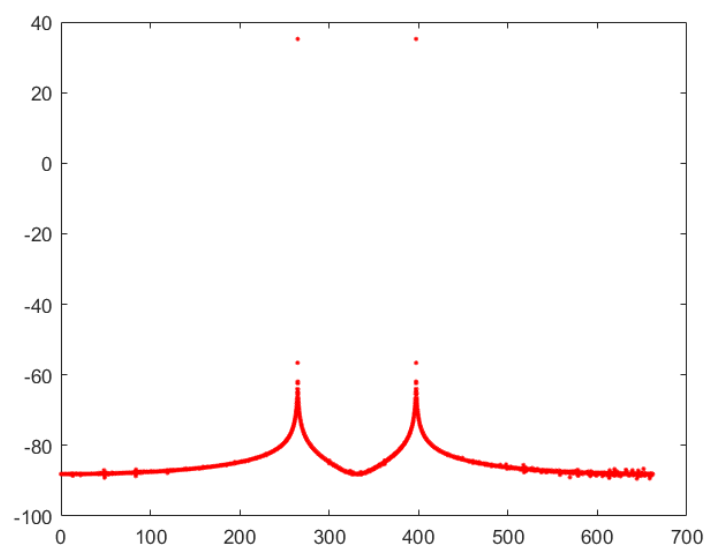
Poniżej zaprezentowane są odpowiednie wykresy. Wykres 1 to bardzo gęsta funkcja sinus, która po oddaleniu wygląda jak pionowe linie. Po przybliżeniu i połączeniu próbek, funkcja wygląda prawidłowo, prezentacja na Rysunku 1:



