

A G H

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

Metody Obliczeniowe w Nauce i Technice
Rozwiązywanie układów równań liniowych metodami iteracyjnymi
Zadanie 7

Maciej Kmąk
Informatyka WI AGH, II rok

1 Treść Zadania

Dany jest układ

$$A x = b,$$

gdzie $A = [a_{i,j}]_{i,j=1}^n$ ma elementy zdefiniowane wzorem (dla parametrów $m = 4$, $k = 7$):

$$a_{i,i} = 7, \quad a_{i,j} = \frac{1}{|i - j| + 4} \quad (i \neq j, i, j = 1, \dots, n).$$

Wektor $x_{\text{zadany}} \in \{-1, 1\}^n$ losujemy z równym prawdopodobieństwem, a następnie definiujemy

$$b = A x_{\text{zadany}}, \quad x_{\text{zadany}} = (x_1, \dots, x_n)^T, \quad x_i \in \{-1, 1\}.$$

Zadanie 1. Metodą Jacobiego rozwiąż układ równań liniowych

$$A x = b,$$

przyjmując kolejno kryterium stopu:

$$\|x^{(i+1)} - x^{(i)}\| < \varepsilon \quad \text{lub} \quad \|A x^{(i)} - b\| < \varepsilon.$$

Porównaj wyniki uzyskane dla obu kryteriów.

Zadanie 2. Wyznacz promień spektralny macierzy iteracji Jacobiego. Sprawdź, czy spełnione są założenia o zbieżności metody dla zadanego układu. Opisz metodę znajdowania promienia spektralnego.

2 Dane techniczne

Eksperymenty przeprowadzono na komputerze osobistym z następującą konfiguracją sprzętową:

- System operacyjny: Windows 11 Pro
- Procesor: 12th Gen Intel (R) Core (TM) i5-1235U @ 1.3 GHz (10 rdzeni)
- Pamięć RAM: 16 GB DDR4

Środowisko programistyczne obejmuje:

- Python 3.12 z bibliotekami:

- NumPy
- Pandas
- time (import time)
- Matplotlib
- Seaborn
- functools.wraps

3 Przebieg Doświadczenia

Zgodnie z zaleceniem prowadzącej, w pierwszej kolejności zbadano promień spektralny macierzy iteracji Jacobiego.

Dla wszystkich rozmiarów n z przedziału $1 \leq n \leq 1000$. Wykorzystano następującą funkcję do obliczania promienia spektralnego:

```

1 def compute_spectral_radius(A):
2     D = np.diag(A)
3     R = A - np.diagflat(D)
4     B = -R / D[:, None]
5     eigs = eigvals(B)
6     return np.max(np.abs(eigs))

```

Listing 1: Funkcja do obliczania promienia spektralnego macierzy iteracji Jacobiego

Gdzie:

- A to macierz układu (wariant b , z $k = 7$, $m = 4$),
- $D = \text{diag}(A)$ – wektor przekątnej macierzy A ,
- $R = A - \text{diag}(D)$ – macierz pozostałości bez części diagonalnej,
- $B = -D^{-1}R$ – macierz iteracji Jacobiego,
- `eigvals(B)` – funkcja NumPy zwracająca wszystkie wartości własne macierzy B ,
- $\rho = \max |\lambda|$ – największa wartość bezwzględna spośród wartości własnych.

Procedura badawcza:

1. Dla każdego n z przedziału $1 \leq n \leq 1000$ skonstruowano macierz A .
2. Dla każdej macierzy A wywołano `compute_spectral_radius(A)`, aby obliczyć $\rho(G)$, gdzie $G = -D^{-1}R$.
3. Zapisano wartości $\rho(G)$ w funkcji n i sprawdzono, dla jakich najmniejszych i największych n zachodzi $\rho(G) < 1$ oraz $\rho(G) \geq 1$.
4. Na podstawie uzyskanych wyników zawęzono rozmiary n do przedziału, w którym promień spektralny przechodzi z wartości mniejszej niż 1 na wartości większe lub równe 1.

Po zidentyfikowaniu przedziału, w którym $\rho(G)$ przekracza 1, skoncentrowano się na dalszej analizie jedynie dla tych wybranych n . W kolejnej części dokumentu przedstawiono szczegółowe wyniki pomiarów promienia spektralnego w wyznaczonym przedziale oraz dalsze obserwacje dotyczące zbieżności metody Jacobiego.

4 Wyniki doświadczenia

W pierwszym etapie obliczeń zmierzono wartości promienia spektralnego $\rho(G)$ dla wybranych rozmiarów układu n . Wyniki zestawiono w Tabeli 1.

