Hierarchiczna kompresja macierzy

Anna Paszyńska[1], **Maciej Paszyński**[2]
[1] Jagiellonian University, Kraków, Poland,
[2] AGH University of Science and Technology, Kraków, Poland

TRANSPORT CIEPŁA

$h'' = 0$

$h(0) = 0 \quad u(1) = 1$

$$\boxed{u''(x) = 0} \text{ dla } x \in (0,1) \quad u(0) = 0, \quad u(1) = 1$$

$$0.0 = x_0 \leq x_1 = \frac{1}{n} \leq x_2 = 2/n \leq \cdots \leq x_{n-1} = \frac{n-1}{n} \leq x_n = 1.0$$

$\frac{1}{n}$

$$u''(x_i) = 0 \text{ dla } i = 0, \ldots, n \quad u''(x_i) = \frac{u_{i-1} - 2u_i + u_{i+1}}{\cancel{2h}} = 0$$

$2h$

$$u_i = u(x_i)$$

$$\boxed{u_{i-1} - 2u_i + u_{i+1} = 0} \text{ dla } i = 1, \ldots, n-1$$

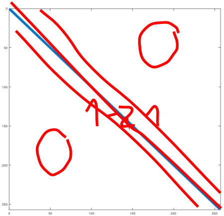
$u_{i-1} \quad u_i \quad u_{i+1}$

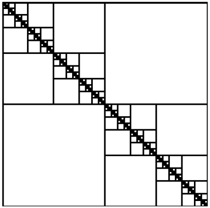
$x_{i-1} \quad x_i \quad x_{i+1}$

$h = \frac{1}{n}$

$0 \quad \frac{i}{n} \quad 1$

$x_0 = 0 \quad x_i = i/n \quad x_n = 1$



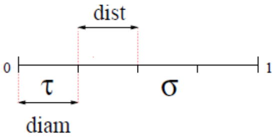


Warunek dopuszczalności: dzielimy dopóki podmacierz nie zawiera jednej nie zerowej wartości $NNZ(A) = 1$

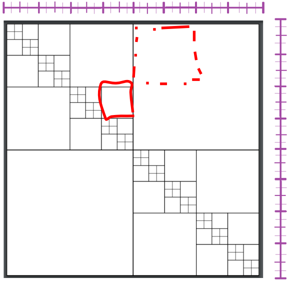
Przykład hierarchicznej kompresji macierzy

Warunek dopuszczalności dla dwóch przedziałów (boków segmentu) τ and σ :
diam$(\tau) \leq$ *dist*(τ, σ) gdzie
diam(τ) to szerokość przedziału, *dist*(τ, σ) to odległość przedziałów

if $diam(\tau) \leq dist(\tau, \sigma)$
then stop recursion



τ z wierszy i kolumny

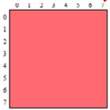


σ

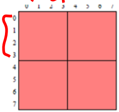
τ, σ OBSZARY SIATKI

WIERSZE I KOLUMNY
= WĘZŁY SIATKI

τ σ



oryginalny τ σ



oryginalny

dist
$\begin{array}{cc} 0 & \frac{1}{7} \\ \frac{1}{7} & 0 \end{array}$

$\{0,\ldots,7\} \times \{0,\ldots,7\}$

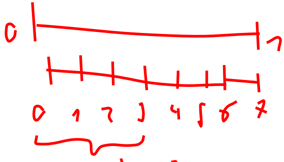
$\mathrm{diam}([0,1]) = 1 \not\leq 0 = \mathrm{dist}([0,1],[0,1]).$

not admissible

$\{0,1,2,3\} \times \{0,1,2,3\}$, $\{0,1,2,3\} \times \{4,5,6,7\}$,
$\{4,5,6,7\} \times \{0,1,2,3\}$, $\{4,5,6,7\} \times \{4,5,6,7\}$.

not admissible

σ

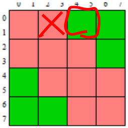


$\mathrm{diam}(\tau) = \frac{3}{7}$

$\mathrm{diam}(\tau) = 1$

$\mathrm{dist}(\tau, \sigma) = 0$

τ



$\mathrm{diam}(\tau) = \frac{1}{7}$

$\mathrm{dist}(\tau, \sigma) = \frac{3}{7}$

DALEJ BYM NIE DZIELIŁ

$\{0,1\} \times \{0,1\}$, $\{0,1\} \times \{2,3\}$, $\{0,1\} \times \{4,5\}$, $\{0,1\} \times \{6,7\}$,
$\{2,3\} \times \{0,1\}$, $\{2,3\} \times \{2,3\}$, $\{2,3\} \times \{4,5\}$, $\{2,3\} \times \{6,7\}$,
$\{4,5\} \times \{0,1\}$, $\{4,5\} \times \{2,3\}$, $\{4,5\} \times \{4,5\}$, $\{4,5\} \times \{6,7\}$,
$\{6,7\} \times \{0,1\}$, $\{6,7\} \times \{2,3\}$, $\{6,7\} \times \{4,5\}$, $\{6,7\} \times \{6,7\}$.

