Laboratorium Metod Numerycznych						
Rok akademicki	Termin	Rodzaj studiów	Kierunek	Prowadzący	Grupa	Sekcja
2017/2018	Czwartek 16:30 - 19:30	Dzienne	INF	mgr inż. Wojciech Łabaj	6	11

Sprawozdanie z ćwiczenia numer 1

Data wykonania ćwiczenia: 2018-04-26
Data oddania sprawozdania:2018-05-21

Temat ćwiczenia:
Interpolacja

Skład podsekcji:
Michał Miciak
Bartłomiej Krasoń

Treść zadania:

Napisać program wyznaczający wartość wielomianu interpolacyjnego Lagrange'a w punktach leżących w przedziale <a,b> dla funkcji interpolowanej f(x).

Funkcja interpolowana: `f(x)= |cos(x)*x|`

Przyjąć(n+1) węzłów – dla n=7, 8, 15, 16

a) Równoodległych($x_i = a + i * h, h = \frac{b-a}{n}, i = 0,1,2, \dots . n$)
b) Dobranych optymalnie($x_i = \frac{1}{2}\ (a + b) + \frac{1}{2}(b - a)\cos\left(\frac{2i+1}{2n+2}\pi\right) i = 0,1, \dots n.$

Przedziały

a) $x \in < -3; 3 >$
b) $x \in < -6; 6 >$

Punkty w których należy wyznaczyć wartości wielomianu Lagrange'a są również równoodległe – dla np = 150

($xp_j = a + j * hp, hp = \frac{b-a}{np}, j = 0,1,2, \dots . np$)

Program oprócz funkcji głównej (main) ma zawierać dodatkowo następujące funkcje pomocnicze:

1) Funkcja wprowadzająca dane(n, np, a, b),
2) Funkcja tablicująca węzły i wartości funkcji interpolowanej f(x) w tych węzłach,
3) Funkcja wyznaczająca wartości wielomianu Lagrange'a w dowolnym punkcie
4) Funkcja wyznaczająca tablice: punktów xp_j, wartości funkcji interpolowanej $f(xp_j)$ i wielomianu Lagrange'a w tych punktach $L_n(xp_j)$ i zapisująca te tablice do pliku.

W oparciu o te tablice zrealizować wykres(np. za pomocą Excela) zawierający:
- wykres funkcji interpolowanej,
- wykres funkcji interpolującej,
- błąd
- węzły

Rozwiązanie

Wzór Lagrange'a:

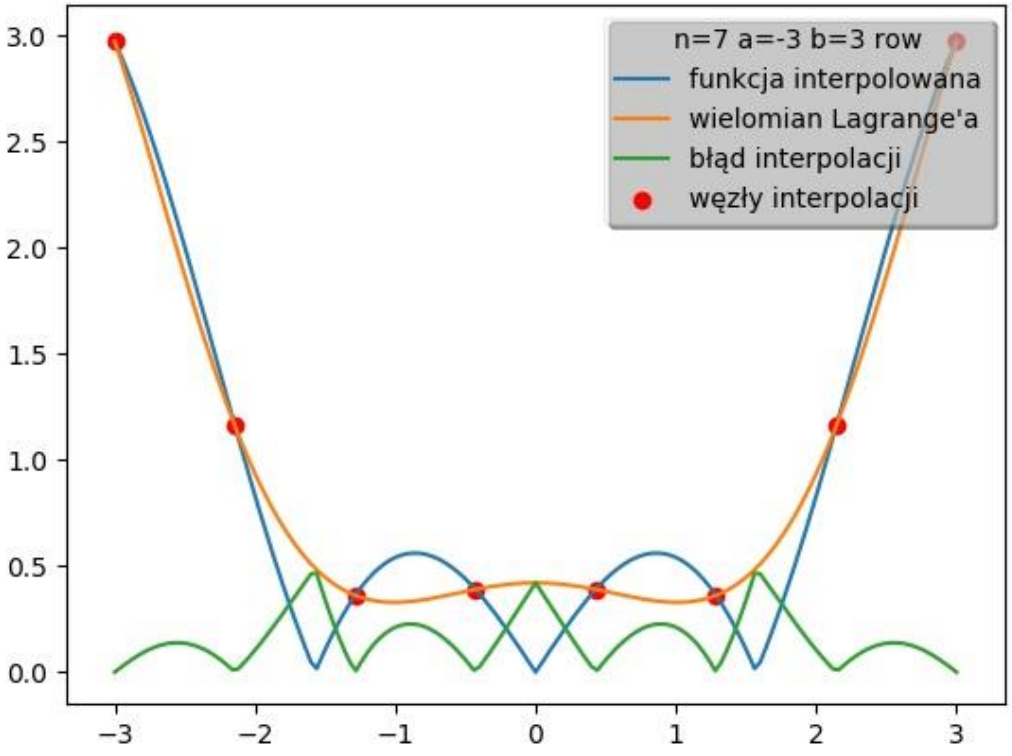
$$L_n(x) = \sum_{i=0}^{n} f_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

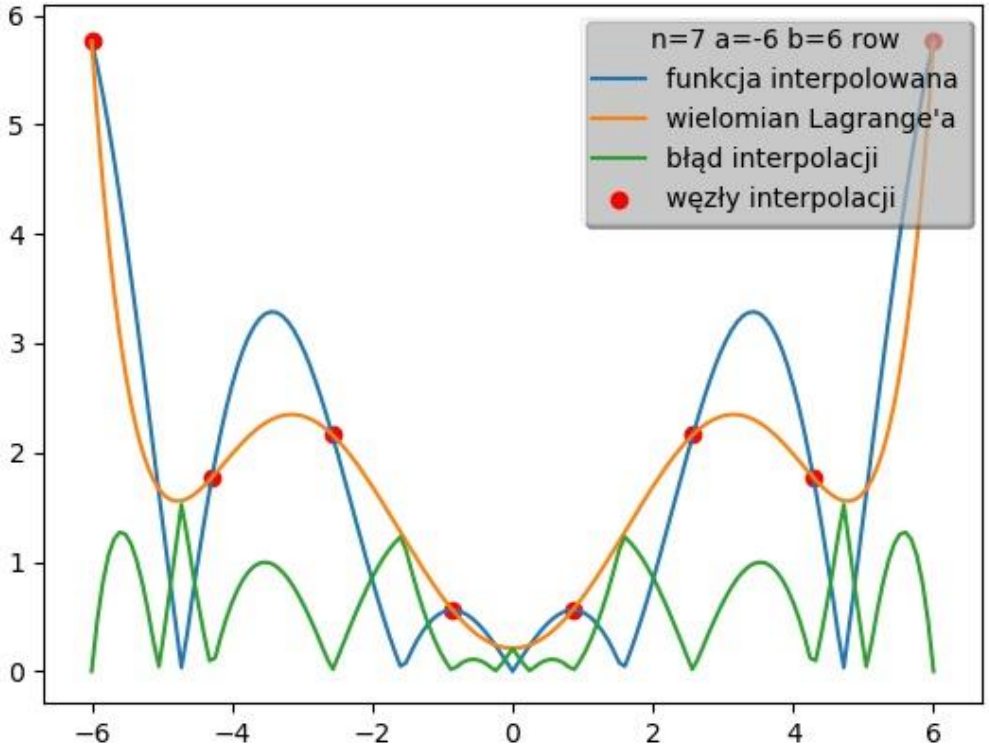
Wzór na błąd:

