

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

ALGORYTMY GEOMETRYCZNE

---

## Ćwiczenie 1

---

*Autor:*

Andrzej Zaborniak  
Informatyka, rok II gr. 4

27.10.2022 r.

---

# 1 Specyfikacja techniczna

Parametry techniczne komputera na którym zostało wykonane ćwiczenie:

Procesor: Intel® Core™ i7-5600U CPU @ 2.60GHz × 4

Karta graficzna: Mesa Intel® HD Graphics 5500 (BDW GT2)

Pamięć RAM: 8,0 GB

System operacyjny: Ubuntu 22.04.1 LTS

Wersja GNOME: 42.4

Użyty język programowania: Python 3.10.6

Wykorzystany program: Jupyter Notebook

## 1.1 Narzędzie graficzne

Wykresy wizualizujące rozmieszczenie punktów na płaszczyźnie oraz tabele zawierające informacje na temat położenia punktów względem prostej  $(a,b)$  zostały wykonane przy użyciu funkcji zawartych w bibliotekach Seaborn oraz Pandas.

## 2 Cel ćwiczenia

Ćwiczenie miało na celu wprowadzić w zagadnienia geometrii obliczeniowej. Głównym zadaniem była implementacja podstawowych predykatów geometrycznych, przeprowadzenie testów oraz wizualizacja. Ćwiczenie polegało na określeniu dla wybranych punktów  $c$  położenie względem prostej  $(a,b)$ . Położenie to można określić obliczając wartość jednego z dwóch wyznaczników.

Wyznacznik macierzy  $3 \times 3$ :

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

Wyznacznik macierzy  $2 \times 2$ :

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

Wartości wyznaczników macierzy  $2 \times 2$  i  $3 \times 3$  zostały policzone za pomocą ręcznej implementacji oraz korzystając z wbudowanej funkcji np.linalg.det(matrix) przyjmującej jako argument macierz zapisaną w postaci tablicy np.Array. Funkcja ta jest w bibliotece numerycznej Numpy.

W zależności od wartości można określić położenie wyznaczonego punktu:

$$\begin{aligned} < 0 &\text{ punkt } c \text{ znajduje się po prawej stronie prostej } ab \\ \det(a, b, c) = &> 0 \text{ punkt } c \text{ znajduje się po lewej stronie prostej } ab \\ &= 0 \text{ punkt } c \text{ znajduje się na prostej } ab \end{aligned}$$

## 3 Treść zadania

### 3.1 Zbiory punktów

Na samym początku należało wygenerować dane zbiory punktów których współrzędne są typu zmiennoprzecinkowego a następnie przedstawić ich graficzną reprezentację.

Zbiory punktów które należało przedstawić:

- a)  $10^5$  losowych punktów o współrzędnych z przedziału [-1000, 1000]

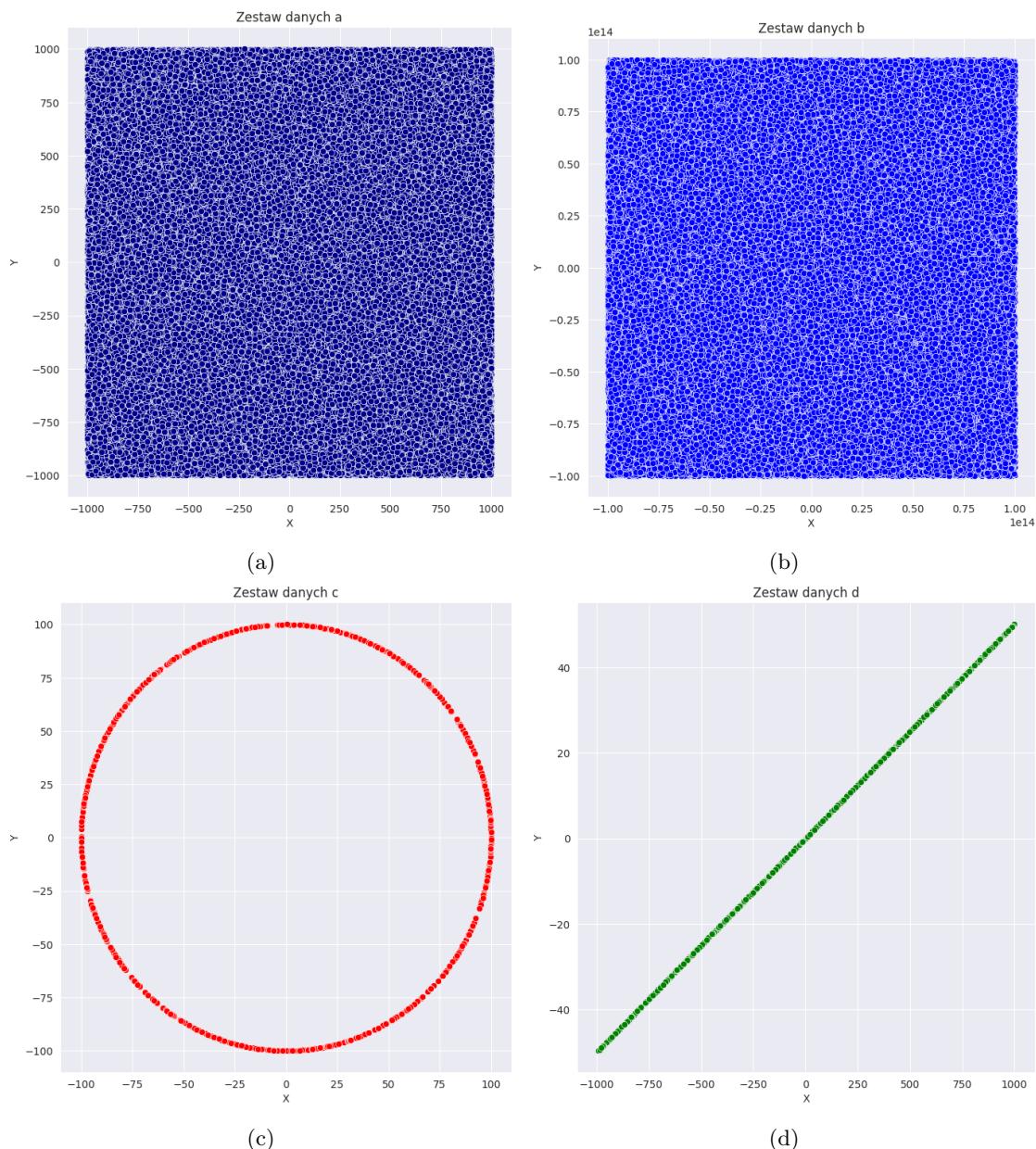
- 
- b)  $10^5$  losowych punktów o współrzędnych z przedziału  $[-10^{14}, 10^{14}]$   
c) 1000 losowych punktów leżących na okręgu o środku  $(0,0)$  i promieniu  $R=100$   
d) 1000 losowych punktów o współrzędnych z przedziału  $[-1000, 1000]$  leżących na prostej wyznaczonej przez wektor  $(a, b)$  przyjmijmy  $a = [-1.0, 0.0]$ ,  $b = [1.0, 0.1]$ .

### 3.2 Tolerancja

Każdy zbiór punktów był badany w określonych tolerancjach:

- $Epsilon = 10^{-10}$
- $Epsilon = 10^{-14}$
- $Epsilon = 10^{-18}$
- $Epsilon = 10^{-20}$

