

Numeryczne wyznaczanie macierzy odwrotnej

Grzegorz Ciesielski

22 lutego 2018

Streszczenie

Problem rozwiązywania układu równań liniowych jest znany matematykom od wieków. Jednym z najwcześniejszych dokumentów poświęconych m. in. temu problemowi jest chiński traktat "The Nine Chapters on the Mathematical Art"¹ z X-II w. p. n. e. Jest to prawdopodobnie pierwszy udokumentowany przypadek wykorzystania macierzy oraz wyznacznika do rozwiązywania układów równań liniowych.

Wyznaczenie macierzy odwrotnej jest problemem trudniejszym niż rozwiązanie układu równań liniowych, jednak ściśle z nim powiązanym. Znalezienie macierzy odwrotnej rozwiązuje układ równań liniowych i umożliwia proste wyznaczanie kolejnych jego rozwiązań, a rozwiązanie pewnej liczby konkretnych układów równań liniowych prowadzi do wyznaczenia macierzy odwrotnej.

W poniższej pracy zajmiemy się problemem numerycznego wyznaczania macierzy odwrotnej z wysoką precyzją.

1 Wprowadzenie

W poniższej pracy będziemy przyjmować następujące oznaczenia:

1. A oznacza odwracalną macierz wymiaru $n \times n$,
2. x jest wektorem długości n ,
3. A_i jest i -tym wierszem macierzy A ,
4. zapis (a_1, a_2, \dots, a_n) , gdzie a_i są wektorami, oznacza macierz, której kolumny są wektorami a_1, a_2, \dots, a_n odpowiednio,
5. zapis $(A_1, \dots, A_k, a_1, \dots, a_n)$, gdzie A_i są macierzami a a_i wektorami oznacza macierz powstałą ze sklejania macierzy A_1, \dots, A_k i wektorów a_1, \dots, a_n wzdłuż wierszy w podanej kolejności.

2 Przegląd metod

Do problemu znajdowania odwrotności macierzy można podejść na dwa sposoby: **pośrednio** i **bezpośrednio**.

Metody pośrednie polegają na sprowadzeniu problemu odwracania macierzy do prostszego problemu, czyli wyznaczenia rozwiązania pojedynczego układu równań liniowych $Ax = b$. Metody bezpośrednie nie korzystają z tego faktu.

2.1 Metody bezpośrednie wyznaczania odwrotności macierzy

Metody bezpośrednie są wygodnymi metodami obliczania macierzy odwrotnej - często można macierz odwrotną zapisać w postaci pewnej liczby wzorów, jednak często nie są numerycznie poprawne lub mają dużą złożoność. W poniższej pracy przedstawimy 2 takie metody ze względu na fakt, iż czasem przydają się do wyznaczania bazy dla rozwiązań iteracyjnych - patrz rozdział 2.3.

2.1.1 Metoda dopełnień algebraicznych

Metoda dopełnień algebraicznych jest bezpośrednią metodą wyznaczania macierzy odwrotnej.

¹Do przeczytania: <http://ctext.org/nine-chapters>

Definicja: $A_{ij} = (-1)^{i+j} \cdot (\text{wyznacznik macierzy powstałej przez usunięcie } i\text{-tego wiersza i } j\text{-tej kolumny z macierzy } A)$.

Twierdzenie: $A \cdot A_{ij}^T = I \cdot \det A$

Dowód: Poniższy dowód wykorzystuje fakt, że $\det A = \sum_{i=1}^n (-1)^{i+j} a_{ij} A_{ij} = \sum_{j=1}^n (-1)^{i+j} a_{ij} A_{ij}$ (rozwiniecie Laplace'a - patrz [Hoh13], rozdział 7).

Zamieńmy i -ty wiersz macierzy A k -tym wierszem. Wówczas z rozwinięcia Laplace'a (i faktu, że powstała macierz jest macierzą osobliwą, gdy $j \neq i$) otrzymamy:

$$\begin{cases} \sum_{j=1}^n a_{kj} A_{ij} = 0 & \text{gdy } k \neq i \\ \sum_{j=1}^n a_{kj} A_{ij} = \det(A) & \text{gdy } k = i \end{cases} \quad (1)$$

oraz podobną tożsamość dla kolumn. Podmieniając zmienne otrzymujemy:

$$A \cdot A_{ij}^T = \left[\sum_{k=1}^n a_{ik} A_{jk} \right]_{ij} \stackrel{(1)}{=} \det A \cdot I \quad (2)$$

Co kończy dowód. Zatem jeśli $\det A \neq 0$, to $\frac{1}{\det A} A_{ij}^T = A^{-1}$.