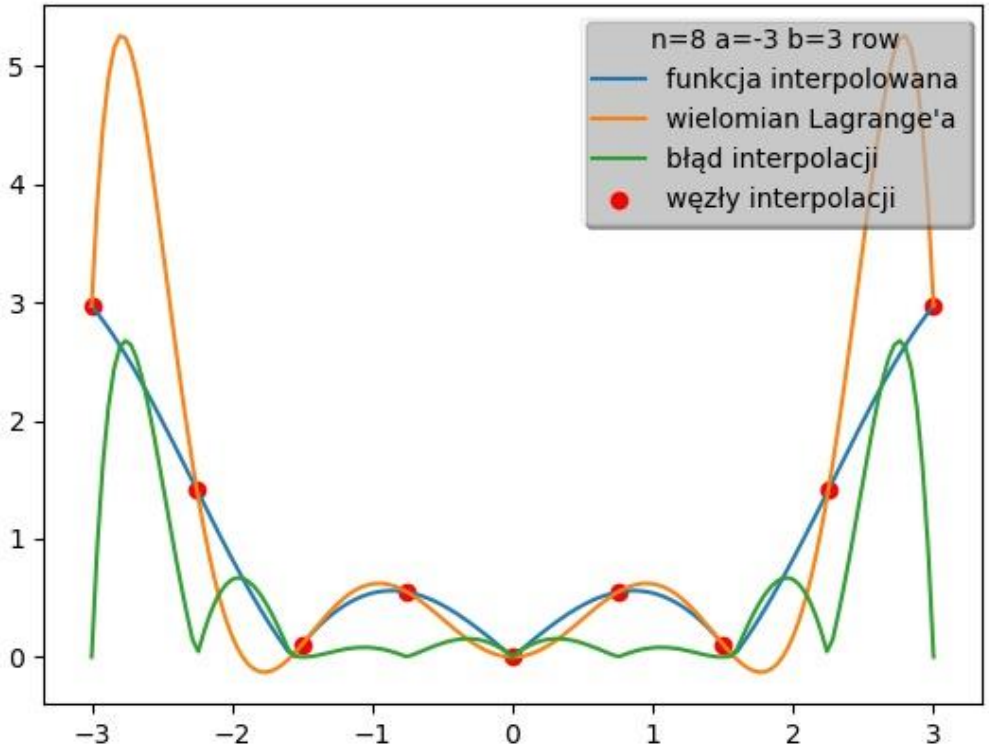
$$R_n(x) = |f(x) - L_n(x)|$$

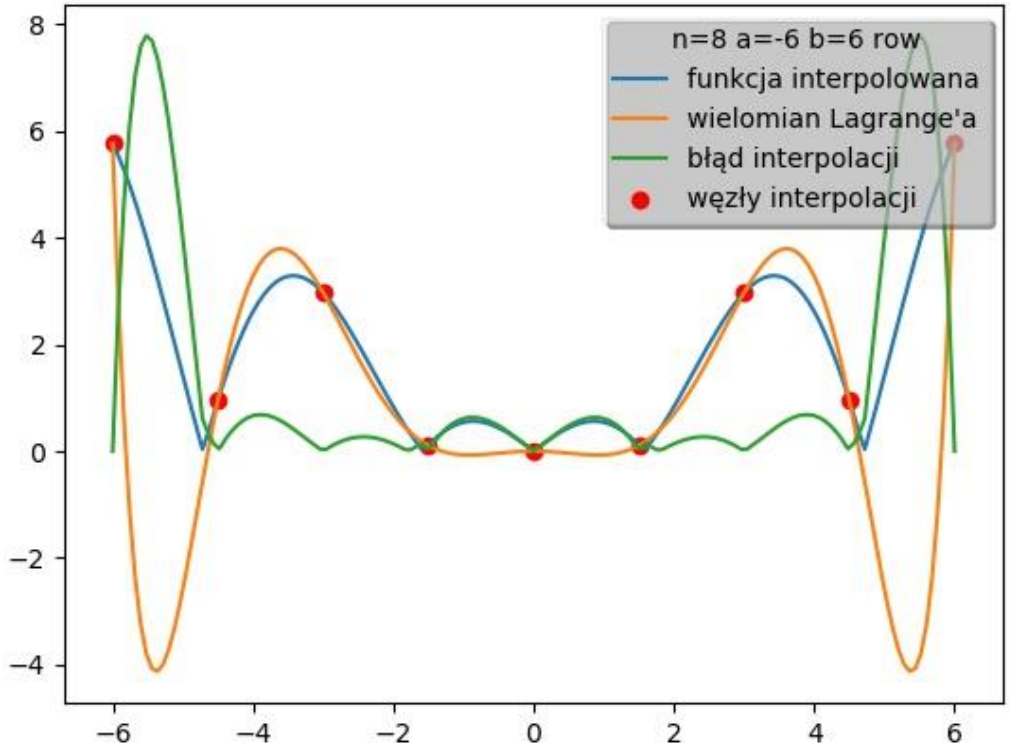
Wykresy

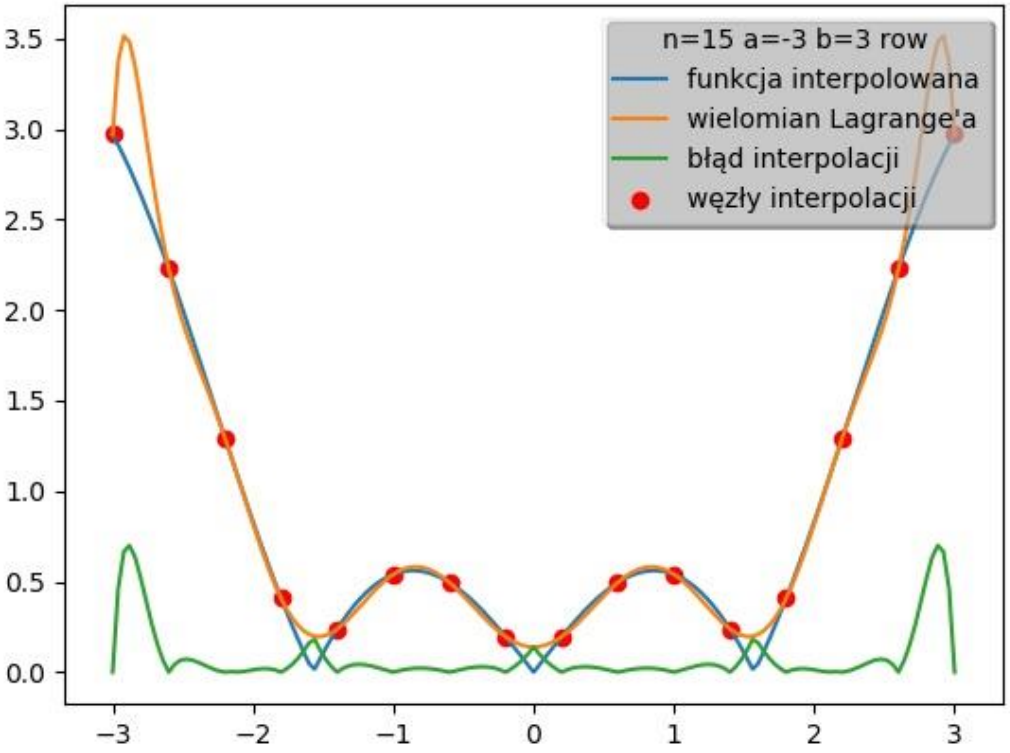
Dla punktów równoodległych:

