



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE



Obliczenia w Matlabie

Tablice

Łukasz Sztangret

Katedra Informatyki Stosowanej i Modelowania



Indeksowanie tablic

W Matlabie tablice indeksowane są za pomocą nawiasu okrągłego (`()`). Odwołanie się do elementu tablicy jest możliwe na dwa sposoby:

- podając nr elementu (elementów),
- podając tablicę logiczną.

Odwołując się za pomocą nr elementu należy pamiętać, że:

- indeksy rozpoczynają się od **1** nie od **0**,
- możliwe jest odwołanie się do kilku elementów podając jako indeks wektor.

Odwołując się za pomocą tablicy logicznej należy pamiętać, że tablica logiczna musi mieć taki sam rozmiar jak tablica, do elementów której się odwołujemy (w przypadku wektorów oba muszą mieć tą samą długość).

Indeksując tablicę pierwszy indeks odnosi się do wiersza, drugi do kolumny.

Indeksowanie tablic

```
>> a=[10,12,14,16,18]
```

```
a =
```

```
    10    12    14    16    18
```

```
>> a(1)
```

```
ans =
```

```
    10
```

Długość tablicy

```
>> a(length(a))
```

```
ans =
```

```
    18
```

Ostatni element
tablicy

```
>> a(end)
```

```
ans =
```

```
    18
```

```
>> a([1,3])
```

```
ans =
```

```
    10    14
```

```
>> a(1:3)
```

```
ans =
```

```
    10    12    14
```

```
>> a(1:end-1)
```

```
ans =
```

```
    10    12    14    16
```

```
>> a(end:-1:1)
```

```
ans =
```

```
    18    16    14    12    10
```

Indeksowanie tablic

```
>> a=[10,12,14,16,18]
```

```
a =
```

```
    10    12    14    16    18
```

```
>> w=a>15
```

```
w =
```

```
    0    0    0    1    1
```

Tablica logiczna

```
>> a(w)
```

```
ans =
```

```
    16    18
```

```
>> a(a>15)
```

```
ans =
```

```
    16    18
```

```
>> a(a==12)
```

```
ans =
```

```
    12
```

```
>> a(a~=12)
```

```
ans =
```

```
    10    14    16    18
```

```
>> a(a>10 & a<18)
```

```
ans =
```

```
    12    14    16
```

Tablice dwuwymiarowe

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

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

```
>> A(2,1)
```

```
ans =
     4
```

```
>> A(1,end)
```

```
ans =
     3
```

```
>> A(end,1)
```

```
ans =
     7
```

```
>> A(1:2,[1,3])
```

```
ans =
     1     3
     4     6
```

```
>> A(1:end,1:2:end)
```

```
ans =
     1     3
     4     6
     7     9
```

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

```
ans =
     1
     4
     7
```

```
>> A(2,:)
```

```
ans =
     4     5     6
```

```
>> A(1:2,:)
```

```
ans =
     1     2     3
     4     5     6
```

To samo co
A(1:end,1)

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

```
ans =
     4     6
```

Tablice wielowymiarowe

```
>> A=zeros(3,3);
>> for i=1:3*3
A(i)=i;
end
>> disp(A)
    1     4     7
    2     5     8
    3     6     9
```

```
>> A=zeros(3,3,3);
>> for i=1:3*3*3
A(i)=i;
end
>> disp(A)
(:,:,1) =
    1     4     7
    2     5     8
    3     6     9
(:,:,2) =
   10    13    16
   11    14    17
   12    15    18
(:,:,3) =
   19    22    25
   20    23    26
   21    24    27
```

```
>> A=zeros(3,3,3);
>> k=1;
>> for i=1:3
for j=1:3*3
A(i,j)=k;
k=k+1;
end
end
>> disp(A)
(:,:,1) =
    1     2     3
   10    11    12
   19    20    21
(:,:,2) =
    4     5     6
   13    14    15
   22    23    24
(:,:,3) =
    7     8     9
   16    17    18
   25    26    27
```

```
>> A=zeros(3,3,3);
>> k=1;
>> for i=1:3*3
for j=1:3
A(i,j)=k;
k=k+1;
end
end
>> disp(A)
(:,:,1) =
    1     2     3
    4     5     6
    7     8     9
   10    11    12
   13    14    15
   16    17    18
   19    20    21
   22    23    24
   25    26    27
(:,:,2) =
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
(:,:,3) =
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
    0     0     0
```

Do elementów tablic wielowymiarowych można odnosić się mniejszą liczbą indeksów

Pierwszy indeks odnosi się do wiersza, drugi do pozostałych wymiarów.

