

# Języki i metody Programowania 1 – sprawozdanie z projektu – aproksymacja średniokwadratowa z bazą wielomianu 4.

Stopnia  
Michał Fereniec

## 1.Opis teoretyczny

Projekt polega na napisaniu/zmodyfikowaniu programu, który będzie dla danych punktów rysował wykres, który będzie „najbardziej pasował” do danych punktów. Aby wiedzieć jak to dopasować skorzystamy z metody aproksymacji średnio kwadratowej z bazą wielomianu 4. Stopnia. Dokładniej mówiąc to nasz aproksymator\_na\_bazie.c aproksymuje punkty przez minimalizację błędu średniokwadratowego za pomocą funkcji bazowych. W gotowym projekcie jest to funkcja bazowa dr. Chwieja z AGH. W moim przypadku będziemy używać wielomianu 4. Stopnia, tzn nasz układ będzie wyglądał następująco:

Moim celem jest znalezienie wielomianu postaci:

$$F(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m$$

Takiego, aby przybliżenie funkcji w punktach  $x_0, x_1, \dots, x_n$  było jak najlepsze.

Funkcję oceny jakości takiego wielomianu można określić w różny sposób:

- Poprzez maksymalną różnicę
- Suma wartości bezwzględnych powinna być jak najmniejsza
- Suma kwadratów różnic powinna być jak najmniejsza (nasza aproksymacja średniokwadratowa)

Ogólnie w aproksymacji średniokwadratowej wielomianowej funkcja błędu jest zdefiniowana następująco:

$$H(a_0, a_1, \dots, a_m) = \sum_{j=0}^n w(x_j) (y_j - \sum_{i=0}^m a_i x_j^i)^2$$

Współczynnik  $w(x_j)$  jest ustaloną funkcją wagową. Najczęściej przyjmuje się, że funkcja wagowa zawsze przyjmuje wartość 1 – wówczas możemy pominąć ten czynnik.

Funkcja osiąga minimum w punkcie, w którym pochodne cząstkowe względem współczynników  $a_0, a_1, \dots, a_m$  są równe 0. W celu znalezienia tego minimum należy rozwiązać układ równań:

$$\frac{\partial H}{\partial a_0} = 0$$

$$\frac{\partial H}{\partial a_1} = 0$$

...

$$\frac{\partial H}{\partial a_m} = 0$$

A po przekształceniu układ ten można sprowadzić do postaci:

$$\begin{array}{ccccccc} a_0(n+1) & + & a_1 \sum_{j=0}^n x_j & + & \dots & + & a_m \sum_{j=0}^n x_j^m & = & \sum_{j=0}^n y_j \\ a_0 \sum_{j=0}^n x_j & + & a_1 \sum_{j=0}^n x_j^2 & + & \dots & + & a_m \sum_{j=0}^n x_j^{m+1} & = & \sum_{j=0}^n y_j x_j \\ \dots & & \dots & & \dots & & \dots & & \dots \\ a_0 \sum_{j=0}^n x_j^m & + & a_1 \sum_{j=0}^n x_j^{m+1} & + & \dots & + & a_m \sum_{j=0}^n x_j^{2m} & = & \sum_{j=0}^n y_j x_j^m \end{array}$$

Tylko, że najwyższą potęgą w moim przypadku będzie 4.

Implementacja w kodzie będzie następująca:

```
double
dft(double a, double b, int n, int i, double x)
{
    double h = (b - a) / (n - 1);
    double h3 = h * h * h;
    int hi[5] = {i - 2, i - 1, i, i + 1, i + 2};
    double hx[5];
    int j;

    for (j = 0; j < 5; j++)
        hx[j] = a + h * hi[j];

    if ((x < hx[0]) || (x > hx[4]))
        return 0;
    else if (x >= hx[0] && x <= hx[1])
        return 3 / h3 * (x - hx[0]) * (x - hx[0]);
    else if (x > hx[1] && x <= hx[2])
        return 1 / h3 * (3 * h * h + 6 * h * (x - hx[1]) - 9 * (x - hx[1]) * (x - hx[1]));
    else if (x > hx[2] && x <= hx[3])
        return 1 / h3 * (-3 * h * h - 6 * h * (hx[3] - x) + 9 * (hx[3] - x) * (hx[3] - x));
    else /* if (x > hx[3]) && (x <= hx[4]) */
        return -3 / h3 * (hx[4] - x) * (hx[4] - x);
}
```

```
double
d2fi(double a, double b, int n, int i, double x)
{
    double h = (b - a) / (n - 1);
    double h3 = h * h * h;
    int hi [5] = {i - 2, i - 1, i, i + 1, i + 2};
    double hx [5];
    int j;

    for (j = 0; j < 5; j++)
        hx[j] = a + h * hi[j];

    if ((x < hx[0]) || (x > hx[4]))
        return 0;
    else if (x >= hx[0] && x <= hx[1])
        return 6 / h3 * (x - hx[0]);
    else if (x > hx[1] && x <= hx[2])
        return 1 / h3 * (6 * h - 18 * (x - hx[1]));
    else if (x > hx[2] && x <= hx[3])
        return 1 / h3 * (6 * h - 18 * (hx[3] - x));
    else /* if (x > hx[3] && (x <= hx[4]) */
        return 6 / h3 * (hx[4] - x);
}
```

```
double
d3fi(double a, double b, int n, int i, double x)
{
    double h = (b - a) / (n - 1);
    double h3 = h * h * h;
    int hi [5] = {i - 2, i - 1, i, i + 1, i + 2};
    double hx [5];
    int j;

    for (j = 0; j < 5; j++)
        hx[j] = a + h * hi[j];

    if ((x < hx[0]) || (x > hx[4]))
        return 0;
    else if (x >= hx[0] && x <= hx[1])
        return 6 / h3;
    else if (x > hx[1] && x <= hx[2])
        return -18 / h3;
    else if (x > hx[2] && x <= hx[3])
        return 18 / h3;
    else /* if (x > hx[3] && (x <= hx[4]) */
        return -6 / h3;
}
```

