Sprawozdanie Algorytmy Geometryczne Ćwiczenie 1

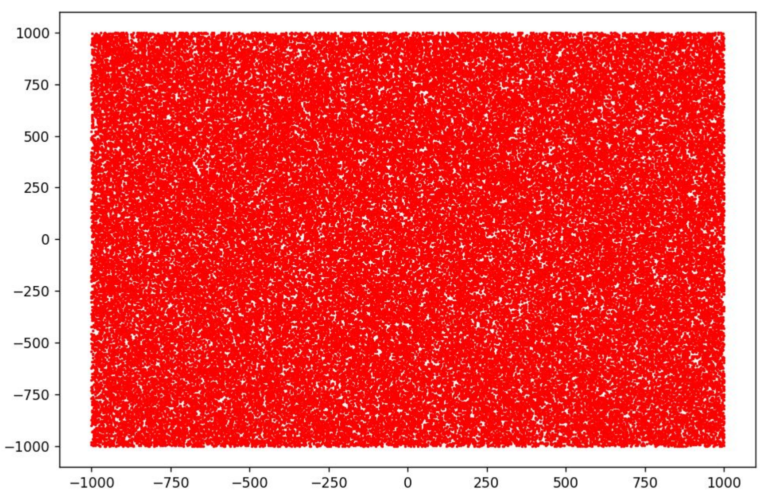
25.10.2020r. Kraków
Norbert Wolniak
Grupa Czw_16.15_A

1. Cel ćwiczenia:
   Celem ćwiczenia jest zaimplementowanie podstawowych predykatów geometrycznych takich jak sprawdzenie orientacji punktu w zależności od wektora w przestrzeni dwuwymiarowej oraz porównanie wyników dla własnej implementacji wyznacznika do tych dostarczanych przez biblioteki.

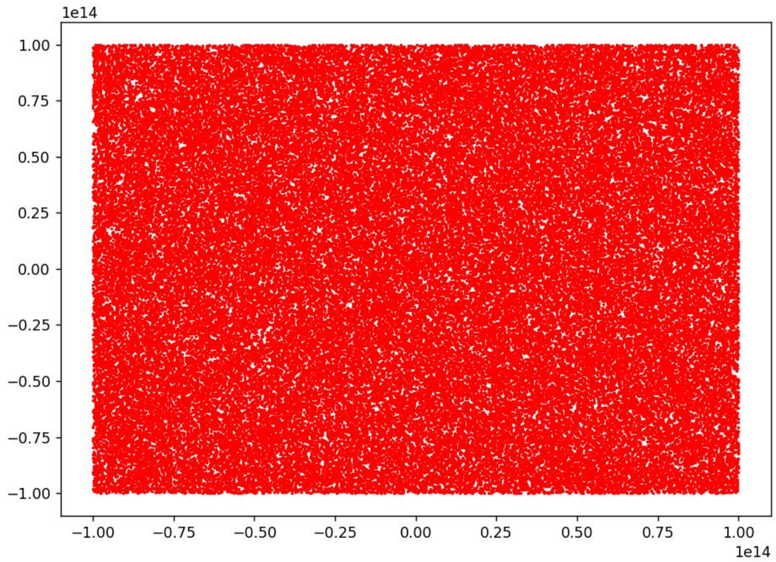
2. Uwagi techniczne:
   Ćwiczenie zostało przeprowadzone w oprogramowaniu Jupyter Notebook, na systemie Windows 10 Home, komputer wyposażony w procesor x64 przy użyciu proponowanego narzędzia graficznego, które napisane jest w języku Python3 i wykorzystuje biblioteki MatPlotLib oraz NumPy.

3. Program ćwiczenia:
   Przygotowane zbiory to :
   A - 10^5 losowych punktów o współrzędnych z przedziału [-1000, 1000],

