

Wprowadzenie do matlaba

3 października 2020

Co to jest matlab?

MATLAB (MATrix LABoratory) – pakiet programowy do wykonywania złożonych obliczeń numerycznych i wizualizacji wyników. Interaktywne środowisko oferujące setki wbudowanych funkcji obliczeniowych, graficznych i animacyjnych. Użytkownik może sam rozbudowywać zakres zastosowań poprzez tworzenie własnych skryptów i programów.

<https://www.mathworks.com/>

Po co nam to? ;)

Products and Services

Search MathWorks.com

Products by Category | Alphabetical

MATLAB[®]

PRODUCT FAMILY

MATLAB

Parallel Computing

- Parallel Computing Toolbox
- MATLAB Parallel Server

Math and Optimization

- Curve Fitting Toolbox
- Optimization Toolbox
- Global Optimization Toolbox
- Symbolic Math Toolbox
- Mapping Toolbox
- Partial Differential Equation Toolbox

AI, Data Science, and Statistics

- Statistics and Machine Learning Toolbox
- Deep Learning Toolbox
- Reinforcement Learning Toolbox
- Deep Learning HDL Toolbox
- Text Analytics Toolbox
- Predictive Maintenance Toolbox

Code Generation

- MATLAB Coder
- Embedded Coder
- HDL Coder
- HDL Verifier
- Filter Design HDL Coder

SIMULINK[®]

PRODUCT FAMILY

Simulink

- System Composer

Event-Based Modeling

- Stateflow
- SimEvents

Physical Modeling

- Simscape
- Simscape Driveline
- Simscape Electrical
- Simscape Fluids
- Simscape Multibody

Real-Time Simulation and Testing

- Simulink Real-Time
- Simulink Desktop Real-Time

Code Generation

- Simulink Coder
- Embedded Coder
- AUTOSAR Blockset
- Fixed-Point Designer
- Simulink PLC Coder
- Simulink Code Inspector
- DO Qualification Kit (for DO-178)
- IEC Certification Kit (for ISO 26262 and IEC 61508)
- HDL Coder
- HDL Verifier

SERVICES

- Software Maintenance
- Training
- Consulting

LICENSE TYPES

- Industry Use
- Student Use
- University Use
- Startup Use
- Primary and Secondary School Use
- Home Use

CLOUD SOLUTIONS

- MATLAB Online
- Simulink Online
- MATLAB Drive
- ThingSpeak
- MATLAB Mobile
- MATLAB Grader
- MATLAB on AWS and Azure

COMMUNITY AND THIRD-PARTY

- File Exchange
- Hardware Support Packages & Services
- Third-Party Products & Services

Trial software

Contact sales

Wprowadzenie do matlaba

3 października 2020

3 / 46

Po co nam to? ;)

Products and Services

Products by Category

Alphabetical

Trial software

Contact sales

APPLICATIONS

Signal Processing

- Signal Processing Toolbox
- Phased Array System Toolbox
- DSP System Toolbox
- Audio Toolbox
- Wavelet Toolbox

Image Processing and Computer Vision

- Image Processing Toolbox
- Computer Vision Toolbox
- Lidar Toolbox

Control Systems

- Control System Toolbox
- System Identification Toolbox
- Predictive Maintenance Toolbox
- Robust Control Toolbox
- Model Predictive Control Toolbox
- Fuzzy Logic Toolbox
- Simulink Control Design
- Simulink Design Optimization
- Reinforcement Learning Toolbox
- Motor Control Blockset

Test and Measurement

- Data Acquisition Toolbox
- Instrument Control Toolbox
- Image Acquisition Toolbox
- OPC Toolbox
- Vehicle Network Toolbox
- ThingSpeak

RF and Mixed Signal

- Antenna Toolbox
- RF Toolbox
- RF Blockset
- Mixed-Signal Blockset
- SerDes Toolbox