To są funkcje, które liczą kolejne pochodne

A to układ równań podany wyżej:

```
for (j = 0; j < nb; j++) {
    for (i = 0; i < 5; i++)
        for (k = 0; k < pts->n; k++) {
            double temp = 1.0;
            for (int r = i+j; r > 0; r--)
                temp *= x[k];
            add_to_entry_matrix(eqs, j, i, temp);
        }
    for (k = 0; k < pts->n; k++) {
        double temp = 1.0;
        for (int r = j; r > 0; r--)
            temp *= x[k];
        add_to_entry_matrix(eqs, j, 5, y[k] * temp);
    }
}
```

Jak widać mamy tu mnożenie macierzy przez nasze  $w(x_i)$ , które przyjmuje się 1 oraz cały przekształcony układ. Jest to kluczowy fragment kodu.

A zapisywanie wartości współczynników, które są w macierzy eqs w ostatniej kolumnie do struktury spl, a dokładniej do tablicy f wewnątrz tej struktury wygląda następująco:

```
if (alloc_spl(spl, 5) == 0) {
    spl->f[0] = get_entry_matrix(eqs, 0, 5); //x to a0
    spl->f[1] = get_entry_matrix(eqs, 1, 5); // f to a1
    spl->f[2] = get_entry_matrix(eqs, 2, 5); // f1 to a2
    spl->f[3] = get_entry_matrix(eqs, 3, 5); // f2 to a3
    spl->f[4] = get_entry_matrix(eqs, 4, 5); // f3 to a4
}
```

## 2. Wywołanie programu

Dla gotowych danych testowych dane.1 efekt jest następujący:

```
micha@micha-VirtualBox:~$ cd pulpit
bash: cd: pulpit: No such file or directory
micha@micha-VirtualBox:~$ cd Pulpit
micha@micha-VirtualBox:~/Pulpit$ cd lmp10
micha@micha-VirtualBox:~/Pulpit/lmp10$ ls
aproxymator_na_bazie.c   gaus      makespl.h   points.h
aproxymator_na_bazie.o  interpolator.c  myplot      points.o
aprox                   main.c       myplot1     prosta.c
debug_base_plot.txt     main.o       myplot2     spl
debug_spline_plot.txt   Makefile     points.c     spl1
```

Następnie:

```
micha@micha-VirtualBox:~/Pulpit/lmp10$ ./aprox
Usage: ./aprox -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
micha@micha-VirtualBox:~/Pulpit/lmp10$ ./aprox -s spl -p test/dane.1 -g myplot -f 5.1 -t 5.7 -n 300
Przed petlaPrzejscie petli j = 0
Przejscie petli j = 1
Przejscie petli j = 2
Przejscie petli j = 3
Przejscie petli j = 4
Przejscie petli j = 5
Przejscie petli j = 6
Przejscie petli j = 7
Przejscie petli j = 8
Przejscie petli j = 9
5 6
30.00000 165.00000 910.17241 5035.34484 27937.34610 297105.12000
165.00000 910.17241 5035.34484 27937.34610 155443.30857 1652964.86482
910.17241 5035.34484 27937.34610 155443.30857 867294.84116 9221901.43678
5035.34484 27937.34610 155443.30857 867294.84116 4852247.12409 51588629.78084
27937.34610 155443.30857 867294.84116 4852247.12409 27218933.47285 289357385.80064
5 6
27937.34610 155443.30857 867294.84116 4852247.12409 27218933.47285 -17081726.22306
0.00000 -79.29724 -875.35486 -7259.86514 -53613.14238 12635973.90789
-0.00000 0.00000 0.18856 3.11596 34.37015 -3500805.84238
0.00000 0.00000 0.00000 -0.00043 -0.00941 430515.79659
0.00000 0.00000 0.00000 0.00000 0.00000 -19817.39387
```

Jak widać nasza macierz jest trójkątna, a po jednej stronie są same zera więc jest idealnie po naszej myśli.