Tabela 1: Wybrane wartości promienia spektralnego $\rho(G)$ dla kilku rozmiarów n

n	$\rho(G)$
25	0.3405478859
50	0.4900500404
100	0.6583923036
200	0.8392368492
300	0.9488049276
400	1.0277159836
500	1.0894438322
600	1.1401561489
700	1.1831982596
800	1.2205899143
900	1.2536448083
1000	1.2832656794

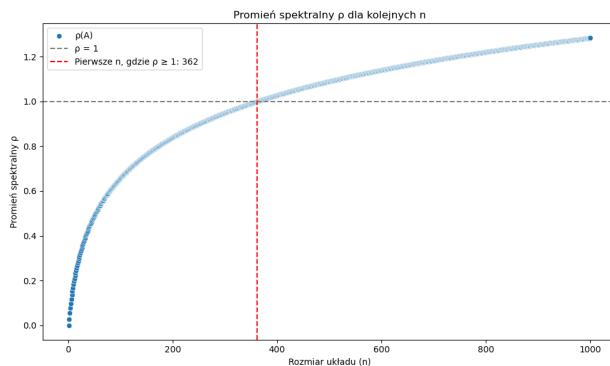
Zgodnie z Tabelą 1, promień spektralny $\rho(G)$ jest mniejszy od 1 dla $n \leq 300$ i większy od 1 dla $n \geq 400$, więc badanie zawężono do przedziału $300 \leq n \leq 400$.

Tabela 2: Promień spektralny $\rho(G)$ dla $360 \leq n \leq 363$

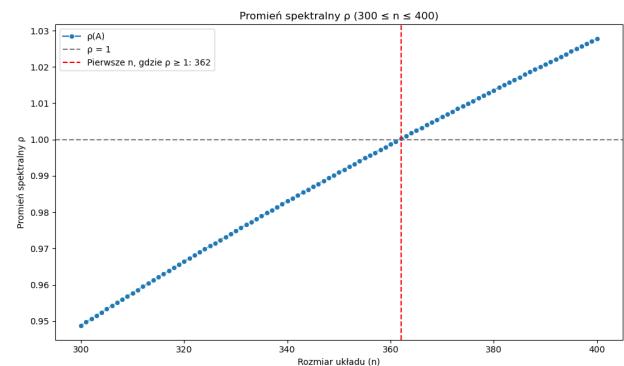
n	$\rho(G)$
360	0.9987199477
361	0.9994820082
362	1.0002420352
363	1.0010000395

Punkt, w którym $\rho(G)$ po raz pierwszy przekracza 1, przypada na $n = 362$.

- Rysunek 1a pokazuje przebieg $\rho(G)$ w funkcji n dla całego zakresu $1 \leq n \leq 1000$.
- Rysunek 1b to powiększenie fragmentu z przedziału $300 \leq n \leq 400$.



(a) Pełny zakres $1 \leq n \leq 1000$.



(b) Powiększenie dla $300 \leq n \leq 400$.

Rysunek 1: (a) Promień spektralny $\rho(G)$ w funkcji n dla $1 \leq n \leq 1000$. (b) Szczegółowy wykres w przedziale $300 \leq n \leq 400$, z zaznaczonym $n = 362$ (próg $\rho = 1$).

5 Dalszy Przebieg Doświadczenia

Na podstawie wyników z badania promienia spektralnego zawężono zestaw rozmiarów n do wartości:

$$n \in \{3, 5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500\},$$

aby potwierdzić, że dla $n = 400$ i $n = 500$ zachodzi $\rho(G) > 1$ i metoda Jacobiego teoretycznie się nie zbiega. Wszystkie dalsze obliczenia wykonano w precyzyji `float64`.

Dla każdej wybranej wielkości n rozwiązywano układ $Ax = b$ metodą Jacobiego, stosując dwa kryteria stopu:

1. $\|x^{(i+1)} - x^{(i)}\| < \varepsilon,$
2. $\|Ax^{(i)} - b\| < \varepsilon,$

gdzie próg ε przyjmowano z zestawu:

$$\varepsilon \in \{10^{-2}, 10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}, 10^{-12}, 10^{-15}\}.$$

Ponadto, dla każdego zestawu parametrów generowano trzy różne wektory początkowe $x^{(0)}$:

1. **Wektor zerowy:**

$$x_{\text{zero}}^{(0)} = [0, 0, \dots, 0]^T \in R^n.$$

2. **Wektor losowany z $\{-100, 100\}$:**

$x_{100}^{(0)}$ — każdy element jest losowo wybrany z zestawu $\{-100, 100\}$.

3. **Wektor „pośredni”:**

$x_{\text{mid}}^{(0)}$ — każdy element jest losowo wybrany z $\{\pm 30, \pm 40, \pm 50, \pm 60\}$.

