AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Metody Obliczeniowe w Nauce i Technice

Sprawozdanie z metod rozwiązywania układów równań liniowych

Maciej Kmąk
Informatyka WI AGH, II rok
14. Czerwca 2025

1 Cel Ćwiczenia

Celem ćwiczenia jest przeanalizowanie i porównanie wybranych metod rozwiązywania układów równań liniowych postaci $Ax = b$ z punktu widzenia:

1. **Złożoności obliczeniowej i zużycia pamięci** różnych algorytmów oraz pomiar rzeczywistych czasów wykonania i wielkości zaalokowanej pamięci.
2. **Stabilności i zbieżności** – analiza współczynnika uwarunkowania $\kappa(A)$ dla metod bezpośrednich oraz promienia spektralnego $\rho(G)$ macierzy iteracji w metodzie Jacobiego.
3. **Dokładności numerycznej** – ocena wpływu błędów zaokrągleń (na przykładzie precyzji `float32` i `float64` z biblioteki `NumPy`) oraz współczynnika uwarunkowania macierzy $\kappa(A)$ na uzyskane rozwiązania (normy $\|x_{\text{zadany}} - x_{\text{obliczony}}\|_\infty$ i $\|\cdot\|_2$).

Badania obejmowały trzy pod-zadania:

Zadanie 6a Macierze pełne(źle uwarunkowana macierz hilbertowska oraz macierz symetryczna); rozwiązanie metodą eliminacji Gaussa.

- Macierz hilbertowska:

$$a_{ij} = \begin{cases} 1, & i = 1, \\ \dfrac{1}{i+j-1}, & i \neq 1, \end{cases} \qquad i, j = 1, 2, \ldots, n.$$

- Macierz symetryczna:

$$a_{ij} = \begin{cases} \dfrac{2i}{j}, & i \geq j, \\ a_{ji}, & i < j, \end{cases} \qquad i, j = 1, 2, \ldots, n.$$

Zadanie 6b Macierz trójdiagonalna; porównanie eliminacji Gaussa i algorytmu Thomasa.

- Macierz trójdiagonalna:

$$A^{n\times n} = \begin{cases} 6, & i = j, \\ \dfrac{1}{i+4}, & j = i+1, \\ \dfrac{6}{i+4+1}, & j = i-1, \\ 0, & |i-j| > 1. \end{cases}$$

Zadanie 7 Macierze o strukturze $a_{ii} = k$, $a_{ij} = 1/(|i-j|+m)$; rozwiązanie metodą Jacobiego z dwoma kryteriami stopu oraz wyznaczenie promienia spektralnego macierzy iteracji.

- Macierz z zadania:

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

2 Dane techniczne

Doświadczenie zostało przeprowadzone na komputerze osobistym o specyfikacji:

- System Operacyjny: Windows 11 Pro
- Procesor: 12th Gen Intel(R) Core(TM) i5-1235U 1.3 GHz
- Język: Python 3.12

3 Przebieg doświadczenia

Zadanie 6a – macierze źle (hilbertowska) i dobrze uwarunkowane ($2i/j$)

1. **Zakresy wymiarów macierzy n:**
   - *Hilbertowska:* $n = 2, 3, \ldots, 100$.
   - *Macierz* $2i/j$*:* $n \in \{2, 3, \ldots, 100\} \cup \{110, 120, \ldots, 500\}$.
2. **Dla każdego** (n, dtype)**:**
   (a) Generacja wektora wzorcowego $\mathbf{x}_{\text{true}} \in \{\pm 1\}^n$.
   (b) Budowa macierzy A i obliczenie $\mathbf{b} = A\mathbf{x}_{\text{true}}$.
   (c) Wyznaczenie współczynnika uwarunkowania
   $$\kappa(A) = \|A\|_\infty \, \|A^{-1}\|_\infty.$$
   (d) Rozwiązanie $A\mathbf{x} = \mathbf{b}$ metodą eliminacji Gaussa.
   (e) Obliczenie błędów:
   $$e_{\max} = \|\mathbf{x}_{\text{true}} - \mathbf{x}_{\text{calc}}\|_\infty, \qquad e_{\text{euk}} = \|\mathbf{x}_{\text{true}} - \mathbf{x}_{\text{calc}}\|_2.$$

Zadanie 6b – macierz trójdiagonalna

1. **Zakresy wymiarów:** $n \in \{2, \ldots, 100\} \cup \{110, 120, \ldots, 1000\}$.
2. **Dla każdego** (n, dtype)**:**
   (a) Generacja wektorów macierzy oraz losowego $\mathbf{x}_{\text{true}} \in \{\pm 1\}^n$.
   (b) Obliczenie $\mathbf{b} = A_{\text{tri}}\, \mathbf{x}_{\text{true}}$.
   (c) Rozwiązanie układu dwiema metodami:
   - *Algorytm Thomasa* (złożoność czasowa $O(n)$, pamięciowa $O(n)$),
   - *Eliminacja Gaussa* (złożoność czasowa $O(n^3)$, pamięciowa $O(n^2)$).

   (d) Pomiar czasu wykonania i zużycia pamięci (moduł `tracemalloc`).
   (e) Obliczenie błędów $e_{\max}$, e_{euk}.

