Paulina Grabowska [redacted]	Metody numeryczne
Aproksymacja – metoda najmniejszych kwadratów	

1. Wstęp

Aproksymacja to technika matematyczna polegająca na konstrukcji rozwiązań przybliżonych, zwłaszcza gdy dokładnego rozwiązania nie można dokładnie przedstawić w formie analitycznej. Jest to szczególnie przydatne podczas analizowania danych, które są zbyt złożone, aby można je było dokładnie opisać za pomocą prostej funkcji lub wzoru.

Istnieje wiele metod aproksymacji, z których każda może być mniej lub bardziej odpowiednia w zależności od rodzaju danych i potrzeb analizy. Jedną z częściej stosowanych technik jest aproksymacja wielomianowa, która polega na szukaniu wielomianu o najmniejszym błędzie w dopasowaniu danych. Inną popularną metodą jest aproksymacja za pomocą funkcji sklejanych, podczas której segmenty cech są łączone w celu uzyskania płynnych przejść między nimi.

Celem aproksymacji jest znalezienie krzywych, które będą przebiegały jak najbliżej punktów testowych. Aproksymacji można używać do obliczania całek oznaczonych lub równań różniczkowych.

2. Metoda najmniejszych kwadratów

Metoda najmniejszych kwadratów jest jedną z najpopularniejszych technik stosowanych w aproksymacji danych. Jej głównym celem jest znalezienie takiej funkcji lub modelu, który minimalizuje sumę kwadratów różnic pomiędzy wartościami rzeczywistymi a wartościami przewidywanymi przez model.

Proces ten polega na dopasowaniu funkcji matematycznej do zestawu danych poprzez minimalizację sumy kwadratów błędów pomiędzy wartościami rzeczywistymi a wartościami obliczonymi przez funkcję aproksymacyjną. Szukana jest taka linia prosta, krzywa lub wielomian, dla których odległości od punktów danych są minimalne, przyjmując jako kryterium optymalizacji sumę kwadratów tych odległości.

Wyznaczane są odpowiednie parametry funkcji aproksymacyjnej poprzez rozwiązanie układu równań uzyskanego poprzez różniczkowanie sumy kwadratów błędów i wyzerowanie jej pochodnych względem parametrów modelu. Rozwiązanie tego układu równań daje optymalne wartości parametrów, które minimalizują sumę kwadratów błędów.

Tworzona jest więc funkcja dwóch zmiennych, *a* i *b*, która dla liczby punktów *n* przyjmuje postać:

$$S(a,b) = \sum_{i=1}^{n}[y_i - y(x_i)]^2 = \sum_{i=1}^{n}[y_i - b - ax_i]^2 \tag{1}$$

Należy znaleźć teraz takie wartości a i b, dla których funkcja S przyjmuje najmniejsze wartości. Dla funkcji wielu zmiennych takie minimum znajduje się w punkcie, w których pochodne cząstkowe tej funkcji po wszystkich zmiennych są równe zero, a więc:

$$\frac{\partial S(a,b)}{\partial a} = 0 \tag{2.1}$$

$$\frac{\partial S(a,b)}{\partial b} = 0 \tag{2.2}$$

Korzystając ze wzorów (1), (2.1) i (2.2), możemy wyliczyć zmienne *a* i *b*. Wzory na te zmienne bo wykonaniu wszystkich przekształceń będą wyglądać następująco:

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \tag{3.1}$$

$$b = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \tag{3.2}$$

Korzystając z tych danych można też wyliczyć współczynnik korelacji *r*, który jest miarą korelacji między zmiennymi x i y. Im ten wynik bliższy 1 lub -1, tym korelacja silniejsza (odpowiednio dodatnia i ujemna); jeśli jest równy 1 lub -1 oznacza to całkowitą korelację zmiennych; natomiast im wartość tego współczynnika bliższa zeru, tym wartość tej korelacji jest mniejsza.

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \tag{4}$$

3. Implementacja numeryczna

Zaczęto od implementacji funkcji *least_squares* (*Fragment kodu 1*), która implementuje metodę najmniejszych kwadratów, która jest używana do znalezienia najlepszego dopasowania linii trendu do zestawu danych. W szczególności, w tym przypadku, funkcja ta oblicza współczynniki a i b aproksymacyjnej funkcji liniowej o postaci $y = ax + b$, która najlepiej pasuje do danych wejściowych.

Na początku funkcja inicjuje zmienne pomocnicze *sum_x, sum_y, sum_xy* i *sum_x_squared* na wartość 0. Zmienne te odpowiadają wartościom sum znajdujących się we wzorami (3.1) i (3.2) i będą wykorzystywane do obliczenia współczynników a i b.