#### 3.2.1 Graficzna reprezentacja zbiorów



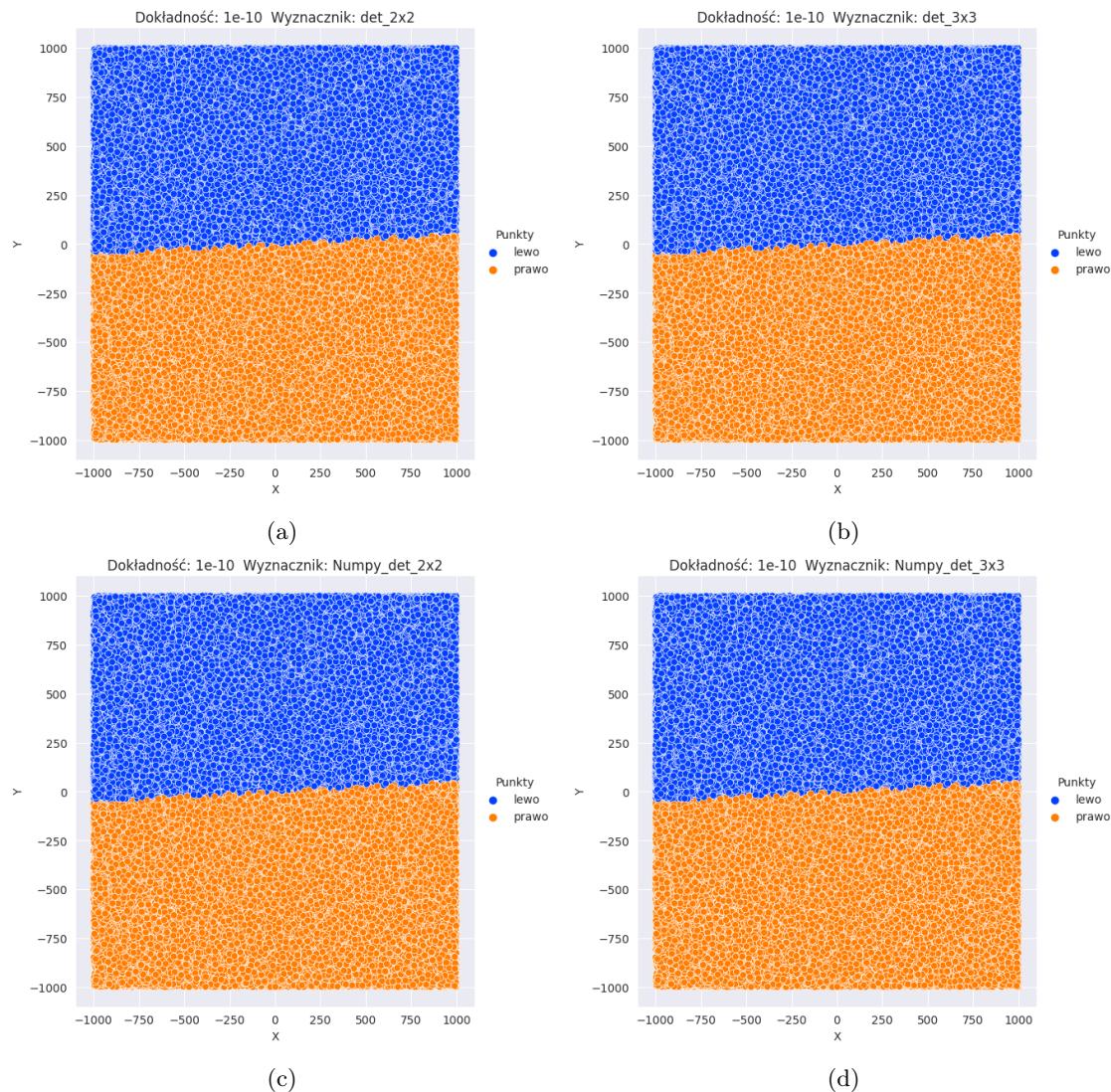
Rysunek 1

## 4 Zbiór punktów a)

Graficzna wizualizacja wraz z tabelami przedstawia rozmieszczenie punktów względem prostej  $ab = [-1.0, 0.0], b = [1.0, 0.1]$ . Każdy wykres zawiera tytuł w którym zostały zawarte informacje takie jak: tolerancja z jaką punkty były klasyfikowane oraz sposób liczenia wyznacznika, odpowiednio:

- det\_2x2, det\_3x3 - oznacza własną implementację obliczania wyznacznika 2x2 i 3x3
- Numpy\_det\_2x2, Numpy\_det\_3x3 - oznacza skorzystanie z funkcji zawartych w bibliotece Numpy do obliczania wyznacznika.

### 4.1 Tolerancja $e = 10^{-10}$



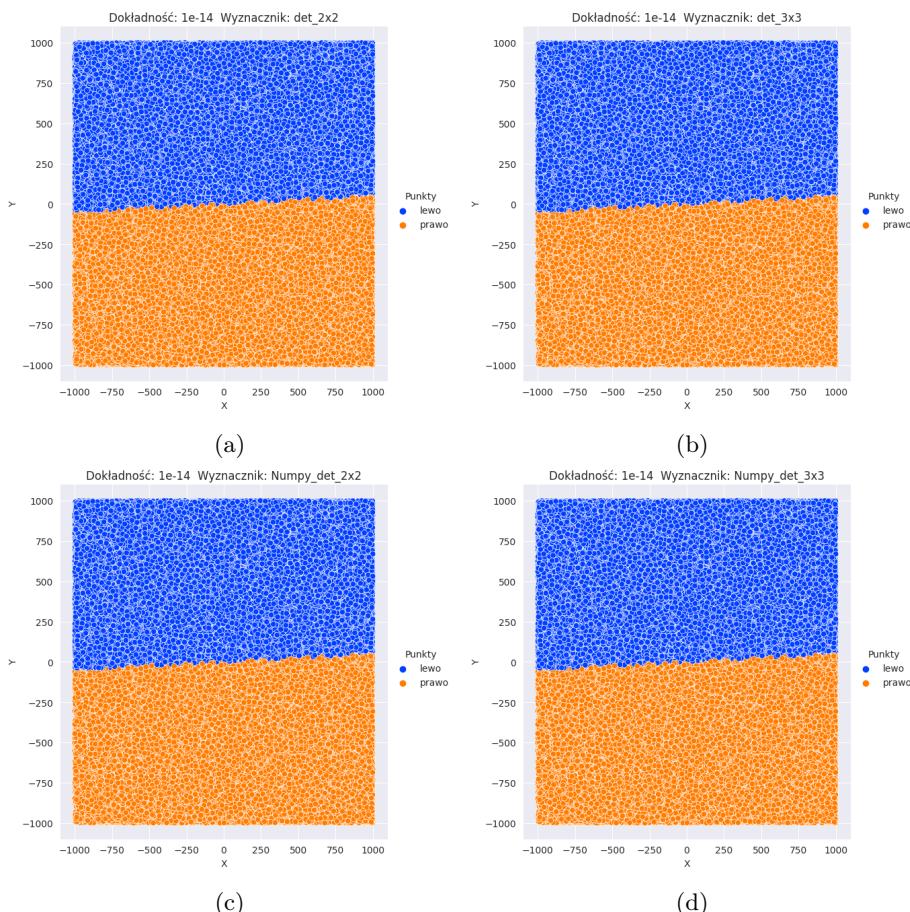
Rysunek 2

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	49881	49881	49881	49881
prawo	50119	50119	50119	50119
środek	0	0	0	0

