

# Zadanie Numeryczne 3

Joanna Szewczyk

05.11.2024r

- Treść zadania:

Wyznacz  $y = A^{(-1)}x$  dla

$$A = \begin{pmatrix} 1.01 & \frac{0.2}{1} & \frac{0.15}{1^3} & & & & & & \\ 0.3 & 1.01 & \frac{0.2}{2} & \frac{0.15}{2^3} & & & & & \\ & 0.3 & 1.01 & \frac{0.2}{3} & \frac{0.15}{3^3} & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ & & & & & 0.3 & 1.01 & \frac{0.2}{N-2} & \frac{0.15}{(N-2)^3} \\ & & & & & & 0.3 & 1.01 & \frac{0.2}{N-1} \\ & & & & & & & 0.3 & 1.01 \end{pmatrix}$$

Oraz  $x = (1, 2, \dots, N)^T$ . Ustalamy  $N = 300$ . Oblicz również wyznacznik macierzy A. Zadanie rozwiąż właściwą metodą (uzasadnij wybór) i wykorzystaj strukturę macierzy. Algorytm proszę zaprogramować samodzielnie; wyjątkowo nie należy stosować procedur bibliotecznych z zakresu algebry liniowej ani pakietów algebry komputerowej (chyba, że do sprawdzenia swojego rozwiązania, co zawsze jest mile widziane). Ponadto, potraktuj  $N$  jako zmienną i zmierz czas działania swojego programu w funkcji  $N$ . Wynik przedstaw na wykresie. Jakiej zależności się spodziewamy?

- Wstęp:

Przechowywanie macierzy w przypadku ogólnym zajmuje  $x \cdot n^2$ , gdzie  $x$  to rozmiar typu danych. Gdy używamy typu float64 zajmującego 8B mamy:

Dla $n = 100$ :	$8B \cdot 100^2 = 80 \text{ kB}$
Dla $n = 1\,000$ :	$8B \cdot 1\,000^2 = 8 \text{ MB}$
Dla $n = 10\,000$ :	$8B \cdot 10\,000^2 = 800 \text{ MB}$
Dla $n = 30\,000$ :	$8B \cdot 30\,000^2 = 7,2 \text{ GB}$

Faktoryzacja LU zachowuje strukturę macierzy wstęgowej, co pozwala istotnie uprościć obliczenia. Wiedząc, że elementy macierzy L oraz U poza wstęgą będą równe zero, możemy pominąć ich wyznaczanie. W efekcie konieczne obliczenia ograniczają się wyłącznie do elementów wstęgi, których liczba jest rzędu  $n$ . Co więcej, liczba składników sumy potrzebnych do obliczenia wartości elementów w macierzy L lub U nigdy nie przekracza szerokości wstęgi.

Wykorzystując te właściwości, można zmniejszyć złożoność rozkładu LU dla macierzy wstęgowej do  $O(n)$ . Rozmiar tablicy w takim schemacie przechowywania wyraża się wzorem:  $x \cdot a \cdot n$ , gdzie  $x$  oznacza rozmiar typu danych, a  $a$  to szerokość wstęgi. W naszym przypadku mamy  $a = 4$  oraz  $x = 8 \text{ B}$ .

Spójrzmy teraz na kilka przykładów, aby porównać efektywność takiego rozwiązania:

Dla $n = 100$ :	$8B \cdot 4 \cdot 100 = 3,2 \text{ kB}$
Dla $n = 1\,000$ :	$8B \cdot 4 \cdot 1\,000 = 32 \text{ kB}$
Dla $n = 10\,000$ :	$8B \cdot 4 \cdot 10\,000 = 320 \text{ kB}$
Dla $n = 30\,000$ :	$8B \cdot 4 \cdot 30\,000 = 960 \text{ kB}$

- Wyniki:

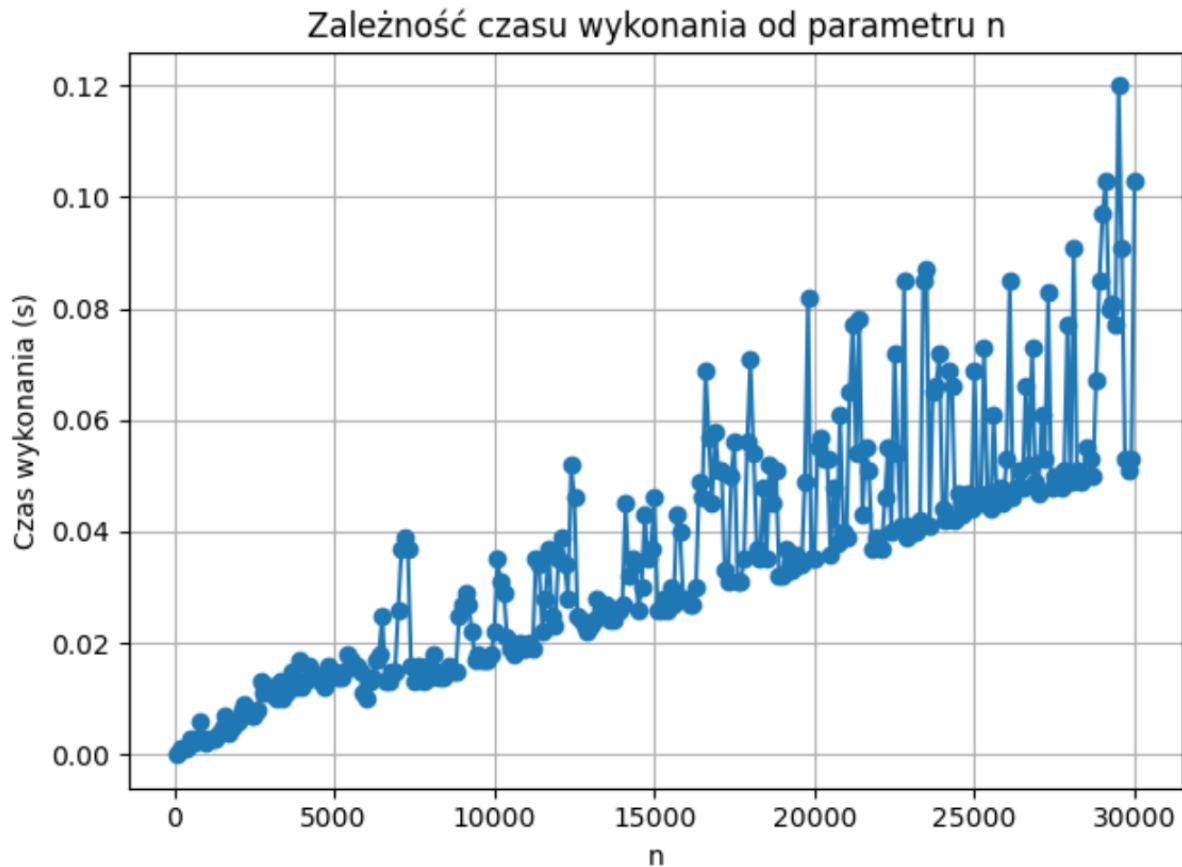
Szukane rozwiązanie dla  $n = 300$  to wektor w postaci  $y = (y_1, y_2, \dots, y_n)^T$ :