the node $\{0,1\} \times \{4,5\}$ *is admissible, because the corresponding domain is* $[0,\frac{1}{4}] \times [\frac{1}{2},\frac{3}{4}]$:

$$\mathrm{diam}\left(\left[0,\frac{1}{4}\right]\right) = \frac{1}{4} = \mathrm{dist}\left(\left[0,\frac{1}{4}\right],\left[\frac{1}{2},\frac{3}{4}\right]\right).$$

dist

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \qquad \frac{\partial u}{\partial y} = 0 \quad x \in (0,1) \quad y \in \{0,1\}$$

$$\boxed{\Delta u(x) = 0} \text{ dla } x \in (0,1)^2 u(0,:) = 0,\ u(1,:) = 1,\ \cancel{u'(:,0) = u'(:,1) = 0}$$

SIATKA

(x_i, y_i)

$$0.0 = x_0 \le x_1 = \frac{1}{n} \le x_2 = \frac{2}{n} \le \cdots x_{n-1} = \frac{n-1}{n} \le x_n = 1.0$$

$$0.0 = y_0 \le y_1 = \frac{1}{n} \le y_2 = \frac{2}{n} \le \cdots y_{n-1} = \frac{n-1}{n} \le y_n = 1.0$$

ZAPISUJĘ RÓWNANIE W PUNKTACH SIATKI

$$\Delta u(x_i, y_j) = 0 \text{ dla } i, j = 0, \ldots, n \qquad \frac{\partial^2 u}{\partial x^2}(x_i, y_j) + \frac{\partial^2 u}{\partial y^2}(x_i, y_j) = 0$$

$$u_{i,j} = u(x_i, y_j)$$

$$\boxed{\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{2h}} \qquad \boxed{\frac{\partial^2 u_{i,j}}{\partial y^2} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{2h}}$$

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\cancel{2h}} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\cancel{2h}} = 0$$

$u_{i,j} = u(x_i, y_j)$

$$\boxed{u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} = 0}$$

JAK SPOSÓB PRZYBLIŻANIA 2GIEJ POCHODNEJ UPRZYWA NA STWORZENIE MACIERZY

Przenumerowanie z siatki na macierz $[i,j] \rightarrow k = (j-1)*n+i$

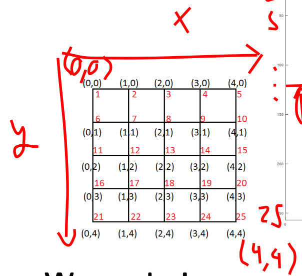
Wiersz $[i,j] \rightarrow k = (j-1)*n+i$ to TEN WIERSZ TO $u_{i,j}$

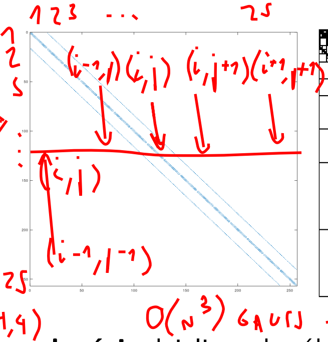
(0,0) - (4,4) → 1 - 25
PUNKTY
WIERSZE WOLNY

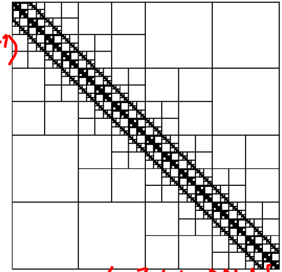
$$u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} = 0$$

$$1*A([i,j],[i-1,j]) - 4*A([i,j],[i,j]) +$$

$$+1*A([i,j],[i+1,j]) + 1*A([i,j],[i,j-1]) + A([i,j],[i,j+1]) = 0$$







$O(N^3)$ GAUSS → $O(N^{3/2})$ SPARSE GAUSS
↓
$O(N)$

Warunek dopuszczalności: dzielimy dopóki podmacierz nie zawiera jednej nie zerowej wartości $NNZ(A) = 1$

