

# Sprawozdanie

Aproksymacja średnio kw. z bazą w. Laguerre

Stanislau Yarmats i Ilya Rohoutseu

Link do repozytorium github: <https://github.com/yarmatss/laguerr>

## 1. Opis teoretyczny

Wielomiany Laguerre'a – wielomiany o współczynnikach rzeczywistych zdefiniowane jako:

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (e^{-x} x^n)$$

Oto pierwsze wielomiany Laguerre'a:

n	$L_n(x)$
0	1
1	$-x + 1$
2	$\frac{1}{2} (x^2 - 4x + 2)$
3	$\frac{1}{6} (-x^3 + 9x^2 - 18x + 6)$
4	$\frac{1}{24} (x^4 - 16x^3 + 72x^2 - 96x + 24)$
5	$\frac{1}{120} (-x^5 + 25x^4 - 200x^3 + 600x^2 - 600x + 120)$
6	$\frac{1}{720} (x^6 - 36x^5 + 450x^4 - 2400x^3 + 5400x^2 - 4320x + 720)$

## 2. Wywoływanie programu

Po pierwsze trzeba skompilować bibliotekę gaus poleceniami:

```
cd gaus
```

```
make
```

Po drugie trzeba skompilować nasz program następującymi poleceniami:

```
cd ..
```

```
make all
```

```

stanisław@ubuntu:~/stud/lab/projekt/laguerr/lmp10$ ./aprox -s spl -p test/dane.1 -g myplot -f 5.1 -t 5
.7 -n 300
stanisław@ubuntu:~/stud/lab/projekt/laguerr/lmp10$ gnuplot

G N U P L O T
Version 5.4 patchlevel 2    last modified 2021-06-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2021
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')

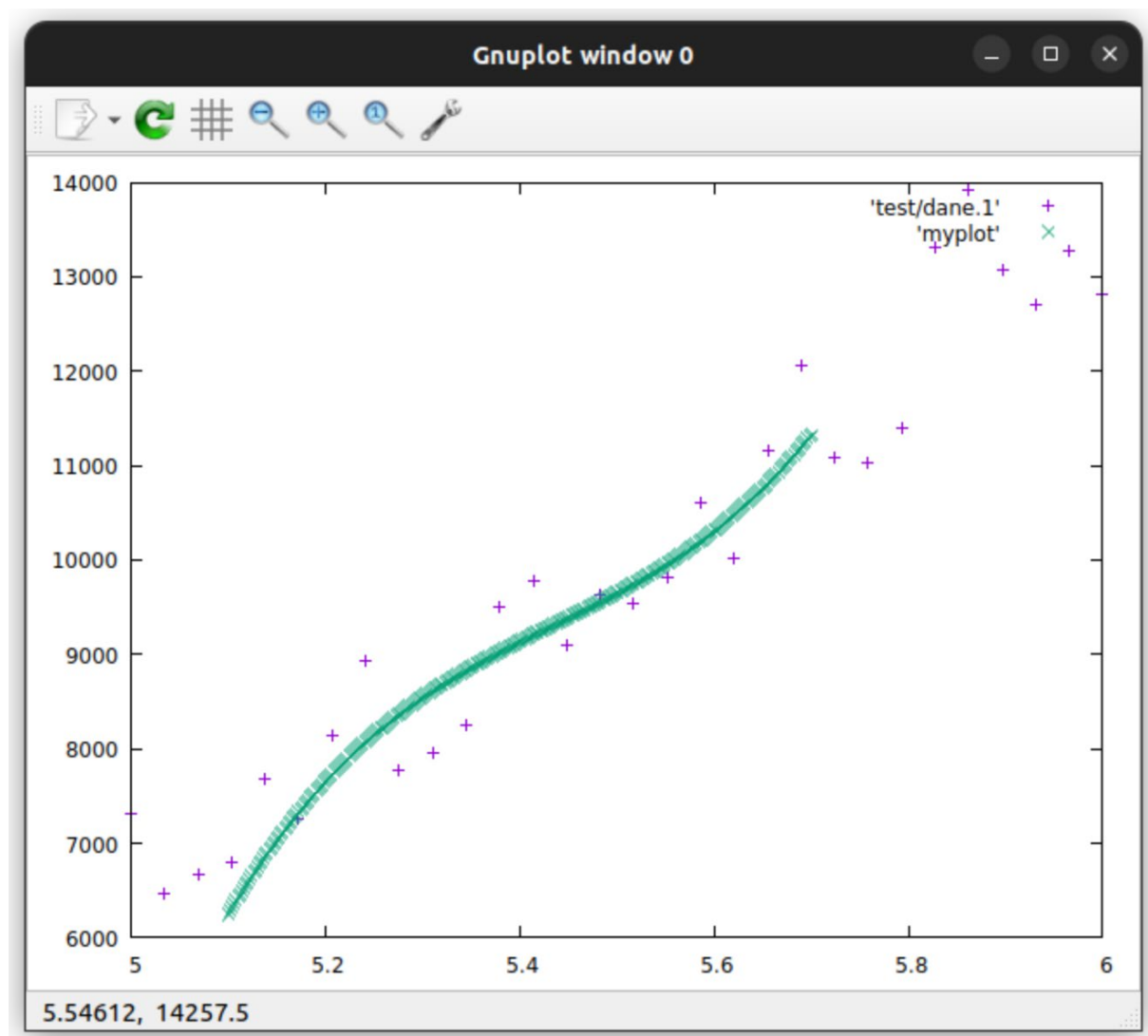
Terminal type is now 'qt'
gnuplot> plot 'test/dane.1', 'myplot'
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland any
way.
gnuplot>

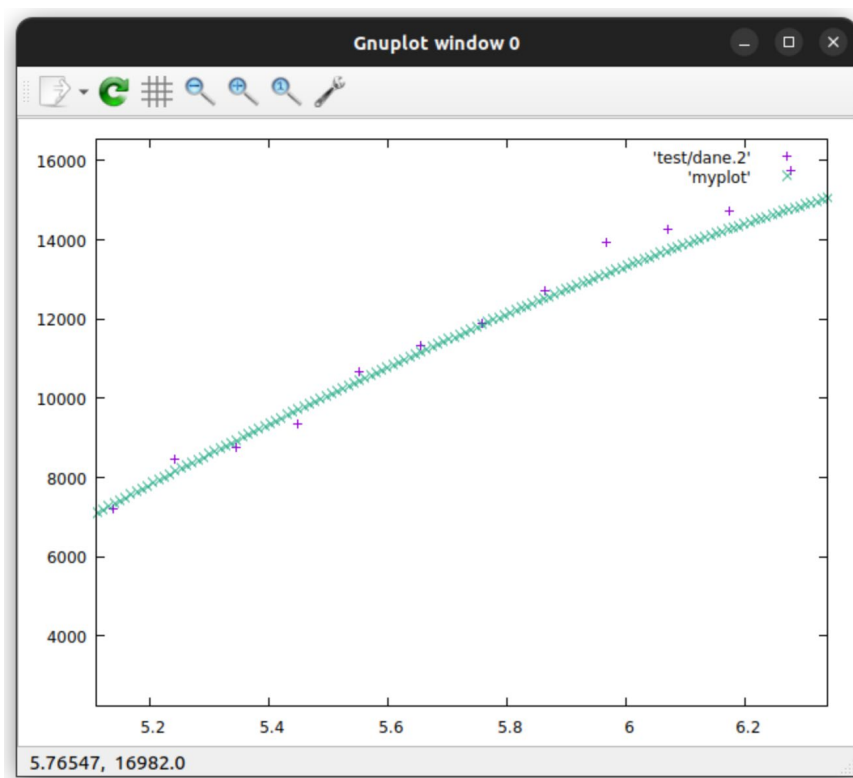
```

