

METODY NUMERYCZNE

ZADANIE 8

Artur Guniewicz

8. Znaleźć wartości funkcji

$$f(x) = \frac{1}{1+5x^2} \quad (7)$$

w punktach $-1, -1 + \frac{1}{32}, -1 + \frac{2}{32}, \dots, 1 - \frac{1}{32}, 1$, a następnie skonstruować wielomian interpolacyjny Lagrange'a oparty na tych węzłach i wartościach funkcji (7) w tych węzłach. Narysować wykres wielomianu interpolacyjnego.

Metoda: metoda Lagrange'a

Współczynniki wielomianu interpolacyjnego można obliczyć w czasie $O(n^2)$ dzięki metodzie Lagrange'a. W tym celu należy skonstruować układ równań ze wzorów:

$$\begin{pmatrix} \varphi_0(x_0) & 0 & 0 & \dots & 0 \\ 0 & \varphi_1(x_1) & 0 & \dots & 0 \\ 0 & 0 & \varphi_2(x_2) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \varphi_n(x_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$$

$$a_0 = \frac{y_0}{\varphi_0(x_0)}$$

$$a_1 = \frac{y_1}{\varphi_1(x_1)}$$

...

$$a_n = \frac{y_n}{\varphi_n(x_n)}$$

Dla danego punktu x wartość wielomianu interpolacyjnego można obliczyć ze wzoru:

$$W(x) = a_0 * \varphi_0(x) + a_1 * \varphi_1(x) + \dots + a_n * \varphi_n(x) = \\ a_0 * (x - x_1) * (x - x_2) * \dots * (x - x_n) + \dots + a_n * (x - x_0) * (x - x_1) * \dots * (x - x_{n-1})$$

Kod programu:

```
/*
*****
*
*           Artur Guniewicz - Zadanie 8
*
*
*****
*/

#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <array>
#include <fstream>

using namespace std;

#define Vector array<long double, 65>

long double calcFunction(long double x);
long double base(const Vector &x, int i, long double y);
void calcLagrange(const Vector &x, const Vector &y, Vector &a);
long double calculate(const Vector &x, const Vector &a, long double y);
void writeToFile(const Vector &x, const Vector &y, const char *f);

int main()
{
    Vector x;
    Vector y;
    Vector a;

    // wypełnianie wektora punktów
    int i = 0;
    for (long double j = -1; j <= 1; j = j + (long double)1.0 / 32)
    {
        x[i] = j;
        y[i] = calcFunction(j);
        i++;
    }
}
```

```

}

// obliczenie współczynników Lagrange'a
calcLagrange(x, y, a);

// zapisywanie punktów obliczonych z wielomianu interpolacyjnego
ofstream file;
file.open("Zadanie8_poly.txt");
for (long double i = -1; i <= 1; i = i + 0.01)
    file << i << "          " << calculate(x, a, i) << endl;

file.close();

// zapisywanie wyników do pliku
writeToFile(x, y, "Zadanie8_function.txt");

return 0;
}

// liczy wartość funkcji dla danego x
long double calcFunction(long double x)
{
    long double temp = 5 * x * x + 1.0;

    return (1.0 / temp);
}

// zwraca element bazy wektora x(y), gdzie i to i-ty element bazy
long double base(const Vector &x, int i, long double y)
{
    long double base = 1;

    for (int j = 0; j < x.size(); j++)
    {
        if (j == i)
            continue;
        else
            base *= (y - x[j]);
    }
}

```

```

    return base;
}

// obliczenie współczynników Lagrange'a
void calcLagrange(const Vector &x, const Vector &y, Vector &a)
{
    for (int i = 0; i < a.size(); i++)
        a[i] = y[i] / base(x, i, x[i]);
}

// obliczenie wartości funkcji dla x
long double calculate(const Vector &x, const Vector &a, long double y)
{
    long double result = 0;

    for (int i = 0; i < a.size(); i++)
        result += a[i] * base(x, i, y);

    return result;
}

// zapisanie wektora x i wartości funkcji y do pliku
void writeToFile(const Vector &x, const Vector &y, const char *f)
{
    ofstream file;
    file.open(f);

    for (int i = 0; i < x.size(); i++)
        file << x[i] << "        " << y[i] << endl;
}

```

Kompilacja:

g++ Zadanie8.cpp -o Zadanie8 && ./Zadanie8

Wyniki:

fragment pliku "Zadanie8_function.txt":

```
-1      0.166667
-0.96875    0.175673
-0.9375     0.185373
-0.90625    0.195831
-0.875      0.20712
-0.84375    0.219319
-0.8125     0.232516
-0.78125    0.246806
-0.75       0.262295
-0.71875    0.279095
-0.6875     0.297329
-0.65625    0.317126
-0.625      0.338624
-0.59375    0.361965
-0.5625     0.387292
-0.53125    0.414743
-0.5        0.444444
-0.46875    0.476501
-0.4375     0.510978
-0.40625    0.547887
-0.375      0.587156
-0.34375    0.628607
-0.3125     0.671916
-0.28125    0.716585
-0.25       0.761905
-0.21875    0.806935
-0.1875     0.850498
-0.15625    0.89121
```

fragment pliku "Zadanie8_poly.txt":

```
-1      0.166667
-0.99    -3.51497
-0.98    -0.883208
-0.97     0.132659
-0.96     0.260362
-0.95     0.214928
-0.94     0.187622
-0.93     0.184751
-0.92     0.189471
-0.91     0.19429
-0.9      0.198177
-0.89     0.201704
-0.88     0.205277
-0.87     0.208998
-0.86     0.212847
-0.85     0.2168
-0.84     0.220849
-0.83     0.224998
-0.82     0.229253
-0.81     0.233618
-0.8      0.238095
-0.79     0.242689
-0.78     0.247402
-0.77     0.252239
-0.76     0.257202
-0.75     0.262295
-0.74     0.267523
-0.73     0.272889
```

Wykres:

uruchomienie z terminalu: `gnuplot --persist`

skrypt do tworzenia wykresu:

```
gnuplot> set title "Wykres wielomianu interpolacyjnego i rzeczywistego"  
gnuplot> set xlabel "x"  
gnuplot> set ylabel "y"  
gnuplot> set xrange [-1:1]  
gnuplot> plot 'Zadanie8_poly.txt' with lines title 'Wielomian interpolacyjny' ,  
'Zadanie8_function.txt' title 'Wielomian rzeczywisty' with points pointtype 6
```

```
GNUPLOT  
Version 5.2 patchlevel 2    last modified 2017-11-01  
  
Copyright (C) 1986-1993, 1998, 2004, 2007-2017  
Thomas Williams, Colin Kelley and many others  
  
gnuplot home:      http://www.gnuplot.info  
faq, bugs, etc:   type "help FAQ"  
immediate help:   type "help" (plot window: hit 'h')  
  
Terminal type is now 'qt'  
gnuplot> set title "Wykres wielomianu interpolacyjnego i rzeczywistego"  
gnuplot> set xlabel "x"  
gnuplot> set ylabel "y"  
gnuplot> set xrange [-1:1]  
gnuplot> plot 'Zadanie8_poly.txt' with lines title 'Wielomian interpolacyjny' , 'Zadanie8_function.txt' title 'Wielomian rzeczywisty' with points pointtype 6  
gnuplot>
```

