

# Rozwiązywanie równań liniowych metodami iteracyjnymi

Krystian Madej, 14.06.2024

## 1. Treść zadania

Macierz  $A$  jest zadana wzorami:

$$\begin{aligned}a_{i,i} &= k \\a_{i,j} &= (-1)^j \frac{m}{j} \quad \text{dla } j > i \\a_{i,i-1} &= \frac{m}{i} \\a_{i,j} &= 0 \quad \text{dla } j < i - 1 \\i, j &= 1, 2, \dots, n \\k &= 10, m = 3\end{aligned}$$

Dany jest układ równań liniowych  $Ax = b$ . Przyjmij wektor  $x$  jako dowolną  $n$ -elementową permutację zbioru  $\{-1, 1\}$  i oblicz wektor  $b$ .

1. Metodą Jacobiego rozwiąż układ równań liniowych  $Ax = b$  przyjmując kryteria stopu:

- $\|x^{(i+1)} - x^{(i)}\| < \rho$  - kryterium krokowe
- $\|Ax^{(i)} - b\| < \rho$  - kryterium wartościowe

Obliczenia wykonaj dla różnych rozmiarów układu  $n$ , dla wektorów początkowych:  $(0, 0, \dots, 0)$  i  $(100, -100, 100, -100, \dots)$ , oraz różnych wartości  $\rho$  w kryteriach stopu. Podaj jak liczono normę. Wyznacz liczbę iteracji oraz sprawdź różnicę w czasie obliczeń dla różnych kryteriów stopu. Sprawdź dokładność obliczeń.

2. Dowolną metodą znajdź promień spektralny macierzy iteracji dla różnych rozmiarów układu. Sprawdź czy spełnione są założenia o zbieżności metody dla zadanego układu. Opisz metodę znajdowania promienia spektralnego.

## 2. Środowisko obliczeń

Obliczenia zostały wykonane przy pomocy języka **C++20** na systemie **Windows 11**, kompilacja 22631.3593, na **12-wątkowym** procesorze **64-**

**bitowym** Intel Core i5-11400H 2.70GHz, kod kompilowany kompilatorem **MSVC** (wersja 19.39). Obliczenia wykonywane na typie `double`.

### 3. Użyte biblioteki i programy pomocnicze

Wykresy rysowano programem Excel z pakietu Microsoft Office.

Do instalacji bibliotek **C++** użyto programu **conan**, wersja 2.3.2.

Najważniejsze użyte biblioteki:

- `<format>` - łatwe formatowanie
- `<numbers>` - stałe matematyczne
- `<future>` - obiekty `std::future` oraz `std::async`
- `<thread>` - wielowątkowość
- `<fstream>` - zapisywanie do plików
- `<chrono>` - zegary i operacje na jednostkach czasu
- Armadillo - algebra liniowa, operacje na macierzach/wektorach

## 4. Sposób obliczeń

### 4.1 Obliczanie normy wektora

Norma z wektora  $\|\cdot\|$  jest normą euklidesową:

$$\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$$

### 4.2 Obliczanie macierzy iteracji

Mając układ równań:

$$Ax = b$$

gdzie:

- $A$  - macierz  $n \times n$
- $x$  - wektora  $n$  niewiadomych
- $b$  - wektor danych

Następnie sprowadzamy macierz  $A$  do postaci sumy dwóch macierzy

$$A = B + R$$

gdzie:

- $B$  - macierz diagonalna, której przekątna jest równa przekątnej macierzy  $A$
- $R = A - B$

Przekształcając układ równań otrzymamy:

$$\begin{aligned}Ax &= b = (B + R)x = b \\ Bx &= -Rx + b \\ Bx &= -(A - B)x + b \\ x &= -B^{-1}(A - B)x + B^{-1}b \\ x &= I - B^{-1}Ax + B^{-1}b\end{aligned}$$

Zapisując ostatnie równanie w postaci iteracyjnej otrzymamy:

$$x^{(i+1)} = I - B^{-1}Ax^{(i)} + B^{-1}b$$

Macierz  $M = I - B^{-1}A$  nazywamy macierzą iteracji.

### 4.3 Obliczanie promienia spektralnego

Promień spektralny macierzy iteracji liczono z definicji jako maksimum wartości bezwzględnych z wartości własnych tej macierzy. Wartości własne macierzy są wyliczane biblioteczną funkcją `arma::gen_eig`.