### 3. Testy

Plik dane.2 był stworzony za pomocą programu gen.c

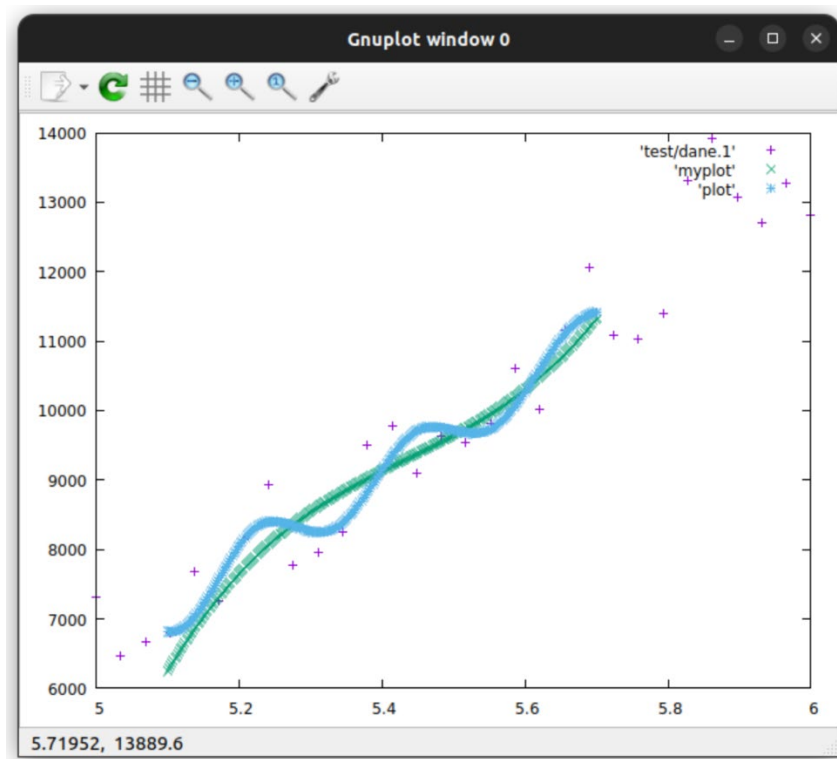
Testy z użyciem plików dane.1 i dane.2:





#### 4. Porównanie działania aproksymatorów

Porównanie nowego aproksymatora z modulem z Imp10:



Na zielono jest zaznaczony wynik działania nowego aproksymatora (myplot), na niebiesko – wynik działania modułu z Imp10 (plot).

