3. Operacje na macierzach

Podstawowym typem danych występującym w MATLAB-ie są **macierze**. Typem uzupełniającym są łańcuchy znakowe.

3.1. Definiowanie macierzy

1. przez wyliczanie elementów
2. przez generowanie elementów
3. budowa z innych macierzy
4. wykorzystanie funkcji
5. mieszanie technik 1. 2. 3. i 4.

3.1.1. Wyliczanie elementów

Elementy macierzy podaje się w nawiasach kwadratowych oddzielając wiersze średnikami lub naciskając klawisz Enter. Elementy w wierszach mogą być oddzielane spacją lub przecinkiem.

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

A =
      1   2
      3   4
```

lub

```
>> A = [1 0
0 1]

A =
     1   0
     0   1
```

Jeżeli w wierszu edycyjnym nie można umieścić wszystkich elementów wiersza macierzy, to można:

- zakończyć wiersz spacją i trzema kropkami
- nacisnąć Enter
- i kontynuować wpisywanie elementów w następnym wierszu edycyjnym:

```
>> A = [1 2 3 4 5 ...
6 7 8 9 10]
A =
    1 2 3 4 5 6 7 8 9 10
```

3.1.2. Generowanie elementów

Do generowania elementów wykorzystuje się wyrażenie: **min : krok : max**. Generuje ono wektor wierszowy którego pierwszym elementem jest **min**, kolejne elementy to **min** + **krok**, **min** + 2***krok**. Elementy generowane są dopóki nie zostanie przekroczona wartość **max**.

```
>> B = 1 : 2 : 5
B =
    1   3   5
```

Można pomijać parametr **krok** w wyrażeniu, wtedy wartość kroku jest przyjmowana jako 1:

```
>> B = 1 : 10
B =
    1  2  3  4  5  6  7  8  9  10
```

3.1.3. Budowa z innych macierzy

Macierz można budować z innych, wcześniej utworzonych macierzy, przy czym odpowiednie wymiary macierzy muszą sobie odpowiadać:

```
>> A = [1 2; 3 4]
A =
    1   2
    3   4
>> B = [5 6; 7 8]
B =
    5   6
    7   8
```

Macierz C zostanie utworzona z macierzy A i B. Macierz A będzie stanowiła górne dwa wiersze, macierz B dolne wiersze macierzy C (operację tą można wykonać ze względu na równą liczbę kolumn macierz A i B).

```
>> C = [A; B]

C =

    1   2
    3   4
    5   6
    7   8
```

Macierz D zostanie utworzona z macierzy A i B. Macierz A będzie stanowiła dwie pierwsze kolumny, macierz B kolejne kolumny macierzy D (operację tą można wykonać ze względu na równą liczbę wierszy macierzy A i B).

```
>> D = [A B]

D =

    1   2   5   6
    3   4   7   8
```

3.1.4. Wykorzystanie funkcji

Funkcja	**Opis**
zeros(n) zeros(n, m)	tworzy macierz wypełnioną zerami o wymiarach $n \times n$ lub $n \times m$
eye(n) eye(n, m)	tworzy macierz jednostkową o wymiarach $n \times n$ lub $n \times m$.
ones(n) ones(n, m)	tworzy macierz o wszystkich elementach równych 1, o wymiarach $n \times n$ lub $n \times m$
rand(n) rand(n, m)	Wypełnia macierz $n \times n$ lub $n \times m$ liczbami pseudolosowymi o rozkładzie jednostkowym na przedziale <0, 1>

randn(n) randn(n, m)	Wypełnia macierz $n \times n$ lub $n \times m$ liczbami pseudolosowymi o rozkładzie normalnym

```
>> C = eye(3)

C =

     1   0   0
     0   1   0
     0   0   1

>> D = zeros(2, 3)

D =

     0   0   0
     0   0   0
```

3.1.5. Mieszanie technik

Wymienione metody generowania macierzy można łączyć, np.:

```
>> A = [1  0  1]

A =

     1   0   1

>> B = [A; 1 : 2 : 5; zeros(1, 3)]

