Michał Rdzany gr.E3 08.11.2017r.

Sprawozdanie
Metody numeryczne 2

2. Interpolacja

Temat 21:

Interpolacja funkcjami kwadratowymi na kwadracie podzielonym na $2n^2$ trójkątów przystających. Tablicowanie funkcji, przybliżenia i błędu w środkach ciężkości trójkątów. Obliczenie błędu średniokwadratowego w tych punktach.

Opis problemu:

Szukamy funkcji $w(x, y)$ interpolującej funkcję $f(x, y)$ na kwadracie o podanych wierzchołkach.

Funkcja w składa się z $2n^2$ funkcji postaci $a + b \cdot x + c \cdot y + d \cdot x^2 + e \cdot y^2 + g \cdot x \cdot y$, gdzie $a, b, c, d, e, g \in \mathbb{R}$, interpolujących funkcję f na $2n^2$ trójkątach przystających, na które dzielimy podany kwadrat.

Szukamy także wartości funkcji $w - f$ w środkach ciężkości tych trójkątów oraz błędu średniokwadratowego w tych punktach.

Opis metody:

Na początek tworzymy macierz wektorów współczynników $M \in M_{6\times 2\times n\times n}(\mathbb{R})$, używając funkcji $makeM(n, f, Z)$, gdzie $Z \in M_{2\times 2\times 2}(\mathbb{R})$, jest macierzą wierzchołków kwadratu, taką że $Z = \begin{bmatrix} W_1 & W_2 \\ W_3 & W_4 \end{bmatrix}, W_i = \underbrace{[w_{i_x}, w_{i_y}]}_{\text{współrzędne x i y}}$ i wierzchołki W_1 i W_4 leżą na tej samej przekątnej.

Funkcja $makeM(n, f, Z)$

Jeżeli $n \notin \mathbb{N}_+$ lub rozmiar macierzy Z jest inny od oczekiwanego, to wypisujemy komunikat o niepoprawności podanych argumentów i kończymy funkcję.

Jeśli z podanych wierzchołków nie powstaje prostokąt, wypisujemy odpowiedni komunikat i kończymy funkcję.
Jeśli ten prostokąt nie jest kwadratem wypisujemy odpowiedni komunikat i dajemy wywołującemu możliwość kontynuowania funkcji, ponieważ opisywana metoda działa na wszystkich prostokątach.

Tworzymy macierze współrzędnych węzłów X i Y.

Dla każdego z n^2 trójkątów:

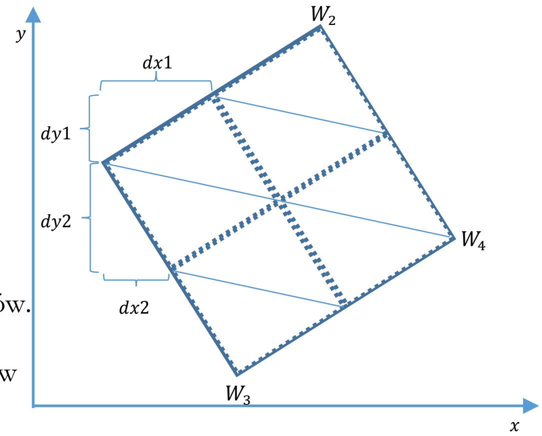
- tworzymy wektor jego 6 węzłów $nodes$.
- Tworzymy macierz A o i-tym wierszu $[1, x, y, x^2, y^2, x \cdot y]$, gdzie x, y są współrzędnymi i-tego węzła z $nodes$.
- Tworzymy wektor $fn = f(nodes)$.
- Rozwiązujemy układ równań z macierzą współczynników A i wektorem danych fn używając funkcji $A\backslash fn$. Macierz A może być bardzo źle uwarunkowana (np. dla dużych $|x|, |y|$ lub dla bliskich sobie węzłów). Otrzymany wektor współczynników wsp' może się wtedy znacznie różnić od dokładnego wektora współczynników wsp, ale nie ma to znaczenia, bo stworzony na podstawie tych współczynników wielomian w' przy danej precyzji obliczeń i tak przecina funkcję f w określonych węzłach. Jeśli A jest osobliwa to rozwiązanie wyznaczamy wolniejszą funkcją $pinv(A) \cdot fn$.
- Wynik wpisujemy do odpowiedniej części macierzy M.