### 4.4 Wartości błędów

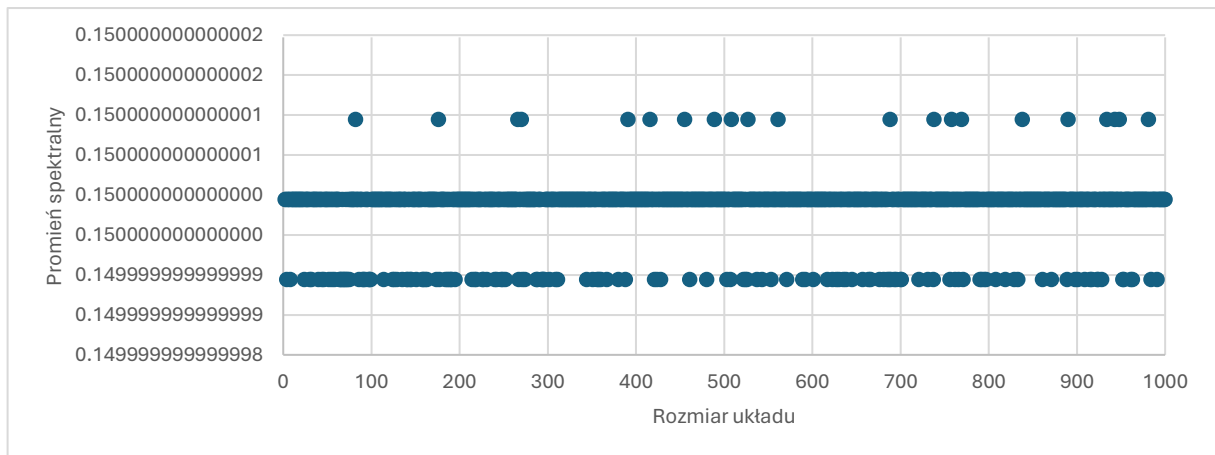
Wartości błędów są liczone normą euklidesową z wektora  $(X - x)$ , gdzie  $X$  jest wektorem zadany, a  $x$  wektorem obliczonym.

## 5. Metoda postępowania

Na początku program tworzy macierz  $A$   $n \times n$  według wzorów opisanych w punkcie 1. Następnie tworzy jest wektor  $x = (1, -1, 1, -1, \dots)$  i oblicza wektor  $b = Ax$ . Następnie tworzone są dwa wektory początkowe, opisane w punkcie 1. Na koniec rozwiązuje się układ równań  $Ax = b$  metodą Jacobiego, dla wszystkich kombinacji  $p$  i warunków stopu.

Obliczenia wykonałem dla  $n \in \{2, 3, \dots, 100\} \cup \{110, 120, \dots, 250\} \cup \{275, 300, \dots, 500\}$ . Ustaliłem limit iteracji na 10000.

