

Obliczanie wyznacznika i macierzy odwrotnej do macierzy rzeczywistej

Jacek Pospychała

26 stycznia 2005 roku

1 Zastosowane algorytmy

1.1 Wyznacznik

Przy obliczaniu wyznacznika wykorzystywane są pośrednie wyniki obliczeń macierzy odwrotnej. Macierz A jest tam m.in. przekształcana w macierz górnotrójkątną U , a z własności macierzy trójkątnych wynika, że wyznacznik równy jest iloczynowi elementów na głównej przekątnej.

Do znalezienia wyznacznika tą metodą potrzebne jest stworzenie $k - 1$ macierzy L_k ($(n - 1)n/2$ dzieleni), gdzie k to liczba kolumn oraz za każdym razem przemnożenie tej macierzy przez A (n^3 mnożeń i $n^2(n - 1)$ dodawań) oraz ostateczne $n - 1$ mnożeń do obliczenia iloczynu elementów na głównej przekątnej przekształconej macierzy A . Oczywiście w programie znajdowanie wyznacznika wymaga jedynie $n - 1$ mnożeń, gdyż pozostałe obliczenia są i tak wykonywane na potrzeby znalezienia macierzy odwrotnej.

1.2 Macierz odwrotna

Do określenia macierzy odwrotnej wykorzystano algorytm rozkładu na czynniki trójkątne. Wynika z niego, że

$$\begin{aligned} A &= LU \\ A^{-1} &= U^{-1}L^{-1} \end{aligned}$$

A macierze U^{-1} i L^{-1} znajduje się w prosty sposób. Pierwszy etap, a więc rozkład na macierz dolnotrójkątną L i górnotrójkątną U , tak by zachodziła równość $A = LU$ został zrealizowany wg. metody Gaussa. Polega ona na tworzeniu dla macierzy A kolejnych macierzy przekształceń L_k , takich, że:

$$U = L_{n-1}L_{n-2}...L_1A = L_zA \quad (1)$$

Macierz L_z to iloczyn macierzy $L_{n-1}L_{n-2}...L_1$. Wynika stąd, że L_z to L^{-1} , a więc tym sposobem znajdowana jest pierwsza macierz odwrotna:

$$\begin{aligned} U &= L_zA|L_z^{-1} \\ L_z^{-1}U &= A \end{aligned}$$

z wcześniejszej zależności $LU = A$ więc:

$$L_z^{-1} = L, L^{-1} = L_z \quad (2)$$

Druga macierz trójkątna, czyli U^{-1} jest budowana z U korzystając z własności:

$$\begin{aligned} U &= [u_{ij}] \\ U^{-1} &= [r_{ij}] \\ r_{kj} &= 0, k < j \\ r_{ii} &= \frac{1}{u_{ij}} \\ r_{kj} &= \frac{-\sum_{i=k}^{j-1} (r_{ki}u_{ij})}{u_{jj}} \end{aligned}$$

Ilość działań na tym etapie jest podobna, jak przy obliczaniu wyznacznika, przy czym dochodzą jeszcze działania związane z tworzeniem macierzy L^{-1} , czyli $n - 1$ mnożeń L_kL .

2 Opis rozwiązania

2.1 Arytmetyka Tvalue

W implementacji zdecydowano się na uniwersalną reprezentację arytmetyki. Arytmetyka ta musi mieć zdefiniowane podstawowe działania: dodawanie, odejmowanie, dzielenie i mnożenie. Dodatkowo, do obliczania błędów, potrzebna jest możliwość porównywania (tylko relacje $<$ i $>$) liczb.

W ramach klasy Tvalue stworzono arytmetyki: zmiennoprzecinkową i przedziałową. W związku z tym, program może rozwiązywać swoje zadanie dla każdego z tych typów. Dodatkowo, w prosty sposób można go rozszerzyć np. o obsługę liczb zespolonych poprzez zdefiniowanie podstawowych działań i relacji. Przy liczbach przedziałowych wykorzystaną bibliotekę dr hab. Andrzeja Marciniaka.

2.2 Działania na macierzach

Do operacji na macierzach stworzono klasę Tmatrix. Oferuje ona przede wszystkim funkcję *odwroc*, która zwraca odwróconą macierz, oraz wyznacznik macierzy danej. Klasa sama w sobie umożliwia obsługę macierzy o dowolnych wymiarach, ale odwracanie możliwe jest tylko dla macierzy kwadratowych. Dodatkowo zdefiniowano operacje dodawania, odejmowania i mnożenia macierzy, oraz przekształcanie w jednostkową I .

