

Wojciech Ładyga - zadanie 19

Język technologia: c++

Aby wykonać ten program należało zaczytać wartości z pliku. W moim programie zaczytałem wartości do 2 wektorów `x` i `y`. Następnie pobrałem długość wektora i zamieściłem w zmiennej `N`. Lecz aby użyć gSL-a należy stosować zwykłe tablice więc musiałem przekonwertować wektor na zwykłą tablicę stopując metodę `copy(x.begin(), x.end(), xx);`.

Korzystając z GSL wyliczyłem splain i wygenerowałem wykres. Do wygenerowania wykresu skorzystałem z metody GNU plotutils:

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

Wyniki splajnu zapisałem do pliku i później wkleiłem do tego pdfa.

Kod programu:

```
/*
 * @Author: Wojciech Ładyga
 * @Date: 2018-12-27
 * @Description: Zad 14
 */
#include <iostream>
#include <iomanip>
#include <string>
#include <fstream>
#include <vector>
#include <gsl/gsl_spline.h>

using namespace std;
void readFile();
void splajn();
//zmienne globalne przechowujące x i y pobrane z pliku
vector<double> x, y;

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

//czytam z pliku
void readFile()
{
```

```

    fstream plik;
    plik.open("dane.txt", ios::in);
    if (plik.good() == true)
    {
        while (!plik.eof()) //aż do końca
        {
            double tmpX, tmpY;
            plik >> tmpX >> tmpY;
            x.push_back(tmpX);
            y.push_back(tmpY);
        }
        plik.close();
    }
}

void splajn()
{
    int N = x.size() - 1;
    //convert vector to array
    double xx[N], yy[N];
    copy(x.begin(), x.end(), xx);
    copy(y.begin(), y.end(), yy);

    printf("#m=0,S=2\n");
    for (int i = 0; i < N; i++)
    {
        printf("%g %g\n", xx[i], yy[i]);
    }
    printf("#m=1,S=0\n");

    //pole robocze do wyliczania splajnu
    gsl_interp_accel *acc = gsl_interp_accel_alloc();
    gsl_spline *spline = gsl_spline_alloc(gsl_interp_cspline, N);

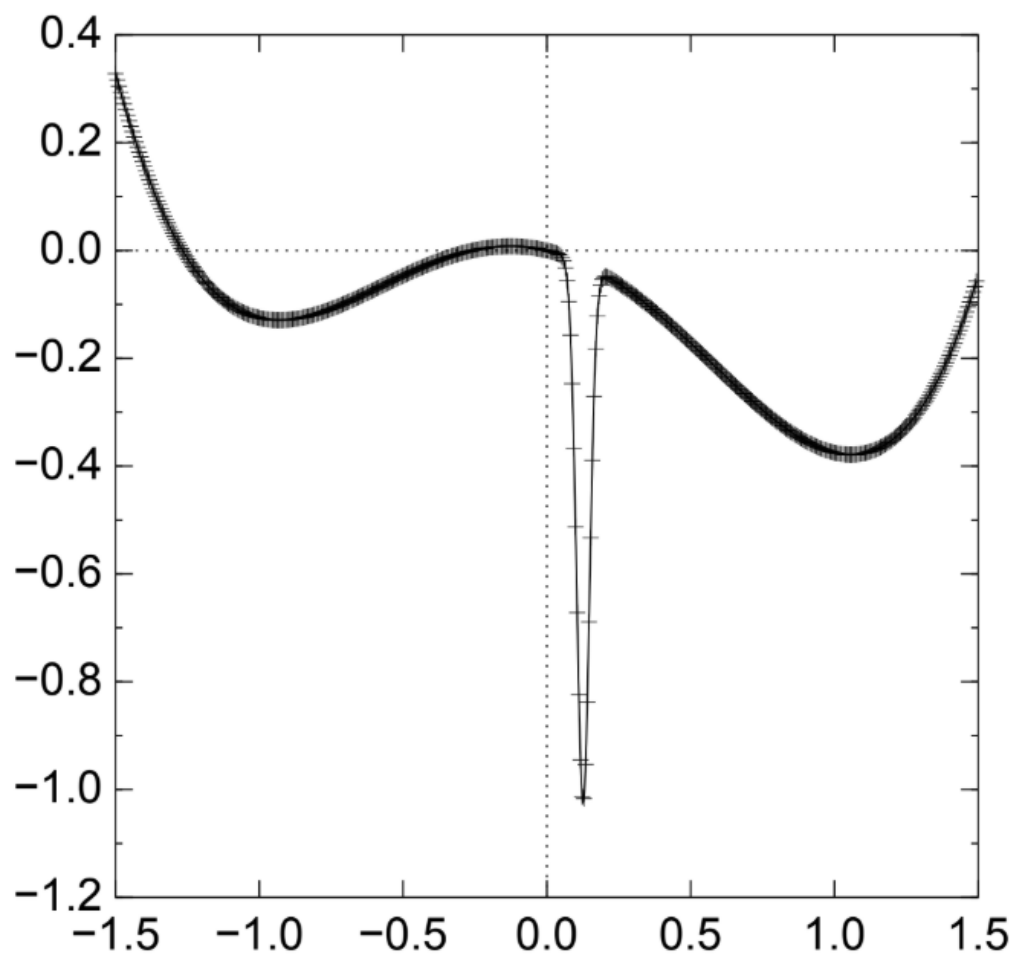
    gsl_spline_init(spline, xx, yy, N);

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

    gsl_spline_free(spline);
    gsl_interp_accel_free(acc);
}