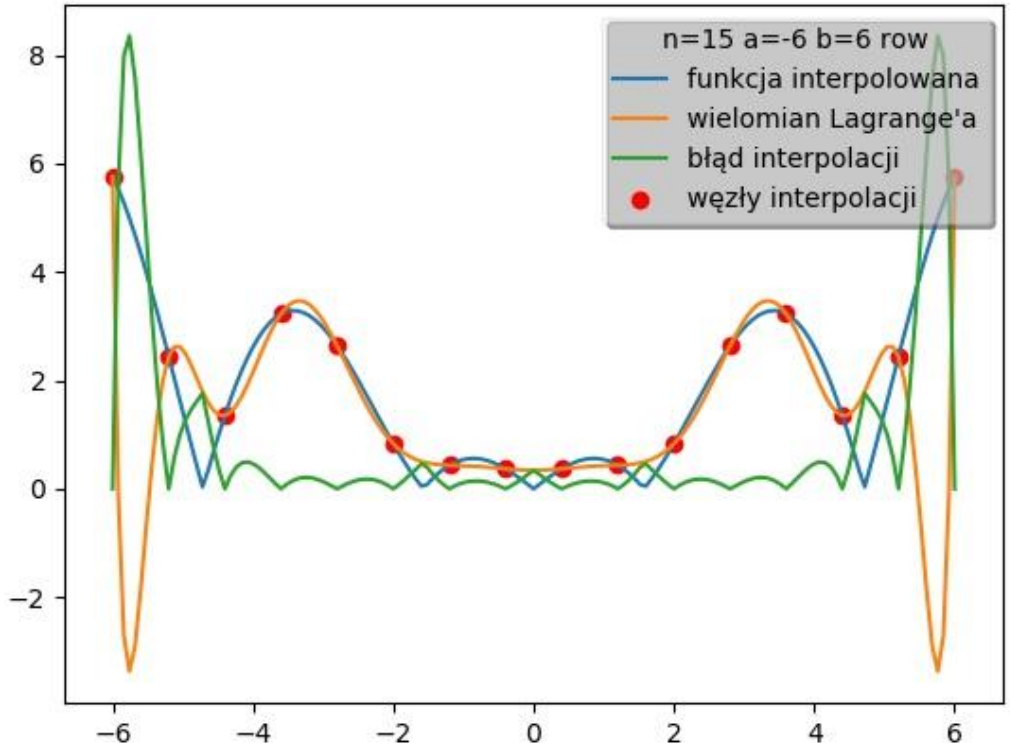


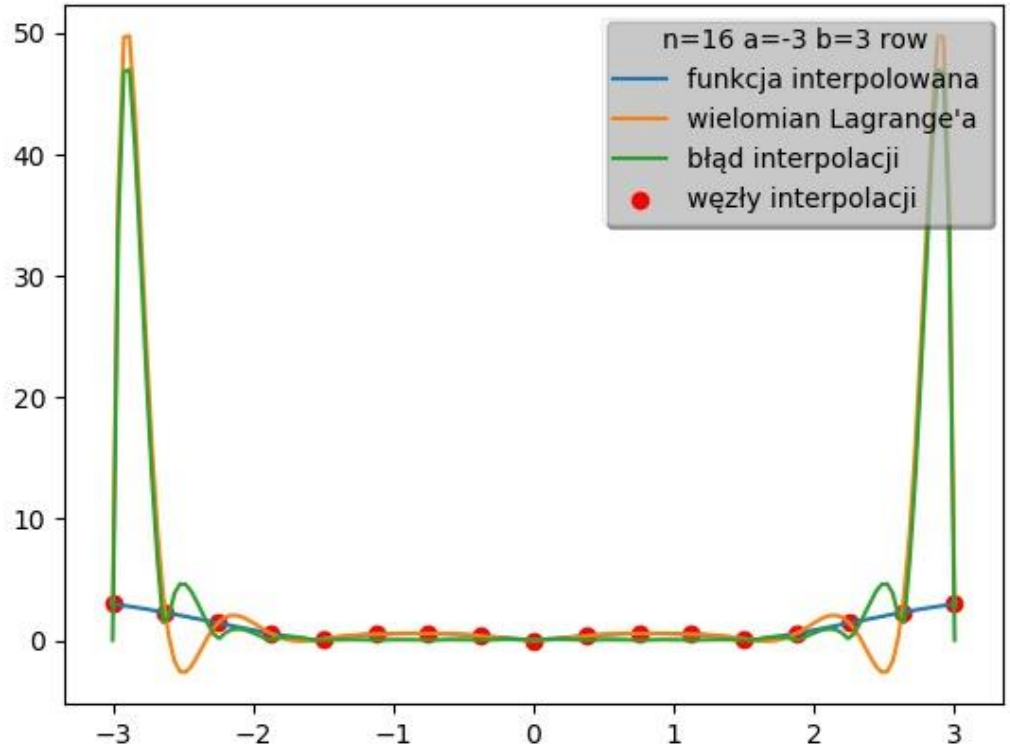

n=7 a=-6 b=6 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
6
5
4
3
2
1
0
−6
−4
−2
0
2
4
6

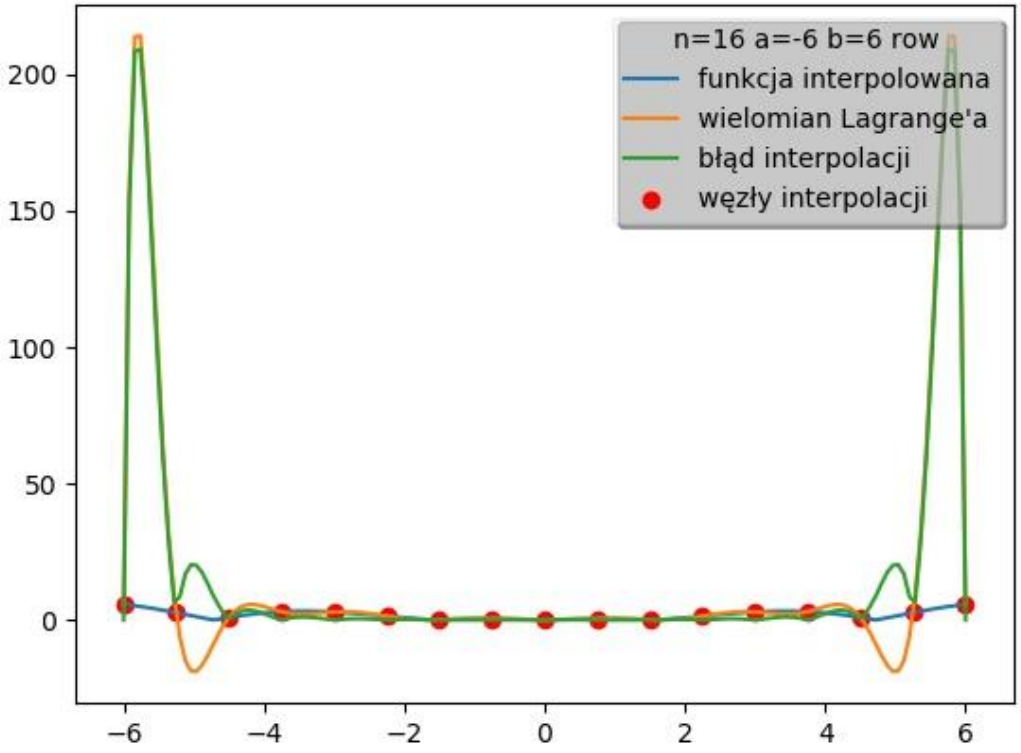

n=8 a=-3 b=3 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