Zwracamy macierz M.

Koniec funkcji $makeM(n, f, Z)$

Aby wyznaczyć wartość funkcji $w(x, y)$ używamy funkcji $findTriangle(x, y, Z, n)$ do znalezienia odpowiedniego wektora w macierzy M, a następnie wywołujemy funkcję $valueW(x, y, wsp)$ która zwraca szukaną wartość.

Funkcja $findTriangle(x, y, Z, n)$



- Tworzymy zmienne $dx1, dy1, dx2, dy2$.

- Z użyciem wyszukiwania binarnego, wyznaczamy mały kwadrat, w którym znajduje się punkt (x, y).

- Wybieramy jeden z 2 trójkątów w tym kwadracie na podstawie odległości punktu (x, y) od niewspólnych wierzchołków tych trójkątów.

- Zwracamy 3 indeksy opisujące położenie wektora współczynników odpowiadającego temu trójkątowi w macierzy M.

Koniec funkcji $findTriangle(x, y, Z, n)$

Funkcja $\boldsymbol{valueW(x, y, wsp)}$

Zwracamy wartość wyrażenia $[1, x, y, x^2, y^2, x \cdot y] \cdot wsp$.

Koniec funkcji $\boldsymbol{valueW(x, y, wsp)}$

Do tablicowania funkcji, przybliżenia, błędu w środkach ciężkości trójkątów i obliczania błędu średniokwadratowego w tych punktach używamy funkcji $tab(f, M, Z)$.

Funkcja $\boldsymbol{tab(f, M, Z)}$

- Tworzymy macierz wszystkich wierzchołków trójkątów.
- Tworzymy macierz $B \in M_{2 \cdot n^2 \times 5}(\mathbb{R})$, gdzie i-ty wiersz macierzy B ma postać $[x,\ y,\ f(x,y),\ w(x,y),\ f(x,y) - w(x,y)]$, gdzie x, y są współrzędnymi i-tego środka ciężkości.

  Współrzędne środka ciężkości trójkąta o współrzędnych wierzchołków C, D, E wyznaczamy ze wzoru $\frac{C+D+E}{3}$.

  Do obliczania $w(x,y)$ wykorzystujemy tylko funkcję $valueW$ ponieważ przeglądamy środki ciężkości w ustalonej kolejności. Dzięki temu tworzymy wiersze macierzy B ze złożonością $O(1)$ zamiast $O(\log(n))$.

- Wyznaczamy błąd średniokwadratowy $sk = \sqrt{\frac{\sum_{i=1}^{2 \cdot n^2} B(i,5)^2}{n}}$.
- Zwracamy B i sk.

Koniec funkcji $\boldsymbol{tab(f, M, Z)}$

Złożoność używanych funkcji:

tab	$O(n^2)$
$valueW$	$O(1)$
$makeM$	$O(n^2)$
$findTriangle$	$O(\log(n))$

Wyznaczenie $w(x,y)$ w k losowych punktach używając opisanej metody ma złożoność $O(n^2 + k \cdot \log(n))$.
Tworzymy macierz M ze złożonością $O(n^2)$ i wyznaczamy $w(x,y)$ ze złożonością $O(\log(n))$.

Przykłady i wnioski:

Skrypty przykładów ustawiają zmienne f, M, B, sk i używają zmiennych n(domyślnie 20) i Z(domyślnie $\begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$)
Przedstawione pod przykładami1-4 wykresy są rysowane dla domyślnych Z i n.
Przykłady1-4 podają czas wykonania i wypisują sk.
Dla $n \leq 8$ dodatkowo wypisują B.
Dla $n \leq 4$ dodatkowo wypisują M.

Funkcja $draw(Z, M, f, i)$ rysuje funkcje w i f dla punktów o współrzędnych z przedziału i(argument opcjonalny).
Funkcja w jest rysowana w 2500 równoodległych punktach.
Wartości w dla punktów należących do kwadratu Z są rysowane na niebiesko, a dla pozostałych na czerwono.
Funkcja ta jest wywoływana na koniec przykładów.

Przykład 1

$$f(x,y) = 3 + 7x^2 + 4y - 3xy$$

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 20$ program podaje $sk = 1.3552 \cdot 10^{-14}$ w $\sim 0.02s$.