Otwierając gnuplota dostaniemy taki wynik:

```
micha@micha-VirtualBox:~/Pulpit/lmp10$ gnuplot

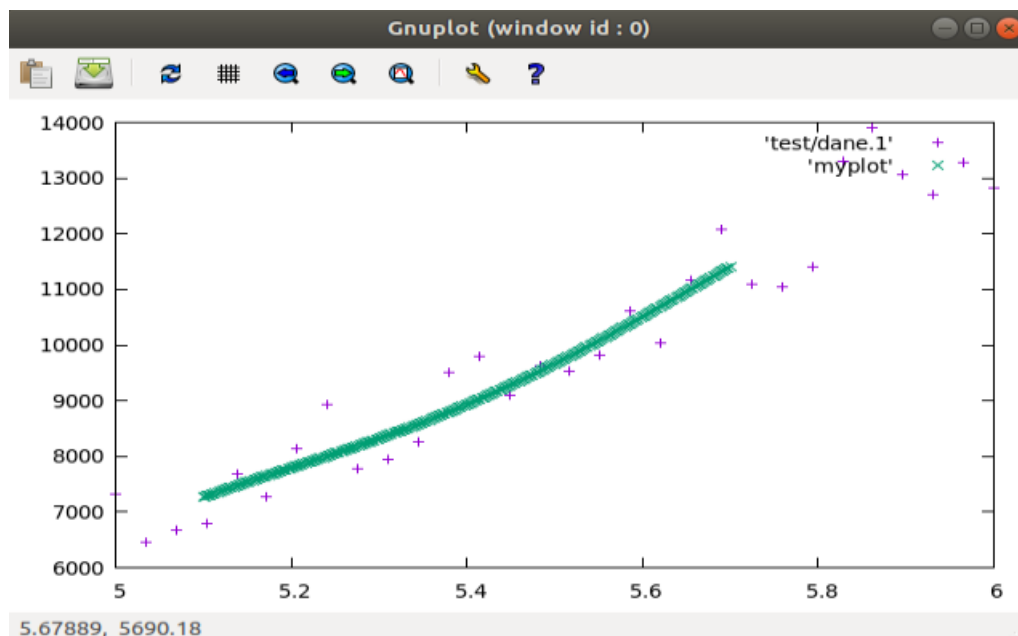
G N U P L O T
Version 5.2 patchlevel 2    last modified 2017-11-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2017
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 'wxt'
gnuplot> plot 'test/dane.1', 'myplot'
Gtk-Message: 23:49:09.405: Failed to load module "canberra-gtk-module"
gnuplot>          load/eval nested too deeply

gnuplot> plot 'test/dane.1', 'myplot'
```



Co znaczy, że dla 300 przedziałów z zakresu od 5.1 do 5.7 nasz aproksymator przepięknie narysował nam wykres, który jest bardzo dobrze dopasowany.