## 6.1 Wartości promienia spekralnego

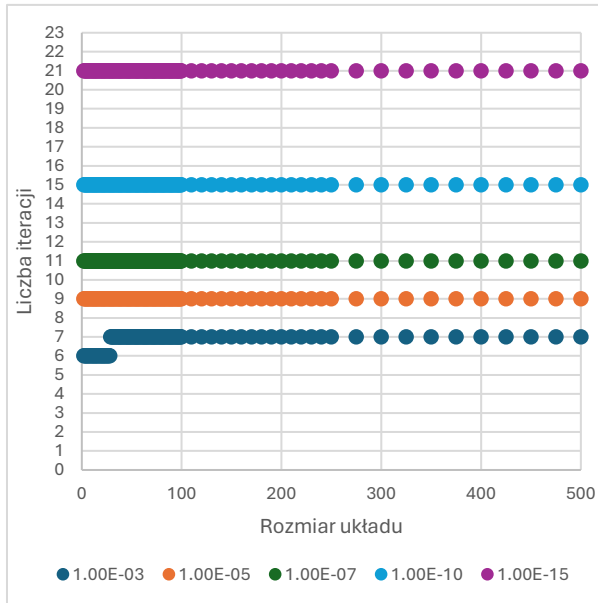


Wykres 1. Promień spektralny

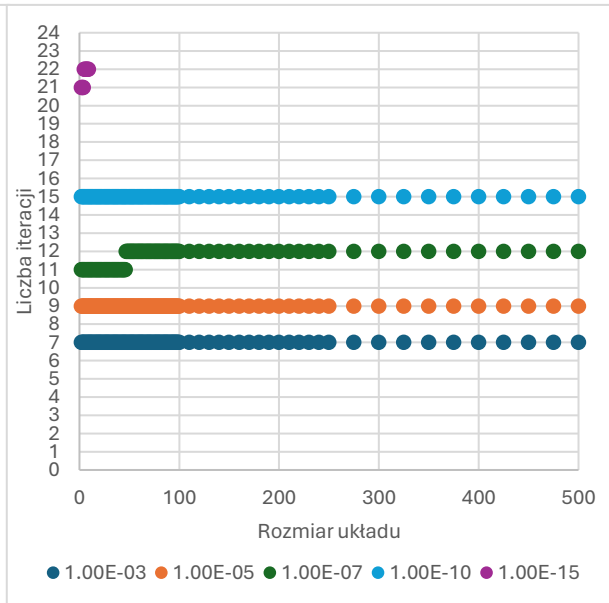
Jak widać na wykresie 1, niezależnie od rozmiaru układu promień spektralny nie zmienia się i jest równy 0.15. Różnice w niektórych miejscach są rzędu  $10^{-15}$  i wynikają z błędów reprezentacji na typie double. Zatem każdy rozmiar układu spełnia twierdzenie o zbieżności procesu iteracyjnego: „Ciąg z dowolnym wektorem startowym  $x^{(0)}$  jest zbieżny do jedyne granicznego  $x^{(inf)}$  wtedy i tylko wtedy, gdy **promień spektralny (spectral radius)** macierzy iteracji jest mniejszy od 1”.

## 6.2 Wyniki obliczeń metody Jacobiego

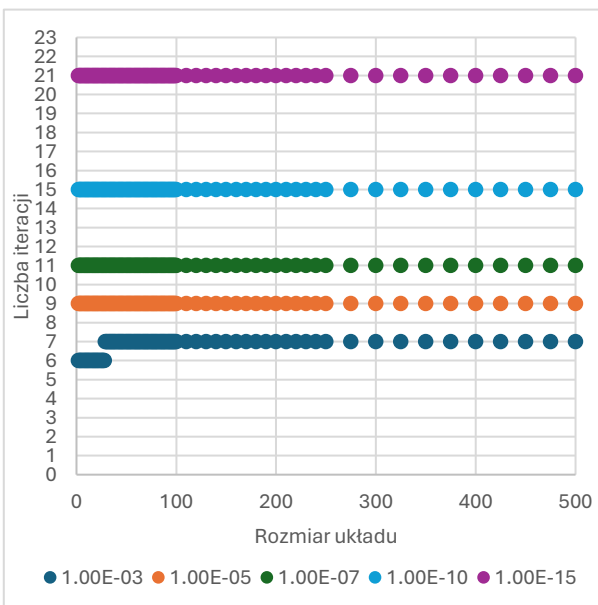
### 6.2.1 Liczba iteracji



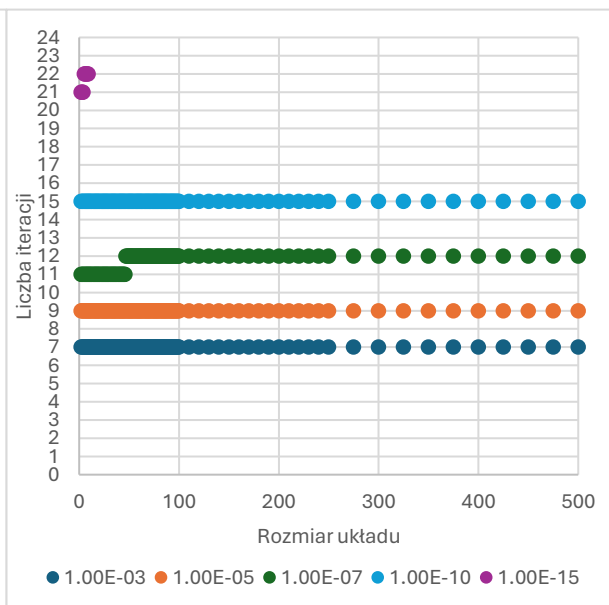
Wykres 2. Liczba iteracji dla niezerowego wektora początkowego i kryterium krokowego



Wykres 3. Liczba iteracji dla niezerowego wektora początkowego i kryterium wartościowego