2.3 Tworzenie macierzy

Program umożliwia tworzenie pustej macierzy, do wypełnienia przez użytkownika, generowanie losowej macierzy, oraz otwieranie macierzy z pliku. Macierze losowe są tworzone poprzez wygenerowanie losowych elementów tak, że: losowany jest znak, oraz losowana jest liczba zmiennoprzecinkowa z przedziału zadanego wykładnikami potęg liczby 10.

Możliwy jest eksport do programu DERIVE, w którym porównywano wyniki i oceniano dokładność.

2.4 Implementacja algorytmu

Poniżej przedstawiono kroki, w których jest rozwiązywane zadanie odwrócenia macierzy. Implementacja jest zgodna z algorytmem przedstawionym teoretycznie na początku sprawozdania.

1. Utworzenie liczb -1, 0, 1. Ponieważ arytmetyka opiera się na liczbach, których typ jest wybierany w trakcie działania programu, znając już typ danych trzeba utworzyć liczby, które będą wykorzystywane w działaniach.
2. Utworzyć macierze L^{-1} (na początku - jednostkowa), U (kopia macierzy wejściowej), oraz L_k (pusta).
3. Dla każdej kolumny macierzy wejściowej, oprócz ostatniej:
 - (a) utworzyć z macierzy jednostkowej L_k macierz Frobeniusa.
 - (b) Wykonać mnożenie macierzy $L = L_k L$, oraz $U = L_k U$
4. Gdy macierze L^{-1} i U są wyznaczone, obliczyć wyznacznik (korzystając z elementów na przekątnej U).
5. Odwrócić macierz U .
6. Zwrócić wynik, będący iloczynem $U^{-1} L^{-1}$

Opisane powyżej kroki są zawarte w funkcji *odwroc* klasy Tmatrix.

3 Błąd obliczeń

Poprawność wyników została zweryfikowana na dwa różne sposoby. W trakcie dopracowywania algorytmu i pisania sprawozdania, porównywano wyniki z programem Derive. Możliwość ta istnieje nadal, dzięki opcji eksportu macierzy do formatu tego programu matematycznego.

Drugi sposób weryfikacji poprawności wynika z własności, że $AA^{-1} = I$. Otóż jeśli macierz A^{-1} nie jest poprawna, wówczas macierz zawierająca wynik tego mnożenia będzie jedynie zbliżona do macierzy jednostkowej. Ta weryfikacja jest przeprowadzana dla każdej obliczanej macierzy. Również w podsumowaniu odwracania macierzy jest informacja o błędzie maksymalnym. Błąd maksymalny jest znajdujący poprzez przeszukanie wynikowej macierzy AA^{-1} i porównanie z macierzą I . Błąd maksymalny to największa różnica od wartości z macierzy I . Jest to więc błąd bezwzględny.

W programie zrezygnowano z obliczania błędu względnego, dlatego że wymaga on obliczenia ilorazu błędu bezwzględnego przez poprawną wartość oczekiwaną. W naszym przypadku, poza główną przekątną wartościami oczekiwanymi jest zero, bo porównujemy macierze jednostkowe, więc konieczne byłoby dzielenie przez zero.