5
4
3
2
1
0
−3
−2
−1
0
1
2
3


n=8 a=-6 b=6 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
8
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6

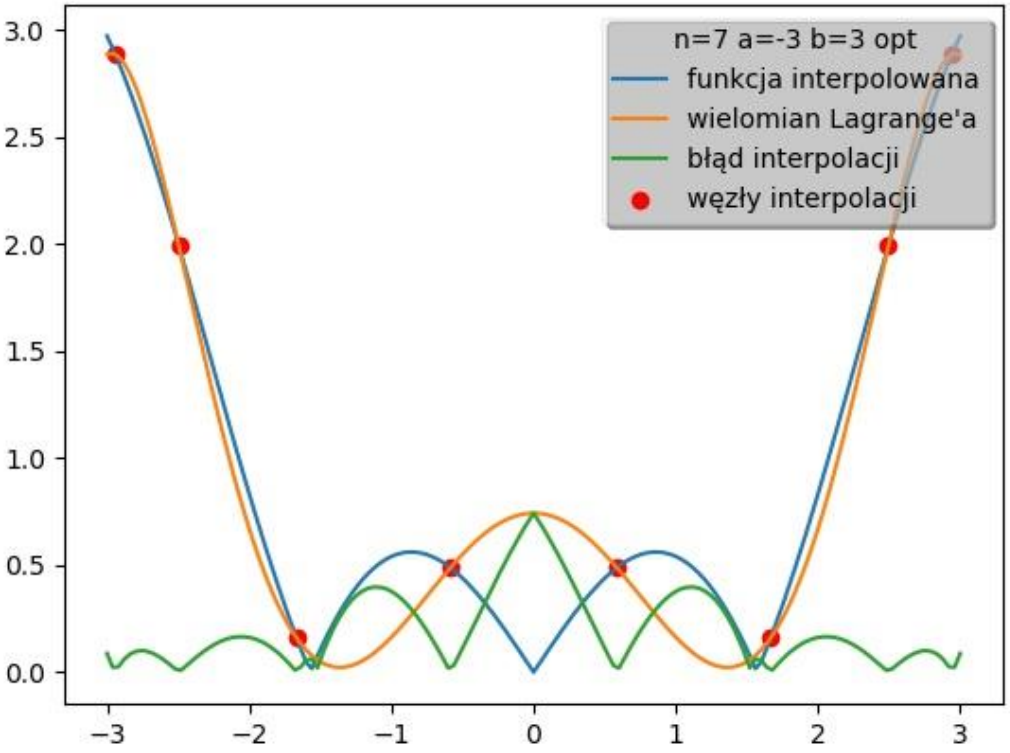

n=15 a=-3 b=3 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−3
−2
−1
0
1
2
3