Wireless Communications

- Communications Toolbox
- WLAN Toolbox
- LTE Toolbox
- 5G Toolbox

Autonomous Systems

- Automated Driving Toolbox
- Robotics System Toolbox
- UAV Toolbox
- Navigation Toolbox
- ROS Toolbox
- Sensor Fusion and Tracking Toolbox
- RoadRunner
- RoadRunner Asset Library
- RoadRunner Scene Builder

FPGA, ASIC, and SoC Development

- HDL Coder
- HDL Verifier
- Deep Learning HDL Toolbox
- Wireless HDL Toolbox
- Vision HDL Toolbox
- Filter Design HDL Coder

Automotive

- Model-Based Calibration Toolbox
- Powertrain Blockset
- Vehicle Dynamics Blockset
- Automated Driving Toolbox
- IEC Certification Kit (for ISO 26262 and IEC 61508)
- Vehicle Network Toolbox
- AUTOSAR Blockset
- RoadRunner
- RoadRunner Asset Library
- RoadRunner Scene Builder

Aerospace

- Aerospace Blockset
- Aerospace Toolbox
- UAV Toolbox
- DO Qualification Kit (for DO-178)

Computational Finance

- Econometrics Toolbox
- Financial Toolbox
- Datafeed Toolbox
- Database Toolbox
- Spreadsheet Link (for Microsoft Excel)
- Financial Instruments Toolbox
- Trading Toolbox
- Risk Management Toolbox

Computational Biology

- Bioinformatics Toolbox
- SimBiology

Po co nam to? ;)

CUSTOMERS (PARTIAL LIST)

AEROSPACE AND DEFENSE

Airbus
BAE Systems
Bell Helicopter
Boeing
European Space Agency
Honeywell
Korean Air
Leonardo
Lockheed Martin
NASA
Raytheon
U.S. Air Force
U.S. Navy

AUTOMOTIVE

BMW
Caterpillar
Continental
Daimler
Ford Motor Company
General Motors
Hyundai
Nissan
Scania
Tata Motors
Tesla Motors
Toyota

ENERGY

Gas Natural Fenosa
Horizon Wind Energy
Hydro-Québec
RWE
Sandia National Laboratories

FINANCIAL SERVICES

Bank of England/PRA
CalPERS
Commerzbank
International Monetary Fund
JP Morgan
Munich Re
State Street Global Advisors
Swiss Re
UniCredit Bank Austria AG

INDUSTRIAL AUTOMATION AND MACHINERY

ABB
ASML
Omron
Schlumberger
Siemens
Tetra Pak

MEDICAL DEVICES

Po co nam to? ;)