Dla $Z = \begin{bmatrix} [10,0] & [20,10] \\ [0,10] & [10,20] \end{bmatrix}$, $n = 20$ program podaje $sk = 1.3262 \cdot 10^{-14}$ w $\sim 0.02s$.

Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 10$ program podaje $sk = 4.8761 \cdot 10^{-15}$ w $\sim 0.01s$.

Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 4$ program podaje $sk = 3.9752 \cdot 10^{-15}$ w $\sim 0.001s$.

Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 2$ program podaje $sk = 0$ w $\sim 0.001s$.

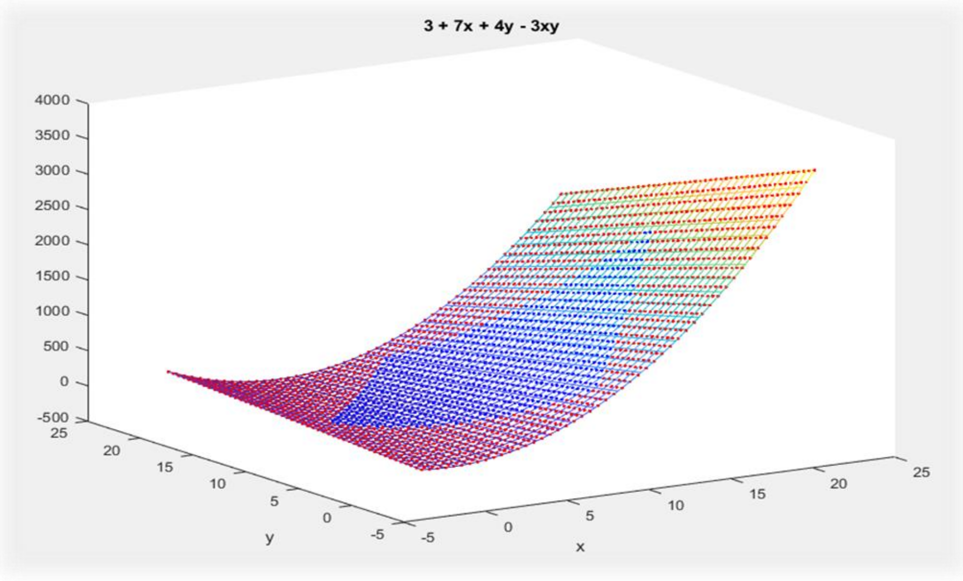
Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 40$ program podaje $sk = 5.0041 \cdot 10^{-15}$ w $\sim 0.08s$.

Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 200$ program podaje $sk = 1.2919 \cdot 10^{-14}$ w $\sim 1.68s$.

Dla $Z = \begin{bmatrix} [0,500] & [500,500] \\ [0,0] & [500,0] \end{bmatrix}$, $n = 200$ program podaje $sk = 3.1902 \cdot 10^{-11}$ w $\sim 1.68s$.

Dla $Z = \begin{bmatrix} [0,500] & [500,500] \\ [0,0] & [500,0] \end{bmatrix}$, $n = 4$ program podaje $sk = 3.4091 \cdot 10^{-11}$ w $\sim 0.002s$.

Dla dowolnych Z i n wszystkie wektory macierzy M są równe $[3,0,4,7,0,-3]^T$, więc $w = f$.
$w = f$ także poza interpolowanym przedziałem, co można zobaczyć używając funkcji $draw$.
$sk \neq 0$ z powodu błędów zaokrągleń, które są większe dla większych $f(x,y)$,więc sk jest większy dla większych $f(x,y)$.
Czas wykonania programu jest proporcjonalny do n^2.



Przykład 2

$$f(x,y) = x \cdot \sin x + 10 \cdot \cos y$$

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 4$ program podaje $sk = 1.7794$ w $\sim 0.03s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 10$ program podaje $sk = 6.9489 \cdot 10^{-2}$ w $\sim 0.01s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 20$ program podaje $sk = 6.4771 \cdot 10^{-3}$ w $\sim 0.03s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 100$ program podaje $sk = 4.3987 \cdot 10^{-5}$ w $\sim 0.5s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 200$ program podaje $sk = 5.4642 \cdot 10^{-6}$ w $\sim 1.75s$.