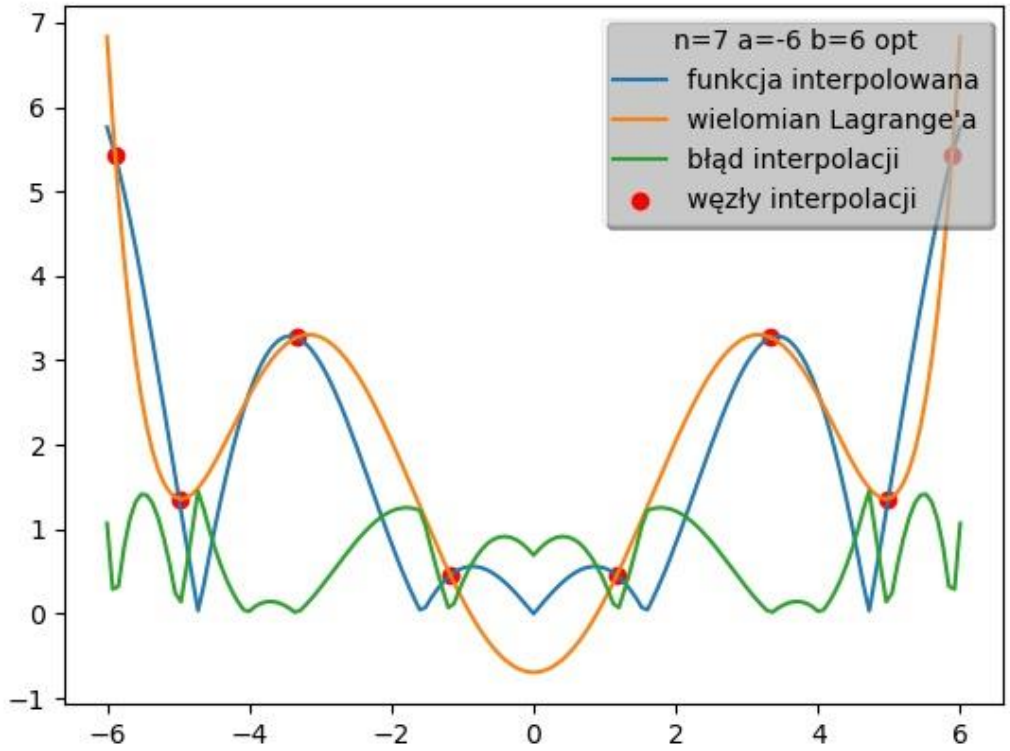

n=15 a=-6 b=6 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
8
6
4
2
0
−2
−6
−4
−2
0
2
4
6

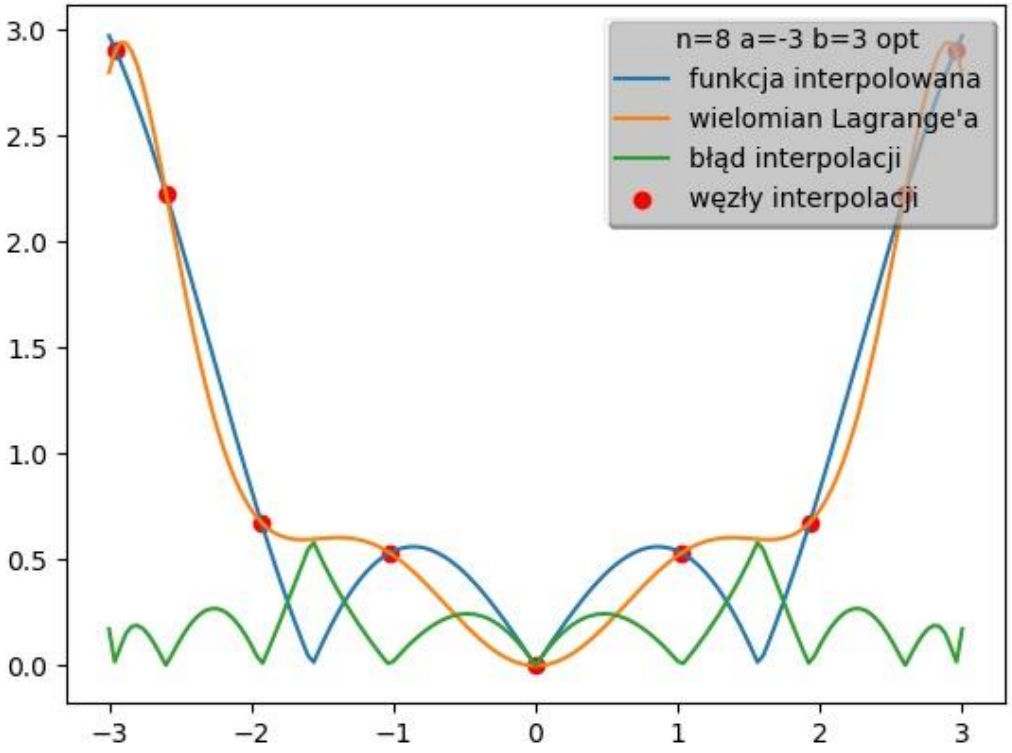

n=16 a=-3 b=3 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
50
40
30
20
10
0
−3
−2
−1
0
1
2
3

