Złożone struktury danych
- tablice jedno i wielowymiarowe

1. Macierze jako przykład tablicy wielowymiarowej

Tablica jest jednorodną strukturą danych ponieważ składa się z elementów tego samego typu. Dostęp do danych zawartych w tablicy jest swobodny co oznacza, że wszystkie elementy mogą być wybrane w dowolnej kolejności i są jednakowo dostępne. W celu wybrania pojedynczego elementu stosuje się indeks (numer elementu w tablicy).

Macierz jest przykładem tablicy dwuwymiarowej. W systemie obliczeniowym MatLab macierze tworzy się wierszami oddzielając kolejne elementy spacjami lub przecinkami. Kolejne wiersze zaś oddziela się średnikami lub znakami powrotu karetki. Cała macierz musi być ujęta w kwadratowe nawiasy. Elementami macierzy mogą być liczby rzeczywiste, zespolone lub poprawne wyrażenia MatLaba.

1.1 „Ręczne” tworzenie macierzy

Przykład 1

„Ręczne” tworzenie macierzy:

- Oddzielanie kolumn macierzy za pomocą spacji i wierszy za pomocą średnika:

```
M1 = [1 2 3; 4 5 6; 7 8 9]
```

- Wynik działania:

```
M1 =
     1     2     3
     4     5     6
     7     8     9
```

- Oddzielanie kolumn macierzy za pomocą przecinków i wierszy za pomocą znaku powrotu karetki:

```
M2 = [0.1,0.2,0.3
0.4,0.5,0.6
0.7,0.8,0.9]
```

- Wynik działania:

```
M2 =
    0.1000    0.2000    0.3000
    0.4000    0.5000    0.6000
    0.7000    0.8000    0.9000
```

- Wypełnianie macierzy poprawnymi wyrażeniami MatLaba:

```
alfa = 120;
beta = 30;
gamma = 30;
M3 = [alfa, beta, gamma; cosd(alfa), cosd(beta), cosd(gamma); alfa+...
beta+gamma,cosd(alfa)*cosd(beta)*cosd(gamma),0]
```

```
% trzy kropki ... oznaczają kontynuację polecenia w nowej linii
```

▪ Wynik działania:

```
M3 =
  120.0000   30.0000   30.0000
   -0.5000    0.8660    0.8660
  180.0000   -0.3750         0
```

1.2 „Automatyczne” tworzenie macierzy

Proces generowania macierzy można wspomóc stosując funkcje wbudowane, które tworzą macierze narzędziowe. Poniżej przedstawiono kilka podstawowych funkcje generujących macierze narzędziowe:

- *eye*(*m*,*n*) – zwraca macierz o wymiarach *m* na *n* wypełnioną zerami, a na przekątnej jedynkami;
- *zeros*(*m*,*n*) – zwraca macierz o wymiarach *m* na *n* wypełnioną zerami;
- *ones*(*m*,*n*) – zwraca macierz o wymiarach *m* na *n* wypełnioną jedynkami;
- *rand*(*m*,*n*) – zwraca macierz o wymiarach *m* na *n* wypełnioną liczbami pseudolosowymi o rozkładzie równomiernym z przedziału od 0 do 1;
- *randn*(*m*,*n*) – zwraca macierz o wymiarach *m* na *n* wypełnioną liczbami pseudolosowymi o rozkładzie normalnym z przedziału od -3 do 3;
- *randi* – zwraca macierz wypełnioną liczbami całkowitymi pseudolosowymi.

2. Wektor jako przykład tablicy jednowymiarowej

Wektor jest przykładem tablicy jednowymiarowej. W systemie obliczeniowym MatLab wektor jest specjalnym przypadkiem macierzy i zawiera tylko jeden wiersz lub kolumnę. Elementami wektora mogą być liczby rzeczywiste, zespolone lub poprawne wyrażenia MatLaba.