zbiór A, tolerancja  $10^{-10}$

Tabela 1

#### 4.2 Tolerancja $e = 10^{-14}$



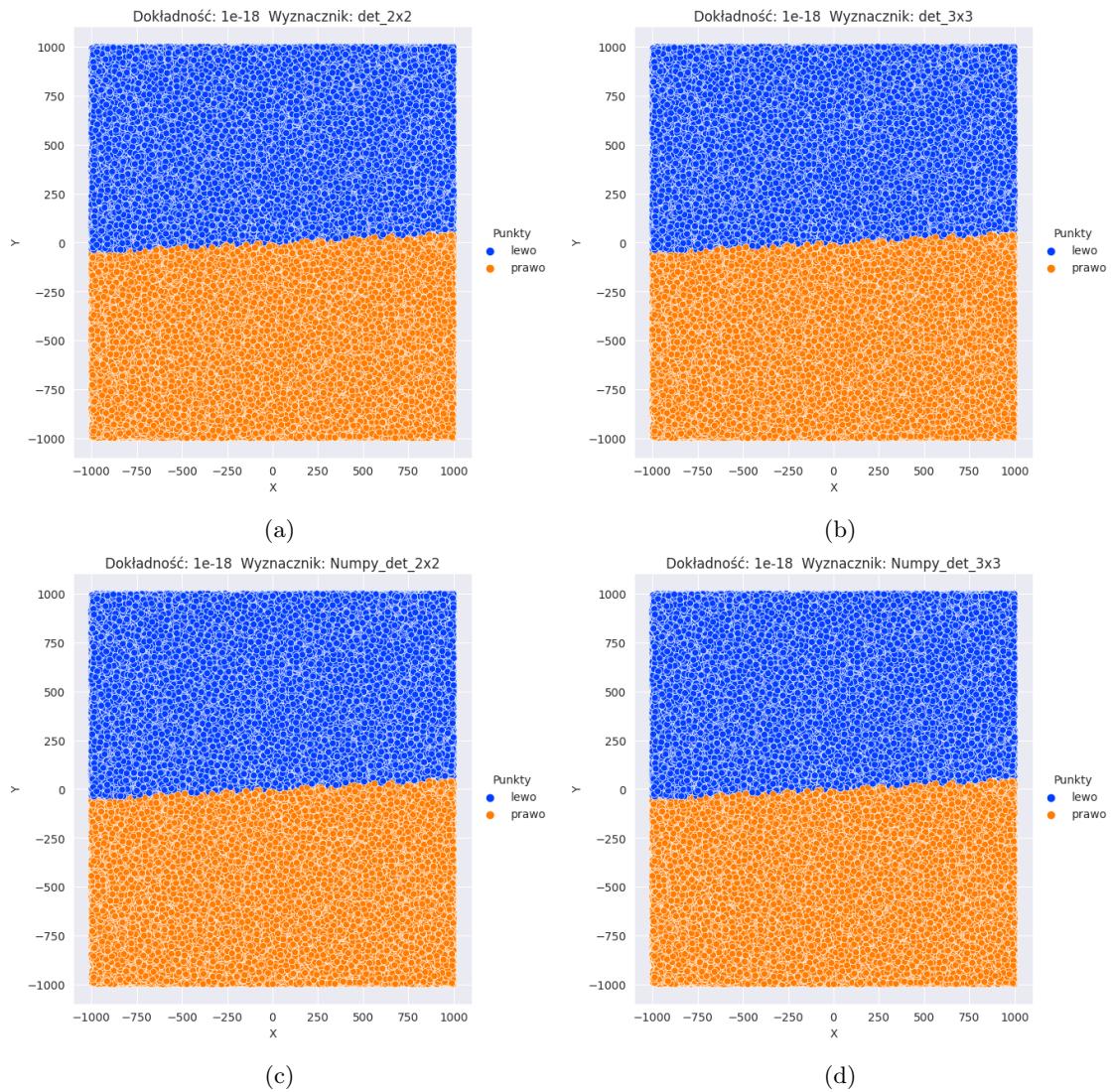
Rysunek 3

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	49881	49881	49881	49881
prawo	50119	50119	50119	50119
środek	0	0	0	0

zbiór A, tolerancja  $10^{-14}$

Tabela 2

### 4.3 Tolerancja $e = 10^{-18}$

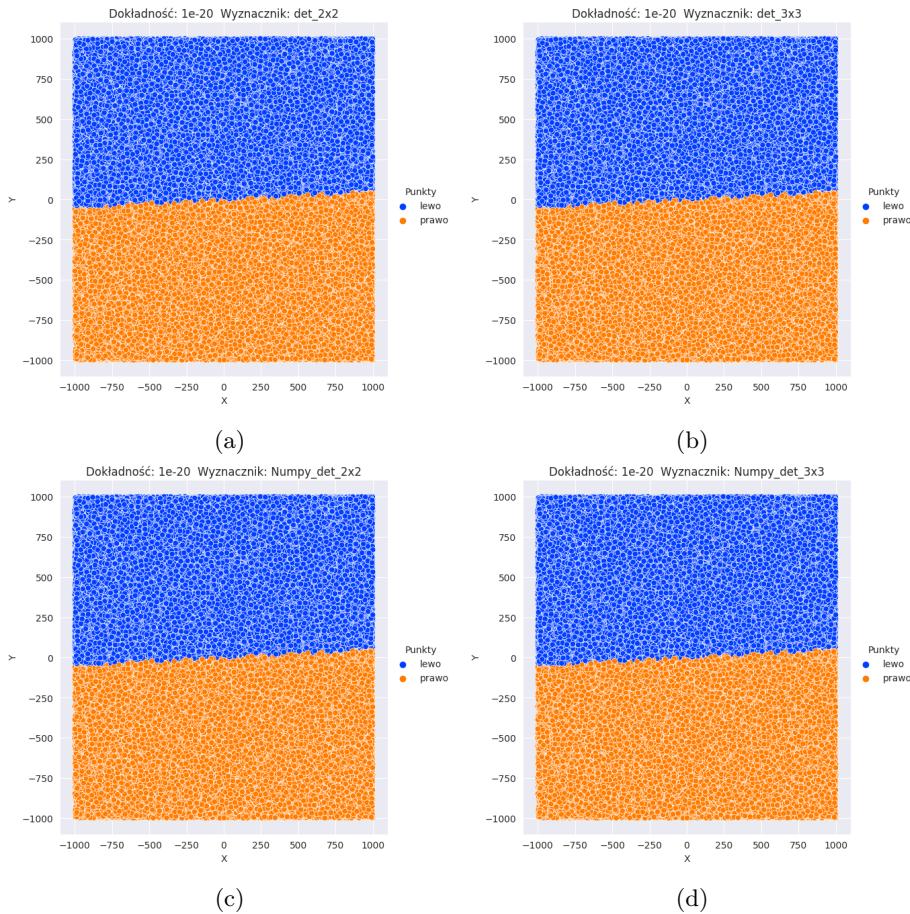


Rysunek 4

	<code>det_2x2</code>	<code>det_3x3</code>	<code>Numpy_det_2x2</code>	<code>Numpy_det_3x3</code>
<b>Punkty</b>				
lewo	49881	49881	49881	49881
prawo	50119	50119	50119	50119
środek	0	0	0	0

zbiór A, tolerancja  $10^{-18}$   
Tabela 3

#### 4.4 Tolerancja $e = 10^{-20}$



Rysunek 5

	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	49881	49881	49881	49881
prawo	50119	50119	50119	50119
środek	0	0	0	0

zbiór A, tolerancja  $10^{-20}$

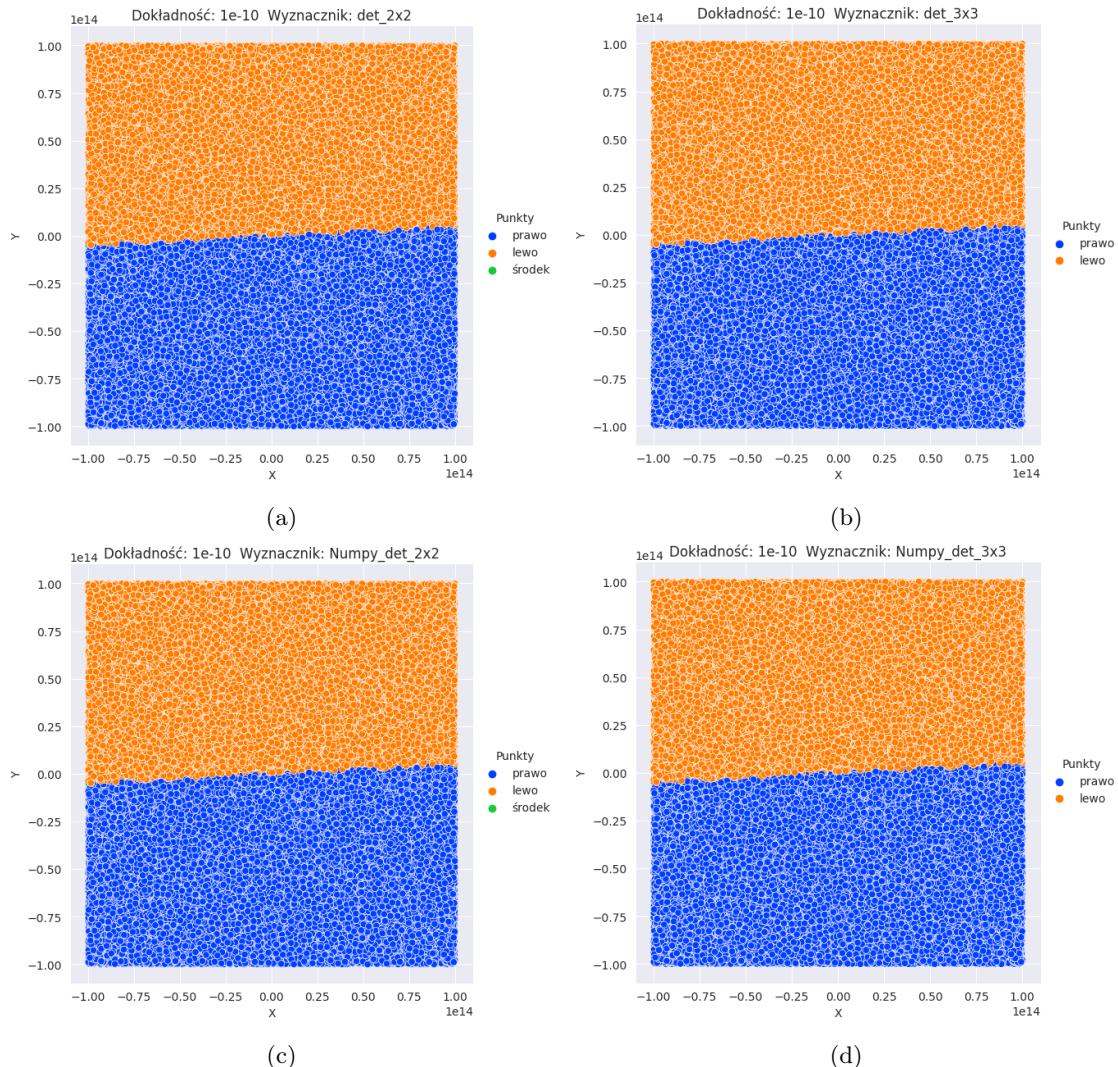
Tabela 4

#### 4.5 Wnioski dla zbioru a

Jak możemy zauważyć z wizualizacji dla pierwszego zbioru punktów, Rysunki o numerach 2,3,4 oraz 5 nie różnią się między sobą tak samo jak tabele 1,2,3 i 4, pomimo że zostały zastosowane różne sposoby liczenia wyznacznika. Wartości tolerancji również nie wpłynęły na podział punktów względem odcinka  $ab$ .

