Wydział WIMiIP	Imię i nazwisko: Zuzanna Będkowska	Rok: 2	Grupa: 1	Data: 01.05.2022
Metody Numeryczne	Temat: Aproksymacja			

Zadanie: wyznaczenie wielomianu aproksymacyjnego dowolnego stopnia

Zgodnie z instrukcją, w trakcie zajęć stworzono następujący kod:

```cpp
#include <iostream>
#include <fstream>
#include <cmath>
#include <iomanip>
#include <vector>

using namespace std;

double fi(double& x, double j) // obliczanie wartosci funkcji fi_j i fi_k
{
    return pow(x, j);
}

double wielomian(double x, vector <double>& a)
{
    double wynik = 0;
    for (int i = 0; i < a.size(); ++i)
    {
        wynik += fi(x, i) * a[i];
    }
    return wynik;
}

int PartialPivoting(vector <vector <double>>& macierz, vector <double>& x, int wiersz)
{
    double maks = -1.0; //tu bedzie element o najwiekszym module
    int p = 0; //indeksy wiersza zamienianego z rozpatrywanym
    for (int j = wiersz; j < macierz[wiersz].size(); ++j)
    {
        if (abs(macierz[j][wiersz]) > maks)
        {
            maks = abs(macierz[j][wiersz]);
            p = j; //zapisujemy nr wiersza
        }
    }
    if (maks == 0)
    {
        return -1; //uklad osobliwy, ABORT THE MISSION
    }
    else if (wiersz == p)
    {
        return 1; //uklad rozwiazywalny, ale zamiana nie jest potrzebna
    }
    else if (maks > 0) //zamiana jest potrzebna
    {
        double pomoc = 0.0;
        for (int i = wiersz; i < macierz.size(); ++i) //nie trzeba zamieniac przed elementem o nr wiersz, bo te elementy sa = 0
        {
            pomoc = macierz[wiersz][i];
            macierz[wiersz][i] = macierz[p][i];
            macierz[p][i] = pomoc;
        }
        pomoc = x[wiersz];
        x[wiersz] = x[p];
        x[p] = pomoc;
        return 1;
    }
}

vector <double> Gauss(int n, vector <vector<double>> & wspolczynniki, vector <double> & wolne)
    {
        vector <double> wyniki (n, 0);
        double mnoznik = 0.0;
        for (int i = 0; i < n - 1; ++i) //petla po wierszach
        {
            for (int j = i + 1; j < n; ++j) //petla po elementach wiersza != 0
            {
                int kontrol = PartialPivoting(wspolczynniki, wolne, i);
                if (kontrol == -1)
                {
                    cout << "UKLAD OSOBLIWY, PRZERWANIE DZIALANIA PROGRAMU!\n";
                }
                mnoznik = wspolczynniki[j][i] / wspolczynniki[i][i];
                for (int k = i; k < n; ++k)
```

```
                {
                    wspolczynniki[j][k] -= mnoznik * wspolczynniki[i][k];
                }
                wolne[j] -= wolne[i] * mnoznik;
            }
        }
        //czesc 2: postepowanie odwrotne
        for (int i = n - 1; i >= 0; --i)
        {
            double skladnik = 0;
            for (int k = i; k < n; k++)
            {
                skladnik += wyniki[k] * wspolczynniki[i][k];
            }
            wyniki[i] = (wolne[i] - skladnik) / wspolczynniki[i][i];
        }
        for (int i = 0; i < n; ++i)
        {
            cout << setw(10) << right << "a" << i << " = " << wyniki[i] << "\n";
        }
        return wyniki;
    }

int main()
{
    int n = 1; //stopien wielomianu
    int wezly = 8;
    vector <double> wagi(wezly, 1.0); //wagi
    vector <double> F(n + 1, 0);
    vector <double> a(n + 1, 0);
    vector <vector <double>> macierz_g(n + 1, F); //macierz
    vector <pair <double, double>> punkty;
    punkty.push_back(make_pair(1.0, 2.0));
    punkty.push_back(make_pair(2.0, 4.0));
    punkty.push_back(make_pair(3.0, 3.0));
    punkty.push_back(make_pair(4.0, 5.0));
    punkty.push_back(make_pair(5.0, 6.0));
```

```
    punkty.push_back(make_pair(6.0, 9.0));
    punkty.push_back(make_pair(7.0, 11.0));
    punkty.push_back(make_pair(8.0, 11.0));

    //uzupelnianie macierzy

    for (int k = 0; k < n + 1; ++k)
    {
        for (int j = 0; j < n + 1; ++j)
        {
            for (int i = 0; i < wezly; ++i)
            {
                macierz_g[k][j] += fi(punkty[i].first, k) * fi(punkty[i].first, j) * wagi[i];
            }
        }
    }
    for (int k = 0; k < n + 1; ++k)
    {
        for (int i = 0; i < wezly; ++i)
        {
            F[k] += fi(punkty[i].first, k) * (punkty[i].second) * wagi[i];
        }
    }
    cout << "Rozwiazanie:\n\t- Ilosc wezlow: " << wezly << "\n\t- Wspolczynniki wielomianu aproksymujacego:\n";
    a = Gauss(n + 1, macierz_g, F);
    cout << "\t -Wezly aproksymacji i ich wartosci: \n";
    for (int i = 0; i < wezly; ++i)
    {
        cout << "\t\t" << i + 1 << ":" << punkty[i].first << "; " << punkty[i].second << "\n";
    }
    cout << "obliczone wartosci funkcji aproksymujacej w wezlach aproksymacji:\n";
    for (int i = 0; i < wezly; ++i)
    {
        cout << "\t\t" << i + 1 << ":" << punkty[i].first << "; " << wielomian(punkty[i].first, a) << "\n";
    }
}
```

Za pomocą wzorów umieszczonych w instrukcji wyznaczono *macierz_g* i wektor F będące częściami układu równań liniowych, którego niewiadomymi są współczynniki wielomianu aproksymującego. Układ ten rozwiązano za pomocą metody Gaussa rozszerzonej o Partial Pivoting zaimplementowanej w trakcie laboratorium 4.

Efekt działania programu jest następujący: