Programowanie i metody numeryczne: podstawy **algebry liniowej** *czyli conieco o macierzach*

Jędrzej Wardyn

25.04.2024

FACULTY OF PHYSICS
UNIVERSITY OF WARSAW

Dodawanie macierzy: przyklad 3×3

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} + \begin{pmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{pmatrix}$$

Dodawanie macierzy: Ogólnie

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \cdots & a_{nn} + b_{nn} \end{pmatrix}$$

Asymptotyczna złożoność obliczeniowa: $O(n^2)$

(O od Ordnung)

Mnożenie macierzy przez skalar: przyklad

$$3 \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{pmatrix}$$

Mnożenie macierzy przez skalar ogólnie

$$c \cdot \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} c \cdot a_{11} & c \cdot a_{12} & \cdots & c \cdot a_{1n} \\ c \cdot a_{21} & c \cdot a_{22} & \cdots & c \cdot a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c \cdot a_{n1} & c \cdot a_{n2} & \cdots & c \cdot a_{nn} \end{pmatrix}$$

Mnożenie macierzy przez skalar ogólnie

$$c \cdot \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} c \cdot a_{11} & c \cdot a_{12} & \cdots & c \cdot a_{1n} \\ c \cdot a_{21} & c \cdot a_{22} & \cdots & c \cdot a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c \cdot a_{n1} & c \cdot a_{n2} & \cdots & c \cdot a_{nn} \end{pmatrix}$$

Asymptotyczna złożoność obliczeniowa: $O(n^2)$...

Mnożenie macierzy przez skalar ogólnie

$$c \cdot \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} c \cdot a_{11} & c \cdot a_{12} & \cdots & c \cdot a_{1n} \\ c \cdot a_{21} & c \cdot a_{22} & \cdots & c \cdot a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c \cdot a_{n1} & c \cdot a_{n2} & \cdots & c \cdot a_{nn} \end{pmatrix}$$

Asymptotyczna złożoność obliczeniowa: $O(n^2)$...

...ale czy można szybciej mnożyć?

(niektóre)Metody Szybszego Mnożenia

- $O(n^{1.585})$ Metoda Karatsuby
- $O(n^{\frac{log(2k-1)}{log(k)}})$ Metoda Toom-Cook k-tego rzędu
- $O(n\log(n)loglog(n))$ Metoda Schönhage-Strassen (algorytm "galaktyczny"), używa szybkiej transformaty Fouriera

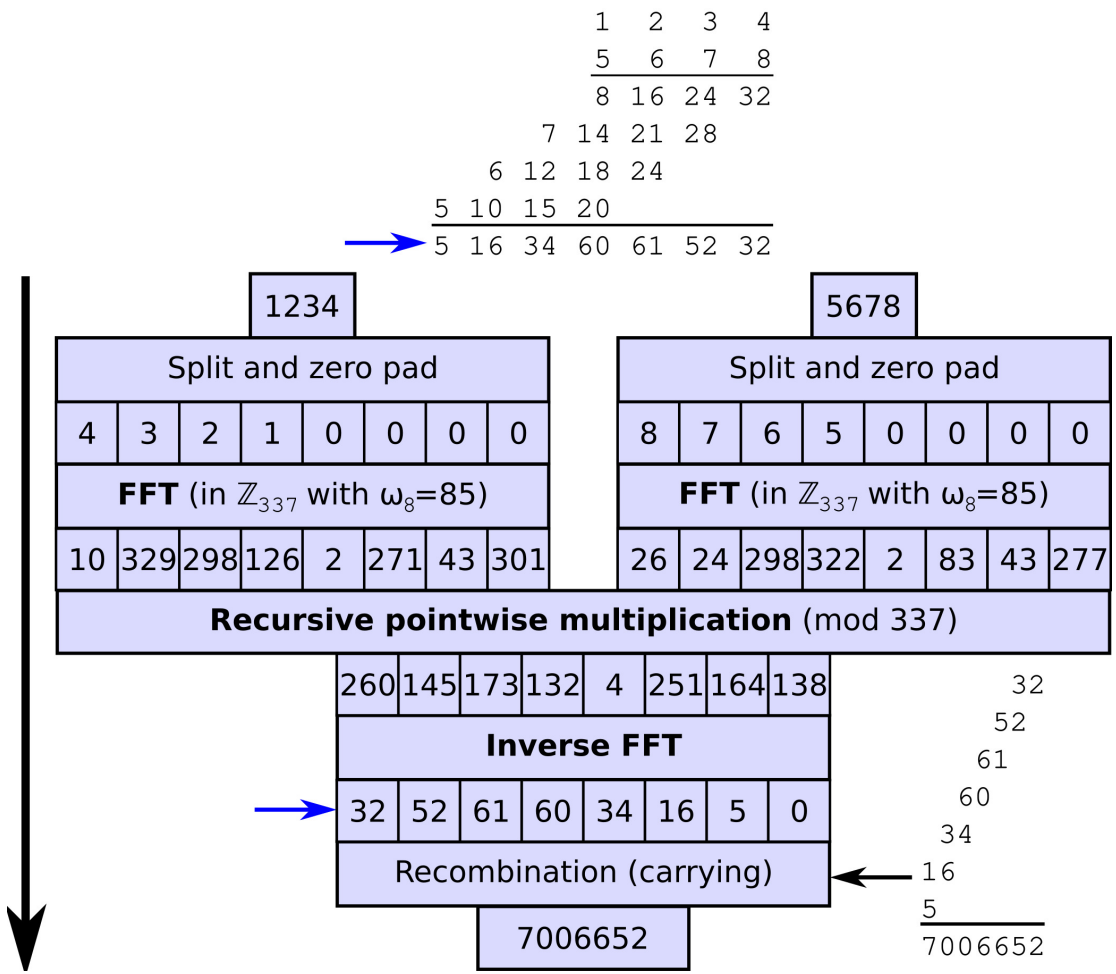
źródła:

`en.wikipedia.org/wiki/Multiplication_algorithm#Computational_complexity_of_multiplication`

`en.wikipedia.org/wiki/Galactic_algorithm`

`en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations#Arithmetic_functions`

Metoda Szybszego Mnożenia Schönhage-Strassen



(niektóre)Metody Szybszego Mnożenia

- $O(n^{1.585})$ Metoda Karatsuby
- $O(n^{\frac{log(2k-1)}{log(k)}})$ Metoda Toom-Cook k-tego rzędu
- $O(n\log(n)loglog(n))$ Metoda Schönhage-Strassen (algorytm "galaktyczny"), używa szybkiej transformaty Fouriera
- $O(nlog(n))$ (2019) Harvey-Hoven algorithm [link]

źródła:

`en.wikipedia.org/wiki/Multiplication_algorithm#Computational_complexity_of_multiplication`

`en.wikipedia.org/wiki/Galactic_algorithm`

`en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations#Arithmetic_functions`

Mnożenie macierzy przez wektor

Example:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{pmatrix}$$

Asymptotyczna złożoność obliczeniowa: $O(n^2)$

Iloczyn wektorowy (cross product)

Iloczyn wektorowy może być również wyrażony jako iloczyn macierzy skośnosymetrycznej i wektora:

Dla dwóch wektorów $\mathbf{u} = [u_1, u_2, u_3]^T$ i $\mathbf{v} = [v_1, v_2, v_3]^T$, iloczyn wektorowy jest równy:

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Mnożenie macierzy tradycyjnie, przykład

$$\begin{bmatrix} -2 & 1 \\ 0 & 4 \end{bmatrix} \times \begin{bmatrix} 6 & 5 \\ -7 & 1 \end{bmatrix} = \begin{bmatrix} -2\times6+1\times-7 & -2\times5+1\times1 \\ 0\times6+4\times-7 & 0\times5+4\times1 \end{bmatrix}$$

$$= \begin{bmatrix} -19 & -9 \\ -28 & 4 \end{bmatrix}$$

Mnożenie macierzy tradycyjnie, w ogólności:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

Mnożenie macierzy tradycyjnie, w ogólności:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

gdzie:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

Asymptotyczna złożoność obliczeniowa: $O(n^3)$, jednakże:

Metoda mnożenia macierzy blokowo (tradycyjnie)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} \times B_{11} + A_{12} \times B_{21} & A_{11} \times B_{12} + A_{12} \times B_{22} \\ A_{21} \times B_{11} + A_{22} \times B_{21} & A_{21} \times B_{12} + A_{22} \times B_{22} \end{bmatrix}.$$

Metoda mnożenia macierzy Strassena

$M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22});$

$M_2 = (A_{21} + A_{22}) \times B_{11};$

$M_3 = A_{11} \times (B_{12} - B_{22});$

$M_4 = A_{22} \times (B_{21} - B_{11});$

$M_5 = (A_{11} + A_{12}) \times B_{22};$

$M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12});$

$M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22}),$

Redukcja mnożeń z 8 do 7. Wtedy:

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{bmatrix}.$$

Metoda mnożenia macierzy Strassena

$M_1 = (A_{11} + A_{22}) \times (B_{11} + B_{22});$

$M_2 = (A_{21} + A_{22}) \times B_{11};$

$M_3 = A_{11} \times (B_{12} - B_{22});$

$M_4 = A_{22} \times (B_{21} - B_{11});$

$M_5 = (A_{11} + A_{12}) \times B_{22};$

$M_6 = (A_{21} - A_{11}) \times (B_{11} + B_{12});$

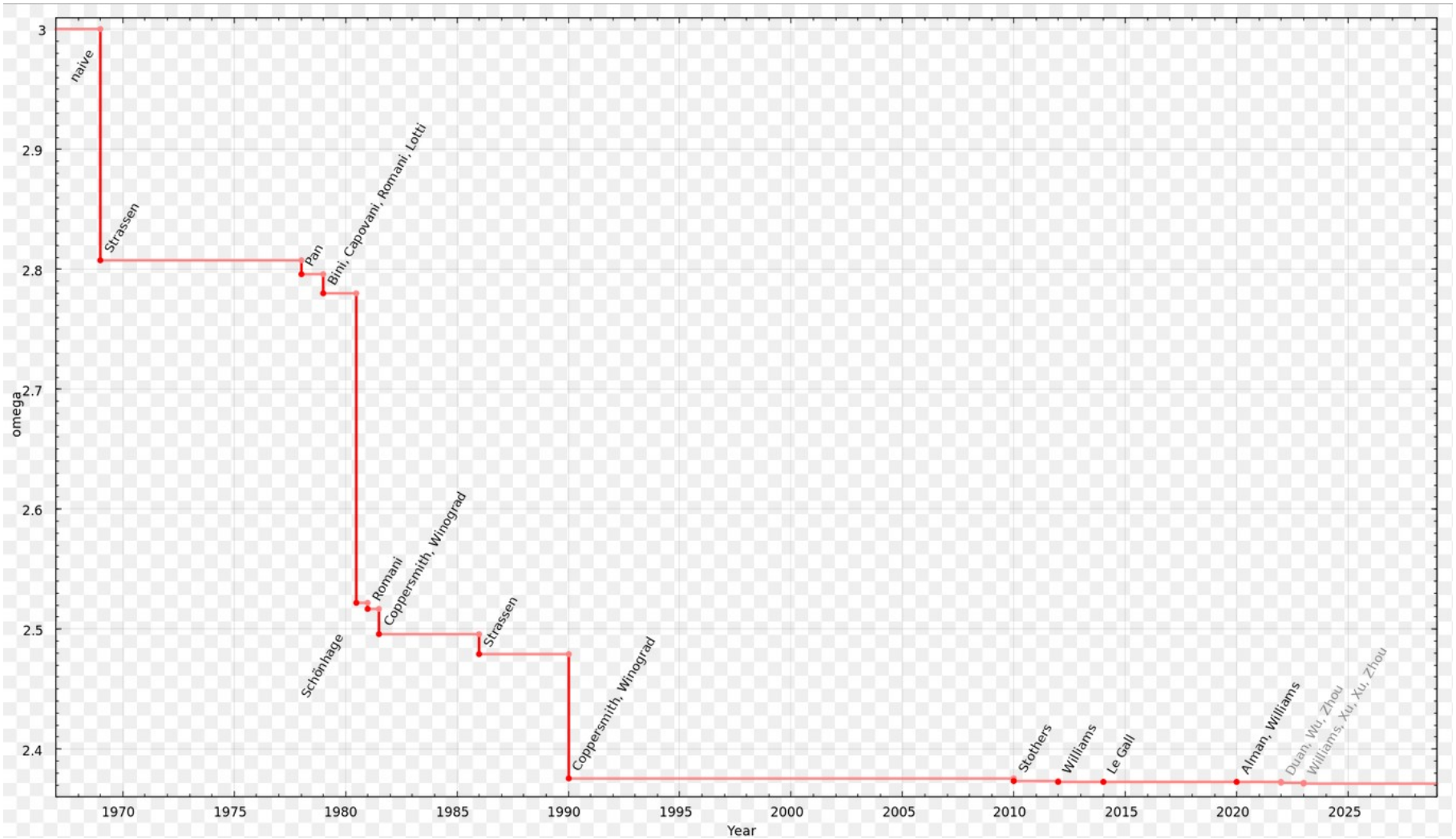
$M_7 = (A_{12} - A_{22}) \times (B_{21} + B_{22}),$

Redukcja mnożeń z 8 do 7. Wtedy:

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{bmatrix}.$$

Asymptotyczna złożoność obliczeniowa: $O(n^{2.807})$, jednakże!

Postęp w redukowaniu złożoności przez lata:



en.wikipedia.org/wiki/Computational_complexityp_of_mathematical_operations#Matrix_algebra

Układy Równań Liniowych

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

Zapisać możemy jako równanie macierzowe:

$$\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b},$$

gdzie postać $\boldsymbol{A}$ oraz $\boldsymbol{b}$ jest znana. Dla uproszczenia zakładamy, że mamy tyle samo równań co niewiadomych ($m = n$).

Przykład rozwiązania równań liniowych (Gauss-Jordan ręcznie)

$$2x + 3y = 5$$

$$4x - y = 2$$

Przykład rozwiązania równań liniowych (Gauss-Jordan ręcznie)

$$2x + 3y = 5$$

$$4x - y = 2$$

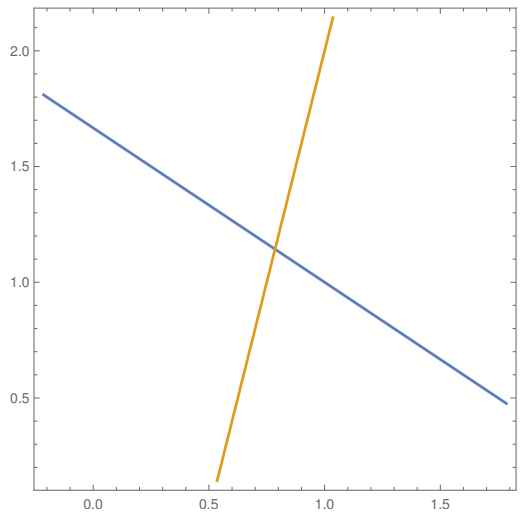
$$\left[\begin{array}{cc|c} 2 & 3 & 5 \\ 4 & -1 & 2 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & \frac{3}{2} & \frac{5}{2} \\ 4 & -1 & 2 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & \frac{3}{2} & \frac{5}{2} \\ 0 & -7 & -8 \end{array}\right]$$

$$\to \left[\begin{array}{cc|c} 1 & \frac{3}{2} & \frac{5}{2} \\ 0 & 1 & \frac{8}{7} \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 0 & \frac{11}{14} \\ 0 & 1 & \frac{8}{7} \end{array}\right]$$

Przykład rozwiązania równań liniowych (Gauss-Jordan ręcznie)

$$2x + 3y = 5$$

$$4x - y = 2$$



$$\left[\begin{array}{cc|c} 2 & 3 & 5 \\ 4 & -1 & 2 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & \frac{3}{2} & \frac{5}{2} \\ 4 & -1 & 2 \end{array}\right] \to \left[\begin{array}{cc|c} 1 & \frac{3}{2} & \frac{5}{2} \\ 0 & -7 & -8 \end{array}\right]$$

$$\to \left[\begin{array}{cc|c} 1 & \frac{3}{2} & \frac{5}{2} \\ 0 & 1 & \frac{8}{7} \end{array}\right] \to \left[\begin{array}{cc|c} 1 & 0 & \frac{11}{14} \\ 0 & 1 & \frac{8}{7} \end{array}\right]$$