Zadanie 7 – metoda Jacobiego

1. **Badanie progu zbieżności:**
   - Konstruowano macierz A dla $n = 1, \ldots, 1000$.
   - Dla każdego n obliczano promień spektralny macierzy iteracji Jacobiego.
   - Zidentyfikowano pierwszy n_0, dla którego $\rho(G) \geq 1$.
2. **Zawężenie badania do** $n \in \{3, 5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500\}$**.**
3. **Schemat pomiarów:**
   - Kryteria stopu:
     $$\|x^{(k+1)} - x^{(k)}\|_\infty < \varepsilon \quad \text{lub} \quad \|Ax^{(k)} - b\|_\infty < \varepsilon,$$
     dla $\varepsilon \in \{10^{-2}, 10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}, 10^{-12}, 10^{-15}\}$.
   - Trzy wektory startowe: zerowy, losowy z $\{\pm 100\}$ i „pośredni" z $\{\pm 30, \pm 40, \pm 50, \pm 60\}$.
4. **Dla każdej kombinacji** $(n, \varepsilon,$ **start):**
   (a) Mierzono liczbę iteracji do zbieżności i średni czas jednej iteracji.
   (b) Obliczano błędy końcowe $e_{\max}$, e_{euk}.

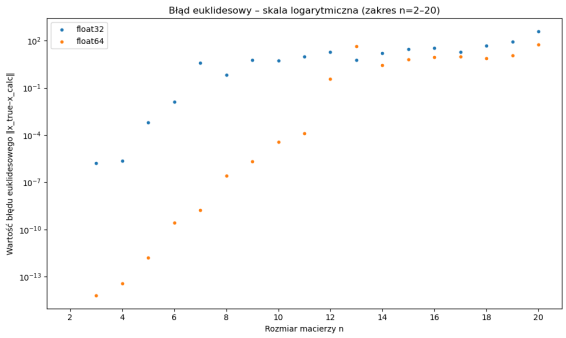
4 Wyniki Doświadczenia

Wyniki doświadczenia 6a cz.1 – macierz źle uwarunkowana

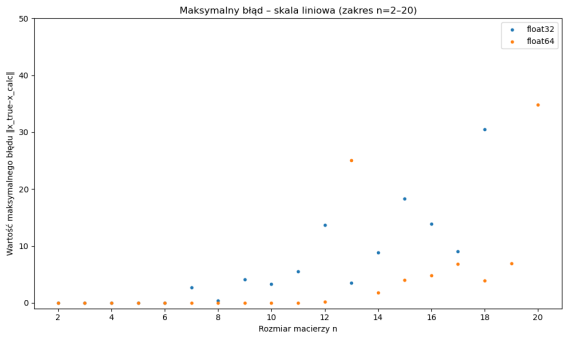
W tabeli 1 oraz na rysunkach 1a,1b i 2 przedstawiono zebrane dla zakresu $n = 2$–20 wartości błędu maksymalnego i euklidesowego oraz współczynnika uwarunkowania macierzy dla precyzji `float32` i `float64`.

Tabela 1: Zadanie 6a cz.1: wyniki eksperymentu dla $n \in \{2, \ldots, 20\}$ (float32 vs. float64)

	float32			**float64**		
n	$e_{\max}$	e_{euk}	κ	$e_{\max}$	e_{euk}	κ
2	0.000e+00	0.000e+00	8.000e+00	0.000e+00	0.000e+00	8.000e+00
3	1.311e-06	1.660e-06	2.400e+02	5.107e-15	6.489e-15	2.400e+02
4	1.907e-06	2.322e-06	1.530e+04	2.975e-14	3.767e-14	1.530e+04
5	4.644e-04	6.392e-04	7.435e+05	1.125e-12	1.559e-12	7.431e+05
6	8.652e-03	1.298e-02	3.266e+07	1.807e-10	2.678e-10	3.216e+07
7	2.677e+00	3.849e+00	1.950e+09	1.162e-09	1.670e-09	1.417e+09
8	4.133e-01	6.608e-01	1.250e+10	1.795e-07	2.677e-07	5.566e+10
9	4.133e+00	5.855e+00	1.487e+11	1.318e-06	2.150e-06	2.019e+12
10	3.290e+00	5.326e+00	1.956e+10	2.235e-05	3.545e-05	8.025e+13
11	5.538e+00	9.358e+00	9.681e+09	8.048e-05	1.328e-04	3.014e+15
12	1.365e+01	1.939e+01	1.139e+10	2.253e-01	3.769e-01	1.090e+17
13	3.458e+00	6.108e+00	4.776e+10	2.501e+01	4.246e+01	3.190e+18
14	8.869e+00	1.603e+01	2.123e+10	1.747e+00	2.809e+00	5.476e+18
15	1.831e+01	2.950e+01	2.395e+10	4.037e+00	6.354e+00	6.797e+18
16	1.388e+01	3.289e+01	6.037e+10	4.837e+00	8.773e+00	7.414e+18
17	9.063e+00	1.842e+01	1.511e+11	6.808e+00	9.786e+00	9.695e+18
18	3.052e+01	4.764e+01	7.394e+10	3.891e+00	7.472e+00	5.371e+19
19	5.168e+01	8.536e+01	7.890e+10	6.956e+00	1.169e+01	3.423e+19
20	2.125e+02	3.766e+02	8.415e+10	3.479e+01	5.495e+01	5.351e+20