Dla każdej kombinacji kryterium stopu oraz wariantu wektora początkowego badane były następujące miary:

- liczba iteracji potrzebna do spełnienia kryterium,
- średni czas wykonania pojedynczej iteracji,
- błąd maksymalny

$$e_{\text{max}} = \|x_{\text{true}} - x^{(N)}\|_\infty,$$

- błąd euklidesowy

$$e_{\text{euk}} = \|x_{\text{true}} - x^{(N)}\|_2.$$

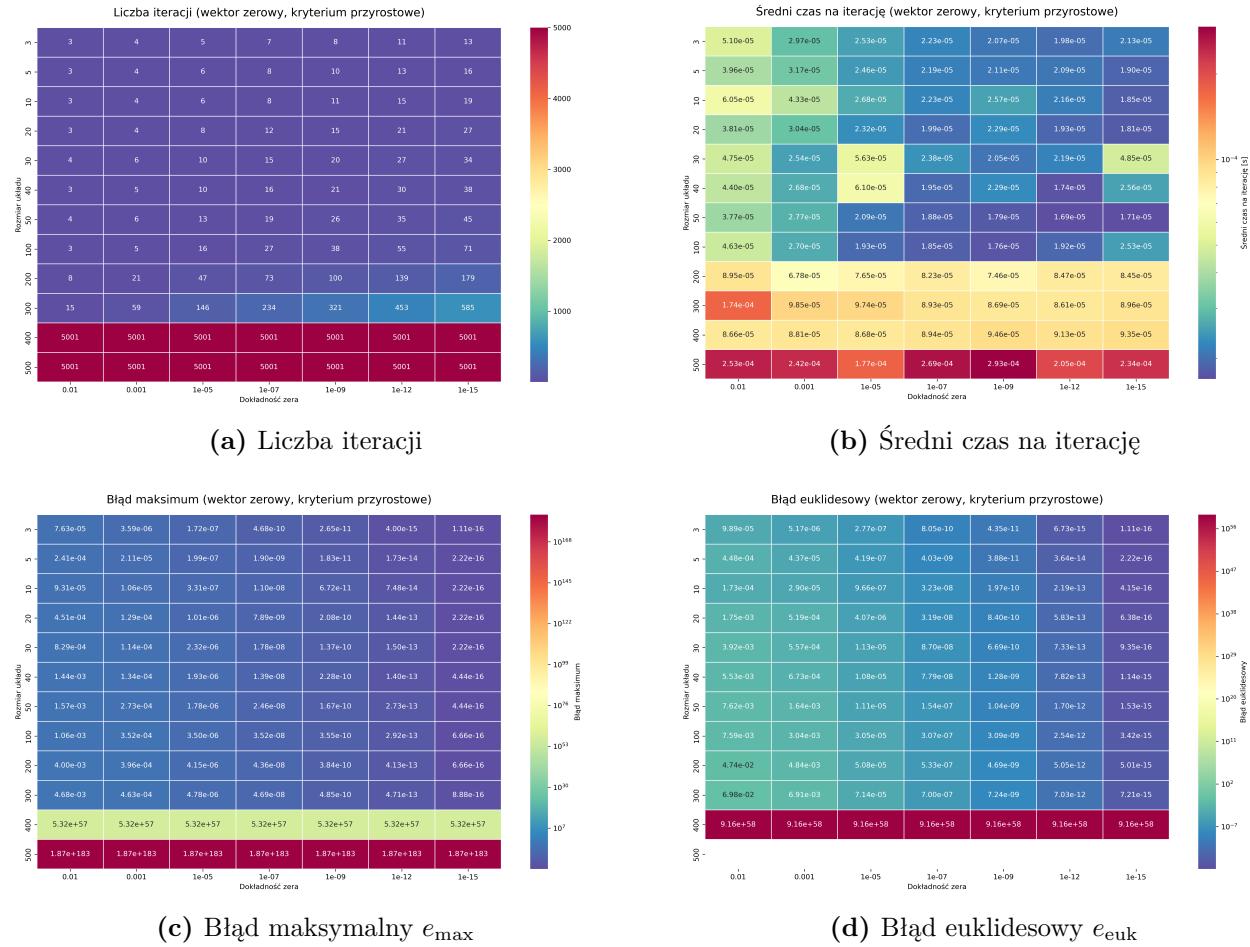
Wszystkie normy w kryteriach stopu obliczane były przy użyciu normy maksimum (∞ -norm), gdzie w praktycznej implementacji użyto funkcji `numpy.linalg.norm(..., ord=np.inf)`. Zdecydowano się na normę maksimum zamiast euklidesowej (2-norm), ponieważ jej obliczenie wymaga jedynie znalezienia największej bezwzględnej wartości wektora (bez sumowania i pierwiastkowania), co jest szybsze obliczeniowo i mniej podatne na błędy numeryczne dla dużych wymiarów.

W dalszej części przedstawimy wyniki jako wykresy typu map ciepła (heatmap) wygenerowane przy użyciu biblioteki Seaborn, aby zwizualizować zależności pomiędzy rozmiarem n , progiem ε i miarami efektywności (liczba iteracji, czas wykonania, błąd).

