

Rozwiązywanie układów równań liniowych metodami iteracyjnymi

Łukasz Wala

*AGH, Wydział Informatyki, Elektroniki i Telekomunikacji
Metody Obliczeniowe w Nauce i Technice 2021/2022*

Kraków, 1 czerwca 2022

1 Problem 1

1.1 Opis problemu

Dany jest układ równań liniowych $\mathbf{Ax}=\mathbf{b}$. Elementy macierzy \mathbf{A} o wymiarze $n \times n$ są określone wzorem:

$$\begin{cases} a_{i,i} = k \\ a_{i,j} = \frac{1}{|i-j|+m} \text{ dla } i \neq j \end{cases} \quad i, j = 1, \dots, n$$

Gdzie $k = 8$, $m = 3$.

Za wektor \mathbf{x} przyjęta zostanie dowolna n -elementowa permutacja ze zbioru $\{1, -1\}$ i obliczony zostanie wektor \mathbf{b} . Układ zostanie rozwiązany metodą Jakobi. Obliczenia zostaną wykonane dla różnych n , dla różnych wektorów początkowych oraz różnych wartości ρ w kryteriach stopu. Wyznaczone zostaną: liczba iteracji, różnica w czasie obliczeń dla obu kryteriów stopu. Sprawdzona zostanie dokładność obliczeń.

Użyte kryteria stopu (norma euklidesowa):

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

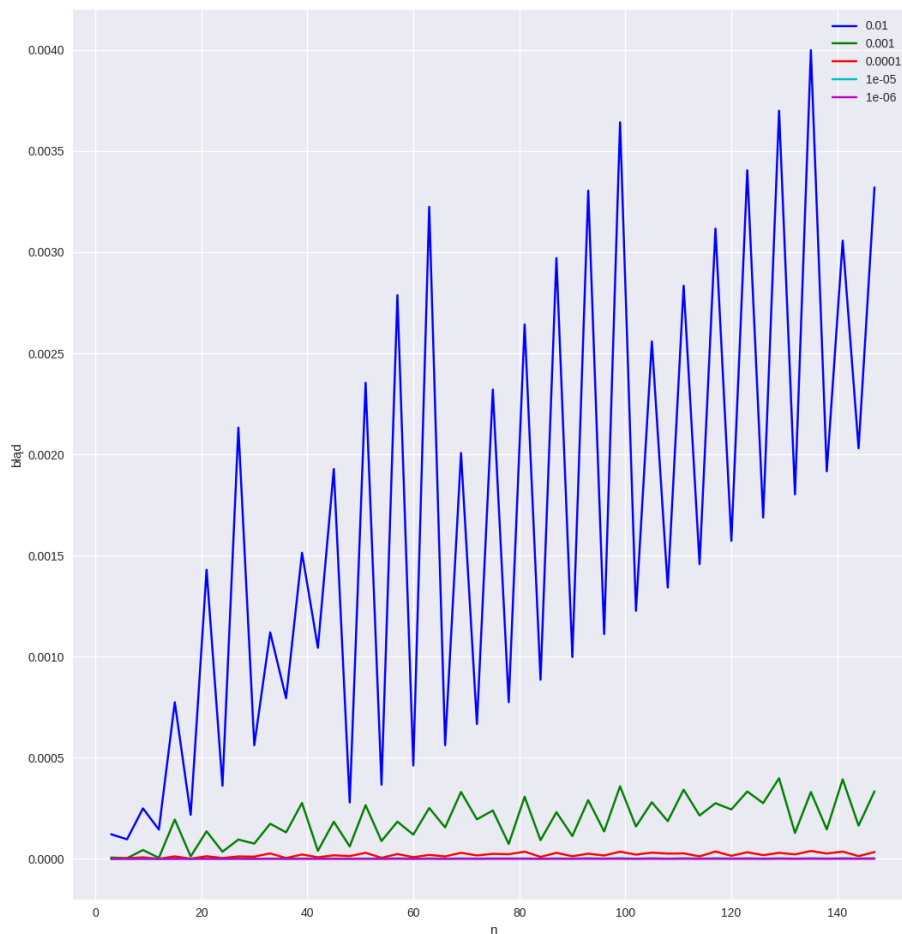
1.2 Opracowanie problemu

Program użyty do rozwiązania układu został napisany w języku Python z użyciem pakietu numpy. Poniżej tabele zależności liczby n oraz precyzji (błąd wyniku, liczba iteracji oraz czas obliczeń) dla wektora początkowego zawierającego same zera oraz rzeczywiste rozwiązanie to $\mathbf{x}=[1, -1, 1, -1, \dots]$:

	0.01	0.001	0.0001	0.00001	0.000001
3	1.21461e-04	6.45239e-06	3.63832e-07	3.63832e-07	2.10000e-08
6	9.67915e-05	3.33425e-06	3.33425e-06	1.15019e-07	3.97211e-09
9	2.49465e-04	4.35374e-05	7.59930e-06	1.32644e-06	4.04122e-08
12	1.44833e-04	5.06499e-06	1.77686e-07	1.77686e-07	6.25783e-09
15	7.74528e-04	1.94437e-04	1.22542e-05	7.72308e-07	1.93885e-07
18	2.17354e-04	1.19462e-05	8.61069e-07	6.84378e-08	6.84378e-08
21	1.42965e-03	1.36194e-04	1.29746e-05	1.23603e-06	1.17750e-07
24	3.61263e-04	3.51678e-05	4.00609e-06	4.64506e-07	5.39479e-08
27	2.13243e-03	9.54138e-05	1.20256e-05	1.51565e-06	1.91026e-07
30	5.61660e-04	7.53908e-05	1.09908e-05	2.36099e-07	3.46098e-08
33	1.11982e-03	1.73729e-04	2.69524e-05	1.64697e-06	2.55512e-07
36	7.94304e-04	1.30859e-04	3.93942e-06	6.84350e-07	1.18885e-07
39	1.51382e-03	2.76484e-04	2.15806e-05	1.68445e-06	1.31478e-07
42	1.04374e-03	3.94455e-05	7.81636e-06	1.54897e-06	6.08309e-08
45	1.92798e-03	1.83768e-04	1.75161e-05	1.66958e-06	1.59139e-07
48	2.78556e-04	6.13257e-05	1.35159e-05	6.56653e-07	1.44739e-07
51	2.35451e-03	2.65404e-04	2.99170e-05	1.62904e-06	1.83629e-07
54	3.66503e-04	8.81516e-05	5.11145e-06	1.23096e-06	7.13909e-08
57	2.78791e-03	1.83892e-04	2.39349e-05	3.11529e-06	2.05489e-07
60	4.61435e-04	1.19642e-04	8.06467e-06	5.43773e-07	1.41200e-07
63	3.22426e-03	2.51372e-04	1.95981e-05	2.89158e-06	2.25441e-07
66	5.61897e-04	1.55462e-04	1.19379e-05	9.17095e-07	7.04549e-08
69	2.00711e-03	3.30835e-04	2.98998e-05	2.70225e-06	2.44220e-07
72	6.66686e-04	1.95256e-04	1.68111e-05	1.44822e-06	1.24764e-07
75	2.32069e-03	2.39331e-04	2.46822e-05	2.54547e-06	2.62514e-07
78	7.74815e-04	7.36645e-05	2.27467e-05	2.16950e-06	2.06929e-07
81	2.64250e-03	3.06970e-04	3.56599e-05	2.41844e-06	2.80944e-07
84	8.85472e-04	9.21784e-05	9.63016e-06	1.00635e-06	1.05165e-07
87	2.97092e-03	2.30665e-04	2.98580e-05	2.31823e-06	3.00079e-07
90	9.97994e-04	1.12765e-04	1.27944e-05	1.45212e-06	1.64816e-07
93	3.30457e-03	2.90377e-04	2.55162e-05	3.64670e-06	3.20445e-07
96	1.11184e-03	1.35356e-04	1.65563e-05	2.02592e-06	2.47912e-07
99	3.64231e-03	3.58681e-04	3.53223e-05	3.47848e-06	3.42554e-07
102	1.22655e-03	1.59873e-04	2.09496e-05	9.94493e-07	1.30387e-07
105	2.55887e-03	2.80012e-04	3.06413e-05	3.35303e-06	3.66916e-07
108	1.34177e-03	1.86237e-04	2.60035e-05	1.35788e-06	1.89714e-07
111	2.83436e-03	3.42094e-04	2.70507e-05	3.26492e-06	3.94063e-07
114	1.45718e-03	2.14366e-04	1.22184e-05	1.81055e-06	2.68296e-07
117	3.11676e-03	2.75045e-04	3.63767e-05	3.21016e-06	2.83290e-07
120	1.57255e-03	2.44177e-04	1.51069e-05	2.36451e-06	1.46424e-07
123	3.40539e-03	3.33059e-04	3.25747e-05	3.18595e-06	3.11601e-07
126	1.68767e-03	2.75590e-04	1.84064e-05	1.23060e-06	2.02718e-07
129	3.69962e-03	3.98547e-04	2.96161e-05	3.19046e-06	3.43701e-07
132	1.80236e-03	1.28150e-04	2.21372e-05	1.59022e-06	2.74807e-07
135	3.99890e-03	3.30473e-04	3.89961e-05	3.22272e-06	3.80284e-07
138	1.91649e-03	1.45657e-04	2.63179e-05	2.02251e-06	1.55438e-07
141	3.05692e-03	3.93150e-04	3.59237e-05	3.28249e-06	2.99934e-07
144	2.02995e-03	1.64273e-04	1.34468e-05	2.53591e-06	2.07692e-07
147	3.32050e-03	3.33699e-04	3.35358e-05	3.37026e-06	3.38701e-07