Przykład Rozkładu Gaussa-Jordana do uzyskania $\boldsymbol{A}^{-1}$

$\boldsymbol{A}|\mathbb{1} =$

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & -3 & 5 & 0 & 0 & 1 \end{array}\right)$$

Przykład Rozkładu Gaussa-Jordana do uzyskania $\boldsymbol{A}^{-1}$

$\boldsymbol{A}|\mathbb{1} =$

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & -3 & 5 & 0 & 0 & 1 \end{array}\right) \Rightarrow \begin{array}{l} r2 = r2 - \frac{1}{2}r1 \\ r3 = r3 - \frac{4}{2}r1 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & -5 & -1 & -2 & 0 & 1 \end{array}\right)$$

Przykład Rozkładu Gaussa-Jordana do uzyskania $\boldsymbol{A}^{-1}$

$\boldsymbol{A}|\mathbb{1} =$

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & -3 & 5 & 0 & 0 & 1 \end{array}\right) \Rightarrow \begin{array}{l} r2 = r2 - \frac{1}{2}r1 \\ r3 = r3 - \frac{4}{2}r1 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & -5 & -1 & -2 & 0 & 1 \end{array}\right)$$

$$\Rightarrow \begin{array}{l} r1 = r1 - \frac{1}{-\frac{1}{2}}r2 \\ r3 = r3 - \frac{-5}{\frac{-1}{2}}r2 \end{array}$$

Przykład Rozkładu Gaussa-Jordana do uzyskania $\boldsymbol{A}^{-1}$

$$\boldsymbol{A}|\mathbb{1} =$$

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & -3 & 5 & 0 & 0 & 1 \end{array}\right) \Rightarrow \begin{array}{l} r2 = r2 - \frac{1}{2}r1 \\ r3 = r3 - \frac{4}{2}r1 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & -5 & -1 & -2 & 0 & 1 \end{array}\right)$$

$$\Rightarrow \begin{array}{l} r1 = r1 - \frac{1}{-\frac{1}{2}}r2 \\ r3 = r3 - \frac{-5}{\frac{-1}{2}}r2 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 0 & 2 & 0 & 2 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 4 & 3 & -10 & 1 \end{array}\right)$$

Przykład Rozkładu Gaussa-Jordana do uzyskania $\boldsymbol{A}^{-1}$

$\boldsymbol{A}|\mathbb{1} =$

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & -3 & 5 & 0 & 0 & 1 \end{array}\right) \Rightarrow \begin{array}{l} r2 = r2 - \frac{1}{2}r1 \\ r3 = r3 - \frac{4}{2}r1 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 1 & 3 & 1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & -5 & -1 & -2 & 0 & 1 \end{array}\right)$$

$$\Rightarrow \begin{array}{l} r1 = r1 - \frac{1}{-\frac{1}{2}}r2 \\ r3 = r3 - \frac{-5}{\frac{-1}{2}}r2 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 0 & 2 & 0 & 2 & 0 \\ 0 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 4 & 3 & -10 & 1 \end{array}\right)$$

$$\Rightarrow \begin{array}{l} r1 = r1 - \frac{2}{4}r2 \\ r2 = r2 - \frac{-\frac{1}{2}}{4}r2 \end{array} \Rightarrow \left(\begin{array}{ccc|ccc} 2 & 0 & 0 & -\frac{3}{2} & 7 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 & -\frac{1}{8} & -\frac{1}{4} & \frac{1}{8} \\ 0 & 0 & 4 & 3 & -10 & 1 \end{array}\right)$$

Dla ogólnego przypadku

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

Potrzebujemy jakiegoś wzoru do rozwiązania macierzy trójkątnej i wektora $\mathbf{b}'$, które dostaniemy, na przykład 4x4:

$$
\begin{bmatrix}
a'_{11} & a'_{12} & a'_{13} & a'_{14} \\
0 & a'_{22} & a'_{23} & a'_{24} \\
0 & 0 & a'_{33} & a'_{34} \\
0 & 0 & 0 & a'_{44}
\end{bmatrix}
\cdot
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4
\end{bmatrix}
=
\begin{bmatrix}
b'_1 \\ b'_2 \\ b'_3 \\ b'_4
\end{bmatrix}
$$

Metoda Gaussa-Jordana dla $\mathbf{Ax} = \mathbf{b}$

$$\begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a'_{33} & a'_{34} \\ 0 & 0 & 0 & a'_{44} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Jeden element już mamy, a kolejny dostajemy odstawiając do tyłu (backsubstitution):

$$x_4 = b'_4/a'_{44} \quad x_3 = \frac{1}{a'_{33}}[b'_3 - x_4 a'_{34}]$$

Metoda Gaussa-Jordana dla $\mathbf{Ax} = \mathbf{b}$

$$\begin{bmatrix} a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a'_{33} & a'_{34} \\ 0 & 0 & 0 & a'_{44} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Jeden element już mamy, a kolejny dostajemy odstawiając do tyłu (backsubstitution):

$$x_4 = b'_4/a'_{44} \quad x_3 = \frac{1}{a'_{33}}[b'_3 - x_4 a'_{34}]$$

Ogólny wzór:

$$x_i = \frac{1}{a'_{ii}}\left[b'_i - \sum_{j=i+1}^{N} a'_{ij} x_j\right]$$

Metoda Gaussa-Seidla dla $\mathbf{Ax} = \mathbf{b}$

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{L_*} + \underbrace{\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{U}.$$

$$L_* \mathbf{x}^{(k+1)} = \mathbf{b} - U\mathbf{x}^{(k)}$$

Metoda Gaussa-Seidla dla $\mathbf{Ax} = \mathbf{b}$

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}}_{L_*} + \underbrace{\begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{U}.$$

$$L_*\mathbf{x}^{(k+1)} = \mathbf{b} - U\mathbf{x}^{(k)}$$

Iteracyjnie liczymy $x_i^{(k+1)}$ przez podstawianie do przodu

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right), \quad i = 1, 2, \ldots, n.$$

https://pl.wikipedia.org/wiki/Metoda_Gaussa-Seidla#Uk%C5%82ad_trzech_r%C3%B3wna%C5%84_liniowych

Rozkład LU dla macierzy $\mathbf{A}$ z $\mathbf{Ax} = \mathbf{b}$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

Rozkład LU dla macierzy $\mathbf{A}$ z $\mathbf{Ax} = \mathbf{b}$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m
\end{aligned}
$$

Rozwiązanie za pomocą rozkładu LU:

1. Rozkład macierzy $\boldsymbol{A}$ na iloczyn dwóch macierzy trójkątnych:

   $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{U}$

2. Rozwiązanie dwóch układów równań z macierzami trójkątnymi:

   $\boldsymbol{L} \cdot \boldsymbol{y} = \boldsymbol{b}$

   $\boldsymbol{U} \cdot \boldsymbol{x} = \boldsymbol{y}$

Wzory dla podstawiania do przodu i do tyłu

Rozwiązanie za pomocą rozkładu LU:

1. $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{U}$
2. $\boldsymbol{L} \cdot \boldsymbol{y} = \boldsymbol{b}$, $\boldsymbol{U} \cdot \boldsymbol{x} = \boldsymbol{y}$

Wzory dla podstawiania do przodu i do tyłu

Rozwiązanie za pomocą rozkładu LU:

1. $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{U}$
2. $\boldsymbol{L} \cdot \boldsymbol{y} = \boldsymbol{b}$, $\boldsymbol{U} \cdot \boldsymbol{x} = \boldsymbol{y}$

Podstawianie do przodu (forward substitution):

$$\begin{cases} y_1 = \frac{b_1}{l_{11}} \\ y_i = \frac{1}{l_{ii}} \left[b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \right] \end{cases} \quad i = 2, 3, \ldots, n$$

Wzory dla podstawiania do przodu i do tyłu

Rozwiązanie za pomocą rozkładu LU:

1. $\boldsymbol{A} = \boldsymbol{L} \cdot \boldsymbol{U}$
2. $\boldsymbol{L} \cdot \boldsymbol{y} = \boldsymbol{b}$, $\boldsymbol{U} \cdot \boldsymbol{x} = \boldsymbol{y}$

Podstawianie do przodu (forward substitution):

$$\begin{cases} y_1 = \frac{b_1}{l_{11}} \\ y_i = \frac{1}{l_{ii}} \left[b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \right] \end{cases} \quad i = 2, 3, \ldots, n$$

Podstawianie do tyłu (backsubstitution):

$$\begin{cases} x_n = \frac{y_n}{u_{nn}} \\ x_i = \frac{1}{u_{ii}} \left[y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right], \end{cases} \quad i = n-1, n-2, \ldots, 1$$

Rozkład LU

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \mathbf{L} \cdot \mathbf{U}?$$

Jak to rozdzielić?

