

Wojciech Ładyga - zadanie 11

Język technologia: c++

Aby wyliczyć jawnie współczynniki wielomianu interpolacyjnego Lagrange's na podstawie zadanych węzłów i funkcji należało wyliczyć wartości funkcji na podstawie podanych węzłów, a następnie rozwiązać taki układ. Wyliczone wyniki pozwoliły wygenerować odpowiedni wykres.

Aby narysować wykres zastosowałem polecenia

```
./a.out > interp.dat  
graph -T ps < interp.dat > interp.ps
```

Wykres został wygenerowany na przedziale $\langle -1, 1 \rangle$ przeskakując od -1 do 1 o 0.001, korzystając z interpolacji.

Generowanie wykresu oparłem o przykład ze strony <https://www.gnu.org/software/gsl/doc/html/interp.html>. Wykorzystywany jest tam gnu plotunit.

W tym zadaniu mamy do dyspozycji 65 węzłów gdyż zaczynając od -1, $-1 + 1/32 + \dots$, i przechodząc aż do 1 skaczemy o $1/32$. Więc wyliczając mamy 65 możliwych argumentów funkcji.

Przykład wyliczenia ilości węzłów stosując excel:

	A	B	C	D	E	F	G
1	-1		0		-0,03125		31
2	-0,96875		1		0		32
3	-0,9375		2		0,03125		33
4	-0,90625		3		0,0625		34
5	-0,875		4		0,09375		35
6	-0,84375		5		0,125		36
7	-0,8125		6		0,15625		37
8	-0,78125		7		0,1875		38
9	-0,75		8		0,21875		39
10	-0,71875		9		0,25		40
11	-0,6875		10		0,28125		41
12	-0,65625		11		0,3125		42
13	-0,625		12		0,34375		43
14	-0,59375		13		0,375		44
15	-0,5625		14		0,40625		45
16	-0,53125		15		0,4375		46
17	-0,5		16		0,46875		47
18	-0,46875		17		0,5		48
19	-0,4375		18		0,53125		49
20	-0,40625		19		0,5625		50
21	-0,375		20		0,59375		51
22	-0,34375		21		0,625		52
23	-0,3125		22		0,65625		53
24	-0,28125		23		0,6875		54
25	-0,25		24		0,71875		55
26	-0,21875		25		0,75		56
27	-0,1875		26		0,78125		57
28	-0,15625		27		0,8125		58
29	-0,125		28		0,84375		59
30	-0,09375		29		0,875		60
31	-0,0625		30		0,90625		61
32					0,9375		62
33					0,96875		63
34					1		64
35							

z formułą $= -1 + (CX/32)$ gdzie cx to numer komurek od 1-65

Kod programu:

```

/*
 * @Author: Wojciech Ladyga
 * @Date: 2019-01-29
 * @Description: Zad 11
 */
#include <iostream>
#include <iomanip>
#include <cmath>

```

```

//interpolacja wielomianowa - lagrangea
using namespace std;
const int N = 66; //wezly

class Lagrange
{
    double suma;
    double tmp;

public:
    double f(double x)
    {
        return 1 / (1 + 5 * pow(x, 2));
    }

    //metoda licząca lagrangea
    double lag(double xi[], double yi[], double xx)
    {
        suma = 0.0;

        for (int i = 0; i < N; i++)
        {
            tmp = yi[i];
            for (int j = 0; j < N; j++)
            {
                if (j != i)
                    tmp = tmp * (xx - xi[j]) / (xi[i] - xi[j]);
            }
            suma += tmp;
        }
    }
};

int main()
{
    Lagrange lagrange;
    double xi[N], yi[N];

    printf("#m=0,S=2\n");

    for (int i = 0; i < N; i++)
    {
        xi[i] = -1.0 + (i / 32.0);
        yi[i] = lagrange.f(xi[i]);
        printf("%g %g\n", xi[i], yi[i]);
    }
    printf("#m=1,S=0\n");

    for (double xx = -0.9; xx < 1.0; xx += 0.001)
    {
        lagrange.lag(xi, yi, xx);
    }
}

```

```
    printf("%g %g\n", xx, lagrange.lag(xi, yi, xx));  
}  
//cout << lagrange.lag(xi, yi);  
return 0;  
}
```

Wyniki działania programu to:

gdzie

-linia ciągła reprezentuje interpolację

-x reprezentują węzły i wartości funkcji wyliczone na ich podstawie