Tabela 1: Błędy obliczeń (wiersze - n , kolumny - precyzja, kryterium stopu 1)

Jak nietrudno było się domyślić, błąd rozwiązania maleje wraz ze malejącą wartością ρ niezależnie od rozmiaru macierzy. Ciężko na pierwszy rzut oka zauważyć zależność n oraz błędu, jednak może być to spowodowane faktem, że program działa aż do uzyskania pewnej dokładności określonej przez kryteria stopu, więc uzyskanie tej samej dokładności dla dużych macierzy wymaga więcej iteracji lub czasu, co zostanie zbadane później.

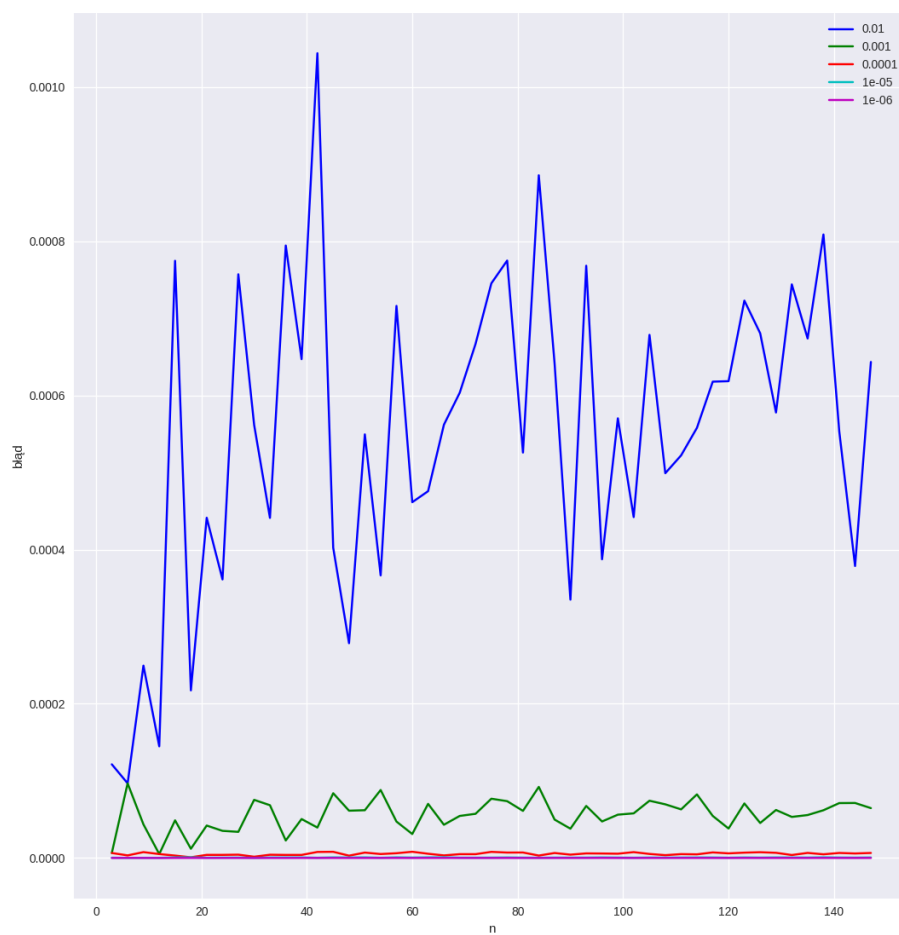


Rysunek 1: Wykres błędów na podstawie powyższej tabeli

Poniżej tabela błędów dla kryterium 2, można w niej zauważyć, że generalnie wyniki są odrobinę bardziej dokładne, co oznacza że kryterium 2 jest bardziej rygorystyczne, lecz zapewne wymaga większej liczby iteracji/czasu.

	0.01	0.001	0.0001	0.00001	0.000001
3	1.21461e-04	6.45239e-06	6.45239e-06	3.63832e-07	2.10000e-08
6	9.67915e-05	9.67915e-05	3.33425e-06	1.15019e-07	1.15019e-07
9	2.49465e-04	4.35374e-05	7.59930e-06	2.31526e-07	4.04122e-08
12	1.44833e-04	5.06499e-06	5.06499e-06	1.77686e-07	6.25783e-09
15	7.74528e-04	4.88126e-05	3.07636e-06	7.72308e-07	4.86740e-08
18	2.17354e-04	1.19462e-05	8.61069e-07	8.61069e-07	6.84378e-08
21	4.41258e-04	4.20365e-05	4.00461e-06	3.81500e-07	3.63436e-08
24	3.61263e-04	3.51678e-05	4.00609e-06	4.64506e-07	5.39479e-08
27	7.57038e-04	3.38734e-05	4.26925e-06	5.38078e-07	6.78171e-08
30	5.61660e-04	7.53908e-05	1.61063e-06	2.36099e-07	3.46098e-08
33	4.41073e-04	6.84282e-05	4.18141e-06	6.48707e-07	3.96402e-08
36	7.94304e-04	2.26793e-05	3.93942e-06	6.84350e-07	2.06526e-08
39	6.46953e-04	5.04971e-05	3.94149e-06	7.19874e-07	5.61889e-08
42	1.04374e-03	3.94455e-05	7.81636e-06	3.06961e-07	6.08309e-08
45	4.02297e-04	8.39446e-05	8.00132e-06	7.62659e-07	7.26941e-08
48	2.78556e-04	6.13257e-05	2.97912e-06	6.56653e-07	3.19033e-08
51	5.49420e-04	6.19319e-05	6.98111e-06	7.86927e-07	4.28497e-08
54	3.66503e-04	8.81516e-05	5.11145e-06	2.96444e-07	7.13909e-08
57	7.16009e-04	4.72290e-05	6.14718e-06	8.00098e-07	5.27756e-08
60	4.61435e-04	3.10593e-05	8.06467e-06	5.43773e-07	3.66652e-08
63	4.75711e-04	7.01884e-05	5.47220e-06	8.07391e-07	6.29478e-08
66	5.61897e-04	4.30739e-05	3.30878e-06	9.17095e-07	7.04549e-08
69	6.03390e-04	5.45324e-05	4.92846e-06	4.45418e-07	7.34192e-08
72	6.66686e-04	5.72821e-05	4.93413e-06	4.25072e-07	3.66201e-08
75	7.45259e-04	7.68584e-05	7.92640e-06	4.63240e-07	4.77739e-08
78	7.74815e-04	7.36645e-05	7.02477e-06	6.70024e-07	6.39078e-08
81	5.25806e-04	6.10815e-05	7.09568e-06	4.81226e-07	5.59028e-08
84	8.85472e-04	9.21784e-05	3.11306e-06	3.25318e-07	3.39963e-08
87	6.41126e-04	4.97784e-05	6.44345e-06	5.00283e-07	6.47580e-08
90	3.35039e-04	3.79800e-05	4.31031e-06	4.89216e-07	5.55261e-08
93	7.68105e-04	6.74955e-05	5.93101e-06	5.21175e-07	7.44846e-08
96	3.87378e-04	4.73334e-05	5.79147e-06	7.08696e-07	8.67234e-08
99	5.70218e-04	5.61541e-05	5.52995e-06	5.44580e-07	5.36292e-08
102	4.42118e-04	5.78648e-05	7.58536e-06	3.60095e-07	4.72117e-08
105	6.78461e-04	7.42429e-05	5.21928e-06	5.71138e-07	6.24987e-08
108	4.99013e-04	6.95788e-05	3.63282e-06	5.07552e-07	7.09120e-08
111	5.22164e-04	6.30233e-05	4.98349e-06	6.01488e-07	7.25973e-08
114	5.57838e-04	8.24736e-05	4.70338e-06	6.96966e-07	3.97575e-08
117	6.17783e-04	5.45179e-05	7.21040e-06	6.36301e-07	5.61521e-08
120	6.18388e-04	3.81869e-05	5.97660e-06	3.70102e-07	5.79298e-08
123	7.22884e-04	7.07013e-05	6.91491e-06	6.76310e-07	6.61461e-08
126	6.80479e-04	4.53541e-05	7.47041e-06	4.99464e-07	8.22776e-08
129	5.77774e-04	6.22422e-05	6.70520e-06	7.22335e-07	5.36768e-08
132	7.43944e-04	5.32575e-05	3.82537e-06	6.61062e-07	4.74895e-08
135	6.73762e-04	5.56810e-05	6.57042e-06	5.42992e-07	6.40737e-08
138	8.08630e-04	6.19079e-05	4.75706e-06	8.59896e-07	6.60863e-08
141	5.53366e-04	7.11686e-05	6.50296e-06	5.94200e-07	5.42944e-08
144	3.78654e-04	7.13134e-05	5.83947e-06	4.78252e-07	3.91692e-08
147	6.43360e-04	6.46559e-05	6.49774e-06	6.53004e-07	6.56251e-08