## 5 Zbiór punktów b)

### 5.1 Tolerancja $e = 10^{-10}$



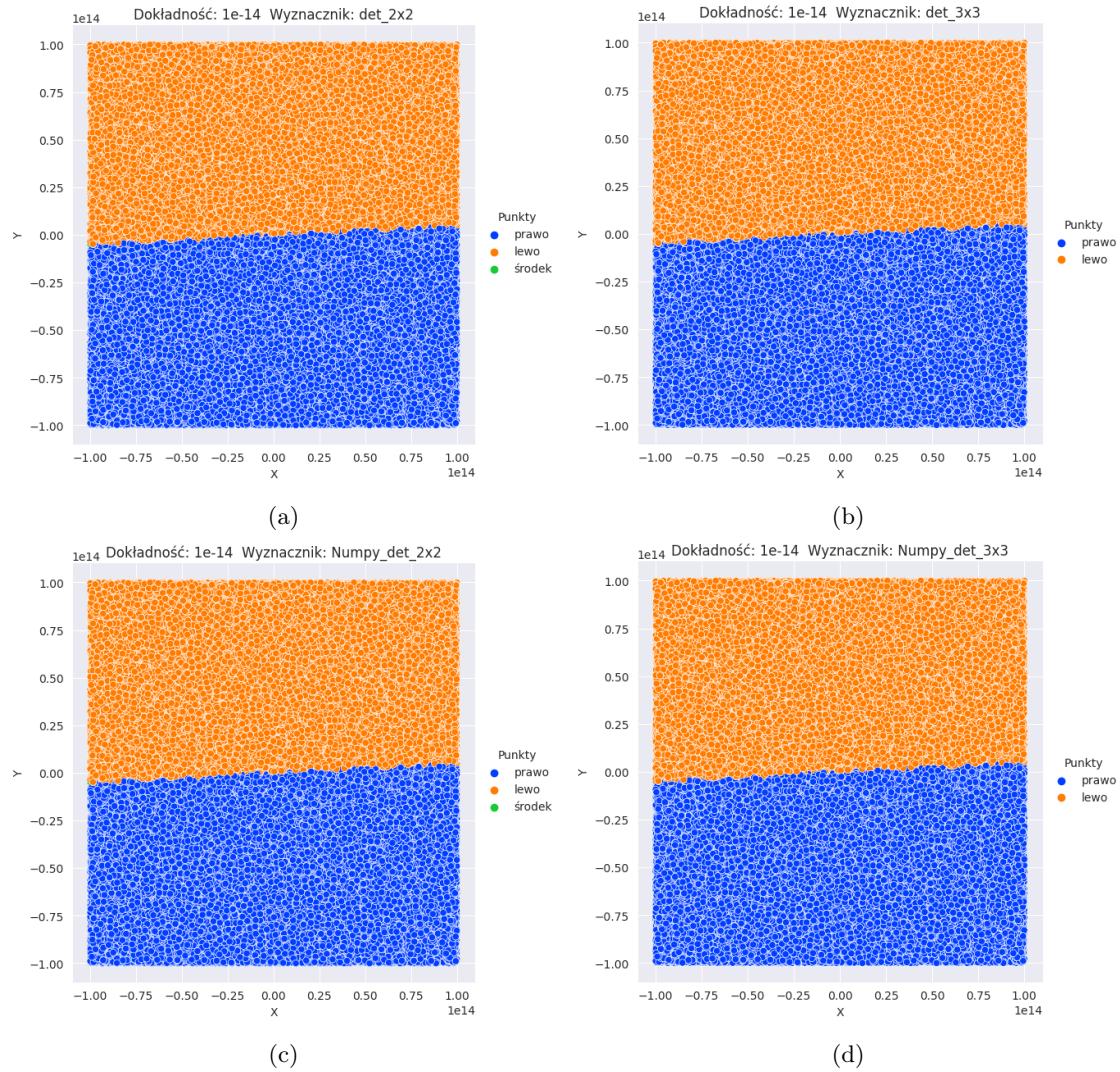
Rysunek 6

	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	49808	49810	49807	49810
prawo	50188	50190	50187	50190
środek	4	0	6	0

zbiór B, tolerancja  $10^{-10}$

Tabela 5

## 5.2 Tolerancja $e = 10^{-14}$



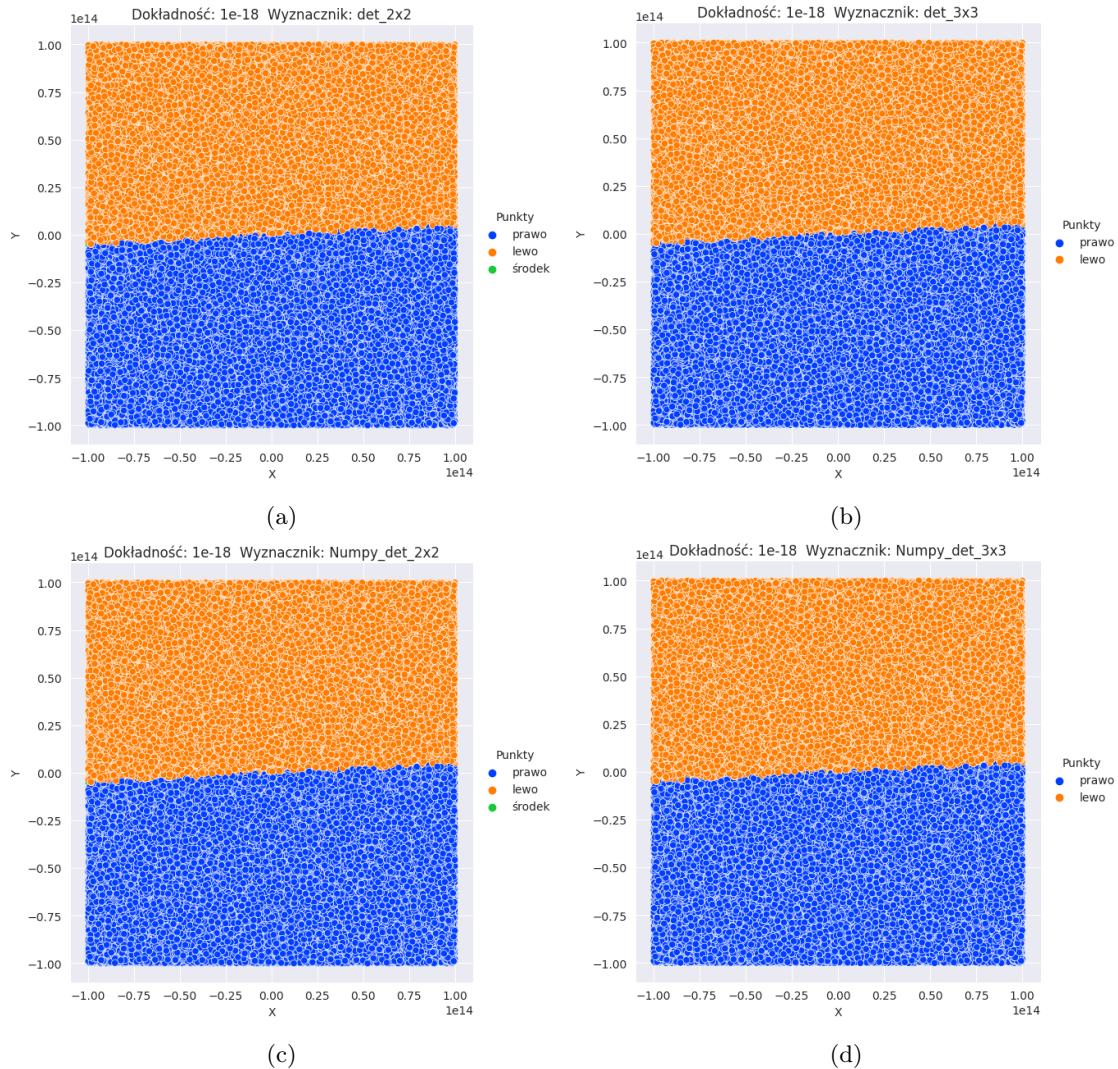
Rysunek 7

	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	49808	49810	49807	49810
prawo	50188	50190	50187	50190
środek	4	0	6	0

