Rozwiązywanie układów równań liniowych metodami bezpośrednimi

Marcin Mikuła

Zadanie 1

Elementy macierzy **A** o wymiarze $n \times n$ są określone wzorem:

$$\begin{cases} a_{1j} = 1 \\ a_{ij} = \dfrac{1}{i+j-1} \quad dla \;\; i \neq 1 \end{cases} \qquad i, j = 1, \ldots, n$$

Przyjmij wektor $\boldsymbol{x}$ jako dowolną n–elementową permutację ze zbioru { 1, -1 } i oblicz wektor **b**. Następnie metodą eliminacji Gaussa rozwiąż układ równań liniowych $\mathbf{A}\boldsymbol{x}=\mathbf{b}$ (przyjmując jako niewiadomą wektor $\boldsymbol{x}$). Przyjmij różną precyzję dla znanych wartości macierzy **A** i wektora **b**. Sprawdź, jak błędy zaokrągleń zaburzają rozwiązanie dla różnych rozmiarów układu (porównaj – zgodnie z wybraną normą – wektory $\boldsymbol{x}$ *obliczony* z $\boldsymbol{x}$ *zadany*). Przeprowadź eksperymenty dla różnych rozmiarów układu.

Zakres n wynosi 3-100, użyte precyzje float32 i float64, przyjęty wektor x składa się naprzemiennie z 1 oraz -1, x = [1, -1, 1, -1, …].

n	Wyniki dla float64
3	1.0 , -1.0 , 1.0 ,
4	1.0 , -1.0 , 1.0 , -1.0 ,
6	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,
8	1.0 , -1.0 , 1.0 , -1.0 , 1.0000001 , -1.0000001 , 1.0 , -1.0 ,
10	1.0 , -0.99999992 , 0.99999887 , -0.999992 , 0.99996863 , -0.99992652 , 0.99989471 , -0.99990954 , 0.9999572 , -0.99999143 ,
12	0.99999941 , -0.99996221 , 0.99919827 , -0.99172934 , 0.95111984 , -0.81988143 , 0.56753005 , -0.31190968 , 0.28013607 , -0.52390499 , 0.81951835 , -0.97011435 ,
14	0.99999839 , -0.99987809 , 0.99693457 , -0.96208463 , 0.72726558 , 0.24775735 , -2.821475 , 7.05684 , -10.817507 , 10.992585 , -7.2147285 , 2.5977962 , 0.10075086 , -0.90425428 ,
16	1.0000003 , -1.0000287 , 1.0007737 , -1.0099144 , 1.0704415 , -1.2924074 , 1.6662906 , -1.3963528 , -1.3618239 , 7.4147673 , -13.573008 , 14.637213 , -9.7861643 , 3.6341052 , -0.11663269 , -0.88726018 ,
18	1.0000001 , -1.0000087 , 1.0002725 , -1.0038892 , 1.029806 , -1.1272942 , 1.2560957 , -0.81278282 , -1.6668658 , 6.959016 , -12.297357 , 12.500419 , -6.4819511 , -0.38410709 , 3.1216439 , -2.4549751 , 1.4011829 , -1.0392044 ,

Tabela 1. Tabela wyników dla precyzji float64

n	Float32	Float64
2	0.00000e+00	0.00000e+00
4	6.64652e-15	3.01871e-13
6	5.44100e-11	3.63798e-10
8	1.97221e+01	1.20335e-07
10	1.81574e+01	1.66203e-04
12	4.69258e+00	1.21398e+00
14	5.70722e+00	2.11156e+01
16	4.89353e+01	2.59432e+01
18	7.33821e+01	2.21990e+01
20	2.65256e+02	8.71401e+02
22	5.08758e+01	5.29518e+01
24	1.80496e+02	1.23774e+02
26	2.56446e+02	1.11393e+02
28	1.48963e+03	1.57776e+03
30	8.07414e+01	1.89246e+02
32	4.76402e+01	6.72210e+01
34	5.92480e+01	5.53441e+02
36	2.16845e+02	3.53450e+02
38	3.07429e+03	3.77185e+02
40	3.75507e+02	2.70230e+02
42	1.26124e+02	3.19532e+02
44	1.44788e+02	3.75632e+02
46	1.80565e+02	1.08220e+03
48	5.65476e+02	1.75233e+03
50	5.09456e+02	2.46086e+02
52	5.14735e+02	1.61072e+04
54	6.67503e+02	5.96323e+02
56	5.09443e+02	1.24163e+02
58	5.34945e+02	1.20849e+03
60	2.17668e+02	2.75634e+02
62	2.74899e+02	1.09973e+03
64	1.84302e+02	3.73795e+02
66	1.65714e+02	6.00618e+02
68	3.28770e+02	5.05802e+02
70	3.16937e+02	9.68115e+02
72	4.22425e+02	4.05753e+02
74	2.06689e+02	2.38284e+02
76	3.72820e+02	4.12513e+02
78	7.22018e+02	2.03857e+02
80	3.30165e+02	5.08628e+02
82	4.24376e+02	7.73650e+02
84	2.74024e+02	1.16940e+04
86	3.03358e+02	1.24309e+04
88	2.93164e+02	6.09119e+04
90	3.00330e+02	1.76003e+03
92	3.95333e+02	1.68947e+03
94	7.66381e+02	1.16441e+03
96	1.51410e+03	1.93032e+04
98	5.48100e+02	1.33434e+03
100	3.35457e+03	3.77706e+03