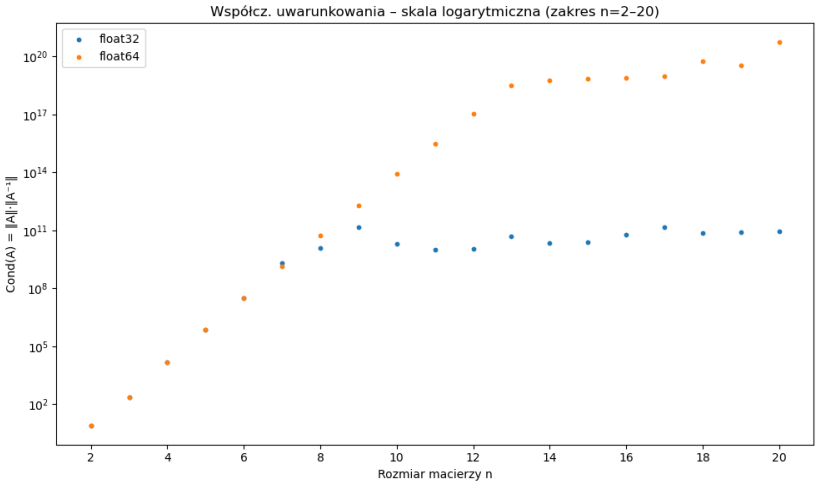
(a) Norma euklidesowa - skala logarytmiczna



(b) Norma maksimum - skala liniowa

Rysunek 1: Wykresy błędów: euklidesowego i maksimum $\|x_{\text{true}} - x_{\text{calc}}\|$ dla precyzji float32 i float64

Macierz A w zadaniu 1 z ćwiczenia 6a to zmodyfikowana macierz Hilberta, w której pierwszy wiersz składa się z samych jedynek ($a_{1j} = 1$), a pozostałe elementy odpowiadają klasycznej definicji macierzy Hilberta. Taka konstrukcja prowadzi do silnie złego uwarunkowania, co możemy zauważyć także na rys. 2.



Rysunek 2: Współczynnik uwarunkowania $\kappa(A)$ (skala logarytmiczna)

Uwarunkowanie i rola $\kappa(A)$. Na rysunku 2 możemy zauważyć, że wartość współczynnika $\kappa(A)$ rośnie bardzo szybko z n, osiągając rzędy 10^{10} dla `float32` oraz 10^{20} dla `float64`. Skorelowane jest to z gwałtownym wzrostem błędu maksimum: dla precyzji `float32` błąd zaczyna istotnie rosnąć już od $n \approx 7$, natomiast dla `float64` dopiero od $n \approx 13$ (por. Tabela 1, kolumna $e_{\max}$). Możemy więc wnioskować, że $\kappa(A)$ stanowi wiarygodny wskaźnik prognozowania utraty dokładności obliczeń.

Wpływ precyzji. Zastosowanie `float64` przesuwa próg istotnego wzrostu błędów w porównaniu do `float32`; niemniej, w skrajnie źle uwarunkowanych przypadkach narastanie błędów staje się zauważalne stosunkowo szybko (w badanym układzie przy $n \approx 13$) pomimo użycia podwójnej precyzji.

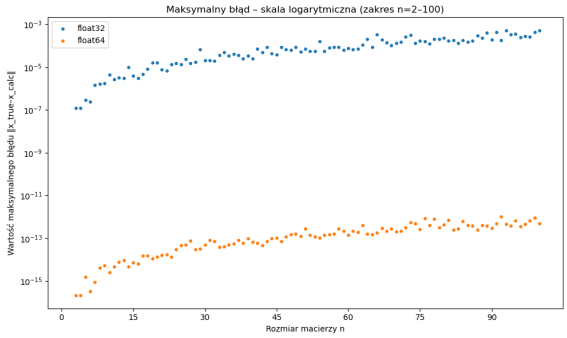
Wyniki doświadczenia 6a cz.2 – macierz dobrze uwarunkowana

W tabeli 2 oraz na rysunkach 3a i 3b przedstawiono wyniki dla zakresu

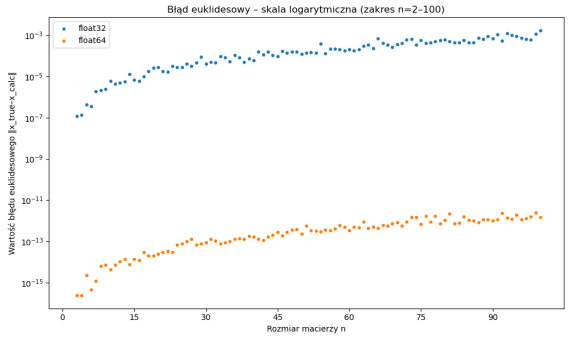
$$n \in \{2, 3, \ldots, 20, 25, 30, 40, 50, 75, 100, 150, 200\}$$