`https://pl.wikipedia.org/wiki/Metoda_LU`

Rozkład LU metodą Doolitlle

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Rozkład LU metodą Doolitlle

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Wzory, które pomogą nam znaleźć elementy macierzy $\mathbf{U}$ i $\mathbf{L}$:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \text{ dla } j \in \{i, i+1, \ldots, n\},$$

$$l_{ji} = \frac{1}{u_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right) \text{dla } j \in \{i+1, i+2, \ldots, n\}.$$

Rozkład LU metodą Doolitlle

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Wzory, które pomogą nam znaleźć elementy macierzy $\mathbf{U}$ i $\mathbf{L}$:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \text{ dla } j \in \{i, i+1, \ldots, n\},$$

$$l_{ji} = \frac{1}{u_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right) \text{dla } j \in \{i+1, i+2, \ldots, n\}.$$

Gdzie naprzemiennie wyznaczamy tymi wzorami wiersz $\mathbf{U}$ a potem wiersz $\mathbf{L}$ i tak dalej (układ n równań n^2 niewiadomych).

Przykłady 3x3 `https://pl.wikipedia.org/wiki/Metoda_LU#Przyk%C5%82ad_(macierz_3x3)`

Rozkład LU metodą Crouta(bliźniacza do Doolitlle)

Przytaczam dla kompletności

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

https://pl.wikipedia.org/wiki/Metoda_LU

Wyznacznik macierzy

Znany wzór permutacyjny [numerycznie niezbyt wydajny]:

$$\mathbf{det}(\mathbf{A}) = \sum_{\sigma \in S_n} (-1)^{\mathrm{Inv}(\sigma)} a_{1\sigma(1)} \cdot a_{2\sigma(2)} \cdot \ldots \cdot a_{n\sigma(n)}$$

Jak uzyskać wyznacznik macierzy za pomocą rozkładu LU:

1. Rozkładamy $\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$.
2. $\mathbf{det}(\mathbf{A}) = \mathbf{det}(\mathbf{L} \cdot \mathbf{U}) = \mathbf{det}(\mathbf{L})\mathbf{det}(\mathbf{U})$
3. Jeśli Doolittle: $\mathbf{det}(\mathbf{A}) = \prod_j^N u_{jj}$,
   jeśli Crout: $\mathbf{det}(\mathbf{A}) = \prod_j^N l_{jj}$

Jak rozwiązać równanie gdy prawa strona $(\mathbf{b} + i\mathbf{d})$ jest zespolona?

$$\mathbf{Ax} = (\mathbf{b} + i\mathbf{d})$$

1. Rozdzielamy $\mathbf{A} = \mathbf{LU}$
2. Podstawiamy wstecz (backsubstitution) $\mathbf{b}$ aby otrzymać część rzeczywistą rozwiązania
3. Podstawiamy wstecz (backsubstitution) $\mathbf{d}$ aby otrzymać część urojoną

Równania zespolone

Jeżeli można rozdzielić macierz oraz wektor z prawej strony na część rzeczywistą i urojoną:

$$(\mathbf{A} + \mathbf{iC})(\mathbf{x} + \mathbf{iy}) = (\mathbf{b} + \mathbf{id})$$

Rozdzielamy to równanie na układ dwóch równań:

$$\begin{cases} \mathbf{Ax} - \mathbf{Cy} + i(\mathbf{Cx} + \mathbf{Ay}) & = \mathbf{b} + i\mathbf{d} \\ \mathbf{Cx} + \mathbf{Ay} + i(\mathbf{Ax} - \mathbf{Cy}) & = \mathbf{d} - i\mathbf{b} \end{cases}$$

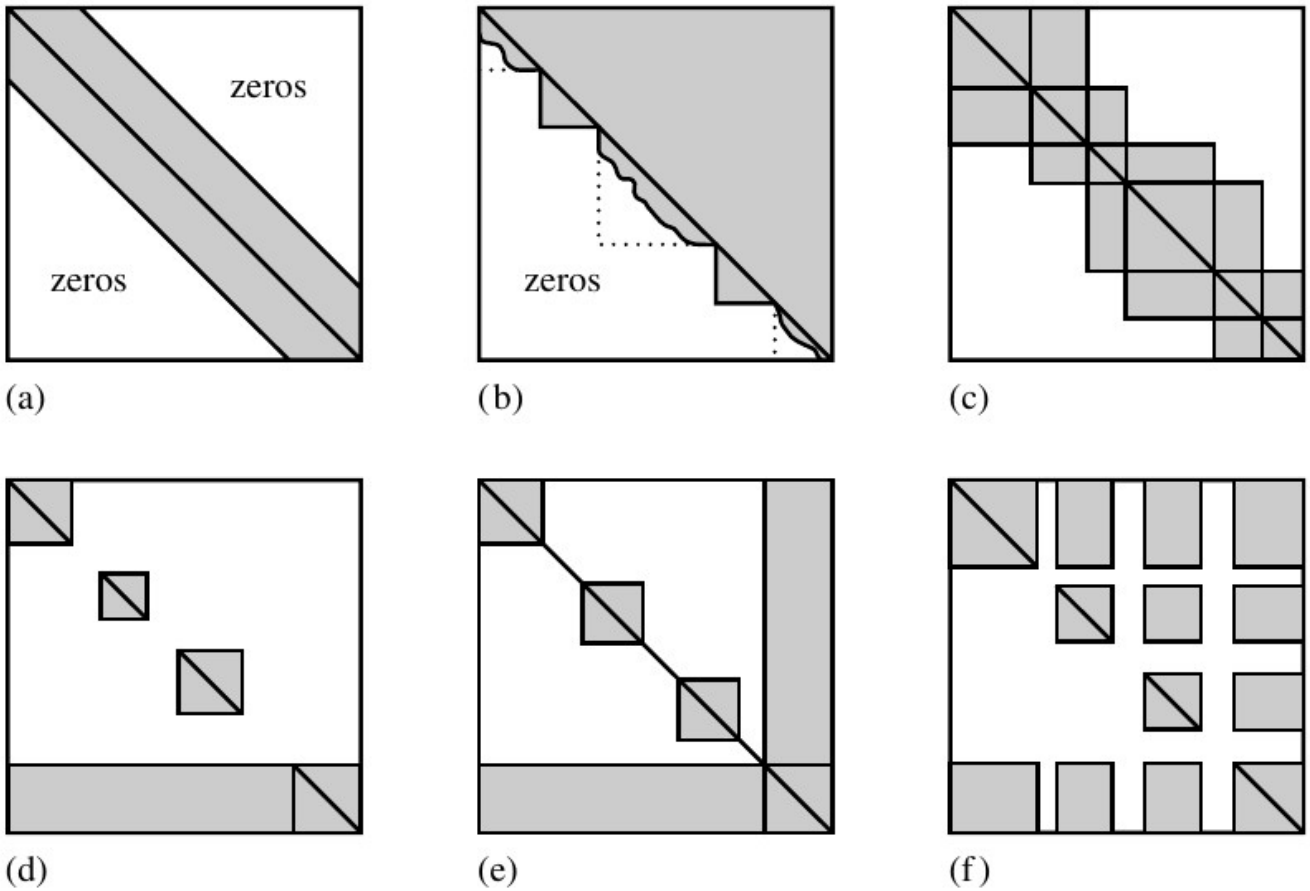
Odpowiadające mu równanie macierzowe:

$$\begin{bmatrix} \mathbf{A} & -\mathbf{C} \\ \mathbf{C} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}$$