3.1 Liczby o dużych wykładnikach

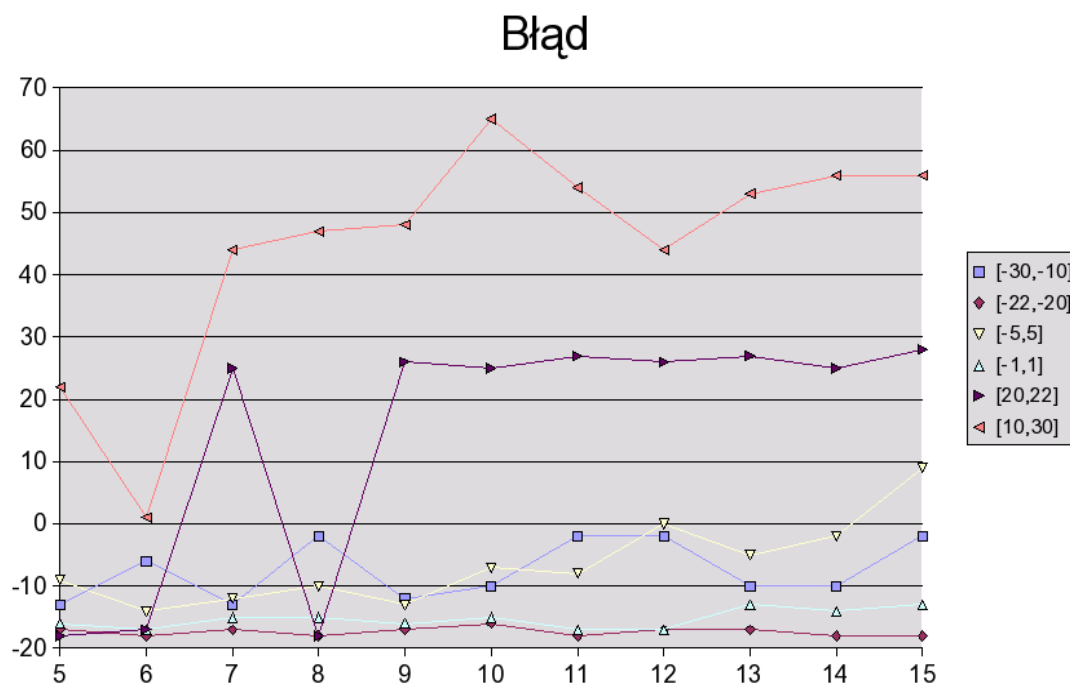
Co ciekawe, porównania z Derive, wykazały, że macierz odwrotna wyliczana przez program, najbardziej poprawna jest na przekątnej. A więc i wyznacznik jest poprawny. Poprawne są również wartości dla elementów ponad przekątną macierzy. Natomiast im większy wykładnik liczb oraz im niższy wiersz macierzy, tym większe mogą być błędy obliczeń pod przekątną. Błędy występują już dla macierzy 5×5 , z elementami o wykładnikach 4-6. Liczby różnią się już na drugim miejscu po przecinku. Im większe wykładniki, tym błędy są większe, a dla liczb o wykładnikach 20, w niektórych miejscach odwróconej macierzy wynikowej niespodziewanie trafiają się potęgi 2, czyli np. liczby 256, 512, 1024. Podsumowując więc, odwracanie macierzy o dużych wykładnikach wiąże się z dużymi błędami. Niezależnie od tego, czy liczby są o zbliżonych wykładnikach, czy różnych. Wynika to z błędów zaokrągleń dla dużych liczb.

3.2 Liczby o małych wykładnikach

Problem opisany wyżej dla dużych liczb nie był tak widoczny dla liczb o wykładnikach w okolicy 10^0 i mniejszych. Kolejnym etapem było więc zweryfikowanie wpływu wielkości macierzy na dokładność obliczeń. Największa przetestowana macierz była rozmiaru 50×50 i zawierała liczby z przedziału $(10^{-22}, 10^{-20})$. Macierz odwrotna była bardzo zbliżona do poprawnej. Iloczyn AA^{-1} na przekątnej miał jedynki, natomiast tam, gdzie powinny być zera, największe liczby były w okolicach 10^{-16} . Stąd wniosek, że im większy rozmiar macierzy, a więc im więcej operacji jest potrzebnych do obliczenia pojedynczego elementu, tym należy się liczyć z większym błędem. Wynika to z niedokładności i błędów mnożenia i dodawania.

3.3 Zależność błędu obliczeń

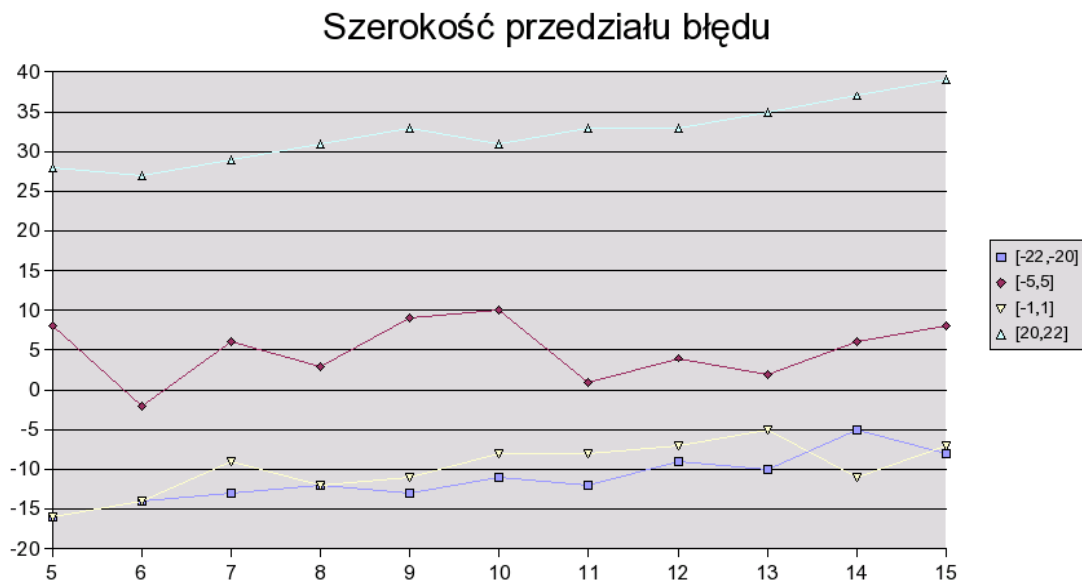
Powyżej omówione zależności pomiędzy wielkością macierzy i rzędem elementów a błędem w następujący sposób zostały zmierzone w programie. Dla macierzy o wielkościach od 5×5 do 15×15 , najmniejsze błędy były rzędu 10^{-20} , a największe w okolicach nawet 10^{70} (dla elementów macierzy $[10^{10}, 10^{30}]$).