6 Wyniki dalszego przebiegu doświadczenia

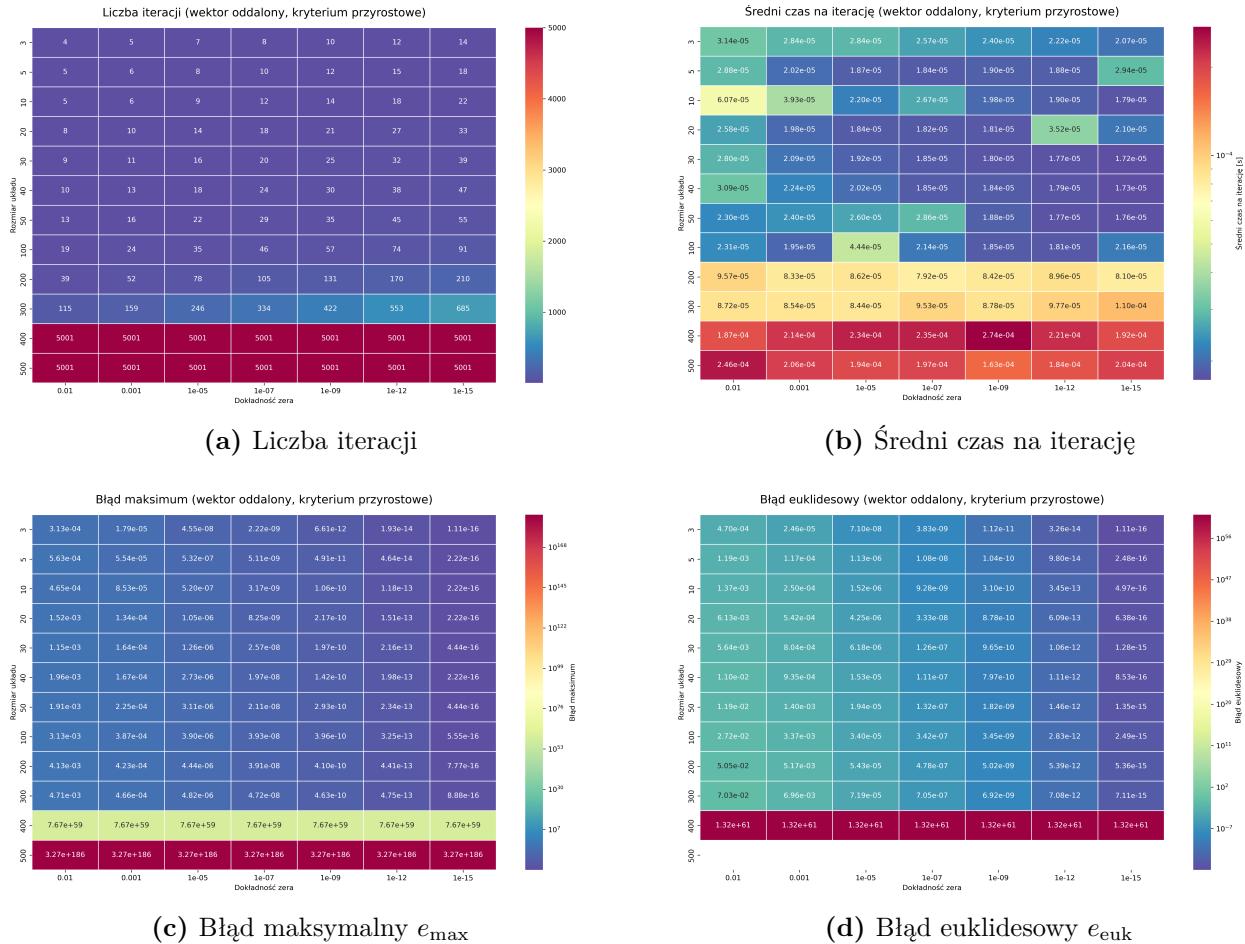
6.1 Kryterium przyrostowe

6.1.1 Wektor zerowy



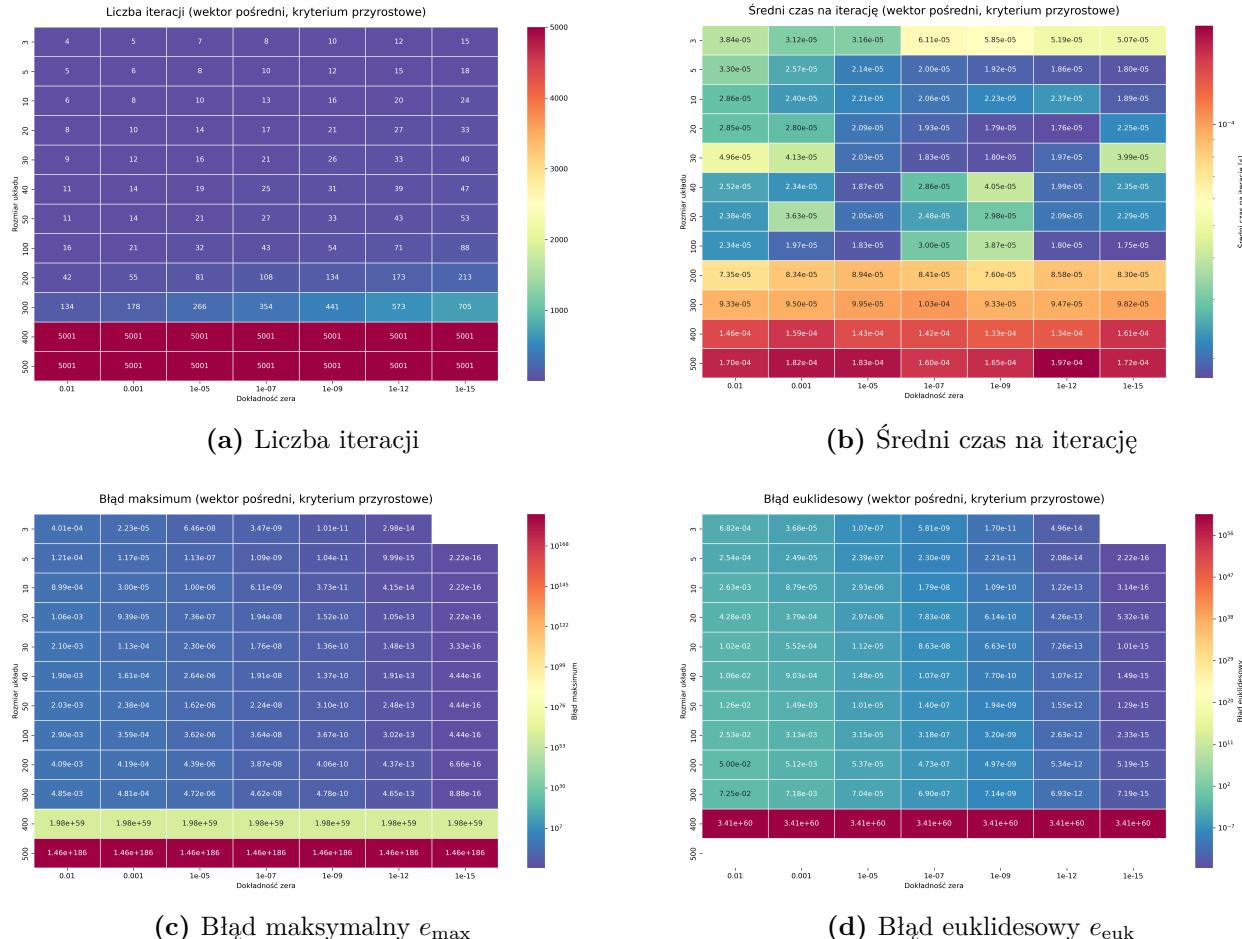
Rysunek 2: Wyniki dla wektora zerowego przy kryterium przyrostowym: (a) liczba iteracji, (b) średni czas na iterację, (c) błąd maksymalny, (d) błąd euklidesowy.