[0.33637669442473483, 1.5981824494515218, 2.27082032493809, 3.0848365457457945, 3.844974106923569, 4.616655346834123, 5.382478675076812, 6.14867296978407, 6.91388200016796, 7.67879847542006, 8.443388685013728, 9.207773833842142, 9.971992233232145, 10.736084291921783, 11.500075150377635, 12.263984698578499, 13.027827622658412, 13.79161536358432, 14.555356853219754, 15.31905920134059, 16.08272812746752, 16.846368288288676, 17.60998351277017, 18.373576978660548, 19.137151345293276, 19.900708855257353, 20.664251412998514, 21.427780646278613, 22.1912979546909, 22.954804548308847, 23.718301478732506, 24.48178966422267, 25.245269910196356, 26.00874292605299, 26.77220933907569, 27.535669705984454, 28.29912452259137, 29.062574231912027, 29.826019231013913, 30.589459876825202, 31.35289649108394, 32.11632936457209, 32.879758760752615, 33.64318491890531, 34.40660805684011, 35.17002837325326, 35.933446049779484, 36.69686125278547, 37.46027413494153, 38.22368483660315, 38.98709348702842, 39.750500205453875, 40.51390510204758, 41.27730827875534, 42.04070983005392, 42.80410984362296, 43.56750840094559, 44.330905577846444, 45.09430144497458, 45.85769606823778, 46.62108950919375, 47.384481825403334, 48.14787307074975, 48.911263295727736, 49.674652547705826, 50.43804087116474, 51.20142830791402, 51.964814897289784, 52.72820067633488, 53.491585679963706, 54.2549699411129, 55.01835349087947, 55.781736358647464, 56.5451185722044, 57.30850015784814, 58.0718811404855, 58.83526154372303, 59.59864138995076, 60.36202070041957, 61.125399495312756, 61.888777793812274, 62.652155614160016, 63.41553297371476, 64.178909889005, 64.94228637577791, 65.7056624490451, 66.46903812312497, 67.23241341168236, 67.99578832776534, 68.75916288383968, 69.52253709182098, 70.28591096310467, 71.04928450859403, 71.81265773872664, 72.57603066349884, 73.33940329248884, 74.10277563487843, 74.86614769947313, 75.62951949472145, 76.39289102873265, 77.15626230929372, 77.91963334388514, 78.68300413969584, 79.44637470363736, 80.20974504235691, 80.97311516225015, 81.73648506947265, 82.4998547699514, 83.26322426939504, 84.02659357330403, 84.78996268697986, 85.55333161553422, 86.31670036389711, 87.08006893682511, 87.84343733890879, 88.60680557457995, 89.3701736481184, 90.13354156365845, 90.896909325195, 91.66027693658934, 92.42364440157482, 93.18701172376204, 93.95037890664373, 94.71374595359973, 95.47711286790134, 96.24047965271575, 97.00384631111015, 97.76721284605554, 98.53057926043061, 99.29394555702521, 100.05731173854373, 100.82067780760845, 101.58404376676252, 102.34740961847304, 103.11077536513372, 103.87414100906781, 104.63750655253044, 105.40087199771113, 106.16423734673681, 106.92760260167256, 107.69096776452535, 108.45433283724536, 109.217697821728, 109.98106271981598, 110.74442753330105, 111.50779226392567, 112.2711569133849, 113.03452148332777, 113.79788597535895, 114.56125039104032, 115.32461473189217, 116.08797899939472, 116.85134319498945, 117.6147073200803, 118.37807137603484, 119.14143536418553, 119.90479928583083, 120.66816314223625, 121.43152693463537, 122.19489066423085, 122.95825433219548, 123.72161793967295, 124.48498148777887, 125.24834497760152, 126.01170841020281, 126.77507178661905, 127.5384351078615, 128.3017983749175, 129.06516158875084, 129.8285247503026, 130.59188786049188, 131.35525092021624, 132.11861393035258, 132.88197689175752, 133.6453398052681, 134.40870267170217, 135.1720654918593, 135.93542826652086, 136.6987909964508, 137.46215368239606, 138.22551632508706, 138.98887892523803, 139.7522414835477, 140.5156040006995, 141.27896647736208, 142.04232891418968, 142.80569131182256, 143.5690536708872, 144.332415991997, 145.0957782757522, 145.85914052274063, 146.6225027335378, 147.3858649087072, 148.14922704880075, 148.912589154359, 149.67595122591152, 150.43931326397697, 151.20267526906363, 151.96603724166948, 152.72939918228263, 153.4927610913813, 154.2561229694344, 155.01948481690138, 155.7828466342328, 156.5462084218705, 157.30957018024748, 158.0729319097886, 158.83629361091033, 159.59965528402142, 160.3630169295225, 161.12637854780678, 161.88974013926008, 162.6531017042609, 163.41646324318054, 164.17982475638362, 164.9431862442278, 165.70654770706426, 166.46990914523755, 167.23327055908624, 167.9966319489423, 168.7599933151321, 169.52335465797586, 170.2867159777881, 171.0500772748778, 171.8134385495483, 172.57679980209772, 173.34016103281877, 174.1035222419992, 174.86688342992144, 175.63024459686332, 176.3936057430977, 177.15696686889262, 177.92032797451168, 178.68368906021396, 179.44705012625408, 180.21041117288226, 180.97377220034465, 181.73713320888308, 182.5004941987355, 183.26385517013577, 184.02721612331396, 184.7905770584962, 185.553937975905, 186.31729887575918, 187.08065975827395, 187.84402062366118, 188.607381472129, 189.3707423038825, 190.13410311912324, 190.89746391804968, 191.660824700857, 192.4241854677375, 193.1875462188801, 193.95090695447107, 194.71426767469347, 195.47762837972778, 196.24098906975146, 197.00434974493925, 197.76771040546322, 198.5310710514928, 199.29443168319483, 200.05779230073358, 200.82115290427063, 201.58451349396557, 202.34787406997512, 203.11123463245397, 203.87459518155407, 204.63795571742568, 205.40131624021623, 206.1646767500715, 206.92803724713468, 207.6913977315471, 208.45475820344794, 209.21811866297435, 209.98147911026155, 210.74483954544255, 211.5081999686488, 212.2715603800096, 213.03492077965254, 213.79828116770324, 214.56164154428564, 215.32500190952186, 216.08836226353236, 216.85172260643577, 217.61508293834936, 218.3784432593883, 219.14180356966645, 219.9051638692961, 220.66852415838787, 221.43188443705097, 222.19524470539292, 222.95860496351983, 223.72196521153668, 224.48532544954642, 225.24868567765122, 226.01204589595142, 226.77540610450333, 227.5387663686332, 228.30202726185823, 229.2172196251906]

Wyznacznik macierzy A = 13.826355108346936

Czas programu to: 0.00059223175048828125

Wykres zależności czasu od parametru n:



- Podsumowanie:

Jak możemy zauważyć jeśli naszą macierz rzadką zapiszemy w postaci LU, będziemy mogli uzyskać złożoność liniową  $O(n)$ .

Podczas rozwiązywania układów równań liniowych bardzo istotna jest znajomość struktury macierzy. Pozwala ona na dobranie odpowiedniego sposobu przechowywania macierzy w pamięci (nie musimy stosować ogromnych tablic, które pochłaniałyby straszne ilości pamięci). Dla rozpatrywanego przez przypadku  $n = 30\,000$  optymalizacja reprezentacji pozwala na zmniejszenie zapotrzebowania na pamięć o ok. 98 %. Umożliwia to również wybranie algorytmu faktoryzacji o najmniejszej złożoności, potencjalnie zmniejszając czas procesora  $n^2$  razy.