Tabela 2: Zadanie 6a cz.2: wyniki eksperymentu dla $n \in \{2, 3, 4, 5, 20, 25, 30, 40, 50, 75, 100, 150, 200\}$ (float32 vs. float64)

	float32			**float64**		
n	$e_{\max}$	e_{euk}	κ	$e_{\max}$	e_{euk}	κ
2	0.000e+00	0.000e+00	3.000e+00	0.000e+00	0.000e+00	3.000e+00
3	1.192e-07	1.192e-07	8.667e+00	2.220e-16	2.483e-16	8.667e+00
4	1.192e-07	1.333e-07	1.650e+01	2.220e-16	2.483e-16	1.650e+01
5	2.980e-07	4.172e-07	2.680e+01	1.554e-15	2.254e-15	2.680e+01
20	1.669e-05	2.821e-05	4.725e+02	1.332e-14	2.423e-14	4.725e+02
25	1.359e-05	2.715e-05	7.424e+02	4.796e-14	7.926e-14	7.424e+02
30	2.128e-05	3.960e-05	1.073e+03	4.996e-14	9.060e-14	1.073e+03
40	2.551e-05	5.840e-05	1.916e+03	7.017e-14	1.665e-13	1.916e+03
50	5.406e-05	1.249e-04	3.002e+03	1.280e-13	2.335e-13	3.002e+03
75	1.726e-04	5.749e-04	6.778e+03	2.736e-13	6.868e-13	6.778e+03
100	5.060e-04	1.644e-03	1.207e+04	4.929e-13	1.527e-12	1.207e+04
150	7.410e-04	2.420e-03	2.720e+04	1.751e-12	5.864e-12	2.720e+04
200	1.211e-03	4.152e-03	4.841e+04	1.390e-12	6.411e-12	4.841e+04



(a) Norma maksimum — skala logarytmiczna



(b) Norma euklidesowa — skala logarytmiczna

Rysunek 3: Błędy obliczeń $\|x_{\text{true}} - x_{\text{calc}}\|$ dla `float32` i `float64`

Błędy numeryczne. Na logarytmicznych wykresach normy maksimum i euklidesowej widać stabilny przebieg błędów: krzywe dla `float32` są przesunięte o stałą wartość w górę (rzędy wielkości wyższe niż dla `float64`), niemniej nie pojawiają się nagłe skoki ani gwałtowne odchylenia.

Uwarunkowanie i rola $\kappa(A)$. Współczynnik $\kappa(A)$ rośnie stopniowo i bez wyraźnych skoków, co potwierdza dobre uwarunkowanie macierzy – zgodnie z przewidywaniami nie obserwuje się istotnej utraty dokładności w rozwiązywaniu układów. Ponadto, w Tabeli 2 widać, że wartości $\kappa(A)$ dla precyzji `float32` i `float64` są identyczne.

Wyniki doświadczenia 6b – algorytm Thomasa vs. eliminacja Gaussa

Poniżej zestawiono wybrane tabele z wynikami obliczeń dla układów o rozmiarach

$$n \in \{2, 3, \ldots, 10, 20, 50, 100, 150, 200, 300, 400, 500, 750, 1000\},$$

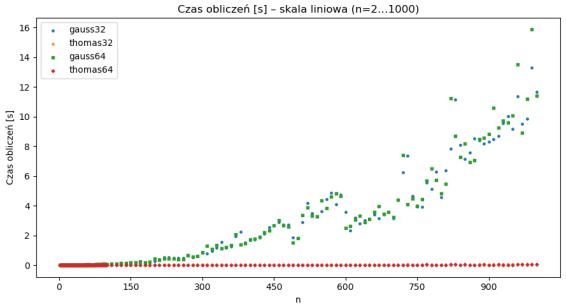
porównujące algorytm Thomasa i pełną eliminację Gaussa.

