Rachunek macierzowy – program 2

Ernest Roszak, Maksymilian Wojnar

1 Wstęp

W programie zostało zaimplementowane rekurencyjne odwracanie macierzy. Językiem implementacji jest Matlab.

2 Pseudokod rozwiązania

Definiujemy funkcję `Inversion`, która wykonuje rekurencyjne odwracanie macierzy A. Przypadek bazowy zachodzi dla rozmiaru macierzy równego 1, wtedy zwracana jest odwrotność macierzy 1×1, czyli macierz $\left[\frac{1}{a_{1,1}}\right]$. W przeciwnym przypadku macierz A dzielona jest na cztery podmacierze. Dla dwóch z nich wywoływana jest funkcja `Inversion`. Następnie, przy pomocy rekurencyjnego mnożenia macierzy, obliczana jest macierz S. Obliczana jest również jej odwrotność, przy pomocy funkcji `Inversion`. Ostatnim krokiem jest obliczenie czterech podmacierzy B, ponownie z użyciem algorytmu rekurencyjnego mnożenia macierzy, po czym podmacierze są łączone w jedną macierz B, która jest ostatecznie zwracana przez funkcję.

function $\textsc{Inversion}(A,\ l)$
  $n \leftarrow \textsc{Size}(A,\ 1)$

  if $n = 1$ **then**
    $B \leftarrow \frac{1}{A}$
  else
    $A_{11} \leftarrow A(1 : n/2, 1 : n/2)$
    $A_{12} \leftarrow A(1 : n/2, n/2 + 1 : end)$
    $A_{21} \leftarrow A(n/2 + 1 : end, 1 : n/2)$
    $A_{22} \leftarrow A(n/2 + 1 : end, n/2 + 1 : end)$

    $A_{11}^{-1} \leftarrow \textsc{Inversion}(A_{11},\ l)$
    $A_{12}^{-1} \leftarrow \textsc{Inversion}(A_{12},\ l)$

    $S_{22} \leftarrow A_{22} - A_{21} \cdot A_{11}^{-1} \cdot A_{12}$
    $S_{22}^{-1} \leftarrow \textsc{Inversion}(S_{22},\ l)$

    $B_{11} \leftarrow A_{11}^{-1} \cdot \left(I_{n/2} + A_{12} \cdot S_{22}^{-1} \cdot A_{21} \cdot A_{11}^{-1}\right)$
    $B_{12} \leftarrow -A_{11}^{-1} \cdot A_{12} \cdot S_{22}^{-1}$
    $B_{21} \leftarrow -S_{22}^{-1} \cdot A_{21} \cdot A_{11}^{-1}$
    $B_{22} \leftarrow S_{22}^{-1}$

$B \leftarrow [B_{11}, B_{12};$
$B_{21}, B_{22}]$
end if
end function

3 Pseudokod mnożenia macierzy z poprzedniego zadania

Definiujemy funkcję `Matmul`, która wykonuje mnożenie macierzy A i B, używając jednej z dwóch metod w zależności od rozmiaru macierzy. Jako argumenty przyjmuje ona dwie macierze – A i B, oraz parametr l. Jeśli liczba wierszy A jest mniejsza lub równa parametrowi l, funkcja korzysta z klasycznej metody mnożenia macierzy. W przeciwnym razie używamy rekurencyjnej metody Bineta.

Funkcja `ClassicMatmul` wykonuje klasyczne mnożenie macierzy. Jej argumentami są dwie macierze A i B. Dla każdego wiersza macierzy A oraz kolumny B funkcja oblicza iloczyn skalarny, a wynik jest zapisywany w odpowiednim miejscu macierzy C.

Funkcja `BinetMatmul` wykorzystuje rekurencyjną metodę mnożenia macierzy, która dzieli macierze A i B na cztery mniejsze podmacierze, następnie wykonuje operacje mnożenia na tych podmacierzach i sumuje je, aby utworzyć wynikową macierz C. Argumentami funkcji są macierze A i B, oraz parametr l. Proces jest powtarzany aż do osiągnięcia wartości parametru l, w którym funkcja przełącza się na klasyczną metodę mnożenia macierzy.

```
function MATMUL(A, B, l)
    if size(A, 1) ⩽ l then
        C ← ClassicMatmul(A, B)
    else
        C ← BinetMatmul(A, B, l)
    end if
    return C
end function

function CLASSICMATMUL(A, B)
    n, m ← size(A)
    m, k ← size(B)
    C ← matrix of size n × k
    for a ← 1 to n do
        for b ← 1 to k do
            sum ← 0
            for c ← 1 to m do
                sum ← sum + A(a, c) · B(c, b)
            end for
            C(a, b) ← sum
        end for
    end for
    return C
end function
```

function BINETMATMUL(A, B, l)
  $n, m \leftarrow size(A)$
  $A_{11} \leftarrow A(1 : n/2, 1 : m/2)$
  $A_{12} \leftarrow A(1 : n/2, m/2 + 1 : end)$
  $A_{21} \leftarrow A(n/2 + 1 : end, 1 : m/2)$
  $A_{22} \leftarrow A(n/2 + 1 : end, m/2 + 1 : end)$

  $m, k \leftarrow size(B)$
  $B_{11} \leftarrow B(1 : m/2, 1 : k/2)$
  $B_{12} \leftarrow B(1 : m/2, k/2 + 1 : end)$
  $B_{21} \leftarrow B(m/2 + 1 : end, 1 : k/2)$
  $B_{22} \leftarrow B(m/2 + 1 : end, k/2 + 1 : end)$

  $C_{11} \leftarrow Matmul(A_{11}, B_{11}, l) + Matmul(A_{12}, B_{21}, l)$
  $C_{12} \leftarrow Matmul(A_{11}, B_{12}, l) + Matmul(A_{12}, B_{22}, l)$
  $C_{21} \leftarrow Matmul(A_{21}, B_{11}, l) + Matmul(A_{22}, B_{21}, l)$
  $C_{22} \leftarrow Matmul(A_{21}, B_{12}, l) + Matmul(A_{22}, B_{22}, l)$

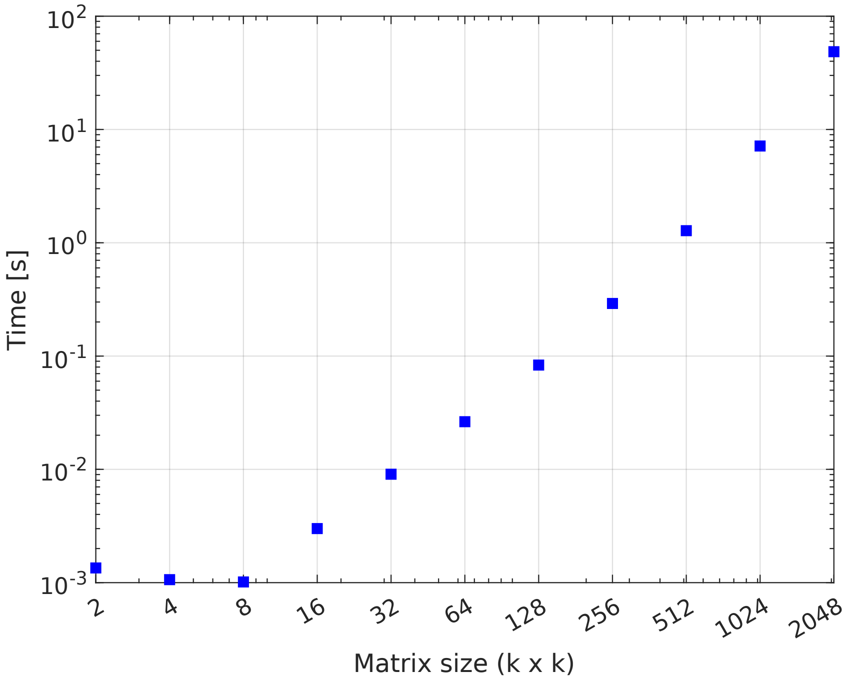
  $C \leftarrow [C_{11}, C_{12};$
    $C_{21}, C_{22}]$
  return C
end function

4 Wyniki

Przygotowaliśmy skrypt, który przeprowadza eksperymenty dotyczące czasu i liczby operacji zmiennoprzecinkowych podczas odwracania macierzy. Skrypt generuje losową macierz A o zadanych rozmiarach oraz wykonuje odwracanie macierzy za pomocą funkcji `Inversion`. Następnie mierzy czas oraz liczbę operacji i zwraca średnią wartość dla każdego rozmiaru macierzy. Każdy eksperyment został powtórzony 5 razy na wylosowanych macierzach zadanego rozmiaru w celu zwiększenia wiarygodności wyników. Do operacji zaliczamy działania dodawania, odejmowania, mnożenia i dzielenia, **nie** jest natomiast zliczana liczba przypisań do zmiennej.

Wykresy zostały wygenerowane dla $l = 64$ (optymalna wartość wyznaczona w poprzednim zadaniu). Wyniki zostały zaprezentowane na wykrsach w skali log-log, aby odpowiednio pokazać różnice w czasie i liczbie operacji.

4.1 Czas odwracania macierzy



4.2 Liczba operacji zmiennoprzecinkowych