B =

     1   0   1

     1   3   5

     0   0   0
```

3.2. Dostęp do elementów macierzy

Do elementów macierzy można odwoływać się podając:

- numer wiersza i numer kolumny elementu
- numery wierszy i numery kolumn elementów.

Jeżeli dana jest macierz **M** to odwołanie **M(k, l)**:

jeżeli **k** i **l** są skalarami daje dostęp do elementu z **k**-tego wiersza i **l**-tej kolumny

jeżeli **k** i **l** są wektorami daje dostęp do elementów znajdujących się w wierszach, których numery znajdują się w wektorze **k**, i kolumnach, których numery znajdują się w wektorze **l**.

Przykłady

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

A =

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

– odwołanie do elementu znajdującego się w drugim wierszu i trzeciej kolumnie macierzy A (liczba 6)

```
>> x = A(2, 3)

x =

     6
```

– odwołanie do elementów trzeciego wiersza macierzy z kolumny 1 i 2:

```
>> x = A(3, [1 2])

x =

     7   8
```

– odwołanie do elementów trzeciego wiersza macierzy (1 sposób):

```
>> x = A(3, [1  2  3])

x =

    7   8   9
```

– odwołanie do elementów trzeciego wiersza macierzy (2 sposób):

```
>> x = A(3, 1 : 3)

x =

    7   8   9
```

– odwołanie do elementów trzeciego wiersza macierzy (3 sposób),

jeżeli potrzebne jest odwołanie do całego wiersza bądź kolumny macierzy można użyć **dwukropka,** który oznacza w tym przypadku „wszystkie wiersze” lub „wszystkie kolumny”:

```
>> x = A(3, : )

x =

    7   8   9
```

– odwołanie do elementów macierzy: A_{11}, A_{13}, A_{31}, A_{33}:

```
>> x = A([1  3], [1  3])

x =

    1   3
    7   9
```

Dostęp do elementów macierzy może być wykorzystywany nie tylko do pobierania elementów macierzy (jak w przykładach powyżej), ale również do nadawania wartości tym elementom:

– wyzerowanie pierwszej kolumny macierzy (A(: , 1) – elementy pierwszej kolumny macierzy A):

```
>> A( : , 1) = 0

A =

    0   2   3
    0   5   6
    0   8   9
```

3.3. Operacje macierzowe

3.3.1. Operatory macierzowe

MATLAB udostępnia zestaw operatorów, umożliwiających wykonanie operacji na macierzach:

+ dodawanie,

– odejmowanie,

* mnożenie,

/ dzielenie

(dzielenie macierzy A przez B jest równoważne mnożeniu A przez odwrotność B: A/B ⇔ A * B^{-1}),

^ potęgowanie

(podnoszenie macierzy A do potęgi np.: 2 jest równoważne obliczeniu A * A;

podnoszenie macierzy A do potęgi np.: 1/2 jest równoważne obliczeniu √A),

’ transpozycję.

***Uwaga!* Operacje na macierzach wymagają odpowiednich rozmiarów macierzy występujących w wyrażeniach.**

Przykłady

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

A =
     1   2
     0   1

>> B = eye(2)

B =
     1   0
     0   1
```

– macierz C powinna być wyliczona jako suma macierzy A i B:

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

```
C =
      2   2
      0   2
```

– macierz D powinna być wyliczona jako iloczyn macierzy A i B:

```
>> D = A * B

D =
      1   2
      0   1
```

– macierz E powinna być równa A^2:

```
>> E = A^2

E =
      1   4
      0   1
```

– macierz F powinna być równa A^{-1}:

```
>> F = A^(-1)

F =
      1   -2
      0   1
```

– macierz G powinna być równa transpozycji A:

```
>> G = A'

G =
      1   0
      2   1
```

***Uwaga!* Operatory maja różne priorytety, np.: mnożenie ma wyższy priorytet niż dodawanie, a potęgowanie i transpozycja niż np. mnożenie. Jeżeli kolejność wykonania operacji ma być inna niż to wynika z priorytetów operatorów należy zastosować nawiasy. Np.: żeby zsumować macierze A i B, a następnie obliczoną sumę przemnożyć przez macierz C należy wykonać operację: (A + B)*C a nie operację A + B * C.**

3.3.2. Funkcje macierzowe

MATLAB udostępnia zestaw funkcji umożliwiających wykonanie operacji na macierzach np.:

Funkcja	**Opis**
sum(A) sum(A, k)	jeżeli A jest wektorem funkcja oblicza sumę elementów wektora jeżeli A jest macierzą funkcja wyznacza wektor, którego kolejne elementy zawierają sumy kolejnych kolumn macierzy A jeżeli w wywołaniu funkcji zostanie podany dodatkowy parametr np. k to wskazuje po którym wymiarze macierz będzie sumowana (1 – po kolumnach, 2 – po wierszach).
prod(A)	jeżeli A jest wektorem funkcja oblicza iloczyn elementów wektora jeżeli A jest macierzą funkcja wyznacza wektor, którego kolejne elementy zawierają iloczyny kolejnych kolumn macierzy A
inv(A)	wyznacza macierz odwrotną do macierzy A
det(A)	oblicza wyznacznik macierzy A
rank(A)	oblicza rząd macierzy A
diag(A)	zwraca główna przekątną macierzy A

Przykłady

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

A =

      1   2

      0   1

>> B = sum(A)

B =

      1   3

>> x = det(A)

x =

      1
```

3.4. Operacje tablicowe

Do grupy operacji tablicowych zaliczone zostały takie operatory i funkcje, które nie traktują macierzy jako elementu algebry lecz jako zbiór odrębnych elementów.

3.4.1. Operatory tablicowe

.* mnożenie tablicowe,

./ dzielenie tablicowe,

.^ potęgowanie tablicowe.

Uwaga! **Nie ma operatorów: .+ i .– gdyż są one niepotrzebne, operatory + i – są z definicji operatorami tablicowymi.**

Przykłady:

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

A =

    1   2
    0   1

>> B = eye(2)

B =

    1   0
    0   1

>> C = A .* B

C =

    1   0
    0   1

>> D = A .^ 2

D =

    1   4
    0   1
```

3.4.2. Funkcje tablicowe

MATLAB udostępnia zestaw funkcji, które przekształcają każdy z elementów macierzy z osobna:

Funkcja	**Opis**
trygonometryczne	
sin(A), cos(A), tan(A)	wyznacza macierz, której elementy są równe sinusom, cosinusom, tangensom odpowiednich elementów macierzy A
logarytmiczne, wykładnicze, potęgowe	
sqrt(A)	zwraca macierz pierwiastków kwadratowych macierzy A
exp(A)	zwraca macierz o elementach równych $e^{A_{ij}}$
log(A)	Logarytm naturalny elementów macierzy A
log2(A)	Logarytm przy podstawie 2 elementów macierzy A
log10(A)	Logarytm dziesiętny elementów macierzy A
działające na liczbach zespolonych	
abs(A)	macierz o elementach równych wartościom bezwzględnym odpowiadających elementów macierzy A
real(A)	macierz części rzeczywistych elementów macierzy A
imag(A)	macierz części urojonych elementów macierzy A

Przykłady

```
>> A = [0  3.1415/2  3.1415]

A =

     0   1.5708   3.1415

>> B = sin(A)

B =

     0   1.0000   0.0001
```

```
>> A = [0  1]

A =

     0   1

>> B = exp(A)

B =

     1.0000   2.7183
```

Ćwiczenia

1. Zdefiniuj macierz A jako: $A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$, macierz B jako: $B=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, macierz C jako: $C=\begin{bmatrix}1 & 1\\ 2 & 1\end{bmatrix}$.

2. Oblicz D jako: A+B, E jako: $(A + B)*C^{-1}$, F jako $(A + B^T)^{-1}$

3. Zdefiniuj macierz G jako: $G=\begin{bmatrix}1 & 2\\ 0 & 0\end{bmatrix}$.

4. Oblicz wyznacznik macierzy G.

5. Oblicz H jako G^{-1}. Jak zinterpretujesz wynik?

6. Funkcja magic(n) tworzy macierz o wymiarach $n\times n$, która jest tzw. kwadratem magicznym (sumy elementów w wierszach, kolumnach i na przekątnych są równe). Korzystając z tej funkcji utwórz macierz M o wymiarach 5x5 i sprawdź czy jest kwadratem magicznym (wykorzystaj odpowiednie funkcje MATLAB-a).

7. Zdefiniuj wektor X tak aby zawierał kolejne wartość całkowite z przedziału od 1 do 100.

8. Zdefiniuj wektor Y tak aby zawierał kwadraty kolejnych elementów wektora X.