Tabela 2: Błędy obliczeń (wiersze - n , kolumny - precyzja, kryterium stopu 2)



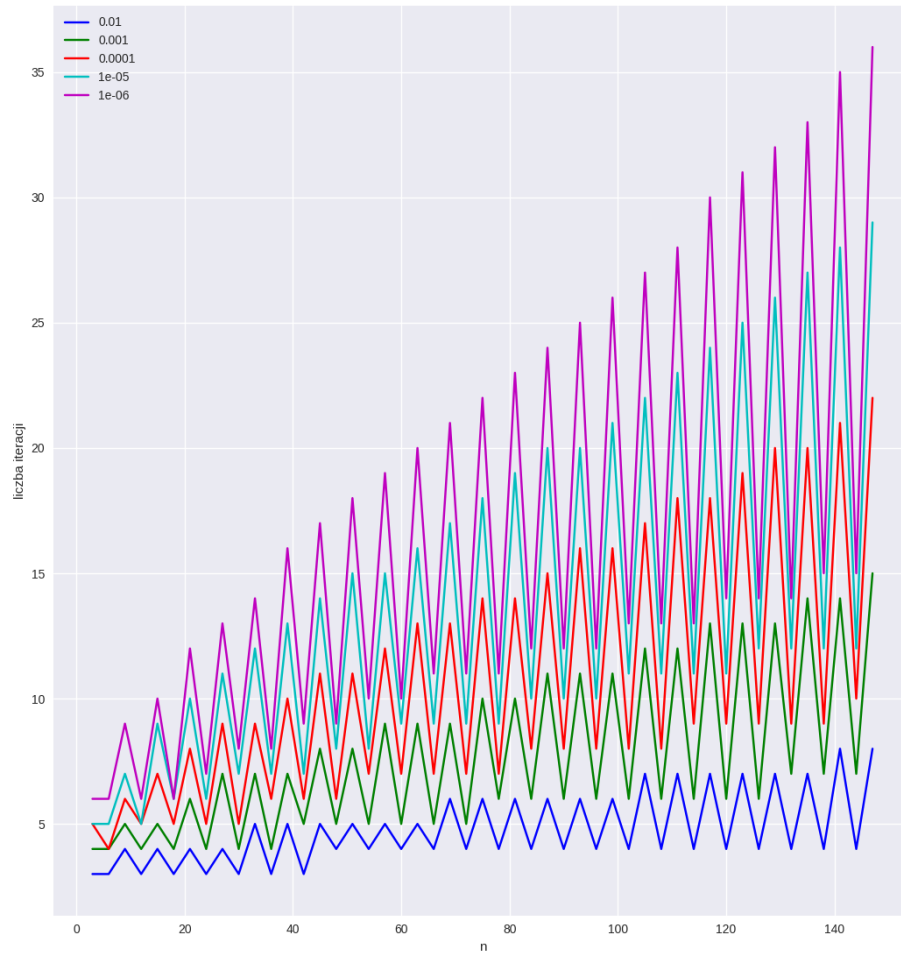
Rysunek 2: Wykres błędów na podstawie powyższej tabeli

Teraz zbadane zostaną liczby iteracji/czas działania dla poszczególnych precyzji oraz wartości n .

	0.01	0.001	0.0001	0.00001	0.000001
3	3	4	5	5	6
6	3	4	4	5	6
9	4	5	6	7	9
12	3	4	5	5	6
15	4	5	7	9	10
18	3	4	5	6	6
21	4	6	8	10	12
24	3	4	5	6	7
27	4	7	9	11	13
30	3	4	5	7	8
33	5	7	9	12	14
36	3	4	6	7	8
39	5	7	10	13	16
42	3	5	6	7	9
45	5	8	11	14	17
48	4	5	6	8	9
51	5	8	11	15	18
54	4	5	7	8	10
57	5	9	12	15	19
60	4	5	7	9	10
63	5	9	13	16	20
66	4	5	7	9	11
69	6	9	13	17	21
72	4	5	7	9	11
75	6	10	14	18	22
78	4	6	7	9	11
81	6	10	14	19	23
84	4	6	8	10	12
87	6	11	15	20	24
90	4	6	8	10	12
93	6	11	16	20	25
96	4	6	8	10	12
99	6	11	16	21	26
102	4	6	8	11	13
105	7	12	17	22	27
108	4	6	8	11	13
111	7	12	18	23	28
114	4	6	9	11	13
117	7	13	18	24	30
120	4	6	9	11	14
123	7	13	19	25	31
126	4	6	9	12	14
129	7	13	20	26	32
132	4	7	9	12	14
135	7	14	20	27	33
138	4	7	9 6	12	15
141	8	14	21	28	35
144	4	7	10	12	15
147	8	15	22	29	36

Tabela 3: Iteracje (wiersze - n , kolumny - precyzja, kryterium stopu 1)

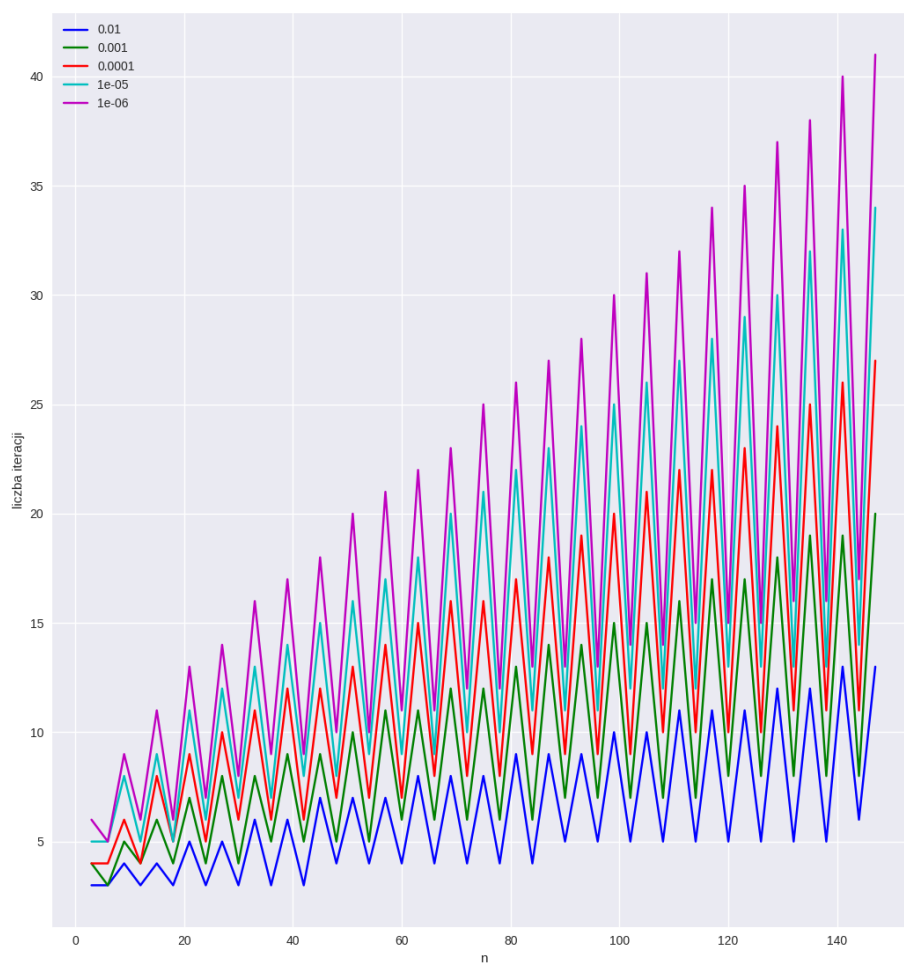
Zgodnie z poprzednimi przypuszczeniami, liczba iteracji rośnie wraz ze wzrastającą precyzją. Co jednak bardzo zastawiające, przyrost jest niejako niezależny dla nieparzystych oraz parzystych wartości n , tzn. gdyby odseparować na osobnych wykresach liczby iteracji dla wartości parzystych i nieparzystych n , oba te wykresy byłyby niemalejące. Podobne zjawisko występuje dla drugiego kryterium stopu.



