



Wprowadzenie do Scilab: macierze

Narzędzia Informatyki

Magdalena Deckert
Izabela Szczęch
Barbara Wołyńska
Bartłomiej Prędkie

Politechnika Poznańska

Instytut Informatyki

Agenda

- Definiowanie macierzy
- Funkcje do definiowania macierzy
- Pusta macierz
- Wymiary macierzy
- Operacje na poszczególnych elementach macierzy
- Dynamiczna zmiana wymiarów macierzy
- Podstawowe operacje na macierzach
- Macierzowe operatory porównania
- Wyznaczanie miejsc zerowych wielomianów
- Elementy algebry liniowej
 - Rozwiązywanie układów równań liniowych

Wprowadzenie

- Macierze są podstawowym typem danych w Scilab.
- Każda macierz definiowana jest przez:
 - liczbę wierszy
 - liczbę kolumn
 - typ danych wypełniających macierz (np. liczby rzeczywiste, całkowite, zmienne bool'owskie, ciągi znaków, wielomiany)
- Macierzami są również:
 - wektory – macierze o wymiarach $1 \times n$ lub $n \times 1$
 - skalary – macierze o wymiarach 1×1

Definiowanie macierzy

Elementy składni do definiowania macierzy:

- **[]** nawiasy kwadratowe zaznaczają początek i koniec macierzy
- **,** przecinek bądź **spacja** oddziela elementy z różnych kolumn
- **;** średnik bądź **Enter** oddziela elementy z różnych wierszy

Definiowanie macierzy

Przejdź do konsoli Scilab. Wykonaj poniższe polecenia definiujące macierze. Czy poniższe przykłady tworzą takie same macierze?

■ Przykład:

- ->A=[1,2,3;4,5,6;7,8,9]

■ Przykład:

- ->B=[1 2 3

- ->4 5 6

- ->7 8 9]

Przykład:

Konsola Scilab

```
-->C=[1 2 3
-->4,5,6; 7 8 ,9]
C =

    1.    2.    3.
    4.    5.    6.
    7.    8.    9.

-->|
```

Funkcje do definiowania macierzy

W Scilab wbudowanych jest wiele funkcji do tworzenia specyficznych macierzy, np. macierzy jednostkowych, macierzy zer czy jedynek.

Wykonaj poniższe przykłady w konsoli Scilab.

- **eye** – tworzy macierz jednostkową o zadanych rozmiarach

- **Przykład:**

- ->A=eye(2,2)

- **Przykład:**

- ->B=eye(2,5)

- **Przykład:**

- ->C=[1 2 3; 4 5 6]

- ->D=eye(C)

```
Konsola Scilab

-->A=eye(2, 2)
A =

    1.    0.
    0.    1.

-->B=eye(2, 5)
B =

    1.    0.    0.    0.    0.
    0.    1.    0.    0.    0.

-->C=[1 2 3; 4 5 6]
C =

    1.    2.    3.
    4.    5.    6.

-->D=eye(C)
D =

    1.    0.    0.
    0.    1.    0.

-->|
```

Funkcje do definiowania macierzy

- **ones** – tworzy macierz o zadanych rozmiarach wypełnioną jedynkami

- **Przykład:**

- ->A=ones(2,5)

- **zeros** – tworzy macierz o zadanych rozmiarach wypełnioną zerami

- **Przykład:**

- ->B=zeros(2,5)

Konsola Scilab

```
-->A=ones(2,5)
```

```
A =
```

```
1.    1.    1.    1.    1.
1.    1.    1.    1.    1.
```

```
-->B=zeros(2,5)
```

```
B =
```

```
0.    0.    0.    0.    0.
0.    0.    0.    0.    0.
```

```
-->
```

Funkcje do definiowania macierzy

- **testmatrix** – tworzy specyficzną macierz kwadratową o zadanych rozmiarach, np. odwróconą macierz Hilberta, kwadrat magiczny

- **Przykład:**

- -> `A=testmatrix('hilb',3)`

- **Przykład:**

- -> `B=testmatrix('magi',3)`

Konsola Scilab

```
-->A=testmatrix('hilb',3)
```

```
A =
```

```
    9.    - 36.    30.  
 - 36.    192.  - 180.  
    30.    - 180.    180.
```

```
-->B=testmatrix('magi',3)
```

```
B =
```

```
    8.     1.     6.  
    3.     5.     7.  
    4.     9.     2.
```

```
-->
```