Tabela 2. Tabela błędów

W tabeli 1 już dla n = 12 pojawiają się znaczące błędy w rozwiązaniu.

Z tabeli 2 wynika że wraz ze wzrostem rozmiaru układu wartości błędów rosną. Wpływ na dokładność wyniku ma także użyta precyzja typu, dla mniejszej wartości błędów są większe.

Zadanie 2

Powtórz eksperyment dla macierzy zadanej wzorem:

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

Porównaj wyniki z tym, co otrzymano w przypadku układu z punktu 1). Spróbuj uzasadnić, skąd biorą się różnice w wynikach. Sprawdź uwarunkowanie obu układów.

n	Wyniki dla float64
3	1.0 , -1.0 , 1.0
4	1.0 , -1.0 , 1.0 , -1.0
6	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0
8	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0
10	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0
12	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0
14	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0
16	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0
18	1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0

Tabela 3. Tabela wyników dla precyzji float64

n	Float32	Float64
2	0.00000e+00	0.00000e+00
4	4.44267e-08	2.48253e-16
6	1.50075e-07	9.74217e-16
8	1.39362e-06	4.67218e-15
10	1.40241e-06	3.08274e-15
12	4.87964e-06	1.98040e-14
14	5.45612e-06	2.27677e-14
16	8.61665e-06	3.80102e-14
18	8.76560e-06	3.65198e-14
20	1.26324e-05	3.80900e-14
22	1.32394e-05	4.75961e-14
24	1.77992e-05	3.68189e-14
26	1.80582e-05	3.97499e-14
28	3.37062e-05	1.01729e-13
30	3.90795e-05	9.93568e-14
32	3.82589e-05	1.22503e-13
34	3.82761e-05	1.27744e-13
36	2.76824e-05	1.81115e-13
38	3.09757e-05	1.91015e-13
40	7.56912e-05	2.56295e-13
42	7.61771e-05	2.56017e-13
44	7.34220e-05	2.20919e-13
46	8.38282e-05	2.43778e-13
48	1.12287e-04	3.16843e-13
50	1.13042e-04	3.46057e-13
52	1.52723e-04	3.67188e-13
54	1.52182e-04	3.95670e-13
56	2.20486e-04	5.07994e-13
58	2.23116e-04	5.38007e-13
60	2.89830e-04	8.55487e-13
62	2.89842e-04	8.52088e-13
64	2.89792e-04	1.02174e-12
66	3.02815e-04	1.20444e-12
68	3.23898e-04	8.29059e-13
70	3.63193e-04	9.25090e-13
72	3.67436e-04	1.20634e-12
74	3.80380e-04	1.20104e-12
76	4.17660e-04	1.89808e-12
78	4.21848e-04	1.96682e-12
80	4.33750e-04	1.57121e-12
82	5.39757e-04	1.97529e-12
84	5.19491e-04	2.30233e-12
86	5.33383e-04	2.28022e-12
88	6.27341e-04	2.21554e-12
90	8.02125e-04	2.39984e-12
92	5.29111e-04	2.24959e-12
94	5.31890e-04	2.29189e-12
96	6.04331e-04	2.46800e-12
98	6.18409e-04	2.51404e-12
100	7.46775e-04	2.28769e-12

Tabela 4. Tabela błędów

n	Float32	Float64
100	7.46775e-04	2.28769e-12
150	2.98700e-03	9.66096e-12
200	5.49645e-03	2.55599e-11
250	1.40134e-02	4.81144e-11
300	1.89213e-02	8.84949e-11
350	2.94801e-02	1.29413e-10
400	4.47038e-02	1.62074e-10
450	6.62354e-02	2.44267e-10
500	1.05383e-01	3.62332e-10

Tabela 5. Tabela błędów dla bardzo dużych n

Nawet dla bardzo dużych n błąd wyniku jest niewielki. Wyniki zawarte w tabelkach bardzo dobrze pokazują wpływ precyzji na wynik, miedzy float32 a float64 różnica to aż 8 rzędów wielkości.