Następnie funkcja przechodzi przez wszystkie dane wejściowe, obliczając sumy wartości x, y, iloczynów $x \cdot y$ oraz kwadratów x^2. Te sumy są niezbędne do wyznaczenia współczynników a i b za pomocą metod najmniejszych kwadratów.

Po obliczeniu sum, funkcja wykorzystuje wzory analogiczne do (3.1) i (3.2), aby obliczyć współczynniki a i b. Współczynnik a jest obliczany na podstawie sumy iloczynów x i y, oraz sumy kwadratów x. Natomiast współczynnik b jest wyliczany na podstawie różnicy iloczynu sum y i kwadratów x oraz iloczynu sum x oraz xy; a następnie wszystko jest dzielone przez iloczyn n i sumy kwadratów x, od czego odejmujemy sumę x podniesioną do kwadratu.

Obliczany jest także współczynnik korelacji r, zgodnie ze wzorem (4); gdzie różnica iloczynów liczby punktów i sumy iloczynu zmiennych x i y oraz iloczynu sum x i y jest dzielona przez iloczynu dwóch pierwiastków: jednego z różnicy iloczynów liczby punktów n i sumy po x^2 oraz iloczynu dwóch sum po x, a drugiego z różnicy iloczynu liczby punktów n i sumy po zmiennych y^2 oraz iloczynu dwóch sum po y.

Warto zauważyć, że obliczenie współczynników a i b metodą najmniejszych kwadratów minimalizuje sumę kwadratów różnic pomiędzy wartościami rzeczywistymi y a wartościami obliczonymi przez aproksymacyjną funkcję liniową.

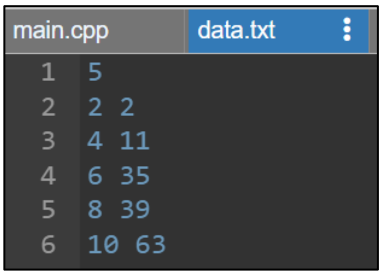
```
void least_squares(double x[], double y[], int n, double &a, double &b, double &r) {
    double sum_x = 0, sum_y = 0, sum_xy = 0, sum_x_squared = 0, sum_y_squared = 0;

    for (int i = 0; i < n; ++i) {
        sum_x += x[i];
        sum_y += y[i];
        sum_xy += x[i] * y[i];
        sum_x_squared += x[i] * x[i];
        sum_y_squared += y[i] * y[i];
    }

    a = (n * sum_xy - sum_x * sum_y) / (n * sum_x_squared - sum_x * sum_x);

    b = (sum_y * sum_x_squared - sum_x * sum_xy) / (n * sum_x_squared - sum_x * sum_x);

    r = ((n * sum_xy) - (sum_x * sum_y)) / (sqrt(n * sum_x_squared - (sum_x * sum_x)) *
       sqrt(n * sum_y_squared - (sum_y * sum_y)));
}

```

Fragment kodu 1: funkcja *least_squares*

Dane, na których działa funkcja, umieszczane są w pliku tekstowym, a funkcja główna zawiera funkcję do odczytania z pliku (*Fragment kodu 2*). Odbywa się to poprzez inicjalizację obiektu *ifstream*, a następnie sprawdzenie warunku otwarcia pliku, aby uniknąć możliwego błędu.

Struktura pliku tekstowego *data.txt* (*Rys. 1*) jest następująca: pierwsza linia zawiera tylko liczbę punktów n. W każdym kolejnym wierszu znajdują się dwie liczby: pierwsza to zmienna x dla danego punktu, a druga to zmienna y. Tak też odczytywany jest plik - najpierw odczytywana jest liczba n i na jej podstawie tworzone są tablice $x[n]$ i $y[n]$, które następnie za pomocą pętli uzupełniane są danymi z pliku tekstowego pliku.

Kiedy czytanie zostanie zakończone, plik tekstowy zostanie zamknięty, uwalniając zasoby systemowe i unikając ewentualnych problemów z plikiem przy późniejszym przetwarzaniu, ponieważ może on być później używany przez kilka aplikacji.



