

Raport Zadanie NUM8

Tomasz Dziób

31.12.2023

1 Wstęp techniczny

Poniższy raport dotyczy zadania numerycznego NUM8 z listy 7. Załączone programy o nazwie *num8_a.cpp* i *num8_b.cpp* zostały napisane w języku *C++*, oraz korzystają z bibliotek *GNU Plot* oraz *GSL*. Przed skompilowaniem programu należy zainstalować powyższe biblioteki.

1.1 Jak uruchomić program?

Razem z załączonymi plikami znajdziemy *Makefile* który służy do uruchomienia programów.

- *make runA* uruchamia program dotyczący podpunktu a
- *make runB* uruchamia program dotyczący podpunktu b

2 Nakreślenie problemu

Aproksymacja jest to proces budowania przybliżonych rozwiązań używany w sytuacjach gdy nie da się go przedstawić dokładnie w postaci analitycznej. Termin aproksymacja występuje w dwóch znaczeniach:

- a) Aproksymacja punktowa Mając N punktów staramy się znaleźć funkcję która będzie przebiegać możliwie “najbliżej” tych punktów. Znamy “kształt” szukanej funkcji (np. stopień, kombinacja funkcji trygonometrycznych), ale jej parametry są nieznane.

$$ax^3 + bx^2 + cx + d$$

- b) Aproksymacja ciągła Znając ustaloną funkcję $g(x)$ której analityczny sposób obliczania jest trudny staramy się skonstruować inną funkcję która będzie bliska funkcji wyjściowej, a równocześnie prostsza obliczeniowo.

3 Użyta metoda

W tym zadaniu zostajemy poproszeni o znalezienie wcześniej podanych współczynników funkcji. Do osiągnięcia tego skorzystamy z *metody najmniejszych kwadratów*.

Metoda najmniejszych kwadratów – Jest to metoda służąca przybliżaniu rozwiązań układów nieokreślonych, czyli takich w których jest więcej równań niż niewiadomych.

Każdy z n punktów które posiadamy można potraktować jako jedno równanie. W zapisie macierzowym wygląda to tak:

$$\begin{bmatrix} (x_1)^3 & (x_1)^2 & x_1 & 1 \\ (x_2)^3 & (x_2)^2 & x_2 & 1 \\ (x_3)^3 & (x_3)^2 & x_3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ (x_n)^3 & (x_n)^2 & x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_n) \end{bmatrix}$$

Po rozwiązaniu powyższego układu równań otrzymujemy rozwiązanie w postaci szukanych współczynników z których możemy obliczyć przybliżoną funkcję.

4 Uzyskany wynik

- a) Wynik wykonania komendy *make runA*:

Współczynniki:
a = 0.100934
b = 4.02306
c = 3.08874
d = 5.63284

b) Wynik wykonania komendy *make runB*:

Ustalone parametry:

$a = 2.137$

$b = 0.69$

$c = 0.5$

$d = 0.42$

Uzyskane parametry:

$a = 2.49791$

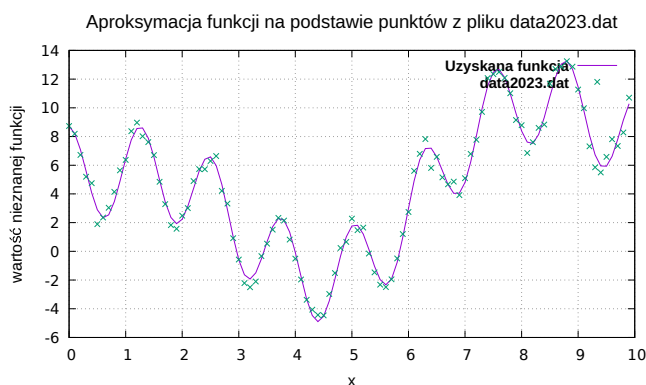
$b = 0.771317$

$c = 0.409573$

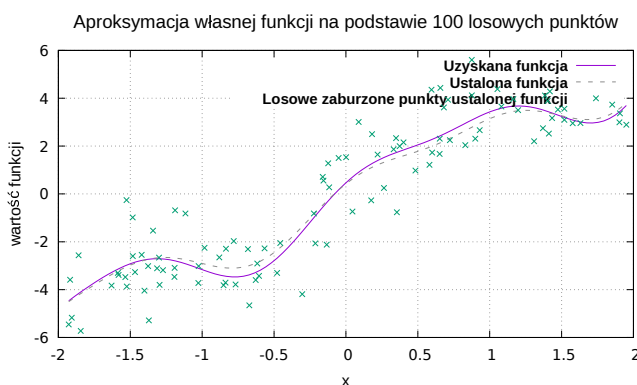
$d = 0.517033$

Analizując wyniki z podpunktu b) uruchamiając program kilka razy będziemy w stanie zauważyć minimalnie różniące się wyniki ze względu na losowość wybieranych punktów.

Wyniki na wykresie prezentują się następująco:



Rysunek 1: Wykres dla podpunktu a)



Rysunek 2: Wykres dla podpunktu b)

- a) Sposób rozwiązywania działa tak jak powinien prowadząc “pomiędzy” punktami przybliżony kształt przyjmowany przez szukaną funkcję.
- b) Dla losowych 100 punktów zakłóconych liczbami pochodzącymi z rozkładu normalnego Gaussa, uzyskujemy podobny wynik. W tym przypadku jesteśmy w stanie zauważyć jak bardzo uzyskane przybliżenie odstaje od wybranej funkcji tzn. zobaczyć wizualnie wartości błędu pomiarowego spowodowanego wygenerowanymi zakłóceniami.

5 Podsumowanie

Aproksymacja metodą najmniejszych kwadratów przydaje się najbardziej gdy potrzebujemy przedstawić funkcję której obliczenie jest wymagające analitycznie lub nie znamy jej parametrów. Używanie jej dla aproksymacji punktowej sprowadza się do rozwiązania układu równań nieokreślonych. Pozwala to na znalezienie zadowalającego nas przybliżenia co zaowocuje w krótszym czasie wykonania.

Choć metoda najmniejszych kwadratów jest potężnym narzędziem, ma też swoje ograniczenia. Na przykład, może nie być odpowiednia, gdy mamy do czynienia z danymi, które są źle uwarunkowane lub mają duże błędy pomiarowe. W takich przypadkach mogą być potrzebne inne techniki numeryczne.