WARUNKI BRZEGOWE

$$\Delta u(x) = 0 \text{ dla } x \in (0,1)^3$$

$\frac{\partial u}{\partial y}$ $\frac{\partial u}{\partial z}$

$$u(0,:,:) = 0,\ u(1,:,:) = 1,\ u(:,0,:) = u(:,1,:) = u(:,:,0) = u(:,:,1) = 0$$

$\frac{\partial u}{\partial z}$ $\frac{\partial u}{\partial z}$

SIATKA

$$0.0 = x_0 \le x_1 = \frac{1}{n} \le x_2 = \frac{2}{n} \le \cdots x_{n-1} = \frac{n-1}{n} \le x_n = 1.0$$

$$0.0 = y_0 \le y_1 = \frac{1}{n} \le y_2 = \frac{2}{n} \le \cdots y_{n-1} = \frac{n-1}{n} \le y_n = 1.0$$

$$0.0 = z_0 \le z_1 = \frac{1}{n} \le z_2 = \frac{2}{n} \le \cdots z_{n-1} = \frac{n-1}{n} \le z_n = 1.0$$

ZAPISUJE Δu W PUNKTACH SIATKI

$$\Delta u(x_i, y_j, z_k) = 0 \text{ dla } i,j,k = 0,\ldots,n \quad u_{i,j} = u(x_i, y_j)$$

$$\frac{\partial^2 u_{i,j,k}}{\partial x^2} + \frac{\partial^2 u_{i.j,k}}{\partial y^2} + \frac{\partial^2 u_{i,j,k}}{\partial z^2} = 0 \qquad \frac{\partial^2 u_{i,j,k}}{\partial x^2} = \frac{u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}}{2h}$$

$$\frac{\partial^2 u_{i,j,k}}{\partial y^2} = \frac{u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k}}{2h} \qquad \frac{\partial^2 u_{i,j,k}}{\partial z^2} = \frac{u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1}}{2h}$$

$$\underbrace{\frac{u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}}{\cancel{2h}}}_{\frac{\partial^2 u}{\partial x^2}} + \underbrace{\frac{u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k}}{\cancel{2h}}}_{\partial^2 u/\partial y^2} + \underbrace{\frac{u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1}}{\cancel{2h}}}_{\frac{\partial^2 u}{\partial z^2}} = 0$$

$$u_{i-1,j,k} - 6u_{i,j} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k} + u_{i,j,k-1} + u_{i,j,k+1} = 0$$

WIĘKSZE

Przenumerowanie z siatki na macierz

$[i,j,k] \rightarrow row = (k-1)(j-1)n^2 + (j-1)n + i$

Wiersz $[i,j,k] \rightarrow row = (k-1)(j-1)n^2 + (j-1)n + i$ to

PUNKTY

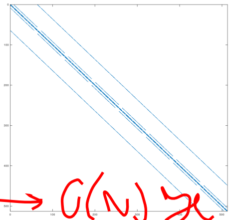
$$u_{i-1,j,k} - 6u_{i,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k} + u_{i,j,k-1} + u_{i,j,k+1} = 0$$

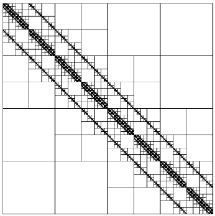
$$1 * A([i,j,k],[i-1,j,k]) - 4 * A([i,j,k],[i,j,k]) + 1 * A([i,j,k],[i+1,j,k])$$
$$+1 * A([i,j,k],[i,j-1,k]) + A([i,j,k],[i,j+1,k]) +$$
$$+1 * A([i,j,k],[i,j,k-1]) + A([i,j,k],[i,j,k+1]) = 0$$

RÓWNANIE (i,j,k)

$O(n^3)$ GAUSS

$O(n^2)$ SKALE GAUSS





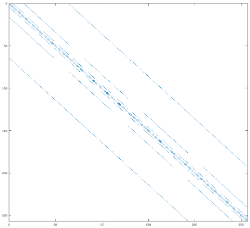
$O(N)$ H

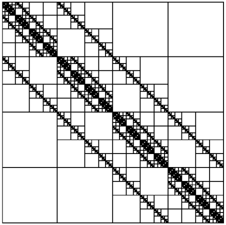
HACKBUSH

$O(N)$ H FORM

LEXING YING

Warunek dopuszczalności: dzielimy dopóki podmacierz nie zawiera jednej nie zerowej wartości $NNZ(A) = 1$