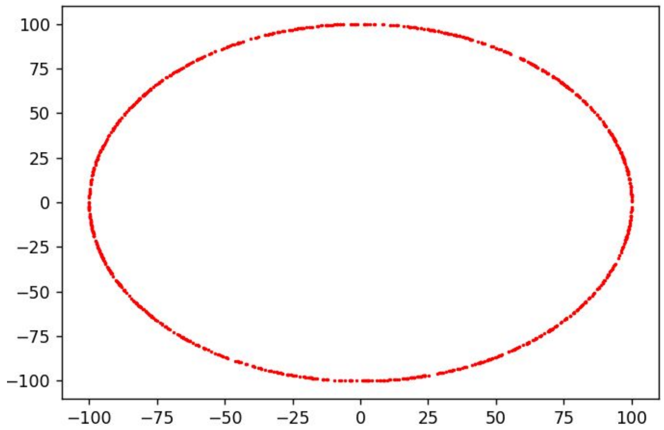
Rysunek 1A



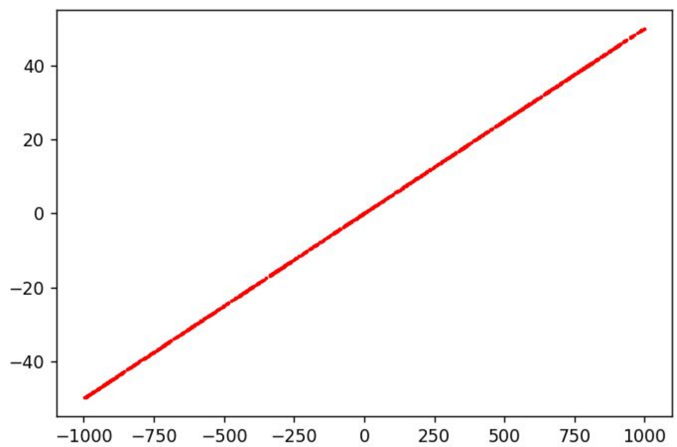
B - 10^5 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$,
Rysunek 1B



C - 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu R=100,
Rysunek 1C



D - 1000 losowych punktów o współrzędnych z przedziału [-1000, 1000] leżących na prostej wyznaczonej przez wektor (a, b), a = [-1.0, 0.0], b = [1.0, 0.1].
Rysunek 1D



Użyte metody i funkcje:
Dostarczone przez narzędzie graficzne oraz:
np.random.uniform(from,to) - losowa liczba rzeczywista z przedziału,
np.linalg.det(matrix) - liczy wyznacznik macierzy, z biblioteki NumPy,
det3x3(p,q,r) i det2x2(p,q,r) - zaimplementowane wyznaczniki
orientation(points,vector,det_,zeroTolerance = 10**(-13)) - zwraca tablice punktów lężących na lewo, prawo i będących współliniowych do wektora. Domyślna tolerancja dla zera to 10^{-13}.
plotPoints(arrayOfPoints) - rysuje punkty przy użyciu narzędzia graficznego, 'arrayOfPoints' to wynikowa tablica orientation().
differences(array,vector,det1,det2) - rysuje różnicę zbiorów punktów (left,right,collinear) po określeniu orientacji względem wektora, wypisuje liczbę tych różnic.

4. Wyniki:
Dla każdego zbioru zostały przeprowadzone podziały punktów względem ich orientacji do wektora (a,b).

Rodzaje wyznaczników : 3x3 oraz 2x2.

$$\det(a,b,c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} \qquad \det(a,b,c) = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$$

Wyznaczniki zaimplementowane samodzielnie oraz pobrane z biblioteki miały różną precyzję obliczeń co skutkowało innym przyporządkowaniem punktów, gdy wartość wyznacznika była bliska zeru dla różnych tolerancji zera. Przetestowane tolerancje dla zera [-10^e:10^e] dla e= -5...-20.

Poniższa tabela prezentuje liczbę różnic w przyporządkowaniu punktów dla każdego ze zbiorów. Porównywane są wyznaczniki zaimplementowane samodzielnie z tymi pobranymi z biblioteki NumPy.

Tabela nr. 1

Współczynnik e	A 3x3	A 2x2	B 3x3	B 2x2	C 3x3	C 2x2	D 3x3	D 2x2
-6	0	0	0	5	0	0	0	0
-7	0	0	0	5	0	0	0	0
-8	0	0	0	5	0	0	0	0
-9	0	0	0	5	0	0	0	0
-10	0	0	0	5	0	0	0	0
-11	0	0	0	5	0	0	0	0
-12	0	0	0	5	0	0	0	282
-13	0	0	0	5	0	0	0	580
-14	0	0	0	5	0	0	131	683
-15	0	0	0	5	0	0	441	724
-16	0	0	0	5	0	0	529	738
-17	0	0	0	5	0	0	549	744
-18	0	0	0	5	0	0	553	744
-19	0	0	0	5	0	0	554	745
-20	0	0	0	5	0	0	554	745