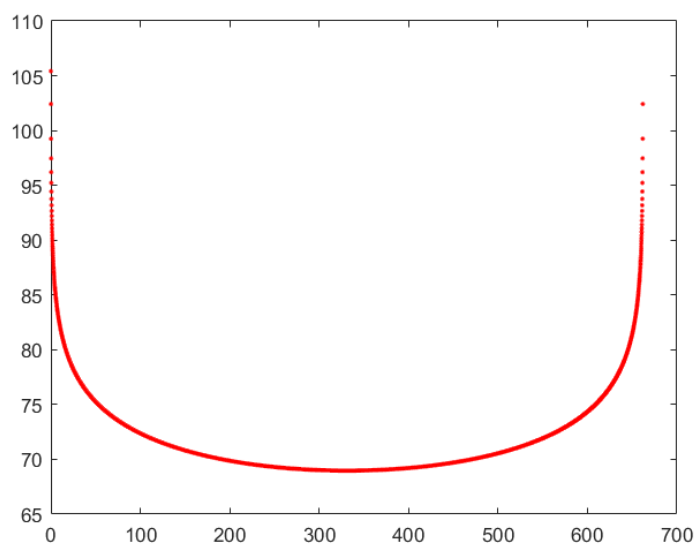
Rysunek 1 Przybliżony widok funkcji



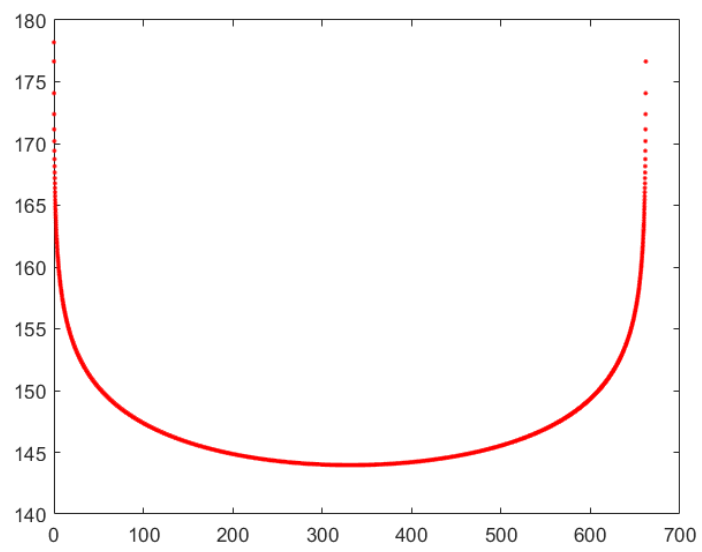
Wykres 1
Zadanie 2.1



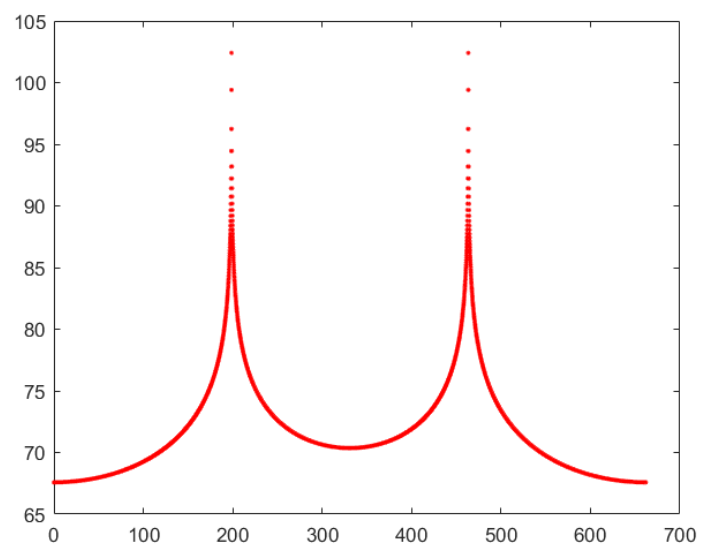
Wykres 2
Zadanie 2.2



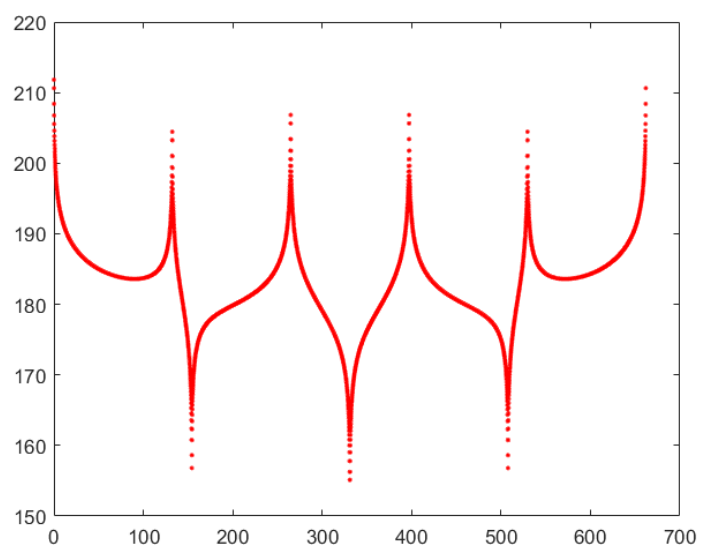
Wykres 3
Zadanie 3.1



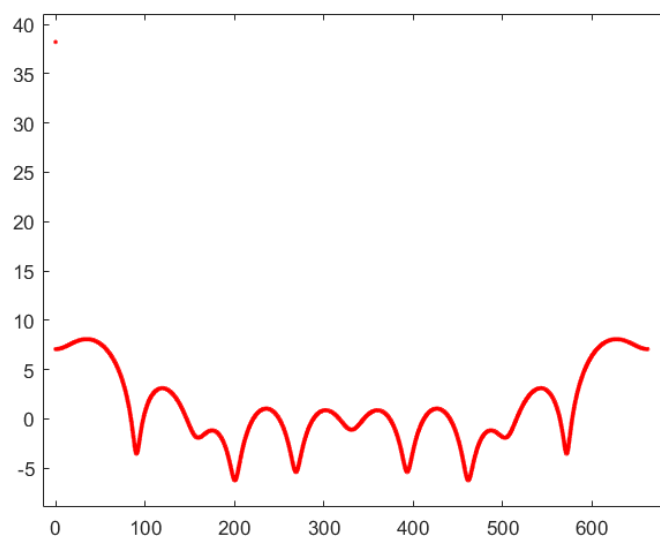
Wykres 4
Zadanie 3.2



