

MACIERZE HIERARCHICZNE

JAN SMÓŁKA

ABSTRACT. Niniejszy dokument zawiera sprawozdanie z wykonania ćwiczenia 3. w ramach przedmiotu Algorytmy Macierzowe w semestrze zimowym roku akademickiego 2023/24.

1. ALGORYTMY

W ramach zadania zaimplementowano algorytm konwersji gęstej reprezentacji macierzy do aproksymującej reprezentacji hierarchicznej, oparty o dekompozycję SVD.

1.1. Struktura danych. H-macierz jest przechowywana w strukturze drzewa czwórkowego. Każdy węzeł zawiera informację o podmacierzy wyjściowej macierzy, zadanej przez zakresy wierszy i kolumn, zapisane w polach *rows* i *cols*, z którą jest związany. Pole *state* informuje o klasyfikacji podmacierzy - *Trivial* dla macierzy o co najmniej jednym rozmiarze mniejszym od 3, *Zero* dla macierzy zerowej, *Divided* dla węzła przechodniego (podmacierzy podzielonej rekurencyjnie) oraz *Compressed* dla macierzy skompresowanej. Węzły odpowiadające podmacierzom skompresowanym zawierają referencje do komponentów dekompozycji Truncated-SVD tej podmacierzy, zaś węzły przechodnie - odesłania do węzłów-dzieci, reprezentujących podział podmacierzy na cztery części.

Dla prostoty w ostatniej instrukcji pseudokodu użyto wywołania "*Children*", oznaczającego podział macierzy na cztery podmacierze, rekurencyjne wywołanie na każdej z nich i umieszczenie w bieżącym węźle referencji do wyników rekurencji.

2. IMPLEMENTACJA

Przytoczoną w pseudokodzie procedurę zaimplementowaną w języku Julia, duży nacisk kładąc na optymalizację wydajności czasu działania i przechowywania informacji w węzłach. W tym celu zminimalizowano liczbę kopii i widoków wyjściowej macierzy koniecznych do wykonania konstrukcji.

2.1. Struktury. Implementacja bezpośrednio odwzorowuje pseudokod. Wywołując główną funkcję można wybrać, czy wynik ma być przechowywany w postaci skompresowanej w nadpisanej macierzy wyjściowej, czy w formie dwóch osobnych macierzy trójkątnych. Dodatkowo zawsze zwracany jest wektor z zapisaną permutacją wierszy.

Algorithm 1 Konstrukcja węzła drzewa H-macierzy**Require:** $A \in M_{n \times m}(\mathbb{R}) \wedge n, m \in \mathbb{N}_+ \wedge rank \in \mathbb{N}_+ \wedge tolerance \in \mathbb{R}_+$

```

function NODE( $A, rows, cols, rank, tolerance$ )
  if  $is\_trivial(rows, cols)$  then
    return Node( $Trivial, rows, cols, A[rows, cols]$ )
  end if

  if  $is\_zero(rows, cols)$  then
    return Node( $Zero, rows, cols, null$ )
  end if

   $svd \leftarrow TruncatedSVD(A[rows, cols], rank + 1)$ 

  if  $svd.S[rank + 1] < tolerance$  then
    return Node( $Compressed, rows, cols, svd$ )
  end if

  return Node( $Divided, rows, cols, Children(matrix, rows, cols)$ )
end function

```

```

struct HMatrix{T<:Number}
  root::Node{T}
  rank::Int
  tolerance::T
  error::Float64
end

function HMatrix(matrix::Matrix{T}, rank::Int, tolerance::T) where {T}
  n, m = size(matrix)

  root = Node(
    matrix,
    1:n,
    1:m,
    rank,
    tolerance
  )

  HMatrix(root, rank, tolerance, root.error)
end

```

FIGURE 1. Struktura reprezentująca H-macierz wraz z konstruktorem

```

@enum State begin
    Divided
    Compressed
    Trivial
    Zero
end

struct SVD{T<:Number}
    U::Matrix{T}
    S::Vector{T}
    Vt::Matrix{T}
end

struct Node{T}
    state::State
    rows::UnitRange{Int}
    cols::UnitRange{Int}
    svd::Option{Union{SVD{T}, Matrix{T}}}
    children::Option{Children}
    error::T
end

```

FIGURE 2. Struktura reprezentująca węzeł drzewa H-macierzy wraz z typami pomocniczymi

3. KOSZT OBLICZENIOWY

3.1. Eksperyment. Implementację przetestowano dla macierzy kwadratowych o rozmiarach będących potęgami 2 ze zbioru $\{8, 9, 10, 11\}$, wypełnione wartościami niezerowymi w 1, 2, 5, 10 i 20 procentach. Przyjęto wartości $rank = 3$ oraz $tolerance = 10^{-2}$.