A dla innych danych:

```

micha@micha-VirtualBox:~/Pulpit/lmp10$ ./aprox -s spl -p test/dane.1 -g moj_przepiekny_plot_2 -f 5.0 -t 7.4 -n 300
Przed petlaPrzejscie petli j = 0
Przejscie petli j = 1
Przejscie petli j = 2
Przejscie petli j = 3
Przejscie petli j = 4
Przejscie petli j = 5
Przejscie petli j = 6
Przejscie petli j = 7
Przejscie petli j = 8
Przejscie petli j = 9
5 6
30.00000 165.00000 910.17241 5035.34484 27937.34610 297105.12000
165.00000 910.17241 5035.34484 27937.34610 155443.30857 1652964.86482
910.17241 5035.34484 27937.34610 155443.30857 867294.84116 9221901.43678
5035.34484 27937.34610 155443.30857 867294.84116 4852247.12409 51588629.78084
27937.34610 155443.30857 867294.84116 4852247.12409 27218933.47285 289357385.80064
5 6
27937.34610 155443.30857 867294.84116 4852247.12409 27218933.47285 -17081726.22306
0.00000 -79.29724 -875.35486 -7259.86514 -53613.14238 12635973.90789
-0.00000 0.00000 0.18856 3.11596 34.37015 -3500805.84238
0.00000 0.00000 0.00000 -0.00043 -0.00941 430515.79659
0.00000 0.00000 0.00000 0.00000 0.00000 -19817.39387
micha@micha-VirtualBox:~/Pulpit/lmp10$ gnuplot

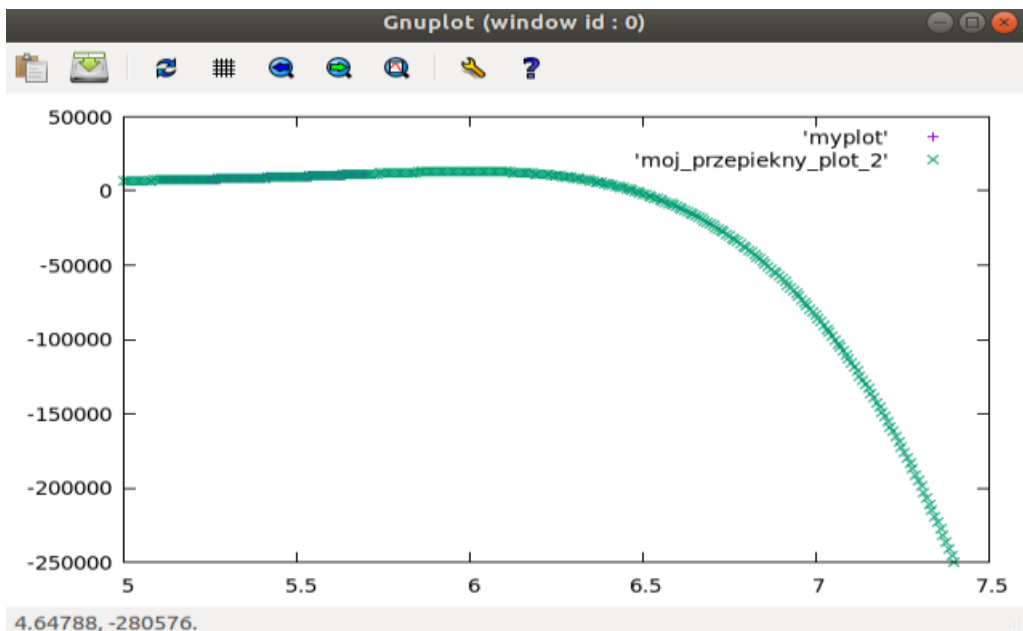
G N U P L O T
Version 5.2 patchlevel 2    last modified 2017-11-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2017
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')

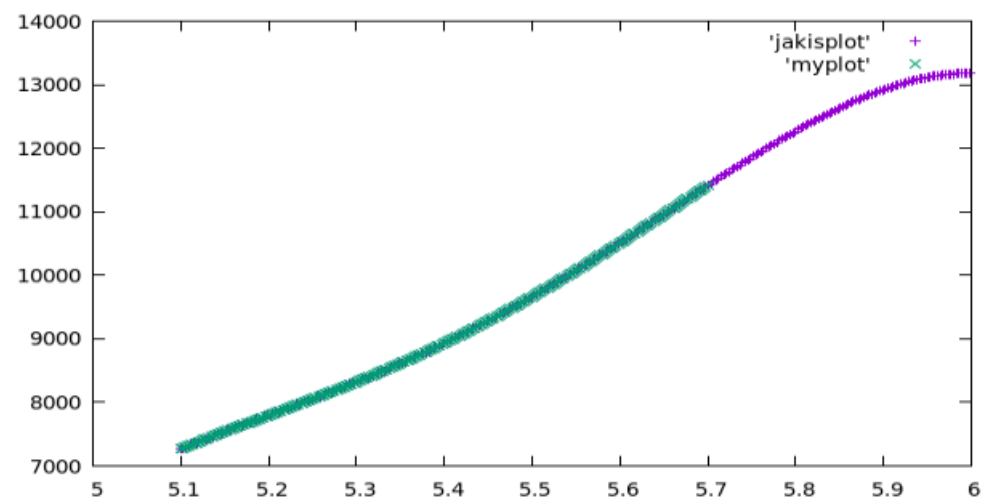
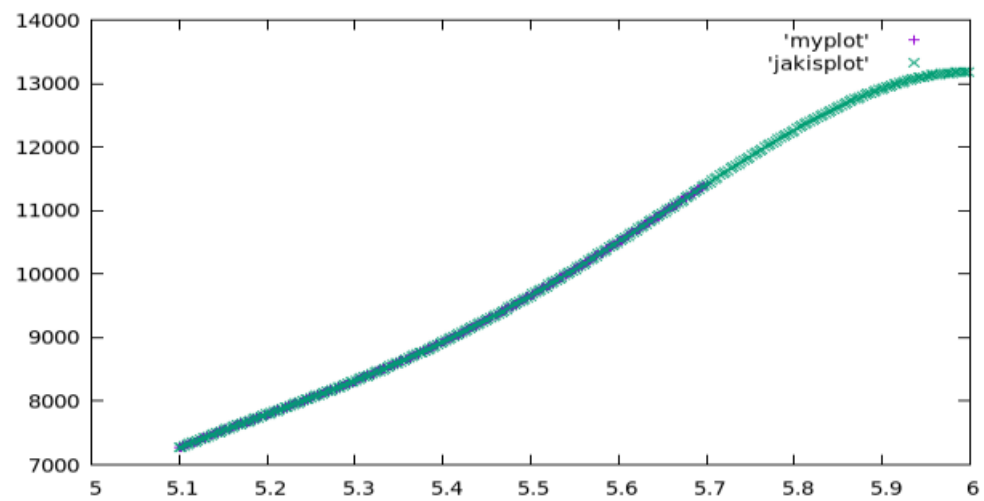
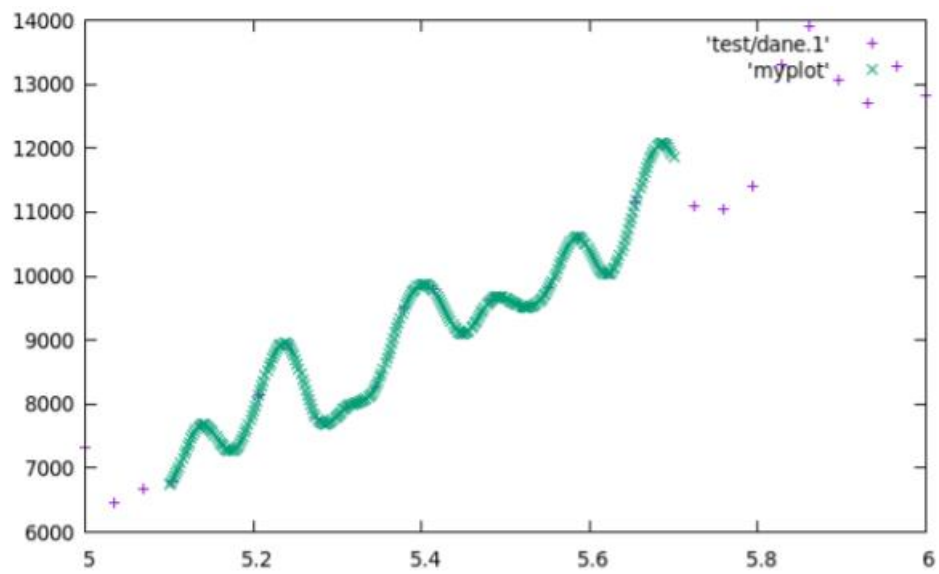
Terminal type is now 'wxt'
gnuplot> plot 'myplot', 'moj_przepiekny_plot_2'

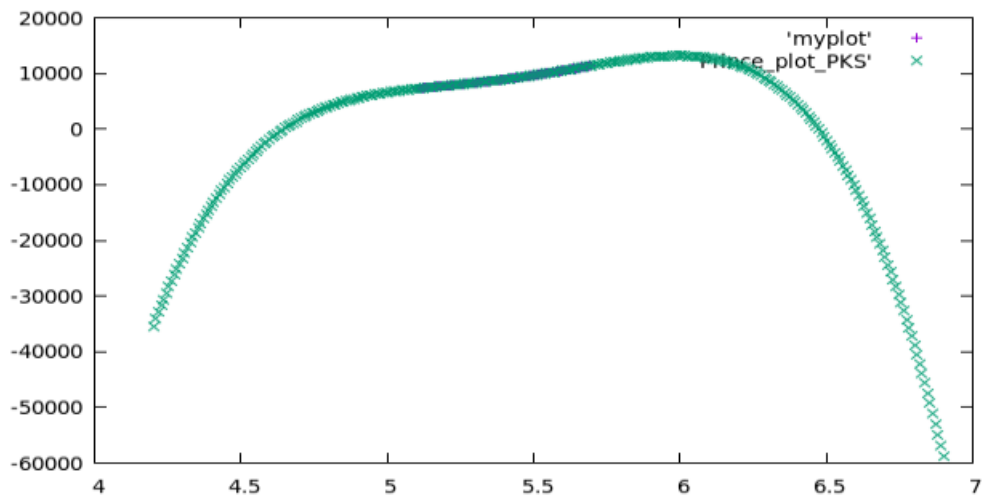
```