Wielkość błędów zależy od wielkości elementów w macierzy wejściowej. Dla liczb o dużych wykładnikach, błąd ma zbliżony (lub większy) wykładnik, a dla liczb o małych wykładnikach (lub ujemnych), błąd jest stosunkowo mniejszy.

Ponieważ błąd był mierzony dla liczb interwałowych, dodatkową informację niesie także szerokość przedziałów błędów. Błąd jest liczony, jako różnica, między wartością oczekiwaną w macierzy jednostkowej (0, lub 1), a wartością

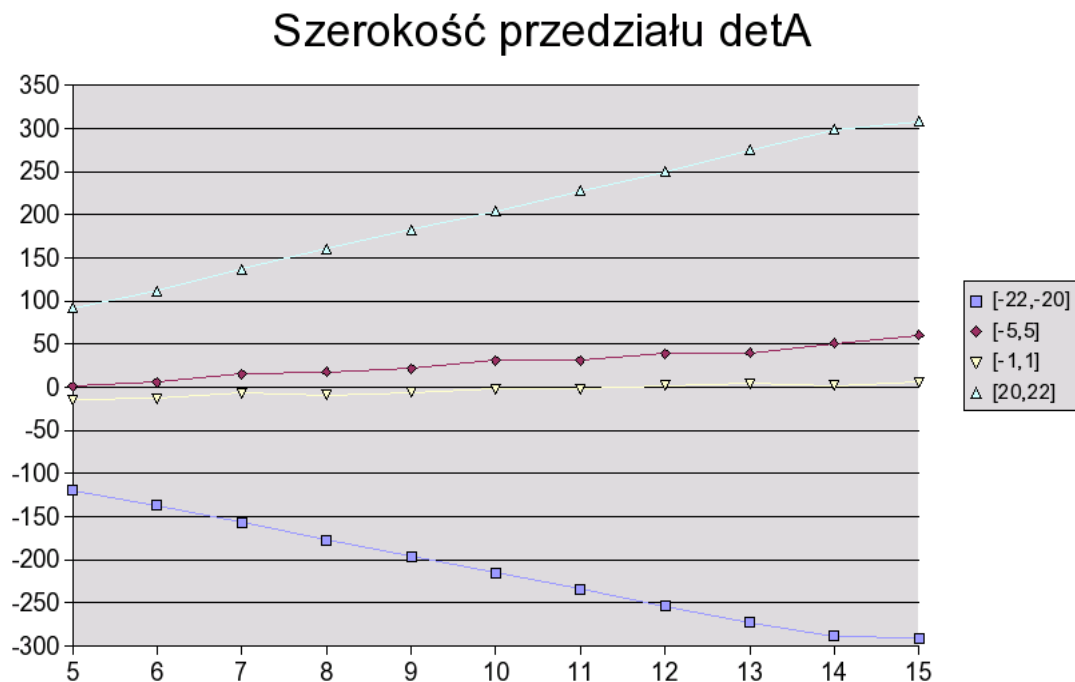
wyliczoną z AA^{-1} . Tak więc szerokość przedziału może być jedynie szerokością przedziału elementu macierzy, który miał największy błąd.



Powyższy wykres potwierdza postawioną hipotezę. A więc, szerokość przedziału błędu odpowiada szerokości przedziałów elementów macierzy. Tj. macierze z elementami o wykładnikach rzędu $[20, 22]$, mają błąd w przedziale o wykładniku 20 i więcej. Macierze z elementami mniejszych rzędów, mają stosunkowo mniejszy przedział błędu.

3.4 Dokładność wyznacznika

Dokładność wyznacznika określa szerokość przedziału, ponieważ wyznacznik również jest liczony w arytmetyce przedziałowej. Zależność szerokości przedziału od rozmiaru macierzy oraz od wykładników elementów przedstawia wykres:



Na podstawie wykresu można stwierdzić, że im większy jest rozmiar macierzy, tym szerokość przedziału $\det A$ jest bardziej ekstremalizowana.