2.1 „Ręczne” tworzenie wektora

Przykład 2

Tworzenie wektora wierszowego w systemie obliczeniowym MatLab:

▪ „Ręczne” tworzenie wektora v_1 (oddzielanie kolejnych elementów kolumn spacją):

```
v1 = [1 2 3 4 5 6 7 8 9]
```

▪ Wynik działania:

```
v1 =
     1     2     3     4     5     6     7     8     9
```

▪ „Ręczne” tworzenie wektora v_2 (oddzielanie kolejnych elementów kolumn przecinkiem):

```
v2 = [1,2,3,4,5,6,7,8,9]
```

▪ Wynik działania:

```
v2 =
     1     2     3     4     5     6     7     8     9
```

Przykład 3

Tworzenie wektora kolumnowego w systemie obliczeniowym MatLab:

▪ „Ręczne” tworzenie wektora v_3 (oddzielanie kolejnych elementów wierszy średnikiem):

```
alfa = 15;
```

```
beta = 30;
v3 = [alfa;beta;alfa/beta;cosd(alfa+beta)]
```

▪ Wynik działania:

```
v3 =
   15.0000
   30.0000
    0.5000
    0.7071
```

2.2 „Automatyczne" – tworzenie wektora

W przypadkach gdy należy utworzyć wektor składający się z liczb z danego zakresu z określonym krokiem można wykorzystać następującą składnię polecenia w MatLabie:

wektor = *wartość początkowa*:*krok*:*wartość końcowa*

Wszystkie trzy wartości w powyższej składni polecenia tworzenia wektora mogą być poprawnymi wyrażeniami arytmetycznymi. Jeśli krok nie zostanie wskazany, wtedy MatLab przyjmuje domyślną jego wartość równą 1.

Ćwiczenie 1

W systemie obliczeniowym MatLab wygenerować następujący wektor wierszowy v_4 = [0,2,4,...,64].

▪ Polecenie tworzenia wektora v_4:

```
v4 = 0:2:64
```

▪ Wynik działania:

```
v4 =
  Columns 1 through 11
     0     2     4     6     8    10    12    14    16    18    20
  Columns 12 through 22
    22    24    26    28    30    32    34    36    38    40    42
  Columns 23 through 33
    44    46    48    50    52    54    56    58    60    62    64
```

System obliczeniowy MatLab posiada również wbudowane funkcje, które umożliwiają tworzenie wektorów specjalnych. Bardzo użytecznymi funkcjami wspomagającymi tworzenie wektorów są:

- *linspace*(*a*,*b*,*n*) – tworzy wektor liniowy o długości *n* w zakresie od *a* do *b*, (funkcja zwraca taki sam wektor jak zapis: *v*=*a*:(*b*-*a*)/(*n*-1):*b*;
- *logspace*(*a*,*b*,*n*) – tworzy wektor logarytmiczny o długości *n* w zakresie od 10^a do 10^b;
- *randperm*(*n*) – tworzy wektor o długości *n* , wypełniony losowo bez powtórzeń liczbami naturalnymi z zakresu od 1 do *n*.

Wektory specjalne, na przykład wektory o określonej długości, złożone z zer lub jedynek, można tworzyć przy użyciu funkcji macierzy narzędziowych (punkt 1.2) np. *zeros, ones, itp.*

Przykład 4

Tworzenie wektora specjalnego z wykorzystaniem funkcji macierzy narzędziowych:

▪ Inicjacja 100-elementowego wektora wierszowego v_5 składającego się z jedynek:

```
v5 = ones(1,100);
```

▪ Inicjacja 10-elementowego wektora kolumnowego v_6 składającego się z zer:

```
v6 = zeros(10,1);
```

3. Indeksowanie

Dostęp do elementów macierzy następuje poprzez wskazanie wiersza i kolumny. I tak wyrażenie $A(i,j)$ w MatLabie znaczy odwołanie do elementu a_{ij} *macierzy A*, znajdującego się w wierszu *i* oraz kolumnie *j*. Zapis taki jest powszechnie stosowany w programach obliczeniowych i językach programowania. Jednakże MatLab zapewnia większe możliwości indeksowania – pozwala bowiem na definiowanie zakresów wierszy i kolumn. Na przykład wyrażenie $A(m{:}n,k{:}l)$ wskazuje wiersze od *m* do *n* i kolumny od *k* do *l* macierzy *A*. Jeśli zakres ma obejmować wszystkie wiersze (lub kolumny) , wówczas wystarczy użyć znaku dwukropka jako indeksu. Dlatego *A*(:,5:20) oznacza odwołanie do kolumny od 5 do 20 oraz wszystkie wiersze macierzy *A*.