Nietrudno zauważyć, że aproksymacja z nowej bazy jest gładzsza i bardziej prosta.

## 5. Wykryte błędy

Funkcja `free_matrix` nie została zdefiniowana w pliku `matrix.h`:

```
13 13 @@ -13,6 +13,8 @@ matrix_t * make_matrix( int rn, int cn );
14 14 matrix_t * read_matrix( FILE *in );
15 15
16 16 + void free_matrix (matrix_t * m);
17 17 +
16 18 void write_matrix( matrix_t *, FILE *out );
17 19
18 20 void put_entry_matrix( matrix_t *, int i, int j, double val );
```

Został zmieniony typ funkcji pomocniczej `xfi` z `double` na `void`:

```
113 113 }
114 114
115 115 /* Pomocnicza f. do rysowania bazy */
116 116 - double
117 117 + void
118 118 xfi(double a, double b, int n, int i, FILE *out)
119 119 {
120 120     double h = (b - a) / (n - 1);
```

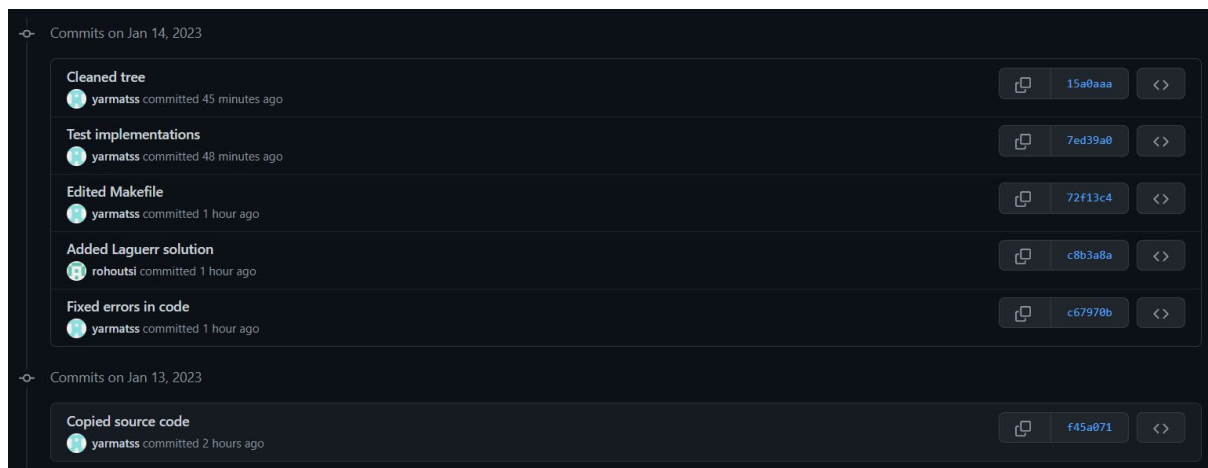
Wynik `valgrind`:

```
==25083== HEAP SUMMARY:
==25083==    in use at exit: 520 bytes in 7 blocks
==25083== total heap usage: 17 allocs, 10 frees, 16,720 bytes allocated
==25083==
==25083== LEAK SUMMARY:
==25083==    definitely lost: 520 bytes in 7 blocks
==25083==    indirectly lost: 0 bytes in 0 blocks
==25083==    possibly lost: 0 bytes in 0 blocks
==25083==    still reachable: 0 bytes in 0 blocks
==25083==    suppressed: 0 bytes in 0 blocks
==25083== Rerun with --leak-check=full to see details of leaked memory
==25083==
==25083== For lists of detected and suppressed errors, rerun with: -s
==25083== ERROR SUMMARY: 6602 errors from 6 contexts (suppressed: 0 from 0)
```

## 6. Podział pracy

Znalezienie informacji teoretycznej: Stanislau Yarmats, implementacja teorii do kodu: wspólnie – Ilya Rohoutseu i Stanislau Yarmats, wykrycie błędów – Ilya Rohoutseu, testy – Stanislau Yarmats.

## 7. Lista commitów



## 8. Wnioski

Aproksymacja średnio kwadratowa z bazą w. Laguerre jest gładzsza i bardziej prostsza niż aproksymacja wymieniona w module Imp10.

## 9. Wykorzystane źródła:

[https://en.wikipedia.org/wiki/Laguerre\\_polynomials?oldid=81223447](https://en.wikipedia.org/wiki/Laguerre_polynomials?oldid=81223447)

[https://pl.frwiki.wiki/wiki/Polyn%C3%B4me\\_de\\_Laguerre](https://pl.frwiki.wiki/wiki/Polyn%C3%B4me_de_Laguerre)