



Πολυτεχνείο Κρήτης
Σχολή Μηχανικών Περιβάλλοντος

Επιστημονικός Προγραμματισμός

8η Εβδομάδα
Πίνακες

1

Πίνακες

- Είναι σύνολα ομοειδών στοιχείων
- Πολλές εφαρμογές μηχανικών χρησιμοποιούν πίνακες
- Το MATLAB αποθηκεύει τα δεδομένα σε πίνακες και εκτελεί όλους τους υπολογισμούς με πράξεις πινάκων.
- Έτσι, για την αποτελεσματική χρήση του MATLAB χρειάζεται ο χρήστης να έχει κατανοήσει τους πίνακες και τις πράξεις πινάκων

2

Μονοδιάστατοι Πίνακες: Διανύσματα

- Ένα διάνυσμα είναι ένας μονοδιάστατος πίνακας
- Παράδειγμα:
- Ένα διάνυσμα γραμμή με 4 στοιχεία $x = [0 \ -1.5 \ 4 \ 7]$
- Ένα διάνυσμα στήλη με 3 στοιχεία $y = \begin{bmatrix} 5 \\ 2.9 \\ 3 \end{bmatrix}$
- $x(3) = 4$ (3^ο στοιχείο του διανύσματος x)
- $y(2) = 2.9$ (2^ο στοιχείο του διανύσματος y)

7-Apr-19

Επιστημονικός Προγραμματισμός

3

3

Δισδιάστατοι Πίνακες (Μητρώα)

- Ένας πίνακας έχει δύο διαστάσεις
- Παράδειγμα:

$$A = \begin{bmatrix} 2.3 & 4.5 & -7.5 \\ 3.2 & -5.1 & 10 \end{bmatrix} \quad B = \begin{bmatrix} 75 & 43 & 79 & 38 \\ 92 & 36 & 37 & 84 \\ 27 & 96 & 56 & 32 \end{bmatrix}$$

Πίνακας 2 x 3

Πίνακας 3 x 4

- $A(2,2) = -5.1$ (στοιχείο στην 2^η γραμμή και 2^η στήλη)
- $B(1,4) = 38$ (στοιχείο στην 1^η γραμμή και 4^η στήλη)

7-Apr-19

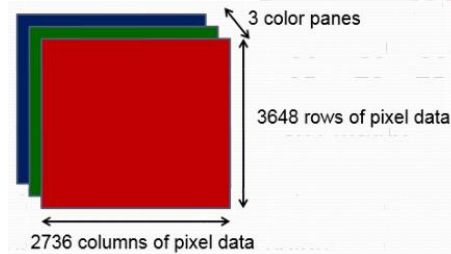
Επιστημονικός Προγραμματισμός

4

4

Πίνακες Περισσότερων Διαστάσεων

- Οι πίνακες μπορούν να έχουν και περισσότερες διαστάσεις
- Για παράδειγμα μια εικόνα jpg 3648 x 2736 pixel μπορεί να εισαχθεί στο MATLAB ως ένας πίνακας με 3 διαστάσεις με 3648 x 2736 x 3 όπου η τρίτη διάσταση αντιπροσωπεύει τις τιμές RGB κάθε στοιχείου



7-Apr-19

Επιστημονικός Προγραμματισμός

5

5

Πίνακες στο MATLAB

Δημιουργία και Επεξεργασία

7-Apr-19

Επιστημονικός Προγραμματισμός

6

6

Δημιουργία Πινάκων

```
>> a = [2.5 4.6 3.5 7]
a =

    2.50    4.60    3.50    7.00

>> b = [2.5; 4.6; 3.5; 7]
b =

    2.5000
    4.6000
    3.5000
    7.0000

>> c=[7.2 5.4 3.5 8; 4.2 2.8 3.2 9]
c =

    7.2000    5.4000    3.5000    8.0000
    4.2000    2.8000    3.2000    9.0000

>> d=[7.2 5.4 3.5; 4.2 2.8 3.2 9]
Error using vertcat
Dimensions of matrices being
concatenated are not consistent.
```

7-Apr-19

Επιστημονικός Προγραμματισμός

7

7

Δημιουργία Πινάκων

Συνάρτηση	Περιγραφή
A:INT:B	Δημιουργεί ένα διάνυσμα γραμμή με τιμές που ξεκινούν από A, αυξάνονται με βήμα INT, και δεν υπερβαίνουν το B
linspace(A, B, N)	Δημιουργεί ένα διάνυσμα γραμμή με N στοιχεία ισοκατανεμημένα μεταξύ A και B
eye(n)	Δημιουργεί έναν μοναδιαίο πίνακα n x n
zeros(m,n)	Δημιουργεί έναν πίνακα μηδενικών m x n
ones(m,n)	Δημιουργεί έναν πίνακα με άσους m x n

Προσοχή: οι εντολές zeros(n) και ones(n) παράγουν n x n πίνακες, όχι διανύσματα!

