

Sprawozdanie z laboratorium  
Metody Numeryczne

# **Interpolacja**

Data odbycia laboratorium: 26-04-2018  
Data oddania sprawozdania: 14-05-2018

Grupa 6  
Sekcja 11  
Martyna Drabińska  
Mateusz Kawulok

## 1. Treść zadania

Napisać program wyznaczający wartości wielomianu interpolacyjnego Lagrange'a w punktach leżących w przedziale  $\langle a; b \rangle$  dla funkcji interpolowanej:

$$f(x) = |\sin(x)|$$

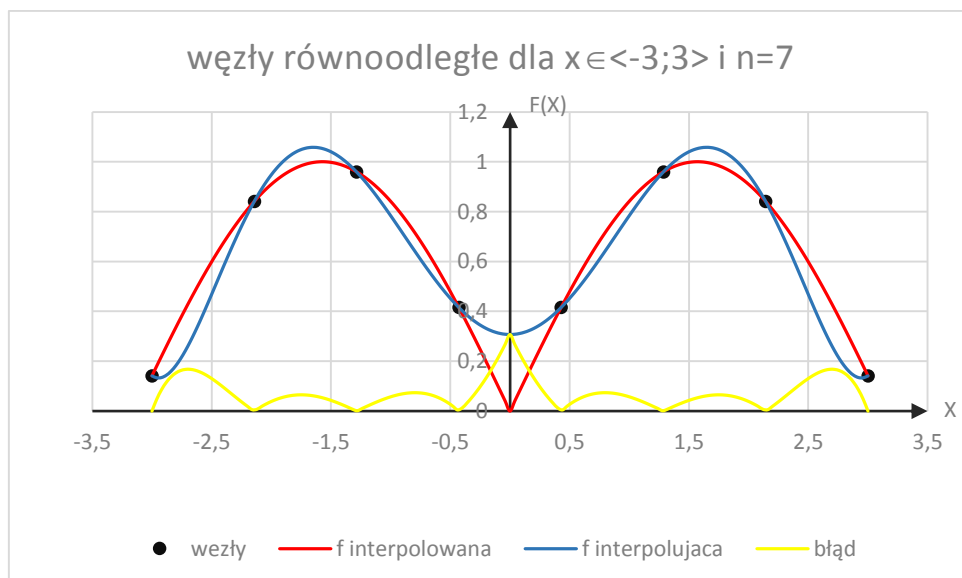
Wartości funkcji interpolującej otrzymaliśmy korzystając ze wzoru Lagrange'a:

$$L_n(x) = \sum_{i=0}^n f_i \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$$

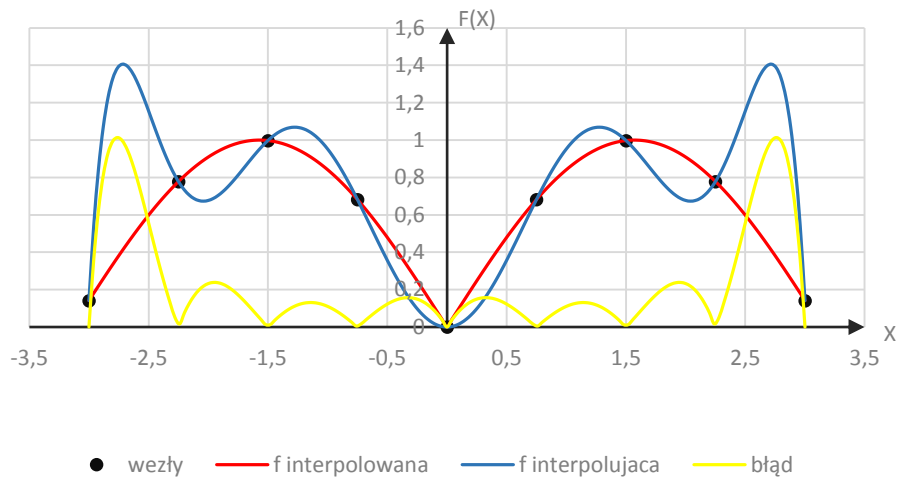
Wartości błędu interpolacji wyznaczaliśmy korzystając ze wzoru:

$$R_n(x) = |f(x) - L_n(x)|$$

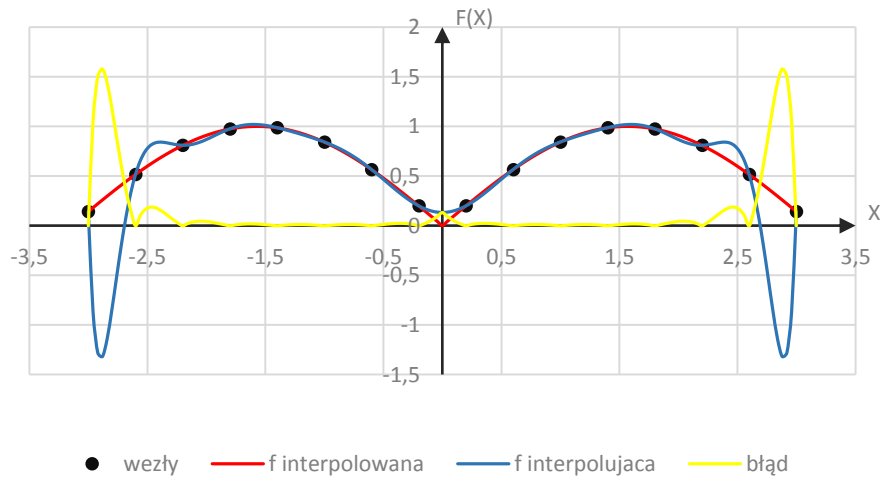
## 2. Otrzymane wykresy



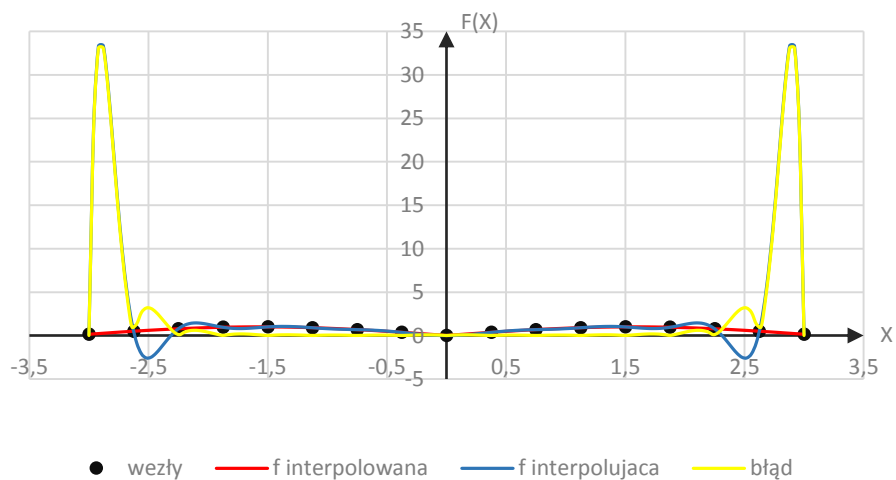
węzły równoodległe dla  $x \in (-3; 3)$  i  $n=8$