Funkcje do definiowania macierzy

- **linspace (x_1 , x_2 , [n])** – tworzy wektor wierszowy zawierających **n** elementów równomiernie rozłożonych z przedziału $\langle x_1; x_2 \rangle$; domyślnie $n=100$

- **Przykład:**

- -> A=linspace (2, 10, 3)

- **Przykład:**

- -> B=linspace (10, -2, 3)

- **Przykład:**

- -> C=linspace (1, 50)

- **Przykład:**

- -> D=linspace (2, 13, 3)

Konsola Scilab

```
-->A=linspace (2, 10, 3)
```

```
A =  
  
    2.    6.   10.
```

```
-->B=linspace (10, -2, 3)
```

```
B =  
  
   10.    4.   - 2.
```

```
-->C=linspace(1,50)
```

```
C =  
  
      column 1 to 10  
  
    1.    1.4949495    1.989899  
  
      column 11 to 19
```

Funkcje do definiowania macierzy

- **rand** – tworzy macierz o zadanych rozmiarach wypełnioną wartościami losowymi z przedziału $<0, 1)$; dodatkowe parametry funkcji sterują np. rozkładem losowanych liczb

- **Przykład:**

- -> `A=rand(2,3)`

Konsola Scilab

```
-->A=rand(2,3)
A  =

    0.2639556    0.2806498    0.7783129
    0.4148104    0.1280058    0.2119030

-->
```

Pusta macierz

■ Przykład:

- -> `A=[]` //tworzy macierz o rozmiarach 0 x 0
- Zapis „`[]`” można wykorzystać do usunięcia zawartości macierzy i zwolnienia pamięci z nią związanej

Przykład:

- -> `A=ones(4,4)`
- -> `A=[]`

Konsola Scilab

```
-->A=ones(4,4)
```

```
A =
```

```
1.    1.    1.    1.
1.    1.    1.    1.
1.    1.    1.    1.
1.    1.    1.    1.
```

```
-->A=[]
```

```
A =
```

```
[]
```

```
-->|
```

Wymiary macierzy

- Funkcja **size (M)** zwraca informacje o wymiarach macierzy M (np.liczbie wierszy i kolumn)

Przykład:

```
- -> A=ones(2,3)
- -> [nr, nc]=size(A)
- -> size(A)
```

- Sprawdź działanie funkcji *size* dla 3-wymiarowej kostki

Przykład:

```
- -> B=ones(2,3,4)
- -> [n1, n2, n3]=size(B)
- -> size(B)
```

Konsola Scilab

```
-->A=ones(2,3)
A =

    1.    1.    1.
    1.    1.    1.

-->[nr, nc]=size(A)
nc =

    3.
nr  =

    2.

-->nr=size(A)
nr  =

    2.    3.

-->size(A)
ans =

    2.    3.

-->
```

Wymiary macierzy

- Funkcję *size* można też wywoływać z dwoma argumentami: **size (M, sel)**, gdzie parametr *sel* przyjmuje następujące wartości:
 - *sel*=1 lub *sel*="r" //funkcja zwraca liczbę wierszy
 - *sel*=2 lub *sel*="c" //funkcja zwraca liczbę kolumn
 - *sel*="*" //funkcja zwraca liczbę wszystkich elementów

Przykład:

- -> A=ones(2,3)

- -> size(A, "*")

Konsola Scilab

```
-->A=ones(2,3)
A =

    1.    1.    1.
    1.    1.    1.

-->size(A, "*")
ans =

    6.

-->
```

Operacje na poszczególnych elementach macierzy

- Operacje na elementach macierzy można wykonać odwołując się:
 - do całej macierzy podając jej nazwę, np. A
 - do elementu w i -tym wierszu i j -tej kolumnie, np. $A(i,j)$
 - do elementów z pewnego zakresu indeksów wierszy lub kolumn, korzystając z operatora „:”

Operacje na poszczególnych elementach macierzy

Odwołanie do całej macierzy

Przykład:

- -> `A=ones(2,3)`
- -> `B=2*ones(2,3)`
- -> `A+B`

Konsola Scilab

```
-->A=ones(2,3)
A  =

    1.    1.    1.
    1.    1.    1.

-->B=2*ones(2,3)
B  =

    2.    2.    2.
    2.    2.    2.

-->A+B
ans =

    3.    3.    3.
    3.    3.    3.

-->
```

Operacje na poszczególnych elementach macierzy

Odwołanie do elementu $A(i,j)$

Przykład:

```
- -> A=testmatrix('magi',3)
- -> A(1,2)
```

UWAGA: Należy pamiętać, że wiersze i kolumny macierzy indeksowane są od 1.