Inne rozkłady

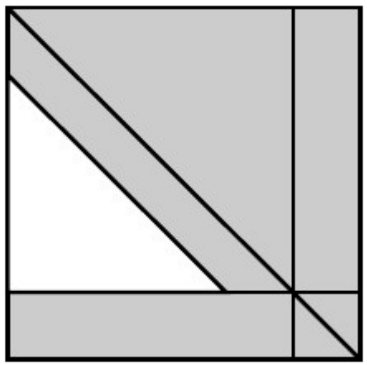
- Metoda SVD: Może być aplikowana do macierzy prostokątnych i macierzy osobliwych ($\mathbf{det}(\mathbf{A}) = 0$)
- Metoda $\mathbf{A} = \mathbf{QR}$: Używając procesu ortogonalizacji Grahama-Schmidta (albo metody Hausholdera)

Rzadkie macierze (1)

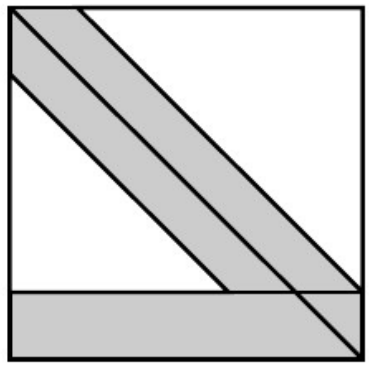


(a) Band diagonal; (b) block triangular; (c) block tridiagonal; (d) singly bordered block diagonal; (e) doubly bordered block diagonal; (f) singly bordered block triangular

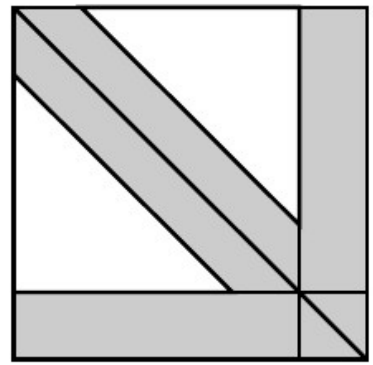
Rzadkie macierze (2)



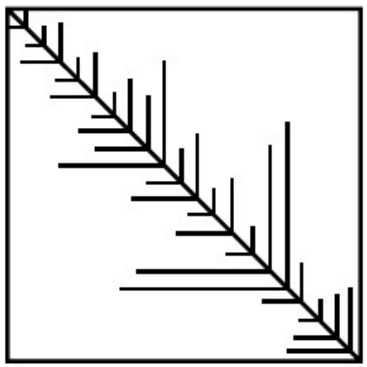
(g)



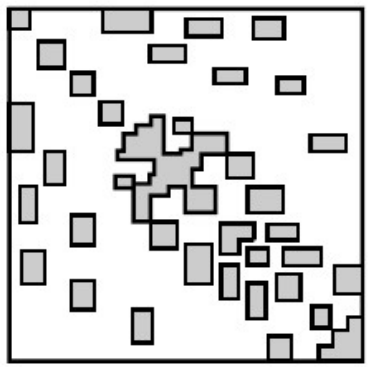
(h)



(i)



(j)



(k)

(g) bordered band-triangular; (h) and (i) singly and doubly bordered band diagonal; (j) and (k) other!

(after Tewarson)

Podstawowe biblioteki

Fortran/C++:

- **LAPACK** (Linear Algebra Package)
- **BLAS** (Basic Linear Algebra Subprograms)
- **Eigen** (C++) (łatwa w obsłudze ale ograniczona)
- **ARPACK** (do metody Arnoldi macierzy rzadkich)

Python:

- **SciPy**: (LAPACK i BLAS)
- **NumPy**

Do obliczeń tensorowych lista bibliotek:

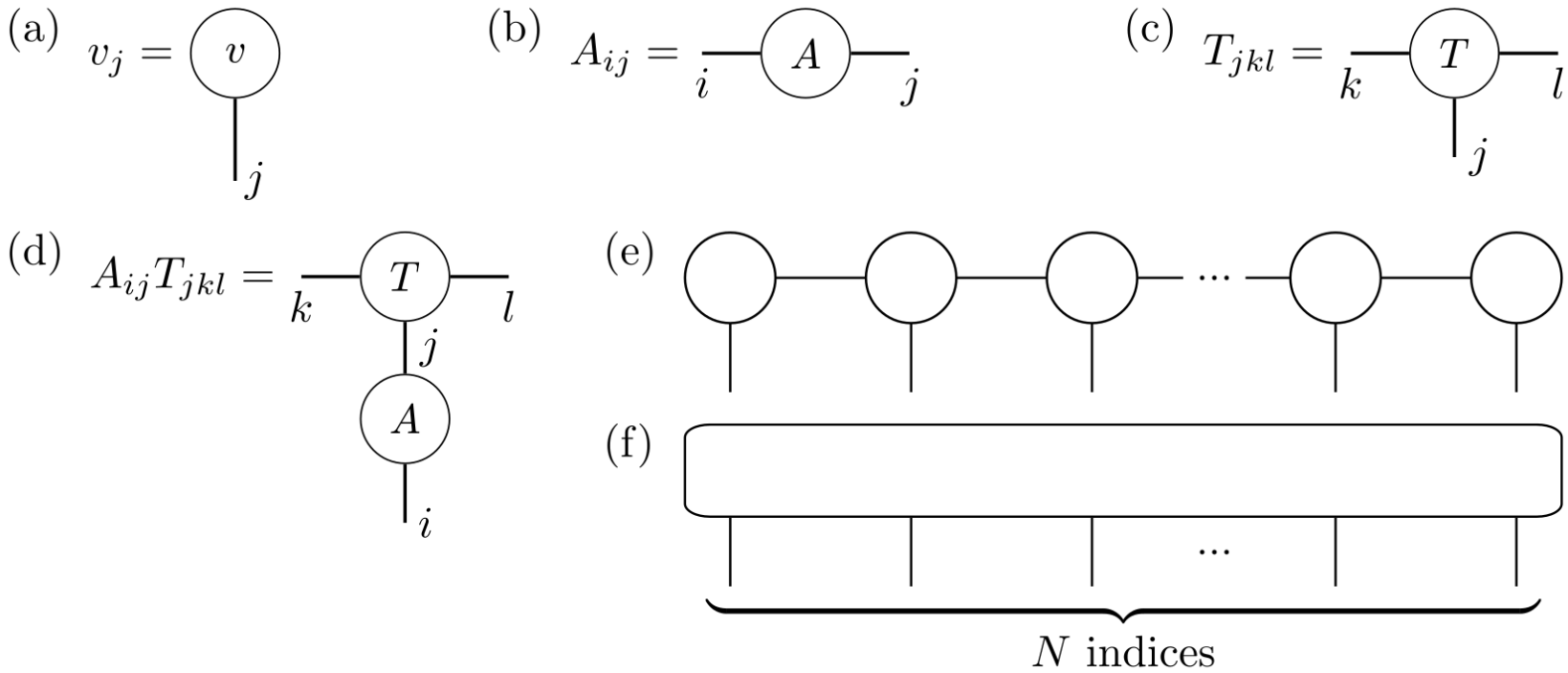
`http://tensornetwork.org/software/`

Źródła

- *Numerical Recipes 3rd Edition: The Art of Scientific Computing* 3rd Edition
- `en.wikipedia.org/wiki/Numerical_linear_algebra` i strony pochodne

Dziękuję za uwagę

Tensor diagrams

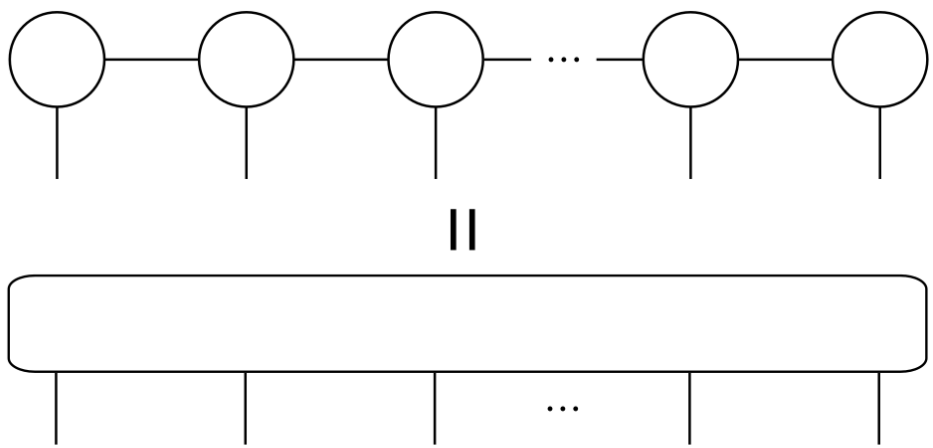


Matrix product state (Tensor train)