Toyota

BIOTECH AND PHARMACEUTICAL

Genentech

GlaxoSmithKline

Mitsubishi Tanabe Pharma

Novartis

Pfizer

Roche

COMMUNICATIONS

AT4 wireless

Ericsson

NEC

Nokia

NTT DOCOMO

Vodafone

ELECTRONICS AND SEMICONDUCTORS

Apple

ATT

Intel

LG Electronics

Qualcomm

Realtek Semiconductor
Corporation

Renesas Electronics

Samsung

Texas Instruments

MEDICAL DEVICES

Cochlear

Johnson & Johnson

Philips Healthcare

Sonova

Weinmann Medical Technology

SOFTWARE AND INTERNET

Amazon

Facebook

Google

ACADEMIA

*More than 6500 institutions
globally, including:*

Carnegie Mellon University

Harvard University

Johns Hopkins University

KTH Royal Institute
of Technology

Massachusetts Institute
of Technology

Max Planck Institute

McGill University

Politecnico di Torino

Stanford University

Technische Universität München

Tsinghua University

University of Cambridge

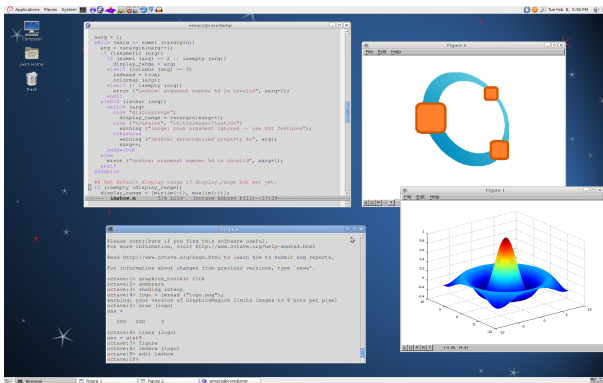
University of Michigan

University of Sydney

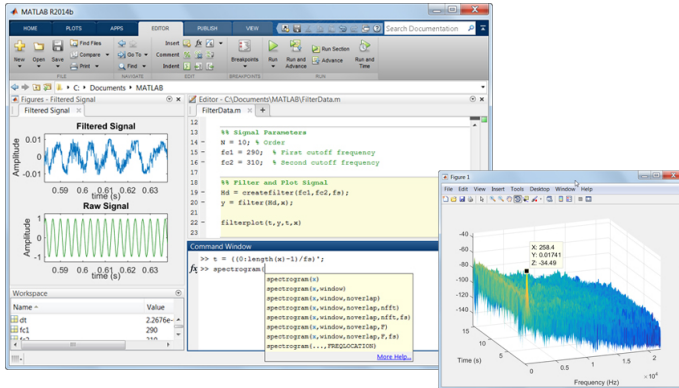
University of Tokyo

GNU Octave – darmowa alternatywa MATLAB-a

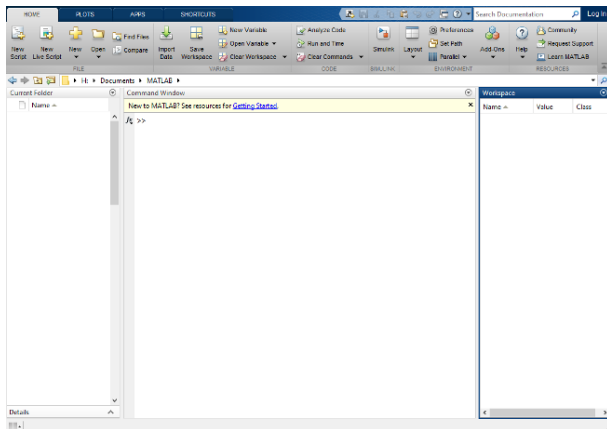
- zdarzają się drobne różnice w składni funkcji – łatwo sprawdzić w dokumentacji