Rysunek 3: Liczba iteracji na podstawie powyższej tabeli

	0.01	0.001	0.0001	0.00001	0.000001
3	3	4	4	5	6
6	3	3	4	5	5
9	4	5	6	8	9
12	3	4	4	5	6
15	4	6	8	9	11
18	3	4	5	5	6
21	5	7	9	11	13
24	3	4	5	6	7
27	5	8	10	12	14
30	3	4	6	7	8
33	6	8	11	13	16
36	3	5	6	7	9
39	6	9	12	14	17
42	3	5	6	8	9
45	7	9	12	15	18
48	4	5	7	8	10
51	7	10	13	16	20
54	4	5	7	9	10
57	7	11	14	17	21
60	4	6	7	9	11
63	8	11	15	18	22
66	4	6	8	9	11
69	8	12	16	20	23
72	4	6	8	10	12
75	8	12	16	21	25
78	4	6	8	10	12
81	9	13	17	22	26
84	4	6	9	11	13
87	9	14	18	23	27
90	5	7	9	11	13
93	9	14	19	24	28
96	5	7	9	11	13
99	10	15	20	25	30
102	5	7	9	12	14
105	10	15	21	26	31
108	5	7	10	12	14
111	11	16	22	27	32
114	5	7	10	12	15
117	11	17	22	28	34
120	5	8	10	13	15
123	11	17	23	29	35
126	5	8	10	13	15
129	12	18	24	30	37
132	5	8	11	13	16
135	12	19	25	32	38
138	5	8	11 8	13	16
141	13	19	26	33	40
144	6	8	11	14	17
147	13	20	27	34	41

Tabela 4: Iteracje (wiersze - n , kolumny - precyzja, kryterium stopu 2)

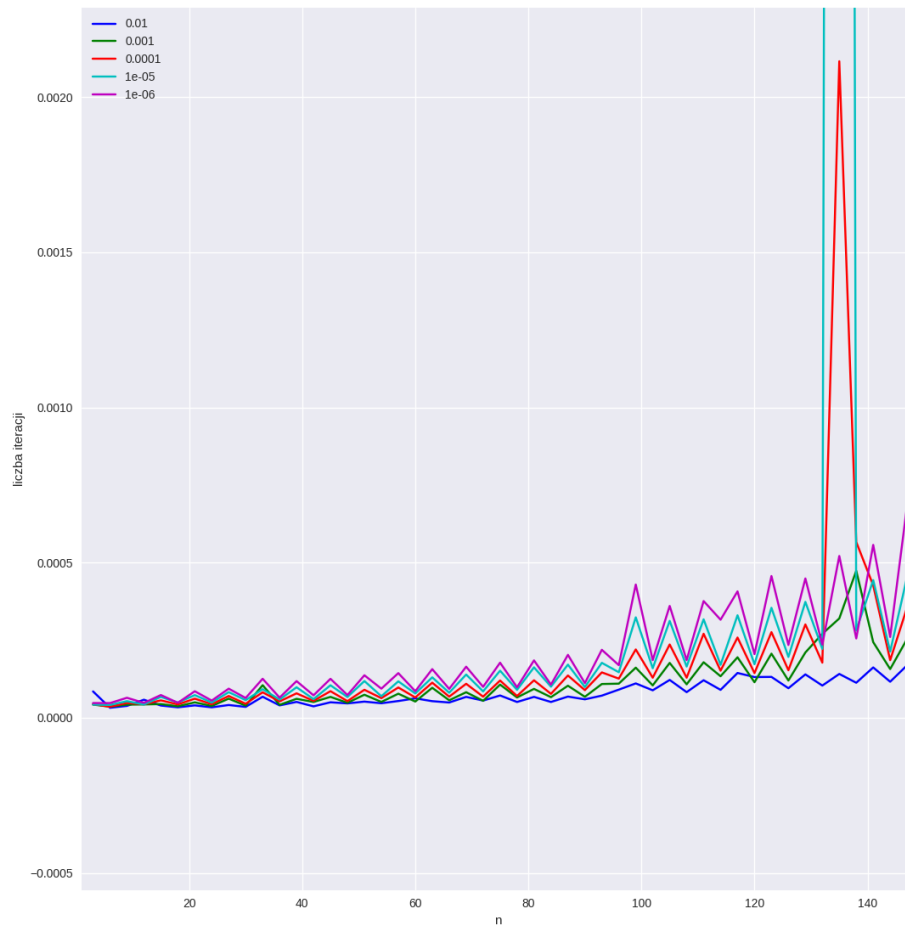


Rysunek 4: Liczba iteracji na podstawie powyższej tabeli

W ogólności dla drugiego, bardziej rygorystycznego kryterium stopu potrzeba więcej iteracji. Poniżej tabele czasu w sekundach:

	0.01	0.001	0.0001	0.00001	0.000001
3	0.0000875	0.0000467	0.0000496	0.0000479	0.0000532
6	0.0000553	0.0000694	0.0000434	0.0000455	0.0000911
9	0.0000851	0.0000927	0.0000794	0.0000591	0.0000679
12	0.0000625	0.0000656	0.0000825	0.0000935	0.0000961
15	0.0000772	0.0000820	0.0000939	0.0001132	0.0001304
18	0.0000753	0.0000722	0.0000730	0.0000877	0.0000925
21	0.0000985	0.0000927	0.0001428	0.0001245	0.0001514
24	0.0000935	0.0001035	0.0001106	0.0001278	0.0001600
27	0.0001264	0.0001535	0.0001547	0.0001626	0.0001926
30	0.0002003	0.0001001	0.0001323	0.0001588	0.0000765
33	0.0001099	0.0000854	0.0000761	0.0001080	0.0001056
36	0.0000687	0.0000479	0.0000546	0.0000613	0.0000694
39	0.0001190	0.0000687	0.0000811	0.0001004	0.0001278
42	0.0000904	0.0000553	0.0000744	0.0000978	0.0000811
45	0.0001194	0.0000758	0.0000896	0.0001173	0.0001295
48	0.0000827	0.0000536	0.0000563	0.0001314	0.0000889
51	0.0000858	0.0001590	0.0003655	0.0002425	0.0002537
54	0.0001256	0.0001256	0.0001025	0.0001073	0.0001309
57	0.0001035	0.0001264	0.0001523	0.0001895	0.0003147
60	0.0000896	0.0000651	0.0000777	0.0000839	0.0001783
63	0.0001538	0.0001783	0.0002410	0.0001719	0.0001745
66	0.0000956	0.0000625	0.0000732	0.0001764	0.0001788
69	0.0000973	0.0000944	0.0001142	0.0001411	0.0001702
72	0.0000806	0.0000761	0.0003283	0.0001762	0.0001783
75	0.0001063	0.0001340	0.0001547	0.0001683	0.0001993
78	0.0000889	0.0000730	0.0000746	0.0000911	0.0001059
81	0.0000992	0.0002117	0.0002501	0.0003147	0.0002294
84	0.0000858	0.0000818	0.0000827	0.0001128	0.0001242
87	0.0001016	0.0001194	0.0001428	0.0003097	0.0002751
90	0.0000956	0.0000775	0.0000868	0.0001082	0.0001197
93	0.0001061	0.0001571	0.0001915	0.0002210	0.0002954
96	0.0001454	0.0001440	0.0001724	0.0001581	0.0014191
99	0.0001471	0.0001960	0.0002384	0.0027018	0.0005896
102	0.0001287	0.0027108	0.0001626	0.0002019	0.0002334
105	0.0035703	0.0002294	0.0002589	0.0024052	0.0004098
108	0.0001583	0.0001814	0.0001416	0.0001955	0.0002174
111	0.0024757	0.0002346	0.0003364	0.0003989	0.0006509
114	0.0001316	0.0011964	0.0001891	0.0001981	0.0002663
117	0.0002456	0.0002358	0.0406570	0.0003750	0.0004618
120	0.0002244	0.0001340	0.0001614	0.0061872	0.0002432
123	0.0001779	0.0021572	0.0003467	0.0003650	0.0009282
126	0.0001872	0.0001454	0.0007567	0.0002885	0.0003605
129	0.0013764	0.0002382	0.0041809	0.0012856	0.0005593
132	0.0016842	0.0001853	0.0057194	0.0002387	0.0004048
135	0.0001967	0.0003278	0.0004053	0.0004618	0.0005331
138	0.0001743	0.0001707	0.0002153	0.0002213	0.0002990
141	0.0002463	0.0042546	0.0003917	0.0004747	0.0005934
144	0.0001466	0.0002179	0.0002005	0.0002601	0.0002675
147	0.0002334	0.0003028	0.0012977	0.0005009	0.0006073