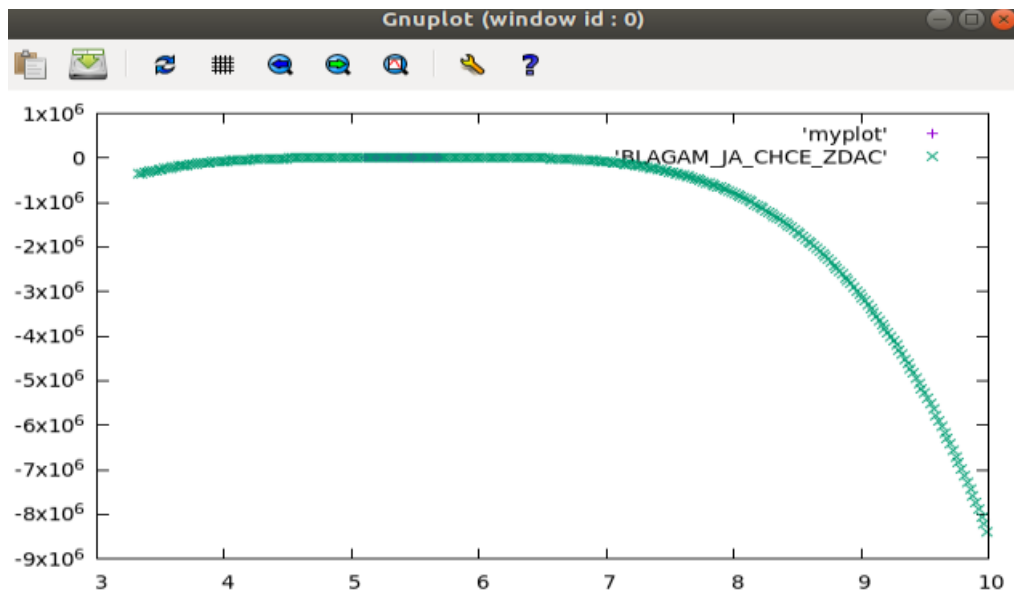
Nasz „przepiekny plot” rysuje jak widać bardzo dokładnie wykres.