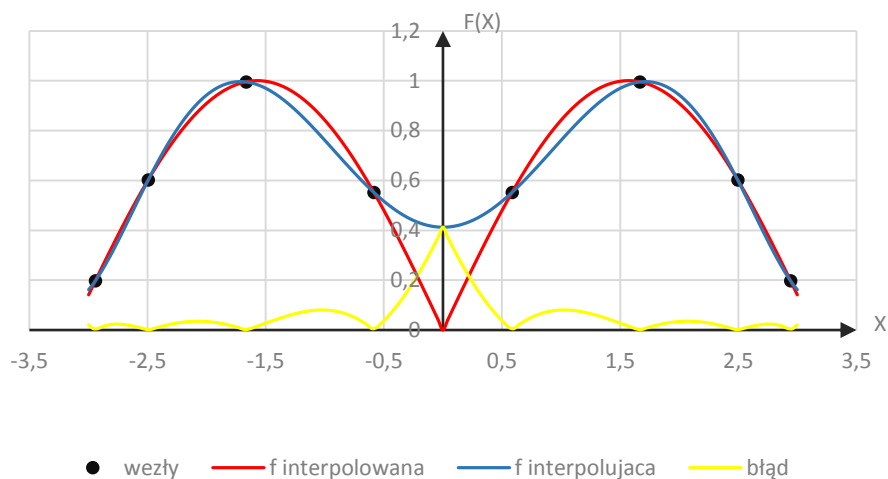
węzły równoodległe dla  $x \in (-3; 3)$  i  $n=15$



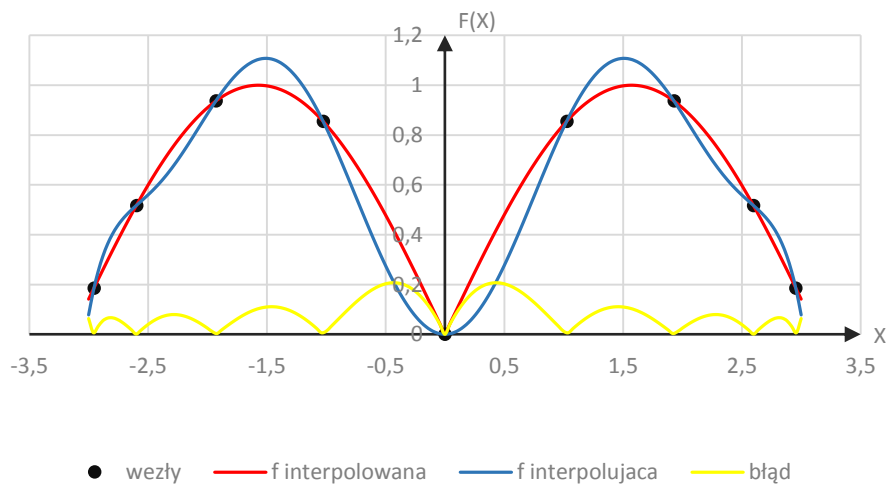
węzły równoodległe dla  $x \in (-3; 3)$  i  $n=16$



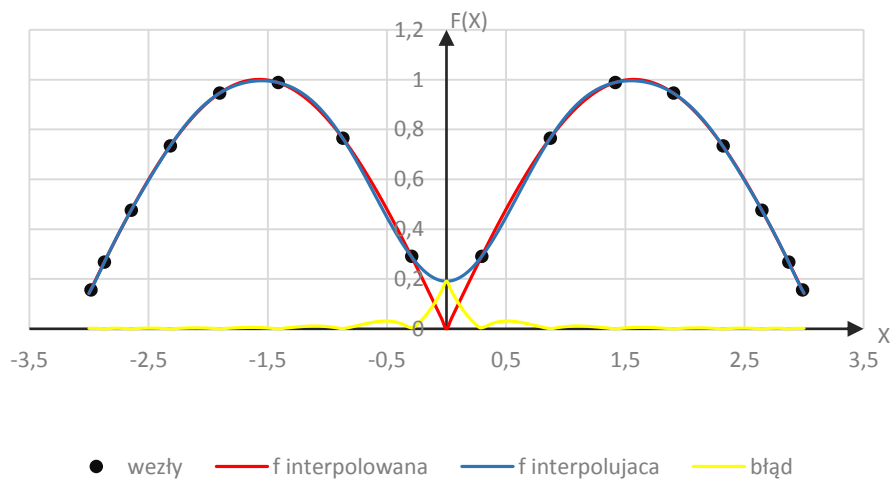
węzły dobrane optymalnie dla  $x \in (-3; 3)$  i  $n=7$