zbiór B, tolerancja  $10^{-14}$

Tabela 6

### 5.3 Tolerancja $e = 10^{-18}$



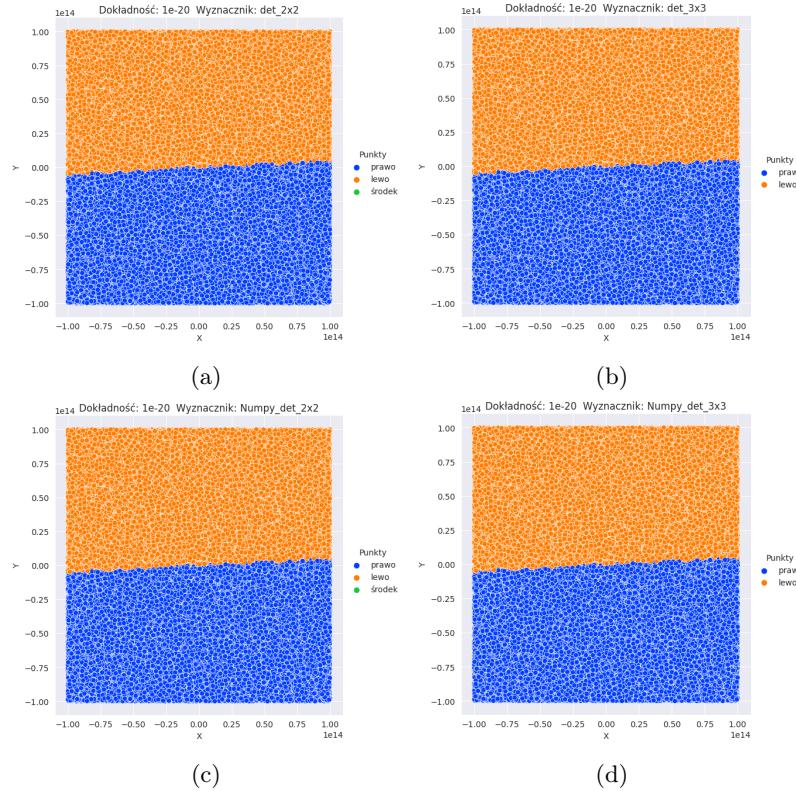
Rysunek 8

	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	49808	49810	49807	49810
prawo	50188	50190	50187	50190
środek	4	0	6	0

zbiór B, tolerancja  $10^{-18}$

Tabela 7

## 5.4 Tolerancja $e = 10^{-20}$



Rysunek 9

	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	49808	49810	49807	49810
prawo	50188	50190	50187	50190
środek	4	0	6	0

zbiór B, tolerancja  $10^{-20}$

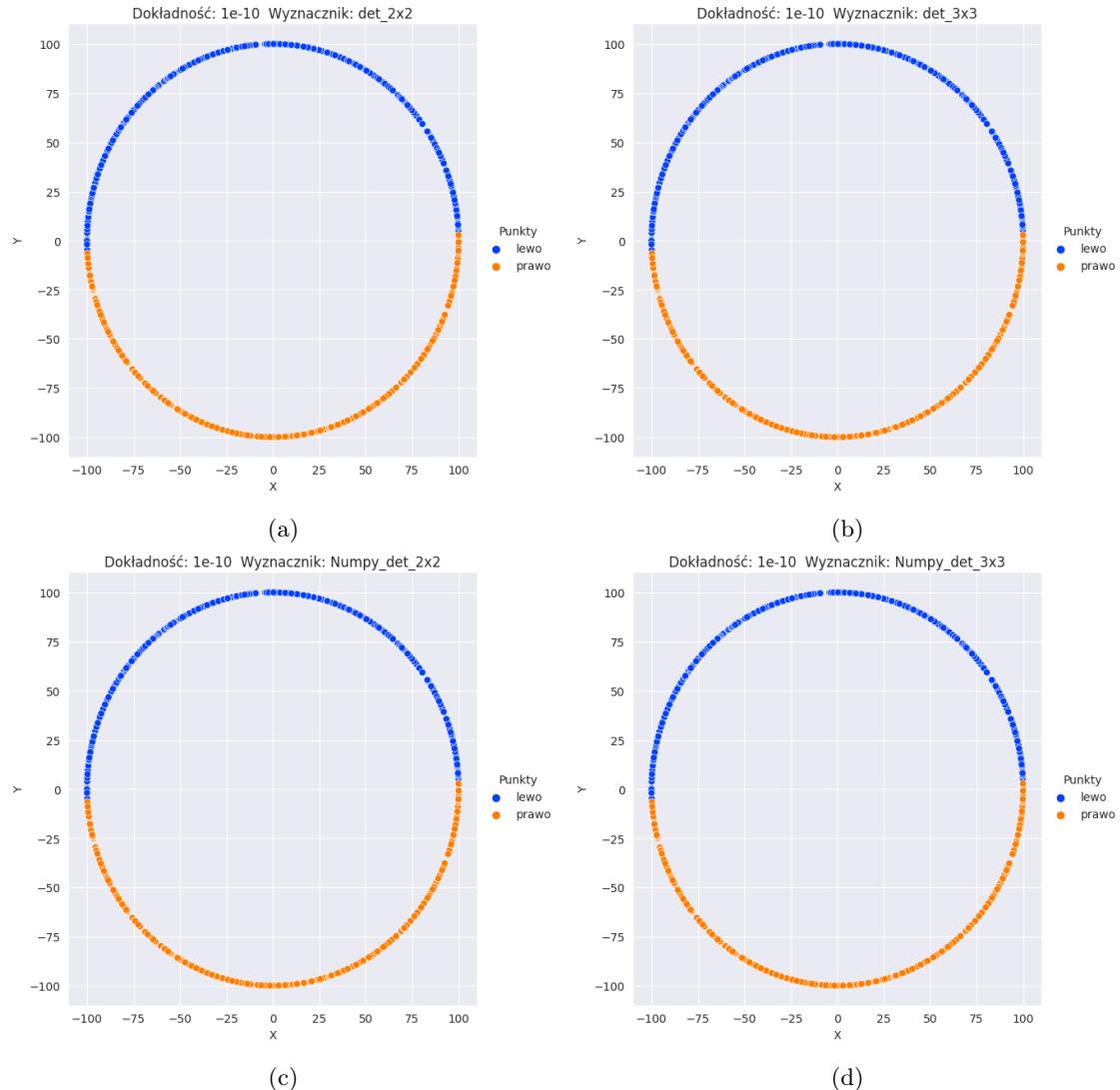
Tabela 8

## 5.5 Wnioski dla zbioru b

Dla drugiego zbioru punktów wartość tolerancji nie wpływała na wynik, zaś korzystanie z różnych metod obliczania wyznacznika pozwoliło nam dostrzec różnice w położeniu punktów. W tym przypadku użycie wyznacznika 3x3 wydaje się bardziej efektywne, ponieważ wyznacznik 2x2 zaliczył punkty na prostej, które tak na prawdę się na niej nie znajdują.

## 6 Zbiór punktów c)

### 6.1 Tolerancja $e = 10^{-10}$



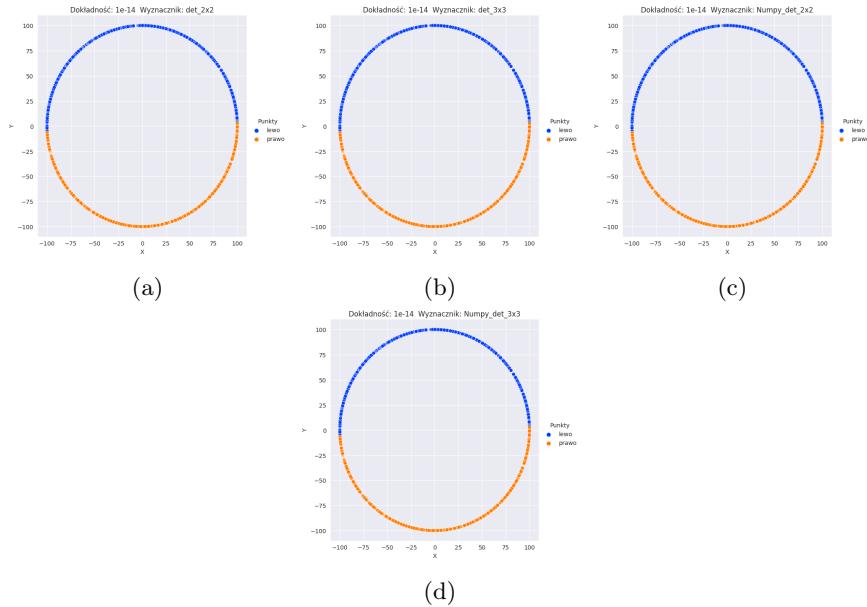
Rysunek 10

	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	502	502	502	502
prawo	498	498	498	498
środek	0	0	0	0