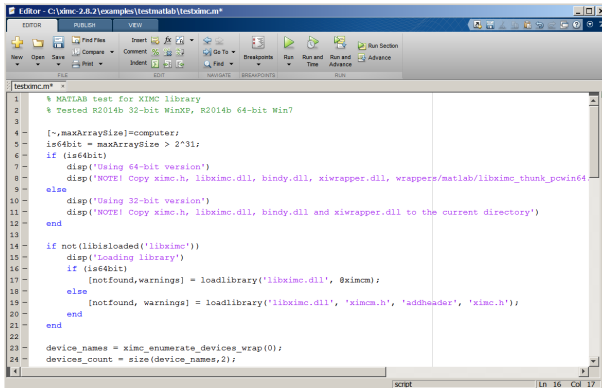
<https://www.gnu.org/software/octave/>



Pulpit w trybie interakcyjnym



Pulpit w trybie edycji



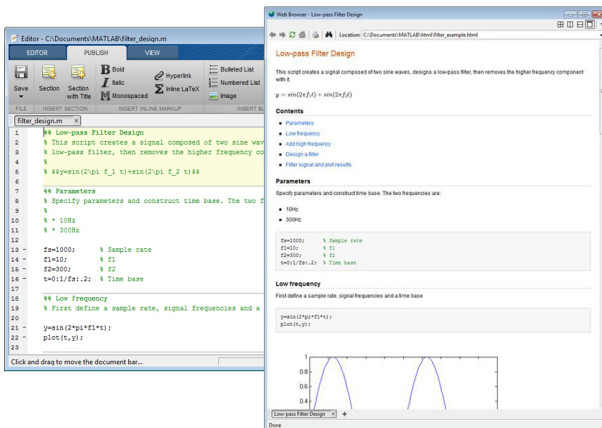
The screenshot shows the MATLAB Editor window with the file `testxmc.m` open. The window has a menu bar (File, Edit, View, Publish) and a toolbar with icons for file operations, editing, and running. The script content is as follows:

```
1 % MATLAB test for XIMC library
2 % Tested R2014b 32-bit WinXP, R2014b 64-bit Win7
3
4 [~,maxArraySize]=computer;
5 is64bit = maxArraySize > 2^31;
6 if (is64bit)
7     disp('Using 64-bit version')
8     disp('NOTE! Copy ximc.h, libximc.dll, bindy.dll, xiwrapper.dll, wrappers/matlab/libximc_thunk_pcwin64
9
10 else
11     disp('Using 32-bit version')
12     disp('NOTE! Copy ximc.h, libximc.dll, bindy.dll and xiwrapper.dll to the current directory')
13 end
14
15 if not(libisloaded('libximc'))
16     disp('Loading library')
17     if (is64bit)
18         [notfound, warnings] = loadlibrary('libximc.dll', @ximcm);
19     else
20         [notfound, warnings] = loadlibrary('libximc.dll', 'ximcm.h', 'addheader', 'ximc.h');
21     end
22 end
23
24 device_names = ximc_enumerate_devices_wrap(0);
25 devices_count = size(device_names,2);
```

The status bar at the bottom indicates the cursor is at line 16, column 17.

Skąd wziąć pomoc?

- Google – mnóstwo dokumentacji on-line
- helpdesk, help, lookfor



Parę uwag ogólnych

- Obsługuje obliczenia interaktywnie – pobiera dane z ekranu i tam je wyświetla.

```
>> 2+2
```

```
ans = 4
```

```
>> 1.5*cos(42*pi/180)
```

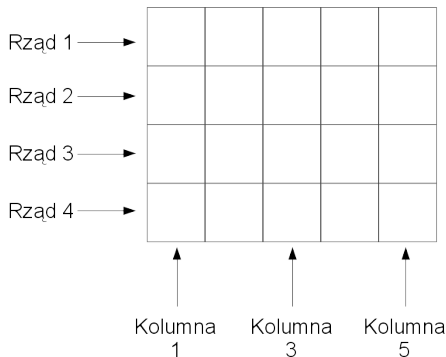
```
ans = 1.1147
```

```
>> sqrt(2)/2
```

```
ans = 0.70711
```

Parę uwag ogólnych

- Podstawowym typem danych jest tablica z danymi różnego typu (np. liczby całkowite, liczby rzeczywiste, łańcuchy znakowe, macierze). Nie trzeba deklarować typu danych.
- Wymiarowanie odbywa się automatycznie.



Parę uwag ogólnych

- Można definiować zmienne o zadanej nazwie.
- Rozróżnia wielkość liter: **a** i **A** to dwie różne nazwy zmiennych.

```
>> x = 2 + 2
```

```
x =
```

```
4
```

```
>> X = 118/x
```

```
X =
```

```
29.5000
```

```
>> x * X
```

```
ans =
```

```
118
```

```
>> promien = 2.8
```

```
promien =
```

```
2.8000
```

```
>> obwod = 2*pi*promien
```

```
obwod =
```

```
17.5929
```

```
>> pole = pi*promien^2
```

```
pole =
```

```
24.6301
```

Parę uwag ogólnych

- Kursor \uparrow przywołuje historię poleceń.
- Średnik na końcu linii zapobiega wyświetleniu wyniku operacji.

```
>> y = 2^2 + log(pi)*sin(x);  
>> y  
y =  
    3.1337
```