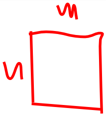




Warunek dopuszczalności: dzielimy dopóki podmacierz nie zawiera jednej nie zerowej wartości $NNZ(A) = 1$

Kompresja macierzy na wartości osobliwe

Niech $n, m, k \in \mathbb{N}$.
Zbiór macierzy o rozmiarze $n \times m$ maksymalnym rzędzie k



$$\mathbf{R(k,n,m)} = \{M \in \mathbb{R}^{n \times m} : rank(M) \leq k\}$$

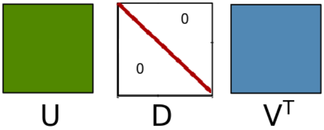
Niech $M \in R(k,n,m)$. **Dekompozycja macierzy na wartości osobliwe (Singular Value Decomposition SVD)** macierzy M to jej forma sfaktoryzowana w postaci

$U^{-1} = U^T$ $\quad$ $V^{-1} = V^T$

$$M = UDV^T$$

gdzie $U \in \mathbb{R}^{nxn}$, $V \in \mathbb{R}^{mxm}$ to macierze unitarne, a $D \in \mathbb{R}^{nxm}$ to macierz diagonalna

$$D_{1,1} \geq D_{2,2} \geq \ldots D_{k,k} \geq D_{k+1,k+1} = \ldots = D_{min(n,m)}, D_{min(n,m)} = 0$$



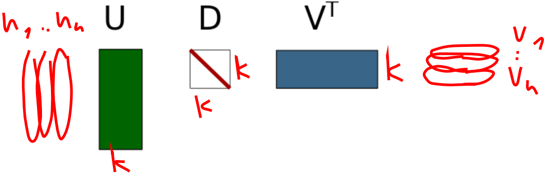
Wartości na przekątnej macierzy diagonalnej D nazywane są **wartościami osobliwymi (singular values)** macierzy M.

Niech $M \in R(k, n, m)$. **Zredukowana dekompozycja na wartości osobliwe (reduced singular value decomposition (rSVD)** macierzy M to jest faktoryzacja w postaci

$$M = UDV^T$$

$$u_i \cdot u_j = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

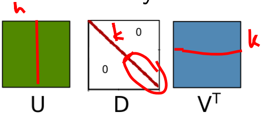
gdzie $U \in \mathbb{R}^{nxk}$, $V \in \mathbb{R}^{mxk}$ mają ortonormalne kolumny / wiersze a macierz $D \in \mathbb{R}^{kxk}$ jest diagonalna i jej wyrazy: $D_{1,1} \geq D_{2,2} \geq ... D_{k,k} > \epsilon > 0$



Uwaga. Z SVD macierzy M możemy uzyskać rSVD poprzez usunięcie kolumn U o indeksach $> k$, wierszy V o indeksach $> k$, oraz wartości osobliwych D o indeksie $> k$.

Lemat.1. (Najlepsza aproksymacja dla ustalonego rzędu)
Niech $M = UDV^T$ to SVD macierzy $M \in \mathbb{R}^{nxm}$

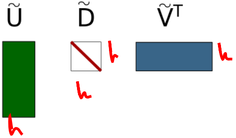
$M =$



odrzucamy wartości osobliwe $D_{k+1,k+1}, \ldots, D_{min(n,m)}$

Niech $\tilde{M} = \tilde{U}\tilde{D}\tilde{V}^T$ gdzie macierze $\tilde{U} = U|_{nxk}$,
$\tilde{D} = diag(D_{1,1}, \ldots, D_{k,k})$, $\tilde{V} = V|_{mxk}$

$\boxed{\|M - \tilde{M}\|_2 = D_{k+1,k+1}}$



$= \tilde{M} \sim M$

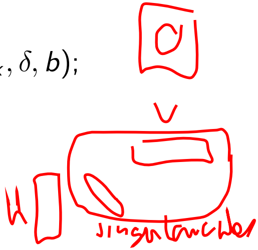
$\|A\|_F = \sqrt{\sum a_{ij}^2}$

Wówczas $\tilde{M}$ jest najlepszą aproksymacją M w sensie
$||M - \tilde{M}|| = min_{N \in R(m,n,k)}||M - N||$,
$||M - \tilde{M}||_2 = D_{k+1,k+1}$, $||M - \tilde{M}||_F = \sqrt{\sum_{i=k+1}^{min(n,m)} D_{i,i}^2}$ w normie spektralnej i w normie Frobeniusa.