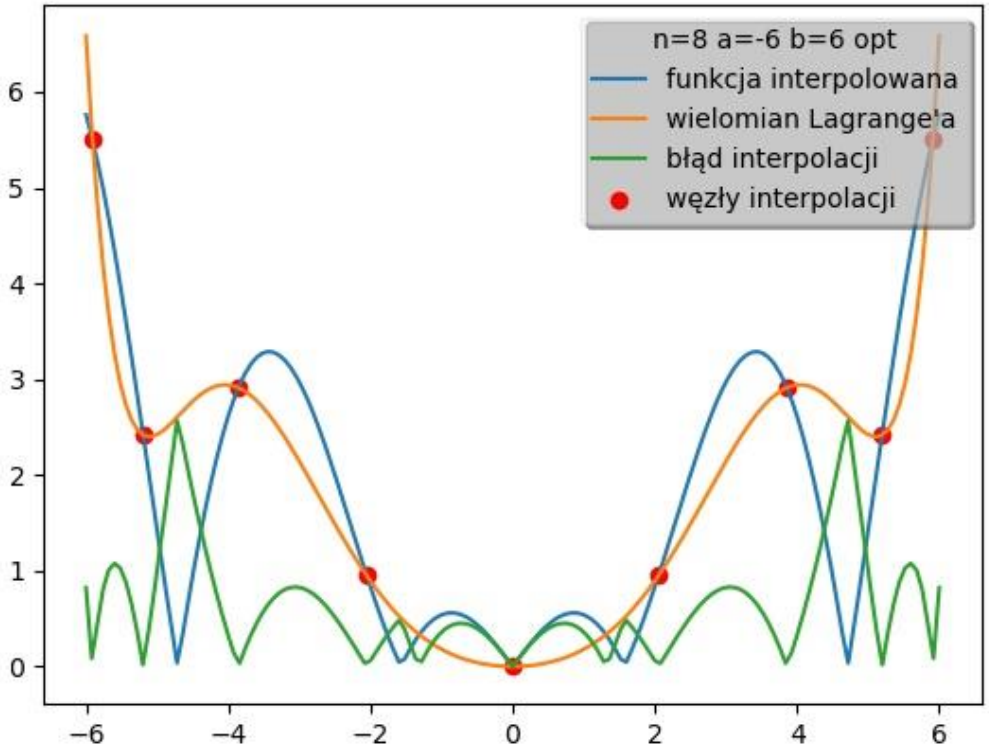

n=16 a=-6 b=6 row
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
200
150
100
50
0
−6
−4
−2
0
2
4
6

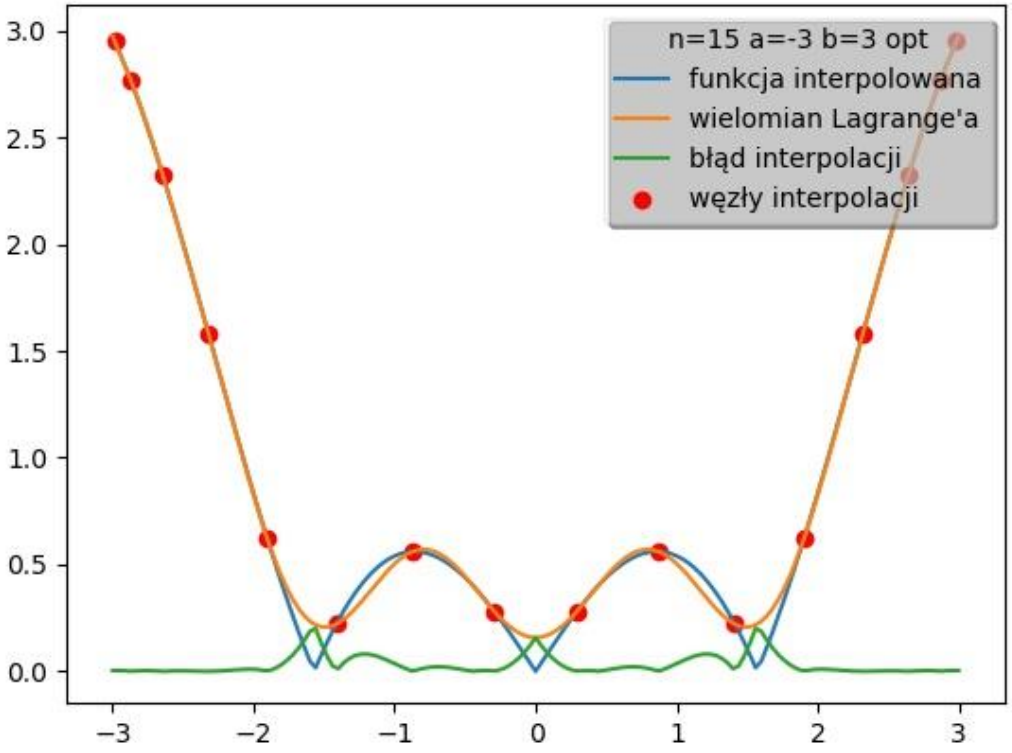
Dla punktów dobranych optymalnie:

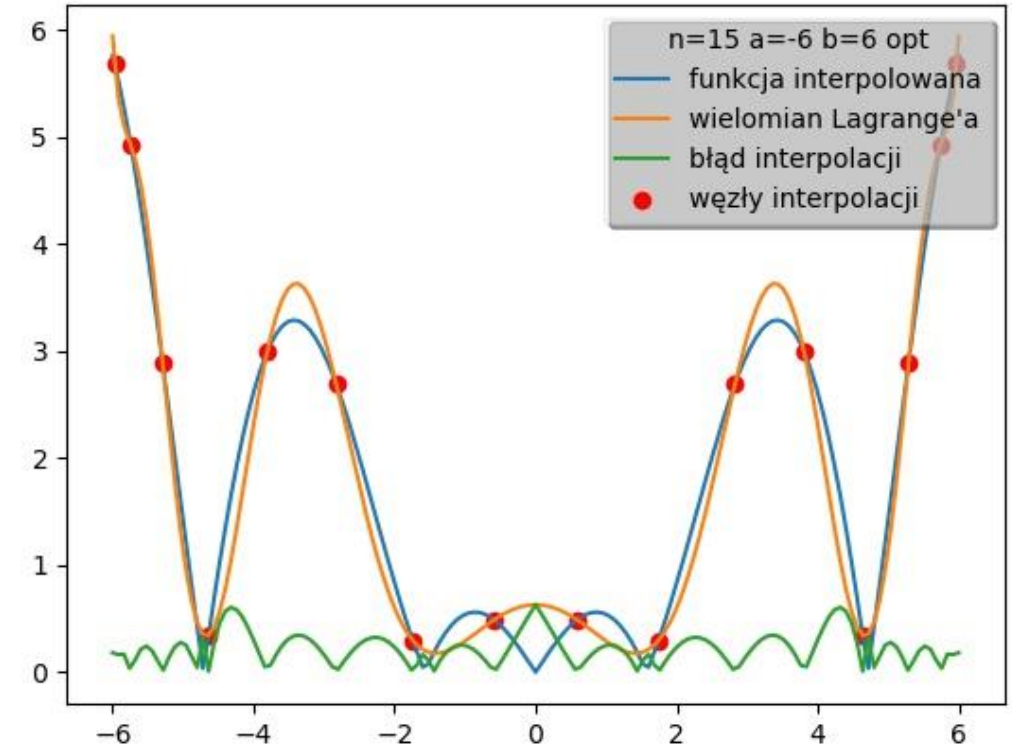


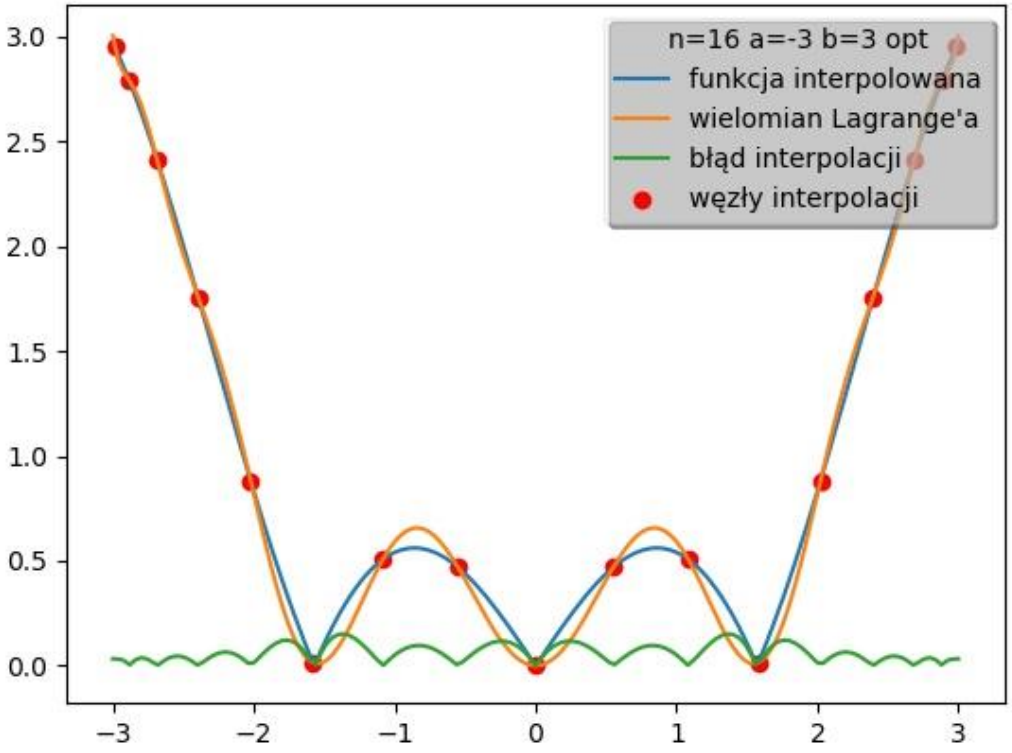


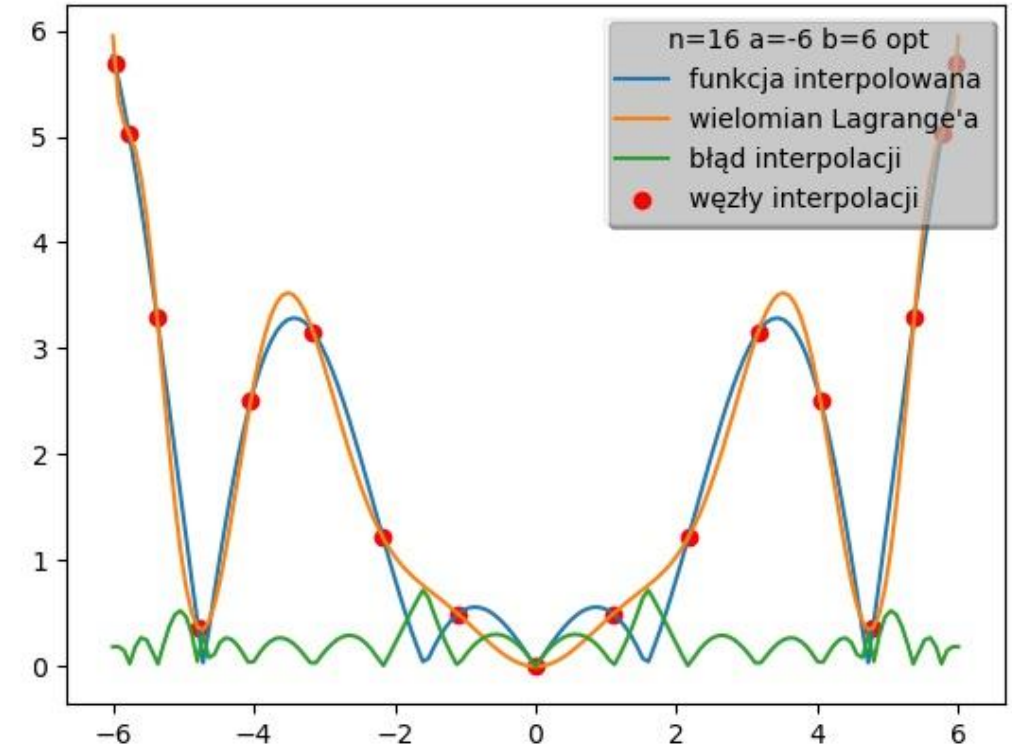

n=8 a=-3 b=3 opt
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−3
−2
−1
0
1
2
3


n=8 a=-6 b=6 opt
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
6
5
4
3
2
1
0
−6
−4
−2
0
2
4
6


n=15 a=-3 b=3 opt
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−3
−2
−1
0
1
2
3


n=15 a=-6 b=6 opt
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
6
5
4
3
2
1
0
−6
−4
−2
0
2
4
6


n=16 a=-3 b=3 opt
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−3
−2
−1
0
1
2
3


n=16 a=-6 b=6 opt
funkcja interpolowana
wielomian Lagrange'a
błąd interpolacji
węzły interpolacji
6
5
4
3
2
1
0
−6
−4
−2
0
2
4
6

Wnioski

1. Błąd interpolacji jest znacznie mniejszy w przypadku, gdy węzły interpolacji są dobrane optymalnie.
2. Dla większej ilości węzłów błąd interpolacji jest mniejszy, aczkolwiek dla metody dobierania węzłów interpolacji równoodlegle, można zaobserwować stosunkowo duży błąd na krańcach przedziału.
3. W przedziale <-3,3> wartości wielomianu Lagrange'a różnią się mniej od wartości funkcji interpolowanej, niż w przedziale <-6,6>.

