

Sprawozdanie

programu aprox

Szymon Półtorak i Sebastian Sikorski

Spis treści

1	Opis teoretyczny zagadnienia	2
2	Opis wywołania programu	3
3	Testy dla różnych zestawów wejściowych	5
4	Porównanie z działającą aproksymacją	6
5	Wnioski	7
6	Błędy znalezione w programie i ich poprawa	7
7	Źródła	7

1 Opis teoretyczny zagadnienia

Wielomiany Hermite’a – wielomiany o współczynnikach rzeczywistych, będące rozwiązaniem równania rekurencyjnego

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$$

Rysunek 1: Wzór rekurencyjny

$$\begin{aligned}H_0(x) &= 1, \\H_1(x) &= 2x.\end{aligned}$$

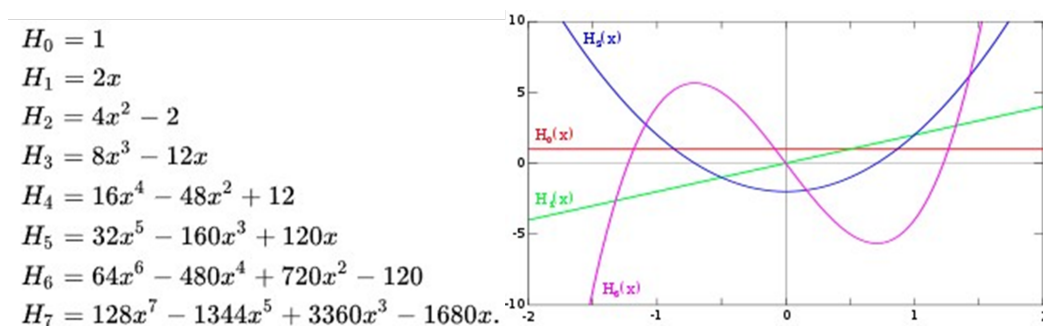
Rysunek 2: Warunki początkowe

Pierwszy z tych wzorów bywa nazywany wzorem Rodrigueza: Wykresy czterech pierwszych

$$\begin{aligned}H_n(x) &= (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} \\H_n(x) &= \frac{2^n}{\sqrt{\pi}} \int_{-\infty}^{\infty} (x + it)^n e^{-t^2} dt \\H_n(x) &= \left. \frac{d^n}{dt^n} e^{-t^2 + 2xt} \right|_{t=0}\end{aligned}$$

Rysunek 3: Wzory wielomianów Hermite’a

wielomianów oraz wzory pierwszych siedmiu:



Rysunek 4: Wykresy pierwszych czterech wielomianów

2 Opis wywołania programu

Aby skompilować program wystarczy użyć komendy „make”. Zostaną utworzone wszystkie pliki wykonywalne.

.git	17.01.2022 12:13	File folder	
aprx	17.01.2022 11:07	File folder	
bin	17.01.2022 12:29	File folder	
gaus	17.01.2022 12:29	File folder	
test	17.01.2022 12:29	File folder	
.gitattributes	15.12.2021 10:57	Text Document	1 KB
.gitignore	17.01.2022 11:07	Text Document	1 KB
aprox	17.01.2022 12:29	File	31 KB
gen	17.01.2022 12:29	File	17 KB
hermit	17.01.2022 12:29	File	27 KB
intrp	17.01.2022 12:29	File	27 KB
Makefile	17.01.2022 12:10	File	4 KB
prosta	17.01.2022 12:29	File	22 KB

Rysunek 5: Utworzone pliki aprox gen hermit intrp i prosta

Pliki:

- gen – odpowiada za generowanie „losowe” danych,
- aprox – program zajmujący się aproksymacją,
- hermit – aproksymacja na bazie wielomianów Hermite’a,
- intrp – interpolator na bazie wielomianów Lagrange’a,
- prosta – dopasowuje do punktów prostą najlepszego dopasowania,