Powyższa tabela nr1. uwidacznia różnice w działaniu różnych wyznaczników. Zbiory które okazały się być problematyczne to:

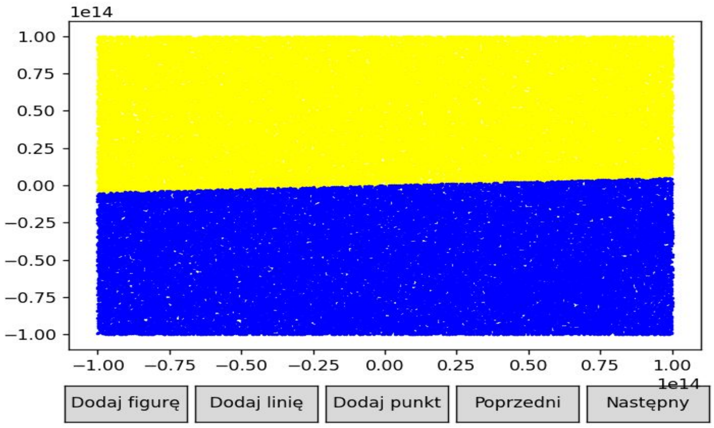
zbiór B - rysunek 1B,

zbiór D - rysunek 1D

Dalsze porównania zastosuję już tylko dla tych dwóch zbiorów.

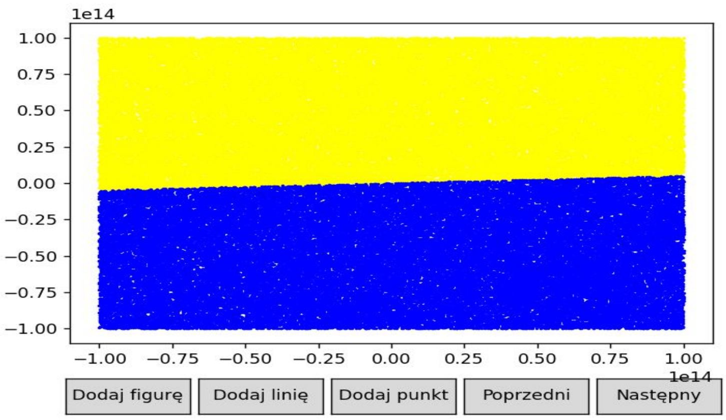
Poniższe rysunki prezentują orientację punktów dla zbiorów B i D:

Rysunek 2Bwyznacznik3x3



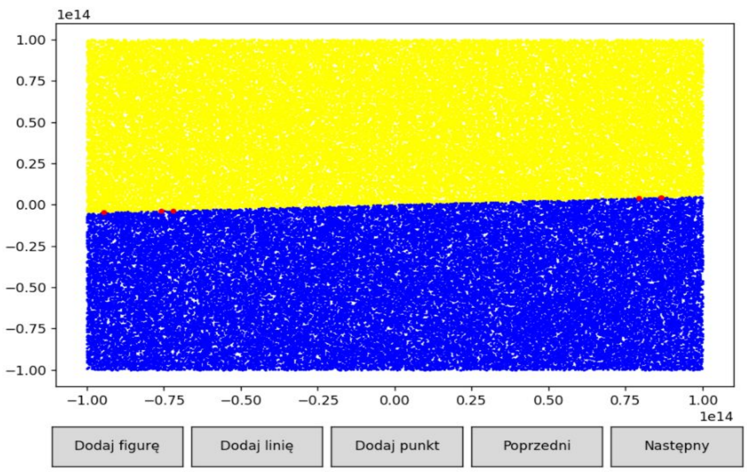

```
Left:  50029
Right:  49971
Collinear:  0
```