Analogicznie możemy osiągnąć:

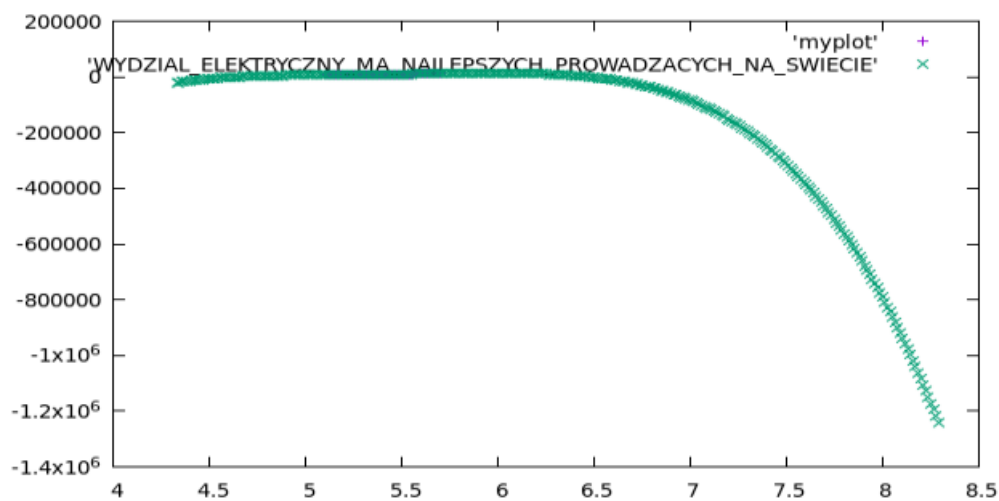




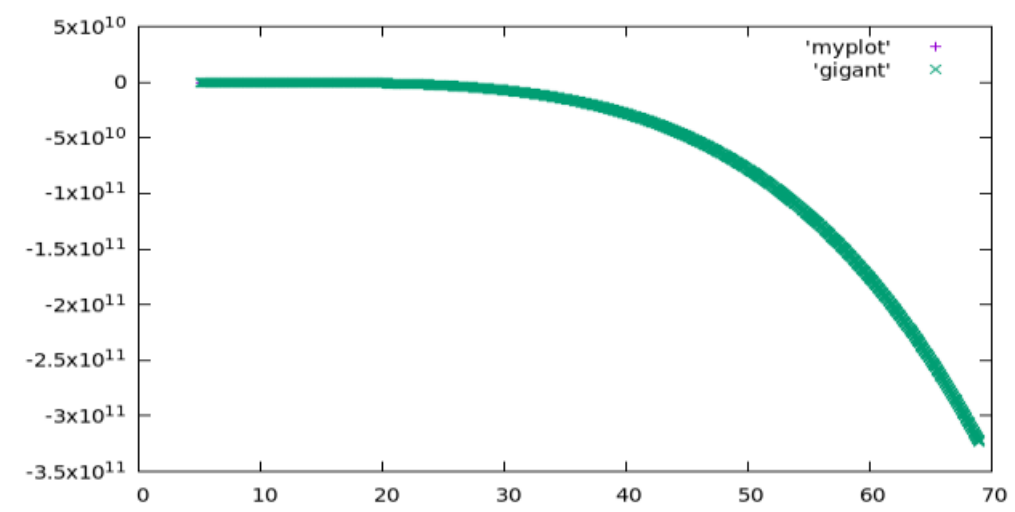
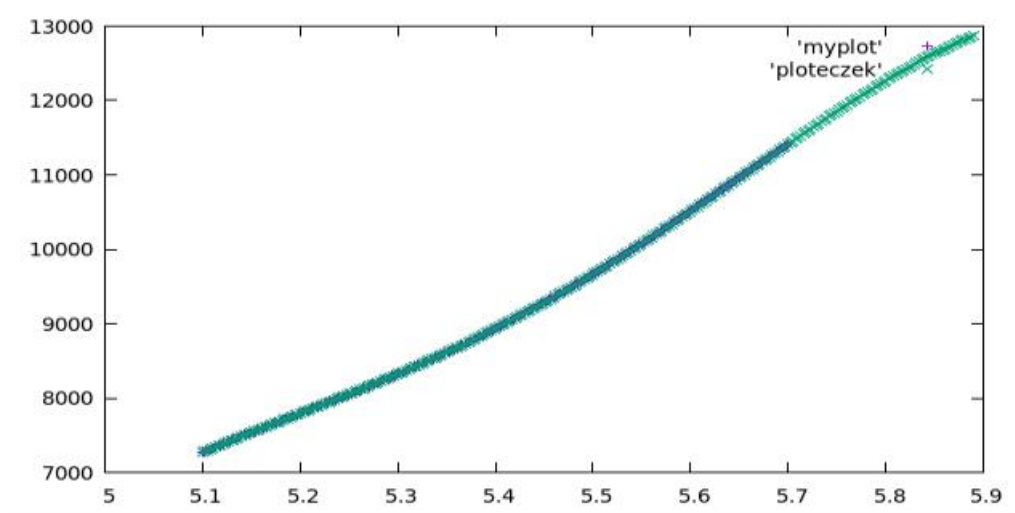
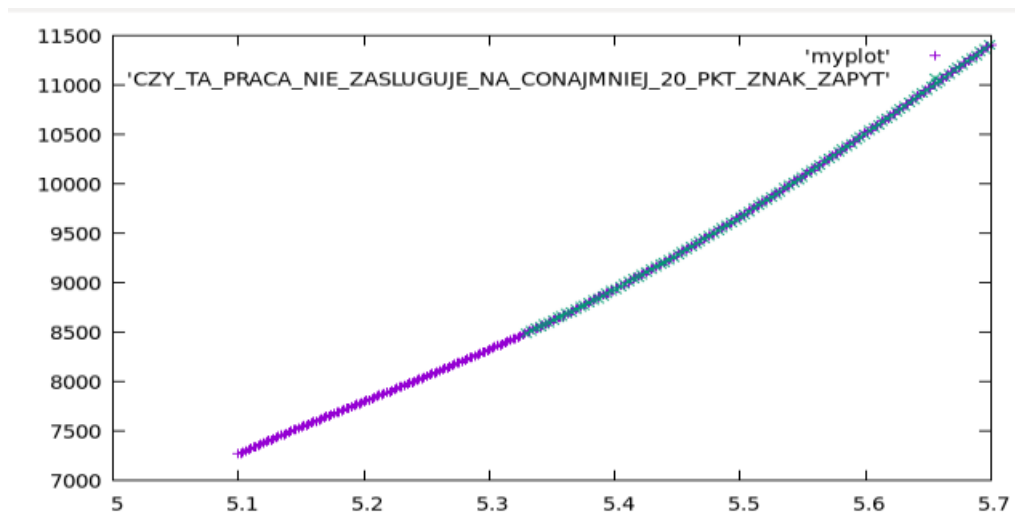
```
micha@micha-VirtualBox:~/Pulpit/lmp10$ ./aprox -s spl -p test/dane.1 -g BLAGAM_JA_CHCE_ZDAC -f 4.20 -t 6.9 -n 300
```



```
2.24391, -1.00192e+07
```



