

# Laboratorium Techniki Obliczeniowej i Symulacyjnej

## Ćwiczenie 2. Podstawowe operacje macierzowe.

Opracował: dr inż. Sebastian Dudzik

### 1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z tworzeniem macierzy i wektorów w programie MATLAB. W ćwiczeniu wprowadzono opis podstawowych operacji macierzowych takich jak transpozycja, odwracanie itp. Dodatkowym celem jest nabycie praktycznych umiejętności posługiwania się funkcjami i operatorami macierzowymi języka MATLAB.

### 2. Wprowadzenie

Podstawowym typem danych języka MATLAB jest macierz (wektor). W pierwszych wersjach programu był to jedyny obsługiwany typ danych. W wersjach nowszych wszystkie typy strukturalne (złożone) są oparte na pojęciu macierzy. Macierz definiuje się jako funkcję, przyporządkowującą parze indeksów:  $i, j$ , wartość liczbową  $a_{i,j}$  (rzeczywistą lub zespoloną) zwaną elementem macierzy. Bardziej obrazowo macierz (dwuwymiarową) można wyobrazić sobie jako prostokątną tablicę liczb. Zapis macierzy o wymiarze  $m \times n$ , gdzie:  $m$  — liczba wierszy,  $n$  — liczba kolumn przedstawia wyrażenie (1).

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (1)$$

Szczególnym przypadkiem macierzy jest wektor. Wyrażenie (2) przedstawia różne zapisy wektorowe. Jeżeli wektor składa się z jednego wiersza, nazywamy go wektorem wierszowym, natomiast jeżeli posiada tylko jedną kolumnę — wektorem kolumnowym.

$$\mathbf{V}_w = \left( a_1 \ a_2 \ \dots \ a_n \right), \quad \mathbf{V}_k = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \quad (2)$$

## 2.1. Tworzenie macierzy

W języku MATLAB istnieje wiele sposobów utworzenia wektorów (macierzy). Poniżej zestawiono trzy spośród nich. Inne sposoby zostaną wprowadzone w opisie realizacji ćwiczenia. Macierze w języku MATLAB tworzy się m.in.:

- poprzez wyliczenie elementów,
- poprzez zastosowanie funkcji generującej,
- z innych macierzy (wektorów).

Aby utworzyć zmienną wektorową przez wyliczenie, elementy wektora należy zamknąć wewnątrz nawiasów kwadratowych, np.:  $\mathbf{a} = [1 \ 2 \ 3 \ 4 \ 5]$ . Utworzony wektor jest wektorem wierszowym ponieważ odstęp (spacja) rozdziela kolumny. Polecenie  $\mathbf{b} = [1; 2; 3; 4; 5]$  tworzy natomiast wektor kolumnowy, gdyż do rozdzielania wierszy używa się średnika (;). Zestawiając ze sobą dwa powyższe sposoby można utworzyć macierz:  $\mathbf{a} = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$ . Powyższe polecenie tworzy macierz zdefiniowaną wyrażeniem (3)

$$\mathbf{a} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad (3)$$

Do tworzenia macierzy i wektorów w programie MATLAB można wykorzystać wiele wbudowanych funkcji generujących. W tab. 1 zestawiono niektóre z nich.

Tab. 1. Wybrane funkcje środowiska MATLAB do generowania macierzy

Nazwa funkcji	Działanie
<code>zeros(m,n)</code>	Generuje macierz o wymiarze $m \times n$ wypełnioną zerami
<code>ones</code>	Generuje macierz o wymiarze $m \times n$ wypełnioną jedynkami
<code>randn(m,n)</code>	Generuje macierz o wymiarze $m \times n$ wypełnioną wartościami pseudolosowymi

Innym sposobem tworzenia wektorów i macierzy jest wykorzystanie zmiennych znajdujących się już w przestrzeni roboczej środowiska MATLAB. Ten sposób tworzenia jest szczególnie przydatny gdy zachodzi potrzeba wykorzystania uzyskanych wcześniej wyników w kolejnym etapie obliczeń. Ten sposób tworzenia zmiennych został objaśniony w części praktycznej.