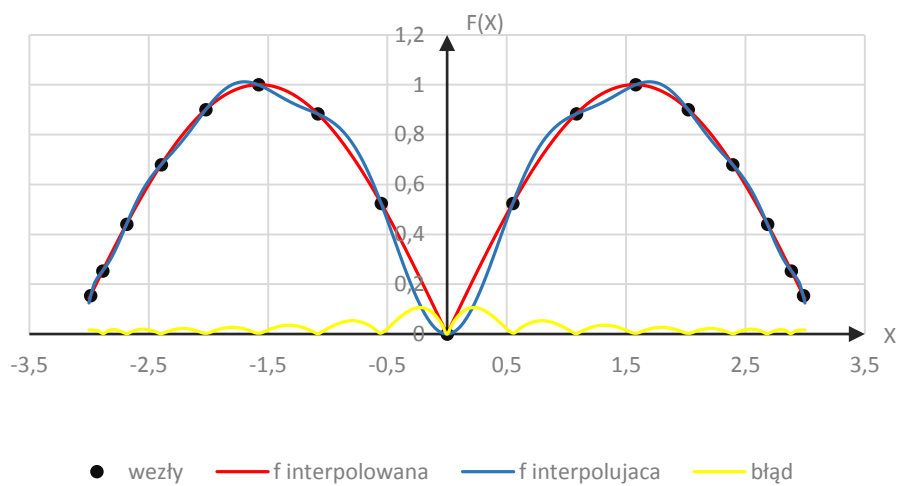
węzły dobrane optymalnie dla  $x \in (-3; 3)$  i  $n=8$