Wykres 4. Liczba iteracji dla zerowego wektora początkowego i kryterium krokowego



Wykres 5. Liczba iteracji dla zerowego wektora początkowego i kryterium wartościowego

Jak widać na wykresach 2-5 liczba iteracji pozostaje taka sama lub zwiększa się o 1, mimo zwiększania rozmiaru układu. Liczka iteracji zwiększa się wraz z dokładnością. Można także zauważyć iż liczby iteracji dla poszczególnych kryteriów stopu nie zmieniają się pomimo użycia innego wektora początkowego. Wartym uwagi jest fakt, że dla kryterium wartościowego obliczenia nie zakańczały się po ustalonym limicie iteracji (równym 10000) już dla niewielkich rozmiarów układu. Jest to widoczne na wykresie jako brak dalszych punktów dla  $\rho = 10^{-15}$ .

Nie jest to spowodowane niezbieżnością układu, gdyż dla każdy z nich spełnia warunek zbieżności (pokazane w punkcie 6.1), a niedoskonałością artmetyki na typie double.

### 6.2.2 Błędy obliczeń

n\rho	Kryterium Krokowe					Kryterium Wartościowe				
	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15
2	0.000121	4.08E-07	9.17E-09	4.64E-12	0	1.81E-05	4.08E-07	9.17E-09	4.64E-12	0
10	0.000173	4.68E-07	1.09E-08	5.41E-12	5.87E-16	1.95E-05	4.68E-07	1.09E-08	5.41E-12	5.87E-16
20	0.000198	4.76E-07	1.12E-08	5.4E-12	1.09E-15	1.96E-05	4.76E-07	1.12E-08	5.4E-12	1.09E-15
30	1.99E-05	4.82E-07	1.13E-08	5.39E-12	1.59E-15	1.99E-05	4.82E-07	1.13E-08	5.39E-12	1.59E-15
40	2.02E-05	4.87E-07	1.14E-08	5.38E-12	1.98E-15	2.02E-05	4.87E-07	1.14E-08	5.38E-12	2.03E-15
50	2.05E-05	4.91E-07	1.15E-08	5.38E-12	2.19E-15	2.05E-05	4.91E-07	1.53E-09	5.38E-12	2.27E-15
60	2.07E-05	4.94E-07	1.16E-08	5.37E-12	2.33E-15	2.07E-05	4.94E-07	1.53E-09	5.37E-12	2.33E-15
70	2.1E-05	4.97E-07	1.16E-08	5.37E-12	2.36E-15	2.1E-05	4.97E-07	1.53E-09	5.37E-12	2.32E-15
80	2.12E-05	4.99E-07	1.17E-08	5.37E-12	2.75E-15	2.12E-05	4.99E-07	1.53E-09	5.37E-12	2.74E-15
90	2.14E-05	5.02E-07	1.17E-08	5.37E-12	2.9E-15	2.14E-05	5.02E-07	1.52E-09	5.37E-12	2.85E-15
100	2.16E-05	5.04E-07	1.18E-08	5.36E-12	2.63E-15	2.16E-05	5.04E-07	1.52E-09	5.36E-12	2.62E-15
200	2.31E-05	5.19E-07	1.21E-08	5.35E-12	4.65E-15	2.31E-05	5.19E-07	1.51E-09	5.35E-12	4.62E-15
300	2.4E-05	5.29E-07	1.23E-08	5.34E-12	5.22E-15	2.4E-05	5.29E-07	1.51E-09	5.34E-12	5.22E-15
400	2.48E-05	5.36E-07	1.24E-08	5.33E-12	6.99E-15	2.48E-05	5.36E-07	1.5E-09	5.33E-12	6.99E-15
500	2.54E-05	5.41E-07	1.25E-08	5.33E-12	1.3E-14	2.54E-05	5.41E-07	1.5E-09	5.33E-12	1.3E-14

