

## Wojciech Ładyga - zadanie 19

Język technologia: c++, GSL

Program miał na zadanie liczyć naturalny spaliny kubiczny na podstawie podanych 65 węzłów i funkcji. Dzięki funkcji i węzłom mogło się wyliczyć wartości funkcji. Mając wartości funkcji i węzły można wyliczyć spaliny i wygenerować funkcję.

Korzystając z GSL wyliczyłem spaliny kubiczny i wygenerowałem wykres. Wykres został wygenerowany na przedziale  $<-1,1>$  przeskakując od -1 do 1 o 0.01. Do wygenerowania wykresu skorzystałem z metody GNU plotutils:

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

Wyniki spaliny zapisałem do pliku i później wkleiłem do tego pdfa

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 exela:

	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: 2018-12-27
 * @Description: Zad 12
 */
#include <iostream>
#include <iomanip>
#include <fstream>
#include <gsl/gsl_linalg.h>

```

```

#include <gsl/gsl_spline.h>

using namespace std;
const int N = 65; //wezly
double f(double x);
void splajn();

int main()
{
    splajn();
    return 0;
}

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

void splajn()
{
    double xi[N], yi[N];

    printf("#m=0,S=17\n");

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

    //inicjacja obszaru roboczego do wyliczania splajnu
    //https://www.gnu.org/software/gsl/doc/html/interp.html
    gsl_interp_accel *acc = gsl_interp_accel_alloc();
    gsl_spline *spline = gsl_spline_alloc(gsl_interp_cspline, N);
    gsl_spline_init(spline, xi, yi, N);

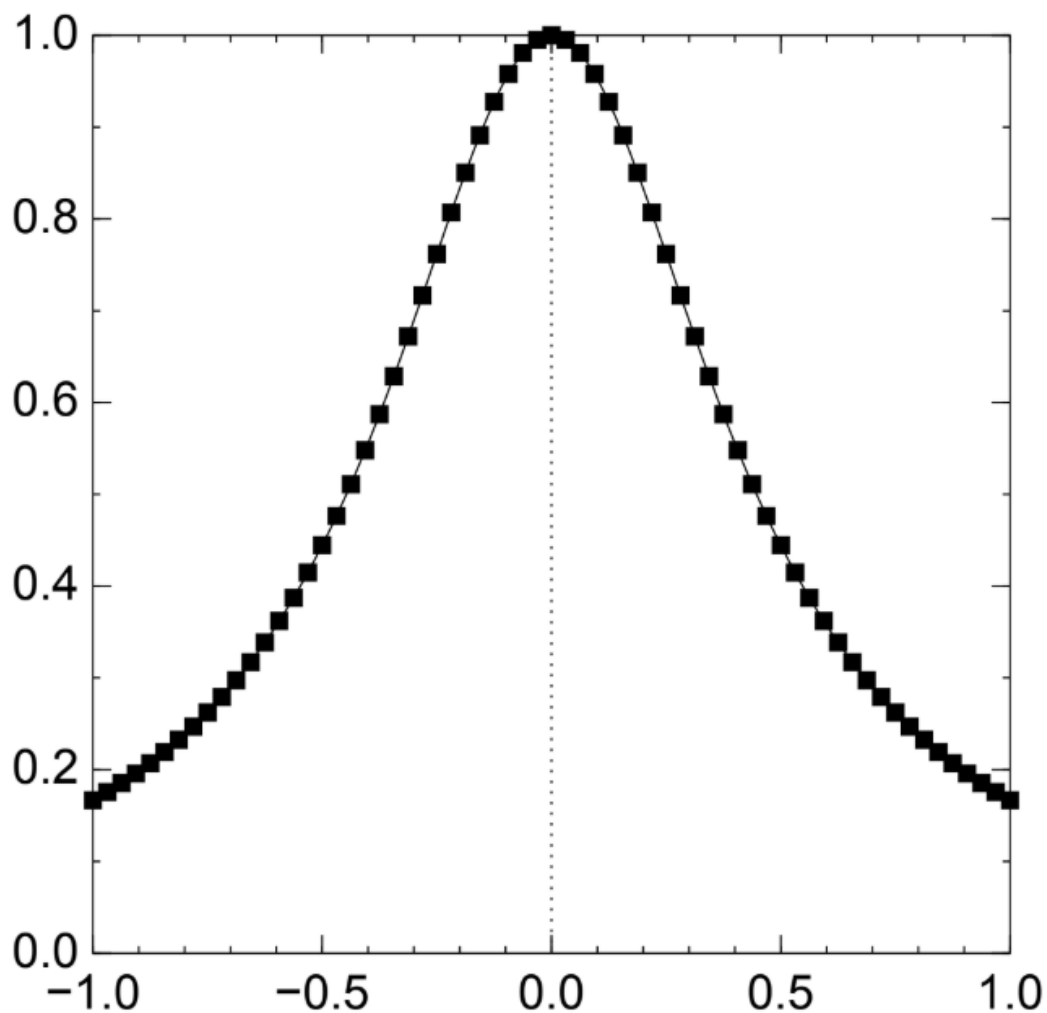
    double j;
    //zapis splajnu do pliku jak i wyliczanie wszystkich wartości stosując splajn
    ofstream myfile;
    myfile.open("wynik.txt");
    for (double i = xi[0]; i < xi[N - 1]; i += 0.01)
    {
        j = gsl_spline_eval(spline, i, acc);
        printf("%g %g\n", i, j);
        myfile << fixed << setprecision(5) << i << " " << j << endl;
    }
    myfile.close();

    //czyszczenie

```

```
gsl_spline_free(spline);  
gsl_interp_accel_free(acc);  
}
```

Wyniki działania programu to:



Oraz wynik splajnu kubicznego:

