26 listopada 2019

Paweł Golik
298868
G1

Interpolacja funkcjami kwadratowymi na obszarze $D = \{(x, y) \in \mathbf{R^2} : |x| + |y| \leqslant 1\}$ podzielonym na $4n^2$ trójkątów przystających. Tablicowanie funkcji, przybliżenia i błędu w środkach ciężkości trójkątów. Obliczanie błędu średniokwadratowego w tych punktach.

Projekt nr 22

1 Wstęp

Zamierzeniem projektu jest zbadanie problemu interpolacji funkcji dwóch zmiennych rzeczywistych, dobrze określonej na obszarze $D = \{(x, y) \in \mathbf{R^2} : |x| + |y| \leqslant 1\}$ przy wykorzystaniu interpolacji na trójkącie funkcjami kwadratowymi w bazie $B = (x, y, x^2, y^2, xy)$, w węzłach wyznaczonych przez wierzchołki oraz środki boków trójkątów wzajemnie przystających do siebie, powstałych przez podział obszaru D na $4n^2$ trójkątów.

W projekcie uwzględniono zagadnienie wpływu liczby trójkątów n na czas i dokładność obliczeń, tablicowanie wartości funkcji, przbliżenia w węzłach oraz błędu średniokwadratowego w środkach ciężkości trójkątów. Analizie poddany został także sposób generowania podziału obszaru na trójkąty.

Algorytm przetestowany został na różnych przypadkach funkcji dwóch zmiennych rzeczywistych $f : D \to \mathbf{R}$ dla różnych wartości n, od 1 do n z krokiem 1. Dla większości funkcji interpolacja była dokładna dla niewielkiego n, z wyjątkiem funkcji szybko zmieniających się. Spadek wielkości błędu obserwowano jedynie do pewnej wartości n.

2 Opis użytych metod

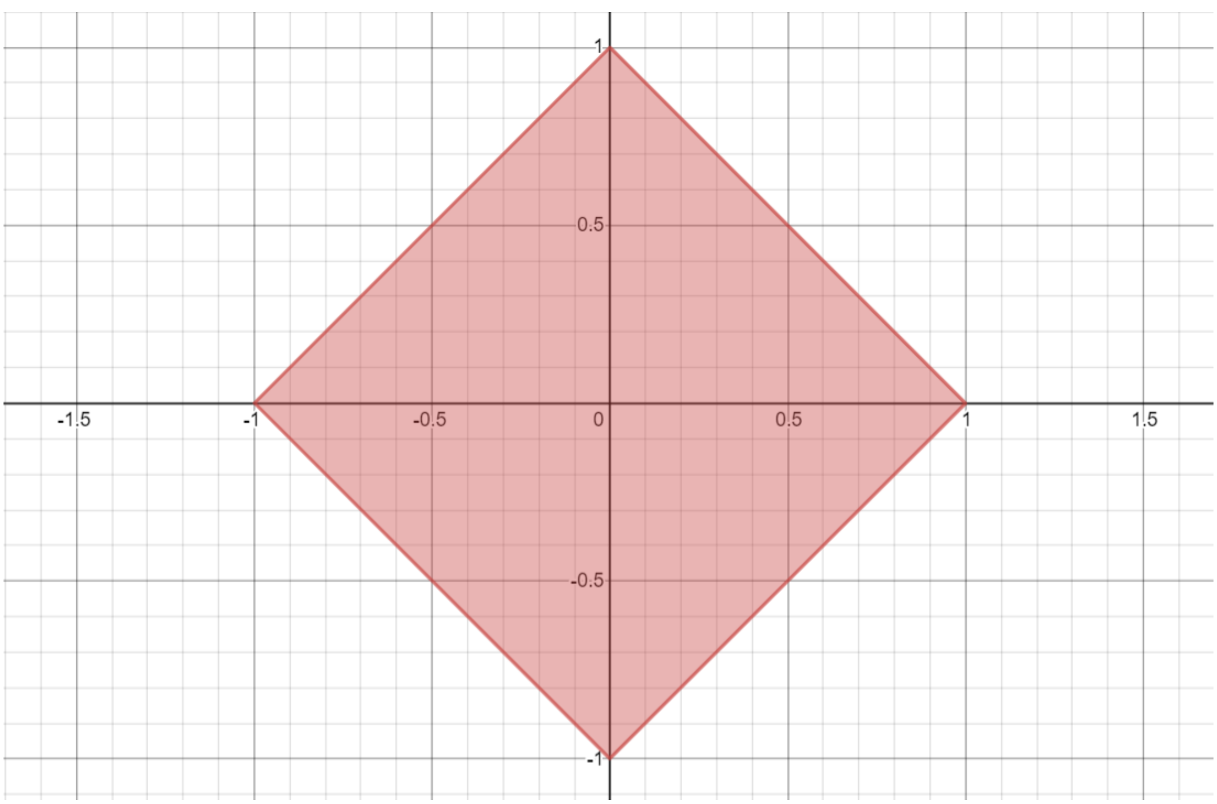
Podział obszaru D na $4n^2$ trójkątów przystających:

Dzielenie obszaru na trójkąty przystające realizowane jest poprzez podzielenie kwadratu na n^2 mniejszych kwadratów, a następnie każdy kwadrat rozdzielony jest na cztery trójkąty wyznaczone poprzez jego przekątne.