3.2. Pomiar czasu. Czas wykonania mierzono za pomocą standardowego makra dostępnego w języku Julia - `@elapsed`. Mimo, że testowanie przebiegało współbieżnie, dokonane pomiary dotyczą fizycznych odczytów zegara procesora, z pominięciem kolejkowania, oczekiwania i przerw systemowych.

3.3. Wizualizacja kompresji. Przygotowano funkcję służącą do przedstawienia skompresowanej macierzy w formie monochromatycznego obrazu. Działanie przedstawia rysunek 5.

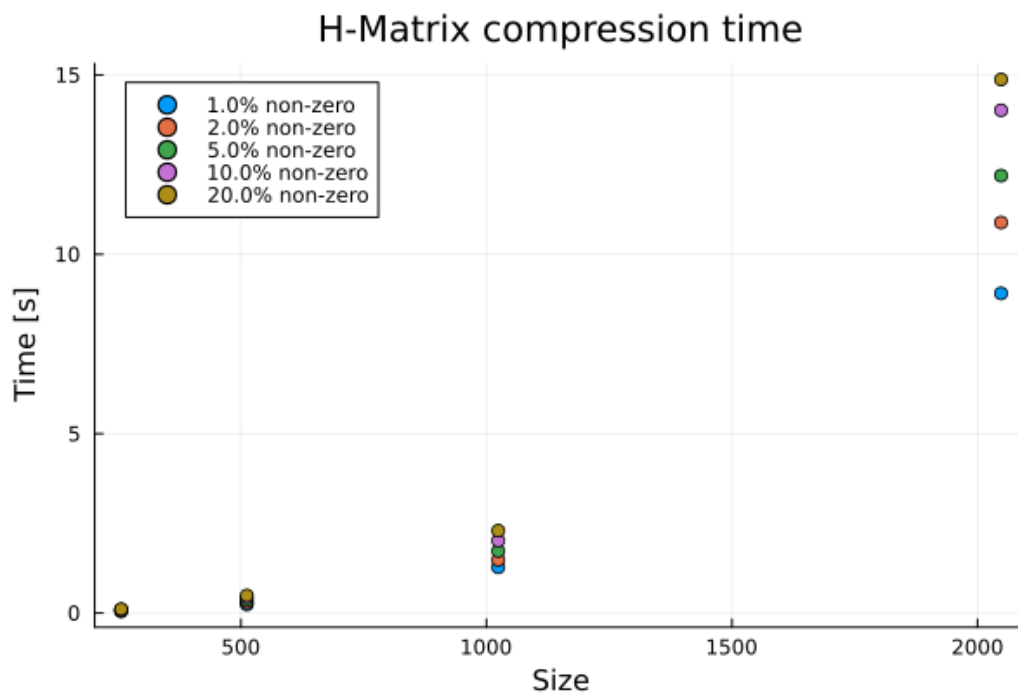


FIGURE 3. Czas działania w zależności od rozmiaru danych wejściowych, dla poszczególnych poziomów wypełnienia macierzy

