

Parametry:

M = 11

N = 14

norma = wierszowa w 1. zadaniu

norma = Frobeniusa w 2. zadaniu

**Raport z Pracowni nr 2****Zadanie 1.**1. Cel zadania

Celem zadania było zbadanie jak zbieżność algorytmu iteracji prostej zależy od parametru alfa, przy normie wierszowej korzystając z metody iteruj\_twierdzenie.

2. Metody

W doświadczeniu wykorzystano zbiór klas udostępnionych w ramach zajęć z kursu Metody numeryczne, stworzonych w języku Python. Została wykorzystana wersja Python 3.11.0. Do obliczeń został wykorzystany komputer z procesorem Ryzen 5 3600.

3. Przyjęte parametry

- $\text{eps} = 1.0\text{E-}10$
- $n = 100$
- $\text{norma} = 0$  (wierszowa)

4. Przebieg doświadczenia i wyniki

Przeprowadzono kilka wstępnych eksperymentów, by określić zakres badanego parametru. Sprawdzono wartości alfa 0.1, 0.5, 0.9, wszystkie dały miarodajne wyniki więc przyjęto, że zakres parametru alfa 0.1-0.9 jest odpowiedni.

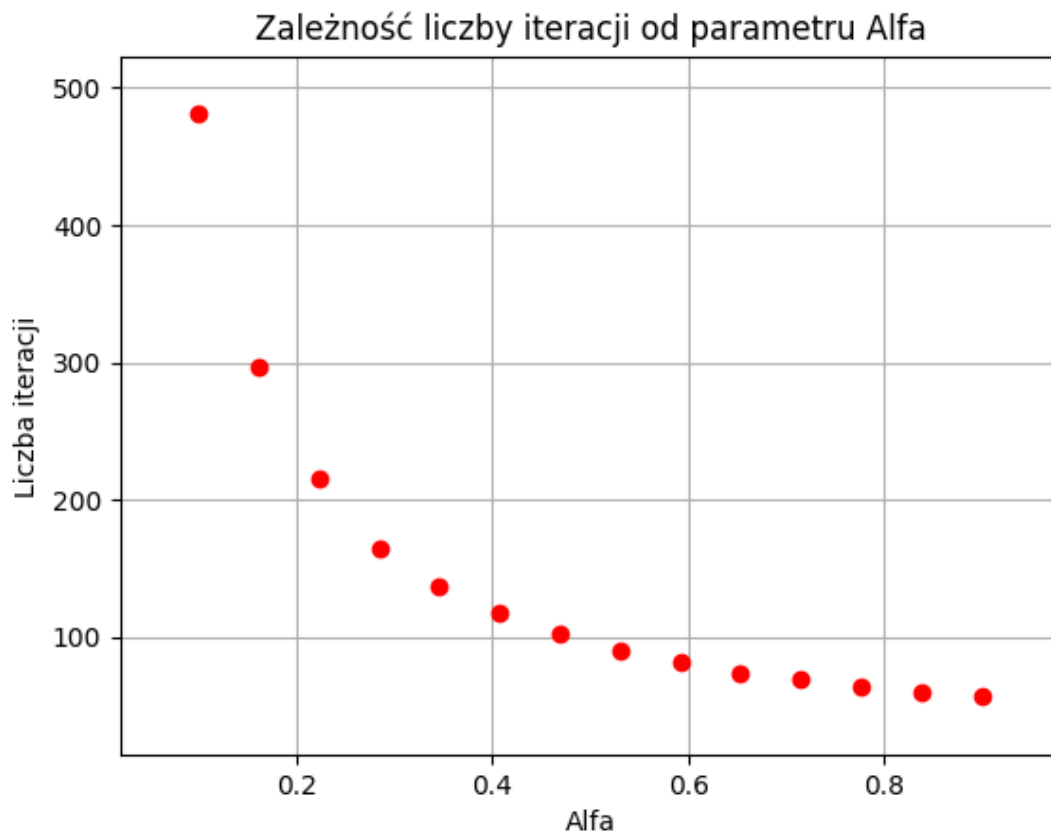
Ustalono  $N = 14$  wartości parametru (0.1, 0.1615, 0.223, 0.2845, 0.346, 0.4075, 0.469, 0.5305, 0.592, 0.6535, 0.715, 0.7765, 0.838, 0.9) oraz dla każdej wartości wykonano 5 eksperymentów, których wybrane charakterystyki (normę macierzy, liczbę iteracji oraz niedokładność rozwiązania) były uśredniane oraz zapisane.