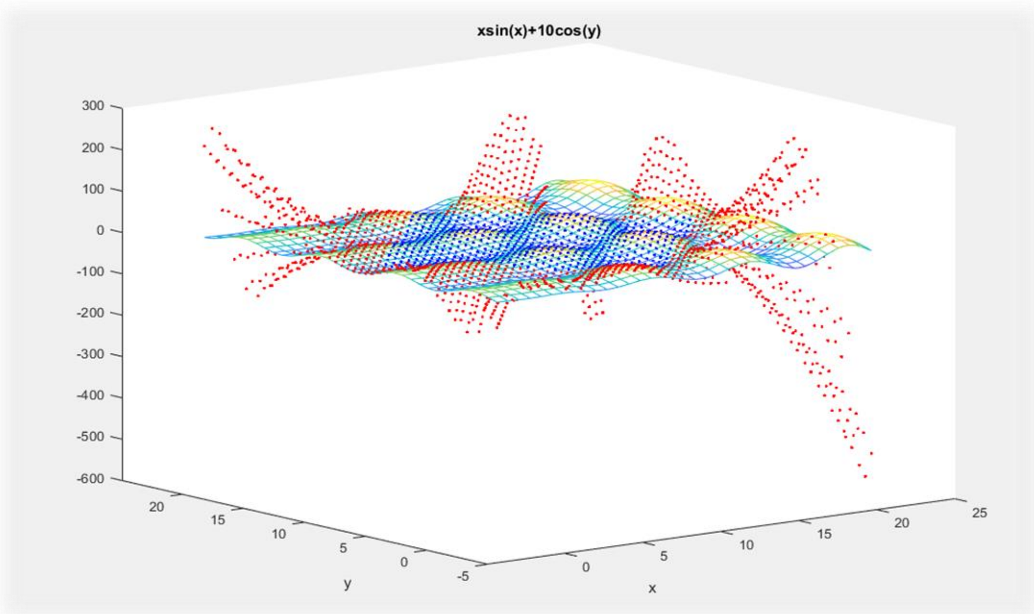
Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 400$ program podaje $sk = 6.8195 \cdot 10^{-7}$ w $\sim 6.7s$.

Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 20$ program podaje $sk = 7.7037 \cdot 10^{-4}$ w $\sim 0.04s$.

Dla $Z = \begin{bmatrix} [0,100] & [5,100] \\ [0,95] & [5,95] \end{bmatrix}$, $n = 20$ program podaje $sk = 7.9651 \cdot 10^{-4}$ w $\sim 0.04s$.

Dla $Z = \begin{bmatrix} [95,5] & [100,5] \\ [95,0] & [100,0] \end{bmatrix}$, $n = 20$ program podaje $sk = 6.0737 \cdot 10^{-3}$ w $\sim 0.04s$.

Błąd średniokwadratowy zwiększa się ze wzrostem odległości między węzłami (dla mniejszych n i dla większych kwadratów). Ponadto sk jest większy dla dużych co do modułu współrzędnych x. Dzieje się tak dlatego, że $f''_{xx}(x,y)$ szybko się zmienia dla dużych $|x|$, a w każdym trójkącie $w''_{xx}(x,y)$ jest stała. Czas wykonania programu jest proporcjonalny do n^2.



Przykład 3

$$f(x,y) = (x \cdot y)^2$$

Dla $Z = \begin{bmatrix} [10,0] & [20,10] \\ [0,10] & [10,20] \end{bmatrix}$, $n = 10$ program podaje $sk = 2.3992$ w $\sim 0.01s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 20$ program podaje $sk = 3.00001 \cdot 10^{-1}$ w $\sim 0.02s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 50$ program podaje $sk = 1.9202 \cdot 10^{-2}$ w $\sim 0.1s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 100$ program podaje $sk = 2.4003 \cdot 10^{-3}$ w $\sim 0.4s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 200$ program podaje $sk = 3.0003 \cdot 10^{-4}$ w $\sim 1.65s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 500$ program podaje $sk = 1.9202 \cdot 10^{-5}$ w $\sim 10s$.