Tabela 5: Czas (wiersze - n , kolumny - precyzja, kryterium stopu 1)



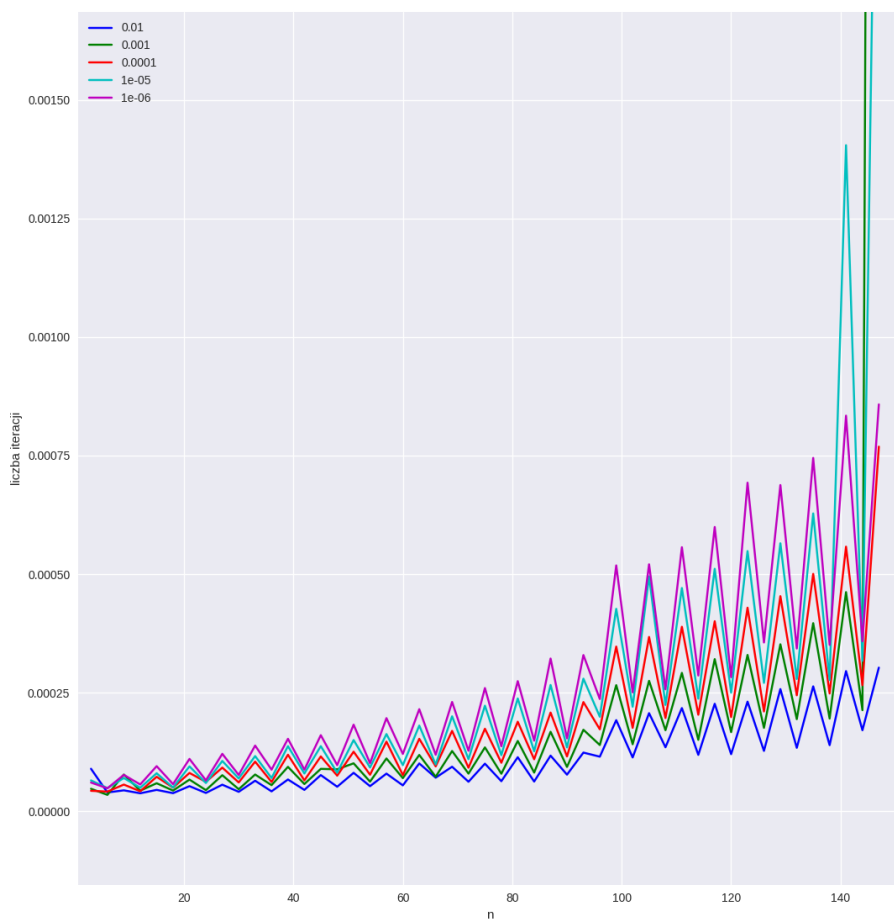
Rysunek 5: Czas działania na podstawie powyższej tabeli

Czas działania jest bardzo niski (z pewnymi odstępstwami dla większych wartości n), co jest zasługą użycia operacji macierzowych dostępnych w pakiecie numpy. W ogólności prezentuje on niewielką tendencję rosnącą z pewnymi nieregularnymi skokami dla dużych wartości n , gdzie prezentują podobne zjawisko jak w przypadku liczby iteracji. Efekt ten był jednak mocno uzależniony od próby.

	0.01	0.001	0.0001	0.00001	0.000001
3	0.0000885	0.0000448	0.0000405	0.0000460	0.0000520
6	0.0000520	0.0000582	0.0000734	0.0000856	0.0000541
9	0.0000689	0.0000970	0.0000622	0.0000677	0.0000753
12	0.0000796	0.0000823	0.0000625	0.0000496	0.0000653
15	0.0001304	0.0000694	0.0000732	0.0000966	0.0000982
18	0.0000706	0.0000985	0.0000601	0.0000527	0.0000651
21	0.0000758	0.0000951	0.0001533	0.0001709	0.0001230
24	0.0000756	0.0001197	0.0001507	0.0001342	0.0000939
27	0.0001285	0.0001628	0.0001502	0.0001769	0.0002279
30	0.0001366	0.0001440	0.0001776	0.0001857	0.0000925
33	0.0001054	0.0000894	0.0000973	0.0001476	0.0001600
36	0.0000994	0.0000601	0.0000746	0.0000670	0.0001132
39	0.0001359	0.0001042	0.0001075	0.0001552	0.0001526
42	0.0001040	0.0000615	0.0000639	0.0001261	0.0000863
45	0.0001435	0.0000920	0.0001085	0.0001378	0.0001724
48	0.0001230	0.0000808	0.0001044	0.0001488	0.0001237
51	0.0001485	0.0001559	0.0003209	0.0002754	0.0003614
54	0.0001411	0.0000703	0.0000792	0.0001056	0.0001009
57	0.0001059	0.0001199	0.0001338	0.0001545	0.0001893
60	0.0000803	0.0000856	0.0000775	0.0000899	0.0001087
63	0.0001833	0.0001509	0.0002174	0.0002193	0.0002525
66	0.0001061	0.0000827	0.0001154	0.0001566	0.0001645
69	0.0001287	0.0001419	0.0001631	0.0002213	0.0004468
72	0.0000954	0.0000937	0.0000906	0.0001061	0.0001936
75	0.0002260	0.0002913	0.0002909	0.0002234	0.0002582
78	0.0001352	0.0001044	0.0001180	0.0001161	0.0001454
81	0.0001812	0.0001636	0.0002263	0.0002744	0.0005000
84	0.0001080	0.0001073	0.0001066	0.0001419	0.0001602
87	0.0001502	0.0001736	0.0004206	0.0002897	0.0004199
90	0.0001278	0.0002105	0.0001416	0.0001376	0.0001526
93	0.0001891	0.0002127	0.0002286	0.0003140	0.0005865
96	0.0001814	0.0003712	0.0001850	0.0001819	0.0002208
99	0.0002198	0.0002406	0.0004840	0.0006256	0.0004938
102	0.0001602	0.0001724	0.0001936	0.0009170	0.0002658
105	0.0002341	0.0050430	0.0004256	0.0004683	0.0007200
108	0.0024714	0.0001760	0.0002422	0.0051222	0.0004179
111	0.0002668	0.0003059	0.0029092	0.0005157	0.0005651
114	0.0032330	0.0001719	0.0002096	0.0025511	0.0003009
117	0.0002780	0.0057919	0.0004408	0.0005164	0.0030587
120	0.0001614	0.0001938	0.0024755	0.0002685	0.0002832
123	0.0027502	0.0003624	0.0004699	0.0014372	0.0006979
126	0.0001879	0.0026448	0.0002863	0.0002663	0.0017917
129	0.0003397	0.0004182	0.0092058	0.0016429	0.0071719
132	0.0062399	0.0548844	0.0002735	0.0003014	0.0003245
135	0.0003462	0.0004387	0.0005884	0.0014827	0.0007622
138	0.0002155	0.0002034	0.0002992	0.0012877	0.0004177
141	0.0007682	0.0043020	0.0005875	0.0006642	0.0008454
144	0.0002103	0.0002539	0.0006621	0.0010614	0.0003817
147	0.0003996	0.0004659	0.0006325	0.0007353	0.0008531

Tabela 6: Czas (wiersze - n , kolumny - precyzja, kryterium stopu 2)

Dla kryterium drugiego czasy działania są bardzo podobne, z bardzo nieznaczną przewagą czasową dla kryterium pierwszego.