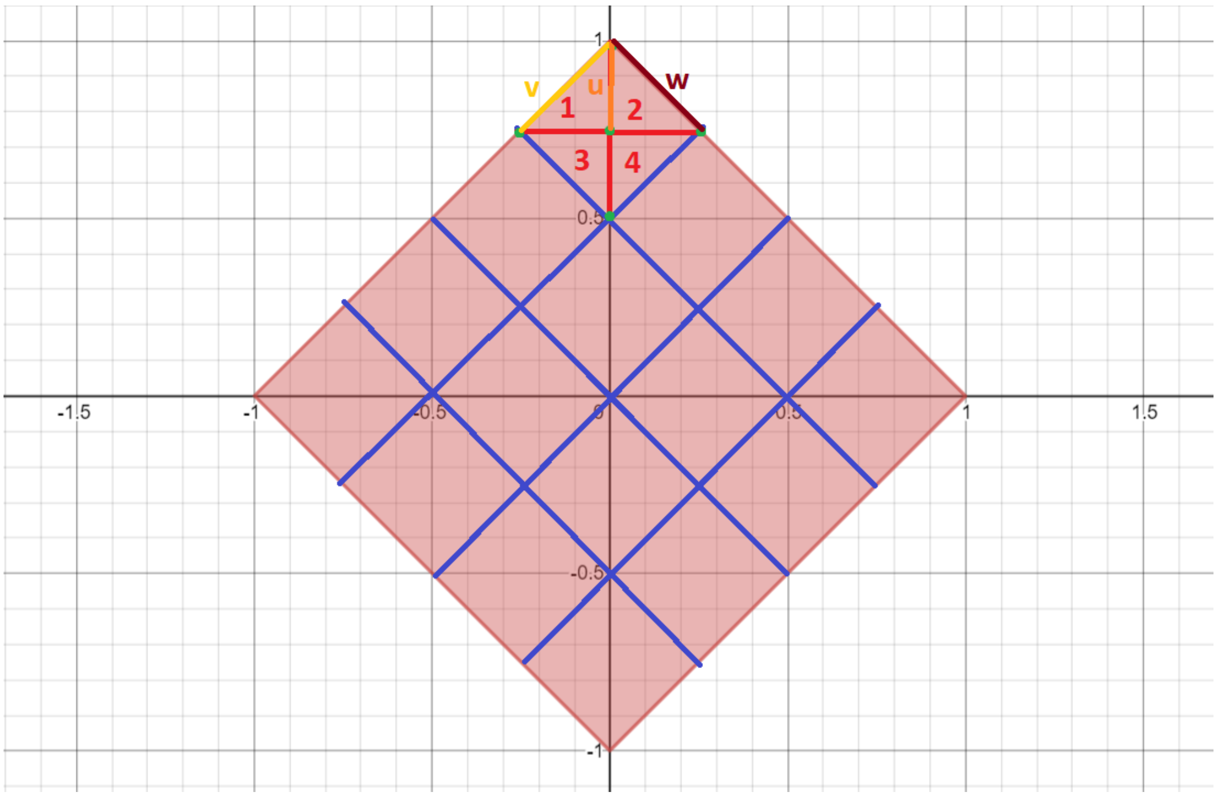
W pierwszej kolejności wyzanczane są wierzchołki czterech "górnych" trójkątów, zaczynając od punktu P = (0,1) przy wykorzystaniu wyznaczonych wektorów $v = [\frac{-1-0}{n}, \frac{0-1}{n}]$, $u = [\frac{0-0}{n}, \frac{0-1}{n}]$ oraz $w = [\frac{1-0}{n}, \frac{0-1}{n}]$.

Następnie za pomocą wektora v generowane są kolejne punkty trójkątów znajdujących się w górnym pasku mniejszych kwadratów, poprzez dodawanie do wierzchołków w pierwszym kwadracie kolejnych wielokrotności wektora v w pętli $for(i = 1 : n-1)$.