zbiór C, tolerancja  $10^{-10}$

Tabela 9

## 6.2 Tolerancja $e = 10^{-14}$

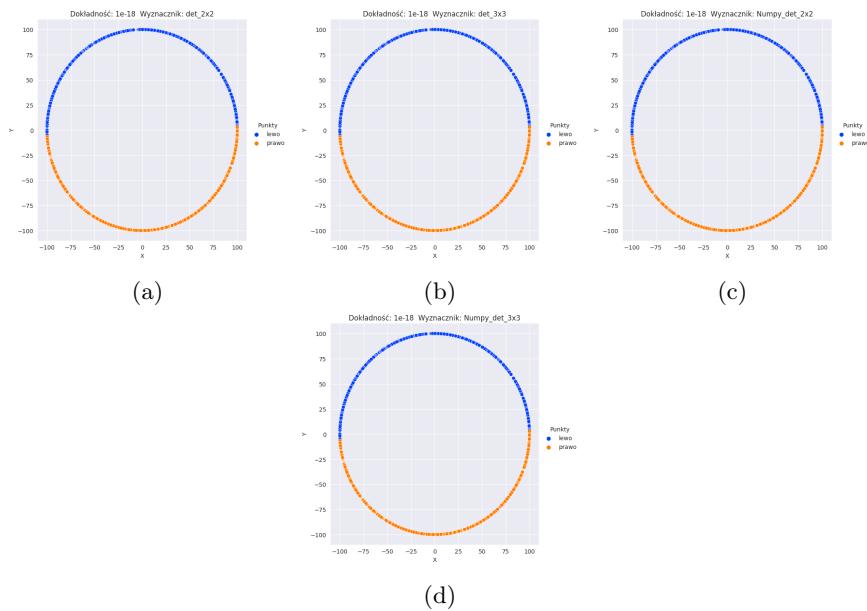


Rysunek 11

	<code>det_2x2</code>	<code>det_3x3</code>	<code>Numpy_det_2x2</code>	<code>Numpy_det_3x3</code>
<b>Punkty</b>				
lewo	502	502	502	502
prawo	498	498	498	498
środek	0	0	0	0

zbiór C, tolerancja  $10^{-14}$   
Tabela 10

## 6.3 Tolerancja $e = 10^{-18}$



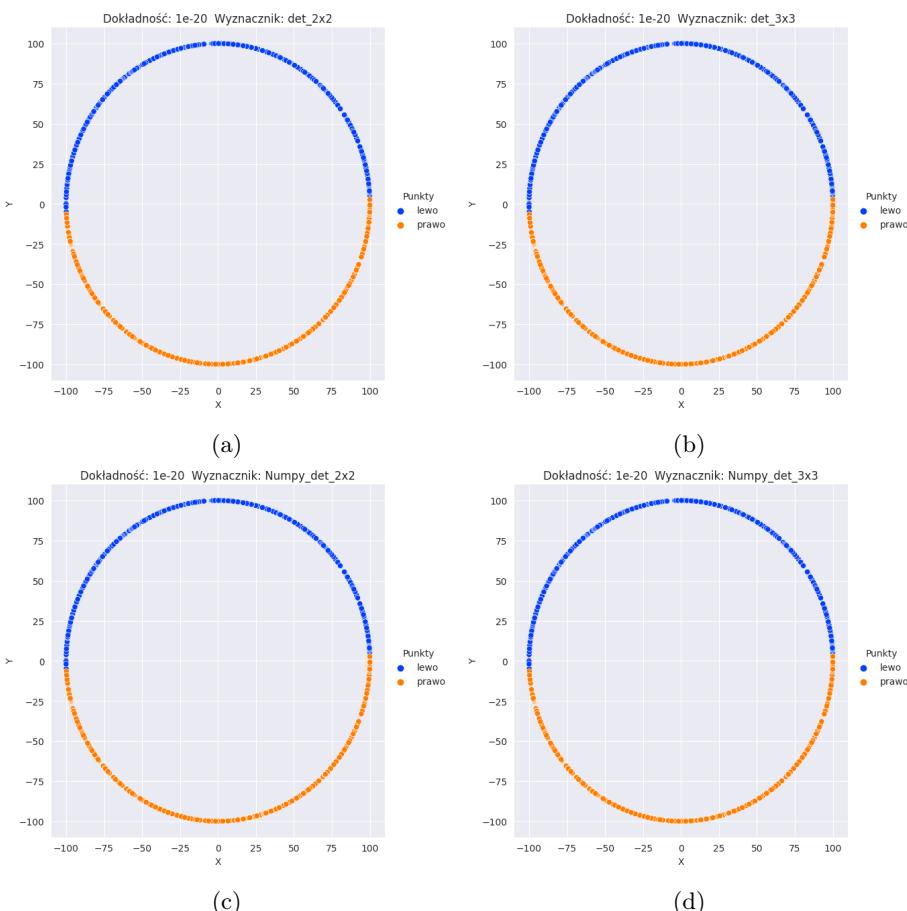
Rysunek 12

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	502	502	502	502
prawo	498	498	498	498
środek	0	0	0	0

zbiór C, tolerancja  $10^{-18}$

Tabela 11

## 6.4 Tolerancja $e = 10^{-20}$



Rysunek 13

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	502	502	502	502
prawo	498	498	498	498
środek	0	0	0	0

zbiór C, tolerancja  $10^{-20}$

Tabela 12

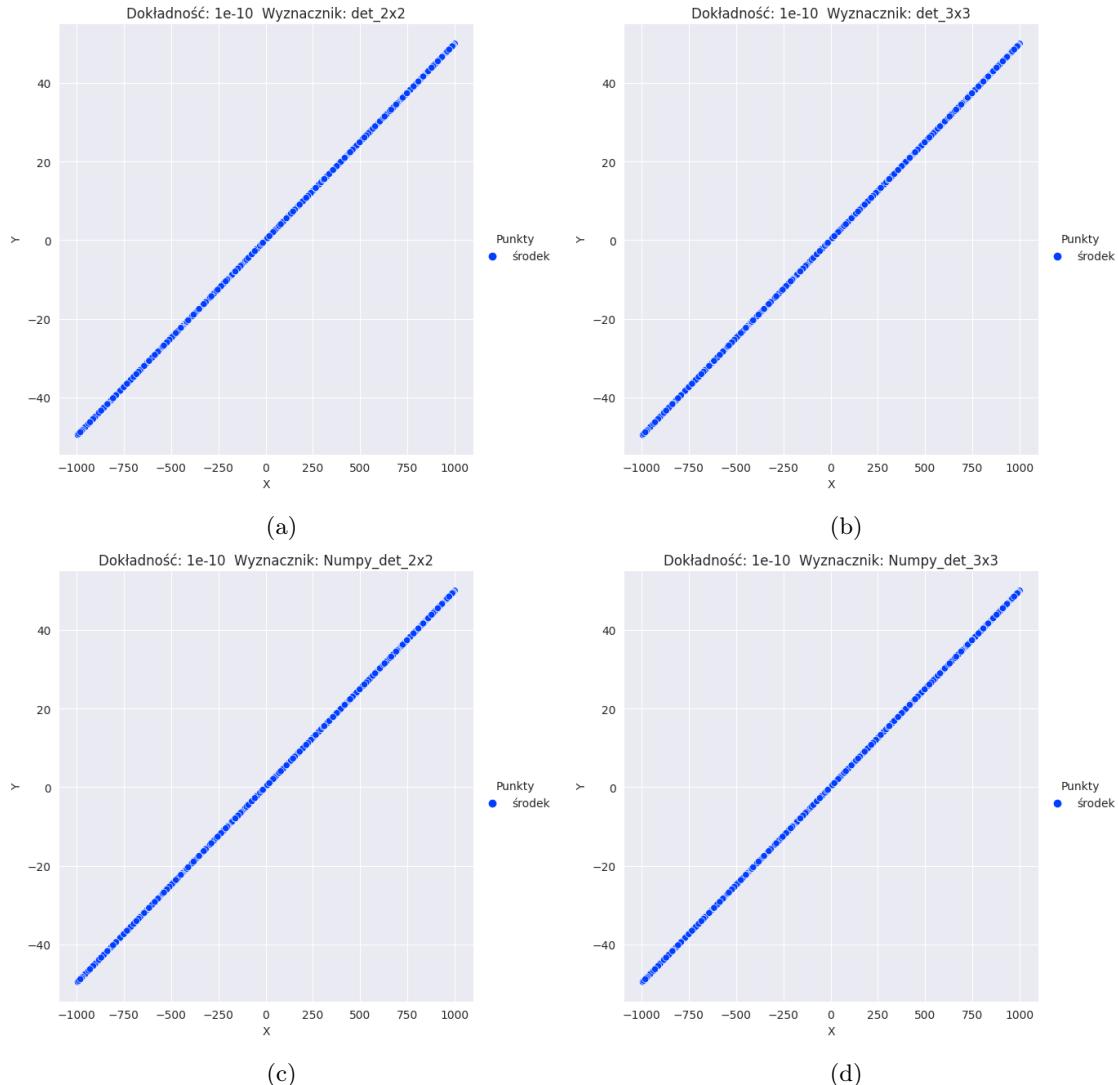
## 6.5 Wnioski dla zbioru c

W zbiorze punktów C doszło do sytuacji podobnej jak dla zbioru A, czyli wyniki podziału punktów nie różnią się od siebie pomimo zastosowania różnej wartości tolerancji oraz innych metod obliczania

wyznacznika.