## 2.2. Dostęp do elementów macierzy

Operacje na macierzach wymagają dostępu do ich elementów. Dostęp ten uzyskuje się poprzez indeksy. Indeks stanowi numer wiersza i/lub kolumny umieszczony wewnątrz

nawiasów zwykłych. Dla przykładu, aby odwołać się do elementu leżącego na przecięciu drugiego wiersza i trzeciej kolumny macierzy opisanej wzorem (3) należy wprowadzić polecenie: `>>a(2,3)`.

Oprócz prostego indeksowania zmiennych w języku MATLAB można odwoływać się do części macierzy lub wektorów. Umożliwia to tzw. notacja dwukropkowa. Dla przykładu, jeżeli w obliczeniach niezbędne są dane zawarte w drugiej kolumnie macierzy `a` — wzór (3) — dane te można uzyskać wprowadzając polecenie `>>a(:,2)`. Dwukropek można odczytać jako „wszystkie wiersze”, natomiast stojąca po przecinku liczba 2 oznacza, że wiersze te wybiera się z drugiej kolumny. Po wprowadzeniu powyższego polecenia do zmiennej `ans` zostanie przypisany wektor postaci:

$$\mathbf{ans} = \begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix} \quad (4)$$

### 2.3. Operacje tablicowe i funkcje wspomagające przetwarzanie macierzy

W poprzednim ćwiczeniu wprowadzono symbole podstawowych operatorów (+), (−), (\*), (/) i (^). W przypadku, gdy argumentami operacji są macierze (lub wektory), operatory te działają macierzowo. Na przykład operator (\*), umożliwia mnożenie macierzy w sensie Cauchy’ego (iloczyn macierzowy). Aby wymusić działanie operatora na poszczególnych elementach macierzy lub wektora należy operator poprzedzić kropką. Taki operator nazywany jest operatorem tablicowym.

W języku MATLAB istnieje wiele funkcji wspomagających przetwarzanie macierzy. W tab. 2 zestawiono niektóre z nich.

Tab. 2. Wybrane funkcje przetwarzające macierze

Nazwa funkcji	Działanie
<code>isempty(x)</code>	Zwraca 1 gdy <code>x</code> jest puste
<code>fliplr(x)</code>	Odwraca kolejność kolumn
<code>tril(x)</code>	Zwraca dolną macierz trójkątną
<code>hilb(N)</code>	Tworzy macierz Hilberta o wymiarze $N \times N$

Praktyczne wykorzystanie funkcji macierzowych opisano w p. 3 niniejszej instrukcji.

### 3. Program ćwiczenia

1. Uruchomienie programu MATLAB.

W ćwiczeniu wykorzystano program MATLAB w wersji 5.3 (R11.1). Uruchomienie programu następuje poprzez skrót na pulpicie (Matlab5.3) lub bezpośrednio z katalogu  $C:\MatlabR11\bin\$ .

2. Uruchomienie programu Wordpad.exe.

Program można uruchomić poprzez wywołanie:  $Start\Programy\Akcesoria\Wordpad$  lub poprzez skrót na pulpicie.

3. Przejście do katalogu roboczego dla grupy laboratoryjnej.

Domyślnym katalogiem startowym (roboczym) programu MATLAB jest  $C:\MatlabR11\work\$ . Zadanie polega na przejściu do podkatalogu katalogu  $work$ . Podkatalog (utworzony na pierwszych zajęciach laboratoryjnych) nazwany jest wybranymi 2 nazwiskami studentów, wchodzących w skład grupy laboratoryjnej.

- (a) Wprowadzić:

```
>>pwd
```

W programie MATLAB każde wprowadzone polecenie zatwierdza się klawiszem <ENTER>. Zwrócić uwagę na ścieżkę dostępu do katalogu bieżącego.

- (b) Wprowadzić:

```
>>cd nazwa_podkatalogu
```

Parametr  $nazwa\_pod-katalogu$  powinien składać się z nazwisk 2 wybranych studentów grupy laboratoryjnej (np.  $>>cd KowalskiNowak$ ).

4. Utworzenie wektorów wierszowych o elementach:  $x_1 : 2, 4, 6, 8, \dots, 20$ ;  $x_2 : 10, 8, 6, 4, 2, 0, -2, -4$ ;  $x_3 : 1, 1/2, 1/3, 1/4, 1/5$ ;  $x_4 : 0, 1/2, 2/3, 3/4, 4/5$ ;  $x_5$ : całkowite liczby parzyste pomiędzy 31 i 75.