Kod

```
#include <iostream>
#include <math.h>
#include <vector>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;
#define M_PI 3.141592653589

// Funkcja pobierajaca dane od uzytkownika
void wejscie(int &n, int &np, int &a, int &b, bool &optymalnie)
{
	cout << "Podaj n: ";

	cin >> n;

	cout << "Podaj np: ";

	cin >> np;

	cout << "Podaj a: ";

	cin >> a;

	cout << "Podaj b: ";

	cin >> b;

	cout << "Jak wyznaczyc wezly:\n 1 - optymalnie\n 0 - rownoodlegle\n";

	cin >> optymalnie;
}
// Funkcja zwraca wartosc funkcji interpolowanej dla danego argumentu x
double funkcjaInterpolowana(double x)
{
	double result = cos(x)*x;

	if (result < 0)
	{
		result = result * (-1.0);
	}
	return result;
}
// Funkcja zwraca wektor zawierajacy wezly
vector<double> wyznaczWezly(int n, int a, int b, bool optymalne)
{
	vector<double> vectorDoZwrotu;
```

```
	if (optymalne)
	{
		const double pi = M_PI;
		double p1 = (double)(a + b) / 2;
		double p2 = (double)(b - a) / 2;
		for (int i = 0; i < (n + 1); i++)
		{
			double pom1 = cos(2 * i + 1);
			double pom2 = (2 * n + 2);
			double x = (p1 + p2 * cos(((2 * i + 1) / (double)(2 * n +
2))*pi));
			vectorDoZwrotu.push_back(x);
		}
	}
	else
	{
		double p = (double)(b - a) / n;
		for (int i = 0; i < (n + 1); i++)
		{
			double x = a + (i*p);
			vectorDoZwrotu.push_back(x);
		}
	}
	return vectorDoZwrotu;
}
// Funckja zwraca wektor wartosci funkcji interpolowanej dla wezlow
vector<double> wyznaczWartosciFunkcjiInterpolowanejDlaWezlow(vector<double> &wezly)
{
	vector<double> vectorDoZwrotu;
	for (int i = 0; i < wezly.size(); i++)
	{
		vectorDoZwrotu.push_back(funkcjaInterpolowana(wezly[i]));
	}
	return vectorDoZwrotu;
}
// Funkcja zwraca wektor wartosci bledow interpolacji
vector<double> wyznaczBladInterpolacji(vector<double> funkcjaInterpolowana,
vector<double> wartosciWielomianuLagrange)
{
	vector<double> vectorDoZwrotu;
	for (int i = 0; i < funkcjaInterpolowana.size(); i++)
	{
		vectorDoZwrotu.push_back(abs(funkcjaInterpolowana[i] -
wartosciWielomianuLagrange[i]));
	}
	return vectorDoZwrotu;
}
// Funkcja zwraca wektor wartosci Wielomianu Lagrange'a dla danych x
vector<double> wyznaczWartosciWielomanuLagrange(vector<double> punkty, int n,
vector<double> wartosciFunkcjiInterpolowanej, vector<double> wezly)
{
	vector<double> vectorDoZwrotu;
	for (int k = 0; k < punkty.size(); k++)
	{
		double wynik = 0;
		for (int i = 0; i < (n + 1); i++)
		{
			double wynikIloczynu = 1;
			for (int j = 0; j < (n + 1); j++)
			{
				if (i != j)
				{
```

```
                                wynikIloczynu *= (punkty[k] - wezly[j]) / (wezly[i]
- wezly[j]);
                        }
                    }
                    wynik += wartosciFunkcjiInterpolowanej[i] * wynikIloczynu;
                }
                vectorDoZwrotu.push_back(wynik);
        }
        return vectorDoZwrotu;
}
// Funkcja wykonujaca obliczenia oraz generujaca raport
void wyjscie(int n, int np, int a, int b, bool optymalnie)
{
        string nazwa = "raport_n=";
        nazwa += to_string(n);
        nazwa += "_ab=";
        nazwa += to_string(a);
        nazwa += to_string(b);
        if (optymalnie)
        {
                nazwa += "_opt";
        }
        else
        {
                nazwa += "_row";
        }

        fstream plik;

        plik.open(nazwa + ".csv", ios_base::out);

        vector<double> punktyX = wyznaczWezly(np, a, b, false);
        vector<double> wartosciFunkcjiInterpolowanej =
wyznaczWartosciFunkcjiInterpolowanejDlaWezlow(punktyX);
        vector<double> wezly = wyznaczWezly(n, a, b, optymalnie);
        vector<double> wartosciFunkcjiInterpolowanejDlaWezlow =
wyznaczWartosciFunkcjiInterpolowanejDlaWezlow(wezly);
        vector<double> wartosciWielomianuLagrange =
wyznaczWartosciWielomanuLagrange(punktyX, n, wartosciFunkcjiInterpolowanejDlaWezlow,
wezly);
        vector<double> blad = wyznaczBladInterpolacji(wartosciFunkcjiInterpolowanej,
wartosciWielomianuLagrange);

        plik << "x,y,L,b,w,fw\n";
        for (int i = 0; i < np + 1; i++)
        {
                plik << punktyX[i] << "," << wartosciFunkcjiInterpolowanej[i] << "," <<
wartosciWielomianuLagrange[i] << "," << blad[i];
                if (i < wezly.size()) plik << "," << wezly[i] << "," <<
wartosciFunkcjiInterpolowanejDlaWezlow[i];
                plik << endl;
        }
        plik.close();
}
int main()
{
        int n;
        int np;
        int a;
        int b;
        bool optymalnie;
```

```
        wejscie(n, np, a, b, optymalnie);
        wyjscie(n, np, a, b, optymalnie);
}
```