Tabela 1. Błędy obliczeń dla niezerowego wektora początkowego

n\rho	Kryterium Krokowe					Kryterium Wartościowe				
	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15
2	0.000121	4.08E-07	9.17E-09	4.64E-12	0	1.81E-05	4.08E-07	9.17E-09	4.64E-12	0
10	0.000173	4.68E-07	1.09E-08	5.41E-12	5.87E-16	1.95E-05	4.68E-07	1.09E-08	5.41E-12	5.87E-16
20	0.000198	4.76E-07	1.12E-08	5.4E-12	1.09E-15	1.96E-05	4.76E-07	1.12E-08	5.4E-12	1.09E-15
30	1.99E-05	4.82E-07	1.13E-08	5.39E-12	1.59E-15	1.99E-05	4.82E-07	1.13E-08	5.39E-12	1.59E-15
40	2.02E-05	4.87E-07	1.14E-08	5.38E-12	1.98E-15	2.02E-05	4.87E-07	1.14E-08	5.38E-12	2.03E-15
50	2.05E-05	4.91E-07	1.15E-08	5.38E-12	2.19E-15	2.05E-05	4.91E-07	1.53E-09	5.38E-12	2.27E-15
60	2.07E-05	4.94E-07	1.16E-08	5.37E-12	2.33E-15	2.07E-05	4.94E-07	1.53E-09	5.37E-12	2.33E-15
70	2.1E-05	4.97E-07	1.16E-08	5.37E-12	2.36E-15	2.1E-05	4.97E-07	1.53E-09	5.37E-12	2.32E-15
80	2.12E-05	4.99E-07	1.17E-08	5.37E-12	2.75E-15	2.12E-05	4.99E-07	1.53E-09	5.37E-12	2.74E-15
90	2.14E-05	5.02E-07	1.17E-08	5.37E-12	2.9E-15	2.14E-05	5.02E-07	1.52E-09	5.37E-12	2.85E-15
100	2.16E-05	5.04E-07	1.18E-08	5.36E-12	2.63E-15	2.16E-05	5.04E-07	1.52E-09	5.36E-12	2.62E-15
200	2.31E-05	5.19E-07	1.21E-08	5.35E-12	4.65E-15	2.31E-05	5.19E-07	1.51E-09	5.35E-12	4.62E-15
300	2.4E-05	5.29E-07	1.23E-08	5.34E-12	5.22E-15	2.4E-05	5.29E-07	1.51E-09	5.34E-12	5.22E-15
400	2.48E-05	5.36E-07	1.24E-08	5.33E-12	6.99E-15	2.48E-05	5.36E-07	1.5E-09	5.33E-12	6.99E-15
500	2.54E-05	5.41E-07	1.25E-08	5.33E-12	1.3E-14	2.54E-05	5.41E-07	1.5E-09	5.33E-12	1.3E-14

Tabela 2. Błędy obliczeń dla zerowego wektora początkowego

Jak widać na tabelach 1. i 2., wartości błędów dla obu wektorów początkowych są niemal identyczne. Dla kryterium krokowego i  $\rho = 10^{-3}$  błąd zwiększa się dla  $n \in [2, 30)$ , po czym gwałtownie spada o jeden rząd wielkości. Podobną sytuację widać dla kryterium wartościowego i  $\rho = 10^{-7}$ . Można też zauważyć, że pomimo wykonania 10000 iteracji, większość wartości dla  $\rho = 10^{-15}$  nie jest nawet bliska spełnieniu kryterium wartościowego. Wartości błędów są niemal identyczne dla większych układów ( $n \geq 50$ ), niezależnie od użytego kryterium.

