

Sprawozdanie

Metody Numeryczne 2, laboratorium 4

Grzegorz Rozdzialik (D4, grupa lab. 2)

5 grudnia 2016

1 Zadanie

Temat **4**, zadanie **46**:

Wzory empiryczne. Baza: $1, x, x^2, \sin x$. Graficzne przedstawienie punktów pomiarowych i funkcji przybliżającej.

Niech $f : \mathbb{R} \rightarrow \mathbb{R}$ będzie funkcją aproksymowaną, $m \in \mathbb{N}$, $x_1, x_2, \dots, x_m \in \mathbb{R}$ oraz $f_1, f_2, \dots, f_m \in \mathbb{R}$ będą wartościami funkcji aproksymowanej w punktach x_i ($f_i = f(x_i)$, $i = 1, 2, \dots, m$).

Niech $g_1, g_2, \dots, g_n : \mathbb{R} \rightarrow \mathbb{R}$ będą funkcjami z bazy.

Należy znaleźć element optymalny

$$f^* = \alpha_1 g_1(x) + \alpha_2 g_2(x) + \dots + \alpha_n g_n(x) \quad (1)$$

, który minimalizuje wyrażenie

$$H(\alpha_1, \alpha_2, \dots, \alpha_n) = \sum_{i=1}^m \left(f_i - f^*(x_i) \right)^2 \quad (2)$$

Zakładamy $m \geq n$.

2 Opis metody

Po obliczeniu pochodnej funkcji H (równanie (2)) ze względu na dowolną zmienną α_i i przyrównaniu jej do zera (ponieważ szukamy ekstremum) otrzymujemy następującą równość:

$$\forall_{k=1,\dots,n} \sum_{j=1}^n \alpha_j \sum_{i=1}^m g_j(x_i) g_k(x_i) = \sum_{i=1}^m f_i g_k(x_i) \quad (3)$$

Zauważmy, że $\sum_{i=1}^m g_j(x_i) g_k(x_i) = \langle g_j, g_k \rangle$ oraz $\sum_{i=1}^m f_i g_k(x_i) = \langle f_i, g_k \rangle$. Zatem równość (3) można zapisać jako:

$$\forall_{k=1,\dots,n} \sum_{j=1}^n \alpha_j \langle g_j, g_k \rangle = \langle f_i, g_k \rangle \quad (4)$$

Otrzymaliśmy więc układ równań normalnych $G\alpha = F$, gdzie

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$
$$G = \begin{bmatrix} \langle g_1, g_1 \rangle & \langle g_1, g_2 \rangle & \dots & \langle g_1, g_n \rangle \\ \langle g_2, g_1 \rangle & \langle g_2, g_2 \rangle & \dots & \langle g_2, g_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle g_n, g_1 \rangle & \langle g_n, g_2 \rangle & \dots & \langle g_n, g_n \rangle \end{bmatrix}$$
$$F = \begin{bmatrix} \langle f_1, g_1 \rangle \\ \langle f_2, g_2 \rangle \\ \vdots \\ \langle f_n, g_n \rangle \end{bmatrix}$$

Po jego rozwiązaniu otrzymujemy współczynniki α_i , a mamy całą postać elementu optymalnego f^* .

W przypadku naszego zadania mamy:

$$\begin{aligned}g_1(x) &= 1 \\g_2(x) &= x \\g_3(x) &= x^2 \\g_4(x) &= \sin x\end{aligned}$$

Ilość punktów pomiarowych n nie może być mniejsza niż 4 ($n \geq 4$). Wtedy element optymalny f^* ma postać:

$$f^*(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 \sin x$$

Po obliczeniu iloczynów skalarnych $\langle g_i, g_j \rangle$ (dla $i, j = 1, 2, 3, 4$) i wstawieniu do macierzy G otrzymujemy:

$$G = \sum_{k=1}^m \begin{bmatrix} m & x_k & x_k^2 & \sin x_k \\ x_k & x_k^2 & x_k^3 & x_k \sin x_k \\ x_k^2 & x_k^3 & x_k^5 x_k^2 & \sin x_k \\ \sin x_k & x_k \sin x_k & x_k^2 \sin x_k & \sin^2 x_k \end{bmatrix}$$

Po obliczeniu iloczynów skalarnych $\langle f_i, g_i \rangle$ (dla $i = 1, 2, 3, 4$) i wstawieniu do wektora F otrzymujemy:

$$F = \sum_{k=1}^m \begin{bmatrix} f_k \\ f_k x_k \\ f_k x_k^2 \\ f_k \sin x_k \end{bmatrix}$$

Zatem należy rozwiązać układ równań $G\alpha = F$, gdzie

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}$$

Element optymalny f^* otrzymujemy wstawiając znalezione współczynniki $\alpha_1, \dots, \alpha_4$ do wzoru (1).