Rysunek 2BwyznacznikNP3x3



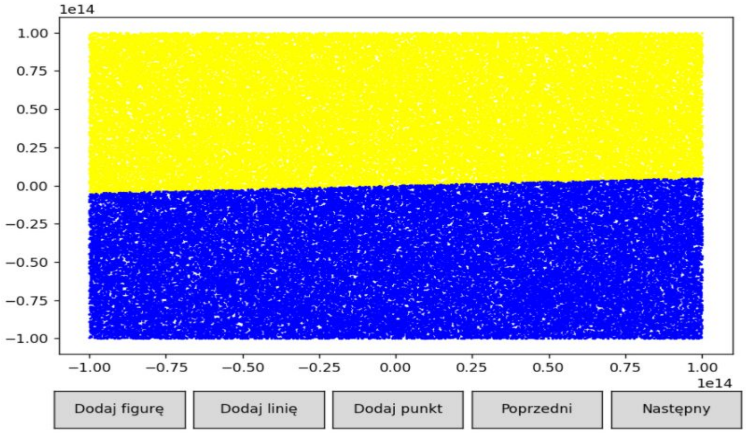

```
Left:  50029
Right:  49971
Collinear:  0
```

Rysunek 2Bwyznacznik2x2



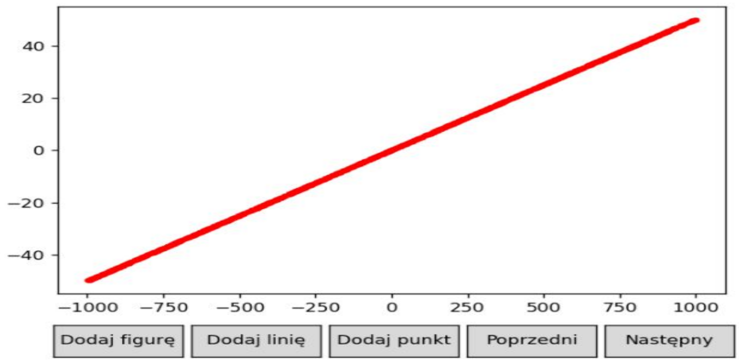

```
Left:  50026
Right:  49969
Collinear:  5
```

Rysunek 2BwyznacznikNP2x2



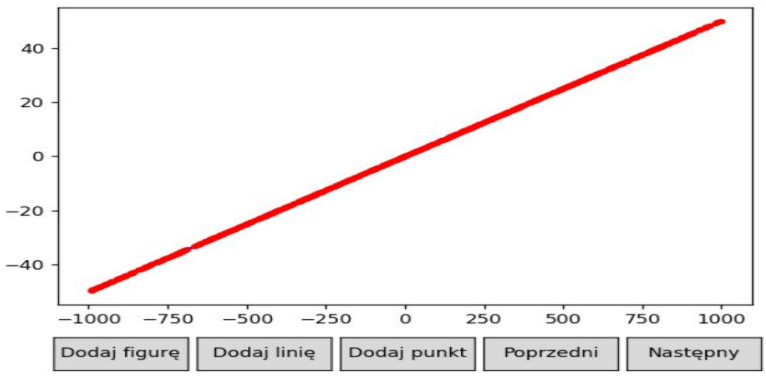

```
Left:  50028
Right:  49972
Collinear:  0
```

Rysunek 2Dwyznacznik3x3



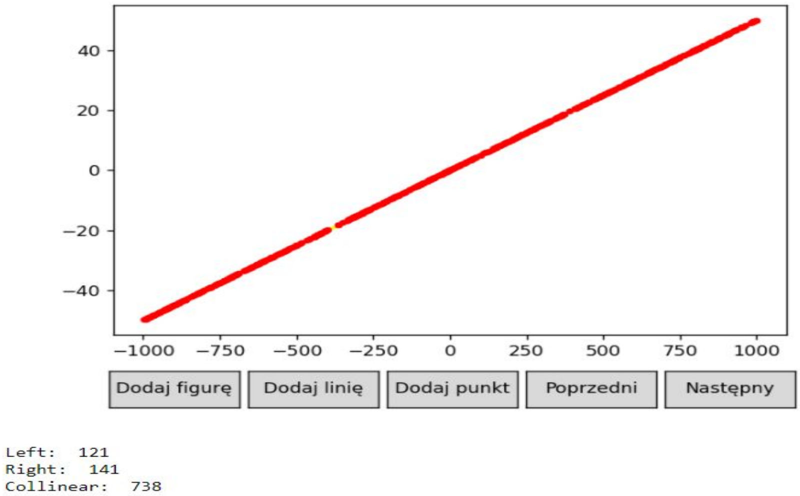

```
Left:  0
Right:  0
Collinear:  1000
```