Tworzenie macierzy i wektorów

```
>> A=zeros(2,3)
A =
    0    0    0
    0    0    0
>> B=zeros(3)
B =
    0    0    0
    0    0    0
    0    0    0
>> C=ones(1,4)
C =
    1    1    1    1
>> D=eye(3)
D =
    1    0    0
    0    1    0
    0    0    1
>> E=eye(2,4)
E =
    1    0    0    0
    0    1    0    0
F=diag(5)
F =
    5
>> G=diag(1:4)
G =
    1    0    0    0
    0    2    0    0
    0    0    3    0
    0    0    0    4
>> H=true(1,3)
H =
    1    1    1
>> I=false(2,1)
I =
    0
    0
>> J=rand(2,3)
J =
    0.7547    0.6797    0.1626
    0.2760    0.6551    0.1190
>> K=randn(1,4)
K =
   -0.0068    1.5326   -0.7697    0.3714
>> L=randi(5,1,5)
L =
    2    4    2    3    4
>> M=randperm(10)
M =
    4    5    10    8    6    3    9    7    1    2
>> N=linspace(0,1,6)
N =
    0    0.2000    0.4000    0.6000    0.8000    1.0000
>> O=logspace(0,3,4)
O =
    1    10    100    1000
```

Łączenie macierzy i wektorów

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

```
A =
     1     2
     3     4
```

```
>> x=[10;11]
```

```
x =
    10
    11
```

```
>> B=[A,x]
```

```
B =
     1     2    10
     3     4    11
```

```
>> y=[20,21,22]
```

```
y =
    20    21    22
```

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

```
C =
     1     2    10
     3     4    11
    20    21    22
```

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

```
A =
     1     2
     3     4
```

```
>> B=[5,6;7,8]
```

```
B =
     5     6
     7     8
```

```
>> horzcat(A,B)
```

```
ans =
     1     2     5     6
     3     4     7     8
```

```
>> vertcat(A,B)
```

```
ans =
     1     2
     3     4
     5     6
     7     8
```

```
>> cat(1,A,B)
```

```
ans =
     1     2
     3     4
     5     6
     7     8
```

```
>> cat(2,A,B)
```

```
ans =
     1     2     5     6
     3     4     7     8
```

```
>> cat(3,A,B)
```

```
ans(:,:,1) =
     1     2
     3     4
ans(:,:,2) =
     5     6
     7     8
```

horzcat (vertcat) – łączy elementy w poziomie (pionie), to samo co [A,B] ([A;B])

Cat – łączy elementy w wybranym kierunku

Łączenie macierzy i wektorów

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

```
A =
     1     2
     3     4
```

```
>> repmat(A,2,1)
```

```
ans =
     1     2
     3     4
     1     2
     3     4
```

```
>> repmat(A,1,2)
```

```
ans =
     1     2     1     2
     3     4     3     4
```

```
>> repmat(A,2)
```

```
ans =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

```
>> repmat(A,[2,2,2])
```

```
ans(:,:,1) =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

```
ans(:,:,2) =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

repmat – zwielokrotnia
tablicę wzdłuż określonego
wymiaru

```
>> repmat(A,[2,2,2,2])
```

```
ans(:,:,1,1) =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

```
ans(:,:,2,1) =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

```
ans(:,:,1,2) =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

```
ans(:,:,2,2) =
     1     2     1     2
     3     4     3     4
     1     2     1     2
     3     4     3     4
```

Łączenie macierzy i wektorów

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

```
A =
```

```
1 2
```

```
3 4
```

```
>> B=5
```

```
B =
```

```
5
```

```
>> C=randi(10,3)
```

```
C =
```

```
6 8 6
```

```
10 8 1
```

```
3 4 1
```

```
>> D=blkdiag(A,B,C)
```

```
D =
```