węzły dobrane optymalnie dla  $x \in (-3; 3)$  i  $n=15$

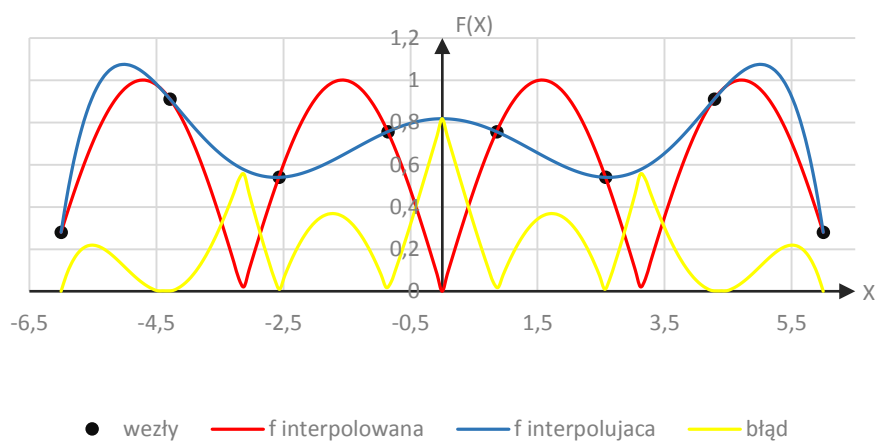


węzły dobrane optymalnie dla  $x \in (-3; 3)$  i  $n=16$

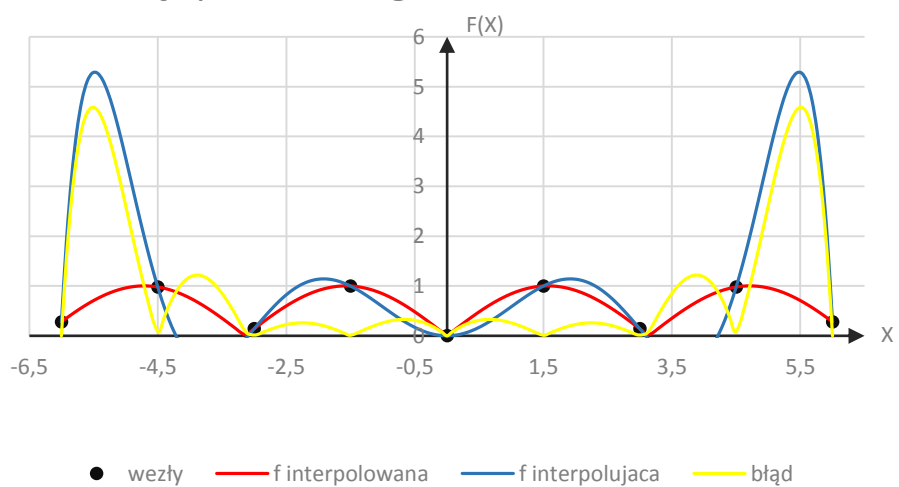


węzły równoodległe dla  $x \in (-6; 6)$

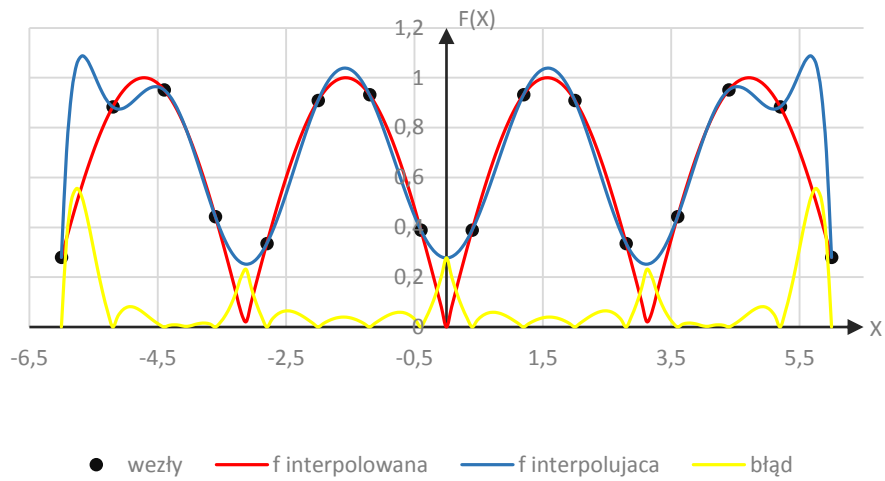
i  $n=7$



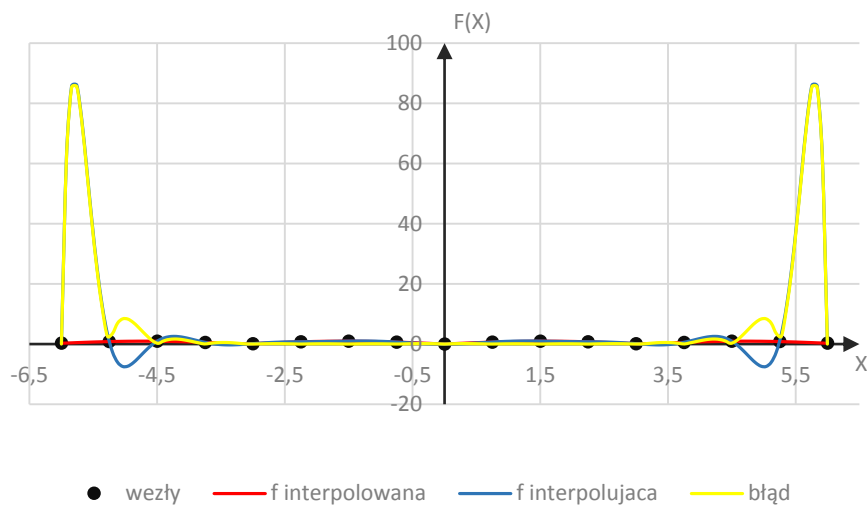
węzły równoodległe dla  $x \in (-6; 6)$  i  $n=8$