6.1.2 Wektor losowany z $\{\pm 100\}$



Rysunek 3: Wyniki dla wektora ± 100 przy kryterium przyrostowym: (a) liczba iteracji, (b) średni czas na iterację, (c) błąd maksymalny, (d) błąd euklidesowy.

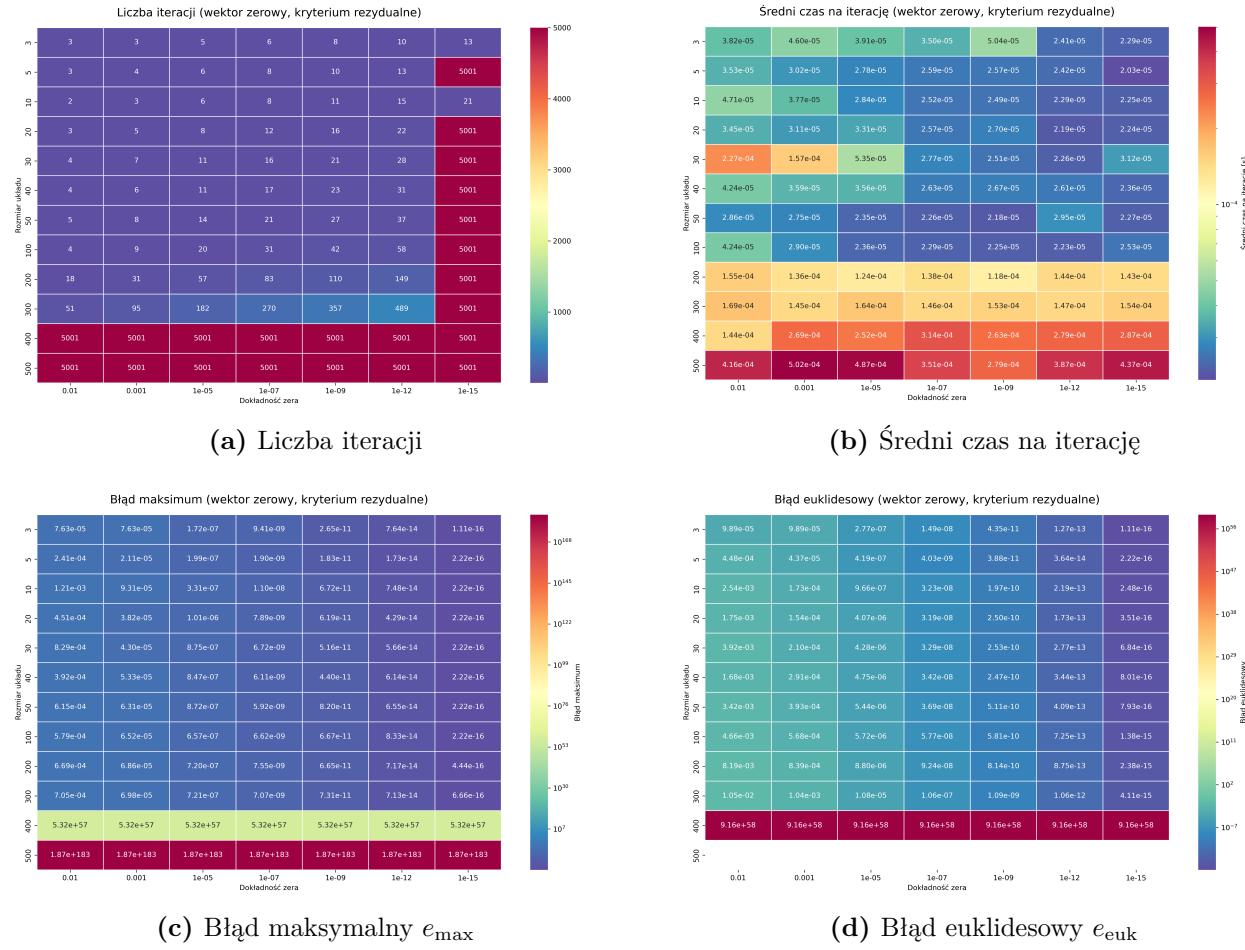
6.1.3 Wektor „pośredni”