Rysunek 2DwyznacznikNP3x3

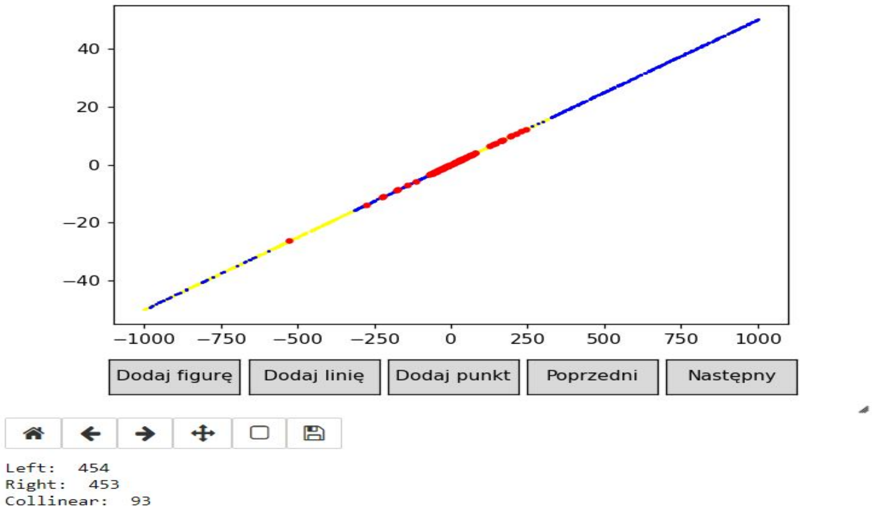



```
Left:  18
Right:  113
Collinear:  869
```

Rysunek 2Dwyznacznik2x2



Rysunek 2DwyznacznikNP2x2



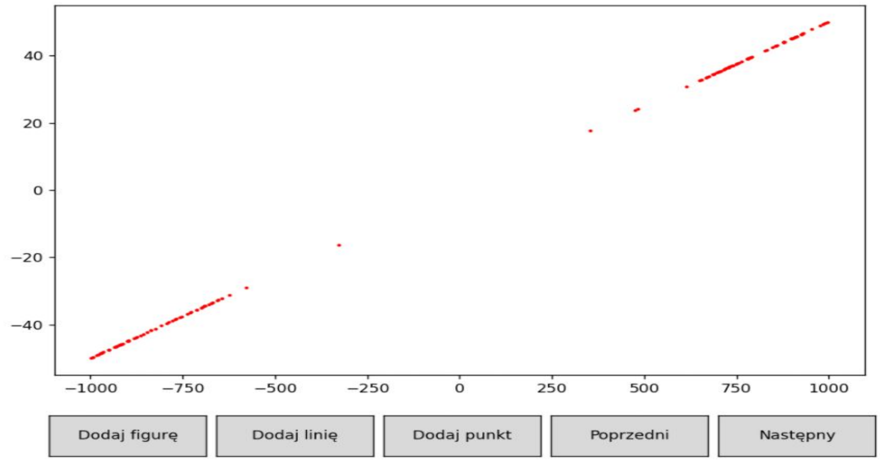
Powyższe porównania zostały przeprowadzone dla tolerancji dla zera o współczynniku e = -14.

5. Wnioski:

Porównując wyniki szczególnie dla zbioru D, gdzie różnice w tolerancji dla zera miały największe znaczenie z powodu punktów współliniowych do a i b dochodzę do wniosku, że wyznaczniki zaimplementowane własnoręcznie mają lepszą precyzję obliczeń. Z wyznaczników 3x3 (rysunek 2Dwyznacznik3x3) i 2x2 (rysunek 2Dwyznacznik2x2) własnych lepszym wyborem okazał się wyznacznik 3x3 w którym granica błędu jest znacznie mniejsza od tej w wyznaczniku 2x2 zarówno tych implementowanych ręcznie jak i pobranych z biblioteki (rysunki 2DwyznacznikNP3x3 i 2DwyznacznikNP2x2). Wpływ na to może mieć postać wyznacznika 2x2 obliczanego z różnic współrzędnych tych punktów.