1	2	0	0	0	0
3	4	0	0	0	0
0	0	5	0	0	0
0	0	0	6	8	6
0	0	0	10	8	1
0	0	0	3	4	1

Wymiary macierzy

```
>> A=[1,2,3;4,5,6]
```

```
A =  
    1    2    3  
    4    5    6
```

```
>> x=[1,3,5,7]
```

```
x =  
    1    3    5    7
```

```
>> length(x)
```

```
ans =  
    4
```

```
>> length(A)
```

```
ans =  
    3
```

```
>> length(A.')
```

```
ans =  
    3
```

length – długość wektora, najdłuższy bok tablicy

```
>> size(x)
```

```
ans =  
    1    4
```

```
>> size(A)
```

```
ans =  
    2    3
```

```
>> size(A,1)
```

```
ans =  
    2
```

```
>> size(A,2)
```

```
ans =  
    3
```

size – wymiary tablicy

```
>> numel(x)
```

```
ans =  
    4
```

```
>> numel(A)
```

```
ans =  
    6
```

```
>> ndims(x)
```

```
ans =  
    2
```

```
>> ndims(A)
```

```
ans =  
    2
```

```
>> a=5
```

```
a =  
    5
```

```
>> ndims(a)
```

```
ans =  
    2
```

```
>> size(a)
```

```
ans =  
    1    1
```

numel – liczba elementów tablicy

ndims – liczba wymiarów tablicy (skalar, wektor i macierz mają po dwa wymiary)

Obroty macierzy

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

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

```
>> fliplr(A)
```

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

```
>> flipud(A)
```

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

```
>> flipdim(A,1)
```

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

```
>> flipdim(A,2)
```

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

```
>> rot90(A)
```

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

```
>> rot90(A,2)
```

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

```
>> rot90(A,3)
```

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

```
>> reshape(A,1,9)
```

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

```
>> reshape(A,9,1)
```

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

min, max

```
>> x=randi(10,1,5)
```

```
x =
```

```
    9    7    4   10    1
```

```
>> a=min(x)
```

```
a =
```

```
    1
```

```
>> [a,b]=min(x)
```

```
a =
```

```
    1
```

```
b =
```

```
    5
```

Wartość minimalna

Pozycja wartości minimalnej

```
>> c=max(x)
```

```
c =
```

```
   10
```

```
>> [c,d]=max(x)
```

```
c =
```

```
   10
```

```
d =
```

```
    4
```

min, max

```
>> x=randi(10,1,5)
```

```
x =
```

```
    3    7    5    4    9
```

```
>> y=5
```

```
y =
```

```
    5
```

```
>> a=min(x,y)
```

```
a =
```

```
    3    5    5    4    5
```

```
>> [a,b]=min(x,y)
```

```
>> x=randi(10,1,5)
```

```
x =
```

```
    8    4    6    1    1
```

```
>> y=randi(10,1,5)
```

```
y =
```

```
    6    8   10    2    6
```

```
>> b=min(x,y)
```

```
b =
```

```
    6    4    6    1    1
```

BŁĄD



min, max

```
>> A=randi(100,4,4)
```

```
A =  
 96  76  90  15  
 35  26  96  26  
 59  51  55  85  
 23  70  14  26
```

```
>> a=min(A)
```

```
a =  
 23  26  14  15
```

```
>> [a,b]=min(A)
```

```
a =  
 23  26  14  15
```

```
b =  
 4  2  4  1
```

```
>> c=min(min(A))
```

```
C =  
 14
```

Wartość minimalna z każdej kolumny

Pozycja wartości minimalnej w każdej kolumnie

min, max

```
>> A=randi(100,4,4)
```

```
A =
```

```
27 12 51 3
61 30 9 93
72 32 27 74
23 43 81 49
```

```
>> [a,b]=min(A,[],1)
```

```
a =
```

```
23 12 9 3
```

```
b =
```

```
4 1 2 1
```

```
>> [a,b]=min(A,[],2)
```

```
a =
```

```
3
```

```
9
```

```
27
```

```
23
```

```
b =
```

```
4
```

```
3
```

```
3
```

```
1
```

Wartość minimalna z
każdego wiersza

Wartość minimalna z każdej
kolumny

min, max

Wartość minimalna z
każdego wiersza

```
>> A=randi(100,2,2,2)
```

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

```
 34  37
```

```
 91  12
```

```
A(:,:,2) =
```

```
 79  25
```

```
 39  41
```

```
>> min(A)
```

```
ans(:,:,1) =
```

```
 34  12
```

```
ans(:,:,2) =
```

```
 39  25
```

Wartość minimalna z każdej
kolumny

```
>> min(A,[],2)
```

```
ans(:,:,1) =
```

```
 34
```

```
 12
```

```
ans(:,:,2) =
```

```
 25
```

```
 39
```

```
>> min(A,[],3)
```

```
ans =
```

```
 34  25
```

```
 39  12
```

