Obliczenia w programie MATLAB

Na zajęciach korzystamy z programu MATLAB, w którym wykonywać będziemy większość obliczeń. Po uruchomieniu programu w zależności od wersji i konfiguracji może pojawić się kilka okienek:

`Command Window` – najważniejsze, podstawowe okno programu. Służy do komunikacji programu z użytkownikiem. **W tym oknie wprowadzamy polecenia** i w nim program wyświetla wyniki obliczeń, ostrzeżenia i błędy.

`Current Directory` albo **`Current Folder`** – pokazuje zawartość bieżącego katalogu. Ścieżka do bieżącego katalogu pokazywana jest też na pasku pod głównym menu. Pliki tworzone w programie będą domyślnie zapisywane właśnie w tym katalogu. **WAŻNE: Matlab będzie widział tylko te pliki**, które są zapisane w katalogach wskazanych **na jego liście ścieżek** albo **w bieżącym katalogu.**

`Command History` – zawiera listę ostatnio wprowadzanych komend, w porządku chronologicznym.

`Workspace` – pokazuje zmienne aktualnie pamiętane w programie. Można je tam obserwować, a także zmieniać (usuwać).

Pomoc do programu (w języku angielskim) znajdziemy w oknie **`Help`**.

Proszę wybrać z menu programu: **Desktop – Desktop Layout – Command Window Only** lub w wersji R2012 i nowszych: **Home – Layout – Command Window Only**. Spowoduje to zamknięcie wszystkich okienek z wyjątkiem okna poleceń.

Obliczenia na liczbach prowadzimy w oknie poleceń (**`Command Window`**) wpisując konkretne działanie i wciskając klawisz ENTER. Znak **`*`** oznacza mnożenie, znak **`/`** oznacza dzielenie, potęgowanie uzyskujemy wpisując przed wykładnikiem potęgi znak `^`, dodawanie i odejmowanie to **`+`** i **`−`**. Separatorem dziesiętnym jest kropka (nie przecinek!), odpowiednikiem mnożenia przez 10 do potęgi *x* jest dodanie na końcu wartości wyrażenia **`e`***x* (np. 8.85e-12 = $8.85 \cdot 10^{-12}$). Jeśli chcemy policzyć wartość dowolnej, znanej w programie funkcji, jej argument wprowadzamy w nawiasie, np:

```
Command Window
    sin(pi/3)
    cos(5e-2)
```

W odpowiedzi na powyższe komendy Matlab obliczy podane wyrażenie i zapamięta jego wynik w zmiennej `ans` (skrót od *answer* - odpowiedź). Możemy też sami nazywać wyniki, które są obliczane albo **deklarować zmienne**. Zmienne wprowadzamy do programu podając ich nazwę, potem znak równości, a po nim przypisując wartość (i oczywiście wciskając ENTER), np. dla zmiennych jednowyrazowych:

```
Command Window
    a=6
    b = 7*4
    B = 1.2e3+a+b; C = 7.02e-12;
```

Postawienie na końcu linii średnika powoduje, że zmienna jest zapamiętywana, ale nie jest wyświetlana na ekranie w momencie jej wprowadzania. Można jednak zobaczyć wartość każdej zapamiętanej zmiennej wpisując jej nazwę i wciskając ENTER.

Macierze w Matlabie

Zmienne w Matlabie są MACIERZAMI, tzn. są zapamiętywane jako zbiór wartości uporządkowanych w określonej liczbie wierszy i kolumn. W szczególnych przypadkach macierze mogą mieć jeden wiersz i jedną kolumnę – zawierają wtedy tylko jeden wyraz. Macierze o większych rozmiarach można wprowadzać na kilka sposobów, z których dwa podstawowe to:

1) po nazwie macierzy i znaku równości wpisujemy elementy w nawiasach kwadratowych, oddzielając je spacjami lub przecinkami, a wiersze rozdzielamy średnikami:

```
d = [5 3.07 6 4e6;0 1 60 33.5;0 0 0 0];
```

2) po lewej stronie znaku równości, po nazwie zmiennej określamy w nawiasie położenie elementu w macierzy, a po prawej stronie znaku równości podajemy wartość tego elementu:
   `zmienna(wiersz,kolumna) = wartość`:

```
d(1,1) = 5;
d(1,2) = 3.07;
d(1,3) = d(1,1)+1;
d(1,4) = 2*2e6;
d(2,1) = 0;
```

... itd.