```
-1.00000 0.16667  
-0.99000 0.16951  
-0.98000 0.17238  
-0.97000 0.17530  
-0.96000 0.17831  
-0.95000 0.18140  
-0.94000 0.18457  
-0.93000 0.18781  
-0.92000 0.19113  
-0.91000 0.19453  
-0.90000 0.19802
```

-0.89000 0.20159  
-0.88000 0.20525  
-0.87000 0.20901  
-0.86000 0.21286  
-0.85000 0.21680  
-0.84000 0.22085  
-0.83000 0.22500  
-0.82000 0.22925  
-0.81000 0.23362  
-0.80000 0.23810  
-0.79000 0.24269  
-0.78000 0.24740  
-0.77000 0.25224  
-0.76000 0.25720  
-0.75000 0.26230  
-0.74000 0.26752  
-0.73000 0.27289  
-0.72000 0.27840  
-0.71000 0.28405  
-0.70000 0.28986  
-0.69000 0.29581  
-0.68000 0.30193  
-0.67000 0.30821  
-0.66000 0.31466  
-0.65000 0.32129  
-0.64000 0.32808  
-0.63000 0.33506  
-0.62000 0.34223  
-0.61000 0.34959  
-0.60000 0.35714  
-0.59000 0.36490  
-0.58000 0.37286  
-0.57000 0.38102  
-0.56000 0.38941  
-0.55000 0.39801  
-0.54000 0.40683  
-0.53000 0.41589  
-0.52000 0.42517  
-0.51000 0.43469  
-0.50000 0.44444  
-0.49000 0.45444  
-0.48000 0.46468  
-0.47000 0.47517  
-0.46000 0.48591  
-0.45000 0.49689  
-0.44000 0.50813  
-0.43000 0.51962  
-0.42000 0.53135  
-0.41000 0.54333  
-0.40000 0.55556  
-0.39000 0.56802

-0.38000 0.58072  
-0.37000 0.59365  
-0.36000 0.60680  
-0.35000 0.62016  
-0.34000 0.63371  
-0.33000 0.64746  
-0.32000 0.66138  
-0.31000 0.67545  
-0.30000 0.68966  
-0.29000 0.70398  
-0.28000 0.71839  
-0.27000 0.73287  
-0.26000 0.74738  
-0.25000 0.76190  
-0.24000 0.77640  
-0.23000 0.79083  
-0.22000 0.80515  
-0.21000 0.81934  
-0.20000 0.83333  
-0.19000 0.84710  
-0.18000 0.86059  
-0.17000 0.87374  
-0.16000 0.88652  
-0.15000 0.89888  
-0.14000 0.91075  
-0.13000 0.92208  
-0.12000 0.93284  
-0.11000 0.94295  
-0.10000 0.95238  
-0.09000 0.96108  
-0.08000 0.96899  
-0.07000 0.97609  
-0.06000 0.98232  
-0.05000 0.98765  
-0.04000 0.99206  
-0.03000 0.99552  
-0.02000 0.99800  
-0.01000 0.99950  
0.00000 1.00000  
0.01000 0.99950  
0.02000 0.99800  
0.03000 0.99552  
0.04000 0.99206  
0.05000 0.98765  
0.06000 0.98232  
0.07000 0.97609  
0.08000 0.96899  
0.09000 0.96108  
0.10000 0.95238  
0.11000 0.94295  
0.12000 0.93284

0.13000 0.92208  
0.14000 0.91075  
0.15000 0.89888  
0.16000 0.88652  
0.17000 0.87374  
0.18000 0.86059  
0.19000 0.84710  
0.20000 0.83333  
0.21000 0.81934  
0.22000 0.80515  
0.23000 0.79083  
0.24000 0.77640  
0.25000 0.76190  
0.26000 0.74738  
0.27000 0.73287  
0.28000 0.71839  
0.29000 0.70398  
0.30000 0.68966  
0.31000 0.67545  
0.32000 0.66138  
0.33000 0.64746  
0.34000 0.63371  
0.35000 0.62016  
0.36000 0.60680  
0.37000 0.59365  
0.38000 0.58072  
0.39000 0.56802  
0.40000 0.55556  
0.41000 0.54333  
0.42000 0.53135  
0.43000 0.51962  
0.44000 0.50813  
0.45000 0.49689  
0.46000 0.48591  
0.47000 0.47517  
0.48000 0.46468  
0.49000 0.45444  
0.50000 0.44444  
0.51000 0.43469  
0.52000 0.42517  
0.53000 0.41589  
0.54000 0.40683  
0.55000 0.39801  
0.56000 0.38941  
0.57000 0.38102  
0.58000 0.37286  
0.59000 0.36490  
0.60000 0.35714  
0.61000 0.34959  
0.62000 0.34223  
0.63000 0.33506

0.64000	0.32808
0.65000	0.32129
0.66000	0.31466
0.67000	0.30821
0.68000	0.30193
0.69000	0.29581
0.70000	0.28986
0.71000	0.28405
0.72000	0.27840
0.73000	0.27289
0.74000	0.26752
0.75000	0.26230
0.76000	0.25720
0.77000	0.25224
0.78000	0.24740
0.79000	0.24269
0.80000	0.23810
0.81000	0.23362
0.82000	0.22925
0.83000	0.22500
0.84000	0.22085
0.85000	0.21680
0.86000	0.21286
0.87000	0.20901
0.88000	0.20525
0.89000	0.20159
0.90000	0.19802
0.91000	0.19453
0.92000	0.19113
0.93000	0.18781
0.94000	0.18457
0.95000	0.18140
0.96000	0.17831
0.97000	0.17530
0.98000	0.17238
0.99000	0.16951