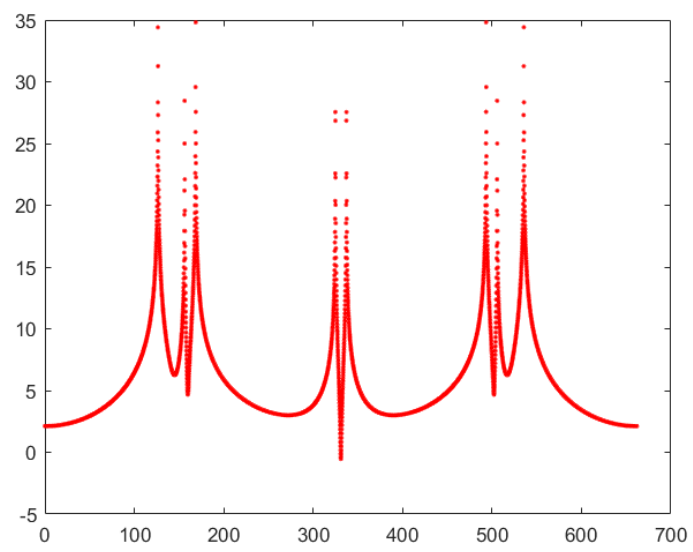
Wykres 5
Zadanie 3.3



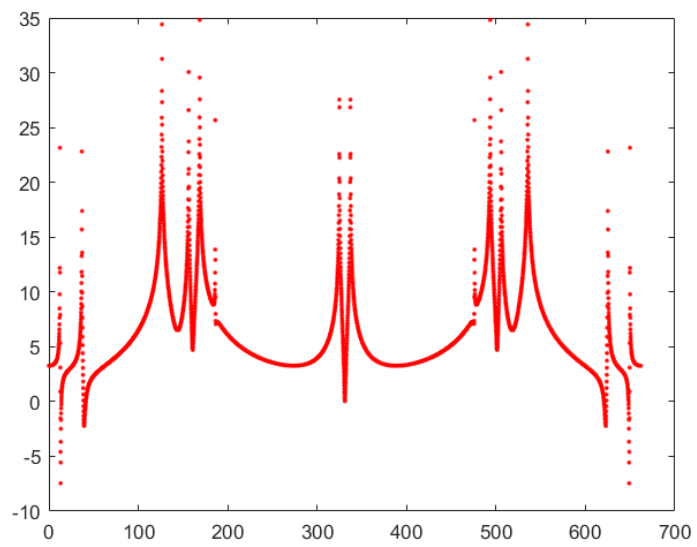
Wykres 6
Zadanie 3.4



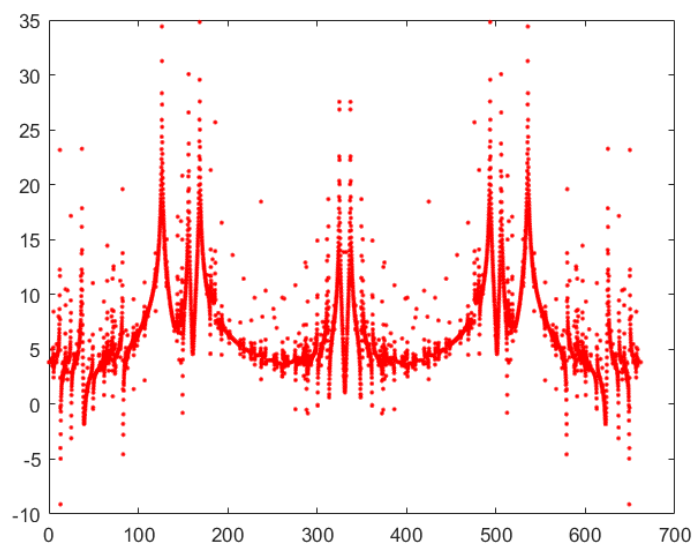
Wykres 7
Zadanie 3.5



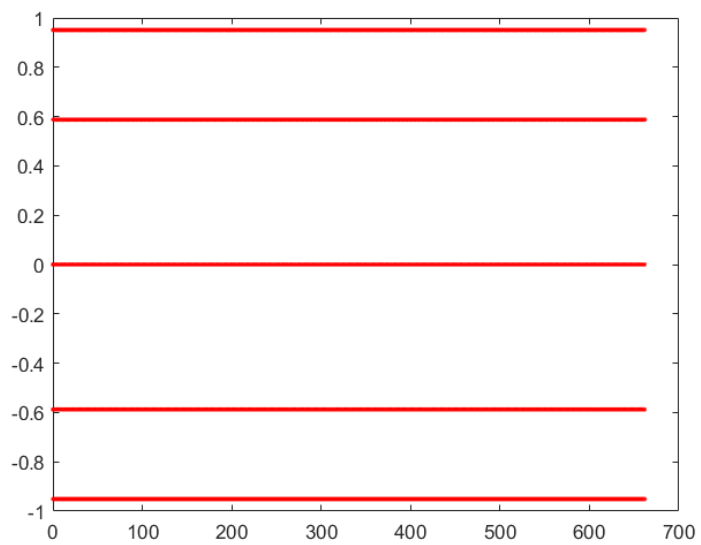
Wykres 8
Zadanie 3.6



Wykres 9
Zadanie 3.7



Wykres 10
Zadanie 3.8



Wykres 11
Zadanie 4.1