Tabela 3: Wyniki eksperymentu – algorytm Thomasa

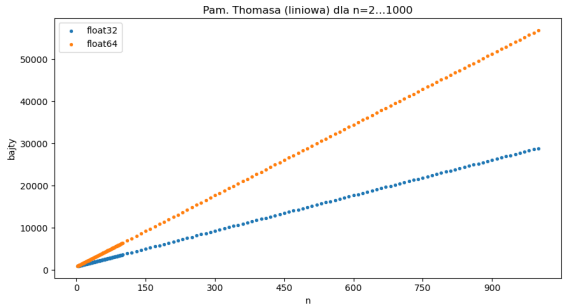
	float32				**float64**			
n	$e_{\max}$	e_{euk}	t [s]	Pamięć [B]	$e_{\max}$	e_{euk}	t [s]	Pamięć [B]
2	0.000e+00	0.000e+00	7.129e-04	9.500e+02	0.000e+00	0.000e+00	4.980e-05	9.600e+02
3	5.960e-08	5.960e-08	6.840e-05	9.120e+02	1.110e-16	1.110e-16	9.160e-05	9.840e+02
4	1.192e-07	1.192e-07	9.850e-05	9.400e+02	1.110e-16	1.110e-16	7.970e-05	1.040e+03
5	1.192e-07	1.333e-07	9.850e-05	9.680e+02	2.220e-16	2.220e-16	9.990e-05	1.096e+03
6	1.192e-07	1.192e-07	1.182e-04	9.960e+02	2.220e-16	3.331e-16	1.187e-04	1.152e+03
7	1.192e-07	1.788e-07	1.486e-04	1.024e+03	2.220e-16	2.719e-16	1.489e-04	1.208e+03
8	1.192e-07	1.788e-07	1.608e-04	1.052e+03	1.110e-16	1.923e-16	1.582e-04	1.264e+03
9	1.192e-07	2.230e-07	2.812e-04	1.080e+03	2.220e-16	3.846e-16	1.920e-04	1.320e+03
10	5.960e-08	5.960e-08	2.036e-04	1.108e+03	2.220e-16	2.937e-16	3.152e-04	1.376e+03
20	1.192e-07	2.308e-07	3.916e-04	1.388e+03	2.220e-16	3.331e-16	3.910e-04	1.936e+03
50	1.788e-07	3.909e-07	1.165e-03	2.228e+03	2.220e-16	7.022e-16	1.235e-03	3.616e+03
100	1.192e-07	6.166e-07	1.894e-03	3.628e+03	2.220e-16	1.047e-15	2.164e-03	6.416e+03
200	1.192e-07	7.633e-07	4.699e-03	6.428e+03	3.331e-16	1.506e-15	5.264e-03	1.202e+04
300	1.788e-07	9.629e-07	7.446e-03	9.308e+03	3.331e-16	1.720e-15	1.083e-02	1.770e+04
400	1.788e-07	1.132e-06	9.170e-03	1.211e+04	2.220e-16	1.769e-15	1.462e-02	2.330e+04
500	1.788e-07	1.176e-06	1.541e-02	1.491e+04	3.331e-16	2.161e-15	1.217e-02	2.890e+04
750	1.192e-07	1.507e-06	2.395e-02	2.191e+04	3.331e-16	2.753e-15	2.380e-02	4.290e+04
1000	1.788e-07	1.761e-06	5.323e-02	2.891e+04	3.331e-16	3.329e-15	3.504e-02	5.690e+04

Tabela 4: Wyniki eksperymentu – eliminacja Gaussa

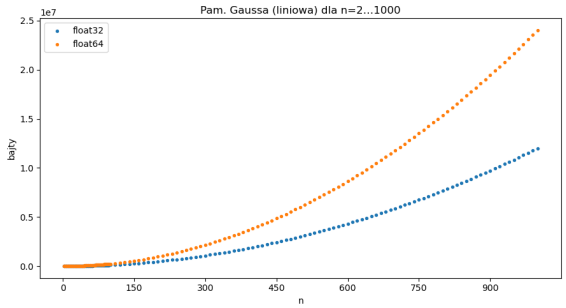
	float32				**float64**			
n	$e_{\max}$	e_{euk}	t [s]	Pamięć [B]	$e_{\max}$	e_{euk}	t [s]	Pamięć [B]
2	0.000e+00	0.000e+00	5.003e-04	1.365e+03	0.000e+00	0.000e+00	1.368e-04	1.384e+03
3	5.960e-08	5.960e-08	2.748e-04	1.376e+03	1.110e-16	1.110e-16	1.781e-04	1.488e+03
4	1.192e-07	1.192e-07	2.613e-04	1.444e+03	1.110e-16	1.110e-16	2.167e-04	1.624e+03
5	1.192e-07	1.333e-07	3.871e-04	1.528e+03	2.220e-16	2.220e-16	3.023e-04	1.792e+03
6	1.192e-07	1.192e-07	4.195e-04	1.628e+03	2.220e-16	3.331e-16	3.638e-04	1.992e+03
7	1.192e-07	1.788e-07	5.984e-04	1.744e+03	2.220e-16	2.719e-16	5.345e-04	2.224e+03
8	1.192e-07	1.788e-07	7.069e-04	1.876e+03	1.110e-16	1.923e-16	9.618e-04	2.881e+03
9	1.192e-07	2.230e-07	1.080e-03	2.024e+03	2.220e-16	3.846e-16	6.944e-04	2.784e+03
10	5.960e-08	5.960e-08	9.397e-04	2.188e+03	2.220e-16	2.937e-16	8.721e-04	3.184e+03
20	1.192e-07	2.308e-07	2.707e-03	5.584e+03	2.220e-16	3.331e-16	2.573e-03	1.054e+04
50	1.788e-07	3.909e-07	1.554e-02	3.102e+04	2.220e-16	7.022e-16	1.490e-02	6.142e+04
100	1.192e-07	6.166e-07	5.359e-02	1.214e+05	2.220e-16	1.047e-15	5.651e-02	2.422e+05
200	1.192e-07	7.633e-07	2.736e-01	4.822e+05	3.331e-16	1.506e-15	4.543e-01	9.638e+05
300	1.788e-07	9.629e-07	8.278e-01	1.083e+06	3.331e-16	1.720e-15	8.710e-01	2.165e+06
400	1.788e-07	1.132e-06	1.678e+00	1.924e+06	2.220e-16	1.769e-15	1.711e+00	3.847e+06
500	1.788e-07	1.176e-06	1.821e+00	3.005e+06	3.331e-16	2.161e-15	1.797e+00	6.009e+06
750	1.192e-07	1.507e-06	4.008e+00	6.757e+06	3.331e-16	2.753e-15	3.948e+00	1.351e+07
1000	1.788e-07	1.761e-06	1.166e+01	1.201e+07	3.331e-16	3.329e-15	1.143e+01	2.402e+07



Rysunek 4: Porównanie czasu obliczeń [s] (skala liniowa) dla metod Thomasa i Gaussa.