Przykład 5

Przykład indeksowania:

▪ Wybór jednego elementu z macierzy:

```
M = [1 2 3;4 5 6;7 8 9]
a22 = M(2,2) %wybranie drugiego wiersza i drugiej kolumny z macierzy M
```

▪ Wynik działania:

```
M =
     1     2     3
     4     5     6
     7     8     9
a22 =
     5
```

▪ Wybór zakresu elementów z macierzy:

```
w1 = M(:,2) %wybranie wszystkich wierszy z drugiej kolumny macierzy M
M2 = M(:,[1 3]) %wybranie wszystkich wierszy z pierwszej i trzeciej
kolumny macierzy M
M3 = M(1:2,1:2) %wybranie od pierwszego do drugiego wiersza i od
pierwszej do drugiej kolumny macierzy M
```

▪ Wynik działania:

```
w1 =
     2
     5
     8
M2 =
     1     3
     4     6
     7     9
M3 =
     1     2
     4     5
```

▪ Wybór jednego elementu wektora:

```
a3 = w1(3) %wybranie trzeciego elementu wektora w1
```

▪ Wynik działania:

```
a3 =
     8
```

MatLab automatycznie określa wymiary macierzy, co oznacza, że nie trzeba ich jawnie deklarować. Wymiary istniejącej macierzy A można odczytać za pomocą polecenia *size*(A) lub ściślej [m,n] = *size*(A) i wówczas liczba wierszy oraz liczba kolumn przypisane są zmienianym, odpowiednio, m oraz n. Długość wektora v (liczbę jego elementów) można sprawdzić za pomocą polecenia *length*(v).

4. Działania arytmetyczne na macierzach

MatLab pozwala w łatwy sposób wykonywać operacje na macierzach, o ile tyko maja one sens pod względem matematycznym, a argumenty są kompatybilne.

Dlatego też działanie:

- A+B lub A-B jest poprawne, jeśli macierz A i B mają ten sam rozmiar;
- A*B jest poprawne, jeśli liczba kolumn macierzy A jest równa liczbie wierszy macierzy B;
- A/B jest poprawne i dla macierzy o tej samej wielkości daje A*B^{-1};
- A^2 jest równoznaczne A*A i ma sens tylko wówczas, gdy macierz A jest kwadratowa.

Gdyby we wszystkich powyższych poleceniach B zastąpić skalarem na przykład b, to działania arytmetyczne nadal byłyby możliwe do przeprowadzenia. W takim przypadku polecenie A+b spowodowałoby dodanie b do każdego elementu A, polecenie A*b (lub b*A) mnożyłoby każdy element A przez b i tak dalej.

Z tego względu, że wektory są traktowane jako jednowierszowe lub jednokolumnowe macierze to polecenie takie jak $w = u * v$, gdzie u oraz v, są wektorami o tej samej długości, np. m na 1, wygenerowało by błąd (ponieważ nie można mnożyć macierzy o wymiarach m na 1 przez macierz o tych samych wymiarach). Natomiast działania $w = u * v'$ oraz $w = u' * v$ wykonane zostałyby poprawnie i zwróciłyby iloczyn skalarny dwóch wektorów.