Można też od razu wpisać jedną wartość w kilku pozycjach macierzy, np. w wierszu trzecim, w kolumnach od pierwszej do czwartej:

```
d(3,1:4) = 0;
```

Oprócz dwuwymiarowych macierzy i pojedynczych wartości – tzw. skalarów, często będziemy się posługiwać macierzą jednowierszową (lub jednokolumnową) zwaną wektorem, której wartości będą tworzyły ciąg, np. narastający liniowo. Służy do tego funkcja **`linspace`**, w której jako argumenty podajemy **wartość początkową**, **wartość końcową** i liczbę **wyrazów**. Możemy też utworzyć taki wektor podając krok o jaki mają zmieniać się kolejne wyrazy, według schematu:

`wartość_początkowa : krok : wartość_końcowa`, np:

```
e = 0:0.1:2;
E = linspace(0,2,21);
```

Jak łatwo zauważyć, oba wektory **`e`** i **`E`** mają taką samą zawartość.

Zapamiętanymi zmiennymi możemy się posługiwać tak samo jak wartościami liczbowymi, w szczególności możemy na ich podstawie tworzyć nowe zmienne:

```
G = sin(d);
h = d.*G
g = h'
```

Ostatnia z powyższych operacji powoduje transpozycję macierzy **h** i przypisanie jej do nowej zmiennej o nazwie **g**. UWAGA: w przypadku macierzy zawierających wartości zespolone, macierz transponowana będzie zawierać wartości sprzężone. Transpozycję bez sprzężenia wykonuje się dodając kropkę przed apostrofem: `.'`. Wartości zespolone wprowadza się dodając symbol jedności urojonej **j** lub **i**:

```
A = 5 + 7j
d = 0.5 + 7e-2j;
```

Przypisanie zmiennej **d** nowej wartości powoduje skasowanie wszystkich zawartych w niej wcześniej danych. Komenda **clear** powoduje skasowanie wszystkich zmiennych zapamiętanych w programie. Aby skasować tylko niektóre zmienne należy podać ich nazwy. Poniższa komenda powoduje skasowanie tylko zmiennych **a** **A** i **b**:

```
clear a A b;
```

UWAGA. Poniższe komendy:

```
zmienna = 15;
zmienna = zmienna + 1;
```

są całkowicie poprawne w języku MATLAB, mimo że druga z nich wydaje się niepoprawna w sensie matematycznym. Pierwsza komenda tworzy w pamięci zmienną o nazwie **zmienna** i przypisuje jej konkretną wartość. Druga komenda robi dokładnie to samo. Wartość po prawej stronie znaku równości jest obliczana, a następnie zapamiętywana jako zmienna o nazwie **zmienna**.

Nazwy w języku MATLAB

Zmienne używane w programie MATLAB mogą być prawie dowolnie nazywane przez użytkownika, jednak należy przestrzegać kilku prostych zasad.

1) Nazwy zmiennych mogą składać się z ciągu znaków zawierających:

   duże i małe litery   znak podkreślenia „_”   cyfry.

2) Należy pamiętać, że MATLAB rozróżnia duże i małe litery (`aa`, `Aa`, `aA`, `AA` to różne zmienne!).

3) Nazwa zmiennej musi zaczynać się od litery (małej lub dużej).

4) Nie wolno używać spacji ani polskich liter: **ą ć ę ł ń ó ś ź ż** (małych ani dużych).

Takie same zasady obowiązują przy nazywaniu skryptów, a także funkcji, które zapisujemy zawsze w pliku o takiej samej nazwie, jak nazwa funkcji, z rozszerzeniem **.m** (nadawanym automatycznie).