7-Apr-19

Επιστημονικός Προγραμματισμός

8

8

Δημιουργία Πινάκων

Συνάρτηση	Περιγραφή
<code>randi([imin imax], [m,n])</code>	Δημιουργεί έναν πίνακα $m \times n$ με ομοιόμορφα κατανεμημένες ακέραιες τιμές από <code>imin</code> έως <code>imax</code>
<code>randn(m,n)</code>	Δημιουργεί έναν πίνακα $m \times n$ με τυχαίους αριθμούς μιας κατανομής με μέσο όρο 0 και τυπική απόκλιση 1
<code>rand(m,n)</code>	Δημιουργεί έναν πίνακα $m \times n$ με ομοιόμορφα κατανεμημένες τιμές στο διάστημα <code>[0,1]</code>
<code>randperm(m,n)</code>	Δημιουργεί ένα διάνυσμα n μοναδικών ακεραίων από τις τιμές 1, 2, ..., m
Προσοχή: <code>rng('shuffle')</code> για αλλαγή των τιμών <code>rand</code>, <code>randi</code> & <code>randn</code>	

7-Apr-19

Επιστημονικός Προγραμματισμός

9

9

Δείκτες Διανυσμάτων

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

Το `x` είναι ένα διάνυσμα με 8 τιμές. Για να τραβήξουμε μια συγκεκριμένη τιμή, πρέπει να δώσουμε το δείκτη της τιμής

- Στο MATLAB οι δείκτες ξεκινούν από το 1 όχι από το 0, όπως σε άλλες γλώσσες προγραμματισμού
- `x(1)` είναι το 1^ο στοιχείο του διανύσματος `x`, `x(2)` είναι το δεύτερο στοιχείο του διανύσματος `x`, κ.ο.κ

7-Apr-19

Επιστημονικός Προγραμματισμός

10

10

Δείκτες Διανυσμάτων

- `x(3:7)` είναι ένα διάνυσμα που περιλαμβάνει από το 3^ο έως το 7^ο στοιχείο του `x`. Δίνει σφάλμα αν το `x` έχει λιγότερα από 7 στοιχεία
- Για να βρούμε πόσα στοιχεία έχει το διάνυσμα `x`
`>> N = length(x)`

7-Apr-19

Επιστημονικός Προγραμματισμός

11

11

Παράδειγμα

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



```
>> x(5)
```

```
ans =
```

```
8
```

```
>> x(3:7)
```

```
ans =
```

```
6 2 8 6 1
```

```
>> x(1:2:8)
```

```
ans =
```

```
3 6 8 1
```

```
>> x([1 3 6])
```

```
ans =
```

```
3 6 6
```

7-Apr-19

Επιστημονικός Προγραμματισμός

12

12

Δείκτες Πινάκων

Έστω πίνακας A στο MATLAB με τουλάχιστον 4 γραμμές και τουλάχιστον 7 στήλες

- A(2,6) είναι το στοιχείο στη γραμμή 2, στήλη 6 του A
- A(2,:) είναι ένα διάνυσμα που αποτελείται από ολόκληρη τη δεύτερη γραμμή του πίνακα A
- A(:,3) είναι ένα διάνυσμα που αποτελείται από ολόκληρη την τρίτη στήλη του πίνακα A
- A(2:3, 2:5) είναι ένας 2 x 4 πίνακας με τα στοιχεία του πίνακα A που βρίσκονται στις γραμμές 2 και 3 και στις στήλες 2, 3, 4 και 5
- Για να βρω αριθμό γραμμών και στηλών >> [rows, cols] = size(A)

7-Apr-19

Επιστημονικός Προγραμματισμός

13

13

Παράδειγμα Πίνακα

```
>> pinakas = randi([0 10], [4 7])
pinakas =
     6     10     10     4     7     7     7
     1     10     5     10     0     8     1
     3      1     8     8     9     8     7
     6     10     1     10    10     4     0
>> pinakas(2,6)
ans =
     8
>> pinakas(:,3)
ans =
    10
     5
     8
     1
>> pinakas(2,3:4)
ans =
     5     10
     8      8
     1     10
>> pinakas(2:4,3:6)
ans =
     5     10     0     8
     8      8     9     8
     1     10    10     4
```