- (a) Wprowadzić:

```
>>x1=[2 4 6 8 10 12 14 16 18 20]
```

Wprowadzony wektor wierszowy składa się z 10 kolumn. Do oddzielania elementów tego samego wiersza (separator kolumn) używa się spacji. Można zauważyć, że elementy wektora są liczbami parzystymi w zakresie od 2 do 20. Tworzenie wektora można zautomatyzować. Wprowadzić:

```
>>x1=2:2:20
```

W zapisie wykorzystano tworzenie wektora ze stałym krokiem:  $wart\_pocz: krok: wart\_końc$ . Porównać wyniki wykonania obu poleceń.

- (b) Wprowadzić:

```
>>x2=[10 8 6 4 2 0 -2 -4]
```

Wprowadzony wierszowy składa się z 8 kolumn. Można zauważyć, że elementy wektora są liczbami parzystymi w zakresie od 10 do -4. Tworzenie wektora można zautomatyzować. Wprowadzić:

```
>>x2=10:-2:-4
```

Porównać wyniki wykonania obu poleceń.

(c) Wprowadzić:

```
>>x3=[1 1/2 1/3 1/4 1/5]
```

Utworzony wektor wierszowy ma 5 kolumn. Można zauważyć, że mianowniki elementów wektora rosną. Generowanie wektora można zautomatyzować tworząc oddzielnie wektory licznika i mianownika i dokonując dzielenia tablicowego.

Wprowadzić:

```
>>l=[1 1 1 1 1] % wektor licznika
```

```
>>m=[1 2 3 4 5] % wektor mianownika
```

```
>>l./m % wektor wynikowy
```

Znaków %, oraz występujących po nich napisów nie trzeba wprowadzać. MATLAB traktuje je jako komentarze i nie wpływają one na działanie wykonywanych poleceń. Wprowadzanie wektora mianownika, można jeszcze skrócić.

Wprowadzić:

```
>>m=1:5 % mianownik jako wektor ze stałym krokiem
```

```
>>l./m % wektor wynikowy
```

Pominięcie wartości kroku powoduje przyjęcie kroku jednostkowego. Porównać wyniki wykonania obu poleceń.

(d) Wprowadzić:

```
>>x4=[0 1/2 2/3 3/4 4/5]
```

Utworzony wektor wierszowy ma 5 kolumn. Można zauważyć, że zarówno mianowniki, jak i liczniki elementów wektora rosną. Generowanie wektora można zautomatyzować, tworząc oddzielnie wektory licznika i mianownika i dokonując dzielenia tablicowego. Wprowadzić:

```
>>l=0:4 % licznik jako wektor ze stałym krokiem
```

```
>>m=1:5 % mianownik jako wektor ze stałym krokiem
```

```
>>l./m % wektor wynikowy
```

(e) Wprowadzić:

```
>>x1=32:2:74
```

Wektor zawiera całkowite liczby parzyste pomiędzy 31 i 75. Został wygenerowany jako wektor ze stałym krokiem (co 2).

(f) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.

(g) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:

```
>>clc
```

5. Utworzenie wektorów kolumnowych o elementach:  $y_1 : 1, 4, 9, 16, \dots, 100$ ;  $y_2 : 1/5, 1/10, 1/15, 1/20, 1/25$ ;  $y_3$ : całkowite liczby nieparzyste pomiędzy 102 i 117.

(a) Wprowadzić:

```
>>y1=[1; 4; 9; 16; 25; 36; 49; 64; 81; 100]
```

Utworzony wektor kolumnowy ma 10 wierszy. Do oddzielania elementów tej samej kolumny (separator wierszy) używa się średnika. Można zauważyć, że

elementy tworzonego wektora są kwadratami kolejnych liczb całkowitych od 1 do 10. Tworzenie wektora można zautomatyzować. Wprowadzić:

```
>>yy=1:10 % utworzenie wektora podstaw
>>y=yy.^2 % utworzenie wierszowego wektora elementów
>>y1=y' % utworzenie wektora kolumnowego (transpozycja)
```

Znaków %, oraz występujących po nich napisów nie trzeba wprowadzać. MATLAB traktuje je jako komentarze i nie wpływają one na działanie wykonywanych poleceń. Porównać wyniki wykonania obu poleceń. Często metodą tworzenia wektorów kolumnowych, jest transpozycja. UWAGA: przed operatorem potęgowania (^) musi wystąpić kropka ponieważ jest to operator tablicowy.