3 Implementacja metody

Implementacja metody podzielona jest na następujące kroki:

1. Obliczenie macierzy G oraz wektora F na podstawie podanych punktów pomiarowych (funkcja `computeApproximationEquation`).
2. Obliczenie współczynników $\alpha_1, \dots, \alpha_4$ (funkcja `computeApproximationCoefficients`).
3. Uzyskanie elementu optymalnego (funkcja `composeApproximationFunction`).

4 Poprawność metody

Dla przykładowych punktów pomiarowych testowanych przeze mnie metoda była poprawna, to jest znajdowała funkcję, która aproksymowała te punkty. Zdarzało się jednak, że aproksymacja ta nie była zbyt dokładna.

Zatem ogólnie można stwierdzić, że metoda jest poprawna. W swoich obliczeniach w sekcji 2 nie stosowałem żadnych restrykcyjnych założeń, zatem powinna być wykonalna dla wszystkich zestawów punktów pomiarowych, z zastrzeżeniem, że dla niektórych może nie być dokładna.

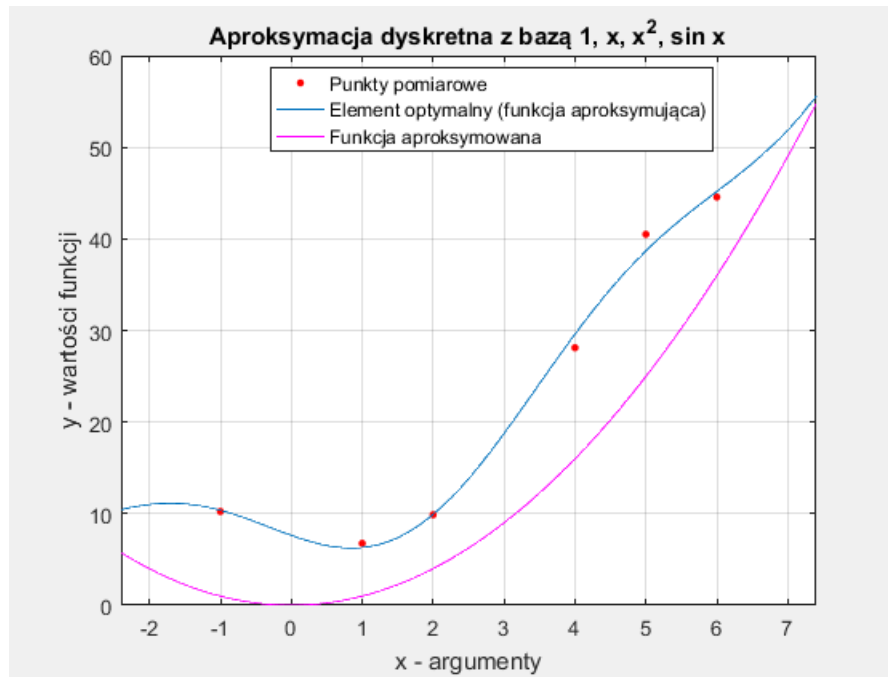
5 Przykłady

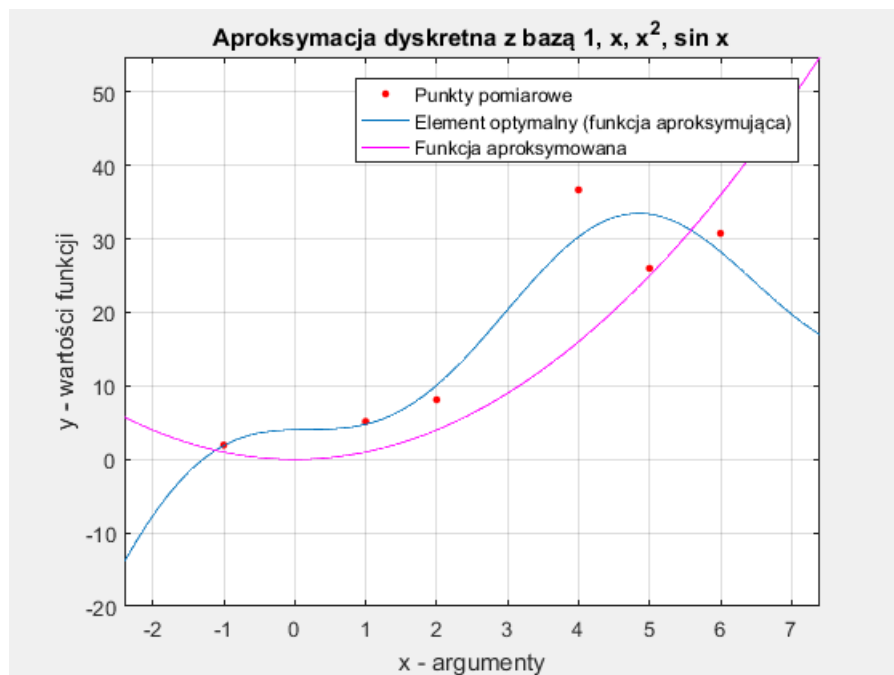
Przykłady zostały wygenerowane poprzez obliczenie danej funkcji aproksymowanej f w węzłach, a następnie zaburzenie tych wartości poprzez dodanie szumu. Dopiero na tych zaburzonych wartościach wykonujemy aproksymację.