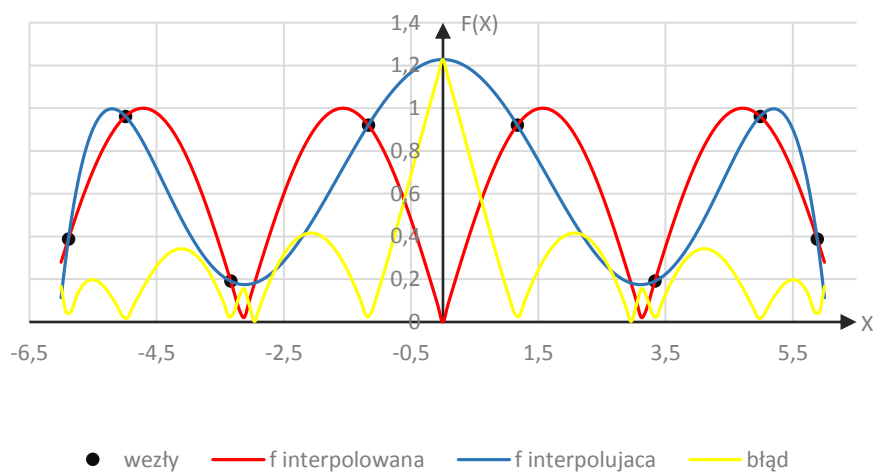
węzły równoodległe dla  $x \in (-6; 6)$  i  $n=15$



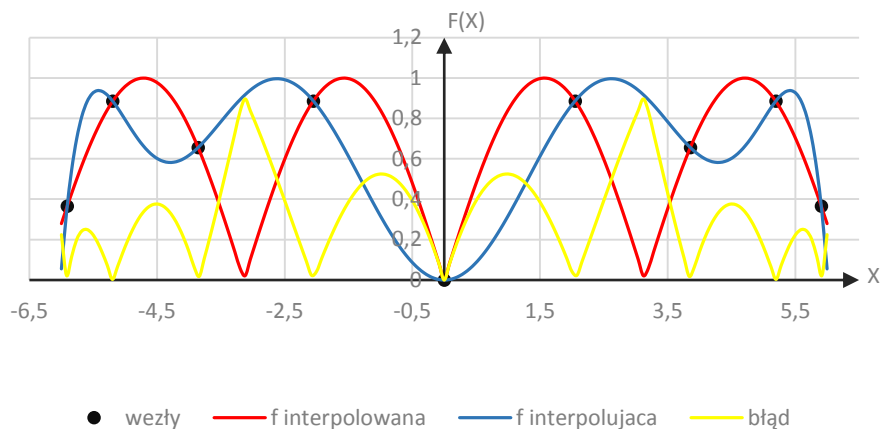
węzły równoodległe dla  $x \in (-6; 6)$  i  $n=16$



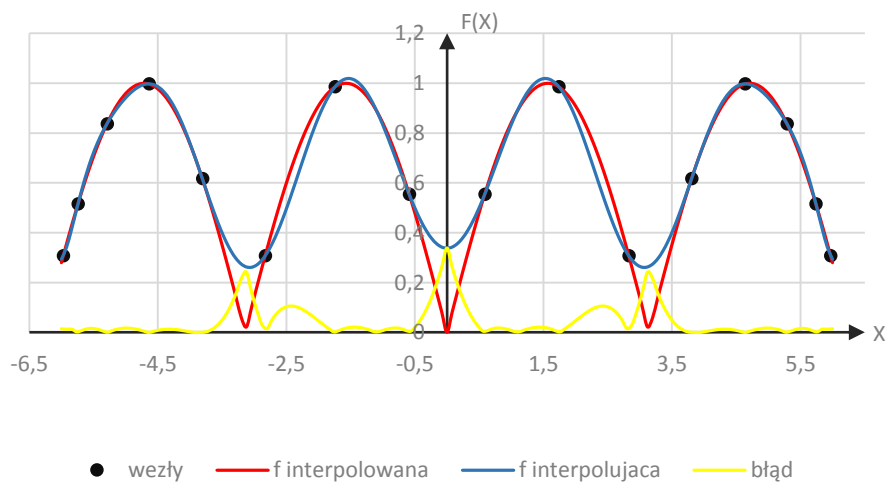
węzły dobrane optymalnie dla  $x \in (-6; 6)$   $n=7$