```

Wyniki działania programu to:



Oraz wynik splajnu kubicznego:

```
-1.5 0.32812
-1.49000 0.30841
-1.48000 0.28926
-1.47000 0.27067
-1.46000 0.25263
-1.45000 0.23513
-1.44000 0.21815
-1.43000 0.20170
-1.42000 0.18577
-1.41000 0.17034
-1.40000 0.15540
-1.39000 0.14095
-1.38000 0.12698
-1.37000 0.11349
-1.36000 0.10046
-1.35000 0.08788
```

-1.34000 0.07574
-1.33000 0.06405
-1.32000 0.05279
-1.31000 0.04195
-1.30000 0.03152
-1.29000 0.02151
-1.28000 0.01189
-1.27000 0.00266
-1.26000 -0.00618
-1.25000 -0.01465
-1.24000 -0.02275
-1.23000 -0.03048
-1.22000 -0.03787
-1.21000 -0.04490
-1.20000 -0.05160
-1.19000 -0.05797
-1.18000 -0.06401
-1.17000 -0.06973
-1.16000 -0.07514
-1.15000 -0.08025
-1.14000 -0.08506
-1.13000 -0.08958
-1.12000 -0.09382
-1.11000 -0.09778
-1.10000 -0.10148
-1.09000 -0.10490
-1.08000 -0.10808
-1.07000 -0.11100
-1.06000 -0.11368
-1.05000 -0.11612
-1.04000 -0.11834
-1.03000 -0.12032
-1.02000 -0.12209
-1.01000 -0.12365
-1.00000 -0.12500
-0.99000 -0.12615
-0.98000 -0.12711
-0.97000 -0.12788
-0.96000 -0.12846
-0.95000 -0.12887
-0.94000 -0.12911
-0.93000 -0.12919
-0.92000 -0.12910
-0.91000 -0.12886
-0.90000 -0.12847
-0.89000 -0.12794
-0.88000 -0.12728
-0.87000 -0.12648
-0.86000 -0.12555
-0.85000 -0.12450
-0.84000 -0.12333

-0.83000 -0.12205
-0.82000 -0.12067
-0.81000 -0.11918
-0.80000 -0.11760
-0.79000 -0.11592
-0.78000 -0.11416
-0.77000 -0.11232
-0.76000 -0.11039
-0.75000 -0.10840
-0.74000 -0.10633
-0.73000 -0.10420
-0.72000 -0.10202
-0.71000 -0.09977
-0.70000 -0.09747
-0.69000 -0.09513
-0.68000 -0.09275
-0.67000 -0.09032
-0.66000 -0.08786
-0.65000 -0.08537
-0.64000 -0.08286
-0.63000 -0.08032
-0.62000 -0.07776
-0.61000 -0.07519
-0.60000 -0.07260
-0.59000 -0.07001
-0.58000 -0.06741
-0.57000 -0.06481
-0.56000 -0.06221
-0.55000 -0.05962
-0.54000 -0.05704
-0.53000 -0.05447
-0.52000 -0.05192
-0.51000 -0.04939
-0.50000 -0.04687
-0.49000 -0.04439
-0.48000 -0.04193
-0.47000 -0.03950
-0.46000 -0.03711
-0.45000 -0.03475
-0.44000 -0.03243
-0.43000 -0.03015
-0.42000 -0.02792
-0.41000 -0.02574
-0.40000 -0.02360
-0.39000 -0.02152
-0.38000 -0.01949
-0.37000 -0.01751
-0.36000 -0.01560
-0.35000 -0.01375
-0.34000 -0.01196
-0.33000 -0.01024