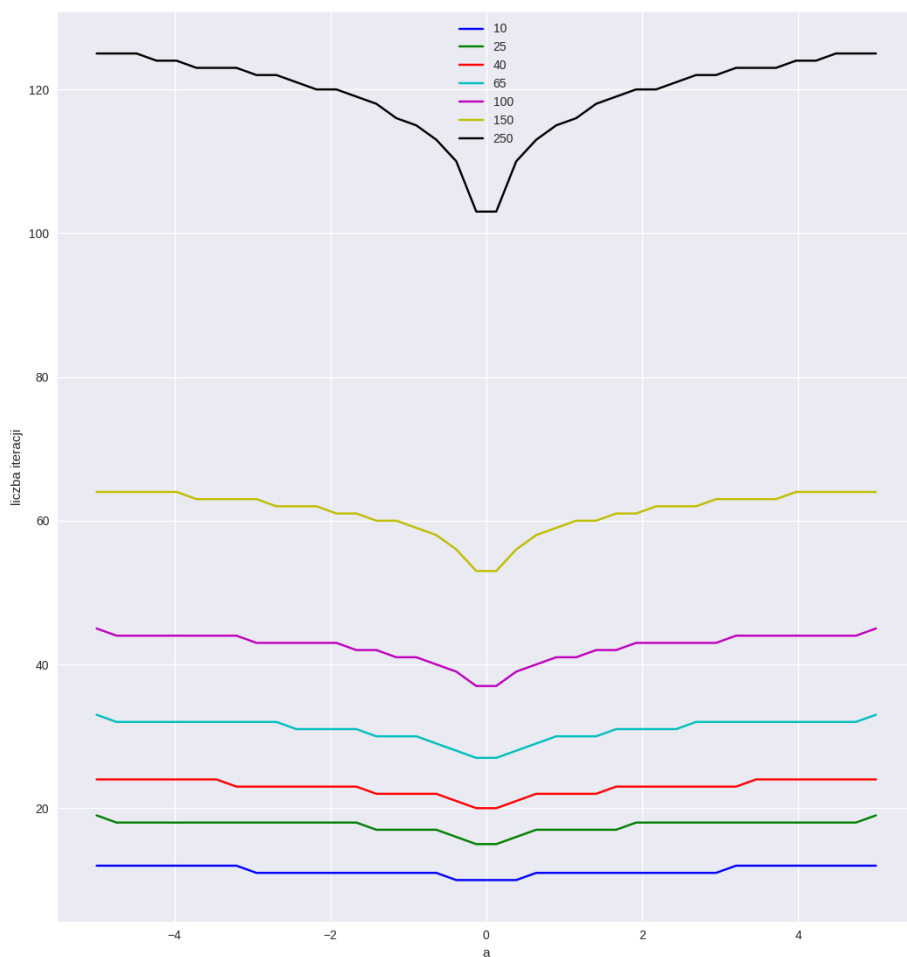
Rysunek 6: Czas działania na podstawie powyższej tabeli

Teraz pozostaje zbadanie wpływu wektora początkowego. Tutaj zostanie użyte tylko kryterium pierwsze oraz precyzja $\rho = 10^{-6}$. Użyte wektory początkowe przyjmą postać $\mathbf{x} = [a, a, \dots, a]$, gdzie a będzie wartościami z zakresu $[-5, 5]$.

	10	25	40	65	100	150	250
-5.000	3.31058e-08	3.22290e-08	5.75751e-08	4.39648e-08	5.06492e-08	6.91441e-08	6.49123e-08
-4.744	3.14080e-08	8.98168e-08	5.46226e-08	7.79389e-08	7.61143e-08	6.55983e-08	6.15835e-08
-4.487	2.97103e-08	8.49908e-08	5.16700e-08	7.37345e-08	7.20000e-08	6.20524e-08	5.82547e-08
-4.231	2.80126e-08	8.01647e-08	4.87174e-08	6.95301e-08	6.78858e-08	5.85066e-08	6.49249e-08
-3.974	2.63148e-08	7.53386e-08	4.57648e-08	6.53257e-08	6.37715e-08	5.49607e-08	6.09901e-08
-3.718	2.46171e-08	7.05126e-08	4.28123e-08	6.11213e-08	5.96572e-08	7.09250e-08	6.74420e-08
-3.462	2.29194e-08	6.56865e-08	3.98597e-08	5.69169e-08	5.55429e-08	6.60336e-08	6.27908e-08
-3.205	2.12216e-08	6.08604e-08	8.53299e-08	5.27126e-08	5.14286e-08	6.11422e-08	5.81396e-08
-2.949	1.03202e-07	5.60344e-08	7.85035e-08	4.85082e-08	7.49462e-08	5.62508e-08	6.32259e-08
-2.692	9.42277e-08	5.12083e-08	7.16771e-08	4.43038e-08	6.84291e-08	7.08485e-08	5.77280e-08
-2.436	8.52536e-08	4.63822e-08	6.48507e-08	7.49213e-08	6.19121e-08	6.41010e-08	6.17384e-08
-2.179	7.62795e-08	4.15562e-08	5.80243e-08	6.70658e-08	5.53950e-08	5.73536e-08	6.52958e-08
-1.923	6.73055e-08	3.67301e-08	5.11980e-08	5.92104e-08	4.88780e-08	6.98093e-08	5.76140e-08
-1.667	5.83314e-08	3.19040e-08	4.43716e-08	5.13550e-08	6.71000e-08	6.05014e-08	5.90221e-08
-1.410	4.93573e-08	7.95165e-08	8.68051e-08	8.12741e-08	5.67769e-08	7.06195e-08	5.90335e-08
-1.154	4.03833e-08	6.53444e-08	7.10223e-08	6.65971e-08	7.35832e-08	5.77796e-08	6.74867e-08
-0.897	3.14092e-08	5.11723e-08	5.52396e-08	5.19201e-08	5.72313e-08	6.19927e-08	6.20452e-08
-0.641	2.24352e-08	3.70002e-08	3.94568e-08	6.95847e-08	6.47535e-08	6.10834e-08	6.19227e-08
-0.385	7.11542e-08	6.70363e-08	5.47349e-08	7.87757e-08	6.15420e-08	6.97421e-08	6.13628e-08
-0.128	2.37181e-08	7.46445e-08	4.21827e-08	5.14558e-08	5.14712e-08	6.10246e-08	6.59519e-08
0.128	2.37181e-08	4.75681e-08	4.21827e-08	4.42720e-08	5.14712e-08	6.10246e-08	6.59519e-08
0.385	7.11542e-08	5.78159e-08	5.47349e-08	7.49308e-08	6.15420e-08	6.97421e-08	6.13628e-08
0.641	2.24352e-08	3.38603e-08	3.94568e-08	6.75268e-08	6.47535e-08	6.10834e-08	6.19227e-08
0.897	3.14092e-08	4.80324e-08	5.52396e-08	5.08187e-08	5.72313e-08	6.19927e-08	6.20452e-08
1.154	4.03833e-08	6.22045e-08	7.10223e-08	6.54957e-08	7.35832e-08	5.77796e-08	6.74867e-08
1.410	4.93573e-08	7.63766e-08	8.68051e-08	8.01727e-08	5.67769e-08	7.06195e-08	5.90335e-08
1.667	5.83314e-08	9.05487e-08	4.43716e-08	5.07655e-08	6.71000e-08	6.05014e-08	5.90221e-08
1.923	6.73055e-08	3.56609e-08	5.11980e-08	5.86209e-08	4.88780e-08	6.98093e-08	5.76140e-08
2.179	7.62795e-08	4.04869e-08	5.80243e-08	6.64763e-08	5.53950e-08	5.73536e-08	6.52958e-08
2.436	8.52536e-08	4.53130e-08	6.48507e-08	7.43318e-08	6.19121e-08	6.41010e-08	6.17384e-08
2.692	9.42277e-08	5.01391e-08	7.16771e-08	4.39883e-08	6.84291e-08	7.08485e-08	5.77280e-08
2.949	1.03202e-07	5.49651e-08	7.85035e-08	4.81927e-08	7.49462e-08	5.62508e-08	6.32259e-08
3.205	2.12216e-08	5.97912e-08	8.53299e-08	5.23970e-08	5.14286e-08	6.11422e-08	5.81396e-08
3.462	2.29194e-08	6.46173e-08	3.98597e-08	5.66014e-08	5.55429e-08	6.60336e-08	6.27908e-08
3.718	2.46171e-08	6.94433e-08	4.28123e-08	6.08058e-08	5.96572e-08	7.09250e-08	6.74420e-08
3.974	2.63148e-08	7.42694e-08	4.57648e-08	6.50102e-08	6.37715e-08	5.49607e-08	6.09901e-08
4.231	2.80126e-08	7.90955e-08	4.87174e-08	6.92146e-08	6.78858e-08	5.85066e-08	6.49249e-08
4.487	2.97103e-08	8.39215e-08	5.16700e-08	7.34190e-08	7.20000e-08	6.20524e-08	5.82547e-08
4.744	3.14080e-08	8.87476e-08	5.46226e-08	7.76234e-08	7.61143e-08	6.55983e-08	6.15835e-08
5.000	3.31058e-08	3.18649e-08	5.75751e-08	4.37959e-08	5.06492e-08	6.91441e-08	6.49123e-08