We can express tensor $T_{s_1 s_2 \cdots s_L}$ as a Matrix product state (MPS):

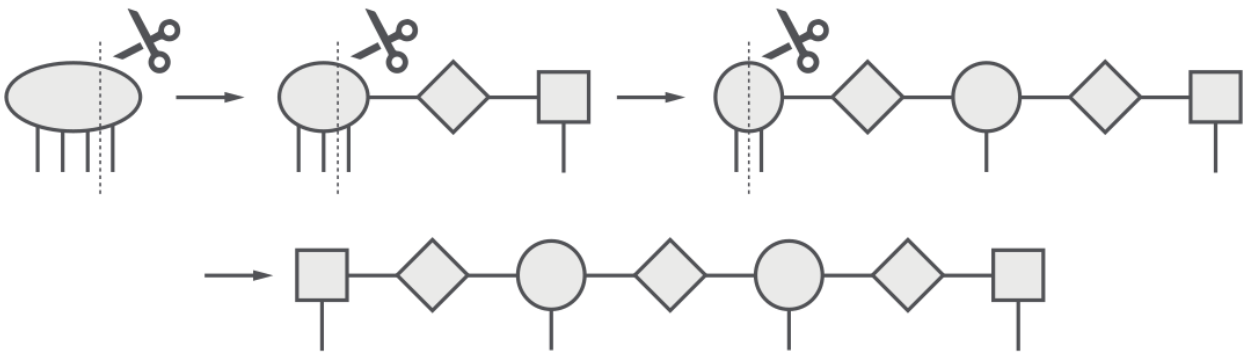
$$T_{s_1 s_2 \cdots s_L} = \sum_{\alpha} A^{s_1}_{\alpha_1} A^{s_2}_{\alpha_1,\alpha_2} A^{s_3}_{\alpha_2,\alpha_3} \cdots A^{s_{L-1}}_{\alpha_{L-1},\alpha_L} A^{s_L}_{\alpha_L}$$

where $A^{s_i}_{\alpha_i,\alpha_{i+1}}$ are matrices, s_i are outer indices, α_i indices are contracted/summed over to recover tensor $T_{s_1 s_2 \cdots s_L}$.

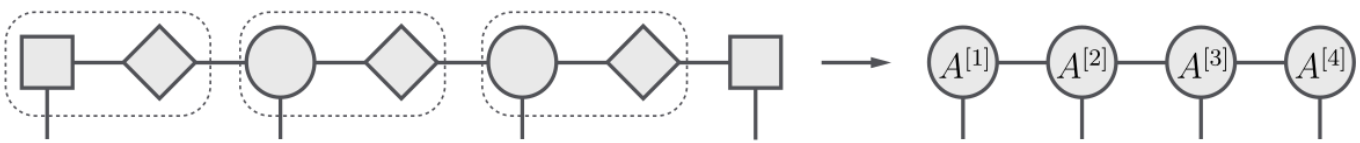


Construction of MPS via SVD

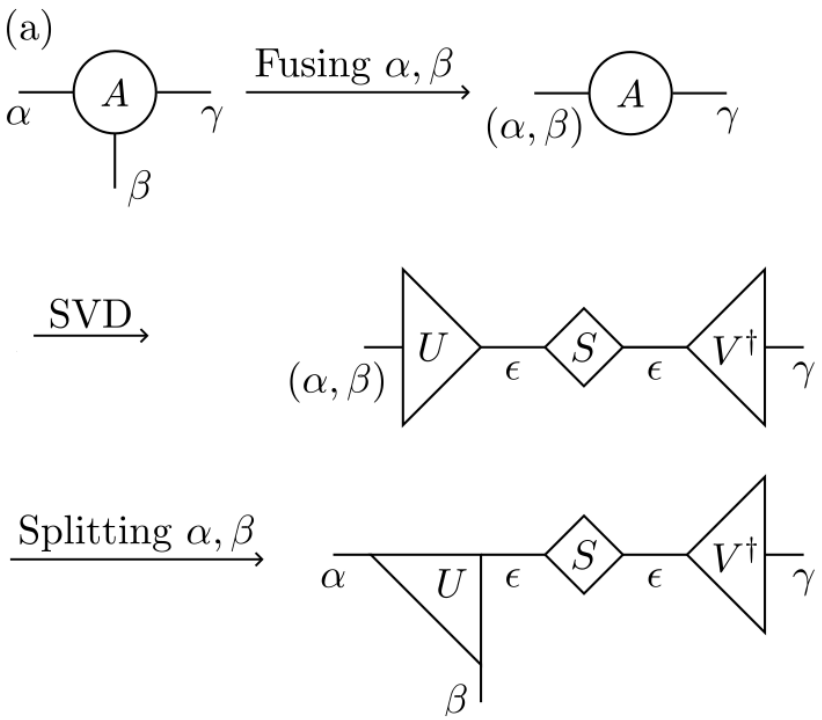
Arbitrary four-party state $|\psi\rangle$ represented by tensor is bipartited into two parts, where one leg undergoes SVD, process is repeated for entire tensor.

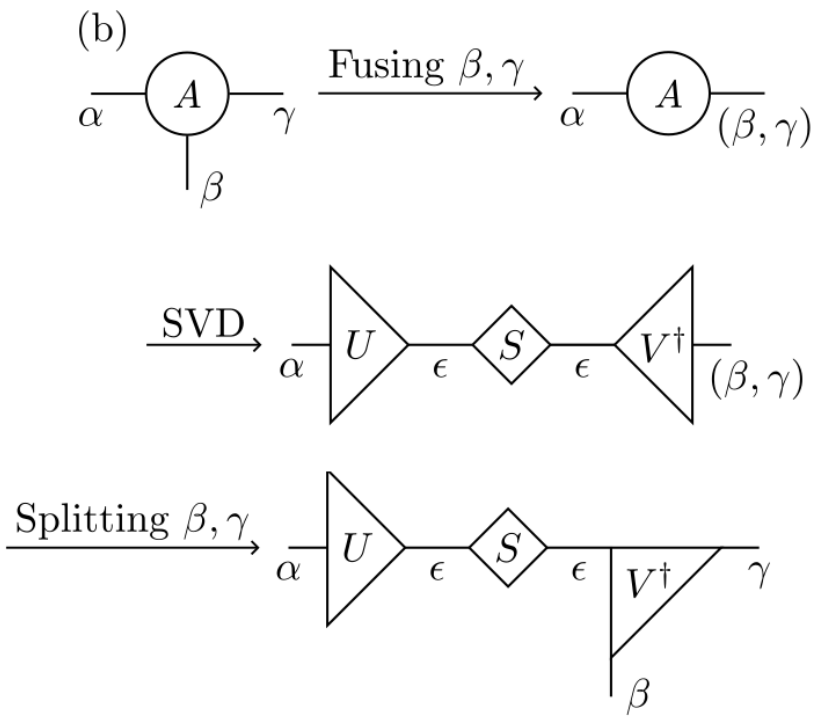


Then tensors can be grouped into an MPS.