<u>Przykład 6</u>

Mnożenie macierzowe dwóch wektorów:

▪ Tworzenie dwóch wektorów pięcioelementowych (u_1 – wektor kolumnowy, u_2 – wektor wierszowy) oraz operacja mnożenia macierzowego, w wyniku której powstaje macierz M_1 o wymiarach 5 na 5:

```
u1 = [2 5 7 19 23]' %transpozycja wektora u1
u2 = linspace(-5,5,5)
M1 = u1*u2
```

▪ Wynik działania:

```
u1 =
     2
     5
     7
    19
    23
```

```
u2 =
   -5.0000   -2.5000         0    2.5000    5.0000
M1 =
  -10.0000   -5.0000         0    5.0000   10.0000
  -25.0000  -12.5000         0   12.5000   25.0000
  -35.0000  -17.5000         0   17.5000   35.0000
  -95.0000  -47.5000         0   47.5000   95.0000
 -115.0000  -57.5000         0   57.5000  115.0000
```

Oprócz dzielenia zwykłego, czyli prawostronnego (/), MatLab pozwala na wykonywanie dzielenia lewostronnego (\). Dzielenie takie stosowane jest w równaniach macierzowych. Za pomocą polecenia $x = A\backslash b$ można rozwiązać równanie $Ax = b$.

Ćwiczenie 2

Rozwiązać układ równań liniowych: $\begin{cases} x + y = 32 \\ 2x - 4y = -2 \end{cases}$

▪ Rozwiązanie układu równań liniowych z wykorzystaniem operatora dzielenia lewostronnego i wyświetlenie wyniku w czytelny sposób w przestrzeni *Command Window*:

```
A = [1 1;2 -4];
b = [32; -2];
x = A\b;
clc
disp('Rozwiązanie układu równań:')
disp(['x=' num2str(x(1)) ', y=' num2str(x(2))])
```

▪ Wynik działania:

```
Rozwiązanie układu równań:
x=21, y=11
```

5. Działania arytmetyczne na tablicach

Działania arytmetyczne przeprowadzane na tablicach pozwalają wykonywać określoną operację arytmetyczną kolejno na wszystkich elementach. Mnożenie, dzielenie i potęgowanie elementu przez element pomiędzy dwiema macierzami lub wektorami o tych samych rozmiarach wykonuje się poprzez wstawienie stosowanego operatora arytmetycznego, poprzedzonego kropką:

- .* - mnożenie elementu przez element;
- ./ - dzielenie elementu przez element;
- .\ - lewostronne dzielenie elementu przez element;
- .^ - potęgowanie elementu przez element

Przykład 7

Mnożenie tablicowe dwóch wektorów wierszowych w systemie obliczeniowym MatLab:

▪ Tworzenie wektora v_8 po przez wykonanie operacji mnożenia tablicowego na wektorach v_6 i v_7:

```
v6 = 1:10;
v7 = 2:2:20;
v8 = v6.*v7
```

▪ Wynik działania:

```
v8 =
     2     8    18    32    50    72    98   128   162   200
```

6. Zadania do samodzielnego wykonania

Wytyczna 1:

Przed przystąpieniem do rozwiązywania zadań należy:

1. Utworzyć nowy folder i nazwać go (*imię_nazwisko_Lab_numer*);
2. Do utworzonego katalogu skopiować wzorzec raportu dołączony do instrukcji, zamienić jego nazwę według wzorca podanego w punkcie 1;
3. Otworzyć raport i wypełnić go swoimi danymi;
4. Uruchomić środowisko obliczeniowe MatLab;
5. W środowisku obliczeniowym MatLab w oknie *Current Directory* lub *Current Folder* (zależy od wersji systemu obliczeniowego) zlokalizować katalog utworzony w punkcie 1.
6. Wszystkie pliki powiązane z zadaniami należy zapisywać w tej lokalizacji.

Wytyczna 2:

Przed przystąpieniem do każdego z zadań należy wykonać następujące czynności:

1. Wyczyścić przestrzeń *Workspace* ze wszystkich zmiennych (polecenie: *clear)*;
2. Wyczyścić przestrzeń *Command Window* ze wszystkich poleceń (polecenie: *clc)*;

Wytyczna 3:

Wytyczna dotyczy osób pracujących na stanowiskach komputerowych zapewnionych przez Uczelnię. Po ocenieniu przez prowadzącego zajęcia należy:

1. Zarchiwizować wyniki swoich prac na własnym zewnętrznym nośniku danych, lub przesyłając je na własny adres e-mail, czy dysk działający w chmurze bądź w inny legalny sposób;
2. Trwale usunąć z przestrzeni dyskowej katalog utworzony według wytycznej 1 w punkcie 1.