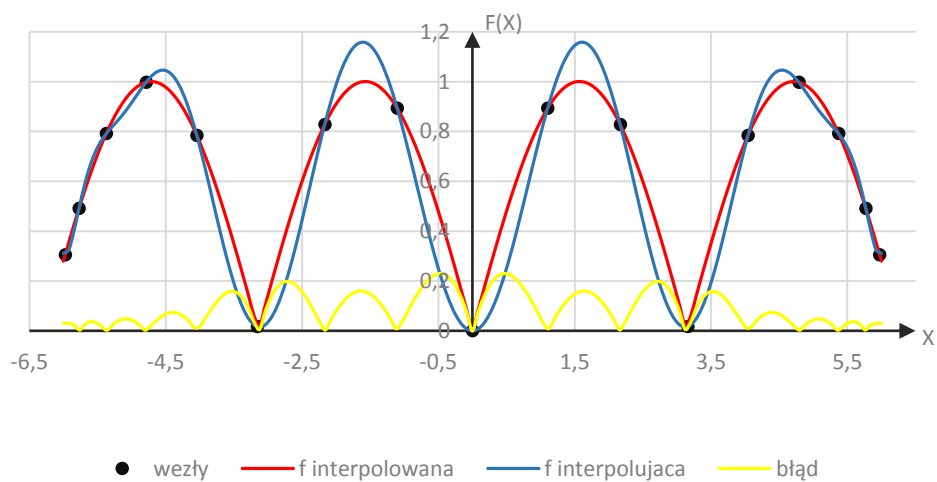
węzły dobrane optymalnie dla  $x \in (-6; 6)$  i  $n=8$



węzły dobrane optymalnie dla  $x \in (-6; 6)$  i  $n=15$



węzły dobrane optymalnie dla  $x \in (-6; 6)$  i  $n=16$



### 3. Wnioski

Najlepsze wyniki uzyskaliśmy dla węzłów dobranych optymalnie gdzie  $n=16$  i  $x \in \langle -3; 3 \rangle$ .

Lepsze wyniki można zaobserwować dla mniejszego przedziału interpolacji. Także zwiększanie liczby węzłów powodowało uzyskanie dokładniejszej funkcji interpolującej. W przypadku węzłów rozłożonych równomiernie można zauważyć znaczny wzrost błędu interpolacji na końcach przedziałów wraz ze wzrostem ilości węzłów. Dla węzłów dobranych optymalnie wartości błędu maleją wraz ze wzrostem ilości węzłów.