2.1.2 Metoda rekurencyjna Strassena

Metoda rekurencyjna Strassena jest metodą obliczania macierzy odwrotnej rekurencyjnie, zaproponowana przez Volkera Strassena bazującą na jego algorytmie szybkiego mnożenia macierzy.

Zauważmy, że:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{21} & A_{11}^{-1} \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1} \\ 0 & I \end{bmatrix} \quad (3)$$

gdzie $\Delta = A_{22} - A_{21} A_{11}^{-1} A_{12}$, oraz:

$$\begin{aligned} A^{-1} &= \begin{bmatrix} I & -A_{11}^{-1} A_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & \Delta^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{21} A_{11}^{-1} & I \end{bmatrix} \\ &= \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1} A_{12} \Delta^{-1} A_{21} A_{11}^{-1} & -A_{11}^{-1} A_{12} \Delta^{-1} \\ -\Delta^{-1} A_{21} A_{11}^{-1} & \Delta^{-1} \end{bmatrix} \end{aligned} \quad (4)$$

Zatem możemy wyznaczyć odwrotność macierzy wyznaczając odwrotność mniejszych macierzy: A_{11} oraz Δ . Warto zwrócić uwagę na fakt, że może się zdarzyć, iż macierz A nie będzie osobliwa, ale macierz A_{11} lub Δ już będzie (np. I^T). Wówczas algorytm nie zadziała poprawnie. Więcej informacji dotyczących tej metody (m. in. jak częściowo naprawić wspomniany błąd oraz analizę złożoności) można znaleźć w artykule "*Triangular Factorization and Inversion by Fast Matrix Multiplication*" [BH74].

2.2 Metody pośrednie

Jeśli znajdziemy sposób na rozwiązywanie układów równań liniowych, to wyznaczenie macierzy odwrotnej staje się wyjątkowo proste: wystarczy rozwiązać n układów równań (gdzie n jest wymiarem odwracanej macierzy macierzy):

$$Ax = E_1, \quad Ax = E_2, \quad \dots \quad Ax = E_n \quad (5)$$

gdzie E_1, \dots, E_n są wektorami jednostkowymi. Wówczas jeśli x_1, \dots, x_n są rozwiązaniami powyższych równań, to $A^{-1} = (x_1, x_2, \dots, x_n)$.

Dowód:

$$A \cdot (x_1, x_2, \dots, x_n) = (Ax_1, Ax_2, \dots, Ax_n) = (E_1, E_2, \dots, E_n) = I \quad (6)$$

2.2.1 Metoda eliminacji Gaussa

Metoda eliminacji Gaussa jest metodą rozwiązywania układu równań liniowych postaci $Ax = b$. Polega ona na wykonywaniu określonych operacji na macierzy A i wektorze b tak, by nie zmienić rozwiązania układu równań, a macierz A przekształcić do prostszej postaci. Zauważmy zatem, że wykonanie następujących operacji na macierzy (A, b) :

1. zamiana dowolnych dwóch wierszy miejscami,
2. pomnożenie dowolnych wierszy przez niezerowy skalar,
3. dodanie do jednego wiersza skalarnej krotności innego wiersza,

nie zmieniają rozwiązania układu $Ax = b$. Po wykonaniu dowolnej liczby powyższych operacji w dowolnej kombinacji otrzymujemy macierz (A', b') .

Zauważmy również, że (o ile macierz A jest nieosobliwa) korzystając z powyższych operacji możemy sprowadzić macierz A do macierzy jednostkowej. Jak to dokładnie zrobić? Postępujemy zgodnie z algorytmem:

```
Data:  $A, b$ 
 $M = (A, b)$  ;
for  $i \leftarrow 1$  to  $n$  do
    Posortuj wiersze macierzy  $M$  malejąco względem modułu  $i$ -tej kolumny od  $i$ -tego
    miejsca;
     $M_i \leftarrow \frac{1}{m_{ii}} \cdot M_i$ ;
    for  $j \leftarrow i + 1$  to  $n$  do
         $M_j \leftarrow M_j - m_{ji} \cdot M_i$ ;
    end
end
for  $i \leftarrow n$  to  $2$  do
    for  $j \leftarrow 1$  to  $i - 1$  do
         $M_j \leftarrow M_j - m_{ji} \cdot M_i$ ;
    end
end
Result:  $M^{n+1}$ 
```

Algorithm 1: Eliminacja Gaussa

Gdzie M_i oznacza i -ty wiersz macierzy M , a M^j oznacza j -tą kolumnę macierzy M .