Precyzja uzyskanych wyników w problemie 2 jest znacznie lepsza niż w problemie 1.

Powodem przez który się tak dzieje może być metoda wybierania pivotu. W zastosowanej implementacji algorytmu Gaussa jako pivot wybierane są zawsze kolejne elementy na przekątnej macierzy. Dla macierzy z problemu drugiego jest to zawsze 2. W przypadku macierzy z problemu pierwszego kolejne elementy z przekątnej redukują się na tyle mocno, że podczas dzielenia wierszy przez wartość pivotu błędy zaokrągleń stają się duże w porównaniu do współczynników oryginalnej macierzy.

Dodatkowo obliczono wskaźnik uwarunkowania macierzy. Wartość ta jest miarą jak bardzo zmieni się rozwiązanie **x** układu równań w stosunku do zmiany **b**. Jeżeli wskaźnik macierzy jest duży to nawet mały błąd **b** może spowodować duże błędy w **x**. Wskaźnik uwarunkowania jest obliczany ze wzoru:

$$\kappa = ||A^{-1}|| \cdot ||A||$$

n	A1	A2
3	2.160000e+02	1.444444
4	2.880000e+03	1.833333
6	2.268000e+05	2.644444
8	1.286208e+07	3.448413
10	8.841438e+08	4.249206
12	4.407939e+10	5.055219
14	2.459224e+11	5.868898
16	1.084972e+11	6.678405
18	1.187493e+14	7.485025
20	4.003917e+11	8.289565
20	4.003917e+11	8.289565
30	6.472370e+11	12.331882
50	1.023324e+13	20.420510
70	4.698734e+13	28.508027
100	2.863136e+14	40.638622
150	1.264886e+15	60.855672
200	1.180762e+16	81.073053
300	1.506072e+17	121.508544
500	3.030274e+18	202.379076

Tabela 6. Porównanie uwarunkowania macierzy z zadania 1 i zadania 2

Zgodnie z oczekiwaniami wskaźnik uwarunkowania macierzy z problemu pierwszego jest znacznie większy od wskaźnika dla macierzy problemu 2.

Zadanie 3

Powtórz eksperyment dla jednej z macierzy zadanej wzorem poniżej (macierz i parametry podane w zadaniu indywidualnym). Następnie rozwiąż układ metodą przeznaczoną do rozwiązywania układów z macierzą trójdiagonalną. Porównaj wyniki otrzymane dwoma metodami (czas, dokładność obliczeń i zajętość pamięci) dla różnych rozmiarów układu. Przy porównywaniu czasów należy pominąć czas tworzenia układu. Opisz, jak w metodzie dla układów z macierzą trójdiagonalną przechowywano i wykorzystywano macierz A.

(m = 4, k = 6)

$$\begin{cases} a_{i,j} = k \\ a_{i,j+1} = \dfrac{1}{i+m} \\ a_{i,j-1} = \dfrac{k}{i+m+1} \quad dla \quad i > 1 \\ a_{i,j} = 0 \quad dla \quad j < i-1 \quad oraz \quad j > i+1 \end{cases} \qquad i, j = 1, \ldots, n$$

n	Wynik eliminacja Gaussa	Wyniki Thomas
3	[1.0 , -1.0 , 1.0 ,]	[1.0 , -1.0 , 1.0 ,]
4	[1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 ,]
6	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]
8	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]
10	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]
12	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]
14	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]
16	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]
18	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]	[1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 , 1.0 , -1.0 ,]