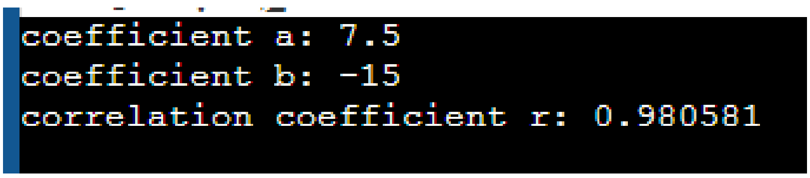
Rys. 1 Plik tekstowy *data.txt* zawierający zadane dane

Po odczytaniu wszystkich danych z pliku wywoływana jest dla nich funkcja *least_squares*, a następnie na konsoli wyświetlane są obliczone współczynniki *a* i *b*.

```
int main() {

    ifstream file("data.txt");

    if (!file.is_open()) {
        cout << "no such file exits" << endl;
        return 1;
    }


    int n;
    file >> n;

    double x[n];
    double y[n];


    for (int i = 0; i < n; ++i) {
        file >> x[i] >> y[i];
    }


    double a, b, r;

    least_squares(x, y, n, a, b, r);

    cout << "coefficient a: " << a << endl;
    cout << "coefficient b: " << b << endl;

    cout << "correlation coefficient r: " << r << endl;


    file.close();

    return 0;
}

```

Fragment kodu 2: funkcja *main*

Po zaimplementowaniu algorytmu sprawdzono poprawność jego działania dla zadanych wcześniej wartości (znajdujących się w pliku *data.txt – Rys.1*). Wynik zgadzał się (*Rys. 2*), co pozwala na stwierdzenie, że algorytm działa poprawnie. Następnie przystąpiono do napisania testów jednostkowych, które mają upewnić się, co do poprawnego działania algorytmu



Rys. 2 Zrzut ekranu konsoli programu dla danych testowych

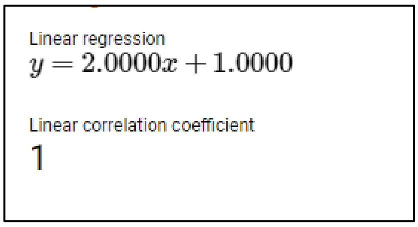
4. Testy jednostkowe

4.1 Test dla danych liniowych

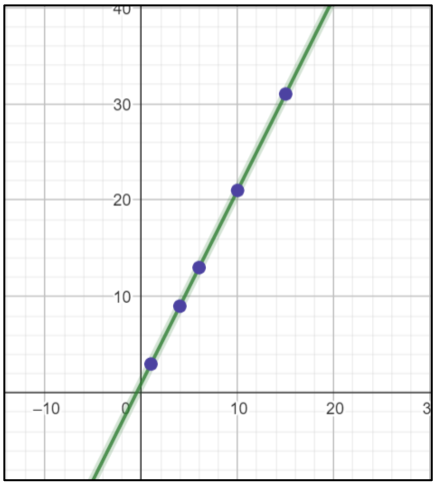
Do pliku tekstowego *data.txt* wprowadzono dane punktów, celem utworzenia przez nie funkcji liniowej:

$$f(x) = 2x + 1.$$

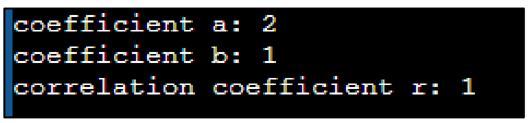
Wybrano punkty o współrzędnych odpowiednio (1, 3), (4, 9), (6, 13), (10, 21), (11, 31). Poprawność wyników sprawdzono w programie PlanetCalc (*Rys. 3a*), a wykresy wygenerowano w programie Geogebra (*Rys. 3b*). Wynik z konsoli załączono na *Rys. 3c*.



Rys. 3a Oczekiwane wartości współczynników *a, b* i wartości współczynnika korelacji *r* obliczone w programie PlanetCalc



Rys. 3b Wykres funkcji utworzonej dla testu jednostkowego dla danych liniowych wygenerowany w programie Geogebra



Rys. 3c Wydruk z konsoli po skompilowaniu programu dla testu jednostkowego dla danych liniowych

4.2 Test dla danych nieregularnych

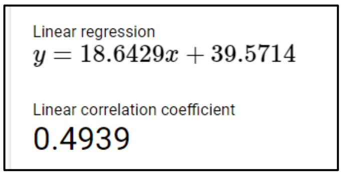
Wybrano punkty o współrzędnych odpowiednio (−3, 86), (−2, −33), (−1, −18), (0, −7), (1 ,1), (2, 49), (3, 199) które tworzą funkcję wielomianową (na *Rys. 4b* oznaczoną kolorem czerwonym) o wzorze :