Poprawność algorytmu: Po pierwsze zauważmy, że wykonujemy jedynie operacje takie, jak wyżej, zatem po wykonaniu algorytmu $M^{1..n}x = M^{n+1}$. Wystarczy zatem pokazać, że $M^{1..n} = I$. Rozważmy zatem pierwszą pętlę *for* względem zmiennej i .

Teza: Po i -tej iteracji tej pętli, $m_{kl} = 0$ dla $l \leq i, l < k \leq n$ oraz $m_{ii} = 1$.

Dowód tezy: Druga część tezy jest oczywista - i -ty wiersz jest po posortowaniu (sortowanie macierzy zapobiega przypadkowi, gdy $m_{ii} = 0$, ponieważ nie może się zdarzyć tak, że wszystkie elementy w kolumnie będą zerami - wówczas macierz byłaby osobliwa) macierzy dzielony przez m_{ii} , zatem $m_{ii} = 1$. Wystarczy pokazać, że $m_{ki} = 0$ dla $i < k \leq n$. Łatwo zauważyć, że zagnieżdżona pętla *for* (względem zmiennej j) zeruje m_{ki} dla $k = i + 1 \dots n$ (ponieważ M_i jest wektorem zaczynającym się od $i - 1$ zer i z jedynką na i -tym miejscu), co kończy dowód tezy.

Stąd wniosek, że po wykonaniu się tej pętli, macierz $M^{1..n}$ jest w postaci górnotrójkątnej z jedynkami na głównej współrzędnej.

Analogicznie pokazujemy, że druga pętla *for* względem zmiennej i zeruje elementy nad przekątną macierzy $M^{1..n}$, zatem w wyniku działania algorytmu macierz $M^{1..n}$ jest macierzą jednostkową, co chcieliśmy pokazać.

2.2.2 Wzory Cramera

Wzory Cramera służą rozwiązywaniu układów równań liniowych postaci $Ax = b$, gdzie $A = [a_{ij}]$, $x = [x_i]$, $b = [b_i]$.

Twierdzenie (Cramer, 1750): Jeżeli macierz A jest macierzą nieosobliwą, to układ $Ax = b$ ma dokładnie jedno rozwiązanie dane wzorem:

$$x_i = \frac{\det(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n)}{\det(a_1, a_2, \dots, a_n)} \quad (7)$$

Gdzie a_i jest i -tą kolumną macierzy A .

Dowód: Jeżeli macierz A jest nieosobliwa, to dla dowolnego wektora b istnieje wektor x spełniający układ równań. Wówczas $a_1x_1 + a_2x_2 + \dots + a_nx_n = b$, a z liniowości wyznacznika względem każdej współrzędnej mamy:

$$\begin{aligned} \det(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n) &= \det(a_1, \dots, a_{i-1}, \sum_{k=1}^n a_k x_k, a_{i+1}, \dots, a_n) \\ &= \sum_{k=1}^n x_k \det(a_1, \dots, a_{i-1}, a_k, a_{i+1}, \dots, a_n) \\ &= x_i \det(a_1, a_2, \dots, a_n) \end{aligned} \quad (8)$$

2.3 Metody iteracyjne rozwiązywania układu równań liniowych

Poniższe metody są szczególnymi przypadkami metod rozwiązywania układu równań liniowych:

$$Ax = b \quad (9)$$

Metody te polegają na aproksymowaniu wektora x w kolejnych iteracjach, począwszy od pewnego przybliżenia początkowego x_0 .

2.3.1 Metoda Richardsona

Metodę iteracyjną Richardsona definiuje wzór:

$$x_{k+1} = (I - \tau A)x_k + \tau b \quad (10)$$

Ma ona jednak sporo wad. Pierwszą z nich jest dobór współczynnika τ - należy go odpowiednio wybrać, gdyż źle dobrany współczynnik skutkuje kiepską zbieżnością lub też rozbieżnością metody. Kolejną - zbieżność. Metoda jest rzadko zbieżna, i jej zbieżność nie zależy od doboru punktu startowego, tylko od dodatniej określoności macierzy A . Oznaczmy przez e_k błąd w i -tej iteracji metody Richardsona, tzn. $e_k = |x - x_k|$. Wówczas:

$$e_{k+1} = e_k - \tau A e_k = (I - \tau A)e_k \quad (11)$$

Zatem:

$$\|e_{k+1}\| = \|(I - \tau A)e_k\| \leq \|I - \tau A\| \|e_k\| \quad (12)$$

Stąd otrzymujemy zbieżność metody, o ile $\|I - \tau A\| < 1$, co nie zawsze musi być spełnione, nawet mimo dowolności τ (i faktu, że macierz A jest nieosobliwa). Ponieważ chcemy, by nasza metoda działała dla każdej macierzy o niezerowym wyznaczniku, niezbędne będą pewne usprawnienia.