Przykład 1 Funkcja aproksymowana: $f(x) = x^2$.
Zaburzenie wartości:

- mnożnik z zakresu $< \frac{1}{2}, 2 >$
- dodanie stałej z zakresu $< 0, 10 >$

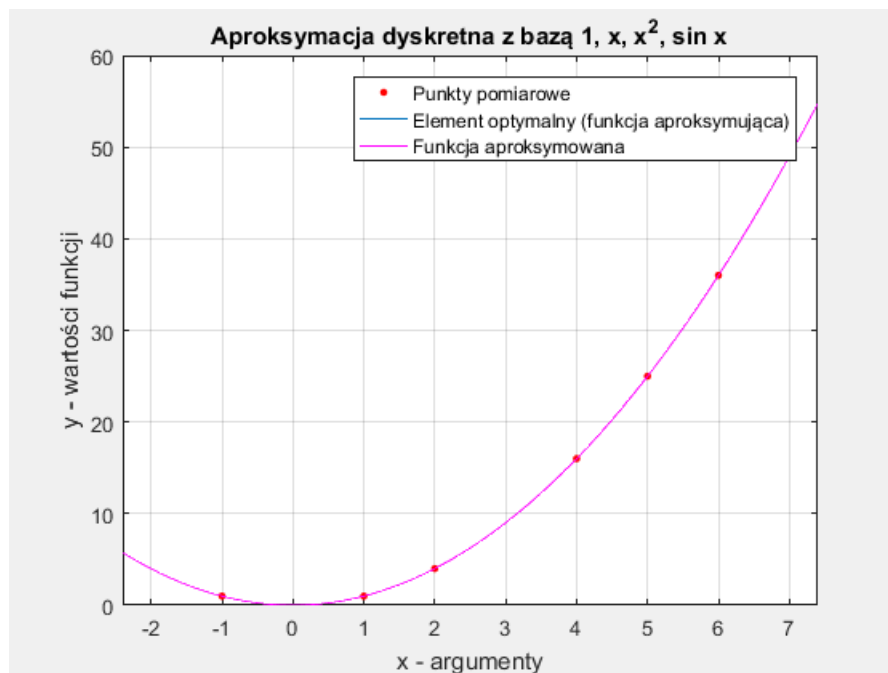
Węzły: $x \in \{-1, 1, 2, 4, 5, 6\}$





Dokładność aproksymacji zależy głównie od wylosowanego szumu. Element optymalny dość dobrze aproksymuje węzły, jednakże ze względu na duże zaburzenie wartości w węzłach nie aproksymuje on dobrze funkcji wejściowej $f(x) = x^2$.

Przykład 2 Funkcja aproksymowana: $f(x) = x^2$.
 Brak zaburzenia wartości.
 Węzły: $x \in \{-1, 1, 2, 4, 5, 6\}$



Metoda dokładnie aproksymuje funkcję z bazy przestrzeni elementu optymalnego, jeżeli dane nie są zaburzone.

6 Wnioski

1. Element optymalny w tym przypadku odwzorowuje dokładnie funkcję z bazy (gdy dane nie są zaburzone).
2. Dla 4 punktów pomiarowych aproksymacja jest dokładna (staje się interpolacją). Dla większej ilości punktów nie ma takiej zależności.
3. Im więcej punktów pomiarowych i im większe moduły zmian między nimi, tym aproksymacja jest gorsza.

7 Funkcja do testowania metody

Funkcja `plotApproximation(x, f, pointsCount, paddingMultiplier)` umożliwia łatwe testowanie metody. Oblicza ona element optymalny, a następnie wyświetla wykres elementu optymalnego oraz nanosi na niego punkty pomiarowe.

Obszar, na jakim rysowany jest wykres jest powiększony o $\text{paddingMultiplier} * (\max_{i \in 1, \dots, m} x_i - \min_{i \in 1, \dots, m} x_i)$ w celu zobrazowania elementu optymalnego również poza zakresem punktów pomiarowych. Domyślna wartość $\text{paddingMultiplier} = \frac{1}{6}$.