Wartość minimalna z każdej
płaszczyzny

mean, sum, prod

```
>> x=randi(10,1,5)
```

```
x =
```

```
1 3 4 9 1
```

```
>> mean(x)
```

```
ans =
```

```
3.6000
```

Średnia
arytmetyczna

```
>> sum(x)
```

```
ans =
```

```
18
```

Suma

```
>> prod(x)
```

```
ans =
```

```
108
```

Iloczyn

```
>> A=randi(10,3,3)
```

```
A =
```

```
1 8 6
```

```
2 7 3
```

```
7 5 8
```

Wartości
średnie z
każdej
kolumny

```
>> mean(A)
```

```
ans =
```

```
3.3333 6.6667 5.6667
```

```
>> mean(A,2)
```

```
ans =
```

```
5.0000
```

```
4.0000
```

```
6.6667
```

Wartości
średnie z
każdego
wiersza

sort

```
>> x=randi(100,1,5)
x =
    63    59    21    31    48
>> a=sort(x)
a =
    21    31    48    59    63
>> a=sort(x,'descend')
a =
    63    59    48    31    21
>> [a,b]=sort(x)
a =
    21    31    48    59    63
b =
     3     4     5     2     1
```

Posortowany wektor x

Pozycje na których
znajdowały się
poszczególne elementy
przed sortowaniem



sort

```
>> A=randi(100,3,3)
```

```
A =
```

```
44  98  26  
19  44  41  
91  12  60
```

```
>> a=sort(A)
```

```
a =
```

```
19  12  26  
44  44  41  
91  98  60
```

```
>> [a,b]=sort(A)
```

```
a =
```

```
19  12  26  
44  44  41  
91  98  60
```

```
b =
```

```
2  3  1  
1  2  2  
3  1  3
```

Każda kolumna sortowana jest oddzielnie

sort

```
>> A=randi(100,3,3)
```

```
A =
```

```
44 98 26
```

```
19 44 41
```

```
91 12 60
```

```
>> a=sort(A,2)
```

```
a =
```

```
26 44 98
```

```
19 41 44
```

```
12 60 91
```

```
>> [a,b]=sort(A,2)
```

```
a =
```

```
26 44 98
```

```
19 41 44
```

```
12 60 91
```

```
b =
```

```
3 1 2
```

```
1 3 2
```

```
2 3 1
```



Sortowanie wierszy

find

```
>> x=randi(100,1,10)
```

```
x =
```

```
7 73 26 75 5 1 2 16 68 100
```

```
>> x<50
```

```
ans =
```

```
1 0 1 0 1 1 1 1 0 0
```

```
>> x(x<50)
```

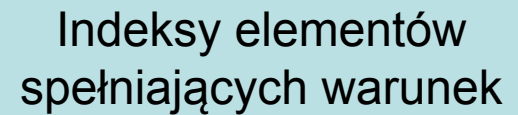
```
ans =
```

```
7 26 5 1 2 16
```

```
>> find(x<50)
```

```
ans =
```

```
1 3 5 6 7 8
```



Indeksy elementów
spełniających warunek

find

```
>> x=randi(100,1,10)
```

```
x =
```

```
7 73 26 75 5 1 2 16 68 100
```

```
>> find(x<50,3)
```

```
ans =
```

```
1 3 5
```

```
>> find(x<50,3,'last')
```

```
ans =
```

```
6 7 8
```

3 pierwsze indeksy
elementów spełniających
warunek

3 ostatnie indeksy
elementów spełniających
warunek



find

```
>> A=randi(10,4,4)
```

```
A =  
    6    9    3   10  
    7    7    8   10  
    3    1    4    5  
    1   10    7    4
```

```
>> A>7
```

```
ans =  
    0    1    0    1  
    0    0    1    1  
    0    0    0    0  
    0    1    0    0
```

```
>> A(A>7)
```

```
ans =  
    9  
   10  
    8  
   10  
   10
```

```
>> find(A>7)
```

```
ans =  
    5  
    8  
   10  
   13  
   14
```

Pojedynczy indeks

```
>> [a,b]=find(A>7)
```

```
a =  
    1  
    4  
    2  
    1  
    2
```

Numer wiersza

```
b =  
    2  
    2  
    3  
    4  
    4
```

Numer kolumny

```
>> [a,b,c]=find(A>7)
```

```
a =  
    1  
    4  
    2  
    1  
    2  
  
b =  
    2  
    2  
    3  
    4  
    4  
  
c =  
    1  
    1  
    1  
    1  
    1
```