$$f(x) = 2x^5 - 3x^4 + 6x^3 - x^2 + 4x - 7.$$

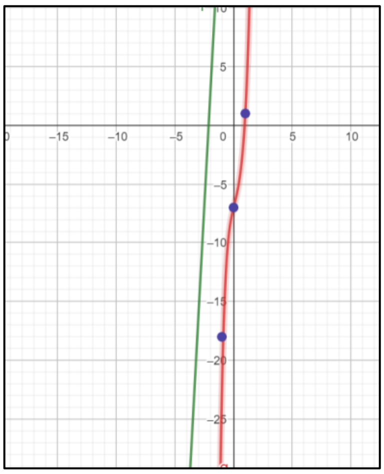
Do pliku tekstowego *data.txt* wprowadzono dane punktów, celem utworzenia przez nie funkcji liniowej (zaznaczonej na *Rys. 4b* na zielono):

$$f(x) = 18.6429\,x + 39.5714.$$

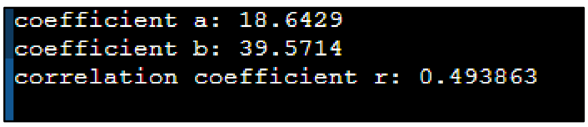
Poprawność wyników sprawdzono w programie PlanetCalc (*Rys. 4a*), a wynik z konsoli załączono na *Rys. 4c*.



Rys. 4a Oczekiwane wartości współczynników *a, b* i wartości współczynnika korelacji *r* obliczone w programie PlanetCalc



Rys. 4b Wykres funkcji utworzonej dla testu jednostkowego dla danych nieregularnych wygenerowany w programie Geogebra. Z przyczyn czytelności nie załączono całego wykresu



Rys. 4c Wydruk z konsoli po skompilowaniu programu dla testu jednostkowego dla danych nieregularnych

4.3 Test dla dużego zestawu danych

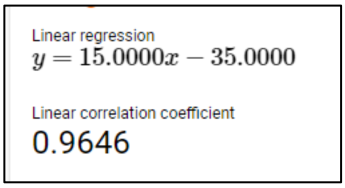
Do pliku tekstowego *data.txt* wprowadzono 16 punktów (od 0 do 15) tak, aby tworzyły funkcję kwadratową (zaznaczoną na *Rys. 5b* na czerwono):

$$f(x) = x^2$$

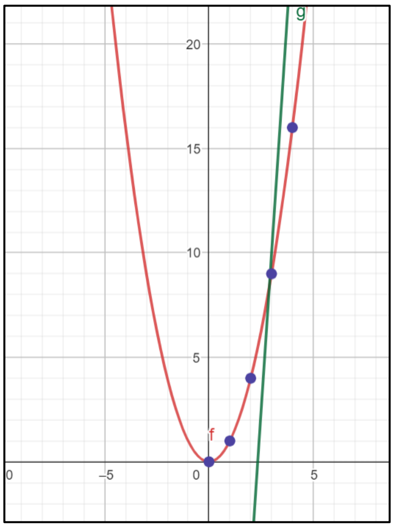
Spodziewaną prostą aproksymacji jest prosta o równaniu (zaznaczona na *Rys. 5b* na zielono):

$$f(x) = 15x - 35$$

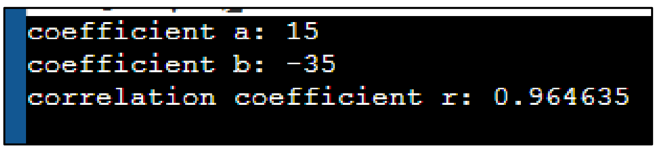
Poprawność wyników sprawdzono w programie PlanetCalc (*Rys. 5a*), a wynik z konsoli załączono na *Rys. 5c*



Rys. 5a Oczekiwane wartości współczynników *a, b* i wartości współczynnika korelacji *r* obliczone w programie PlanetCalc



Rys. 5b Wykres funkcji utworzonej dla testu jednostkowego dla dużej liczby danych wygenerowany w programie Geogebra. Z przyczyn czytelności nie załączono całego wykresu



Rys. 5c Wydruk z konsoli po skompilowaniu programu dla testu jednostkowego dla dużej liczby danych

4.4 Test dla danych niespełniających założeń metody

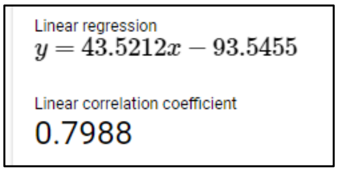
Do pliku tekstowego *data.txt* wprowadzono 10 punktów (od 0 do 9) tak, aby tworzyły funkcję eksponencjalną (zaznaczoną na *Rys. 6b* na czerwono), której punkty o wiele bardziej wyraźnie układają się na krzywej, a nie na prostej:

$$f(x) = 2^x$$

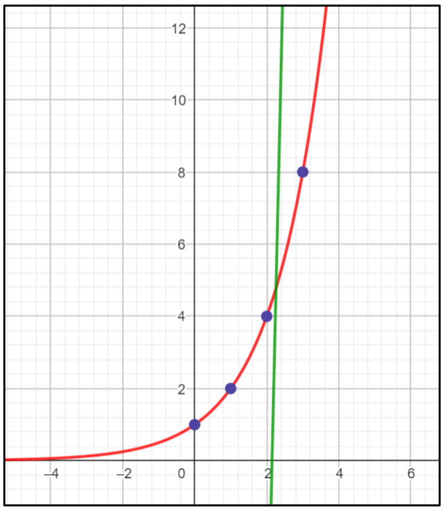
Spodziewaną prostą jest prosta o równaniu (zaznaczona na *Rys. 6b* na zielono):

$$f(x) = 43.5212x - 93.5455$$

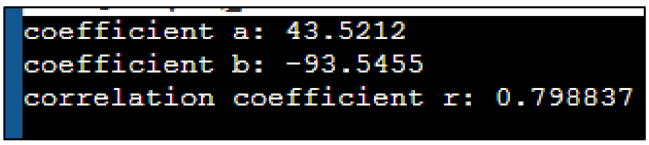
Poprawność wyników sprawdzono w programie PlanetCalc (*Rys. 6a*), a wynik z konsoli załączono na *Rys. 6c*



Rys. 6a Oczekiwane wartości współczynników *a, b* i wartości współczynnika korelacji *r* obliczone w programie PlanetCalc



Rys. 6b Wykres funkcji utworzonej dla testu jednostkowego dla danych niespełniających założeń metody wygenerowany w programie Geogebra. Z przyczyn czytelności nie załączono całego wykresu



Rys. 6c Wydruk z konsoli po skompilowaniu programu dla testu jednostkowego dla danych niespełniających założeń metody

5. Podsumowanie wyników

Otrzymane wartości są identyczne z otrzymanymi – algorytm działa więc bez zarzutu i zwraca poprawne wartości. Co jednak wymaga podkreślenia to to, że w zależności od funkcji zmienia się współczynnik korelacji. Im funkcja bardziej liniowa (funkcja liniowa, kwadratowa) tym współczynnik korelacji bardziej zbliżony jest do 1, co oznacza że wyliczona prosta aproksymacyjna jest całkiem, dobrze skorelowana z wykresem. Natomiast w przypadku funkcji łukowych – takich jak wielomianowe, eksponencjalne czy logarytmiczne należy się spodziewać mniejszego współczynnika korelacji, z racji zakrzywienia wykresów i prostej aproksymacji.

Tabela 1: Zestawienie otrzymanych wyników

Numer testu rodzaje funkcji	Oczekiwana wartość *a*	Otrzymana wartość *a*	Oczekiwana wartość *b*	Otrzymana wartość *b*	Oczekiwana wartość *r*	Otrzymana wartość *r*
1 f. liniowa	2	2	1	1	1	1
2 f. wielomianowa	18.6429	18.6429	39.5714	39.5714	0.4939	0.4939
3 f. kwadratowa	15	15	-35	-35	0.9646	0.9646
4 f. eksponencjalna	43.5212	43.5212	-93.544	-93.544	0.7988	0.7988

6. Wnioski

Metoda najmniejszych kwadratów jest skuteczną techniką do dopasowania modelu matematycznego do zestawu danych. Dzięki niej możliwe jest znalezienie najlepiej dopasowanej funkcji do danych, nawet jeśli nie są one idealnie liniowe. Warto zauważyć, że wyniki uzyskane za pomocą metody najmniejszych kwadratów powinny być interpretowane ostrożnie. Konieczne jest zrozumienie ograniczeń metody oraz kontekstu danych, aby poprawnie zinterpretować wyniki i wyciągnąć sensowne wnioski.

Metoda najmniejszych kwadratów może być wrażliwa na odstające punkty w danych. W takich przypadkach istnieją różne techniki, takie jak współczynnik korelacji, który pokazuje nam odchylenie prostej aproksymacji od rzeczywistych danych.

7. Źródła

Wykłady z Metod Numerycznych autorstwa dr hab. Danuty Szeligi

Prezentacja Metody Numeryczne. Aproksymacja – metoda najmniejszych kwadratów autorstwa dr. Hab. Inż. Marcina Hojnego.

Wikipedia – artykuły o aproksymacji, metodzie najmniejszych kwadratów

https://qcg.home.amu.edu.pl/pliki/Aproksymacja.pdf

Program Geogebra: https://www.geogebra.org/calculator

Program PlanetCalc: https://planetcalc.com