

Метод облічення

Лабораторна робота 1

1 Матриці

1.1 Створення матриць

Матриці можна сформулювати через:

- заміну елементів
- генерацію елементів
- складання з інших матриць

1.1.1 Заміна елементів

$$\begin{aligned} A &= [1\ 2\ 3; 7\ 8\ 9] \\ A &= [1,2,3; 7,8,9] \\ A &= [1\ 2\ 3 \\ &\quad 7,8,9] \end{aligned}$$

1.1.2 Генерація елементів

Одним із специфічних операторів в програмі Matlab/Octave є двокрапка. Уможливило вона творення векторів при рівномірно розкладених елементах. Команда:

$$p : q : r$$

утворить вектор постаті:

$$[p, p + q, p + 2q, \dots, r]$$

Якщо параметр q замінений буде ужитий крок $q = 1$.

Приклади застосування:

```
>> 1:7
ans = 1 2 3 4 5 6 7
>> 1:3:10
ans = 1 4 7 10
```

Таблиця 1: Функції, допомагаючи генерації матриці

Функція	Опис
<code>eye(n)</code>	одична матриця у вимірах $n \times n$
<code>eye(n,m)</code>	або $n \times m$
<code>ones(n)</code>	матриця сповнена одиницями
<code>ones(n,m)</code>	
<code>rand(n)</code>	матриця сповнена псевдовипадковими числами
<code>rand(n,m)</code>	у одноманітний розклад на $[0,1]$
<code>zeros(n)</code>	матриця сповнена нулями
<code>zeros(n,m)</code>	

1.1.3 Склеювання

Способом, що часто зустрічається, є також створення матриці з інших матриць.

Приклад:

```
>> A = 1:4 ;

>> B = 1:5 ;
>> C = 0 ;
>> D = [A, C ; B ]
D = 1 2 3 4 0
    1 2 3 4 5
```

Слід притому пам'ятати, щоб виміри з'єднаних матриць впогоджувалися. У протилежному випадку спроба утворення матриці закінчиться повідомленням про помилку.

1.2 Видалення частини матриці

Команда

```
>> []
```

визначає пусту матрицю. Приписування такої матриці до рядка або колони спричинить їх усунення, наприклад:

```
>> D(:,2:3) = []
D = 1 4 0
    1 4 5
>> D(1,:) = []
D = 1 3 4 5
```

Увага: Таким чином ми можемо усувати цілі рядки, колони або їх групи. Усунення поодинокого елемента матриці немало б сенсу і генерує помилку.

Таблиця 2: Арифметичні оператори

Оператор	матричний	табличний
додавання	+	+
віднімання	-	-
множення	*	.*
ділення правостороннє	/	./
ділення лівостороннє	\	.\
піднесення до квадрату	^	.^
транспозиція	'	!'

2 Операції на матрицях

2.1 Арифметичні оператори (матричні і табличні)

Matlab/Octave уможливує виконання двох видів арифметичні операції на матрицях. Матричні оператори визначені правилами матрицевої алгебри. Натомість табличні оператори трактують матриці як таблиці чисел і уможливають виконання операції на елементах матриці, що відповідають їй. Кожен оператор матриці має свого табличного відповідника, що передує крапкою. Винятком є оператор додавання і віднімання, які з визначення оперують на елементах матриці.

Увага: Щоб виконати яку-небудь табличну операцію матриці мусять мати такі самі розміри або одна з них мусить бути скаляром.

Увага: Згідно з визначенням добутків(у сенсі Cauchy'ego) двох матриць може бути визначений тільки тоді коли число колон першої матриці є рівне числу ряду другої матриці. Множення матриць не є змінне!

Як відомо ділення є зворотною операцією до множення ($\frac{a}{b} = a \cdot b^{-1}$). Matlab уможливує два види ділення. Правобічне ділення двох матриць сформульовано наступним чином:

$$A/B = A \cdot B^{-1}$$

У свою чергу лівобічне множення:

$$A \setminus B = A^{-1} \cdot B$$

У аналогічний спосіб сформульовано є табличне ділення. Кожен елемент c_{ij} матриці $C = A./B$ є рівний $c_{ij} = a_{ij} \cdot \frac{1}{b_{ij}}$. У свою чергу для табличного ділення правобічного, чи коли $C = A.\setminus B$ маємо: $c_{ij} = \frac{1}{a_{ij}} \cdot b_{ij}$.