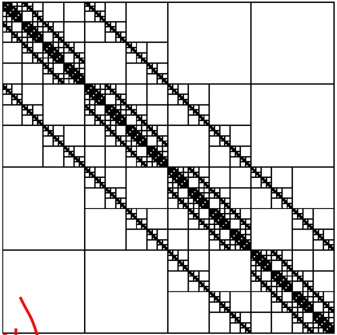
rSVD kompresja bloku macierzy: *node* = **CompressMatrix**($t_{min}, t_{max}, s_{min}, s_{max}, \delta, b$)

Require: $t_{min}, t_{max}, s_{min}, s_{max}$ - range of indexes of block, δ *compression threshold*, b *maximum rank*

if block $(t_{min}, t_{max}, s_{min}, s_{max})$ consist of zeros **then**
    create new node v; $v.rank \leftarrow 0$; $v.size \leftarrow size(t_{min}, t_{max}, s_{min}, s_{max} s, t)$; **return** v;
end if
$[U, D, V] \leftarrow truncatedSVD(t_{min}, t_{max}, s_{min}, s_{max}, \delta, b)$;
$\sigma \leftarrow diag(D)$;
$rank \leftarrow rank(D)$
create new node v; $v.rank \leftarrow rank$;
$v.singularvalues \leftarrow \sigma(1 : rank)$;
$v.U \leftarrow U(*, 1 : rank)$;
$v.V \leftarrow D(1 : rank, 1 : rank) * V(1 : rank, *)$;
$v.sons \leftarrow \emptyset$; $v.size \leftarrow size(t_{min}, t_{max}, s_{min}, s_{max})$;
return v;



space - time formulation



$O(N \log_2 N)$

epsilon=0.1, maxrank=64

Block A, rank=rank(A), size=size(A)
UDV = svd(A)
singular_values = diag(D) (posortowane malejąco)

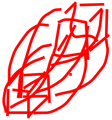
Admissibility condition

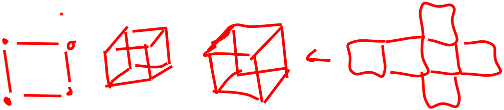
max {i: |singular_values(i)| >epsilon} <=maxrank

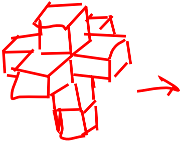
AND

max {i: |singular_values(i)| >epsilon} <=size/2

[jeśli admissibility condition jest spełniony to kompresujemy za pomocą SVD, jeśli nie, to dzielimy rekurencyjnie na 4]

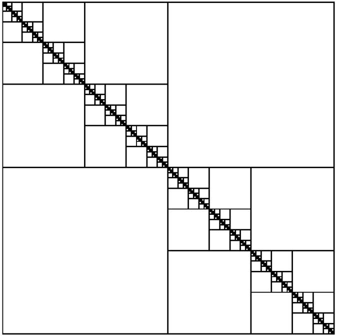






4D cube

Cztero-wymiarowa metoda różnic skończonych



epsilon=1, maxrank=32

Block A, rank=rank(A), size=size(A)
UDV = svd(A)
singular_values = diag(D) (posortowane malejąco)

Admissibility condition

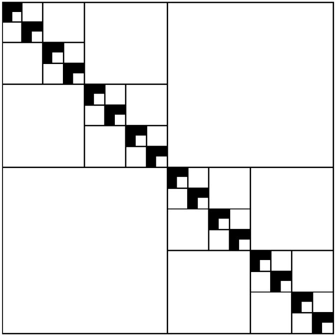
max {i: |singular_values(i)| >epsilon} <=maxrank

AND

max {i: |singular_values(i)| >epsilon} <=size/2

[jeśli admissibility condition jest spełniony
to kompresujemy za pomocą SVD, jeśli nie,
to dzielimy rekurencyjnie na 4]

Cztero-wymiarowa metoda różnic skończonych



epsilon=6, maxrank=256

Block A, rank=rank(A), size=size(A)
UDV = svd(A)
singular_values = diag(D) (posortowane malejąco)

Admissibility condition

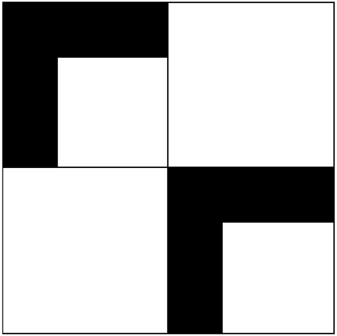
max {i: |singular_values(i)| >epsilon} <=maxrank