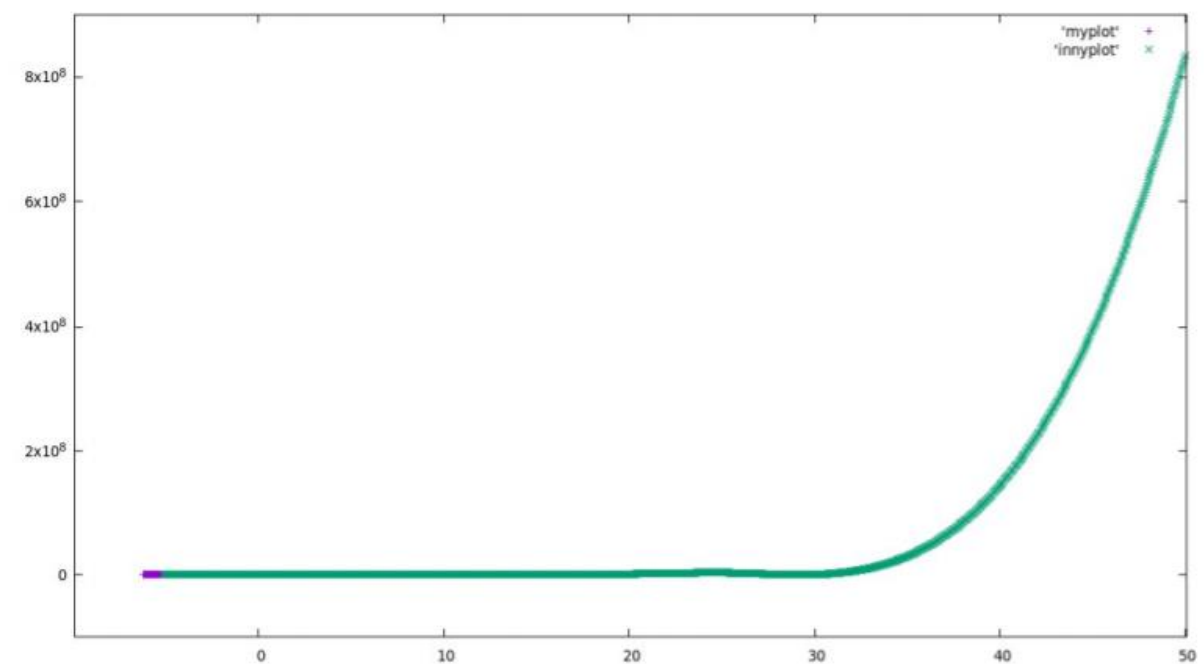
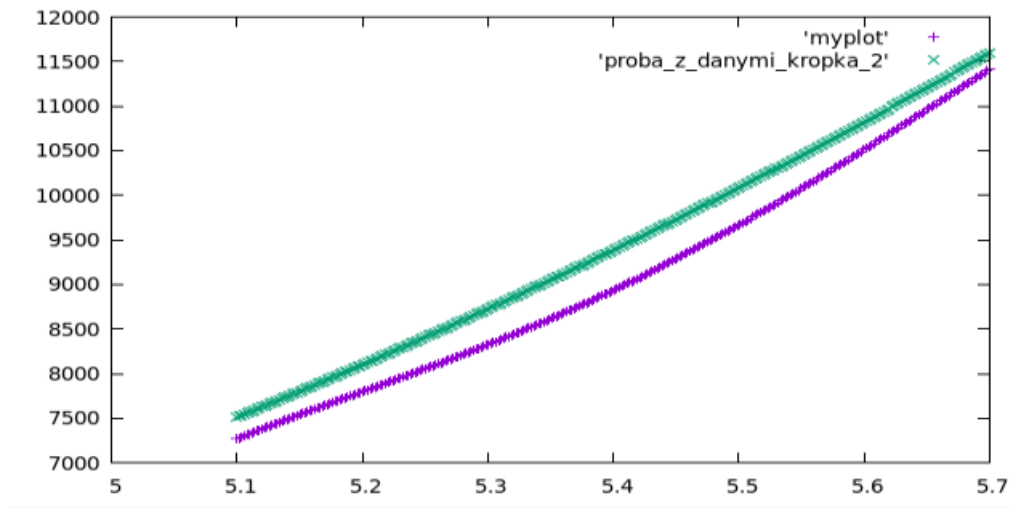