(b) Wprowadzić:

```
>>y2=[1/5; 1/10; 1/15; 1/20; 1/25] Utworzony wektor kolumnowy ma 5 wierszy. Generowanie wektora można zautomatyzować, tworząc oddzielnie wektory licznika i mianownika, dokonując dzielenia tablicowego oraz transpozycji. Wprowadzić:
```

```
>>l=[1 1 1 1 1] % wektor liczników elementów
>>m=[5 10 15 20 25] % wektor mianowników elementów
>>y2=(l./m)' % wektor wynikowy
```

Porównać wynik z wynikiem uzyskanym poprzednio. Można zauważyć, że mianowniki elementów wektora tworzą ciąg arytmetyczny o stałej różnicy równej 5. Generowanie wektora można dodatkowo uprościć. Wprowadzić:

```
>>m=5:5:25 % wektor mianowników jako wektor ze stałym krokiem
>>y2=(l./m)' % wektor wynikowy
```

Porównać wynik z wynikami uzyskanym poprzednio.

(c) Wprowadzić:

```
>>y3=[103; 105; 107; 109; 111; 113; 115; 117]
```

Generowanie wektora można zautomatyzować. Wprowadzić:

```
>>y3=(103:2:117)'
```

Aby zamienić wektor wierszowy ze stałym krokiem na kolumnowy, należy użyć operatora transpozycji, oraz nawiasów zwykłych. Porównać wynik z wynikiem uzyskanym poprzednio.

(d) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.

(e) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:

```
>>clc
```

6. Przeprowadzanie operacji tablicowych na wektorach kolumnowych o następujących elementach:  $x = (3 \ 2 \ 6 \ 8)$  i  $y = (4 \ 1 \ 3 \ 5)$ . W celu realizacji zadania wykonać poniższe polecenia.

(a) Utworzyć w przestrzeni roboczej wektory  $x$  i  $y$ .

Wprowadzić:

```
>>x=[3 2 6 8]'; y=[4 1 3 5]'
```

Polecenia języka MATLAB można wprowadzać w jednej linii. W takim przypadku należy je rozdzielić średnikami.

- (b) Dodać sumę elementów wektora  $x$  do wektora  $y$ .  
Wprowadzić:  
`>>suma_x=sum(x)`  
Obliczyć w pamięci sumę elementów  $x$  i porównać z wynikiem w MATLABIE.  
Funkcja `>>sum(x)` sumuje elementy wektora lub macierzy. Wprowadzić:  
`>>wynik_sumowania=suma_x+y`  
Dodawanie liczby (zmienna `suma_x`) do wektora powoduje dodanie tej liczby do każdego elementu wektora. Obliczyć w pamięci sumę wartości zmiennej `suma_x` i elementów wektora  $y$ . Porównać wynik z wartością zmiennej `wynik_sumowania`.
- (c) Podnieść każdy element wektora  $x$  do potęgi określonej przez odpowiadający mu element  $y$ .  
Wprowadzić:  
`>>wynik_potegowania=x.^y`  
W nazwach zmiennych utworzonych w przestrzeni roboczej programu MATLAB, nie może być znaków diakrytycznych (również polskich —  $\text{ą}$ ,  $\text{ę}$ ,  $\text{ź}$  itp.). Do potęgowania „element po elemencie” wykorzystuje się operator (^) poprzedzony kropką. Taki operator nazywany jest operatorem tablicowym.
- (d) Podzielić każdy element w  $x$  przez odpowiadający mu element w  $y$ .  
Wprowadzić:  
`>>wynik_dzielenia=x./y`  
Do dzielenia „element po elemencie” wykorzystuje się operator (/) poprzedzony kropką. Taki operator nazywany jest operatorem tablicowym.
- (e) Pomnożyć każdy element w  $x$  przez odpowiadający mu element w  $y$  a wyniki zapisać w wektorze  $z$ .  
Wprowadzić:  
`>>z=x.*y`  
Do mnożenia „element po elemencie” wykorzystuje się operator (\*) poprzedzony kropką. Taki operator nazywany jest operatorem tablicowym.
- (f) Zsumować elementy wektora  $z$  i przypisać wynik do zmiennej  $w$ .  
Wprowadzić:  
`>>w=sum(z)`
- (g) Wprowadzić:  
`>>x'*y-w`  
Zwrócić uwagę na brak kropki przy operatorze (\*). Nie jest to operator tablicowy. Zinterpretować wynik biorąc pod uwagę wcześniejsze etapy obliczeń
- (h) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (i) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