Dowodzi się, że metoda Richardsona jest (nawet dość szybko) zbieżna dla macierzy symetrycznych. Dlaczego by więc, zamiast szukać rozwiązania układu $Ax = b$, nie rozwiązać za pomocą metody Richardsona równoważnego układu równań $A^T A x = A^T b$? Wówczas macierz $A^T A$ jest symetryczna, a metoda szybko zbiega. Jednak pojawia się pewien problem. Otóż $\text{cond}_2(A^T A) = \text{cond}_2(A)^2$, zatem w przypadku macierzy o umiarkowanie słabym uwarunkowaniu będziemy otrzymywać dość spore błędy, czego chcemy uniknąć.

2.3.2 Szeregi Neumanna

Szeregi Neumanna są szeregami przedstawiającymi odwrotności pewnych macierzy za pomocą macierzowych szeregów potęgowych. Jak się zaraz okaże, można je wykorzystać do poprawiania dokładności odwrotności macierzy, o ile mamy dostępne odpowiednie jej przybliżenie.

Twierdzenie (Neumann): Jeśli A jest macierzą wymiaru $n \times n$ taką, że $\|A\| < 1$, to macierz $I - A$ jest nieosobliwa oraz:

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k \quad (13)$$

Dowód: Prosty dowód faktu, że macierz $I - A$ jest nieosobliwa pozostawiamy czytelnikowi. Udowodnimy, że szereg sum częściowych: $\sum_{k=0}^m A^k$ jest zbieżny do $(I - A)^{-1}$ przy $m \rightarrow \infty$.

Wystarczy sprawdzić, że:

$$(I - A) \sum_{k=0}^m A^k \xrightarrow{m \rightarrow \infty} I \quad (14)$$

Zauważmy, że:

$$(I - A) \sum_{k=0}^m A^k = \sum_{k=0}^m (A^k - A^{k+1}) = A^0 - A^{m+1} = I - A^{m+1} \quad (15)$$

A skoro $\|A^{m+1}\| \leq \|A\|^{m+1} \xrightarrow{m \rightarrow \infty} 0$, więc zachodzi (14).

Twierdzenie to jest przykładem zastosowania potęgowych szeregów macierzy do wyznaczania odwrotności macierzy $I - A$. Jednak my skupimy się w poniższej pracy na prostym wniosku z powyższego twierdzenia:

Wniosek: Jeżeli A, B są macierzami wymiaru $n \times n$ oraz $\|I - AB\| < 1$, to są one nieosobliwe oraz:

$$A^{-1} = B \sum_{k=0}^{\infty} (I - AB)^k, \quad B^{-1} = \sum_{k=0}^{\infty} (I - AB)^k A. \quad (16)$$

Dowód: Z poprzedniego twierdzenia macierz AB jest nieosobliwa oraz

$$B^{-1}A^{-1} = (AB)^{-1} = \sum_{k=0}^{\infty} (I - AB)^k \quad (17)$$

Wymnażając powyższą równość obustronnie przez A z prawej lub B z lewej, otrzymujemy (16).

3 Podejście do problemu

Nasze rozwiązanie opiera się na metodzie eliminacji Gaussa (rozdział 2.2.1), ze względu na dobrą złożoność tej metody, dokładność oraz (praktyczną) poprawność numeryczną. W rozwiązaniu wykorzystujemy również wniosek z twierdzenia Neumanna w celu poprawienia dokładności wyniku.