Tabela 7. Wyniki dla precyzji float64

n	Gauss	Thomas
2	0.00000e+00	0.00000e+00
4	2.48253e-16	2.48253e-16
6	3.33067e-16	3.33067e-16
8	2.71948e-16	2.71948e-16
10	2.93737e-16	2.93737e-16
12	2.93737e-16	2.93737e-16
14	3.68219e-16	3.68219e-16
16	3.14018e-16	3.14018e-16
18	4.00297e-16	4.00297e-16
20	3.84593e-16	3.84593e-16
22	4.00297e-16	4.00297e-16
24	4.71028e-16	4.71028e-16
26	4.71028e-16	4.71028e-16
28	4.71028e-16	4.71028e-16
30	4.83935e-16	4.83935e-16
32	4.83935e-16	4.83935e-16
34	4.96507e-16	4.96507e-16
36	4.96507e-16	4.96507e-16
38	4.96507e-16	4.96507e-16
40	5.43896e-16	5.43896e-16
42	5.43896e-16	5.43896e-16
44	5.87475e-16	5.87475e-16
46	6.37775e-16	6.37775e-16
48	5.55112e-16	5.55112e-16
50	5.97873e-16	5.97873e-16
52	5.97873e-16	5.97873e-16
54	5.97873e-16	5.97873e-16
56	7.10890e-16	7.10890e-16
58	7.10890e-16	7.10890e-16
60	7.10890e-16	7.10890e-16
62	7.10890e-16	7.10890e-16
64	7.52990e-16	7.52990e-16
66	7.52990e-16	7.52990e-16
68	7.52990e-16	7.52990e-16
70	7.85046e-16	7.85046e-16
72	8.15844e-16	8.15844e-16
74	8.45521e-16	8.45521e-16
76	8.45521e-16	8.45521e-16
78	8.45521e-16	8.45521e-16
80	8.74190e-16	8.74190e-16
82	9.01949e-16	9.01949e-16
84	9.01949e-16	9.01949e-16
86	9.01949e-16	9.01949e-16
88	9.01949e-16	9.01949e-16
90	9.08757e-16	9.08757e-16
92	9.08757e-16	9.08757e-16
94	9.08757e-16	9.08757e-16
96	9.35491e-16	9.35491e-16
98	9.35491e-16	9.35491e-16
100	9.35491e-16	9.35491e-16

Tabela 8. Tabela błędów

n	Gauss	Thomas
100	9.35491e-16	9.35491e-16
150	1.16969e-15	1.16969e-15
200	1.34149e-15	1.34149e-15
250	1.46869e-15	1.46869e-15
300	1.53837e-15	1.53837e-15
350	1.65047e-15	1.65047e-15
400	1.78328e-15	1.78328e-15
450	1.86438e-15	1.86438e-15
500	2.05316e-15	2.05316e-15

Tabela 9. Tabela błędów dla bardzo dużych n

Błędy dla każdego n są dokładnie takie same, co wynika z faktu że metody Gaussa i Thomasa są u podstaw takie same, tylko metoda Thomasa ogranicza się do działania na elementach trójdiagonali.

n	Gauss	Thomas
20	0.0000000000	0.0000000000
40	0.0000000000	0.0000000000
60	0.0156254768	0.0000000000
80	0.0000000000	0.0000000000
100	0.0156347752	0.0000000000
120	0.0312464237	0.0000000000
140	0.0312585831	0.0000000000
160	0.0312325954	0.0000000000
180	0.0468499660	0.0000000000
200	0.0468766689	0.0000000000
220	0.0625019073	0.0000000000
240	0.0937483311	0.0000000000
260	0.1115143299	0.0156219006
280	0.1093780994	0.0000000000
300	0.1249995232	0.0000000000
320	0.1406226158	0.0000000000
340	0.1718664169	0.0000000000
360	0.1874673367	0.0000000000
380	0.2031514645	0.0000000000
400	0.2343466282	0.0000000000
420	0.2505497932	0.0000000000
440	0.2812550068	0.0000000000
460	0.3124766350	0.0000000000
480	0.3281273842	0.0000000000
500	0.4062368870	0.0000000000

Tabela 10. Zestawienie czasu działania obu metod

W metodzie Thomasa czasy działania są tak krótkie że nie udało mi się ich zarejestrować, co nie zaskakuje ponieważ wykonuje ona bowiem znaczniej mniej operacji. Algorytm Thomasa wygrywa również w kwestii zajętości pamięci, ponieważ przy założeniu, że oryginalne macierze nie mogą być modyfikowane, wykorzystana implementacja używa dwóch dodatkowych tablic o rozmiarze n, natomiast eliminacja Gaussa potrzebuje dodatkowej macierzy o rozmiarze n x (n + 1).