7. Obliczanie wartości wyrażeń z wykorzystaniem wektorów. Mając dany wektor  $t$  złożony z elementów o wartościach od 1 do 2 z krokiem 0.2. W celu realizacji zadania wykonać poniższe polecenia.

- (a) Wprowadzić:  
`>>t=1:0.2:2` Utworzenie wektora wierszowego do wykorzystania w dalszych obliczeniach.
- (b) Wprowadzić:  
`>>log(2+t+t.^2)`  
 Polecenie oblicza wartość wyrażenia:  $\ln(2 + t + t^2)$ . Wewnątrz nawiasów zwykłych wykorzystano operator potęgowania tablicowego ponieważ jednym z argumentów jest wektor  $\mathbf{t}$  i potęgowanie dotyczy poszczególnych elementów tego wektora. UWAGA: Do obliczenia wartości logarytmu naturalnego zmiennej  $x$ , w języku MATLAB służy funkcja  $\log(x)$ .
- (c) `>>exp(t).*(1+cos(3.*t))`  
 Polecenie oblicza wartość wyrażenia:  $e^t \cdot (1 + \cos 3t)$ . Zastosowano operator mnożenia tablicowego ponieważ lewa i prawa strona iloczynu funkcji wykładniczej i wyrażenia w nawiasie są wektorami, i mnożenie zachodzi „element po elemencie”.
- (d) `>>cos(t).^2+sin(t).^2`  
 Polecenie oblicza wartość wyrażenia:  $\cos^2 3t + \sin^2 3t$ . Zastosowano operator potęgowania tablicowego ponieważ wartości zwracane przez funkcje trygonometryczne to wektory.
- (e) Wprowadzić:  
`>>acot(t)` Polecenie oblicza wartość funkcji:  $\cot^{-1}$ , czyli funkcji  $\text{arcctg}(t)$ . UWAGA: Argumenty funkcji trygonometrycznych w języku MATLAB podaje się w radianach. Wartości odwrotnych funkcji trygonometrycznych są zwracane w radianach.
- (f) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (g) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

8. Utworzenie wektora elementów spełniających następującą zależność:

$$x_n = \frac{(-1)^{n+1}}{2n-1}, \quad n = 1, 2, \dots$$

- (a) Wprowadzić:  
`>>n=1:100`  
 Polecenie tworzy wektor elementów o wartościach od 1 do 100 z krokiem 1. Jest to wektor indeksów (numerów) elementów szeregu.
- (b) Wprowadzić:  
`>>((-1).^(n+1))./(2.*n-1)`  
 Polecenie oblicza wartość każdego elementu szeregu na podstawie powyższego równania. Ważnym elementem wyrażenia języka MATLAB są nawiasy zwykłe. UWAGA: do ustalenia kolejności działań w wyrażeniach języka MATLAB używa się tylko nawiasów zwykłych. Nawiasy klamrowe i kwadratowe mają specjalne znaczenie, nie związane z wykonywaniem działań arytmetycznych.



- (c) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (d) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

9. Przetwarzanie macierzy o następującej postaci:

$$A = \begin{bmatrix} 2 & 4 & 1 \\ 6 & 7 & 2 \\ 3 & 5 & 9 \end{bmatrix},$$

z wykorzystaniem notacji dwukropkowej. W celu realizacji zadania wykonać poniższe polecenia.