7-Apr-19

Επιστημονικός Προγραμματισμός

14

14

Χρήσιμες Συναρτήσεις Δεικτών

Συνάρτηση	Περιγραφή
size(A)	Δίνει τις διαστάσεις του πίνακα. Για πίνακα 2 διαστάσεων δίνει πρώτα αριθμό των γραμμών και μετά τον αριθμό των στηλών
length(A)	Δίνει τη μεγαλύτερη διάσταση ενός πίνακα. Είναι πολύ χρήσιμη για διανύσματα, γιατί δίνει τον αριθμό των στοιχείων

7-Apr-19

Επιστημονικός Προγραμματισμός

15

15

Βρόχοι και Διανύσματα / Πίνακες

7-Apr-19

Επιστημονικός Προγραμματισμός

16

16

Βρόχοι και Διανύσματα

- Τι θα δώσει ο παρακάτω κώδικας;

```
x = [3 -5 6 2 -8 6 -1 4];
for k = 1: length(x)
    if x(k) < 0
        x(k) = 0;
    end
end
x
x =
    3     0     6     2     0     6     0     4
```

- $k=1$ $x(1) < 0$; Όχι
- $k=2$ $x(2) < 0$; Ναι $x(2)=0$
- $k=3$ $x(3) < 0$; Όχι
- $k=4$ $x(4) < 0$; Όχι
- $k=5$ $x(5) < 0$; Ναι $x(5)=0$
- $k=6$ $x(6) < 0$; Όχι
- $k=7$ $x(7) < 0$; Ναι $x(7)=0$
- $k=8$ $x(8) < 0$; Όχι

7-Apr-19

Επιστημονικός Προγραμματισμός

17

17

Βρόχοι και Πίνακες

- Τι θα δώσει ο παρακάτω κώδικας;

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = 5;
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

18

18

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

19

$r = 1$ και $c = 1$
 $A(1,1) < 0$;
 Όχι
 $A(1,1) > 5$;
 Όχι
 Νέος Πίνακας B
 $B(1,1) = 3$
 B =
 3

19

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

20

$r = 1$ και $c = 2$
 $A(1,2) < 0$;
 Ναι
 $B(1,2) = 0$
 B =
 3 0

20

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

21

$r = 1$ και $c = 3$

$A(1,3) < 0$;

Όχι

$A(1,3) > 5$;

Ναι

$B(1,3) = 5$

B =
3 0 5

21

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

22

$r = 2$ και $c = 1$

$A(2,1) < 0$;

Όχι

$A(2,1) > 5$;

Όχι

$B(2,1) = 2$

B =
3 0 5
2 0 0

22

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

23

$r = 2$ και $c = 2$

$A(2,2) < 0$;

Ναι

$B(2,2) = 0$

B =

3	0	5
2	0	0

23

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

24

$r = 2$ και $c = 3$

$A(2,3) < 0$;

Όχι

$A(2,3) > 5$;

Ναι

$B(2,3) = 5$

B =

3	0	5
2	0	5

24

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

25

$r = 3$ και $c = 1$

$A(3,1) < 0$;

Ναι

$B(3,1) = 0$

B =

3	0	5
2	0	5
0	0	0

25

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

26

$r = 3$ και $c = 2$

$A(3,2) < 0$;

Όχι

$A(3,2) > 5$;

Όχι

$B(3,2) = 4$

B =

3	0	5
2	0	5
0	4	0

26

Βρόχοι και Πίνακες

```
A = [3 -5 6; 2 -8 6; -1 4 5];
[numr numc] = size(A);
for r = 1: numr
    for c = 1: numc
        if A(r,c) < 0
            B(r,c) = 0;
        elseif A(r,c) > 5
            B(r,c) = 5;
        else
            B(r,c) = A(r,c);
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

$r = 3$ και $c = 3$

$A(3,3) < 0$;

Όχι

$A(3,3) > 5$;

Όχι

$B(3,3) = 5$

B =

3	0	5
2	0	5
0	4	5

27

27

Δήλωση Χώρου Μνήμης

- Στο προηγούμενο πρόγραμμα, ο πίνακας B δημιουργήθηκε μέσα στους βρόχους και μεγάλωνε σταδιακά. Όταν ένας πίνακας αλλάζει διάσταση στο MATLAB η διαρκής δήλωση της αλλαγής του χώρου μνήμης είναι χρονοβόρα
- Αν γνωρίζετε από την αρχή το μέγεθος ενός πίνακα καλό είναι να το δηλώνετε π.χ. με τη δημιουργία ενός πίνακα γεμάτου κάποιες τιμές (συχνά μηδενικά) στις διαστάσεις που γνωρίζετε

7-Apr-19

Επιστημονικός Προγραμματισμός

28

28

Δήλωση Χώρου Μνήμης