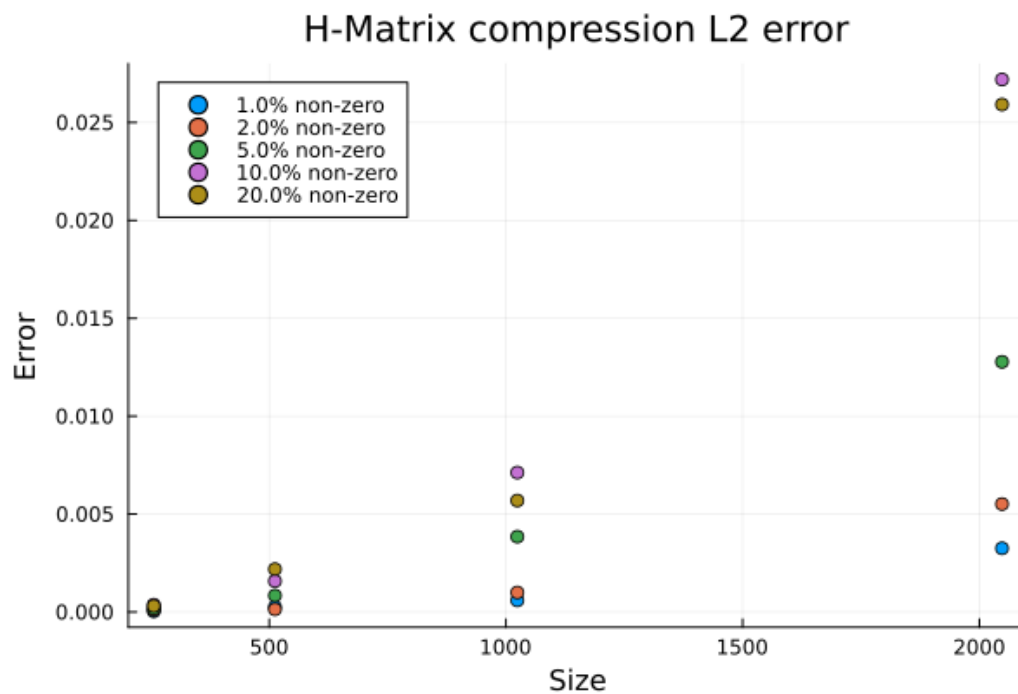


FIGURE 4. Błąd kompresji w zależności od danych wejściowych, dla poszczególnych poziomów wypełnienia macierzy

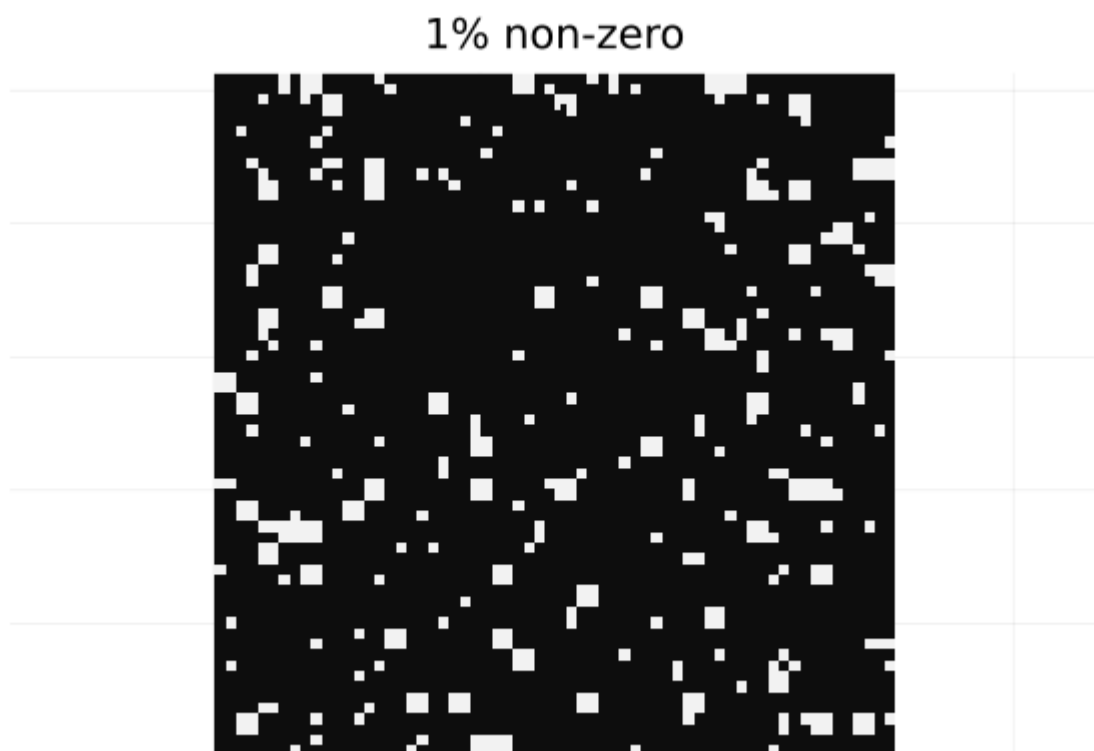


FIGURE 5. Wizualizacja przykładowej macierzy o 1% wartości niezerowych; kolor czarny oznacza składowe zerowe