- (a) Wprowadzić:  
`>>A=2 4 1; 6 7 2; 3 5 9`  
Polecenie tworzy w przestrzeni roboczej macierz zadaną powyższym równaniem. Tak, jak to opisano w poprzednich punktach, poszczególne wiersze oddzielone są średnikami a kolumny odstępami. Utworzona macierz ma rozmiar  $3 \times 3$ .
- (b) Wprowadzić:  
`>>x1=A(1, :)`  
Polecenie tworzy wektor wierszowy `x1` przypisując do niego pierwszy wiersz macierzy `A`. Dwukropek występujący w miejscu numeru kolumny, oznacza wybranie wszystkich kolumn pierwszego wiersza (czyli cały pierwszy wiersz).
- (c) Wprowadzić:  
`>>y3=A(:, 3)`  
Polecenie tworzy wektor kolumnowy `y3` przypisując do niego trzecią kolumnę macierzy `A`. Dwukropek występujący w miejscu numeru wiersza oznacza wybranie wszystkich wierszy trzeciej kolumny (czyli całą trzecią kolumnę). Wprowadzić:  
`>>x23=A(2:3, :)`  
Polecenie tworzy macierz `x23` o rozmiarze  $2 \times 3$ . W celu wybrania drugiego i trzeciego wiersza zamiast indeksu wybrano zakres za pomocą dwukropka. `2 : 3` w miejscu numeru wiersza oznacza wiersze od drugiego do trzeciego. Dwukropek w miejscu indeksu kolumny oznacza wszystkie kolumny. Z macierzy `A` wybierane są wszystkie kolumny drugiego i trzeciego wiersza.
- (d) Wprowadzić:  
`>>y12=A(:, 1:2)`  
Polecenie tworzy macierz `y12` o rozmiarze  $3 \times 2$ . W celu wybrania elementów macierzy `A` zastosowano notację dwukropkową, analogicznie jak w poprzednim podpunkcie. Porównać wynik działania polecenia z wynikiem uzyskanym w poprzednim podpunkcie.
- (e) Wprowadzić:  
`>>y13=A(:, [1 3])`

Polecenie tworzy macierz `y13` o rozmiarze  $3 \times 2$ . Macierz ta zbudowana jest z elementów wszystkich wierszy oraz pierwszej i trzeciej kolumny macierzy `A`.

(f) Wprowadzić:

```
>>sk=sum(A)
```

Polecenie oblicza sumy elementów leżących w kolumnach (sumuje po kolumnach) macierzy `A` i umieszcza je w wektorze wierszowym `sk` o rozmiarze  $1 \times 3$ . Przeliczyć ręcznie sumy elementów w wierszach macierzy `A`. Porównać wyniki z uzyskanymi w MATLAB'ie.

(g) Wprowadzić:

```
>>sw=sum(A')
```

Polecenie oblicza sumy elementów leżących w wierszach (sumuje po wierszach) macierzy `A` i umieszcza je w wektorze kolumnowym `sk` o rozmiarze  $3 \times 1$ . Przeliczyć ręcznie sumy elementów w kolumnach macierzy `A`. Porównać wyniki z uzyskanymi w MATLAB'ie.

(h) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.

(i) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:

```
>>clc
```

10. Sprawdzenie poprawności wyrażeń języka MATLAB dla następujących zmiennych:

$$x = \begin{bmatrix} 1 & 4 & 8 \end{bmatrix}, \quad y = \begin{bmatrix} 2 & 1 & 5 \end{bmatrix}, \quad A = \begin{bmatrix} 3 & 1 & 6 \\ 5 & 2 & 7 \end{bmatrix}.$$

Zaobserwować wyniki działania poniższych poleceń. Czy w wyniku wykonania poleceń wystąpiły błędy? Wyjaśnić przyczyny powstania ewentualnych błędów. Do wyjaśnienia wykorzystać polecenie `>>whos`.

(a) Wprowadzić:

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

Polecenia tworzą zadane zmienne w przestrzeni roboczej.

(b) Wprowadzić:

```
>>x+y
```

(c) Wprowadzić:

```
>>x+A
```

(d) Wprowadzić:

```
>>x'+y
```

(e) Wprowadzić:

```
>>A-[x' y']
```

(f) Wprowadzić:

```
>>[x; y']
```

(g) Wprowadzić:

```
>>[x; y]
```

- (h) Wprowadzić:  
`>>A-3`
- (i) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (j) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

11. Wyjaśnienie wyników działania poleceń języka MATLAB operujących na macierzy:

$$A = \begin{bmatrix} 2 & 7 & 9 & 7 \\ 3 & 1 & 5 & 6 \\ 8 & 1 & 2 & 5 \end{bmatrix}.$$

W celu realizacji zadania wykonać poniższe polecenia.