### 3. Testy dla różnych danych wejściowych

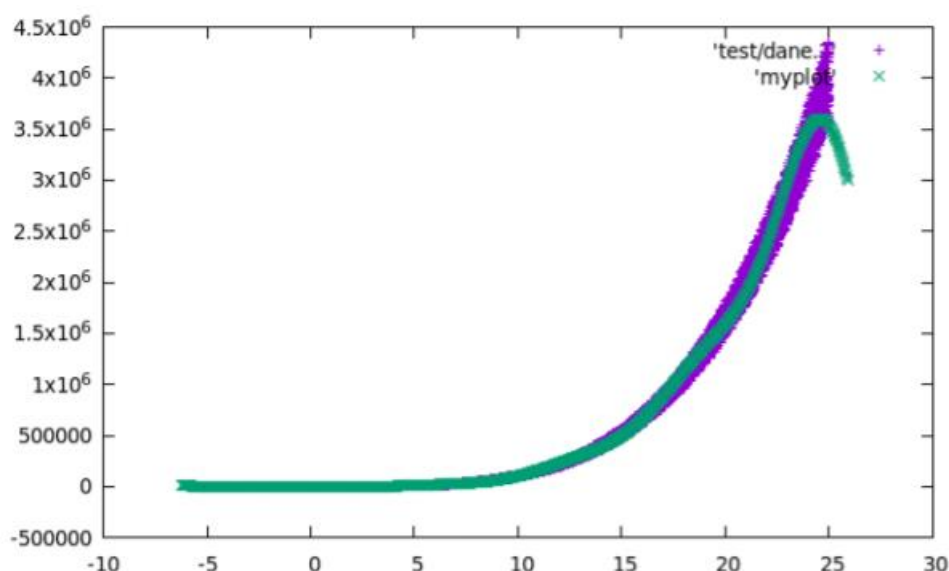
Przy użyciu programu gen w folderze test wygenerowałem dane testowe dane.2 i dane.3. Plik dane.3 zawiera milion punktów, więc nie zamieszczam żadnych wykresów (które zwyczajnie nie chciały się wygenerować)

```
micha@micha-VirtualBox:~/Pulpit/lmp10/test$ cd ..  
micha@micha-VirtualBox:~/Pulpit/lmp10$ ./aprox -s spl -p test/dane.2 -g proba_z_danymi_kropka_2 -f 5.1 -t 5.7 -n 300
```



## 4. Porównanie działania z domyślną metodą (funkcje bazowe dr. Chwieja)

Tak wygląda porównanie wraz z istniejącą aproksymacją:



Działanie programu jest bardzo podobne, myślę że jest również odrobinę szybsze. Jednakże uważam, że w moim przypadku jest bardziej dokładne, ponieważ nawet dla dużych zestawów liczb wykres bardzo ładnie się dopasowuje, jak i dla małych, a w przypadku gdy dana aproksymacja ma dany średni zakres (w porównaniu z danymi, które były testowane) nie jest ona idealna. Czas działania jest również porównywalny.

## 5. Wnioski

Aproksymacja średniokwadratowa z bazą wielomianu 4. Stopnia jest dobrą aproksymacją, na pewno dokładniejszą niż 2. Stopnia. Jednakże jest to podobna metoda aproksymacji, do gotowych. W akceptowalnym czasie pozwala oszacować najlepsze dopasowane punkty, które po zagęszczeniu (w dużej ilości) ładnie układają się w wykres dopasowany do naszych punktów. Program nie doświadczał żadnych problemów i działał w podobnym czasie co początkowy.

Zdecydowanie jestem zadowolony, że udało mi się zrobić tak duży projekt samemu, pomimo tego, iż od początku zajęć nie byłem czołowym programistą. Jednakże pisanie programu, zwłaszcza tak dużego, sprawiło mi wielką przyjemność, a satysfakcja z tego że działa jest ogromna.