Tabela 7: Błędy (wiersze - a , kolumny - n , kryterium stopu 1)

Nie widać dużego wpływu wektora początkowego na ostateczną precyzję, może jednak jego wybór odzwierciedla się w liczbie iteracji i czasie działania:



Rysunek 7: Liczba iteracji na poniższej tabeli

I rzeczywiście na podstawie wykresu można zauważyć, że im bliżej zera, tym mniej potrzeba iteracji do osiągnięcia akceptowalnego wyniku, więc odległość wektora początkowego ma wpływ na liczbę iteracji. Do tego można przypuszczać, że im większa odległość wektora początkowego (w sensie normy euklidesowej), tym większa liczba iteracji.

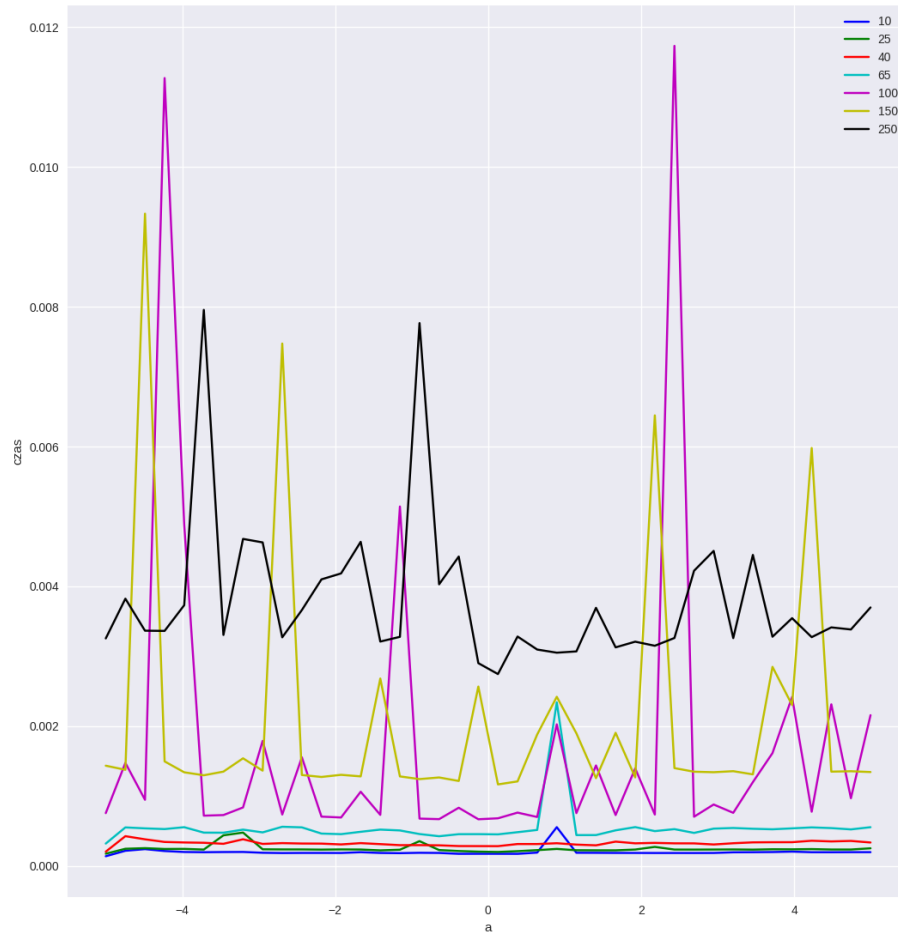
	10	25	40	65	100	150	250
-5.000	12	19	24	33	45	64	125
-4.744	12	18	24	32	44	64	125
-4.487	12	18	24	32	44	64	125
-4.231	12	18	24	32	44	64	124
-3.974	12	18	24	32	44	64	124
-3.718	12	18	24	32	44	63	123
-3.462	12	18	24	32	44	63	123
-3.205	12	18	23	32	44	63	123
-2.949	11	18	23	32	43	63	122
-2.692	11	18	23	32	43	62	122
-2.436	11	18	23	31	43	62	121
-2.179	11	18	23	31	43	62	120
-1.923	11	18	23	31	43	61	120
-1.667	11	18	23	31	42	61	119
-1.410	11	17	22	30	42	60	118
-1.154	11	17	22	30	41	60	116
-0.897	11	17	22	30	41	59	115
-0.641	11	17	22	29	40	58	113
-0.385	10	16	21	28	39	56	110
-0.128	10	15	20	27	37	53	103
0.128	10	15	20	27	37	53	103
0.385	10	16	21	28	39	56	110
0.641	11	17	22	29	40	58	113
0.897	11	17	22	30	41	59	115
1.154	11	17	22	30	41	60	116
1.410	11	17	22	30	42	60	118
1.667	11	17	23	31	42	61	119
1.923	11	18	23	31	43	61	120
2.179	11	18	23	31	43	62	120
2.436	11	18	23	31	43	62	121
2.692	11	18	23	32	43	62	122
2.949	11	18	23	32	43	63	122
3.205	12	18	23	32	44	63	123
3.462	12	18	24	32	44	63	123
3.718	12	18	24	32	44	63	123
3.974	12	18	24	32	44	64	124
4.231	12	18	24	32	44	64	124
4.487	12	18	24	32	44	64	125
4.744	12	18	24	32	44	64	125
5.000	12	19	24	33	45	64	125

Tabela 8: Liczba iteracji(wiersze - a , kolumny - n , kryterium stopu 1)