Wpisując ./<wybrany program> wyświetli się instrukcja jego wywołania.

```
szymon@DESKTOP-QQRFA1Q:~/Projekt$ ./hermit
Usage: ./hermit -s spline-file [-p points-file] [ -g gnuplot-file [-f from_x -t to_x -n n_points
] ]

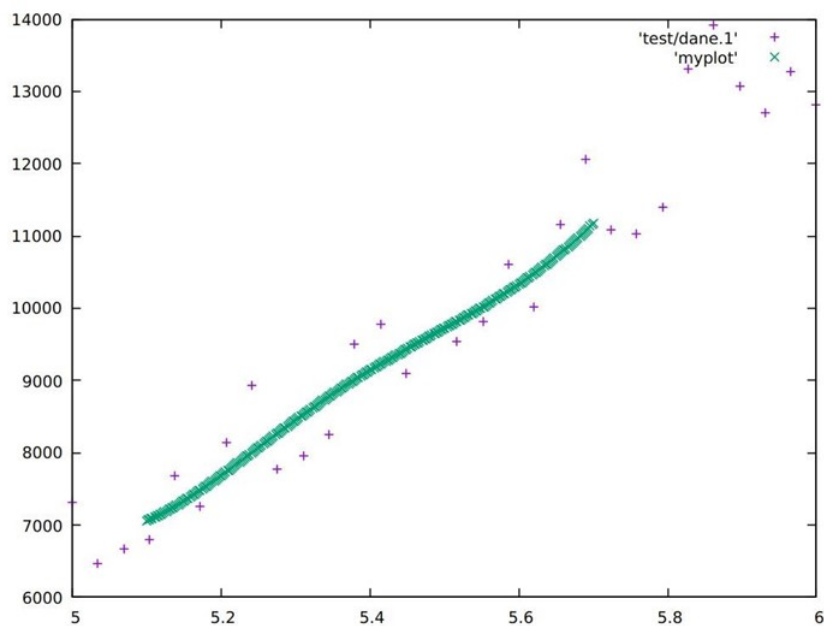
    if points-file is given then
        reads discrete 2D points from points-file
        writes spline approximation to spline-file
        - number of points should be >= 4
    else (points-file not given)
        reads spline from spline-file
    endif
    if gnuplot-file is given then
        makes table of n_points within <from_x,to_x> range
        - from_x defaults to x-coordinate of the first point in points-file,
        - to_x defaults to x-coordinate of the last point
        - n_points defaults to 100
        - n_points must be > 1
    endif
```

Rysunek 6: Wywołanie bez argumentów

Wywołanie:

```
./hermit -s spl -p test/dane.1 -g myplot -f 5.1 -t 5.7 -n 300
```

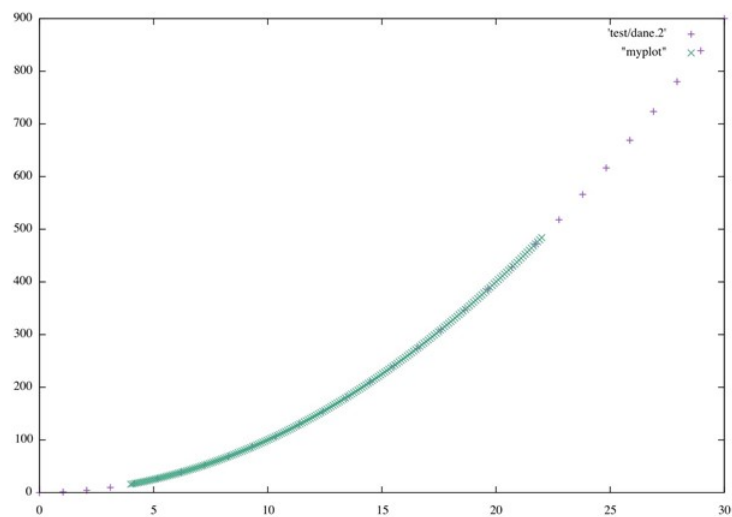
Następnie trzeba otworzyć program rysujący funkcję przykładowo gnupłota i użyć komendy plot np. plot 'test/dane.1', 'myplot'