Rysunek 4: Wyniki dla wektora „pośredniego” przy kryterium przyrostowym: (a) liczba iteracji, (b) średni czas na iterację, (c) błąd maksymalny, (d) błąd euklidesowy.

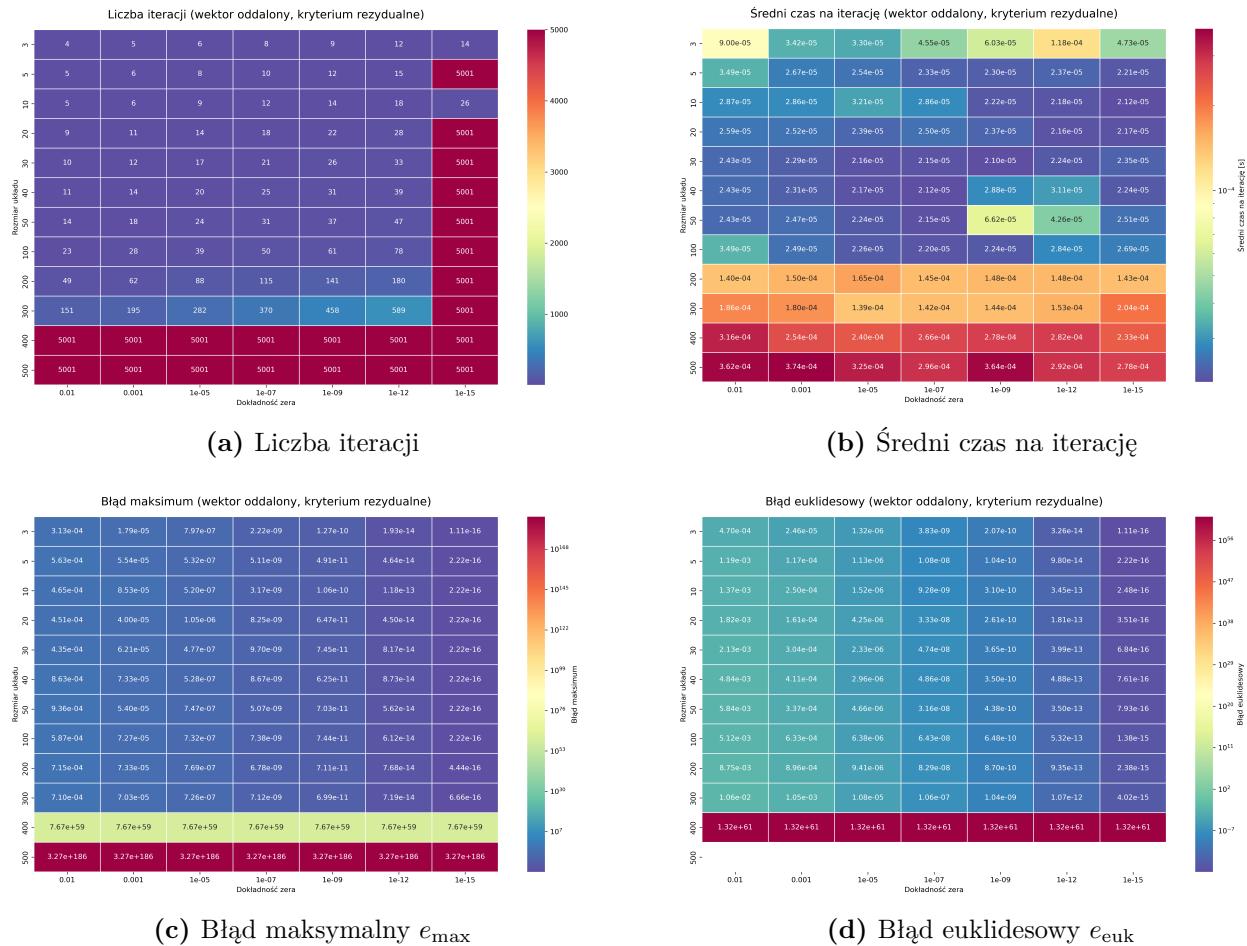
6.2 Kryterium residualne

6.2.1 Wektor zerowy



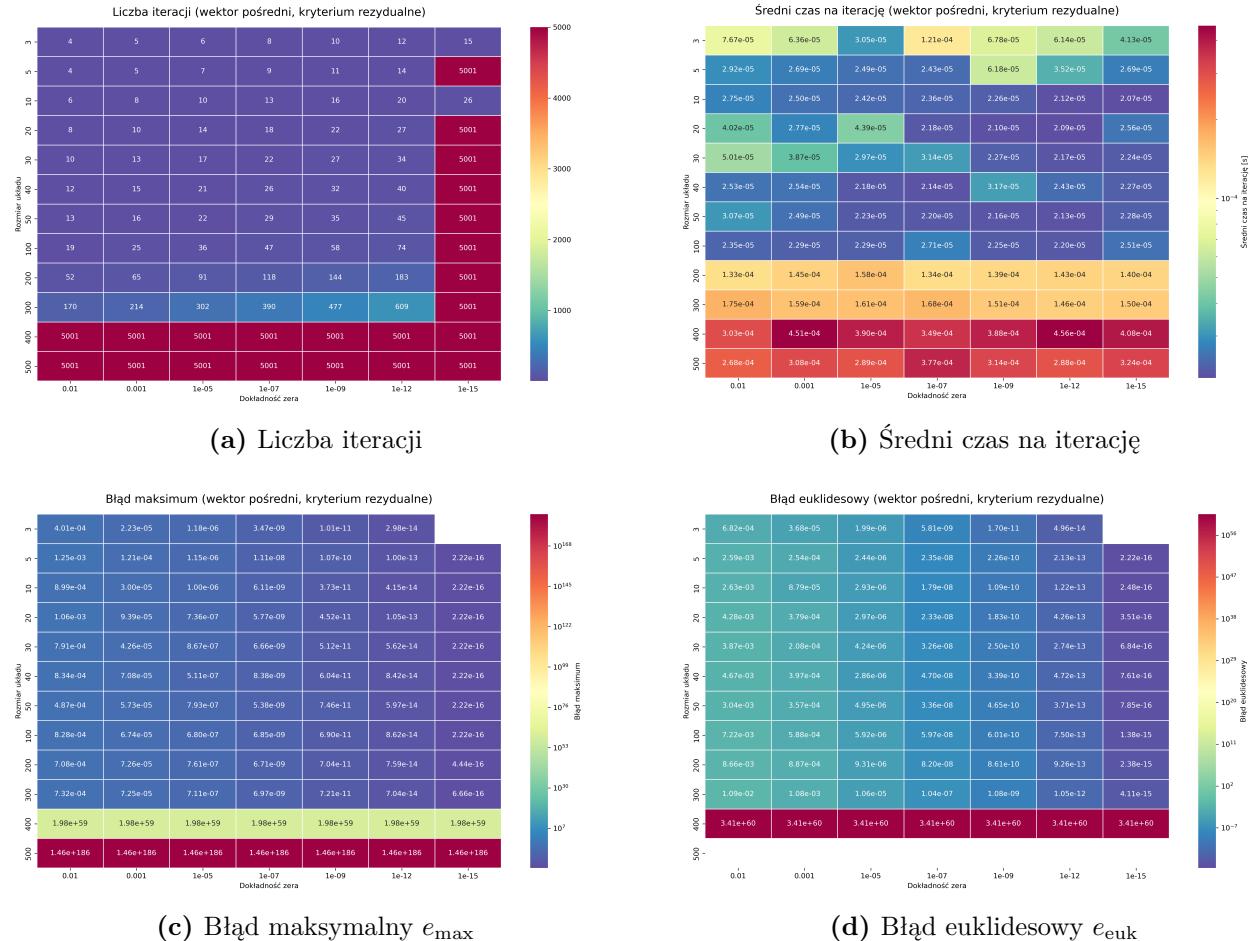
Rysunek 5: Wyniki dla wektora zerowego przy kryterium residualnym: (a) liczba iteracji, (b) średni czas na iterację, (c) błąd maksymalny, (d) błąd euklidesowy.

6.2.2 Wektor losowany z $\{\pm 100\}$



Rysunek 6: Wyniki dla wektora ± 100 przy kryterium residualnym: (a) liczba iteracji, (b) średni czas na iterację, (c) błąd maksymalny, (d) błąd euklidesowy.