AND

max {i: |singular_values(i)| >epsilon} <=size/2

[jeśli admissibility condition jest spełniony to kompresujemy za pomocą SVD, jeśli nie, to dzielimy rekurencyjnie na 4]

Cztero-wymiarowa metoda różnic skończonych



epsilon=7, maxrank=256

Block A, rank=rank(A), size=size(A)
UDV = svd(A)
singular_values = diag(D) (posortowane malejąco)

Admissibility condition

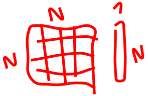
max {i: |singular_values(i)| >epsilon} <=maxrank

AND

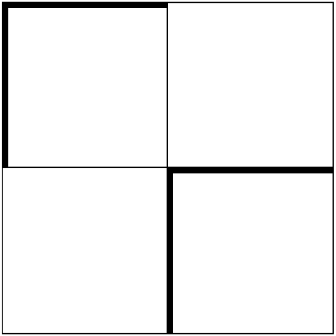
max {i: |singular_values(i)| >epsilon} <=size/2

[jeśli admissibility condition jest spełniony to kompresujemy za pomocą SVD, jeśli nie, to dzielimy rekurencyjnie na 4]

h² bloków
rank 1



O(?)



Block A, rank=rank(A), size=size(A)
UDV = svd(A)
singular_values = diag(D) (posortowane malejąco)

Admissibility condition

max {i: |singular_values(i)| >epsilon} <=maxrank

AND

max {i: |singular_values(i)| >epsilon} <=size/2

[jeśli admissibility condition jest spełniony
to kompresujemy za pomocą SVD, jeśli nie,
to dzielimy rekurencyjnie na 4]

epsilon=10, maxrank=256

h² bloków

rank 1

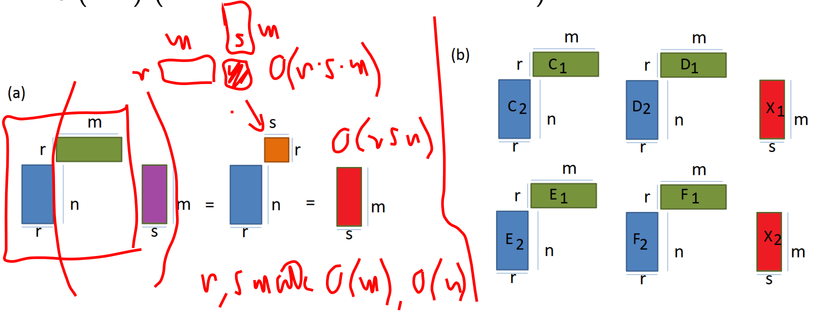
$A \in N \times N$

$\frac{N}{k}$

ILE WYNOSI KOSZT MNOŻENIA [illegible] WEKTOR

Mnożenie macierzy skompresowanej przez wektor

- **lewy panel (liść macierzy hierarchicznej)** koszt mnożenia macierzy skompresowanej przez s wektorów wynosi $\mathcal{O}(rms + rns)$. Gdy $n = m = N \gg r$ koszt wynosi $\mathcal{O}(Nrs)$
- **lewy panel (przypadek w węzłach drzewa macierzy hierarchicznej)** mnożenie macierzy której cztery ćwiartki zostały skompresowane za pomocą SVD zgodnie ze wzorem $\begin{bmatrix} C_2 * (C_1 * X_1) + D_2 * (D_1 * X_2) \\ E_2 * (E_1 * X_1) + F_2 * (F_1 * X_2) \end{bmatrix}$. Koszt obliczeniowy to $\mathcal{O}(Nrs)$ (normalne dodawanie wektorów).



Matrix vector multiplication: $Y = \textbf{matrix\_vector\_mult}(v, X)$

Require: node v, X *vectors to multiply*

```
if v.sons = ∅ then
    if v.rank = 0 then
        return zeros(size(A).rows);
    end if
    return v.U * (v.V * X);
end if
rows ← size(X).rows;
X_1 ← X(1 : rows/2, *); X_2 ← X(rows/2 + 1 : rows, *);
Y_1^(1) ← matrix_vector_mult(v.sons(1), X_1);
Y_1^(2) ← matrix_vector_mult(v.sons(2), X_2);
Y_2^(1) ← matrix_vector_mult(v.sons(3), X_1);
Y_2^(2) ← matrix_vector_mult(v.sons(4), X_2);
return [Y_1^(1) + Y_1^(2) ; Y_2^(1) + Y_2^(2)];
```

$$\begin{bmatrix} Y_1^{(1)} + Y_1^{(2)} \\ Y_2^{(1)} + Y_2^{(2)} \end{bmatrix};$$