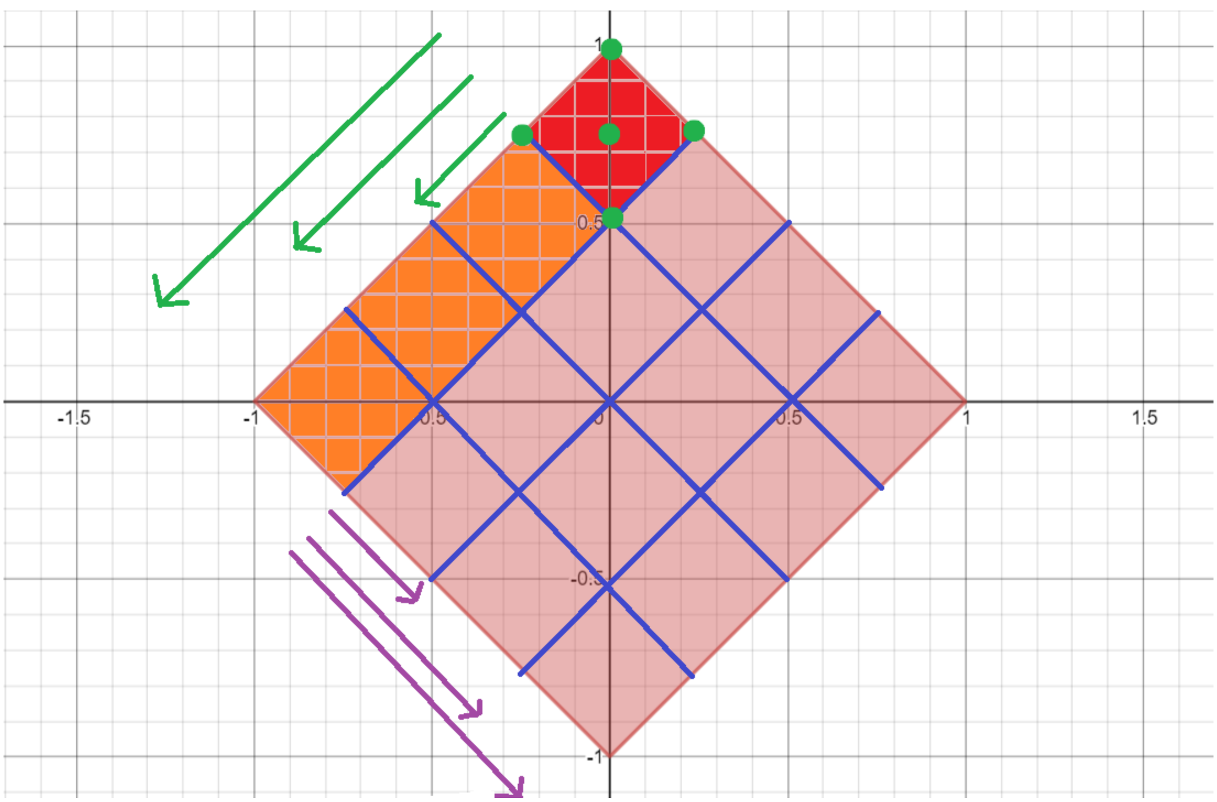
Po wygenerowaniu pierwszego paska kwadratów, generujemy kolejne $n - 1$ pasków za pomocą przesunięcia górnych kwadratów kolejnymi wielokrotnościami wektora w w pętli $for(i = 1 : n - 1)$.



Rysunek 1: Obszar D



Rysunek 2: Przykładowy podział D dla $n = 4$



Rysunek 3: Generowanie kolejnych punktów za pomocą wektorów u i w.

Znajdowanie funkcji interpolującej:

Niech $P_n := \{p : p(x,y) = \sum_{0 \leqslant \kappa+\mu \leqslant n} a_{\mu\kappa} x^\mu y^\kappa\}$, gdzie $a \in \mathbf{R}$.
Naszym celem jest znalezienie funkcji $p \in P_n$, takiej że dla funkcji interpolowanej $f = f(x,y)$:

$$p(x_i, y_j) = f(x_i, y_j) \quad \forall i, j \quad 0 \leqslant i + j \leqslant n$$

W naszym przypadku interpolacja odbywa się za pomocą funkcji kwadratowych, zatem $n = 2$, a zgodnie z przyjętą przez nas bazą przestrzeni wielomianów dwóch zmiennych, stopnia co najwyżej n:

$$P_n : B = (x, y, x^2, y^2, xy)$$

. wielomian interpolujący p ma postać:

$$p(x,y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2$$

Aby wyznaczyć współczynniki wielomianu potrzebujemy 6 węzłów (3 wierzchołków trójkąta oraz 3 środków jego boków) :

$$p(x_0, y_0) = a_0 + a_1 x_0 + a_2 y_0 + a_3 x_0 y_0 + a_4 x_0^2 + a_5 y_0^2 = f(x_0, y_0)$$

$$p(x_1, y_1) = a_0 + a_1 x_1 + a_2 y_1 + a_3 x_1 y_1 + a_4 x_1^2 + a_5 y_1^2 = f(x_1, y_1)$$

$$p(x_2, y_2) = a_0 + a_1 x_2 + a_2 y_2 + a_3 x_2 y_2 + a_4 x_2^2 + a_5 y_2^2 = f(x_2, y_2)$$

$$p(x_3, y_3) = a_0 + a_1 x_3 + a_2 y_3 + a_3 x_3 y_3 + a_4 x_3^2 + a_5 y_3^2 = f(x_3, y_3)$$

$$p(x_4, y_4) = a_0 + a_1 x_4 + a_2 y_4 + a_3 x_4 y_4 + a_4 x_4^2 + a_5 y_4^2 = f(x_4, y_4)$$

$$p(x_5, y_5) = a_0 + a_1 x_5 + a_2 y_5 + a_3 x_5 y_5 + a_5 x_5^2 + a_5 y_5^2 = f(x_5, y_5)$$

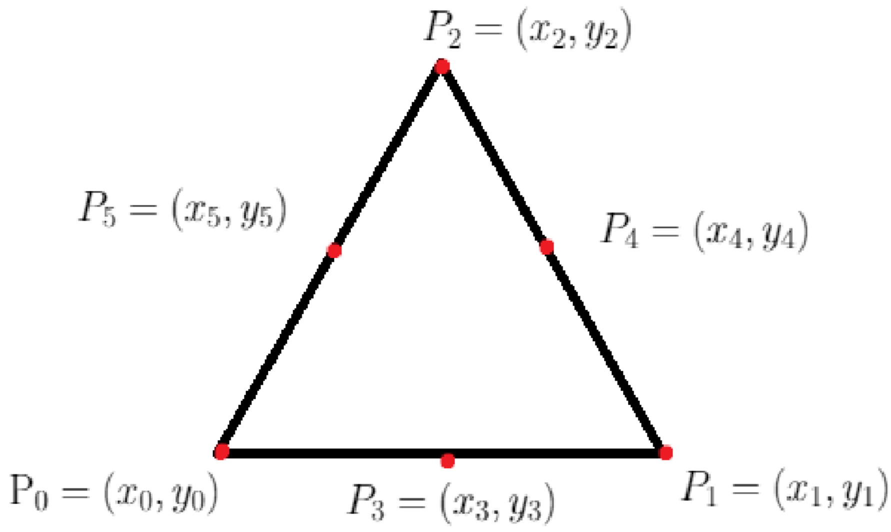
Otrzymaliśmy układ równań:

$$\begin{bmatrix} 1 & x_0 & y_0 & x_0 y_0 & {x_0}^2 & {y_0}^2 \\ 1 & x_1 & y_1 & x_1 y_1 & {x_1}^2 & {y_1}^2 \\ 1 & x_2 & y_2 & x_2 y_2 & {x_2}^2 & {y_2}^2 \\ 1 & x_3 & y_3 & x_3 y_3 & {x_3}^2 & {y_3}^2 \\ 1 & x_4 & y_4 & x_4 y_4 & {x_4}^2 & {y_4}^2 \\ 1 & x_5 & y_5 & x_5 y_5 & {x_5}^2 & {y_5}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} f(x_0, y_0) \\ f(x_1, y_1) \\ f(x_2, y_2) \\ f(x_3, y_3) \\ f(x_4, y_4) \\ f(x_5, y_5) \end{bmatrix}$$

Rozwiązanie układu to współczynniki wielomianu p, które otrzymaliśmy przy wykorzystaniu metody GEPP, zbieżnej dla każdej macierzy nieosobliwej. Po skonstruowaniu wielomianu p obliczamy jego wartość w środku ciężkości trójkąta:

$$C = (\frac{x_0 + x_1 + x_2}{3}, \frac{y_0 + y_1 + y_2}{3})$$

oraz zapisujemy wartość $p(C)$ w tablicy S wraz z wartością $f(C)$ oraz błędem $|p(C) - f(C)|$.



Rysunek 4: Węzły

Obliczanie błędu średniokwadratowego:

Po obliczeniu wartości błędów w środkach ciężkości wszystkich $4n^2$ trójkątów, dzięki ich tablicowaniu, łatwo możemy obliczyć błąd średniokwadratowy:

$$MSE = \frac{1}{4n^2} \sum_{k=1}^{k=4n^2} (f(C_k) - p(C_k))^2,$$

gdzie $C_k = (x_{C_k}, y_{C_k})$ - środek ciężkości k-tego trójkąta

Sprawdzenie poprawności:

Poprawność działania algorytmu zrealizowano przez obliczenie wartości w punktach za pomocą danej funkcji i porówanie wyników - błąd jest małego rzędu, zatem można uznać znalezienie wielomianu p za poprawne .

Inne:

Czas trwania interpolacji dla danego n mierzymy za pomocą wbudowanej funkcji *tic ... toc*. Czas zapisywanany jest w tablicy, w celu wykreślenia odpowiedniego wykresu ułatwiającego zinterpretowanie wyników projektu.

3 Prezentacja przykładów oraz wyników

Działanie programu dla funkcji dobrze określonych na D

Początkowo program tworzy siatkę trójkątów dla danego n, przystępuje do obliczania wartości w węzłach za pomocą wywołania funkcji interpolowanej. Używając metody GEPP wyznacza współczynniki wielomianu interpolującego i oblicza jego wartość w środkach ciężkości trójkąta, wartość funkcji intepolowanej w tych punktach oraz błąd, po czym zapisuje je w tablicy. Każda z tych czynności wykonywana jest dla każego z trójkątów, stąd liczba iteracji wynosi $4n^2$.

Operacje powtarzane są dla każdego n z zakresu $1 : k$, gdzie $k \in \mathbf{N}$ i jest ustalone na początku programu, w celu zbadania zależności zmiany błędu i czasu działania programu od liczby podziałów n.

Przykładowe interpolacje:

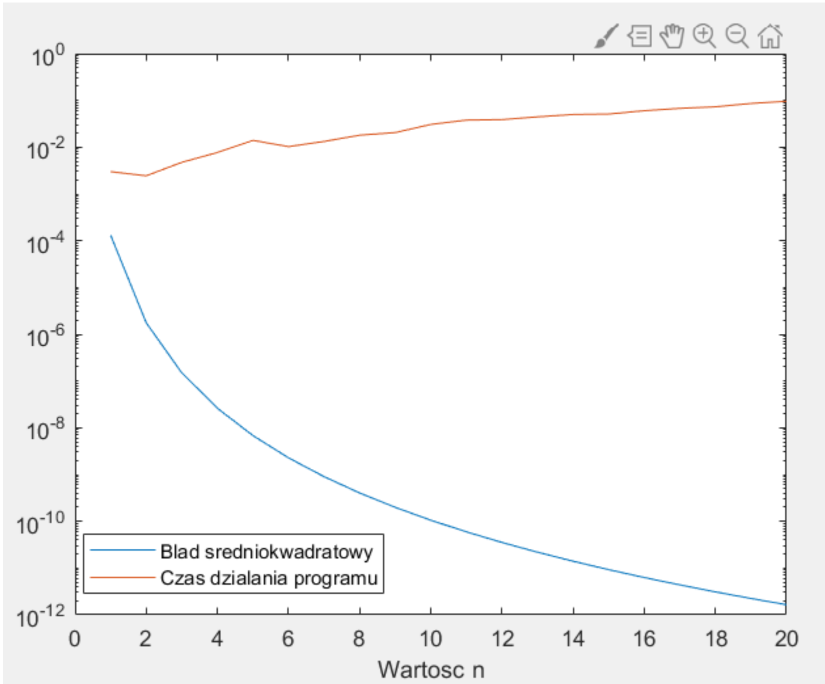
Na poniższych rysunkach sporządzono wykresy błędu średniokwadratowego *(niebieski)* oraz czasu trwania programu *(w sek.) (czerwony)* w zależności od liczby podziałów n dla różnych funkcji interpolowanych poprawnie określonych na D. Wykorzystano funkcję *semilogy* w celu lepszego zobrazowania rzędów wielkości.