- Nazwa zmiennej Matlaba musi się rozpoczynać literą i może składać się z dowolnej ilości liter, cyfr i znaków podkreślenia.

```
>> Koty3 = 'czarny'
```

```
Koty3 =
```

```
czarny
```

```
>> 3koty = 'czarny'
```

```
3koty = 'czarny'
```

```
↑
```

```
Error: Unexpected MATLAB expression.
```

- Nie musimy deklarować zmiennych ani podawać ich rozmiaru.

```
>> kat_1 = 30
```

```
kat_1 =
```

```
30
```

```
>> kolor = 'czerwony'
```

```
kolor =
```

```
czerwony
```

```
>> predkosc = [-0.4 0.8 1.12 1.32 2.01]
```

```
predkosc =
```

```
-0.4000    0.8000    1.1200    1.3200    2.0100
```

- Jeśli pojawia się nowa zmienna, Matlab automatycznie tworzy ją i przydziela jej odpowiednią ilość pamięci.
- Jeśli zmienna o takiej nazwie już była, jej poprzednia wartość zostanie zastąpiona nową.

```
>> przyspieszenie = -2.71
```

```
przyspieszenie =
```

```
-2.7100
```

```
>> przyspieszenie = [-2.71 -2.69 -2.67]
```

```
przyspieszenie =
```

```
-2.7100    -2.6900    -2.6700
```

- Gdy chcemy zobaczyć co jest w zmiennej to wpisujemy jej nazwę w oknie poleceń i naciskamy enter.

```
>> Koty3
```

```
Koty3 =
```

```
czarny
```

- Polecenie `clear` pozwala pozbyć się niepotrzebnych już zmiennych.
- Pierwsza linia po komentarzach każdego skryptu powinna zawierać instrukcję:

```
clear all;
```

Podstawowe operacje arytmetyczne na liczbach

+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
^	potęgowanie

Odmiany tablicy dwuwymiarowej.

- skalar – tablica o rozmiarze 1×1
- wektor wierszowy – tablica o jednym wierszu
- wektor kolumnowy – tablica o jednej kolumnie

5

3
7
4

5	88	3	11
---	----	---	----

9	6	3
5	7	2

Definiowanie tablicy wymaga uwzględnienia następujących reguł:

- elementy w wierszu tablicy muszą być oddzielone spacją lub przecinkiem
 - średnik lub znak nowego wiersza (enter) kończy wiersz tablicy i powoduje przejście do następnego
 - cała lista elementów tablicy musi być ujęta w nawiasy kwadratowe
- ```
>>A = [1 5 3 4; 0 2 8 7];
```
- Odwołanie do elementu tablicy w nawiasie okrągłym:
    - **a(1,7)**: wartość elementu tablicy a z 1. wiersza i 7. kolumny,
    - **b(4)**: wartość 4. elementu tablicy b.

- Tablica  $2 \times 4$  zawierająca 8 elementów:

```
>>A = [1 5 3 4; 0 2 8 7];
```

Średnik na końcu polecenia powoduje, że nie pojawi się wynik operacji na ekranie. Otrzymamy tablicę o dwóch wierszach i czterech kolumnach.

```
>> A
```

```
A =
 1 5 3 4
 0 2 8 7
```



- Tablica  $1 \times 5$  zawierająca 5 elementów:

```
>> b = [1 2 3 4 5]
```

```
b =
```

```
 1 2 3 4 5
```

- Tablica  $3 \times 1$  zawierająca 3 elementy:

```
>> c = [1; 2; 3]
```

```
c =
```

```
 1
 2
 3
```

# Szybkie generowanie elementów z równym krokiem

- wykorzystujemy wyrażenie ogólne o postaci: `min : max`.

```
>> x=1:6
```

```
x =
```

```
1 2 3 4 5 6
```

- wykorzystujemy wyrażenie: `min : krok : max`

```
>> y=3:2.3:9
```

```
y =
```

```
3.0000 5.3000 7.6000
```