Dla podanych danych wyglądają one następująco:

<i>Alfa</i>	<i>  D  </i>	<i>Iteracje</i>	<i>Niedokładność</i>
<i>0.1</i>	0.957404	480.80	3.330669e-16
<i>0.1615</i>	0.933292	296.20	4.440892e-16
<i>0.223</i>	0.910658	214.80	4.218847e-16
<i>0.2845</i>	0.887047	165.00	4.218847e-16
<i>0.346</i>	0.867246	136.80	3.552714e-16
<i>0.4075</i>	0.847630	117.40	3.108624e-16

0.469	0.830397	102.60	3.774758e-16
0.5305	0.810919	90.20	3.774758e-16
0.592	0.793398	82.00	4.662937e-16
0.6535	0.776247	73.60	3.774758e-16
0.715	0.764690	69.20	3.774758e-16
0.7765	0.746788	63.20	3.663736e-16
0.838	0.736613	59.80	3.330669e-16
0.9	0.723655	56.40	3.774758e-16

Sporządzono również wykres zależności liczby iteracji od parametru alfa:



Rys 1. Zależności liczby iteracji od parametru alfa

## 5. Wnioski

Wraz ze wzrostem parametru alfa zmniejsza się liczba iteracji potrzebna do uzyskania odpowiednio małego błędu. Zbieżność iteracji prostej oznacza, że dla pewnego zbioru wartości początkowych danej funkcji, wartości te będą się zbliżać do jednej znanej wartości po kolejnych iteracjach. Im większa jest zbieżność, tym szybciej będą się one zbliżać do tej wartości, co oznacza, że mniej iteracji będzie potrzebnych, aby osiągnąć pożądany stopień precyzji.

Można wysunąć następujące wnioski: im większa liczba alfa tym większa zbieżność algorytmu iteracji prostej.

## Zadanie 2.

### 1. Cel zadania

Celem zadania było zbadanie wpływu liczby iteracji ( $k$ ) na efektywność uzyskiwania rankingu stron metodą iteracji Seidela oraz metodą potęgową przy normie Frobeniusa korzystając z metody iteruj.

### 2. Metody

W doświadczeniu wykorzystano zbiór klas udostępnionych w ramach zajęć z kursu Metody numeryczne, stworzonych w języku Python. Została wykorzystana wersja Python 3.11.0. Do obliczeń został wykorzystany komputer z procesorem Ryzen 5 3600.

### 3. Przyjęte parametry

- $nn = 100$  (liczba stron)
- $norma = 2$  (Frobeniusa)

### 4. Przebieg doświadczenia i wyniki

Przeprowadzono kilka wstępnych eksperymentów, by określić zakres badanego parametru. Sprawdzone wartości  $k$ : 50, 400, 700, wszystkie dały miarodajne wyniki więc przyjęto, że zakres parametru  $k$  50-700 jest odpowiedni.

Ustalono  $N = 14$  wartości parametru (50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700) oraz dla każdej wartości wykonano 5 eksperymentów, których wybrane charakterystyki (liczbę linków na stronach, niedokładność Seidela oraz niedokładność potęgową) były uśredniane oraz zapisane.

Dla podanych danych wyglądają one następująco:

<i>Liczba iteracji</i>	<i>Liczba linków</i>	<i>Niedokładność Seidela</i>	<i>Niedokładność potęgowa</i>
50	100.219000	2.965984e-02	1.540353e-01
100	99.845000	1.180383e-02	6.433322e-02
150	100.179000	4.247412e-03	3.045789e-02
200	100.034000	1.442809e-03	3.500974e-02
250	99.770000	5.283390e-04	2.347826e-02
300	99.740000	1.872694e-04	3.695528e-02
350	100.252000	8.404843e-05	1.623517e-02
400	100.040000	3.081202e-05	3.085048e-01
450	100.212000	1.159709e-05	1.883886e-02
500	99.713000	3.584487e-06	6.972944e-02
550	99.859000	1.571645e-06	1.513039e-02
600	100.096000	4.890449e-07	9.843341e-03
650	100.012000	1.751280e-07	7.494618e-03
700	99.729000	8.931777e-08	4.241745e-02

## 5. Wnioski

Należy zauważyć, że dla tej samej liczby iteracji niedokładność wyników jest mniejsza w przypadku metody iteracji Seidela niż w przypadku metody potęgowej.

W metodzie iteracji Seidela wyraźnie wydać, że niedokładność spada wraz ze wzrostem ilości iteracji, tak silnej zależności nie można zaobserwować w przypadku metody potęgowej.

Efektywność iteracji Seidela, wraz ze wzrostem liczby iteracji, jest coraz większa w porównaniu do efektywności metody potęgowej.