2.2 Оператори порівнянь

В програмі Matlab/Octave можна порівнювати цілі матриці (таблиці). Умовою є такі самі розміри обох операндів (Винятком є ситуація, коли один з них є скаляром. Тоді той скаляр порівнюваний з кожним другим елементом операнду). Результатом дії операторів порівнянь є матриця (таблиця)

Таблиця 3: Оператори порівняння

Оператор	Означення
==	дорівнює
~=	різність
>	більше ніж
<	менше ніж
>=	більше або дорівнює
<=	менше або дорівнює

Таблиця 4: Оператори і функції логічні

Оператор	Логічна функція
~	заперечення
&	кон'юнкція
	диз'юнкція
<i>xor</i>	симетрична різниця

такої самої величини що і операнд сповнена одиницями де відповідаючи собі елементи виконують звіт і нулі в протилежному випадку.

І так наприклад, якщо $A = [-2 \ 3 \ 0]$, $B = [3 \ 3 \ 3]$, вираз $A >= B$ має вартість $[0 \ 1 \ 0]$, натомість $A \sim B$ буде рівне $[1 \ 0 \ 1]$.

2.3 Логічні оператори

Найновіша версія програми відкриває доступ трьом видам операторів і функцій логічних. То оператори:

- „елементові”(element-wise) - виконують операції на елементах матриці трактуючи число 0 як логічний нуль і кожен іншу вартість як логічну одиницю.
- бітові (bit-wise) - оперують на цілковитих або побудованих з них матрицях таких самих розмірів. Виконують логічні операції на бітових вартостях відповідних чисел.
- „швидкі”(short-circuit) - оператори альтернативи і кон'юнкції, які працюють на логічних висловах заключаючи вартість скалярну. У випадку коли по перевірці першого операнду логічна цінність виразу визначена наступна евалюація операнду оминає. (відповідних операторів `||` і `&&` в C/C++)

Оскільки на лабораторії будуть вживані головні і перші із згаданих логічних операторів, то тільки вони подані в таблиці 4. Докладний опис всіх, однаково логічних як і інших операторів можна відшукати [тут](#).

3 Програмування

3.1 Умовна інструкція if

Умовна інструкція if

Загальна постать інструкції:

```
if wyrażenie_warunkowe1
  instrukcje1
elseif wyrażenie_warunkowe2
  instrukcje2
else
  instrukcje_koncowe
endif
```

Інструкція вибору switch

```
switch wyrażenie
case wartosc1
  instrukcje1
case wartosc2
  instrukcje2
...
otherwise
  instrukcje
endswitch
```

Цикл for

```
for zmienna = macierz_wartosci
  instrukcje
endfor
```

Цикл while

```
while wyrażenie
  instrukcje
endwhile
```

Цикл do-until

```
do
  instrukcje
until wyrażenie
```

Інструкція break

Спричиняє припинення виконання даної петлі. Відкидний є тільки один рівень заглибини.

Інструкція return

Спричиняє безумовне припинення виконання даного документу (функції) і повернення до його (її) місця виклику.

У синтаксисі мови Matlab інструкції `if`, `switch`, `for`, `while` завершується командою `end`, а петля `do-until` не виступає.

4 Функція

Основним способом творення власної функції є її установка змісту разом з визначенням в папці з розширенням `.m`. Загальна постать визначення функції виглядає наступним чином:

```
function [wartosc1, wartosc2, ...] = nazwa_funkcji(parametr1, parametr2, ...)
instrukcje
endfunction
```

Прикладова функція записана в папці `porownaj.m`:

```
% Tutaj mozemy dac opis funkcji, ktory bedzie
% wyswietlany przez program gdy wpiszemy w linii
% polecen: help porownaj
```

```
% A ta linia wyswietlona nie zostanie
```

```
function k = porownaj(a,b)
```

```
    if (a<b)
        k = -1 ;
    elseif (a>b)
        k = 1 ;
    else
        k = 0 ;
    end
endfunction
```

Однаково параметри як і вартість функції є опціональні. Назва функції з визначення не мусить бути така ж як назва файлу, але функцію можна викликати через подання назви файлу. Змінні, утворені у середині такої

функції місцеві, тобто не видимі в документі, з якого лишилася вона спричинена. Подібно змінні, утворені поза функцією (робочі, що знаходиться в просторі), не доступні з її середини. Щоб змінні, існуючі в спільній пам'яті програми, були доступні у середині функції мусять бути задекларовані як глобальні (команда `global`).

Характерною ознакою Matlab'a/Octave є те, що аргументи функції копіювані всього лише тоді коли модифіковані в її тілі.