- różne kroki w różnych wierszach:

```
z=[1:10; 1:2:20]
```

```
z =
```

```
1 2 3 4 5 6 7 8 9 10
1 3 5 7 9 11 13 15 17 19
```

- wykorzystując funkcję: `linspace(początek, koniec, liczba_element)`

```
>> x = linspace(0, 10, 5)
```

```
x =
```

```
 0 2.5000 5.0000 7.5000 10.0000
```

# Proste operacje na tablicach

```
>> b = [7 9 3]
```

```
b =
 7 9 3
```

```
>> b(2) = 11
```

```
b =
 7 11 3
```

```
>> b(4) = 2
```

```
b =
 7 11 3 2
```

```
>> b(6) = 13
```

```
b =
 7 11 3 2 0 13
```

# Wybór fragmentu tablicy

- Można wybrać fragment tablicy i, np. zapisać go w innej zmiennej, podając wybrane rzędy i kolumny tablicy:

```
fragment = tablica(rz1:rz2, kol1:kol2)
```

rz1 i rz2 oznaczają początkowy i końcowy rząd fragmentu, a kol1 i kol2 początkową i końcową kolumnę fragmentu.

# Wybór fragmentu tablicy

```
>> tablica = rand(4,5)
```

```
tablica =
```

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| 0.967363 | 0.755262 | 0.782967 | 0.805672 | 0.743997 |
| 0.419995 | 0.694198 | 0.972953 | 0.698025 | 0.035414 |
| 0.075887 | 0.731045 | 0.671238 | 0.550291 | 0.873163 |
| 0.574697 | 0.033348 | 0.838181 | 0.153805 | 0.774990 |

- Wybór kolumny:

```
>> frag1 = tablica(:,3)
```

```
frag1 =
```

```
0.78297
0.97295
0.67124
0.83818
```

# Wybór fragmentu tablicy

```
>> tablica = rand(4,5)
```

```
tablica =
```

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| 0.967363 | 0.755262 | 0.782967 | 0.805672 | 0.743997 |
| 0.419995 | 0.694198 | 0.972953 | 0.698025 | 0.035414 |
| 0.075887 | 0.731045 | 0.671238 | 0.550291 | 0.873163 |
| 0.574697 | 0.033348 | 0.838181 | 0.153805 | 0.774990 |

- Wybór wiersza:

```
>> frag2 = tablica(2,:)
```

```
frag2 =
```

|          |          |          |          |          |
|----------|----------|----------|----------|----------|
| 0.419995 | 0.694198 | 0.972953 | 0.698025 | 0.035414 |
|----------|----------|----------|----------|----------|

# Wybór fragmentu tablicy

```
>> tablica = rand(4,5)
tablica =
 0.967363 0.755262 0.782967 0.805672 0.743997
 0.419995 0.694198 0.972953 0.698025 0.035414
 0.075887 0.731045 0.671238 0.550291 0.873163
 0.574697 0.033348 0.838181 0.153805 0.774990
```

- Wybór fragmentu tablicy:

```
>> frag3 = tablica(1:3, 4:5)
frag3 =
 0.805672 0.743997
 0.698025 0.035414
 0.550291 0.873163
```



# Proste operacje na tablicach

```
>> x = [1 2 3]
```

```
x =
```

```
 1 2 3
```

```
>> y = [2; 1; 5]
```

```
y =
```

```
 2
```

```
 1
```

```
 5
```

```
>> z = [2 1 0];
```

```
>> a = x + z
```

```
a =
```

```
 3 3 3
```

```
>> b = x + y
```

```
??? Error using ==> plus
```

```
Matrix dimensions must agree
```

```
>> a = x.*z
```

```
a =
```

```
 2 2 0
```

```
>> b = 2*a
```

```
b =
```

```
 4 4 0
```

Można także korzystać z funkcji do wyliczenia wartości elementów tablicy: `sqrt(x)`, `sin(x)` itp.