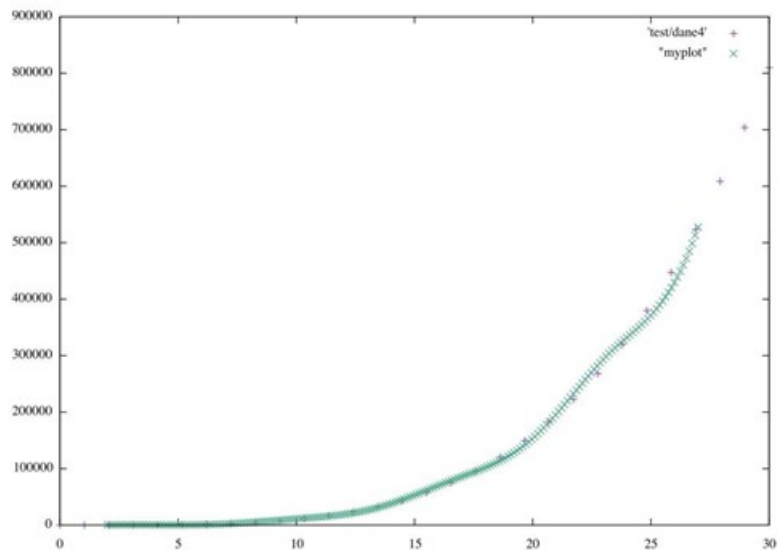
Rysunek 7: Przykładowy wynik aproksymacji

3 Testy dla różnych zestawów wejściowych

Po użyciu komendy `make test_hermit` można zobaczyć 2 różne wykresy rysowane przez naszą funkcję.