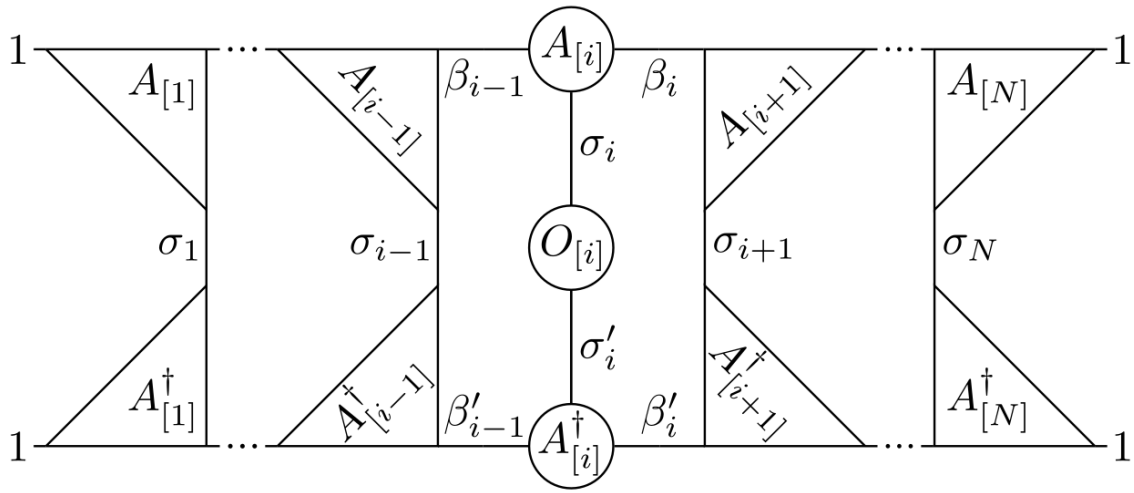
Singular value decomposition (SVD)



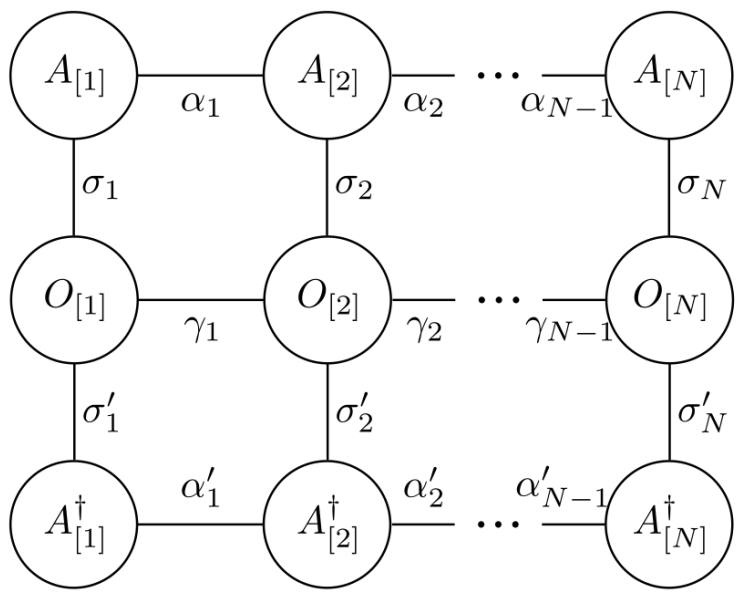


Matrix product operator (MPO)

$\langle\psi|\hat{O}_{[i]}|\psi\rangle =$



$\langle\psi|\hat{O}|\psi\rangle =$

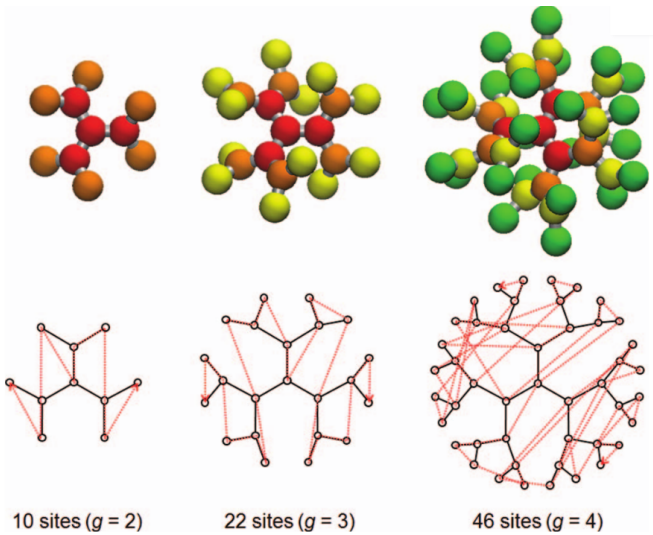


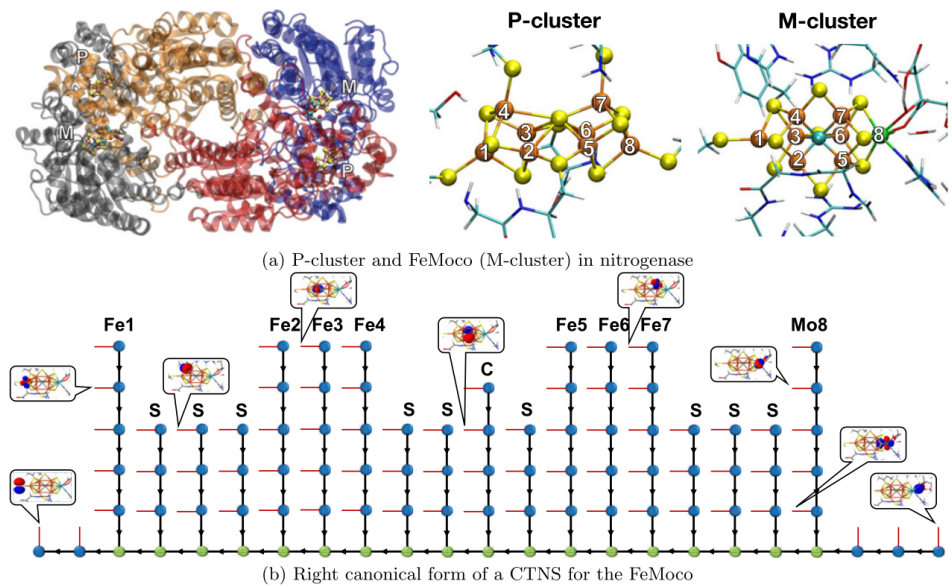
Hardware for computing TN methods

The cost of DMRG scales with system size N as $O(ND^3)$, where the so-called bond dimension D regulates how expressive the underlying matrix product state (MPS) variational ansatz is. We consider lattice models in two spatial dimensions, with square lattices of size 10×10 (free fermions) and 20×20 (transverse field Ising model), for which the required MPS bond dimension is known to scale at least as $\exp(\sqrt{N})$. Using half of a TPU v3 pod (namely 1024 TPU v3 cores), we reach an unprecedentedly large bond dimension $D = 2^{16} = 65\,536$, for which optimizing a single MPS tensor takes about 2 min.

Martin Ganahl et al. PRX Quantum 4, 010317 (2023)

TN in quantum chemistry:
Tree Tensor Network (TTN),
Comb Tensor Network States(CTNS)





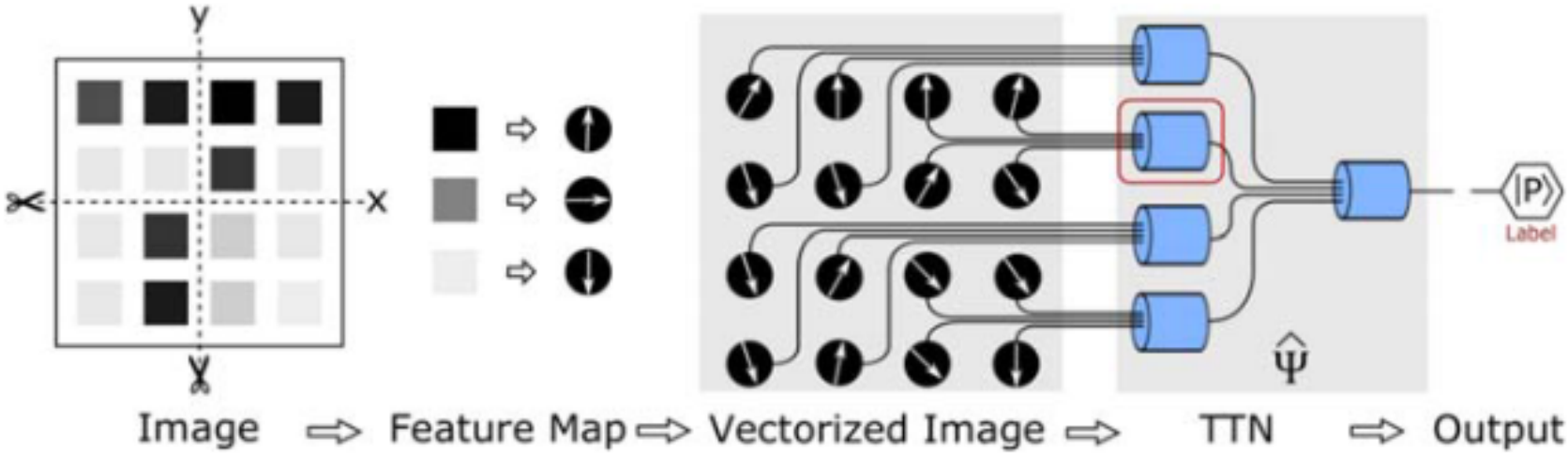
(a) P-cluster and FeMoco (M-cluster) in nitrogenase