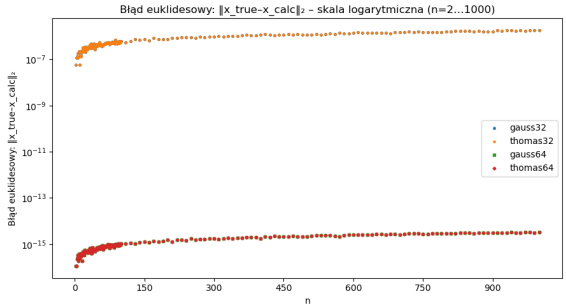


(a) Algorytm Thomasa



(b) Eliminacja Gaussa

Rysunek 5: Porównanie zużycia pamięci (w bajtach) w zależności od n.



Rysunek 6: Porównanie błędu euklidesowego $e_{\text{euk}} = \|x_{\text{true}} - x_{\text{calc}}\|_2$ w skali logarytmicznej.

Czas obliczeń: Z porównania (Rys. 4) wynika, że czas rozwiązania układu trójdiagonalnego algorytmem Thomasa rośnie liniowo wraz ze wzrostem rozmiaru n, co potwierdza jego złożoność $\mathcal{O}(n)$. Natomiast pełna eliminacja Gaussa wykazuje typowy dla siebie przebieg sześcienny $\mathcal{O}(n^3)$, przez co dla większych n staje się znacznie wolniejsza.

Zużycie pamięci: Rysunek 5a pokazuje, że algorytm Thomasa wykorzystuje pamięć proporcjonalnie do n (trzy wektory długości n), natomiast eliminacja Gaussa (rys. 5b) – ze względu na pełne przechowywanie macierzy $n \times n$ – wymaga przestrzeni kwadratowej $\mathcal{O}(n^2)$. Dodatkowo wyższa precyzja (`float64`) skutkuje dwukrotnym zwiększeniem zużycia pamięci w obu metodach.

Dokładność rozwiązań: Jak ilustruje wykres błędu euklidesowego (Rys.6), obie metody osiągają porównywalną dokładność: wartości $e_{\max}$ i e_{euk} nie różnią się pomiędzy algorytmem Thomasa a eliminacją Gaussa. Jedynym czynnikiem wpływającym na wielkość błędów jest zastosowana precyzja zmiennoprzecinkowa (`float32` vs. `float64`) a nie sam algorytm.

Wyniki doświadczenia 7 – metoda Jacobiego

W pierwszym etapie badania obliczono promień spektralny $\rho(G)$ dla wybranych rozmiarów układu n. Wyniki przedstawiono w Tabeli 5. Z Tabeli 5b widać, że promień spektralny po raz pierwszy przekracza 1 przy $n = 362$. Na Rysunku 7 zaprezentowano przebieg $\rho(G)$ w całym badanym zakresie oraz powiększenie fragmentu wokół $n \approx 362$. Badanie promienia spektralnego macierzy iteracyjnej G wykazało, że $\rho(G) < 1$ dla $n \leq 300$, a $\rho(G) > 1$ już od $n = 362$. W dalszych testach skupiono się więc na „zbieżnych" układach $n \leq 300$ oraz „niezbieżnych" $n \geq 400$, rozwiązując je metodą Jacobiego w precyzji `float64`.