Przykłady:

```
- -> A=(0,1)
- -> A=(1,12)
```

```
Konsola Scilab

-->A=testmatrix('magi',3)
A  =

      8.      1.      6.
      3.      5.      7.
      4.      9.      2.

-->A(1,2)
ans =

      1.

-->A(0,1)
      !--error 21
Nieprawidłowy indeks.

-->A(1,12)
      !--error 21
Nieprawidłowy indeks.

-->
```

Operacje na poszczególnych elementach macierzy

Odwołanie przez operator „:”

- Ogólnie, składnia **i:[s]:j** pozwala wygenerować wektor liczb z przedziału $\langle i;j \rangle$ z krokiem s ; domyślnie $s=1$

Przykłady:

- -> $v=2:6$
- -> $v=3:2:10$
- -> $v=10:-2:3$
- -> $v=3:-2:10$

```
Konsola Scilab

-->v=2:6
v =

    2.    3.    4.    5.    6.

-->v=3:2:10
v =

    3.    5.    7.    9.

-->v=10:-2:3
v =

   10.    8.    6.    4.

-->v=3:-2:10
v =

    []

-->|
```

Operacje na poszczególnych elementach macierzy

Odwołanie przez operator „:”

- Operator „:” można też wykorzystać do określania zakresów indeksów wierszy i kolumn macierzy

Przykłady:

- -> A=testmatrix('magi',5)

- -> A(1:2, 3:4)

//wypisze elementy macierzy A
z wiersza 1 i 2 i kolumn 3 oraz 4

```
Konsola Scilab

-->A=testmatrix('magi',5)
A =

    17.    24.     1.     8.    15.
    23.     5.     7.    14.    16.
     4.     6.    13.    20.    22.
    10.    12.    19.    21.     3.
    11.    18.    25.     2.     9.

-->A(1:2,3:4)
ans =

     1.     8.
     7.    14.

-->
```

Operacje na poszczególnych elementach macierzy

Odwołanie przez operator „:”

Przykład:

- -> A=testmatrix('magi',5)

- -> A([1 5], :)=A([5 1],:)

//zamieni miejscami wiersz
pierwszy i piąty macierzy A

Przykład:

- -> B=testmatrix('magi',5)

- -> B(:, [1 5])=B(:, [5 1])

Konsola Scilab

```
-->A=testmatrix('magi',5)
```

```
A =
```

17.	24.	1.	8.	15.
23.	5.	7.	14.	16.
4.	6.	13.	20.	22.
10.	12.	19.	21.	3.
11.	18.	25.	2.	9.

```
-->A([1 5], :)=A([5 1],:)
```

```
A =
```

11.	18.	25.	2.	9.
23.	5.	7.	14.	16.
4.	6.	13.	20.	22.
10.	12.	19.	21.	3.
17.	24.	1.	8.	15.

```
-->|
```

Operacje na poszczególnych elementach macierzy

Odwołanie przez operator „:”

Podsumowanie wykorzystania operatora „:”

A	cała macierz
A(:, :)	cała macierz
A(i:j,k)	elementy z wierszy od i-tego do j-tego w kolumnie k-tej
A(i,j:k)	elementy z wiersza i-tego w kolumnach od j-tej do k-tej
A(i,:)	cały wiersz i-ty
A(:,j)	cała j-ta kolumna

Dynamiczna zmiana wymiarów macierzy

- Wymiary macierzy mogą się zmieniać dynamicznie w zależności od danych w macierzy.

Przykłady:

- -> A=[1 2 3; 4 5 6] // macierz 2 x 3

- -> A(3,1)=7

- -> A(:,3)=[]

Konsola Scilab

```
-->A=[1 2 3; 4 5 6]
```

```
A =
```

1.	2.	3.
4.	5.	6.

```
-->A(3,1)=7
```

```
A =
```

1.	2.	3.
4.	5.	6.
7.	0.	0.

```
-->A(:,3)=[]
```

```
A =
```

1.	2.
4.	5.
7.	0.

```
-->
```

Dynamiczna zmiana wymiarów macierzy

Przykład:

- -> A=[1 2 3; 4 5 6]

- -> B=[7;8]

- -> C=[A,B]

Konsola Scilab

```
-->A=[1 2 3; 4 5 6]
```

```
A =
```

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

```
-->B=[7;8]
```

```
B =
```

```
7.  
8.
```

```
-->C=[A,B]
```

```
C =
```

```
1.    2.    3.    7.  
4.    5.    6.    8.
```

```
-->
```