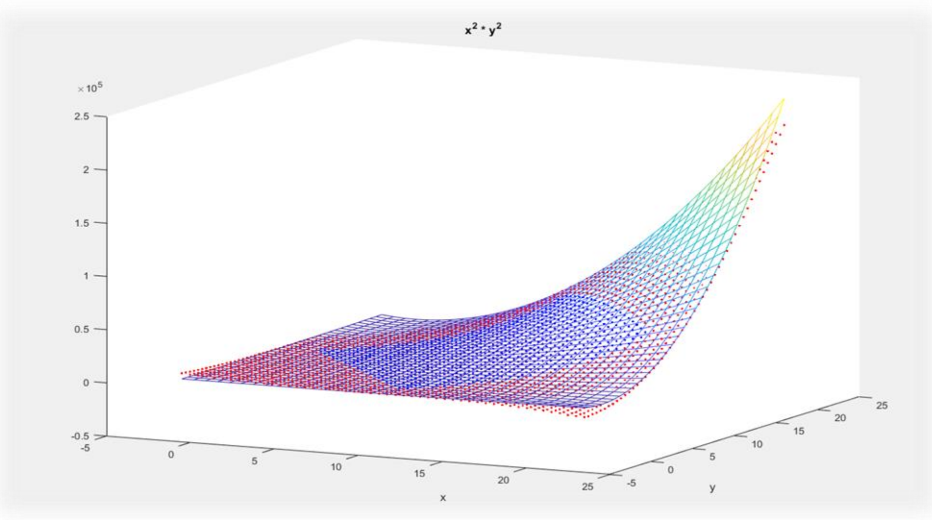
Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 100$ program podaje $sk = 2.5003 \cdot 10^{-5}$ w $\sim 0.42s$.

Dla $Z = \begin{bmatrix} [0,100] & [5,100] \\ [0,95] & [5,95] \end{bmatrix}$, $n = 100$ program podaje $sk = 4.6306 \cdot 10^{-4}$ w $\sim 0.42s$.

Dla $Z = \begin{bmatrix} [95,5] & [100,5] \\ [95,0] & [100,0] \end{bmatrix}$, $n = 100$ program podaje $sk = 4.6306 \cdot 10^{-4}$ w $\sim 0.42s$.

Dla $Z = \begin{bmatrix} [95,100] & [100,100] \\ [95,95] & [100,95] \end{bmatrix}$, $n = 100$ program podaje $sk = 9.0283 \cdot 10^{-4}$ w $\sim 0.42s$.

Błąd średniokwadratowy zwiększa się ze wzrostem odległości między węzłami (dla mniejszych n i dla większych kwadratów). Ponadto sk jest większy dla dużych co do modułu współrzędnych x, y. Dzieje się tak dlatego, że $f''_{xy}(x,y)$ szybko się zmienia dla dużych $|x|, |y|$, a w każdym trójkącie $w''_{xy}(x,y)$ jest stała. Czas wykonania programu jest proporcjonalny do n^2.



Przykład 4

$$f(x,y) = \frac{\ln(x+1)}{\ln(y+2)}$$

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 10$ program podaje $sk = 1.3273 \cdot 10^{-3}$ w $\sim 0.01s$.

Dla $Z = \begin{bmatrix} [0,10] & [10,20] \\ [10,0] & [20,10] \end{bmatrix}$, $n = 20$ program podaje $sk = 1.8196 \cdot 10^{-4}$ w $\sim 0.02s$.