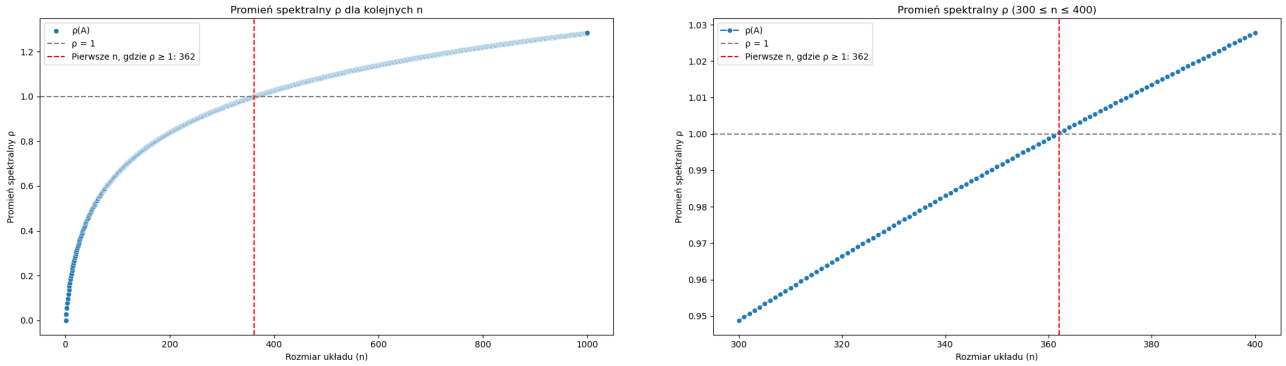
(a) $n \in \{100, 200, 300, 400\}$

n	$\rho(G)$
100	0.6584
200	0.8392
300	0.9488
400	1.0277

(b) $\rho(G)$ dla $360 \leq n \leq 363$

n	$\rho(G)$
360	0.99872
361	0.99948
362	1.00024
363	1.00100

Tabela 5: Promień spektralny $\rho(G)$: porównanie wybranych wartości

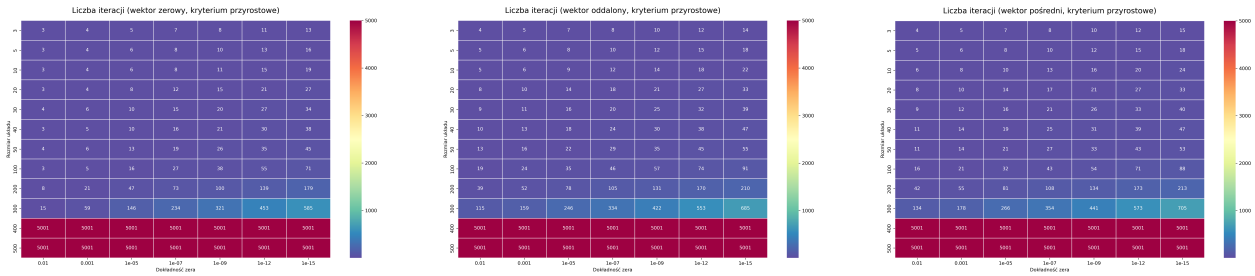


(a) Zakres $1 \leq n \leq 1000$ **(b)** Fragment $300 \leq n \leq 400$

Rysunek 7: Promień spektralny $\rho(G)$ w zależności od n.

Jako kryteria stopu przyjęto normę przyrostu $\|x^{(k+1)} - x^{(k)}\|_\infty < \varepsilon$ i residualną $\|Ax^{(k)} - b\|_\infty < \varepsilon$, oraz trzy typy wektora początkowego: zerowy, losowy z $\{\pm 100\}$ i „pośredni" z $\{\pm 30, \pm 40, \pm 50, \pm 60\}$.

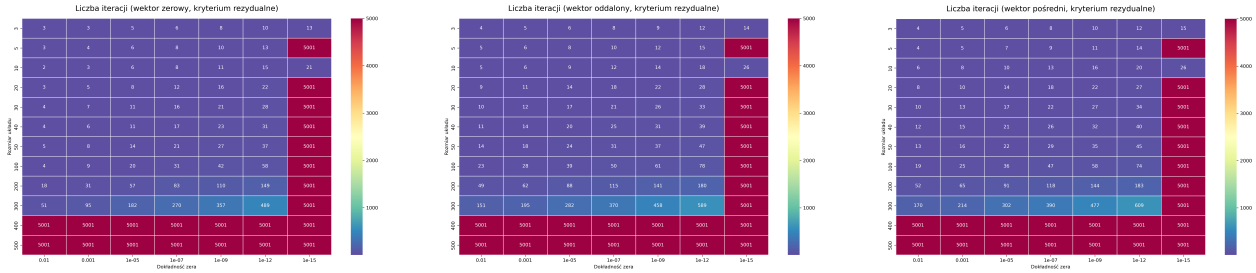
Kryterium przyrostowe – liczba iteracji



(a) Wariant zerowy **(b)** Wariant ± 100 **(c)** Wariant „pośredni"

Rysunek 8: Liczba iteracji do spełnienia kryterium przyrostowego $\|x^{(k+1)} - x^{(k)}\|_\infty < \varepsilon$.

Kryterium residualne – liczba iteracji

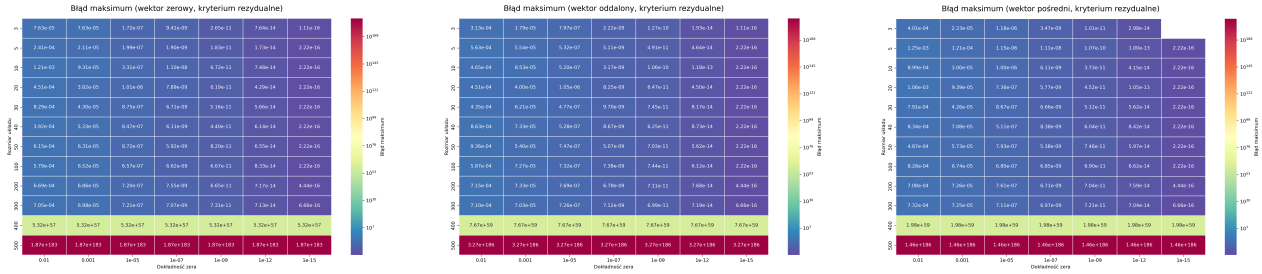


(a) Wariant zerowy **(b)** Wariant ±100 **(c)** Wariant „pośredni”

Rysunek 9: Liczba iteracji do spełnienia kryterium residualnego $\|Ax^{(k)} - b\|_\infty < \varepsilon$.