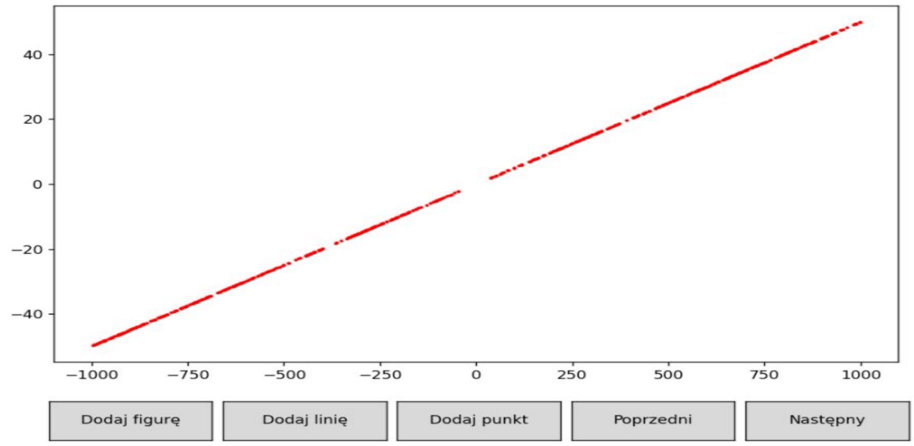
Poniższe rysunki prezentują różnicę w przydzieleniu punktów odpowiednio dla wyznacznika 3x3 i wyznacznika NP 3x3 (rysunek 3D3x3) oraz dla wyznacznika 2x2 i wyznacznika NP 2x2 (rysunek 3D2x2)

Rysunek 3D3x3



Różnica w wyniku : 131

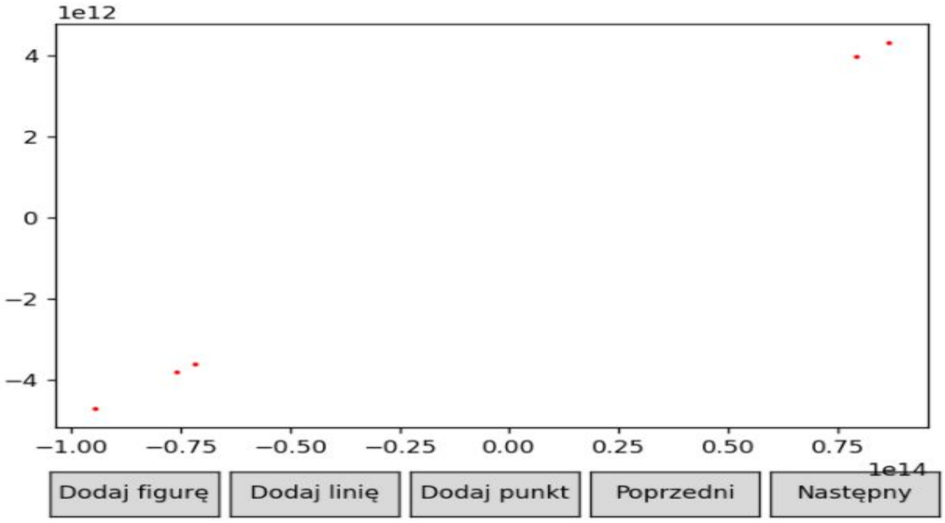
Rysunek 3D2x2



Różnica w wyniku : 683

W zbiorze B z powodu zakresu danych rzędu 10^{14} i stosunkowo małych współrzędnych punktów a i b (wektor ab) w wyznaczniku 2x2 (rysunek 2Bwyznacznik2x2) może wystąpić różnica liczb rzędu 10^{28} której wynik jest mało precyzyjny i może wynieść nieprawdziwe 0.0 ponieważ precyzja obliczeń nie pozwala na dokonanie dokładnej różnicy tych liczb na pozycjach bliskiej zeru co czasem prowadzi do błędnego przydzielenia punktów co widać w tabeli (tabela nr1) w kolumnie B 2x2 i na rysunku (rysunek 2Bwyznacznik2x2).

Poniższy rysunek prezentuje punkty które są źle przydzielane w zbiorze B dla wyznacznika 2x2.




```
Różnica w wyniku : 5
```

Konkluzja:

Własnoręcznie zaimplementowany wyznacznik 3x3 okazał się najlepszy uwzględniając że tolerancja dla zera wynosi 10^{-14}.