Podstawowe operacje na macierzach

Operacje macierzowe		Operacje tablicowe (na odpowiadających sobie elementach macierzy)	
+	dodawanie		
-	odejmowanie		
*	mnożenie	.*	mnożenie
/	prawostronne dzielenie $x/y = xy^{-1}$	./	prawostronne dzielenie
\	lewostronne dzielenie $x \backslash y = x^{-1}y$	.\	lewostronne dzielenie
^ lub **	potęgowanie	.^	potęgowanie
`	transponowanie macierzy i sprzężenie liczby zespólonej	.'	transponowanie macierzy

Podstawowe operacje na macierzach

- Na macierzach: $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $B = \begin{bmatrix} 5 & 6 & 7 & 8 \end{bmatrix}$ $C = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
wykonaj następujące działania:
 - $A+B$
 - $A+1$
 - $A+C$
 - $B*C$
 - $A*B$
 - $A.*B$
 - dodaj pierwszą kolumnę macierzy A z macierzą C

Podstawowe operacje na macierzach - wynik

Konsola Scilab

```
-->A=[1 2; 3 4]
```

```
A =
```

```
1.    2.  
3.    4.
```

```
-->B=[5 6; 7 8]
```

```
B =
```

```
5.    6.  
7.    8.
```

```
-->C=[1;2]
```

```
C =
```

```
1.  
2.
```

```
-->A+B
```

```
ans =
```

```
6.    8.  
10.   12.
```

```
-->A+1
```

```
ans =
```

```
2.    3.  
4.    5.
```

```
-->A+C
```

```
!--error 8
```

```
Niezgodne dodawanie.
```

```
-->B*C
```

```
ans =
```

```
17.  
23.
```

```
-->A*B
```

```
ans =
```

```
19.   22.  
43.   50.
```

```
-->A.*B
```

```
ans =
```

```
5.    12.  
21.   32.
```

```
-->A(:,[1])+C
```

```
ans =
```

```
2.  
5.
```

```
-->
```

Podstawowe operacje na macierzach

- Na wektorach: $u=[1;2;3]$ i $v=[4\ 5\ 6]$ wykonaj mnożenia

```
Konsola Scilab

-->u=[1
-->2
-->3]
u =

    1.
    2.
    3.

-->v=[5 6 7]
v =

    5.    6.    7.

-->u*v
ans =

    5.    6.    7.
   10.   12.   14.
   15.   18.   21.

-->v*u
ans =

   38.

-->u .*v
!--error 9999
inconsistent element-wise operation

-->v .*u
!--error 9999
inconsistent element-wise operation

-->
```

Macierzowe operatory porównania

- Do macierzy o takich samych wymiarach można stosować następujące operatory porównania:

<code>and(A, „r”)</code>	logiczne „i” na elementach wierszy macierzy A
<code>and(A, „c”)</code>	logiczne „i” na elementach kolumn macierzy A
<code>or(A, „r”)</code>	logiczne „lub” na elementach wierszy macierzy A
<code>or(A, „c”)</code>	logiczne „lub” na elementach kolumn macierzy A
<code><, &, </code>	operatory działające na odpowiadających sobie elementach macierzy

- Wynik zastosowania operatorów porównania jest macierzą wartości typu boolean.

Macierzowe operatory porównania

Przykłady:

- -> A=[1 2 7; 6 9 8]

- -> A>3

- -> B=[4 5 6; 7 8 9]

- -> A>B

- -> or(A>B, "r")

Konsola Scilab

```
-->A=[1 2 7; 6 9 8]
```

```
A =
```

```
1.    2.    7.  
6.    9.    8.
```

```
-->A>3
```

```
ans =
```

```
F F T  
T T T
```

```
-->B=[4 5 6; 7 8 9]
```

```
B =
```

```
4.    5.    6.  
7.    8.    9.
```

```
-->A>B
```

```
ans =
```

```
F F T  
F T F
```

```
-->or(A>B, "r")
```

```
ans =
```

```
F T T
```

```
-->|
```

Macierzowe operatory porównania

Przykłady:

- -> A=[1 2 7; 6 9 8]

- -> A>3

- -> A(A>3)=0

//wszystkie elementy
macierzy A o wartościach >3
zostaną zamienione na 0.