## 7 Zbiór punktów d)

### 7.1 Tolerancja $e = 10^{-10}$



Rysunek 14

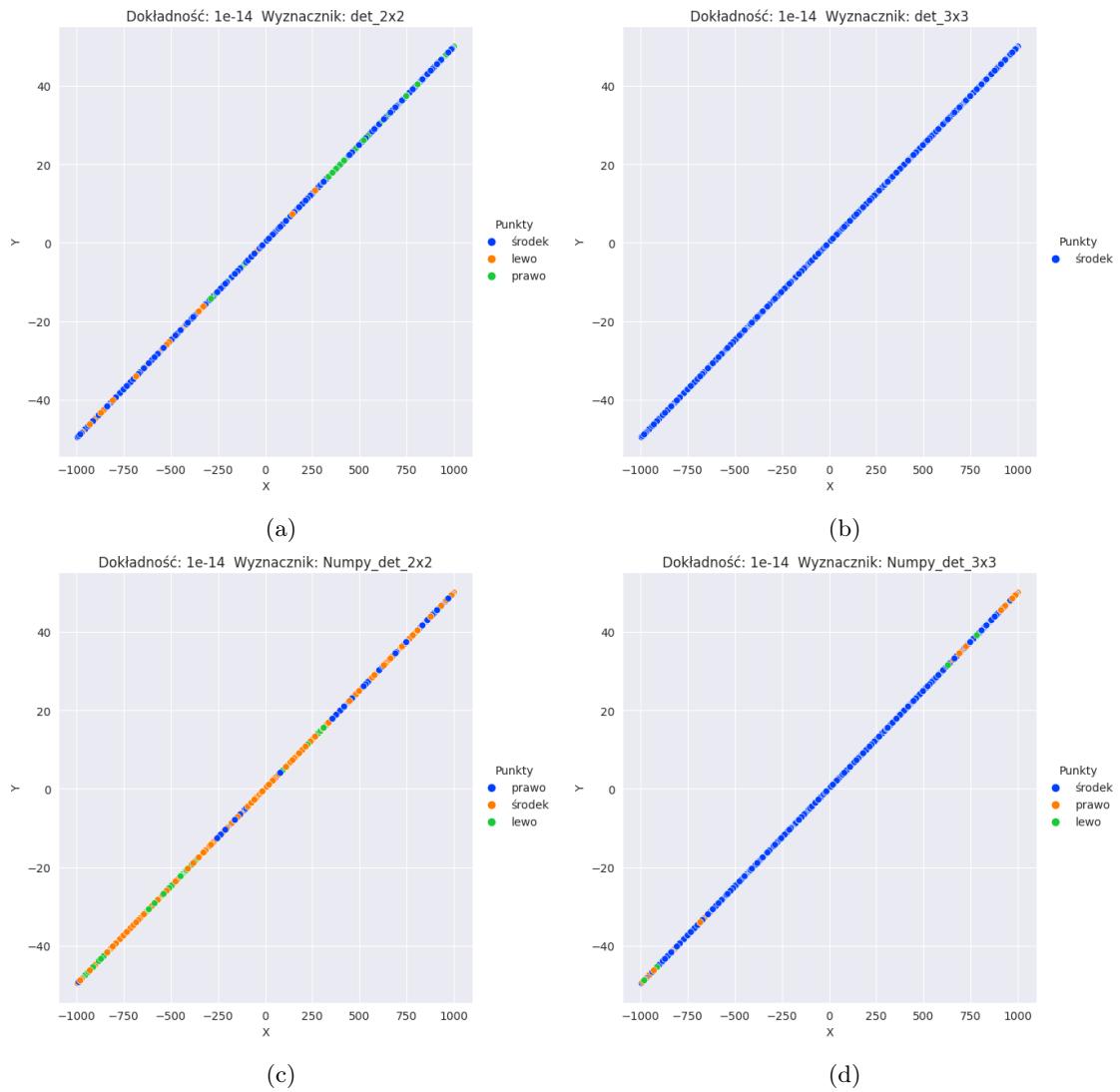
	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
Punkty				
lewo	0	0	0	0
prawo	0	0	0	0
środek	1000	1000	1000	1000

zbiór D, tolerancja  $10^{-10}$

Tabela 13

Dla zbioru d sytuacja jest znacznie ciekawsza, ponieważ dla tolerancji wynoszącej  $10^{-10}$  wszystkie punkty leżą na odcinku ab bez względu na zastosowany sposób liczenia wyznacznika.

## 7.2 Tolerancja $e = 10^{-14}$



Rysunek 15

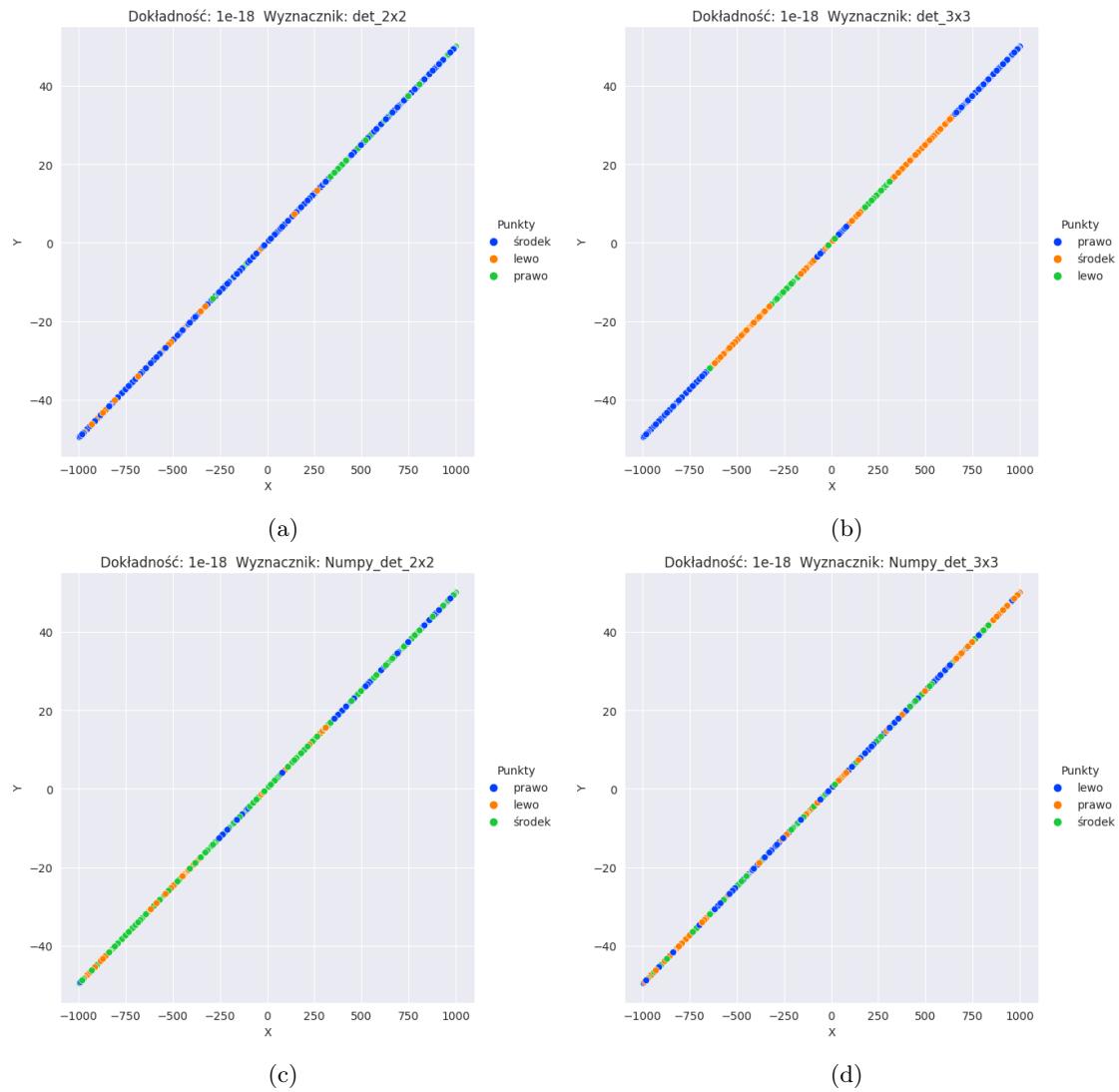
Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	127	0	155	25
prawo	131	0	180	91
środek	742	1000	665	884

zbiór D, tolerancja  $10^{-14}$

Tabela 14

Dla *Epsilon* wynoszącego  $10^{-14}$  det\_2x2 ponownie pozwala nam określić dokładniejsze położenie punktów zarówno w wersji implementowanej ręcznie jak i użytej z biblioteki. Natomiast dla wyznacznika 3x3 wyniki są znacznie mniej sprecyzowane, jeszcze gorszy efekt potrafimy uzyskać przy użyciu metody obliczania zaimplementowanej w bibliotece Numpy.