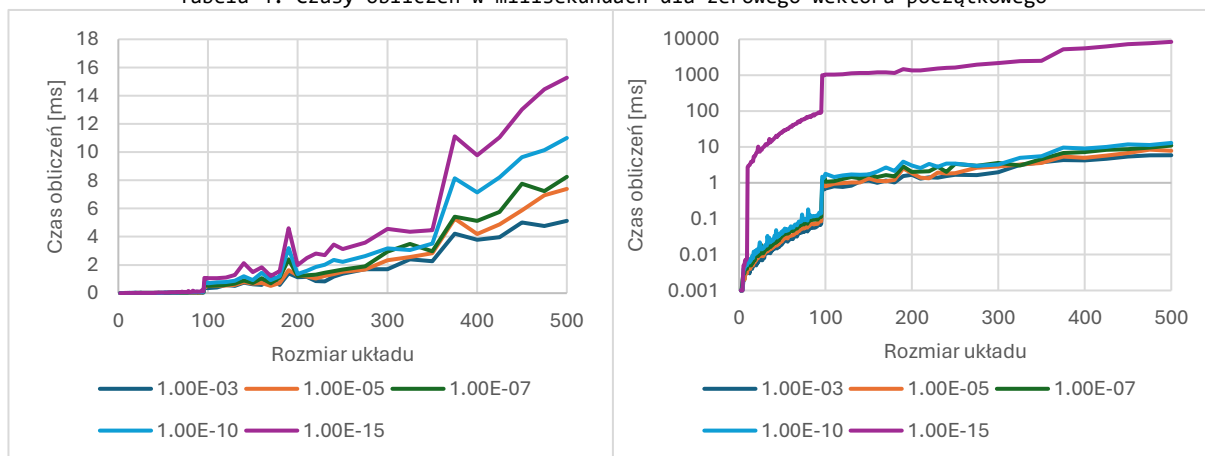
### 6.2.3 Czas obliczeń

n\rho	Kryterium Krokowe					Kryterium Wartościowe				
	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15
2	0.004	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
10	0.002	0.007	0.006	0.01	0.007	0.004	0.004	0.008	0.008	2.75
20	0.005	0.005	0.005	0.007	0.01	0.005	0.007	0.009	0.011	6.704
30	0.008	0.008	0.01	0.013	0.018	0.012	0.012	0.014	0.018	11.118
40	0.012	0.011	0.018	0.017	0.023	0.014	0.016	0.02	0.026	16.39
50	0.027	0.017	0.023	0.027	0.037	0.02	0.024	0.033	0.039	25.108
60	0.021	0.024	0.031	0.039	0.055	0.027	0.034	0.045	0.061	36.959
70	0.054	0.032	0.038	0.051	0.076	0.04	0.046	0.061	0.081	52.355
80	0.037	0.042	0.056	0.068	0.093	0.099	0.104	0.139	0.181	68.953
90	0.041	0.049	0.06	0.081	0.111	0.06	0.075	0.097	0.122	84.388
100	0.357	0.516	0.517	0.703	1.061	0.69	0.805	1.069	1.768	1040.643
200	1.119	1.262	1.209	1.355	1.996	1.688	1.865	2.016	3.048	1354.908
300	1.692	2.328	2.939	3.174	4.563	1.955	2.863	3.552	3.248	2165.11
400	3.784	4.2	5.12	7.135	9.778	4.26	5.01	7.188	9.088	5595.885
500	5.125	7.389	8.246	11.002	15.274	5.808	7.808	10.943	12.939	8481.038

Tabela 3. Czasy obliczeń w milisekundach dla niezerowego wektora początkowego

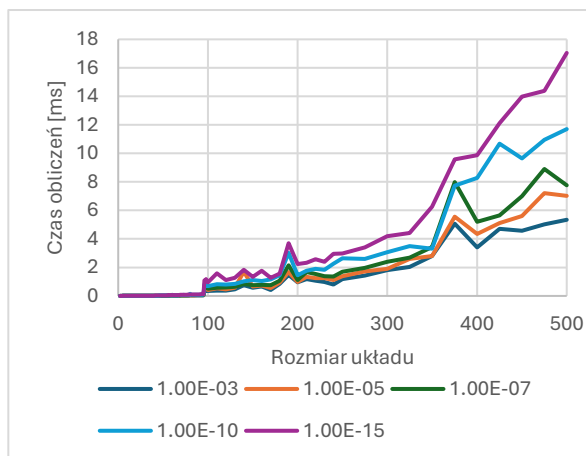
n\rho	Kryterium Krokowe					Kryterium Wartościowe				
	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15	1.00E-03	1.00E-05	1.00E-07	1.00E-10	1.00E-15
2	0	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
10	0.003	0.003	0.003	0.007	0.006	0.004	0.003	0.003	0.004	2.584
20	0.003	0.005	0.006	0.007	0.01	0.006	0.006	0.007	0.009	7.952
30	0.008	0.008	0.01	0.013	0.017	0.01	0.01	0.012	0.016	10.259
40	0.009	0.011	0.013	0.017	0.024	0.014	0.015	0.018	0.028	17.053
50	0.013	0.016	0.022	0.026	0.037	0.022	0.024	0.031	0.048	27.297
60	0.086	0.027	0.03	0.039	0.054	0.033	0.039	0.056	0.158	38.649
70	0.026	0.032	0.04	0.05	0.072	0.045	0.051	0.061	0.079	50.616
80	0.061	0.068	0.067	0.146	0.1	0.056	0.063	0.085	0.108	68.583
90	0.039	0.049	0.06	0.078	0.109	0.063	0.079	0.115	0.134	84.486
100	0.352	0.423	0.51	0.69	0.96	0.699	0.857	1.138	1.369	1045.612
200	0.957	0.987	1.111	1.449	2.24	1.184	1.405	1.937	2.233	1328.087
300	1.82	1.906	2.396	3.063	4.19	1.877	2.519	3.036	3.485	2132.187
400	3.4	4.338	5.205	8.279	9.863	5.264	5.542	7.191	8.611	5577.827
500	5.341	7.017	7.753	11.702	17.045	5.861	8.242	10.101	13.342	8559.699