Przykłady $1 - 4$ to testowe przykłady algorytmu będące różnymi funkcjami dobrze określonymi na D, których zmienność na obszarze nie jest zbyt duża - funkcje nie oscylują. Przykłady te obrazują działanie programu dla funkcji często występujących w matematyce takich jak: exp, log, sin, cos.

Przykład 5 to wariant interpolacji wielomianu.

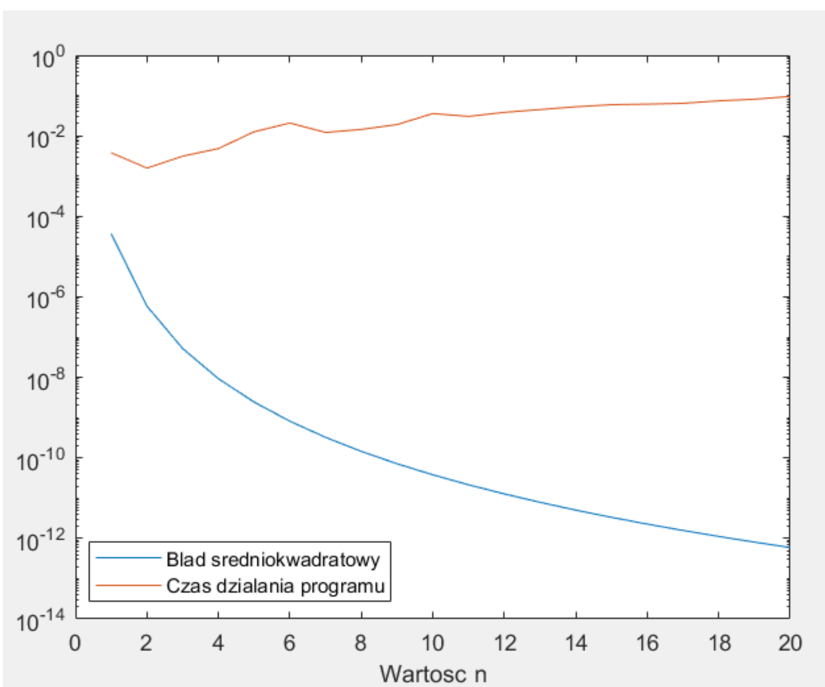
Przykłady $6, 7$ to interpolacja funkcji oscylujących na D - funkcje te bardzo szybko się zmieniają.

Przykład 1:



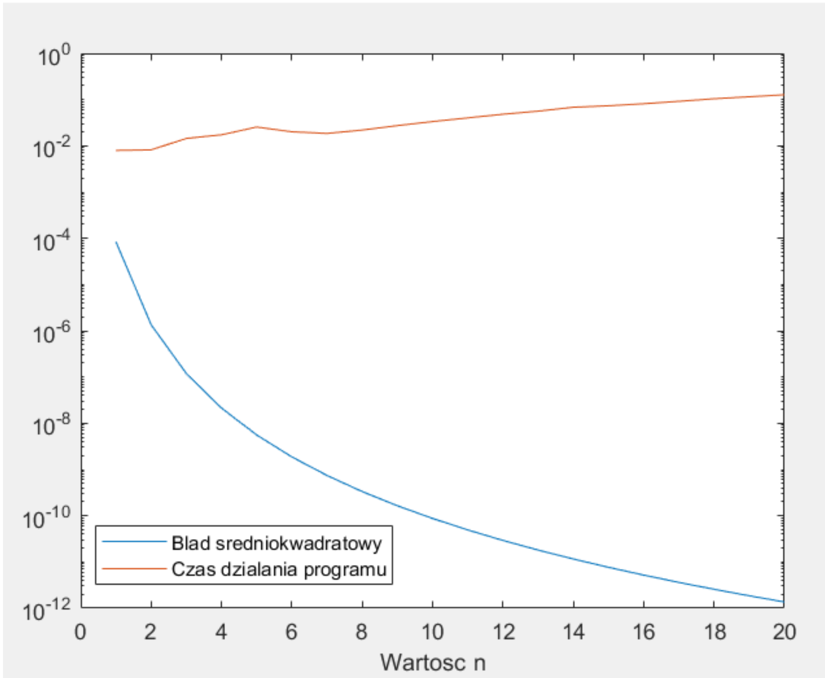
Rysunek 5: $f(x, y) = e^x + e^y$

Przykład 2:



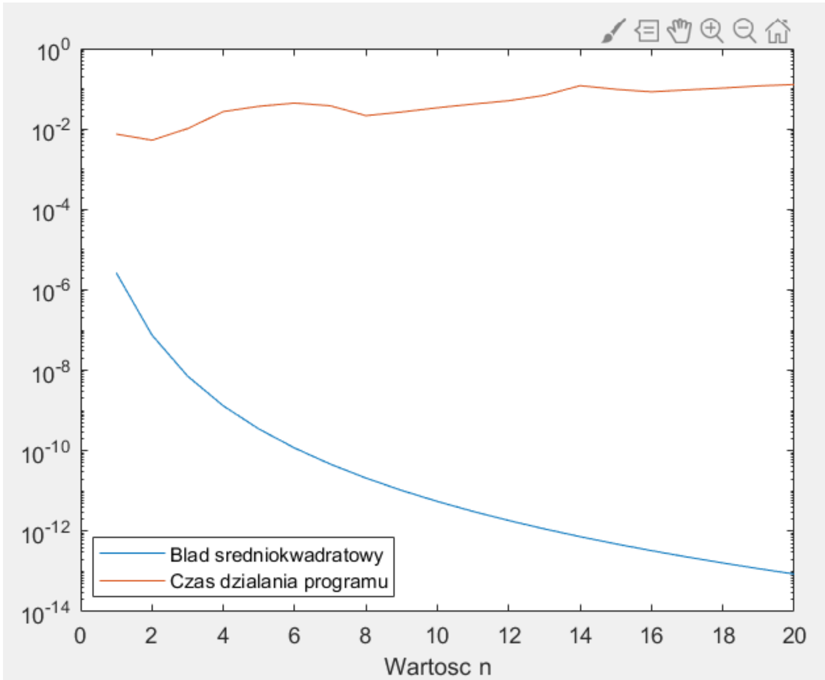
Rysunek 6: $f(x, y) = sin(x) + cos(y)$

Przykład 3:



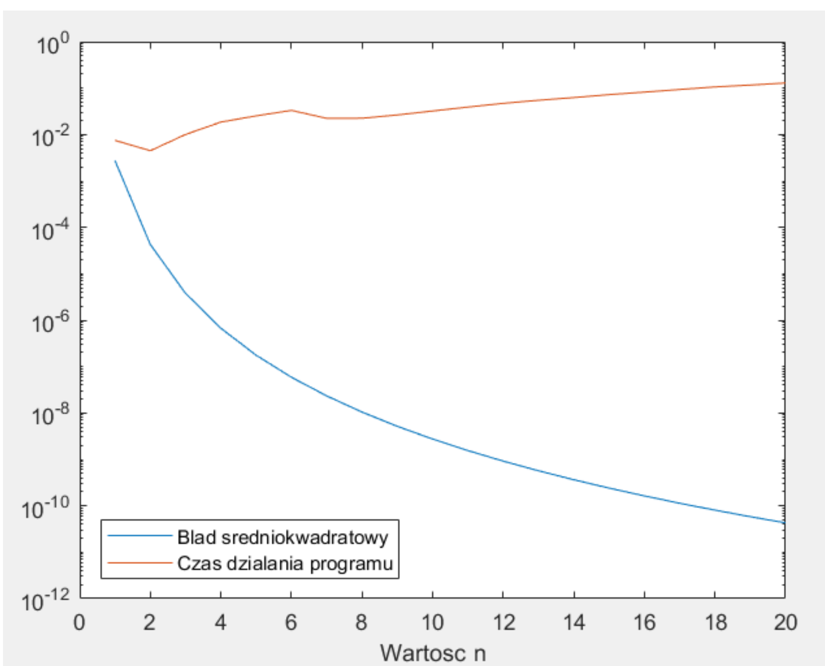
Rysunek 7: $f(x,y) = log(\sqrt{e^{xcos(x)-ysin(y)}})$

Przykład 4:



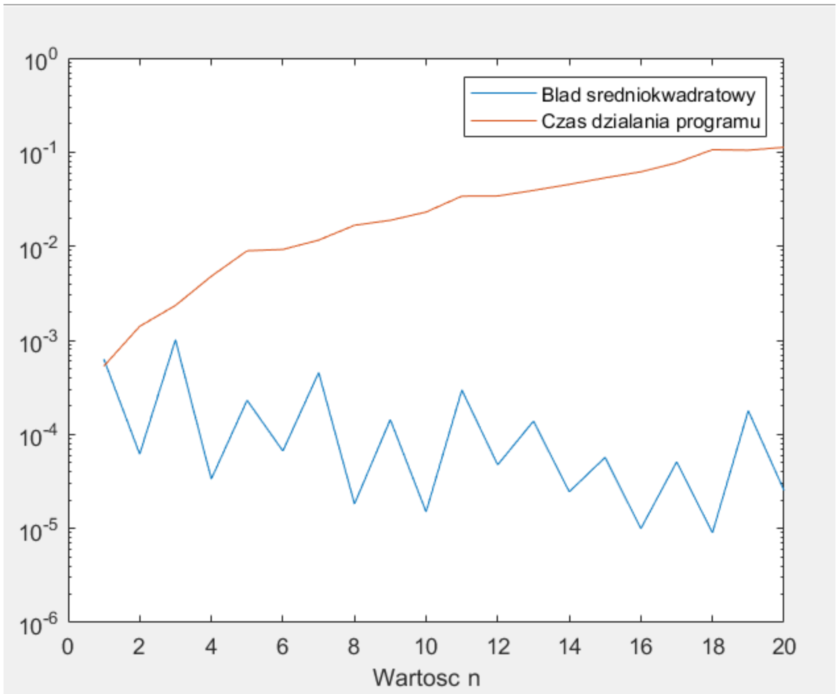
Rysunek 8: $f(x,y) = \sqrt{|xy| + 1}$

Przykład 5:



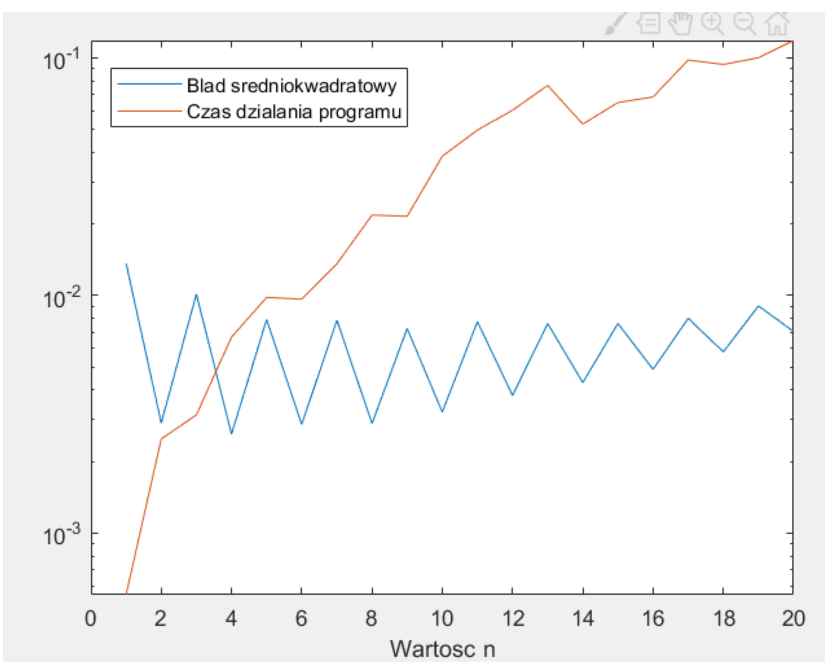
Rysunek 9: $f(x,y) = x^3 + y^3 + xy + 1$

Przykład 6:



Rysunek 10: $f(x,y) = \frac{1}{log|x+y|+20}$

Przykład 7:



Rysunek 10: $f(x,y) = \frac{1}{log|x-y|+5}$

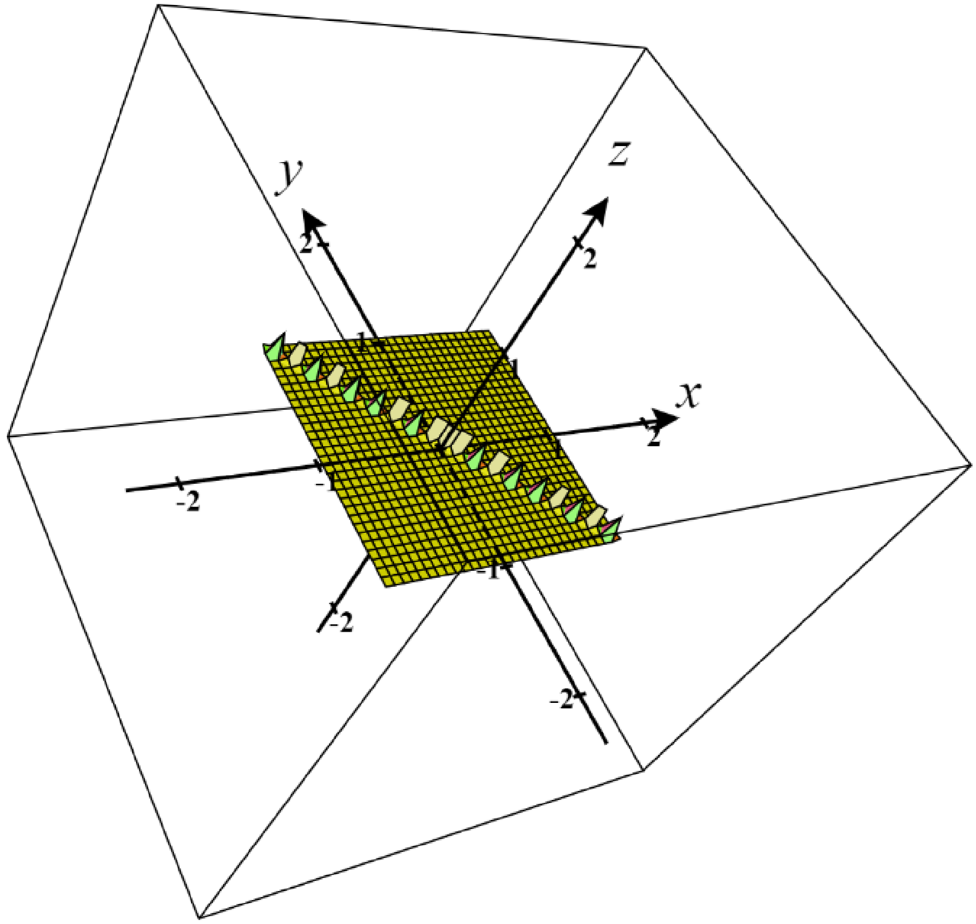
Interpretacja

Metoda interpolacji funkcjami kwadratowymi podzielonym na $4n^2$ trójkątów dobrze radzi sobie z aproksymacją funkcji interpolowanej już dla niewielkiego n. Dla wielu przypadków juz dla podziału na cztery trójkąty ($n = 1$) błąd jest rzędu około 10^{-3}, a uzyskanie błędu rzędu 10^{-10} wymaga jedynie $n = 10$. We wszystkich przypadkach błąd szybko maleje dla początkowych n, a począwszy od ok. $n = 16$ wielkość błędu nie zmiejsza się znacząco, natomiast następuje wzrost czasu trwania programu do ok. 1 sek co może powodować zbyt długie działanie programu wykorzystującego powyższą metodę wielokrotnie przy żądanej dużej dokładności. Wtedy lepiej jest zastosować inną metodę.