Argumenty przyjmowane przez tą funkcję opisane są w pliku `plotApproximation.m` (nad nagłówkiem funkcji).

Przykładowe wywołanie funkcji: `plotApproximation([0 1 2 3], [5 6 8 -5])`

Dodatkowo aby ułatwić sprawdzanie tej metody aproksymacji dla funkcji został stworzony skrypt `approximateFunction.m`. Zawiera on obliczenie wartości danej funkcji f , a następnie dodanie *szumu* do każdego węzła aproksymacji:

$$f_i = \text{rand}(S_{\max} - S_{\min}) + f(x_i) * (\text{rand}(P_{\max} - P_{\min}) + P_{\min})$$

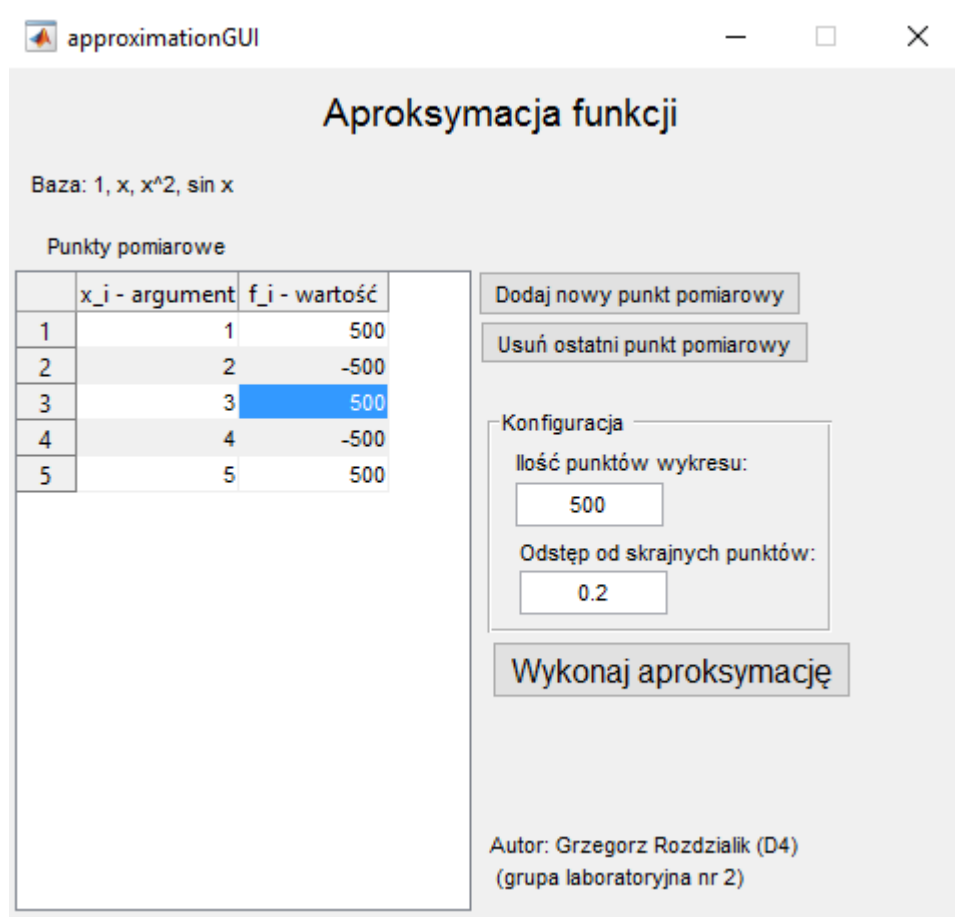
gdzie $S_{\min}, S_{\max} \in \mathbb{R}$ są stałymi, które zostaną dodane do każdej obliczonej wartości funkcji, $P_{\min}, P_{\max} \in \mathbb{R}$ oznaczają zakres procentowy przez który zostaną pomnożone obliczone wartości funkcji.

Dzięki temu można faktycznie sprawdzić poprawność aproksymacji dla zaburzonych danych.

8 Interfejs graficzny

Do metody został dołączony interfejs graficzny umożliwiający łatwe wprowadzanie punktów pomiarowych, zmianę parametrów (współczynnik powięk-

szenia obszaru, a także ilość punktów wewnątrz obszaru, na których zostanie narysowany wykres) i dokonanie z tymi danymi aproksymacji.



Rysunek 1: Interfejs graficzny

9 Bibliografia

1. Informacje z wykładu *Metod numerycznych 2* (wydział MiNI PW, dr Iwona Wróbel), w szczególności temat *Aproksymacja średniokwadratowa dyskretna*.