3.1 Opis metody

Metoda eliminacji Gaussa służy do rozwiązywania układu równań liniowych $Ax = B$, a jak pokazaliśmy na początku rozdziału 2.2 każdą z takich metod można rozszerzyć do znajdowania odwrotności macierzy. W przypadku metody Gaussa rozwiązywanie w ten sposób wielu układów równań: $Ax_1 = b_1, Ax_2 = b_2, \dots, Ax_k = b_k$ naraz jest wyjątkowo proste w implementacji - Wystarczy zastąpić macierz (A, b) macierzą $(A, b_1, b_2, \dots, b_k)$. Wówczas rozwiązania wymienionych układów po wykonaniu algorytmu będą ostatnimi k kolumnami wynikowej macierzy, podobnie jak w zwykłym algorytmie. Zatem aby znaleźć odwrotność macierzy, w metodzie Gaussa należy przyjąć za macierz M macierz (A, I) , która w wyniku iteracji algorytmu przejdzie na (I, A^{-1}) (patrz: paragraf **Poprawność algorytmu**).

3.1.1 Poprawność numeryczna algorytmu eliminacji Gaussa

Twierdzenie (Wilkinson): Jeżeli w wyniku działania algorytmu eliminacji Gaussa dla równania $Ax = b$ otrzymamy zaburzone rozwiązanie $\hat{A}\hat{x} = b$, to:

$$\frac{\|A - \hat{A}\|_{\infty}}{\|A\|_{\infty}} \leq Kn^3 \rho_n \nu, \quad (18)$$

gdzie ν jest precyzją arytmetyki, K niewielką stałą, a ρ_n współczynnikiem wzrostu zdefiniowanym następująco:

$$\rho_n = \frac{\max_{i,j,k} |a_{i,j}^{(k)}|}{\max_{i,j} |a_{i,j}|} \quad (19)$$

Dowód pominiemy - można go znaleźć w książce *"Accuracy and Stability of Numerical Algorithms"* ([Hig02]). Aby zatem udowodnić poprawność numeryczną, musimy ograniczyć współczynnik ρ_n . Okazuje się, że zachodzi ogólne ograniczenie $\rho_n < 2^{n-1}$ (znalezienie przykładowej takiej macierzy pozostawiamy czytelnikowi), a w większości przypadków $\rho_n < n^{\frac{2}{3}}$ (patrz: [Hig02]). Zatem algorytm eliminacji Gaussa jest niemalże poprawny numerycznie - dla dużych n i źle uwarunkowanych macierzy wyniki mogą być kiepskie w arytmetyce zbyt niskiej precyzji, ale na to trudno poradzić.

3.1.2 Skalowany wybór elementów głównych

Zastosujmy eliminację Gaussa do znalezienia odwrotności macierzy:

$$A = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix} \quad (20)$$

W wyniku działania metody macierz (A, I) przechodzi na macierze:

$$\begin{bmatrix} \epsilon & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \epsilon^{-1} & \epsilon^{-1} & 0 \\ 0 & 1 - \epsilon^{-1} & -\epsilon^{-1} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \epsilon^{-1} & \epsilon^{-1} & 0 \\ 0 & 1 & \frac{\epsilon^{-1}}{\epsilon^{-1}-1} & \frac{1}{1-\epsilon^{-1}} \end{bmatrix} \quad (21)$$

Jeżeli ϵ będzie dostatecznie mały, to $fl\left(\frac{\epsilon^{-1}}{\epsilon^{-1}-1}\right) = 1$, zatem macierz przejdzie dalej na:

$$\dots \rightarrow \begin{bmatrix} 1 & 0 & 0 & \frac{\epsilon^{-1}}{\epsilon^{-1}-1} \\ 0 & 1 & 1 & \frac{1}{1-\epsilon^{-1}} \end{bmatrix} \quad (22)$$

Czyli algorytm w wyniku zwróci macierz

$$\begin{bmatrix} 0 & 1 \\ 1 & \frac{1}{1-\epsilon^{-1}} \end{bmatrix} \neq A^{-1} = \begin{bmatrix} \frac{1}{\epsilon-1} & \frac{-1}{\epsilon-1} \\ \frac{-1}{\epsilon-1} & \frac{1}{1-\epsilon^{-1}} \end{bmatrix} \approx \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \quad (23)$$

Problem wynikanie z samego faktu, że ϵ jest mały, ale z tego, że jest on mały **względem** pierwszej kolumny. Jak więc ten problem naprawić? Zauważmy, że gdyby wpierw zamienić wiersze miejscami, problem by nie występował. Zatem pewnym (i, jak się okaże, dobrym) pomysłem może być zaczynanie metody eliminacji od wierszy zaczynających się elementem największym w stosunku do pozostałych elementów w swoim wierszu.