Warto zauważyć, że dla kryterium residualnego i tolerancji $\varepsilon = 10^{-15}$ w większości przypadków (por. Rys. 9a, 9b, 9c) iteracje dochodziły do ustawionego limitu, mimo to końcowe wartości błędu maksymalnego $e_{\max}$ były zgodne z ograniczeniami arytmetyki liczb zmiennoprzecinkowych (zob. Rysunek 10).

Kryterium residualne – błąd maksymalny $e_{\max}$



(a) Wariant zerowy **(b)** Wariant ±100 **(c)** Wariant „pośredni”

Rysunek 10: Błąd maksymalny $e_{\max} = \|x_{\text{true}} - x^{(N)}\|_\infty$ przy kryterium residualnym.

Ostateczne wnioski płynące z analizy promienia spektralnego oraz przebiegu iteracji przedstawiają się następująco:

- Dla $n \leq 300$ ($\rho(G) < 1$) metoda Jacobiego konwerguje, zaś gdy $\rho(G) > 1$ (od $n = 362$), brak jest zbieżności w zadanym limicie kroków.
- Kryterium przyrostowe generuje zwykle mniejszą liczbę iteracji niż residualne, ale prowadzi do wyższych błędów końcowych.
- Kryterium residualne zapewnia lepszą kontrolę dokładności kosztem większej liczby iteracji i dłuższego czasu pojedynczej iteracji.
- Wariant zerowy wektora startowego daje najszybszą zbieżność i najniższe błędy, natomiast wektor „oddalony” (±100) wymaga najwięcej iteracji, zwłaszcza przy mniejszych tolerancjach.

5 Podsumowanie Zagadnienia

Badania obejmowały trzy problemy: (6a) układy o pełnych macierzach dobrze i źle uwarunkowanych, (6b) układy trójdiagonalne, (7) układy rozwiązywane metodą Jacobiego. Poniżej syntetyzujemy najważniejsze wnioski.

Uwarunkowanie a metody bezpośrednie (Zadanie 6a)

- **Macierz hilbertowska (złe uwarunkowanie).** Współczynnik uwarunkowania osiągał rzędy $10^{10} - 10^{20}$; odpowiadało to lawinowemu wzrostowi błędu $e_{\max}$ od $n \approx 7$ (`float32`) i $n \approx 13$ (`float64`). Podwójna precyzja tylko *opóźniała* katastrofę numeryczną – nie usuwała jej przy skrajnie źle uwarunkowanych macierzach.
- **Macierz $2i/j$ (dobre uwarunkowanie).** $\kappa(A)$ rosło wolno (maks. $\approx 10^5$), a profil błędów pozostawał gładki; różnice między `float32` i `float64` sprowadzały się do stałego przesunięcia o kilka rzędów wielkości, zgodnego z oczekiwaniami dla różnych precyzji liczb zmiennoprzecinkowych.

Współczynnik uwarunkowania $\kappa(A)$ jest wiarygodnym predyktorem jakości rozwiązania dla metod bezpośrednich: nagły skok κ zwiastuje równie nagły wzrost błędu, zaś wysoka precyzja nie zastąpi dobrego uwarunkowania macierzy.

Macierze trójdiagonalne – Thomas vs. Gauss (Zadanie 6b)

- **Złożoność.** Algorytm Thomasa potwierdził złożoność czasową i pamięciową $O(n)$; pełna eliminacja Gaussa – odpowiednio $O(n^3)$ i $O(n^2)$. Przy $n = 1000$ Thomas był ok. **200**$\times$ szybszy i **400**$\times$ oszczędniejszy pamięciowo.
- **Dokładność.** Obie metody dawały identyczne wartości $e_{\max}$ oraz e_{euk}. O błędzie decydowała wyłącznie precyzja (`float32`/`float64`), a nie dobór algorytmu.

W przypadku układów trójdiagonalnych algorytm Thomasa pozwala uzyskać porównywalną dokładność rozwiązań przy znacznie niższych kosztach obliczeniowych i pamięciowych niż pełna eliminacja Gaussa.

Metoda Jacobiego – granice zbieżności (Zadanie 7)

Promień spektralny i zbieżność

Promień spektralny $\rho(G)$ stanowi dobry wskaźnik zbieżności metod iteracyjnych – układy konwergują jedynie wtedy, gdy $\rho(G) < 1$.

Kryteria stopu i wektory startowe

- **Kryterium przyrostowe** wymagało mniej iteracji, lecz kończyło z większymi błędami.
- **Kryterium residualne** zapewniało niższy błąd kosztem większej liczby iteracji i dłuższego czasu na iterację.
- Wektor startowy *zerowy* zbiegał najszybciej i z najmniejszą liczbą iteracji, podczas gdy wektor ± 100 wymagał największej liczby kroków, szczególnie przy mniejszych tolerancjach $\varepsilon \leq 10^{-9}$.

Ograniczenia arytmetyki zmiennoprzecinkowej

Przy kryterium residualnym i tolerancji $\varepsilon = 10^{-15}$ iteracje w większości przypadków dochodziły do ustawionego limitu (por. Rys. 9 i 10), mimo to uzyskane wartości błędu maksymalnego $e_{\max}$ pozostawały w granicach około 10^{-14}, co uwidacznia wpływ ograniczonej precyzji zmiennoprzecinkowej.