# Dodawanie wektorów

- Suma wektorów to wektor o składowych równych sumie ich składowych.
- Odejmowanie, to dodawanie wektora  $-\vec{B}$ .

W Matlabie:

```
>> A = [1 2 3];
```

```
>> B = [2 4 6];
```

```
>> C = A + B
```

```
C =
```

```
 3 6 9
```

```
>> D = B - A
```

```
D =
```

```
 1 2 3
```

# Mnożenie wektora przez skalar

Mamy  $\vec{A} = (A_x, A_y, A_z)$  i skalar  $m$ :

$$\vec{B} = m\vec{A}$$

taki, że  $\vec{B} = (mA_x, mA_y, mA_z)$

W Matlabie:

```
>> A = [1 2 4]
```

```
A =
```

```
1 2 4
```

```
>> B = 4*A
```

```
B =
```

```
4 8 16
```

# Operacje arytmetyczne na tablicach i macierzach

|             |                 |      |                |                    |      |                   |
|-------------|-----------------|------|----------------|--------------------|------|-------------------|
| Potęgowanie | <code>.^</code> | albo | <code>^</code> | <code>a.^ b</code> | albo | <code>a^ b</code> |
| Mnożenie    | <code>.*</code> | albo | <code>*</code> | <code>a.* b</code> | albo | <code>a*b</code>  |
| Dzielenie   | <code>./</code> | albo | <code>/</code> | <code>a./ b</code> | albo | <code>a/b</code>  |
|             | <code>.\</code> | albo | <code>\</code> | <code>b.\ a</code> | albo | <code>b\ a</code> |

*UWAGA:* `56/8=8\56`

|             |                |                  |
|-------------|----------------|------------------|
| Dodawanie   | <code>+</code> | <code>a+b</code> |
| Odejmowanie | <code>-</code> | <code>a-b</code> |
| Przypisanie | <code>=</code> | <code>a=b</code> |

# Macierz

## Macierz:

układ liczb, symboli lub wyrażeń zapisanych w postaci prostokątnej tablicy. Macierze jednowskaźnikowe nazywa się często wektorami wierszowymi lub kolumnowymi, co wynika z zastosowań macierzy w algebrze liniowej. W informatyce macierze modeluje się zwykle za pomocą (najczęściej dwuwymiarowych) tablic.

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & a_{22} & \dots & A_{2n} \\ \vdots & & & \vdots \\ A_{m1} & A_{m2} & \dots & A_{mn} \end{bmatrix}$$

# Macierz transponowana

Macierz transponowana do macierzy  $A$  o rozmiarze  $m \times n$ , to macierz  $A^T$  o rozmiarze  $n \times m$ :

$$(A^T)_{i,j} = A_{j,i}$$

- wiersze są zamienione z kolumnami
- transpozycja zamienia wektor kolumnowy w wierszowy i vice versa
- $(A^T)^T = A$

# Macierz transponowana

Przykład:

$$\begin{bmatrix} 0 & 7 & 3 & -2 \\ -1 & 2.5 & 13 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & -1 \\ 7 & 2.5 \\ 3 & 13 \\ -2 & 5 \end{bmatrix}$$

W Matlabie transpozycja to ':

```
octave:2> A = [0 7 3 -2; -1 2.5 13 5]
A =
 0.00000 7.00000 3.00000 -2.00000
 -1.00000 2.50000 13.00000 5.00000
octave:3> A'
ans =
 0.00000 -1.00000
 7.00000 2.50000
 3.00000 13.00000
 -2.00000 5.00000
```



# Dodawanie/odejmowanie macierzy

Jeśli  $A$  i  $B$  to macierze o takim samym rozmiarze  $m \times n$ , to sumę  $A + B$  otrzymujemy dodając do siebie poszczególne elementy.