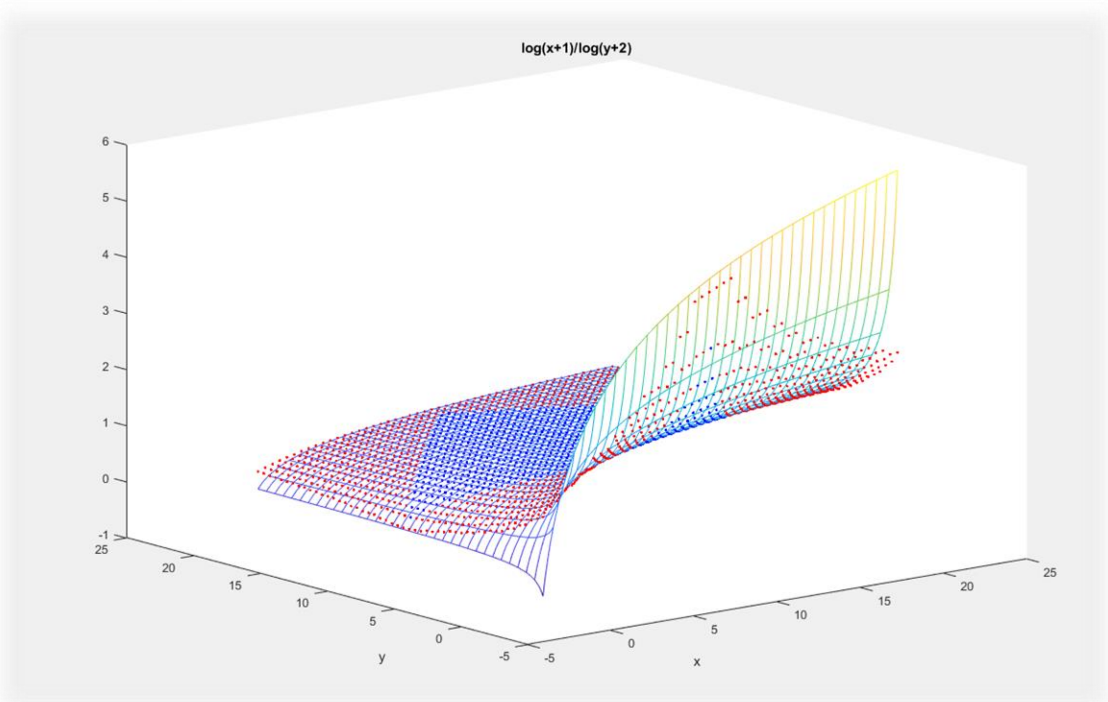
Dla $Z = \begin{bmatrix} [0,5] & [5,5] \\ [0,0] & [5,0] \end{bmatrix}$, $n = 100$ program podaje $sk = 1.1107 \cdot 10^{-6}$ w $\sim 0.45s$.

Dla $Z = \begin{bmatrix} [15,20] & [20,20] \\ [15,15] & [20,15] \end{bmatrix}$, $n = 100$ program podaje $sk = 1.5051 \cdot 10^{-10}$ w $\sim 0.45s$.

Dla $Z = \begin{bmatrix} [10^{10},20] & [10^{10}+5,20] \\ [10^{10},15] & [10^{10}+5,15] \end{bmatrix}$, $n = 100$ program podaje $sk = 1.4089 \cdot 10^{-6}$ w $\sim 1.9s$.

Dla $Z = \begin{bmatrix} [-0.5, 4.5] & [4.5, 4.5] \\ [-0.5,-0.5] & [4.5,-0.5] \end{bmatrix}$, $n = 100$ program podaje $sk = 1.0617 \cdot 10^{-5}$ w $\sim 0.45s$.

Błąd średniokwadratowy zwiększa się ze wzrostem odległości między węzłami (dla mniejszych n i dla większych kwadratów). Ponadto sk jest większy dla x lub y bliskim -1. Dzieje się tak dlatego, że f''_{xx}, f''_{xy} szybko się zmieniają dla x lub y bliskim -1, a w każdym trójkącie w''_{xx}, w''_{xy} są stałe. $f''_{yy} = \frac{\ln(x+1)\cdot(\ln(y+2)+2)}{(y+2)^2\cdot(\ln(y+2))^3}$ szybko się zmienia nie tylko dla y bliskim $-1, -2$, ale też dla $x \gg y$, więc sk jest większy także w tym przypadku. Czas wykonania programu jest proporcjonalny do n^2. Przy $x \gg y$ czas wykonania programu znacznie się wydłużył. Jest to spowodowane osobliwością macierzy A w funkcji $makeM$ przy używanej precyzji obliczeń. W takim przypadku korzystamy z metody $pinv$, która jest wolniejsza.



Przykład 5

$$f(x, y) = \frac{x}{e^{x^2+y^2}}$$

Przykład używa zmiennej n(domyślnie 10). Dla $Z = \begin{bmatrix} [-2.5, 2.5] & [2.5, 2.5] \\ [-2.5, -2.5] & [2.5, -2.5] \end{bmatrix}, \begin{bmatrix} [-2, 0] & [0, 2] \\ [0, -2] & [2, 0] \end{bmatrix}, \begin{bmatrix} [-3, 0] & [0, 3] \\ [0, -3] & [3, 0] \end{bmatrix}$ tworzy macierz M i wywołuje funkcję $draw(Z, M, f)$. Przykład służy graficznemu przedstawieniu działania programu.

$n = 1$ $\qquad\qquad$ $n = 8$