Operacje na macierzach

MATLAB to akronim powstały z angielskich słów ***mat**rix **lab**oratory* (laboratorium macierzowe). Jak sama nazwa wskazuje, program został stworzony do przeprowadzania obliczeń na macierzach. Oprócz standardowych operacji macierzowych znanych z matematyki, umożliwia też szersze wykorzystanie zapisu zmiennych w postaci macierzy, poprzez działania „z kropką”. Te działania to mnożenie, dzielenie i potęgowanie, które w odróżnieniu od zwykłego mnożenia, dzielenia i potęgowania wykonywane jest jednocześnie na wszystkich elementach macierzy, jednak w taki sposób jakby były to zwykłe działania na pojedynczych liczbach (możemy jednym działaniem pomnożyć przez siebie odpowiadające sobie wyrazy w dwóch macierzach). W tym celu zamiast symbolu **`*`** używamy symbolu poprzedzonego kropką **`.*`**. Podobnie można dzielić i potęgować elementy jednej macierzy przez elementy drugiej – używając symboli: **`./`** oraz **`.^`**. Jako przykład rozpatrzymy mnożenie.

`*` Mnożenie macierzy (ang. Matrix multiply)

`X*Y` jest wynikiem mnożenia macierzy `X` i `Y`. Każdy skalar (macierz o rozmiarze 1x1) można mnożyć przez dowolną zmienną. Jeśli `X` i `Y` nie są skalarami, liczba kolumn w `X` musi być równa liczbie wierszy w `Y`. Wynik mnożenia będzie miał tyle wierszy ile ma macierz `X` i tyle kolumn ile ma macierz `Y`.

Przykład:

```
X = [1 2 3;-1 -2 -3]; % dwa wiersze, trzy kolumny
Y = [10 20 -20 -10; 30 40 -40 -30; 50 60 -60 -50]; % trzy wiersze,
                                                      cztery kolumny
Z = X*Y
Z =
   220   280  -280  -220
  -220  -280   280   220
```

Wyrazy w *k*-tym wierszu `X` są mnożone przez wyrazy w *n*-tej kolumnie `Y`. Poszczególne iloczyny są dodawane i w ten sposób powstaje element macierzy `Z`, który jest umieszczany w jej *k*-tym wierszu, w *n*-tej kolumnie. Wyraz `Z(1,2)` powstaje z pierwszego wiersza `X` i drugiej kolumny `Y`:

```
1*20 + 2*40 + 3*60 = 280
```

`.*` Mnożenie wyrazów w macierzach (ang. Array multiply)

`X.*Y` jest wynikiem mnożenia każdego wyrazu w macierzy `X` przez odpowiedni wyraz macierzy `Y`. Macierze `X` i `Y` muszą mieć takie same rozmiary.

```
X = [1 2 3;-1 -2 -3]; % dwa wiersze, trzy kolumny
Y = [10 20 -20; 30 -40 -30]; % dwa wiersze, trzy kolumny

Z = X.*Y
Z =
   10.00    40.00   -60.00
  -30.00    80.00    90.00
```

Z linii poleceń można wywołać okno pomocy i otworzyć je na wybranej stronie poleceniem `doc`. Wpisz w Command Window: `doc *` aby zobaczyć informacje nt. operatorów i znaków specjalnych.

Zadania do samodzielnego wykonania

1) Proszę utworzyć macierz **g** o wymiarach 5x4 (pięć wierszy, cztery kolumny) i określić jakie macierze (o jakim rozmiarze i zawartości) zostaną utworzone w wyniku wykonania kolejnych poleceń.

```
g =[-3   8  -2   1   1
     7  -4   1   6  -1
     2   5  -8   9 -10
     1   5  -9  -7  -3]';

a = g(:,2)

b = g(4,:)

c = g(4:5,1:3)

d = g(1:2:5,:)
```

2) Użyj komendy `sum` aby obliczyć:

a) sumę liczb naturalnych nie większych niż *k*: $A = \sum_{i=1}^{k} i$

b) sumę liczb naturalnych, parzystych, nie większych niż *k*: $B = \sum_{i=1}^{k/2} 2i$

c) sumę liczb naturalnych, nieparzystych, nie większych niż *k*: $C = \sum_{i=1}^{k/2} (2i-1)$

d) sumę iloczynów liczb naturalnych nie większych niż *m*, pomnożonych przez ich pierwiastki: $D = \sum_{i=1}^{m} i\sqrt{i}$. Podaj wynik z dokładnością do trzech miejsc po przecinku.

e) sumę iloczynów liczb naturalnych nie większych niż sto, podzielnych przez 9, pomnożonych przez ich pierwiastki: $E = 9\sqrt{9} + 18\sqrt{18} + \ldots + 99\sqrt{99}$. Podaj wynik w zaokrągleniu do trzech cyfr znaczących, do sześciu cyfr znaczących i do ośmiu cyfr znaczących.