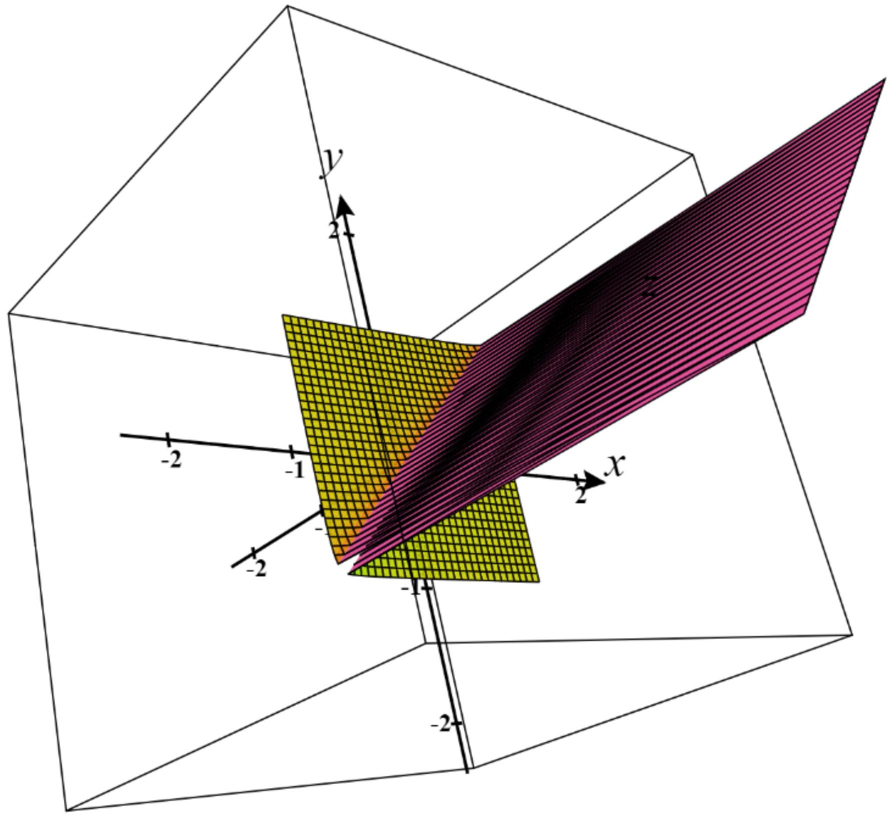
W przykładzie 5, dla interpolacji wielomianu spadek błędu następuje wolniej, co jest skutkiem próby interpolowania wielomianu innym wielomianem p. Dla $n = 18$ błąd jest rzędu 10^{-10} co jest wynikiem gorszym niż dla "zwykłych" funkcji osiągających dla pdoobnego podziału błąd rzędu 10^{-12}.

W przykładach $6, 7$, błąd oscyluje i nie zmniejsza się wraz ze wzrostem n. Dla podziału obszaru D dla $n = 19$ błąd jest mniejszy niż dla $n = 2$. Trudno jest przewidzieć zachowanie błędu dla kolejnych n, a czas trwania programu rośnie przechodząc wraz ze wzrostem n w niekorzystne dla nas wartości.

Zjawisko to jest spowodowane oscylacją funkcji interpolowanej $f(x, y) = \frac{1}{log|x+y|+20}$ dla (x, y) znajdujących się w pobliżu $x = y$, co uniemożliwia "nadążanie" wielomianowi interpolującemu za funkcją f. Stąd błąd także oscyluje. Analogiczne zjawisko na jeszcze większą skalę zaobserwowano dla funkcji $f(x, y) = \frac{1}{log|x-y|+5}$



Rysunek 11: Rzeczywisty przebieg funkcji interpolowanej $f(x,y) = \frac{1}{log|x+y|+20}$



Rysunek 12: Rzeczywisty przebieg funkcji interpolowanej $f(x,y) = \frac{1}{log|x-y|+5}$

4 Wnioski

Algorytm pozwala na interpolację o złożoności liniowej względem liczby trójkątów, na które dzielimy obszar D. Dla funkcji szybko oscylujących otrzymana dokładność jest znacząco mniejsza niż dla podobnego podziału z funkcją zmieniającą się "normalnie". W takich przypadkach należy rozważyć inną metodę interpolacji lub wprowadzić modyfikacje algorytmu.

Dla większości przypadków błąd oraz czas odpowiednio maleje i rośnie w podobny sposób wraz ze wzrostem liczby podziałów n. Dalsze zwiększanie n powinno być jednak przemyślane z uwagi na coraz większy czas wykonania i powinno być stosowane dla małej liczby interpolacji, w których wysoka precyzja stanowi czynnik kluczowy dla projektu.

Literatura

[1] https://www.mathworks.com

[2] http://www.mini.pw.edu.pl/ iwrobel/MN2_informatyka_zima_2019-20/Materialy