Konsola Scilab

```
-->A=[1 2 7; 6 9 8]
```

```
A =
```

```
1.    2.    7.  
6.    9.    8.
```

```
-->A>3
```

```
ans =
```

```
F F T  
T T T
```

```
-->A(A>3)=0
```

```
A =
```

```
1.    2.    0.  
0.    0.    0.
```

```
-->|
```

Wyznaczanie miejsc zerowych wielomianów

- Funkcja **roots (p)** służy do wyznaczania miejsc zerowych dowolnych wielomianów.

Jako parametr p podajemy współczynniki przy kolejnych potęgach (uporządkowane wg malejących potęg).

- **Przykłady:**

Wyznacz miejsca zerowe wielomianu:

$$x^3 - 6x^2 + 11x - 6 = 0$$

- -> p=[1 -6 11 -6]

- -> roots(p)

Konsola Scilab

```
-->p=[1 -6 11 -6]
```

```
p =
```

```
1. - 6. 11. - 6.
```

```
-->roots(p)
```

```
ans =
```

```
3.
```

```
2.
```

```
1.
```

```
-->
```

Elementy algebry liniowej

- Wybrane funkcje algebry liniowej:

$\det(A)$	obliczanie wyznacznika macierzy A
$\text{spec}(A)$	obliczanie wartości własnych macierzy A
$\text{rank}(A)$	obliczanie rzędu macierzy A
$\text{diag}(A)$	wyznaczanie elementów leżących na głównej przekątnej macierzy A
$\text{inv}(A)$	wyznaczenie macierzy odwrotnej do macierzy A

Rozwiązywanie układów równań liniowych

Za pomocą macierzy można rozwiązywać układy równań liniowych.

Układ równań liniowych:

$$\begin{cases} x + 2y + 3z = 4 \\ 3x + 4y + 5z = 14 \\ 2x + 6y + 6z = 20 \end{cases}$$

można zapisać macierzowo jako: $A \cdot X = b$, gdzie

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 2 & 6 & 6 \end{bmatrix} \quad b = \begin{bmatrix} 4 \\ 14 \\ 20 \end{bmatrix}$$

UWAGA:

- układ ma **jedno rozwiązanie**, gdy
 $\text{rank}(A) = \text{rank}([A \ b]) = \text{liczba niewiadomych w układzie}$
- układ ma **nieskończenie wiele rozwiązań**, gdy
 $\text{rank}(A) = \text{rank}([A \ b]) < \text{liczba niewiadomych w układzie}$
- układ **nie ma rozwiązań**, gdy
 $\text{rank}(A) \neq \text{rank}([A \ b])$

Rozwiązywanie układów równań liniowych

- Równanie macierzowe $A \cdot X = b$, gdzie

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 2 & 6 & 6 \end{bmatrix} \quad b = \begin{bmatrix} 4 \\ 14 \\ 20 \end{bmatrix}$$

ma 1 rozwiązanie.

Porównaj wartości poniższych funkcji:

-->rank(A)

-->rank([A b])

Konsola Scilab

```
-->A=[1 2 3
```

```
-->3 4 5
```

```
-->2 6 6]
```

```
A =
```

```
1.    2.    3.
```

```
3.    4.    5.
```

```
2.    6.    6.
```

```
-->b=[4
```

```
-->14
```

```
-->20]
```

```
b =
```

```
4.
```

```
14.
```

```
20.
```

```
-->rank(A)
```

```
ans =
```

```
3.
```

```
-->rank([A b])
```

```
ans =
```

```
3.
```

Rozwiązywanie układów równań liniowych

Rozwiązanie równania macierzowego: $A * X = b$

- **Sposób I:**

--> $X = A \backslash b$

Polecenie rozwiązuje zadany układ za pomocą operatora dzielenia w lewo. Ponieważ nie jest możliwe dzielenie wektora przez macierz, operator odwraca macierz A , a następnie mnoży ją lewostronnie przez wektor b .

- **Sposób II:**

--> $X = \text{inv}(A) * b$

Polecenie rozwiązuje zadany układ równań bezpośrednio: funkcja $\text{inv}(A)$ odwraca macierz A , a następnie wynik mnożony jest przez wektor b .

Rozwiązywanie układów równań liniowych

Konsola Scilab

```
-->A
```

```
A =
```

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

```
-->b
```

```
b =
```

```
4.  
14.  
20.
```

```
-->X=A\b
```

```
X =
```

```
2.5  
6.  
- 3.5
```

```
-->X=inv(A)*b
```

```
X =
```

```
2.5  
6.  
- 3.5
```

```
-->
```

Literatura

Materiały przygotowane na podstawie "Introduction to Scilab".

<http://www.scilab.org/support/documentation/tutorials>