	10	25	40	65	100	150	250
-5.000	0.0001397	0.0001760	0.0002053	0.0003209	0.0007567	0.0014353	0.0032563
-4.744	0.0002179	0.0002465	0.0004268	0.0005522	0.0014753	0.0013766	0.0038245
-4.487	0.0002415	0.0002553	0.0003819	0.0005386	0.0009477	0.0093334	0.0033667
-4.231	0.0002143	0.0002434	0.0003440	0.0005283	0.0112736	0.0014968	0.0033636
-3.974	0.0002005	0.0002449	0.0003366	0.0005553	0.0048840	0.0013406	0.0037315
-3.718	0.0001972	0.0002353	0.0003324	0.0004776	0.0007203	0.0012982	0.0079567
-3.462	0.0001998	0.0004389	0.0003178	0.0004764	0.0007288	0.0013504	0.0033059
-3.205	0.0002010	0.0004783	0.0003824	0.0005205	0.0008378	0.0015407	0.0046799
-2.949	0.0001893	0.0002389	0.0003161	0.0004809	0.0017903	0.0013642	0.0046299
-2.692	0.0001855	0.0002372	0.0003273	0.0005615	0.0007365	0.0074754	0.0032718
-2.436	0.0001872	0.0002370	0.0003216	0.0005529	0.0015552	0.0013020	0.0036619
-2.179	0.0001872	0.0002344	0.0003204	0.0004640	0.0007064	0.0012741	0.0041010
-1.923	0.0001864	0.0002372	0.0003085	0.0004556	0.0006943	0.0013046	0.0041862
-1.667	0.0001974	0.0002358	0.0003273	0.0004895	0.0010626	0.0012822	0.0046384
-1.410	0.0001867	0.0002236	0.0003130	0.0005209	0.0007317	0.0026822	0.0032113
-1.154	0.0001838	0.0002329	0.0002983	0.0005085	0.0051436	0.0012820	0.0032787
-0.897	0.0001891	0.0003557	0.0002985	0.0004573	0.0006781	0.0012434	0.0077679
-0.641	0.0001869	0.0002282	0.0002959	0.0004258	0.0006704	0.0012670	0.0040302
-0.385	0.0001743	0.0002143	0.0002854	0.0004554	0.0008335	0.0012164	0.0044274
-0.128	0.0001750	0.0002055	0.0002847	0.0004559	0.0006688	0.0025673	0.0029023
0.128	0.0001748	0.0002010	0.0002840	0.0004528	0.0006824	0.0011675	0.0027463
0.385	0.0001740	0.0002124	0.0003145	0.0004838	0.0007639	0.0012105	0.0032833
0.641	0.0001917	0.0002270	0.0003142	0.0005157	0.0007012	0.0018809	0.0030971
0.897	0.0005555	0.0002444	0.0003259	0.0023410	0.0020270	0.0024209	0.0030520
1.154	0.0001893	0.0002255	0.0003061	0.0004423	0.0007567	0.0018985	0.0030704
1.410	0.0001886	0.0002244	0.0002959	0.0004425	0.0014386	0.0012562	0.0036941
1.667	0.0001872	0.0002241	0.0003488	0.0005102	0.0007288	0.0019057	0.0031283
1.923	0.0001869	0.0002356	0.0003245	0.0005565	0.0013964	0.0012677	0.0032096
2.179	0.0001860	0.0002735	0.0003307	0.0004992	0.0007367	0.0064473	0.0031512
2.436	0.0001862	0.0002353	0.0003242	0.0005274	0.0117342	0.0014005	0.0032623
2.692	0.0001860	0.0002348	0.0003238	0.0004735	0.0007050	0.0013492	0.0042241
2.949	0.0001867	0.0002356	0.0003078	0.0005348	0.0008800	0.0013418	0.0045085
3.205	0.0001960	0.0002370	0.0003257	0.0005441	0.0007608	0.0013568	0.0032611
3.462	0.0001974	0.0002344	0.0003381	0.0005329	0.0012000	0.0013111	0.0044508
3.718	0.0001998	0.0002396	0.0003402	0.0005252	0.0016153	0.0028496	0.0032804
3.974	0.0002058	0.0002387	0.0003407	0.0005386	0.0024178	0.0023031	0.0035458
4.231	0.0001972	0.0002418	0.0003614	0.0005524	0.0007770	0.0059810	0.0032744
4.487	0.0001969	0.0002360	0.0003517	0.0005424	0.0023134	0.0013502	0.0034139
4.744	0.0001981	0.0002358	0.0003588	0.0005238	0.0009696	0.0013559	0.0033841
5.000	0.0001981	0.0002532	0.0003362	0.0005553	0.0021579	0.0013444	0.0036995

Tabela 9: Czas (wiersze - a , kolumny - n , kryterium stopu 1)

Chciałoby się przypuszczać, że w przypadku czasu działania wystąpi podobne zjawisko jak dla liczby iteracji — mniejsza odległość od rzeczywistego wyniku skutkuje lepszym czasem, jednak wyniki ciężko zinterpretować w taki sposób. Do tego wyniki były bardzo niekonsekwentne w zależności od wywołania programu.



Rysunek 8: Czas działania na podstawie powyższej tabeli

2 Problem 2

2.1 Opis problemu

Przy użyciu dowolnej metody zostanie znaleziony promień spektralny macierzy iteracji z poprzedniego problemu (dla różnych rozmiarów układu — takich, dla których znajdowane były rozwiązania układu). Sprawdzane zostanie, czy spełnione są założenia o zbieżności metody dla zadanego układu. Opisane zostanie metoda znajdowania promienia spektralnego.

2.2 Opracowanie problemu

Promień spektralny jest wartością maksymalną spośród wartości bezwzględnych wartości własnych macierzy, tj.:

$$\rho(A) = \max\{|\lambda_1|, \dots, |\lambda_n|\}$$

Wartościami własnymi macierzy nazywamy natomiast pierwiastki wielomianu charakterystycznego tej macierzy:

$$w_A(\lambda) = \det(A - \lambda I)$$

gdzie I jest macierzą jednostkową. Do policzenia wartości własnych wielomianu użyta zostanie funkcja `numpy.linalg.eigvals`.

Niech $\epsilon = x^{(t)} - x$, gdzie x jest wektorem rzeczywistych rozwiązań układu równań. Wówczas macierz M taką, że

$$\epsilon^{(t)} = M^t \cdot \epsilon^{(0)}$$

nazywamy macierzą iteracji. Macierz iteracji dla metody Jakobiego ma postać

$$M = D^{-1}(L + U)$$

Gdzie (niech A będzie macierzą układu równań)

$$A = D + L + U$$

D jest macierzą diagonalną, z diagonalnych elementów macierzy A , L — poddiagonalną, U — nadddiagonalną. Mając te informacje, można łatwo obliczyć promienie spektralne dla macierzy iteracji.

n	promień	n	promień
3	0.05843	52	0.48718
4	0.08253	53	0.49123
5	0.10423	54	0.49521
6	0.12398	55	0.49913
7	0.14212	56	0.50299
8	0.15891	57	0.50678
9	0.17455	58	0.51052
10	0.18918	59	0.51421
11	0.20294	60	0.51784
12	0.21593	61	0.52141
13	0.22823	62	0.52494
14	0.23992	63	0.52841
15	0.25105	64	0.53184
16	0.26167	65	0.53522
17	0.27184	66	0.53855
18	0.28159	67	0.54184
19	0.29096	68	0.54509
20	0.29997	69	0.54829
21	0.30865	70	0.55146
22	0.31703	71	0.55458
23	0.32512	72	0.55766
24	0.33295	73	0.56071
25	0.34053	74	0.56372
26	0.34788	75	0.56669
27	0.35502	76	0.56963
28	0.36194	77	0.57253
29	0.36868	78	0.57540
30	0.37523	79	0.57823
31	0.38161	80	0.58104
32	0.38782	81	0.58381
33	0.39388	82	0.58655
34	0.39979	83	0.58926
35	0.40556	84	0.59194
36	0.41120	85	0.59460
37	0.41671	86	0.59722
38	0.42209	87	0.59982
39	0.42736	88	0.60239
40	0.43252	89	0.60493
41	0.43757	90	0.60745
42	0.44252	91	0.60994
43	0.44737	92	0.61241
44	0.45213	93	0.61485
45	0.45680	94	0.61727
46	0.46137	95	0.61967
47	0.46587	96	0.62204
48	0.47028	97	0.62439
49	0.47462	98	0.62672
50	0.47888	99	0.62903
51	0.48306	100	0.63131

Tabela 10: Promienie spektralne

Warunkiem wystarczającym zbieżności metody iteracyjnej jest

$$\rho(M) < 1$$

Jak nietrudno zauważyć, dla wszystkich badanych wartości n promień spektralny macierzy iteracji jest mniejszy od jedynki, dla nich metoda jest zbieżna.

2.3 Wnioski

Metoda Jakobiego jest jedną z najprostszych metod iteracyjnych pozwalających na poprawienie znanego, ale niedokładnego rozwiązania układu równań liniowych. Na jakość uzyskanego rozwiązania wpływa głównie wybrana precyzja, natomiast liczba iteracji i czas działania zależą od wybranej precyzji, rozmiaru macierzy oraz tego, jak blisko wektor początkowy znajduje się rzeczywistego rozwiązania. Spośród dwóch wykorzystanych tutaj kryteriów, drugie okazało się być bardziej rygorystyczne i skutkowało dokładniejszymi wynikami pomimo takiej samej precyzji. Warunkiem wystarczającym zbieżności metody Jakobiego dla danego układu równań jest to, żeby promień spektralny był mniejszy od jedynki.