Tabela 4. Czasy obliczeń w milisekundach dla zerowego wektora początkowego

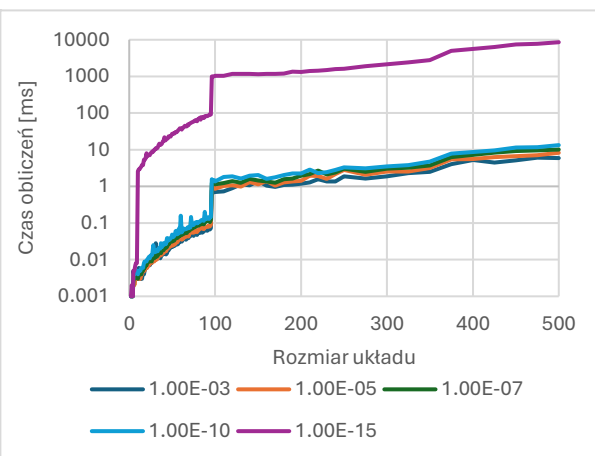


Wykres 6. Czasy obliczeń w milisekundach dla niezerowego wektora początkowego i kryterium krokowego.

Wykres 7. Czasy obliczeń w milisekundach dla niezerowego wektora początkowego i kryterium wartościowego, skala logarytmiczna.



Wykres 7. Czasy obliczeń w milisekundach dla zerowego wektora początkowego i kryterium krokowego.



Wykres 9. Czasy obliczeń w milisekundach dla zerowego wektora początkowego i kryterium wartościowego, skala logarytmiczna.

Jak widać na tabelach 3. i 4. oraz wykresach 6-9, czas obliczeń zwiększa się wraz z rozmiarem układu. Podobnie na czas wpływa dokładność obliczeń. Dobrze widoczny jest nagły skok w czasie obliczeń dla  $\rho = 10^{-15}$  oraz kryterium wartościowego. Następuje on dokładnie tam, gdzie obliczenia zaczęły zakańczać się nie pod wpływem spełnienia kryterium stopu, lecz przekraczania limitu iteracji. Nie widać znaczących różnic w zależności od użycia wektora zerowego lub niezerowego. Na wykresach jest też widoczny nagły skok w czasie obliczeń blisko  $n = 100$ . Najbardziej prawdopodobnym powodem jest implementacja wewnątrz biblioteki Armadillo.

## 7. Wnioski

Dla zadanego układu równań wartość promienia spektralnego jest praktycznie stała, nie zależy od rozmiaru układu. Dodatkowo dowodzi, że każdy układ jest zbieżny metodą iteracyjną.

Liczba iteracji dla danej dokładności obliczeń jest względnie stała, w niektórych przypadkach rośnie co najwyżej o 1, po czym dalej nie zmienia się. Zwiększa się natomiast wraz ze zmniejszaniem wartości  $\rho$ . Na liczbę iteracji nie wpływa to czy rozpoczniemy obliczenia od wektora zerowego czy niezerowego. Ze względu na ograniczenie artmetyki zmiennoprzecinkowej obliczenia dla wartościowego warunku stopu i  $\rho = 10^{-15}$  nie były zbieżne, pomimo iż udowodniona została zbieżność dla układu.

W większości przypadków wartości błędów pozostają niewielkie. Dla większych rozmiarów układów czas obliczeń jest niemal identyczny dla obu kryteriów. Podobnie jak w przypadku liczby iteracji, wartości wektora początkowego nie wpłynęły na wartości błędów.



Czas obliczeń dla tej samej wartości  $\rho$  był większy dla kryterium wartościowego. Zwiększał się wraz z rozmiarem układu. Wybór wektora początkowego nie wpłynął istotnie na czas obliczeń. Dla kryterium wartościowego i  $\rho = 10^{-15}$  obliczenia trwały najdłużej, co nie jest dziwne biorąc pod uwagę liczbę wykonanych iteracji.