Prawdziwe wartości

find

```
>> A=randi(3,3)-1
```

```
A =  
 2  2  0  
 2  1  1  
 0  0  2
```

```
>> [a,b,c]=find(A)
```

```
a =
```

```
1  
2  
1  
2  
2  
3
```

Numer
wiersza

```
b =
```

```
1  
1  
2  
2  
3  
3
```

Numer
kolumny

```
c =
```

```
2  
2  
2  
1  
1  
2
```

Prawdziwe
wartości

find

```
>> A=randi(10,3,3,3)
```

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

```
 9 10  2  
 1 10  9  
 8  1  3
```

```
A(:,:,2) =
```

```
 8  4 10  
 1  7  6  
 2  8  2
```

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

```
 7  8  3  
 7 10  2  
 7  6  9
```

```
>> find(A>8)
```

```
ans =
```

```
 1  
 4  
 5  
 8  
16  
23  
27
```

Pojedynczy indeks

```
>> [a,b]=find(A>8)
```

```
a =
```

```
 1  
 1  
 2  
 2  
 1  
 2  
 3
```

Numer wiersza

```
b =
```

```
 1  
 2  
 2  
 3  
 6  
 8  
 9
```

Numer kolumny i płaszczyzny



Tablice komórkowe

W przeciwieństwie do zwykłych tablic, każdy element tablicy komórkowej może być innego typu.

Tworzymy je używając nawiasów `{}` zamiast `[]`.

```
>> tab={1, [1 2 3], [4;5;6], 'AGH'}
```

```
tab =
```

```
    [1]    [1x3 double]    [3x1 double]    'AGH'
```

```
>> tab1=cell(1,3)
```

```
tab1 =
```

```
    []    []    []
```

Tablice komórkowe

```
>> tab={1, [1 2 3], [4;5;6], 'AGH'};
>> tab{1}
ans =
     1
>> tab{2}
ans =
     1     2     3
>> tab{3}
ans =
     4
     5
     6
>> tab{4}
ans =
AGH
```

To, co jest w komórce

```
>> tab(1)
ans =
     1
>> tab(2)
ans =
 [1x3 double]
>> tab(3)
ans =
 [3x1 double]
>> tab(4)
ans =
'AGH'
```

Komórka o wymiarach 1x1

Tablice komórkowe

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

```
A =
```

```
1 2
3 4
```

```
>> B={A}
```

```
B =
```

```
[2x2 double]
```

```
>> C=num2cell(A)
```

```
C =
```

```
[1] [2]
[3] [4]
```

```
>> D=cell2mat(B)
```

```
D =
```

```
1 2
3 4
```

Komórka 1x1
zawierająca tablicę 2x2

Tablica komórek 2x2
zawierająca liczby

```
>> E=cell2mat(C)
```

```
E =
```

```
1 2
3 4
```

```
>> who
```

Your variables are:

```
A B C D E
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	2x2	32	double	4*8=32
B	1x1	144	cell	112+4*8=144
C	2x2	480	cell	4*112+4*8=480
D	2x2	32	double	
E	2x2	32	double	

Tablice komórkowe

```
>> A={ [1,2,3;4,5,6]; [7,8;9,0] }
```

```
A =
```

```
 [2x3 double]
```

```
 [2x2 double]
```

```
>> A{1}
```

```
ans =
```

```
 1  2  3
```

```
 4  5  6
```

```
>> A{2}
```

```
ans =
```

```
 7  8
```

```
 9  0
```

```
>> A{1}(1,:)
```

```
ans =
```

```
 1  2  3
```

```
>> A{1}(A{2}(2,:) < 5,:)
```

```
ans =
```

```
 4  5  6
```

Prezentacja udostępniona na licencji **Creative Commons: Uznanie autorstwa, Na tych samych warunkach 3.0.** Pewne prawa zastrzeżone na rzecz autorów. Zezwala się na dowolne wykorzystywanie treści pod warunkiem wskazania autorów jako właścicieli praw do prezentacji oraz zachowania niniejszej informacji licencyjnej tak długo, jak tylko na utwory zależne będzie udzielana taka sama licencja. Tekst licencji dostępny jest na stronie:

<http://creativecommons.org/licenses/by-sa/3.0/deed.pl>