Аргументи функції передавані через вартість, однак їх копіювання настає всього лише тоді коли модифіковані в її тілі. (У зразковій функції *porownaj* аргументи функції небули копіювані)

Більше цікавого на тему функції можна відшукати [тут](#).

5 Вправи

1. Утвори власний документ з розширенням .m.
2. Перевір результати нижче згаданих команд вписуючи їх до документу і виконайте їх.

```
% TWORZENIE MACIERZY
% Wymienianie element [U+FFFD]w
A = [1 2 ...
      3; 7 8 9]

B = [1, 2, 3; 7, 8, 9]

C = [1 2 3
      7,8,9]

% Generowanie
% Operator :
w1 = 1 : 0.31 : 3
w2 = -3:7
w3 = 10:-2:0

% Wykorzystanie funkcji
X = ones(2)
Y = eye(3)
Z = rand(2,5) % round(10*rand(3))

% Zbudowanie z innych macierzy
U = [A, X ; Z]

% Mieszanie powyzzszych sposobow
V = [ [9:-1:7; 1:3], zeros(2,1); 1 1 1 1]

%-----
% ODWOLYWANIE SIE DO ELEMENTOW MACIERZY
V(1,3)
V(3,1)
V(:,2)
V(2,:)
V(2, 1:3)
V(2, [1 3 4])
V([1 3], [1 3 4])
V(7)
V(:)
```



```

%-----
% USUWANIE FRAGMENTOW MACIERZY
V(:,4) = []
V(2,:) = []
% V(1,1) = [] % !!!

%-----
% OPERACJE NA MACIERZACH
A = [1 0 0;
      2 3 -1;
      0 7 2]

B = [1 0 3;
      -1 5 2;
      2 2 2]

A+B
A-B
A+2

2*A
2.*A

disp('Mnozenie macierzowe')
MM1 = A*B
MM2 = B*A

disp('Mnozenie tablicowe')
MT1 = A.*B
MT2 = B.*A

disp(['Dzielenie', ' ', 'macierzowe'])
DM1 = A/B
DM2 = A\B
DM3 = B/A
DM4 = B\A

disp('Dzielenie tablicowe')
DT1 = A./B
DT2 = A.\B
DT3 = B./A
DT4 = B.\A

disp(['Potego' ' wanie'])
PM = A^2
A*A

```

```

PT = A.^2
A.*A

disp(['Transpozycja'])
A.' % transpozycja
A'

C = [1-i 2i    3;
     -1  5+3i  2+3i;
      3+i  2    2]
C.'
C' % hermitowskie sprzezenie macierzy (dla m. zespolonych)

% Operacje porownan i funkcje logiczne
A == B
A ~= B
2 < A
% A > B
% A < B
% A >= B
% A <= B

~A
A & B
% A | B
% xor(A,B)

A = [2 0 7;
     0 0 1]

all(A)
any(A)

%-----
%----- INSTRUKCJE STERUJACE -----%

                % INSTRUKCJA "if"

a = 2
b = 4

if (a<b)
    k = -1
elseif (a>b)
    k = 1
else
    k = 0

```

```

endif

                                % INSTRUKCJA "switch"
c = 2 ;
switch c
    case 0
        zeros(2)
    case 1
        disp('wpisano 1')
    otherwise
        disp(['wpisano ' num2str(c) ' tym razem'])
endswitch

                                % P[U+FFFD]TLA "for"
suma=0;
wekt = 1:10 ;
for k = wekt
    suma=suma+k;
endfor
S=suma;

                                % P[U+FFFD]TLA "while"
suma=0;
k=1;
while k<=10
    suma=suma+k;
    k=k+1;
endwhile
S=suma;

                                % INSTRUKCJA "break"
                                % zatrzymuje wykonywanie p[U+FFFD]tli "for" lub "while" ...
                                % ... i powoduje wyjście poza petle
suma=0;
i=1;
while i<=10
    suma=suma+i; break
    i=i+1;
endwhile
S=suma;

```

3. Напиши функцію rogownaj.m описану в розділі 4. Виклич її з лінії команд також з середини вище згаданого документа.
4. Напиши функцію fun1.m, яка лічитиме вартість многочлена $x^3 + x^2 - 3x - 3$. Параметр функції повинен приймати не тільки вартості ска-

лярні,але також матриці (вектори).

5. Напиши функцію, яка в результаті поверне квадратну матрицю на елементи доль, що є цілковитими числами з границею $< 0, 10 >$ і її слід (Слід матриці то сума елементів, лежачих на її головній діагоналі).
6. Напиши функцію silnia (факторіал).