0 b = 0

x_1 x_2

PAINT

dodawanie wektorów

SZUKAMY ITERACYJNIE

$\boxed{A x_n} = b$

zgadnięte wcześniej

$r = A x_n - b$

k iteracji

poprawienie $x_n \to x_{n+1}$

$\|r\| < \varepsilon$

$O(N \log_2 N \, k)$

Dodawanie macierzy skompresowanych $Y = \textbf{matrix\_matrix\_add}(A, B)$

Require: node v, node w

if $v.sons = 0$ **and** $w.sons = 0$ **and** $v.rank = 0$ **and** $w.rank = 0$ **then**
  return Zero matrix of proper dimensions
end if
if $v.sons = 0$ **and** $w.sons = 0$ **and** $v.rank! = 0$ **and** $w.rank! = 0$ **then**
  return $\boxed{\text{rSVDofCompressed}}([v.U1 w.U1], [v.V1 w.V1])$
end if
if $v.sons > 0$ **and** $w.sons > 0$ **then**

A1	A2
A3	A4

+

B1	B2
B3	B4

=

$A_1 + B_1$	$A_2 + B_2$
$A_3 + B_3$	$A_4 + B_4$

return

Addition((A1,B1),	Addition (A2,B2))
Addition (A3,B3)	Addition(A4,B4)

end if

STRATNA KOMPRESJA

$0 + 0 = 0$

$v.U_1$ $w.U_1$ $v.V_1$ $w.V_1$

Dodawanie macierzy skompresowanych: $Y = matrix\_matrix\_add(A, B)$

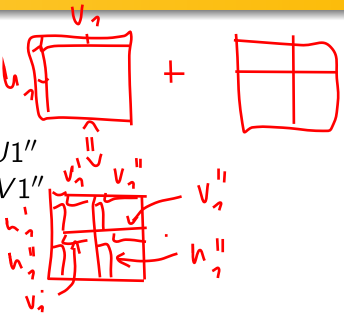
Require: node v, node w

if $v.sons = 0$ **and** $w.sons > 0$ **then**

$A = U1V1$

Podziel $U1$ na 2 czesci $U1 = U1'U1''$

Podziel $V1$ na 2 czesci $V1 = V1'V1''$



U1’*V1’	U1’*V1”
U1”*V1’	U1”*V1”

B1	B2
B3	B4

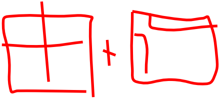
return

Addition((U1’*V1’,B1),	Addition (U1’*V1”,B2))
Addition (U1”*V1’,B3)	Addition(U1’*V1”,B4)

end if

if $v.sons > 0$ AND $w.sons = 0$ **then**

Symetrycznie do poprzedniego przypadku



end if

if v, w to liczby **then**

return $[v + w]$

end if

Mnożenie macierzy skompresowanych: $Y = \textbf{matrix\_matrix\_mult}(A, B)$

Require: node v, node w

```
if v.sons = 0 AND w.sons = 0 then
    if v.rank = 0 AND w.rank = 0 then
        return Zero matrix of proper dimensions
    else
        if v.rank! = 0 AND w.rank! = 0 then
            return v.U(v.V * w.U) * w.V
        end if
    end if
else
    if v.sons > 0 AND w.sons > 0 then
```

$\boxed{0} \cdot \boxed{0} = \boxed{0}$

$O(n)$

$\cancel{(v.U * v.V) * (w.U * w.V)}$

$O(n^2)$

+ addition

* multiply

A1	A2
A3	A4

B1	B2
B3	B4

$A_1 * B_1 + A_2 * B_3$ | $A_1 * B_2 + A_2 * B_4$

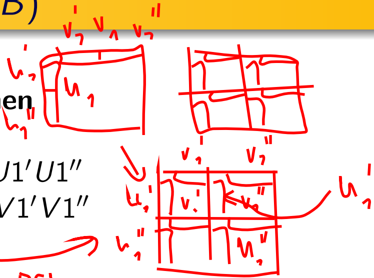
$A_3 * B_1 + A_4 * B_3$ | $A_3 * B_2 + A_4 * B_4$

return

Addition(Multiply(A1,B1), Multiply(A2,B3))	Addition(Multiply(A1,B2), Multiply(A2,B4))
Addition(Multiply(A3,B1), Multiply(A4,B3))	Addition(Multiply(A3,B2), Multiply(A4,B4))

```
    end if
```