```
A = [3 -5 6; 2 -8 6; -1 4 5];
B = zeros(size(A)); % Δημιουργία πίνακα B με
[numr numc] = size(A); % 0 σε όλα τα στοιχεία
for r = 1: numr
    for c = 1: numc
        if A(r,c)<0
            B(r,c)=0;
        elseif A(r,c)>5
            B(r,c)=5;
        else
            B(r,c)=5;
        end
    end
end
B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

29

29

Λογικοί και Σχεσιακοί Τελεστές με Πίνακες

7-Apr-19

Επιστημονικός Προγραμματισμός

30

30

Λογικοί και Σχεσιακοί Τελεστές

Τελεστής	Περιγραφή
$A == B$	Έλεγχος ανά στοιχείο. Δίνει 1 εάν $a_{ij} == b_{ij}$, αλλιώς 0
$A > B$	Έλεγχος ανά στοιχείο. Δίνει 1 εάν $a_{ij} > b_{ij}$, αλλιώς 0
$A < B$	Έλεγχος ανά στοιχείο. Δίνει 1 εάν $a_{ij} < b_{ij}$, αλλιώς 0
$A \& B$	Έλεγχος ανά στοιχείο λογικό ΚΑΙ Δίνει 0 εάν είτε το a_{ij} είτε το b_{ij} είναι 0. Αλλιώς 1
$A B$	Έλεγχος ανά στοιχείο λογικό ΚΑΙ Δίνει 0 εάν και το a_{ij} και το b_{ij} είναι 0. Αλλιώς 1

Οι A και B πρέπει να έχουν ίδιες διαστάσεις εκτός και αν ένα από αυτά είναι ένας αριθμός μόνο (όχι πίνακας / διάνυσμα)
Θυμηθείτε 0 = FALSE και Μη-Μηδενικό = TRUE

7-Apr-19

Επιστημονικός Προγραμματισμός

31

31

Παράδειγμα

```
>> A = randi([-10 10], [3 5])
```

```
A =
```

```

     3    -5    -7    10    -6
     4     4    -8    -3     5
     5     3     0     2    -5
```

```
>> A > 5
```

```
ans =
```

```

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

7-Apr-19

Επιστημονικός Προγραμματισμός

32

32

Παράδειγμα

```
>> A = randi([0 3], [2 2])
A =
     3     0
     1     1

>> B = randi([0 3], [2 2])
B =
     2     1
     1     3

>> A == B
ans =
     0     0
     1     0
```

7-Apr-19

Επιστημονικός Προγραμματισμός

33

33

Παράδειγμα

```
>> A = randi([0 3], [2 2])
A =
     3     0
     1     1

>> B = randi([0 3], [2 2])
B =
     2     1
     1     3

>> A > B
ans =
     1     0
     0     0
```

7-Apr-19

Επιστημονικός Προγραμματισμός

34

34

Παράδειγμα

```
>> A = randi([0 3], [2 2])
A =
     3     0
     1     1

>> B = randi([0 3], [2 2])
B =
     2     1
     1     3

>> A >= B
ans =
     1     0
     1     0
```

7-Apr-19

Επιστημονικός Προγραμματισμός

35

35

Κείμενο στο MATLAB

Strings & Cells

7-Apr-19

Επιστημονικός Προγραμματισμός

36

36

Δείκτες σε μια μεταβλητή κειμένου

- Κάθε γράμμα αποθηκεύεται σαν ξεχωριστό στοιχείο ενός κανονικού πίνακα

```
>> onoma='MariaPapadaki'
```

```
onoma = MariaPapadaki
```

```
>> onoma(1)
```

```
ans = M
```

```
>> onoma(4)
```

```
ans = i
```

```
>> onoma(15)
```

```
Index exceeds matrix dimensions.
```

7-Apr-19

Επιστημονικός Προγραμματισμός

37

37

Πίνακες τύπου Cell

- Οι απλοί πίνακες δεν συμπεριφέρονται καλά με κείμενα, οι πίνακες τύπου cell είναι καλύτεροι
- Οι πίνακες τύπου cell δουλεύουν εξαιρετικά με μεταβλητές κειμένου και με διαφορετικούς τύπους μεταβλητών
- Για τους πίνακες cell χρησιμοποιούνται άγκιστρα {} αντί για αγκύλες []. Πέρα από αυτό, οι δείκτες δουλεύουν με τον ίδιο τρόπο

7-Apr-19

Επιστημονικός Προγραμματισμός

38

38

Παράδειγμα Πίνακα Cell

```
>> mines = {'Φεβρουάριος' 'Μάρτιος' 'Απρίλιος' 'Μάιος'}
mines =
    'Φεβρουάριος' 'Μάρτιος' 'Απρίλιος' 'Μάιος'

>> mines{1}
ans = 'Φεβρουάριος'

>> mines{1}
ans = Φεβρουάριος

>> mines{4}
ans = Μάιος
```

7-Απρ-19

Επιστημονικός Προγραμματισμός

39

39

Παράδειγμα Πίνακα Cell