### 4. Kod programu

```
#define _USE_MATH_DEFINES
#include <iostream>
#include<vector>
#include<cmath>
#include <fstream>
#include <string>
using namespace std;

/* Funkcja tworząca tablice n+1 równoodległych x z przedziału <a;b> */
vector<double> stworzTabliceX(double a, double b, double n) {

    vector<double> X;
    double h = (b - a) / n;
    for (int i = 0; i <= n; i++) {
        double x = a + i*h;
        X.push_back(x);
    }
    return X;
}

/* Funkcja obliczająca wartości funkcji |sin(x)| dla x z przekazanego wektora */
vector<vector<double> > obliczY(vector<double> X) {

    vector<double> Y;
    vector<vector<double> > XY;
    XY.push_back(X);
    for (int i = 0; i < X.size(); i++) {
        Y.push_back(abs(sin(X[i])));
    }
    XY.push_back(Y);
    return XY;
}

/*Funkcja tworząca wektor n+1 optymalnie dobranych x */
vector<double> stworzTabliceXOptymalnych(double a, double b, double n) {
    vector<double> X;
```



```

        for (int i = 0; i <= n; i++) {
            double x = 0.5*(a + b) - 0.5*(b - a)*cos((2 * i + 1) / (2 * n + 2)*
M_PI);
            X.push_back(x);
        }
        return X;
    }

    /* Obliczenie wartosci Lagrange'a w punkcie x*/
    double obliczWartosciLagrange(vector<vector<double> > XY, double x) {
        double Lnx = 0;
        for (int i = 0; i < XY[0].size(); i++) {
            double iloczyn = 1;
            for (int j = 0; j < XY[0].size(); j++) {
                if (j != i) {
                    iloczyn *= (x - (XY[0][j])) / ((XY[0][i]) - (XY[0][j]));
                }
            }
            Lnx += (XY[1][i])*iloczyn;
        }
        return Lnx;
    }

    /*zapisanie danych do pliku csv*/
    void zapisz(vector<vector <double> > XY, vector<vector <double> > XYL, int n, string
nazwa) {

        ofstream plik;
        if (!n) {
            nazwa += "r_.csv";
            plik.open(nazwa);
        }
        else {
            nazwa += "o_.csv";
            plik.open(nazwa);
        }
        plik << " wezly:" << endl;
        for (int i = 0; i < XY.size(); i++)
        {
            for (int j = 0; j < XY[0].size(); j++)
            {
                plik << XY[i][j] << ";";
            }
            plik << endl;
        }

        plik << "Lagrange" << endl;
        for (int i = 0; i < XYL.size(); i++)
        {
            for (int j = 0; j < XYL[0].size(); j++)
            {
                plik << XYL[i][j] << ";";
            }
            plik << endl;
        }

        plik << "wartosci bledu:" << endl;
        for (int j = 0; j < XYL[0].size(); j++)
        {
            plik << abs(XYL[1][j] - XYL[2][j]) << ";";
        }
        plik << endl;
    }

```

```

        plik.close();
    }

int main(int argc, char** argv) {

    cout << "Podaj przedzial: " << endl;
    double a, b;
    cin >> a >> b;

    cout << "podaj n:" << endl;
    double n;
    cin >> n;

    cout << "podaj nazwe: " << endl;
    string nazwa;
    cin>>nazwa;

    //rownoodlegle
    vector<double> x = stworzTabliceX(a, b, n);
    vector<vector<double> > XY = obliczY(x); //wektor wezlow

    //150
    vector<double> X_150 = stworzTabliceX(a, b, 150);
    vector<vector<double>> XY_150 = obliczY(X_150);vector<double> LN;

    for (int i = 0; i <= 150; i++) {
        double l = obliczWartosciLagrange(XY, XY_150[0][i]);
        LN.push_back(l);
    }
    XY_150.push_back(LN);

    zapisz(XY, XY_150, 0, nazwa);

    //optymalne
    vector<double> Xoptymalne = stworzTabliceXOptymalnych(a, b, n);
    vector<vector<double> > XY_optymalne = obliczY(Xoptymalne); //wektor wezlow
    optymalnych
    //150
    vector<double> X_150o = stworzTabliceX(a, b, 150);
    vector<vector<double> > XY_150o = obliczY(X_150o);

    vector<double> LNo;
    for (int i = 0; i <= 150; i++) {
        double l = obliczWartosciLagrange(XY_optymalne, XY_150o[0][i]);
        LNo.push_back(l);
    }
    XY_150o.push_back(LNo);
    zapisz(XY_optymalne, XY_150o, 1, nazwa);

    return 0;
}

```