Mnożenie macierzy skompresowanych: $Y = \textbf{matrix\_matrix\_mult}(A, B)$



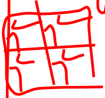
Require: node v, node w

if $v.sons = 0$ **and** $w.sons > 0$ **then**

$A = U1V1$

Podziel $U1$ na 2 czesci $U1 = U1'U1''$

Podziel $V1$ na 2 czesci $V1 = V1'V1''$



U1'*V1'	U1'*V1''
U1''*V1'	U1''*V1''

B1	B2
B3	B4

return

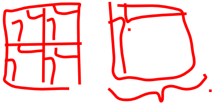
Addition(Multiply(U1'*V1',B1), Multiply(U1'*V1'',B3))	Addition(Multiply(U1'*V1',B2), Multiply(U1'*V1'',B4))
Addition(Multiply(U1''*V1',B1), Multiply(U1''*V1'',B3))	Addition(Multiply(U1''*V1',B2), Multiply(U1''*V1'',B4))

end if

if $v.sons > 0$ **and** $w.sons = 0$ **then**

Symetrycznie do poprzedniego przypadku

end if



if v, w to liczby **then** return v.w

Zadanie ~~5~~ 4

- Proszę wybrać ulubiony język programowania.
- Proszę wygenerować macierz o rozmiarze $2^{3k} = 2^k * 2^k * 2^k$ dla $k = 2, 3, 4$ o strukturze opisującej topologię trójwymiarowej siatki zbudowanej z elementów sześciennych (wiersz = wierzchołek, niezerowe losowe wartości w kolumnach – sąsiadujące wierzchołki siatki)
- Proszę użyć rekurencyjną procedurę kompresji macierzy z Zadania 3.
- Proszę narysować macierz skompresowaną używając rysowacza z zadania 3
- Proszę przemnożyć macierz skompresowaną przez wektor (20 punktów)
- Proszę przemnożyć macierz skompresowaną przez samą siebie (20 punktów)

Raporty (dla par 2 osobowych)

$A \to \mathcal{H}$, $\hookrightarrow \tilde{A}$ $\quad \mathcal{H}x = \bar{y}$, $\tilde{A}x = \tilde{y}$

Proszę przygotować następujący raport

- Proszę w raporcie opisać pseudo-kod swojego rekurencyjnego algorytmu
- Proszę w raporcie umieścić wybrane najbardziej istotne fragmenty kodu
- Proszę narysować wykres: oś pozioma rozmiar macierzy 2^{3k} dla $k = 2, 3, 4$, oś pionowa czas mnożenia macierzy skompresowanej przez wektor.
- Proszę dopasować wykres αN^{β} do wykresów czasu mnożenia. Jaka jest złożoność eksperymentalna?
- Proszę napisać dekonstrukcje macierzy gęstej z macierzy skompresowanej i porównać wyniki mnożenia macierzy przez wektor licząc sumę różnicy kwadratów $||Ax - Hx||_F^2 = \sum_i (Ax_i - Hx_i)^2$

NORMA FROBENIUSA^2

$\tilde{y}$ $\quad \bar{y}$

Raporty (dla par 2 osobowych)

$A \rightarrow \mathcal{H}$ $\mathcal{H}^2 = \mathcal{H} \cdot \mathcal{H}$

Proszę przygotować następujący raport

- Proszę w raporcie opisać pseudo-kod swojego rekurencyjnego algorytmu
- Proszę w raporcie umieścić wybrane najbardziej istotne fragmenty kodu
- Proszę narysować wykres: oś pozioma rozmiar macierzy 2^{3k} dla $k = 2, 3, 4$, oś pionowa czas mnożenia macierzy skompresowanej przez samą siebie.
- Proszę dopasować wykres αN^{β} do wykresów czasu mnożenia. Jaka jest złożoność eksperymentalna?
- Proszę napisać dekonstrukcje macierzy gęstej z macierzy skompresowanej i porównać wyniki mnożenia macierzy licząc sumę różnicy kwadratów $||A^2 - H^2||_F^2 = \sum_{i,j}(A^2{}_{ij} - H^2{}_{ij})^2$

$A^2 - \mathcal{H}^2$ NORMA FROBENIUSA