Przykład:

$$\begin{bmatrix} 0 & 4 & 8 \\ -2 & 12 & -8 \end{bmatrix} + \begin{bmatrix} -4 & -7 & 0 \\ 16 & -9 & 13 \end{bmatrix} = \begin{bmatrix} -4 & -3 & 8 \\ 14 & 3 & 5 \end{bmatrix}$$

W Matlabie:

```
octave:4> A = [0 2; -8 12]
A =
 0 2
 -8 12
octave:5> B = [-5 7; 19 23]
B =
 -5 7
 19 23
octave:6> A+B
ans =
 -5 9
 11 35
```

# Dodawanie/odejmowanie macierzy

Własności:

- przemienne:  $A + B = B + A$
- łączne:  $(A + B) + C = A + (B + C)$
- Macierz zerowa:  $A + 0 = 0 + A = A$ ;  $A - A = 0$
- $(A + B)^T = A^T + B^T$

# Mnożenie macierzy przez skalar

Mnożymy macierz przez skalar (liczbę) mnożąc wszystkie elementy macierzy przez tą liczbę:

$$-2 \begin{bmatrix} 3 & 8 & -1 \\ -5 & 11 & 7 \end{bmatrix} = \begin{bmatrix} -6 & -16 & 2 \\ 10 & -22 & -14 \end{bmatrix}$$

- $(\alpha + \beta)A = \alpha A + \beta A$ ;  $(\alpha\beta)A = (\alpha)(\beta A)$
- $\alpha(A + B) = \alpha A + \alpha B$
- $0 \cdot A = 0$ ;  $1 \cdot A = A$

W Matlabie:

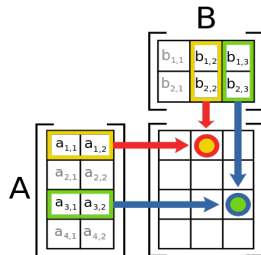
```
octave:7> A = [0 7 3 -2; -1 2.5 13 5]
A =
 0.00000 7.00000 3.00000 -2.00000
 -1.00000 2.50000 13.00000 5.00000
octave:8> 3*A
ans =
 0.00000 21.00000 9.00000 -6.00000
 -3.00000 7.50000 39.00000 15.00000
```

# Mnożenie macierzy

Jeśli  $A$  to macierz  $m \times p$  a  $B$  to macierz  $p \times n$  można obliczyć  $C = AB$  o rozmiarze  $m \times n$ :

$$C_{i,j} = \sum_{k=1}^p a_{i,k} b_{k,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \dots + a_{i,p}b_{p,j}; \quad i = 1, \dots, m; \quad j = 1, \dots, n$$

- rozmiar macierzy  $A$  i  $B$  musi być zgodny: liczba kolumn  $A$  musi być równa liczbie wierszy  $B$
- by znaleźć element  $i, j$  macierzy  $C = AB$ , potrzebne są  $i$ -ty rząd macierzy  $A$  i  $j$ -ta kolumna macierzy  $B$
- policz odpowiednie iloczyny, np. 3. wyraz  $i$ -tego wiersza  $A$  i 3. wyraz  $j$ -tej kolumny  $B$
- zsumuj iloczyny



# Mnożenie macierzy

Własności:

- $0A = 0$ ,  $A0 = 0$  (0 to skalar lub zgodna macierz zerowa)
- $IA = A$ ,  $AI = A$  ( $I$  – macierz jednostkowa, czyli ma jedynki na przekątnej i zera poza)
- $(AB)C = A(BC)$ , czyli  $ABC$
- $\alpha(AB) = (\alpha A)B$ ,  $\alpha$  to skalar
- $A(B + C) = AB + AC$ ,  $(A + B)C = AC + BC$
- $(AB)^T = B^T A^T$

# Mnożenie macierzy – Matlab

```
octave:14> A=[1 6; 4 3]
A =
 1 6
 4 3
octave:15> B = [2 5; -3 1]
B =
 2 5
 -3 1
octave:16> A*B
ans =
 -16 11
 -1 23
octave:17> B*A
ans =
 22 27
 1 -15
```

Mnożenie macierzy nie jest przemienne!