```
>> strcmp(mines, 'Μάρτιος') % Σύγκριση Κειμένου
ans =
    0    1    0    0

>> strcmp(mines, 'μάρτιος')
ans =
    0    0    0    0

>> strcmpi(mines, 'μάρτιος')
ans =
    0    1    0    0

>> strncmp(mines, 'Μά', 2)
ans =
    0    1    0    1
```

7-Απρ-19

Επιστημονικός Προγραμματισμός

40

40

Σχόλια

- Οι απλοί πίνακες είναι πολύ βολικοί για ένα μεγάλο πλήθος μαθηματικών υπολογισμών και για πολλές εφαρμογές προγραμματισμού
- Όλα τα στοιχεία σε έναν απλό πίνακα πρέπει να είναι του ίδιου τύπου. Δεν μπορείς να έχεις κάποια στοιχεία με τιμές διπλής ακρίβειας και κάποια με ακέραιες τιμές
- Ένας πίνακας τύπου cell επιτρέπει ανάμιξη διαφορετικών ειδών μεταβλητών με διαφορετικά μεγέθη. Είναι χρήσιμο για μεταβλητές με κείμενο, και για αποθήκευση μεταβλητών με κείμενο και μεταβλητών με αριθμούς μαζί

7-Apr-19

Επιστημονικός Προγραμματισμός

41

41

Εξάσκηση

- Δοκιμάστε τις παρακάτω εντολές

```
>> x = [8 4 -5 2 -3]
>> x(3)
>> x(10)
>> x(2:4)
>> x([1 4])
>> N = length(x)
>> x(2:4) = [1 2 3]
```

7-Apr-19

Επιστημονικός Προγραμματισμός

42

42

Εξάσκηση

- Δοκιμάστε τις παρακάτω εντολές

```
>> A = [8 4 -5 2; -3 5 9 4; 5 6 1 -7]
>> A(3,2)
>> A(1,6)
>> A(2,:)
>> A(:,1)
>> A(2:3,3:4)
>> [numr numc] = size(A)
>> A(2,2:3) = [0 0]
```

7-Apr-19

Επιστημονικός Προγραμματισμός

43

43

Εξάσκηση

- Δοκιμάστε τις παρακάτω εντολές

```
>> A = [8 4 -5 2; -3 5 9 4]
>> B = [5 -8 2 4; 5 6 1 -7]
>> A > 0
>> A == 0
>> A == B
>> A > B
```

7-Apr-19

Επιστημονικός Προγραμματισμός

44

44

Χρήσιμες Συναρτήσεις για Πίνακες

sum, max, find, etc.

7-Apr-19

Επιστημονικός Προγραμματισμός

45

45

Συνάρτηση	Περιγραφή
$\min(A)$	Αν A διάνυσμα δίνει το ελάχιστο. Αν A πίνακας δίνει διάνυσμα με το ελάχιστο κάθε στήλης
$\max(A)$	Αν A διάνυσμα δίνει το μέγιστο. Αν A πίνακας δίνει διάνυσμα με το μέγιστο κάθε στήλης
$\min(A,B)$	Πίνακας με τα ελάχιστα από κάθε a_{ij} , b_{ij} . Οι A και B πρέπει να έχουν ίδιες διαστάσεις
$\max(A,B)$	Πίνακας με τα μέγιστα από κάθε a_{ij} , b_{ij} . Οι A και B πρέπει να έχουν ίδιες διαστάσεις
$\text{sum}(A)$	Αν A διάνυσμα δίνει το άθροισμα. Αν A πίνακας δίνει διάνυσμα με το άθροισμα κάθε στήλης
$\text{find}()$	Βρίσκει τη θέση όλων των στοιχείων ενός διανύσματος που ικανοποιούν τη συνθήκη στην ()

7-Apr-19

Επιστημονικός Προγραμματισμός

46

46

Συνάρτηση sum

```
>> x = [2 4 7 -3 5]

>> total = sum(x) % Αθροίζει όλα τα στοιχεία του x
total = 15

>> sum(x>0) % Η x>0 παράγει ένα διάνυσμα [1 1 1 0 1]
ans = 4 % n sum(x>0) δείχνει πόσα στοιχεία του x > 0

>> nzerofour = sum(0 < x & x < 4)
nzerofour = 1
```

ΠΟΤΕ μη γράψετε >> sum=sum(x)
Διόρθωση >> clear sum

7-Apr-19

Επιστημονικός Προγραμματισμός

47

47

Συνάρτηση sum σε πίνακες

A = 14	80	3	68	39	71	4	70
42	96	85	76	66	3	9	32
92	66	94	75	17	27	83	95