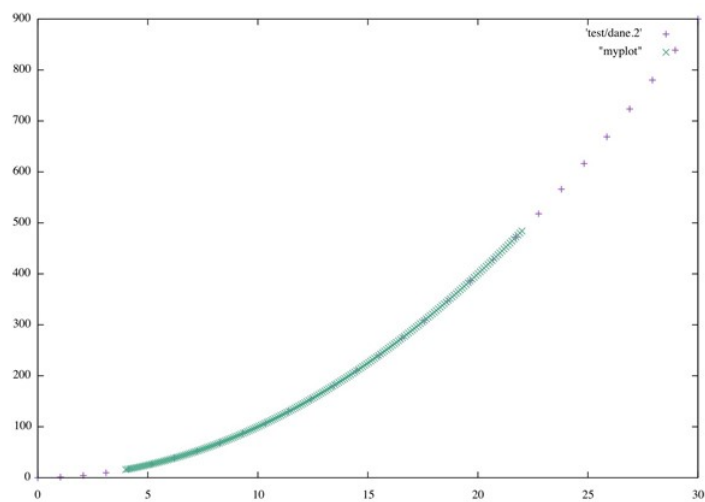


Rysunek 8: Dane testowe 1

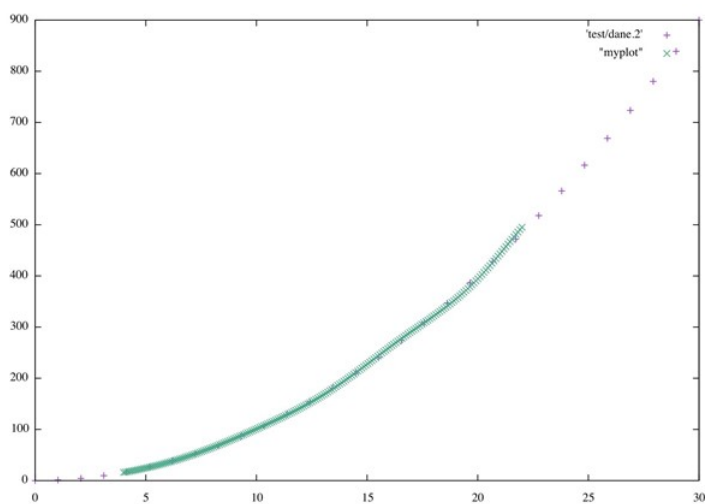


Rysunek 9: Dane testowe 2

4 Porównanie z działającą aproksymacją



Rysunek 10: Użycie wielomianów hermite'a



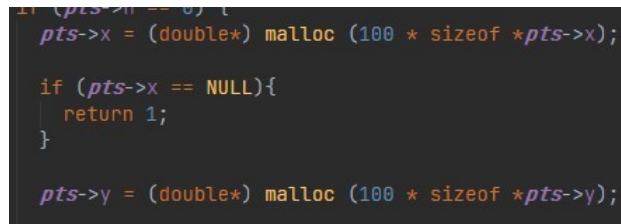
Rysunek 11: Użycie zwykłej zaproksymacji

5 Wnioski

Wprowadzenie bazy wielomianów hermite'a działa prawidłowo, niektóre funkcje (przykład wyżej) otrzymują lepsze wyniki aproksymacji niż nawet ze standardowym programem ./aprox.

6 Błędy znalezione w programie i ich poprawa

Poprawiliśmy brak rzutowania podczas mallocowania wskaźników:



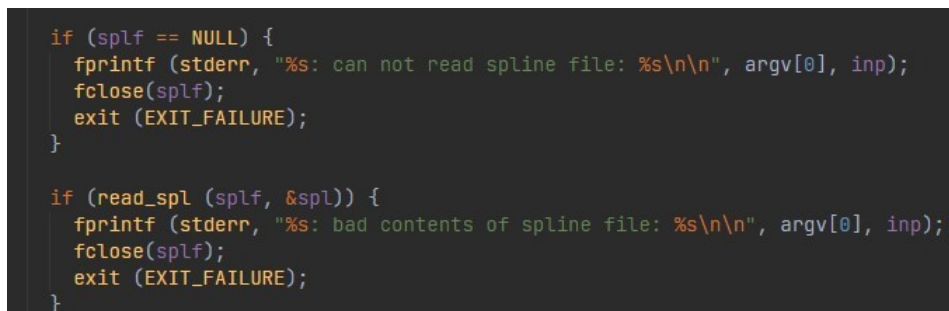
```
if (pts->n == 0) {
    pts->x = (double*) malloc (100 * sizeof *pts->x);

    if (pts->x == NULL){
        return 1;
    }

    pts->y = (double*) malloc (100 * sizeof *pts->y);
```

Rysunek 12: Użycie zwykłej zaproksymacji

Dodaliśmy zamykanie plików w przypadku obsługi fclose:

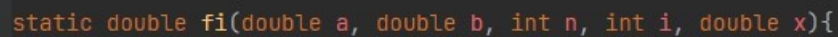


```
if (splf == NULL) {
    fprintf (stderr, "%s: can not read spline file: %s\n\n", argv[0], inp);
    fclose(splf);
    exit (EXIT_FAILURE);
}

if (read_spl (splf, &spl)) {
    fprintf (stderr, "%s: bad contents of spline file: %s\n\n", argv[0], inp);
    fclose(splf);
    exit (EXIT_FAILURE);
}
```

Rysunek 13: Użycie zwykłej zaproksymacji

Zmieniliśmy funkcje nie dekladowane w bibliotekach .h na funkcje typu static:



```
static double fi(double a, double b, int n, int i, double x){
```

Rysunek 14: Użycie zwykłej zaproksymacji

7 Źródła

1. https://pl.wikipedia.org/wiki/Wielomiany_Hermite'a