3.1.3 Poprawianie wyniku

Wynik zwracany przez algorytm eliminacji Gaussa może być nieco zaburzony, jednak o ile macierz A nie jest źle uwarunkowana, to zwracana macierz B spełnia warunek $\|AB - I\| < 1$ (testy wykazują, że w arytmetyce podwójnej precyzji dla macierzy o wskaźniku uwarunkowania $< 10^{11}$ warunek ten jest spełniony). W związku z tym, możemy zastosować wniosek (16) z twierdzenia Neumanna, by poprawić dokładność rozwiązania wykorzystując elementarne działania na macierzach. Testy wykazały, że przybliżenie:

$$A^{-1} \approx B \sum_{k=0}^1 (I - AB)^k = 2B - BAB \quad (24)$$

jest w większości przypadków bliższe A^{-1} niż B . Aby zwiększyć skuteczność tej metody, dobrze jest obliczać to przybliżenie w arytmetyce większej precyzji, by nie utracić precyzji podczas odejmowania $2B - BAB$.

3.2 Testy

Zasadnicze testy przeprowadziliśmy na macierzach wymiarów od 2×2 do 500×500 włącznie, o elementach będących losowymi liczbami rzeczywistymi z przedziału $[-1, 1)$ z rozkładem jednostajnym. Większość obliczeń wykonywaliśmy w arytmetyce zmiennoprzecinkowej podwojonej precyzji (*Float64*) - za wyjątkiem poprawiania wyniku szeregiem Neumanna, gdzie jednorazowo zwiększaliśmy precyzję.

Za sposób pomiaru dokładności obliczonej odwrotności macierzy A przyjmujemy normę macierz $AB - I$, gdzie B jest macierzą zwróconą przez algorytm.

Testowaliśmy 4 metody - metodę *inv* z biblioteki języka Julia, naiwną metodę eliminacji Gaussa (GE), metodę Gaussa ze skalowanym wyborem elementów głównych (GE+S) oraz metodę Gaussa ze skalowaniem i poprawianiem wyniku szeregiem Neumanna (GE+SN).

algorytm	rozmiary macierzy	średni czas obliczeń (ms)	średni błąd		maksymalny błąd	
			$\ \cdot\ _2$	$\ \cdot\ _\infty$	$\ \cdot\ _2$	$\ \cdot\ _\infty$
<i>inv</i>	2 - 10	1.096	3.3003e-18	5.1819e-18	9.0987e-17	1.5646e-16
GE		0.644	4.9205e-17	8.4921e-17	1.9496e-15	3.3219e-15
GE+S		0.682	3.0657e-18	4.5612e-18	8.3143e-17	1.3089e-16
GE+SN		3.873	1.4610e-18	2.4476e-18	4.2996e-17	8.0713e-17
<i>inv</i>	12 - 40	12.268	3.6982e-17	7.5121e-17	4.0667e-16	7.9410e-16
GE		14.986	2.5088e-15	4.5129e-15	3.3555e-14	5.6023e-14
GE+S		19.600	3.7326e-17	7.2352e-17	4.4568e-16	8.0256e-16
GE+SN		114.275	7.0999e-18	1.7567e-17	7.9766e-17	1.9303e-16
<i>inv</i>	50 - 100	270.467	8.4363e-16	2.4783e-15	3.9652e-15	1.1755e-14
GE		353.700	1.1750e-13	2.3986e-13	5.6964e-13	1.1506e-12
GE+S		382.167	8.6812e-16	2.1135e-15	4.1039e-15	9.9631e-15
GE+SN		1600.43	9.3680e-17	4.3422e-16	4.4304e-16	2.0630e-15

Tabela 1: Porównanie algorytmów

Powyższa tabela przedstawia osiągnięte wyniki. Naiwny algorytm Gaussa zdecydowanie nie spełnia wymagań - daje wyniki dużo gorsze od pozostałych metod. Eliminacja Gaussowska z wyborem elementów głównych daje wyniki bardzo zbliżone do funkcji bibliotecznej. Z kolei dodanie jeszcze poprawiania wyniku zdecydowanie polepsza wyniki, jednak kosztem czasu - algorytm musi przeprowadzić kilka kosztownych obliczeniowo mnożeń macierzy, co odbija się na szybkości działania.

Literatura

- [BH74] James R. Bunch and John E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.
- [Hig02] Nicholas J Higham. *Accuracy and stability of numerical algorithms*, volume 80. Siam, 2002.
- [Hoh13] Franz E Hohn. *Elementary matrix algebra*. Courier Corporation, 2013.