### 7.3 Tolerancja $e = 10^{-18}$



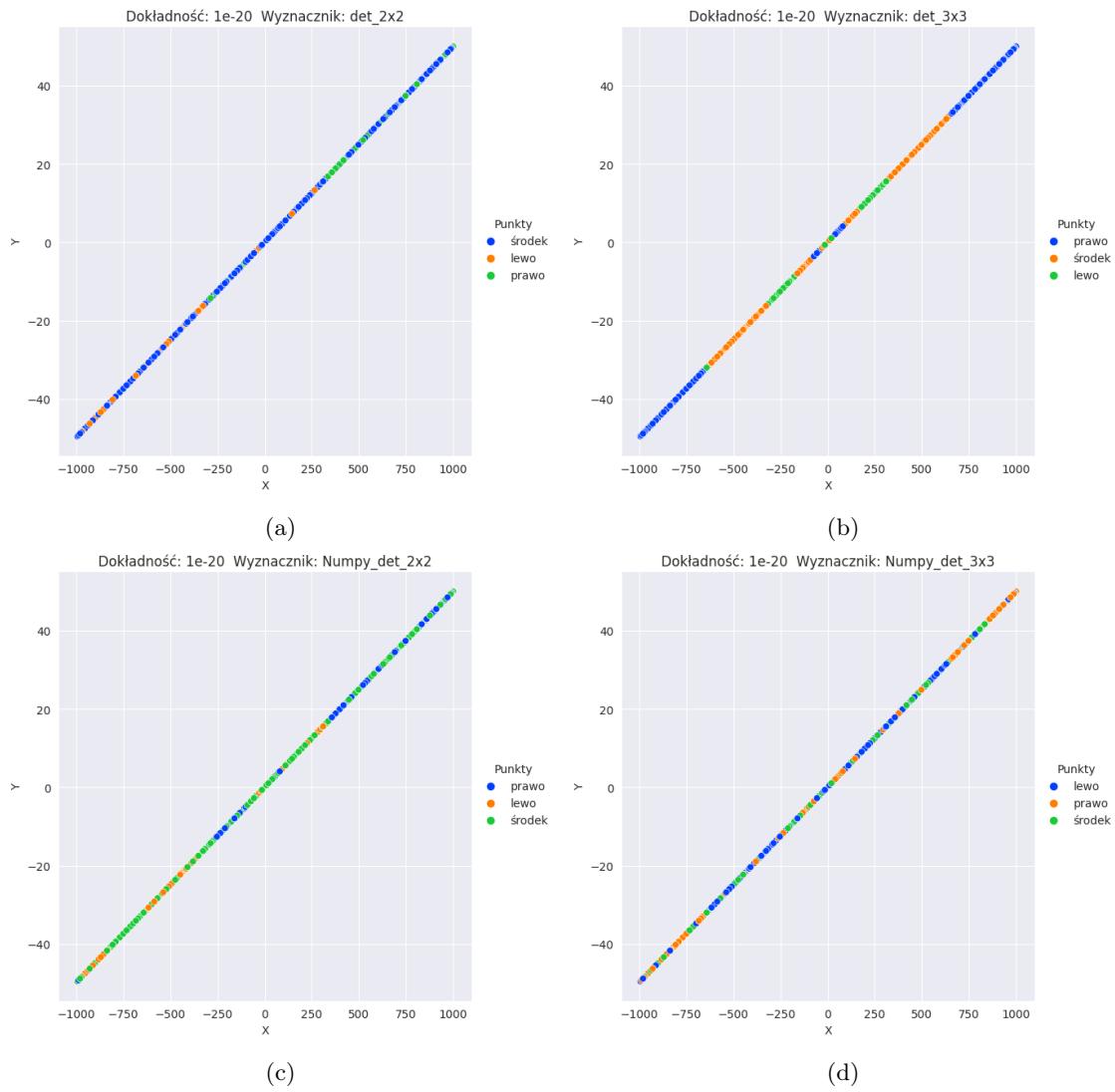
Rysunek 16

Punkty	<i>det_2x2</i>	<i>det_3x3</i>	<i>Numpy_det_2x2</i>	<i>Numpy_det_3x3</i>
lewo	137	183	166	410
prawo	138	367	186	309
środek	725	450	648	281

zbiór D, tolerancja  $10^{-18}$

Tabela 15

## 7.4 Tolerancja $e = 10^{-20}$



Rysunek 17

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	137	183	166	410
prawo	138	367	186	309
środek	725	450	648	281

zbiór D, tolerancja  $10^{-20}$

Tabela 16

## 7.5 Wnioski dla zbioru d

Przy *Epsilonie* równym  $10^{-18}$  oraz  $10^{-20}$  możemy zauważać znaczącą przewagę w dokładności ręcznie implementowanej funkcji obliczania wyznacznika dla macierzy 2x2, sposób ten daje naj-dokładniejsze wyniki, ponieważ dla 1000 losowo wybranych punktów z prostej  $ab$  aż 725 zostało poprawnie zakwalifikowanych, zaś dla funkcji obliczania wyznacznika dostarczonej przez bibliotekę Numpy jedyne 281 punktów zostało poprawnie określonych.

---

## 8 Porównanie tabel

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	49881	49881	49881	49881
prawo	50119	50119	50119	50119
środek	0	0	0	0

zbiór A, dla wszystkich epsilonów taki sam wynik

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	49808	49810	49807	49810
prawo	50188	50190	50187	50190
środek	4	0	6	0

zbiór B, dla wszystkich epsilonów taki sam wynik

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	502	502	502	502
prawo	498	498	498	498
środek	0	0	0	0

zbiór C, dla wszystkich epsilonów taki sam wynik

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	0	0	0	0
prawo	0	0	0	0
środek	1000	1000	1000	1000

zbiór D, tolerancja  $10^{-10}$

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	127	0	155	25
prawo	131	0	180	91
środek	742	1000	665	884

zbiór D, tolerancja  $10^{-14}$

Punkty	det_2x2	det_3x3	Numpy_det_2x2	Numpy_det_3x3
lewo	137	183	166	410
prawo	138	367	186	309
środek	725	450	648	281

zbiór D, tolerancja  $10^{-18}$  oraz  $10^{-20}$

Tabela 17

---

## 9 Wnioski

Ćwiczenie miało na celu porównanie wyników dla różnego rodzaju tolerancji jak i precyzji obliczeń. Z wyżej przedstawionych wyników klasyfikacja punktów dla zbiorów a) (*Rysunek 1a*) i c) (*Rysunek 1c*) nie dała wyników umożliwiających porównanie metod. Z uwagi iż w tych przypadkach dla wszystkich badanych tolerancji obydwa sposoby klasyfikowania położenia punktów względem prostej wydawały się równie skuteczne.

Inna sytuacja miała miejsce dla zbioru b (*Rysunek 1b*) przy którym zastosowanie metody obliczania wyznacznika macierzy  $3 \times 3$  pozwoliło nam pominać punkty błędnie zinterpretowane jako należące do prostej.

Największą przewagę ręcznie implementowanej metody obliczania wyznacznika  $2 \times 2$  jesteśmy w stanie zauważać w zbiorze d (*Rysunek 1d*), ponieważ to właśnie ta metoda dała nam najwięcej poprawnie rozpoznanych punktów.

Patrząc na wykresy *Rysunek 16a* oraz *Rysunek 16b* jesteśmy w stanie zaobserwować iż dla metody obliczania wyznacznika  $2 \times 2$  znaczna ilość punktów jest poprawnie rozpoznawana na całej długości prostej, zaś dla  $\det_{\_}3 \times 3$  głównie dla punktów których współrzędne osiągają wartości bliższe punktom a i b.