- (a) Wprowadzić:  
`>>A=[2 7 9 7; 3 1 5 6; 8 1 2 5]`  
Polecenie tworzy zmienną A w przestrzeni roboczej.
- (b) Wprowadzić:  
`>>A(:, [1 4])`
- (c) Wprowadzić:  
`>>A([2 3], [3 1])`
- (d) Wprowadzić:  
`>>reshape(A,2,6)`
- (e) Wprowadzić:  
`>>A(:)`
- (f) Wprowadzić:  
`>>fliplr(A)`
- (g) Wprowadzić:  
`>>[A;A(end,:)]`
- (h) Wprowadzić:  
`>>sum(A,2)`
- (i) `>>[[A; sum(A)] [sum(A,2); sum(A(:))]]`
- (j) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (k) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

12. Przetwarzanie macierzy o następującej postaci:

$$A = \begin{bmatrix} 2 & 7 & 9 & 7 \\ 3 & 1 & 5 & 6 \\ 8 & 1 & 2 & 5 \end{bmatrix}.$$

W celu realizacji zadania wprowadzić poniższe polecenia.

- (a) Wprowadzić:  
`>>A=[2 7 9 7; 3 1 5 6; 8 1 2 5]`  
Polecenie tworzy zmienną  $A$  w przestrzeni roboczej.
- (b) Wprowadzić:  
`>>x=A(2, :)`  
Polecenie przypisuje do wektora wierszowego  $x$  drugi wiersz macierzy  $A$ .
- (c) Wprowadzić:  
`>>b=A(:, [2 4])`  
Polecenie przypisuje kolumny macierzy  $A$  o parzystych indeksach do macierzy  $b$
- (d) Wprowadzić:  
`>>c=reshape(A,4,3)`  
Polecenie dokonuje macierzy  $A$
- (e) Wprowadzić:  
`>>1./A`  
Polecenie oblicza odwrotności każdego elementu macierzy  $A$ .
- (f) Wprowadzić:  
`>>sqrt(A)`  
Polecenie oblicza pierwiastek kwadratowy każdego elementu macierzy  $A$ .
- (g) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (h) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

13. Tworzenie macierzy, której pierwsza i piąta kolumna to zera a trzecia i czwarta to jedynki. Do utworzenia macierzy wykorzystać funkcje `>>ones(...)` i `>>zeros(...)`. Każdą z funkcji można wykorzystać tylko raz.

- (a) Wprowadzić:  
`>>A=zeros(5,5)`  
Tworzy macierz o rozmiarze  $5 \times 5$  wypełnioną zerami przy pomocy funkcji `>>zeros(...)`.
- (b) Wprowadzić:  
`>>A(:, [3 4])=A(:, [3 4])+ones(5,2)`  
Dodaje do elementów macierzy  $A$  leżących w trzeciej i czwartej kolumny macierz o rozmiarze  $5 \times 2$ , wypełnioną jedynkami, utworzoną przy pomocy funkcji `>>ones(...)`.
- (c) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (d) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

14. Rozwiązywanie następującego układu równań liniowych:

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

Powyższy układ można zapisać macierzowo:  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ , gdzie:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 2 & 6 & 6 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 14 \\ 20 \end{bmatrix}.$$

W takim przypadku, rozwiązanie układu, dane jest jako:  $\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}$ .

(a) Wprowadzić:

```
>>A=[1 2 3; 3 4 5; 2 6 6]; b=[4; 14; 20]
```

Polecenie tworzy macierz  $\mathbf{A}$  o rozmiarze:  $3 \times 3$  oraz kolumnowy wektor  $\mathbf{b}$  o rozmiarze  $3 \times 1$ .

(b) Wprowadzić:

```
>>x=A\b
```

Polecenie rozwiązuje zadany układ równań za pomocą operatora dzielenia w lewo. Ponieważ nie jest możliwe dzielenie wektora przez macierz, operator odwraca macierz  $\mathbf{A}$  a następnie mnoży ją lewostronnie przez wektor  $\mathbf{b}$ . Daje to rozwiązanie opisane powyższym równaniem macierzowym.