-0.32000	-0.00858
-0.31000	-0.00699
-0.30000	-0.00547
-0.29000	-0.00403
-0.28000	-0.00266
-0.27000	-0.00137
-0.26000	-0.00016
-0.25000	0.00098
-0.24000	0.00203
-0.23000	0.00300
-0.22000	0.00389
-0.21000	0.00469
-0.20000	0.00540
-0.19000	0.00603
-0.18000	0.00656
-0.17000	0.00701
-0.16000	0.00736
-0.15000	0.00763
-0.14000	0.00780
-0.13000	0.00787
-0.12000	0.00785
-0.11000	0.00774
-0.10000	0.00753
-0.09000	0.00722
-0.08000	0.00681
-0.07000	0.00631
-0.06000	0.00570
-0.05000	0.00500
-0.04000	0.00420
-0.03000	0.00330
-0.02000	0.00230
-0.01000	0.00120
0.00000	-0.00000
0.01000	-0.00130
0.02000	-0.00271
0.03000	-0.00430
0.04000	-0.00640
0.05000	-0.01055
0.06000	-0.02194
0.07000	-0.05412
0.08000	-0.13249
0.09000	-0.28676
0.10000	-0.52322
0.11000	-0.79118
0.12000	-0.98547
0.13000	-1.00960
0.14000	-0.85174
0.15000	-0.59500
0.16000	-0.34975
0.17000	-0.18117
0.18000	-0.09323

0.19000	-0.05834
0.20000	-0.04885
0.21000	-0.04869
0.22000	-0.05126
0.23000	-0.05452
0.24000	-0.05797
0.25000	-0.06152
0.26000	-0.06516
0.27000	-0.06887
0.28000	-0.07266
0.29000	-0.07653
0.30000	-0.08048
0.31000	-0.08449
0.32000	-0.08858
0.33000	-0.09274
0.34000	-0.09696
0.35000	-0.10125
0.36000	-0.10560
0.37000	-0.11001
0.38000	-0.11449
0.39000	-0.11902
0.40000	-0.12360
0.41000	-0.12824
0.42000	-0.13292
0.43000	-0.13765
0.44000	-0.14243
0.45000	-0.14725
0.46000	-0.15211
0.47000	-0.15700
0.48000	-0.16193
0.49000	-0.16689
0.50000	-0.17187
0.51000	-0.17689
0.52000	-0.18192
0.53000	-0.18697
0.54000	-0.19204
0.55000	-0.19712
0.56000	-0.20221
0.57000	-0.20731
0.58000	-0.21241
0.59000	-0.21751
0.60000	-0.22260
0.61000	-0.22769
0.62000	-0.23276
0.63000	-0.23782
0.64000	-0.24286
0.65000	-0.24787
0.66000	-0.25286
0.67000	-0.25782
0.68000	-0.26275
0.69000	-0.26763

0.70000	-0.27248
0.71000	-0.27727
0.72000	-0.28202
0.73000	-0.28670
0.74000	-0.29133
0.75000	-0.29590
0.76000	-0.30039
0.77000	-0.30482
0.78000	-0.30916
0.79000	-0.31342
0.80000	-0.31760
0.81000	-0.32168
0.82000	-0.32567
0.83000	-0.32955
0.84000	-0.33333
0.85000	-0.33700
0.86000	-0.34055
0.87000	-0.34398
0.88000	-0.34728
0.89000	-0.35044
0.90000	-0.35347
0.91000	-0.35636
0.92000	-0.35910
0.93000	-0.36169
0.94000	-0.36411
0.95000	-0.36637
0.96000	-0.36846
0.97000	-0.37038
0.98000	-0.37211
0.99000	-0.37365
1.00000	-0.37500
1.01000	-0.37615
1.02000	-0.37709
1.03000	-0.37782
1.04000	-0.37834
1.05000	-0.37862
1.06000	-0.37868
1.07000	-0.37850
1.08000	-0.37808
1.09000	-0.37740
1.10000	-0.37647
1.11000	-0.37528
1.12000	-0.37382
1.13000	-0.37208
1.14000	-0.37006
1.15000	-0.36775
1.16000	-0.36514
1.17000	-0.36223
1.18000	-0.35901
1.19000	-0.35547
1.20000	-0.35160

1.21000	-0.34740
1.22000	-0.34287
1.23000	-0.33798
1.24000	-0.33275
1.25000	-0.32715
1.26000	-0.32118
1.27000	-0.31484
1.28000	-0.30811
1.29000	-0.30099
1.30000	-0.29348
1.31000	-0.28555
1.32000	-0.27721
1.33000	-0.26845
1.34000	-0.25926
1.35000	-0.24962
1.36000	-0.23954
1.37000	-0.22901
1.38000	-0.21802
1.39000	-0.20655
1.40000	-0.19460
1.41000	-0.18216
1.42000	-0.16923
1.43000	-0.15580
1.44000	-0.14185
1.45000	-0.12737
1.46000	-0.11237
1.47000	-0.09683
1.48000	-0.08074
1.49000	-0.06408