<u>Zadanie 1</u>

Wykorzystać okno poleceń *Command Window* i utworzyć następujące zmienne:

1. Wektor kolumnowy $v_1 = (2, 12, 45, 3, 5, 7, 8, 12, 0, 6)$;
2. Wektor wierszowy v_2 = (0, 3, 6,...,30);
3. Wektor kolumnowy v_3 = (1, 2, 3,...,100);
4. Stuelementowy wektor wierszowy v_4 wypełniony elementami z zakresu od -2π do 2π;
5. Wektor v_5 wypełniony losowo bez powtórzeń liczbami naturalnymi z zakresu 1 do 100;
6. Macierz $M_1 = \begin{bmatrix} 3 & 4 & 8 \\ 0 & 6 & 9 \\ 3 & 7 & 1 \end{bmatrix}$;
7. Macierz $M_2 = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix}$;
8. Macierz M_3 o wymiarach 10 na 12 wypełnioną elementami pseudolosowymi;
9. Macierz $M_4 = \begin{bmatrix} \cos(45°) & -\sin(45°) & 0 \\ \sin(45°) & \cos(45°) & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Polecenia i wyniki ich działań zamieścić w raporcie.

Zadanie 2

Wygenerować macierz M o wymiarze 10 na 10, która reprezentuje tabliczkę mnożenia rys 1. Prawidłowo utworzoną macierz M zapisać do pliku.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Rys. 1. Macierz o wymiarze 10 na 10, tabliczka mnożenia

Polecenia i wyniki ich działań zamieścić w raporcie.

Wskazówka 1:

Aby wygenerować macierz M należy wykonać odpowiednie operacje macierzowe.

Zadanie 3

Wczytać do przestrzeni *Workspace* macierz M zapisaną do pliku w zadaniu 2 i stosując odpowiednie indeksowanie utworzyć następujące zmienne:

1. w_1 (zmienna ma zawierać całą piątą kolumnę macierzy M);
2. w_2 (zmienna ma zawierać cały pierwszy wiersz macierzy M);
3. M_2(zmienna ma zawierać elementy macierzy M od 3 do 6 wiersza i od 6 do 9 kolumny).
4. M_3(zmienna ma zawierać cały pierwszy, trzeci, szósty i dziewiąty wiersz macierzy M).

Polecenia i wyniki ich działań zamieścić w raporcie.

Zadanie 4

Równanie linii prostej ma postać $y = ax+b$, gdzie wartość a oraz b są stałe. Napisać skrypt, który będzie obliczał współrzędne y linii przy nachyleniu a = 0,2 oraz b = -4 dla następujących wartości współrzędnych x:

x = -5, -3, 0, 1.25, 3, 4, 5, 6, 7, 8, 9, 10.

Skrypt i wynik jego działań zamieścić w raporcie.

Wskazówka 2:

Polecenia nie powinny zawierać żadnych operatorów tablicowych, ponieważ obliczenie obejmuje mnożenie wektora przez skalar *a,* a następnie dodawanie kolejnego skalara *b.*

Zadanie 5

Utworzyć wektor t o dwudziestu elementach od 1, 2, 3,...,20. Następnie obliczyć:

- $x = tsin(t)$;
- $y = \frac{t-1}{t+1}$;
- $z = \frac{sin(t^2)}{t^2}$.

Polecenia i wyniki ich działań zamieścić w raporcie.

Zadanie 6

Napisać skrypt, który będzie rozwiązywał następujący układ równań liniowych:

$$\begin{cases} v + 2x - 5y - z = -1009 \\ 2v - 3x + y + 5z = 120 \\ v - x + y - z = 11 \\ 3v + 4x - 9y + 7z = 333 \end{cases},$$

i wyświetlał uzyskane wynik w czytelny sposób w przestrzeni *Command Window.*

Rozwiązanie układu równań: v=324.0909, x=622.7273, y=481.3636, z=171.7273.

Skrypt i wynik jego działań zamieścić w raporcie.