6.2.3 Wektor „pośredni”



Rysunek 7: Wyniki dla wektora „pośredniego” przy kryterium residualnym: (a) liczba iteracji, (b) średni czas na iterację, (c) błąd maksymalny, (d) błąd euklidesowy.

7 Opracowanie Danych

Kryterium przyrostowe

Zgodnie z oczekiwaniami dla $n = 400$ oraz $n = 500$ liczba iteracji osiąga wartość 5001 (przy `max_iter=5000` oznacza to przekroczenie limitu), a błędy dla tych rozmiarów są ogromne. Ponadto, zgodnie z przewidywaniami, liczba iteracji rośnie wraz ze wzrostem wymagań co do dokładności oraz rozmiaru macierzy – trudniej osiągnąć wymaganą dokładność, a rosnący promień zbieżności wydłuża czas konwergencji. Średni czas wykonywania pojedynczej iteracji jest krótszy w przypadku mniejszych układów i przy luźniejszych kryteriach dokładności (mniejsze ε). Te spostrzeżenia obserwujemy we wszystkich trzech wariantach wektora początkowego. Dla wektora „pośredniego” przy progach dokładności 10^{-15} udało się „trafić” w rozwiązanie, i zarówno błąd maksymalny ($\|x_{\text{true}} - x^{(N)}\|_\infty$), jak i błąd euklidesowy ($\|x_{\text{true}} - x^{(N)}\|_2$) wynosiły 0.0 (należy jednak ciągle pamiętać o ograniczeniach reprezentacji liczb w arytmetyce `float64`).

Kryterium rezydualne

Podobnie jak w przypadku kryterium przyrostowego, dla $n = 400$ i $n = 500$ liczba iteracji osiąga 5001 (przy ustawnionym limicie `max_iter=5000`), co oznacza, że metoda nie zbiega w zadanej liczbie kroków, a błędy końcowe są bardzo duże. Widzimy również, że wraz ze wzrostem wymagań na dokładność (mniejsze wartości ε) oraz rozmiaru układu n , liczba iteracji systematycznie rośnie. Wzrost rozmiaru n oraz zaostrzenie progu ε powodują znaczący wzrost zarówno całkowitego czasu obliczeń, jak i liczby iteracji. Obserwacje te są spójne dla wszystkich trzech wariantów wektora początkowego.

Warto zaznaczyć, że w przypadku kryterium rezydualnego dla tolerancji $\varepsilon = 10^{-15}$ w większości przypadków (por. Rysunki 5, 6, 7) nie udało się zakończyć iterowania przed osiągnięciem limitu, mimo to uzyskiwano niższe wartości błędów końcowych niż przy kryterium przyrostowym.

Ponadto kryterium rezydualne wykaływało się większym nakładem czasowym na pojedynczą iterację (np. dla $n = 200$ średni czas na iterację wynosił około $1,2 \times 10^{-4}$ s, podczas gdy w kryterium przyrostowym było to około 8×10^{-5} s).

8 Podsumowanie Zagadnienia

Przeprowadzone badania potwierdzają, że zarówno wzrost rozmiaru układu n , jak i zaostrzenie wymagań co do dokładności (mniejsze wartości tolerancji ε) prowadzą do zwiększonej liczby iteracji oraz wydłużenia czasu obliczeń. W szczególności, gdy promień spektralny $\rho(G)$ przekracza 1 (dla $n \geq 362$), metoda iteracyjna nie konwerguje w zadanej liczbie kroków, co skutkuje brakiem poprawnego rozwiązania.

- **Kryterium przyrostowe** generuje mniejszą liczbę iteracji w porównaniu z kryterium rezydualnym, co przekłada się na krótszy całkowity czas obliczeń. Jednak dokładność uzyskiwanych rozwiązań jest niższa (większe błędy końcowe) niż przy kryterium rezydualnym.
- **Kryterium rezydualne** wymaga zwykle większej liczby iteracji i dłuższego czasu pojedynczej iteracji, ale zapewnia lepszą kontrolę błędu końcowego (niższe wartości norm $\|\cdot\|_\infty$ i $\|\cdot\|_2$).
- Wariant „zerowy” ($x^{(0)} = [0, \dots, 0]^\top$) w wielu przypadkach prowadził do najszybszego zbiegania (najmniejszej liczby iteracji) oraz najniższych błędów końcowych, szczególnie dla ostrzejszych tolerancji ($\varepsilon = 10^{-15}$) i niewielkich układów.
- Wariant „pośredni” (losowany ze zbioru $\{\pm 30, \pm 40, \pm 50, \pm 60\}$) dawał nieznacznie wolniejsze zbieganie niż „zerowy”, choć wciąż lepsze niż „oddalony” przy łagodniejszych tolerancjach.
- Wariant „oddalony” (losowany ze zbioru $\{\pm 100\}$) zwykle wymagał największej liczby iteracji i generował wyższe czasy, zwłaszcza dla większych układów i małych ε .