(b) Right canonical form of a CTNS for the FeMoco

[left] N. Nakatani; G. Kin-Lic Chan J. Chem. Phys. 138, 134113 (2013)

[right] Zhendong Li 2021 Electron. Struct. 3 014001 (2021)

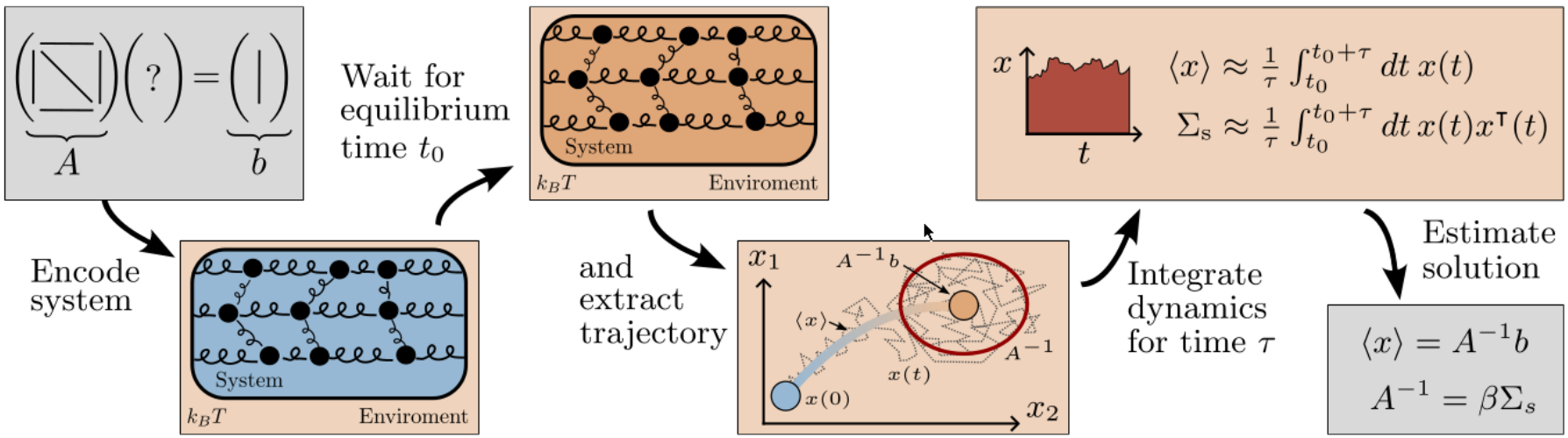
TN in classical and quantum machine learning



Ding Liu et al 2019 New J. Phys. 21 073059 (2021)

https://tensornetwork.org/ml/

Example of a thermodynamic system solving linear algebra problem



Linear System Protocol

1. Given a linear system $Ax = b$, set the potential of the device to

$$V(x) = \frac{1}{2} x^\intercal A x - b^\intercal x \tag{7}$$

at time $t = 0$.

2. Choose equilibration tolerance parameters $\varepsilon_{\mu 0}, \varepsilon_{\Sigma 0} \in \mathbb{R}^+$, and choose the equilibration time

$$t_0 \geqslant \widehat{t}_0, \tag{8}$$

where $\widehat{t}_0$ is computed from the system's physical properties or using heuristic methods based on Eqs. (12), (14). Allow the system to evolve under its dynamics until $t = t_0$, which ensures that $\left\| \langle x \rangle - A^{-1} b \right\| / \| A^{-1} b \| \leqslant \varepsilon_{\mu 0}$ and $\left\| \Sigma - \beta^{-1} A^{-1} \right\| / \| \beta^{-1} A^{-1} \| \leqslant \varepsilon_{\Sigma 0}$.

3. Choose error tolerance parameter ε_x and success probability P_ε, and choose the integration time

$$\tau \geqslant \widehat{\tau}, \tag{9}$$

where $\widehat{\tau}$ is computed from the system's physical properties, Eq. (12) or (14). Use an analog integrator to measure the the time average

$$\bar{x} = \frac{1}{\tau} \int_{t_0}^{t_0 + \tau} dt\, x(t), \tag{10}$$

which satisfies $\| A\bar{x} - b \| / \| b \| \leqslant \varepsilon_x$ with probability at least P_δ.

https://arxiv.org/abs/2308.05660 (2023)

Example of a thermodynamic system solving linear algebra problem

Problem	Digital SOTA	This work (Overdamped)	This work (Underdamped)
Linear System	$O(\min\{d^\omega, d^2\sqrt{\kappa}\})$	$O(d\kappa^2\varepsilon^{-2})$	$O(d\sqrt{\kappa}\varepsilon^{-2})$
Matrix Inverse	$O(d^\omega)$	$O(d^2\kappa\varepsilon^{-2})$	$O(d^2\kappa\varepsilon^{-2})$
Lyapunov Equation	$O(d^3)$	$O(d^2\kappa\varepsilon^{-2})$	$O(d^2\kappa\varepsilon^{-2})$
Matrix Determinant	$O(d^3)$	$O(d\kappa\ln(\kappa)^3\varepsilon^{-2})$	$O(d\ln(\kappa)^3\varepsilon^{-2})$

TABLE I. **Comparison of asymptotic complexities of linear algebra algorithms.** Here, d is the matrix dimension, κ is the condition number, and ϵ is the error. For our thermodynamic algorithms, the complexity depends on the dynamical regime, i.e., whether the dynamics are overdamped and underdamped. For the digital SOTA case, the complexity of solving symmetric, positive definite linear systems, matrix inverse, Lyapunov equation, and matrix determinant problems are respectively for algorithms based on: conjugate gradient method [28], fast matrix multiplication/inverse [29], Bartels-Stewart algorithm [30], and Cholesky decomposition [31]. $\omega \approx 2.3$ denotes the matrix multiplication constant.

Example of a thermodynamic system solving linear algebra problem

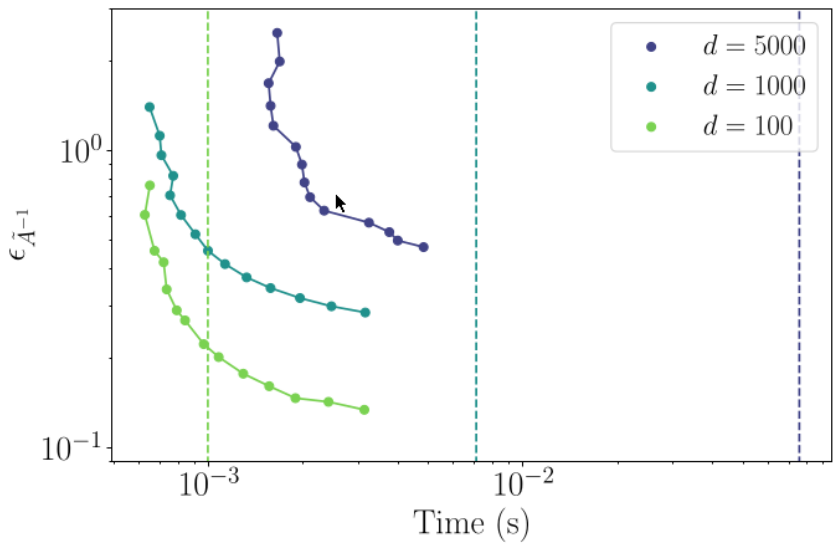


FIG. 6. **Comparison of the error $\varepsilon_{\tilde{A}^{-1}}$ of the thermodynamic algorithm (TA) to solve linear systems with the Cholesky decomposition as a function of total runtime.** Dimensions are $d = 100, 1000, 5000$, respectively in light green, light blue, and purple are shown, as well as the corresponding Cholesky decomposition times as dashed lines. Here the condition numbers are respectively $\{120, 1189, 5995\}$. Calculations were performed on an Nvidia Tesla A10 GPU.