(c) Wprowadzić:

```
>>x=inv(A)*b
```

Polecenie rozwiązuje zadany układ równań bezpośrednio na podstawie powyższej zależności. Funkcja `>>inv(A)` odwraca macierz  $\mathbf{A}$ , następnie wynik mnożony jest przez wektor  $\mathbf{b}$ .

(d) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.

(e) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:

```
>>clc
```

15. Utworzenie i przetwarzanie macierzy elementów o wartościach losowych, o rozkładzie normalnym.

(a) Wprowadzić:

```
>>randn('seed',123456789)
```

```
>>F=randn(5,10)
```

Pierwsze z poleceń inicjuje wartość zarodka generatora pseudo-losowego języka MATLAB. Generowanie liczb pseudolosowych polega na tworzeniu nowego elementu ciągu liczbowego poprzez wykonanie pewnych operacji na elemencie poprzednim. Pierwsze z poleceń inicjuje wartość elementu początkowego. Drugie z poleceń tworzy macierz o rozmiarze:  $5 \times 10$ , wypełnioną wartościami losowymi o rozkładzie normalnym, zerowej wartości oczekiwanej i jednostkowej wariancji.

- (b) Wprowadzić:  
`>>avg_k=mean(F)`  
Polecenie oblicza wartości średnie w każdej z kolumn macierzy F i umieszcza je w wektorze wierszowym `avg_k`.
- (c) Wprowadzić:  
`>>avg_w=mean(F')`  
Polecenie oblicza wartości średnie w każdym z wierszy macierzy F i umieszcza je w wektorze kolumnowym `avg_w`.
- (d) Wprowadzić:  
`>>std_k=std(F)`  
Polecenie oblicza wartości odchyłeń standardowych elementów każdej kolumny macierzy F i umieszcza je w wektorze kolumnowym `std_k`.
- (e) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (f) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

16. Przetwarzanie macierzy reprezentującej oceny punktowe pięciu studentów (kolumny) z czterech testów (wiersze). Macierz ma postać:

$$A = \begin{bmatrix} 89 & 97 & 55 & 72 & 95 \\ 100 & 92 & 63 & 85 & 91 \\ 82 & 96 & 71 & 91 & 82 \\ 90 & 98 & 48 & 83 & 70 \end{bmatrix}$$

- (a) Wprowadzić:  
`>>A=[89 97 55 72 95; 100 92 63 85 91; 82 96 71 91 82; ...  
90 98 48 83 70]`  
Polecenie tworzy macierz z ocenami punktowymi studentów.
- (b) Wprowadzić:  
`>>A_s=sort(A)`  
Polecenie sortuje macierz A po kolumnach i przypisuje wynik do macierzy `A_s`.
- (c) Wprowadzić:  
`>>mean(A)`  
Polecenie oblicza średnią punktową każdego studenta. Który student uczy się najlepiej?
- (d) Wprowadzić:  
`>>median(A)`  
Polecenie oblicza medianę ocen każdego studenta. Mediana to wartość środkowa uporządkowanego ciągu liczb. W przypadku parzystej ciągu o parzystej liczbie elementów jest ona równa średniej dwóch środkowych elementów.
- (e) Wprowadzić:  
`>>mean(mean(F))`  
Polecenie oblicza ogólną średnią punktową wszystkich pięciu studentów.

- (f) Skopiować zawartość okna poleceń programu MATLAB do programu Wordpad.
- (g) Wyczyścić zawartość okna poleceń programu MATLAB poleceniem:  
`>>clc`

#### 4. Opracowanie sprawozdania

W sprawozdaniu należy umieścić polecenia oraz wyniki ich działania skopiowane w trakcie ćwiczenia z okna środowiska MATLAB. Do Każdej linii kodu oraz do każdego wyniku, należy dodać komentarz objaśniający.

##### **Przykład**

`...2+round(6/9+3*2)/2-3` — obliczenie wartości wyrażenia. Funkcja `round(6/9+3*2)` zaokrągla wynik działania `6/9+3*2` do najbliższej liczby całkowitej...