3) Oblicz przykłady z zadania 2 używając pętli **`for`** ... **`end`**, np.:

```
A=0;
for i=1:k,
  A=A+i;
end
```

Zapisz jeden wybrany z powyższych przykład jako funkcję i wypróbuj jej działanie.

Macierz diagonalna, macierz jednostkowa, wyznacznik macierzy

Szczególnym rodzajem macierzy jest **macierz diagonalna** - macierz kwadratowa, której wszystkie współczynniki leżące poza główną przekątną (diagonalą) są równe zeru. Wyznacznik macierzy diagonalnej jest równy iloczynowi wszystkich elementów na głównej przekątnej. Definicja takiej macierzy w Matlabie jest bardzo prosta: tworzy się ją wywołując funkcję `diag` i podając jako argumenty wartości na głównej przekątnej, np:

```
Md = diag([1 2 3 4])
```

Macierz jednostkowa to taka macierz diagonalna, w której wszystkie elementy na głównej przekątnej mają wartość 1. Tworzy się ją wywołując funkcję `eye` i podając jako argument rząd macierzy, np:

```
M1 = eye(4)
```

W tworzeniu macierzy mogą być pomocne funkcje tworzące macierz zerową – złożoną z samych zer, lub macierz złożoną z samych jedynek:

```
zeros(1,3)
M0 = zeros(3)
Mones = ones(4)
```

Wyznacznik dowolnej macierzy kwadratowej liczy funkcja `det`:

`wyznacznik_Md = det(Md)`

Jeśli wyznacznik macierzy jest zerowy, to mówimy, że macierz jest osobliwa.

Zadania do samodzielnego wykonania

4) Utwórz macierz `G` przepisując do niej pierwsze 4 wiersze z macierzy `g`, następnie usuń z macierzy `g` trzeci wiersz:

`G = g(1:4,:),  g(3,:) = []`

wykonaj działania: `g*G,  G*g,  g.*G,  G.*g,  g(1,:)*G(:,1),  G(1,:)*g(:,1)` i porównaj wyniki działań. Które z powyższych działań daje taki sam wynik jak suma iloczynów: `-3*-3 + 7*8 + 2*-2 + 1*1` ?

Czy mnożenie macierzy jest przemienne?

Rozwiązywanie układów równań

Rozwiązywanie układów równań w sposób tradycyjny (metodą wyznaczników) polega na obliczeniu wyznacznika głównego **W** równania i tylu wyznaczników $\mathbf{W_i}$ ile jest zmiennych. Następnie dzieląc $\mathbf{W_i}$ przez **W** otrzymujemy kolejne szukane zmienne. W Matlabie rozwiązanie prostego układu równań sprowadza się do wykonania jednego dzielenia.

Przykład

Dany jest układ równań z trzema niewiadomymi *x*, *y* i *z*:

$$x + y + z = 6$$

$$2x + y - z = 3$$

$$4x + 5y - 3z = 13$$

Zapisujemy układ równań w postaci macierzowej:

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \\ 4 & 5 & -3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 13 \end{bmatrix}, \text{ czyli } \boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}, \text{ gdzie: } \boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \\ 4 & 5 & -3 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 6 \\ 3 \\ 13 \end{bmatrix}$$

w Matlabie macierze ***A*** i ***b*** definiujemy następującymi komendami:

```
A = [1 1 1;2 1 -1;4 5 -3]
b = [6 3 13]'
```

Rozwiązanie w sposób tradycyjny:

- liczymy wyznacznik główny: `W = det(A)`
- liczymy wyznacznik pierwszy: `W1 = det(A1)`, gdzie **`A1`** jest macierzą powstałą z macierzy **`A`** przez zastąpienie pierwszej kolumny wartościami macierzy ***b***, czyli: `A1=A; A1(:,1)=b;`
- liczymy wyznacznik drugi: `W2 = det(A2)`, gdzie **`A2`** jest macierzą powstałą z macierzy **`A`** przez zastąpienie drugiej kolumny wartościami macierzy ***b***, czyli: `A2=A; A2(:,2)=b;`
- liczymy wyznacznik trzeci: `W3 = det(A3)`, gdzie **`A3`** jest macierzą powstałą z macierzy **`A`** przez zastąpienie trzeciej kolumny wartościami macierzy ***b***, czyli: `A3=A; A3(:,3)=b;`
- liczymy wartości *x*, *y*, *z* dzieląc odpowiedni wyznacznik przez wyznacznik główny:

```
x=W1/W,    y=W2/W,    z=W3/W
```

Rozwiązanie macierzowe

Wartości *x*, *y* i *z* znajdujemy wykonując dzielenie:

```
X = A\b
```

x, *y* i *z* będą kolejnymi wyrazami macierzy `X`.

Zadania do samodzielnego wykonania

5) Zapisz poniższe układy równań w postaci macierzowej (***A·x*** = ***b***) – tak jak na poprzedniej stronie. Zdefiniuj w Matlabie odpowiednie macierze i rozwiąż układy równań, a następnie sprawdź otrzymane wyniki (sprawdź, czy faktycznie wymnożenie macierzy ***A*** przez wektor ***x*** jest równe wektorowi ***b***).

$$\begin{aligned} x - 2y + 3z &= 6 \\ 4x + 5y + 6z &= 32 \\ 7x + 8y + 9z &= 50 \end{aligned}$$

Zapisujemy układ równań w postaci macierzowej:

***A*1·*x*1** = ***b*1**

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 4x_4 &= 30 \\ 4x_2 + 5x_3 + 6x_4 &= 47 \\ 6x_3 + 7x_4 &= 46 \\ 8x_4 &= 32 \end{aligned}$$

Zapisujemy układ równań w postaci macierzowej:

***A*2·*x*2** = ***b*2**

$$\begin{aligned} x_1 \cdot 10^{-6} + 2x_2 + 3x_3 + 4x_4 &= 29 \\ 4x_2 \cdot 10^{-6} + 5x_3 + 6x_4 &= 39 \\ 6x_3 \cdot 10^{-6} + 7x_4 &= 28 \\ 80x_4 \cdot 10^{-7} &= 3.2 \cdot 10^{-5} \end{aligned}$$

Zapisujemy układ równań w postaci macierzowej:

***A*3·*x*3** = ***b*3**

6) Sprawdź, czy rozwiązanie drugiego układu równań (***x*2**) jest też rozwiązaniem trzeciego układu równań, czyli: czy spełnione będzie równanie ***A*3·*x*2** = ***b*3**. Czy to możliwe, aby układ równań miał więcej niż jedno rozwiązanie?

UWAGA: Następne zajęcia (*Rozwiązywanie układów równań metodami nieiteracyjnymi i iteracyjnymi*) zaczynamy wejściówką. Przygotuj się z następujących zagadnień:

Co oznaczają określenia macierzy: kwadratowa, trójkątna, osobliwa, jednostkowa, zerowa?

Jakie warunki muszą być spełnione, aby można było mnożyć macierz przez macierz?

Oblicz wyznacznik podanej macierzy (drugiego, trzeciego lub czwartego rzędu).

Rozwiąż (dowolną metodą) zadany układ równań (trzeciego lub czwartego rzędu).

Opisz metody rozwiązywania układów równań.

- nieiteracyjne: Metoda eliminacji Gaussa, Metoda eliminacji Jordana (Metoda eliminacji zupełnej), Rozkład LU,
- iteracyjne: Metoda iteracji prostych, Metoda Jacobiego, Metoda Gaussa-Seidla.

Napisz w Matlabie skrypt realizujący jedną z powyższych metod rozwiązywania układów równań.

Dlaczego stosuje się metody iteracyjne?