```
>> Csums=sum(A) % Αθροισμα στηλών
Csums = 148 242 182 219 122 101 96 197

>> Rsums=sum(A,2) % Αθροισμα γραμμών
Rsums = 349
        409
        549

>> Tsum=sum(sum(A))
Tsum = 1307
```

7-Apr-19

Επιστημονικός Προγραμματισμός

48

48

Συνάρτηση max

```
>> x = [2 4 7 7 -3 5]
```

```
>> megisto = max(x)
```

```
megisto = 7
```

```
>> [megisto thesi] = max(x)
```

```
megisto = 7
```

```
thesi = 3
```

Προσοχή: Δίνει μόνο την πρώτη θέση ενός μέγιστου

7-Apr-19

Επιστημονικός Προγραμματισμός

49

49

Συνάρτηση max σε πίνακες

A = 14	80	3	68	39	71	4	70
42	96	85	76	66	3	9	32
92	66	94	75	17	27	83	95

```
>> megista=max(A) % Μέγιστο κάθε στήλης
```

```
megista =
```

92	96	94	76	66	71	83	95
----	----	----	----	----	----	----	----

```
>> megisto=max(max(A)) % Μέγιστο στοιχείο πίνακα
```

```
megisto =
```

```
96
```

7-Apr-19

Επιστημονικός Προγραμματισμός

50

50

Συνάρτηση find

- Βρίσκει τους δείκτες όλων των στοιχείων ενός πίνακα που ικανοποιούν μια συνθήκη
- Παράδειγμα:

```
>> x = [2 4 7 7 -3 5]
x =
     2     4     7     7    -3     5
>> theseis5 = find(x==5)
theseis5 =     6
>> theseis7 = find(x==7)
theseis7 =     3     4
>> pano_apo4 = find(x>4)
pano_apo4 =     3     4     6
>> find(x>3 & x<7)
ans =         2     6
```

7-Apr-19

Επιστημονικός Προγραμματισμός

51

51

Συνδυασμός max και find

```
>> x = [2 4 7 7 -3 5]
x =     2     4     7     7    -3     5
>> megisto = max(x)
megisto =     7
>> thesimeg = find(x==megisto)
thesimeg =     3     4
```

Εδώ δίνει όλες τις θέσεις με τιμές ίσες με το μέγιστο

7-Apr-19

Επιστημονικός Προγραμματισμός

52

52

Εξάσκηση

- Δοκιμάστε τις παρακάτω εντολές στο MATLAB

```
>> A = [-1  5  7  3  6  10  5]
>> elaxisto = min(A)
>> sum(A == 5)
>> find(A == 5)
>> sum(A > 3 & A < 10)
>> find(A > 3 & A < 10)
```

7-Apr-19

Επιστημονικός Προγραμματισμός

53

53

Εξάσκηση

- Δοκιμάστε τις παρακάτω εντολές στο MATLAB

```
>> A = [-1  5  7;  6  10  5]
>> min(A)
>> min(A, [], 2)
>> min(min(A))
>> sum(A==5)
>> sum(A==5, 2)
>> sum(sum(A > 3 & A < 10))
>> [row col] = find(A > 3 & A < 10)
```

7-Apr-19

Επιστημονικός Προγραμματισμός

54

54



Πολυτεχνείο Κρήτης
Σχολή Μηχανικών Περιβάλλοντος

Διανύσματα και πίνακες

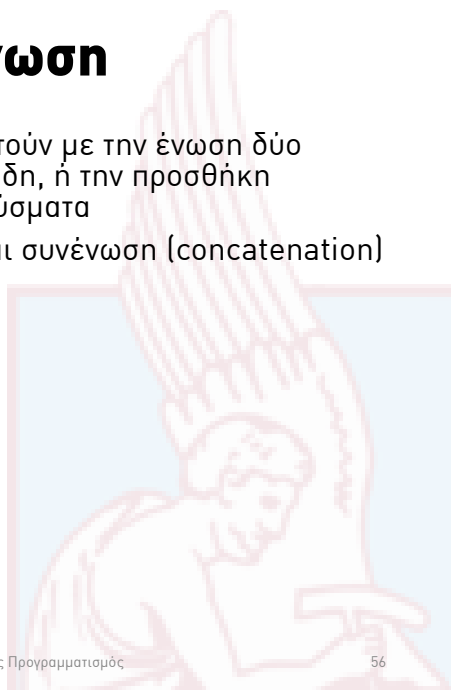
Μέρος 2^ο

55

Συνένωση

- Διανύσματα μπορούν να οριστούν με την ένωση δύο διανυσμάτων που υπάρχουν ήδη, ή την προσθήκη στοιχείων σε υπάρχοντα διανύσματα
- Η διαδικασία αυτή ονομάζεται συνένωση (concatenation)
- Για παράδειγμα:

```
>> v = 2:5;
>> x = [33 11 2];
>> w = [v x]
w =
    2    3    4    5   33   11    2
>> neww = [v 44]
neww =
    2    3    4    5   44
```



56

Τροποποίηση διανυσμάτων

- Τα στοιχεία ενός διανύσματος μπορούν να τροποποιηθούν, π.χ.,
`vec(3) = 11`
- Ένα διάνυσμα μπορεί επίσης να επεκταθεί με αναφορά σε στοιχεία που δεν υπάρχουν ακόμα· αν υπάρχει κενό ανάμεσα στο τελευταίο υπάρχον στοιχείο του διανύσματος και στο νέο στοιχείο (ή στοιχεία), το κενό συμπληρώνεται με μηδενικά, π.χ.,
`>> vec = [3 9];`
`>> vec(4:6) = [33 2 7]`
`vec =`
`3 9 0 33 2 7`
- Η επέκταση διανυσμάτων δεν είναι πολύ καλή ιδέα, οπότε καλό θα ήταν να αποφεύγεται

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

57

57

Στοιχεία πινάκων

- Αν ο χρήστης θέλει να αναφερθεί σε ένα στοιχείο πίνακα, πρέπει να χρησιμοποιήσει το όνομα της μεταβλητής του πίνακα ακολουθούμενο από τον δείκτη της γραμμής και κατόπιν από τον δείκτη της στήλης, σε παρένθεση

```
>> mat = [1:3; 6 11 -2]
mat =
     1     2     3
     6    11    -2
>> mat(2,1)
ans =
     6
```

ΠΑΝΤΑ αναφορά πρώτα στη γραμμή και μετά στη στήλη

- Ο τρόπος αυτός ονομάζεται δεικτοδότηση με δείκτες (subscripted indexing)
- Επιπλέον, είναι δυνατή η αναφορά σε ένα υποσύνολο ενός πίνακα
 - Για αναφορά σε ολόκληρη τη m-οστή γραμμή: `mat(m,:)`
 - Για αναφορά σε ολόκληρη τη n-οστή στήλη: `mat(:,n)`

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

58

58

Δεικτοδότηση μητρώων

- Για αναφορά στην τελευταία γραμμή ή στήλη, χρησιμοποιείται η ενσωματωμένη δεσμευμένη λέξη **end**, π.χ., η **mat(end,m)** είναι η m-οστή τιμή της τελευταίας γραμμής
- Ο χρήστης μπορεί να τροποποιήσει ένα στοιχείο ή υποσύνολο ενός πίνακα χρησιμοποιώντας μια εντολή ανάθεσης τιμής
- Γραμμική δεικτοδότηση (linear indexing): Αν χρησιμοποιηθεί ένας μόνο δείκτης με έναν πίνακα, το MATLAB προσπελαύνει τον πίνακα στήλη-στήλη (σαν να τον «ξετυλίγει»).
 - Σημείωση: γενικά, αυτό είναι καλό να αποφεύγεται

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

59

59

Διαστάσεις πινάκων

- Υπάρχουν αρκετές συναρτήσεις για την εύρεση των διαστάσεων ενός διανύσματος ή πίνακα:
 - Η **length(vec)** επιστρέφει το πλήθος των στοιχείων ενός διανύσματος
 - Η **length(mat)** επιστρέφει τη μεγαλύτερη διάσταση (γραμμές ή στήλες) ενός πίνακα
 - Η **size** επιστρέφει το πλήθος των γραμμών και των στηλών ενός διανύσματος ή ενός πίνακα

Σημαντική σημείωση: οι τιμές αυτές πρέπει να ανατίθενται σε ξεχωριστές μεταβλητές: **[r c] = size(mat)**

 - Η **numel** επιστρέφει τον συνολικό αριθμό των στοιχείων που περιέχονται σε ένα διάνυσμα ή πίνακα
- Όταν γράφετε κώδικα, είναι προτιμότερο να καλύπτετε όλες τις περιπτώσεις: μην υποθέτετε ότι οι διαστάσεις ενός διανύσματος ή πίνακα είναι γνωστές – χρησιμοποιείτε τις συναρτήσεις **length** ή **size** για να προσδιορίζετε τις διαστάσεις!

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

60

60

Συναρτήσεις αλλαγής διαστάσεων

- Η **reshape** αλλάζει τις διαστάσεις ενός πίνακα σε έναν οποιονδήποτε πίνακα με το ίδιο πλήθος στοιχείων
- Η **rot90** περιστρέφει έναν πίνακα αριστερόστροφα (αντίθετα προς τη φορά των δεικτών του ρολογιού) κατά 90 μοίρες
- Η **fliplr** κατοπτρίζει τον πίνακα από αριστερά προς τα δεξιά (δηλαδή η πρώτη στήλη γίνεται τελευταία στήλη, κ.ο.κ.)
- Η **flipud** κατοπτρίζει τη πίνακας από πάνω προς τα κάτω (δηλαδή η πρώτη γραμμή γίνεται τελευταία γραμμή, κ.ο.κ.)
- Η **repmat** αναπαράγει έναν πίνακα· δημιουργεί $m \times n$ αντίγραφα του πίνακα

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

61

61

Κενά διανύσματα

- Ένα κενό διάνυσμα είναι ένα διάνυσμα στο οποίο δεν υπάρχει αποθηκευμένη καμία τιμή· ορίζεται με χρήση κενών αγκυλών []
- Για να διαγραφεί ένα στοιχείο ενός διανύσματος, μπορεί να ανατεθεί το κενό διάνυσμα σε αυτό το στοιχείο
- Για να διαγραφεί μια ολόκληρη γραμμή ή στήλη ενός πίνακα, μπορεί να ανατεθεί το κενό διάνυσμα [] στη γραμμή ή στήλη
Διαγραφή μεμονωμένων στοιχείων ενός πίνακα δεν είναι δυνατή
- «Σταδιακή δημιουργία» ενός διανύσματος: ο χρήστης ξεκινάει με το κενό διάνυσμα και προσθέτει στοιχεία σε αυτό (συνήθως με έναν βρόχο)
Δεν είναι αποδοτικό - χρήση μόνο όταν δεν υπάρχει άλλη επιλογή

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

62

62

Πίνακες ως ορίσματα συναρτήσεων

- Ολόκληροι πίνακες (διανύσματα ή πίνακες) μπορούν να μεταβιβαστούν ως ορίσματα σε συναρτήσεις· αυτή είναι μια πολύ χρήσιμη δυνατότητα!
- Το αποτέλεσμα θα έχει τις ίδιες διαστάσεις με το όρισμα εισόδου

- Για παράδειγμα:

```
>> vec = randi([-5 5], 1, 4)
vec =
    -3     0     5     1
>> av = abs(vec)
av =
     3     0     5     1
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

63

63

Συναρτήσεις πινάκων

- Υπάρχουν αρκετές χρήσιμες ενσωματωμένες συναρτήσεις που έχουν γραφεί ειδικά για να εφαρμόζονται σε διανύσματα ή σε στήλες πινάκων:

- min: η ελάχιστη τιμή
- max: η μέγιστη τιμή
- sum: το άθροισμα όλων των στοιχείων
- prod: το γινόμενο όλων των στοιχείων
- cumprod: σωρευτικό, ή τρέχον, γινόμενο
- cumsum: σωρευτικό, ή τρέχον, άθροισμα

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

64

64

sum, cumsum: παραδείγματα με διανύσματα

- Η συνάρτηση sum επιστρέφει το άθροισμα όλων των στοιχείων· η συνάρτηση cumsum υπολογίζει το τρέχον άθροισμα καθώς διατρέχει (προσθέτει) τα στοιχεία του διανύσματος (4, κατόπιν 4+-2, έπειτα 4-2+5, και τέλος 4-2+5+11)

```
>> vec = [4 -2 5 11];
>> sum(vec)
ans =
    18
>> cumsum(vec)
ans =
    4    2    7   18
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

65

65

prod, cumprod: παραδείγματα

- Αυτές οι συναρτήσεις έχουν την ίδια μορφή με τις sum/cumsum, αλλά υπολογίζουν γινόμενα

```
>> v = [2:4 10]
v =
    2    3    4   10
>> cumprod(v)
ans =
    2    6   24  240
>> mat = randi([1, 10], 2,4)
mat =
    2    2    5    8
    8    7    8   10
>> prod(mat)
ans =
   16   14   40   80
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

66

66

Συναρτήσεις diff και meshgrid

- Η diff επιστρέφει τις διαφορές μεταξύ των διαδοχικών στοιχείων ενός διανύσματος
- Η meshgrid δέχεται ως ορίσματα εισόδου δύο διανύσματα, και επιστρέφει ως ορίσματα εξόδου δύο πίνακες που ορίζουν χωριστά τις τιμές των x και y

```
>> [x y] = meshgrid(1:3,1:2)
```

```
• x =
```

```
•   1   2   3
```

```
•   1   2   3
```

```
• y =
```

```
•   1   1   1
```

```
•   2   2   2
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

67

67

Λογικά διανύσματα

- Το αποτέλεσμα της χρήσης σχεσιακών τελεστών με διανύσματα και πίνακες είναι ένα λογικό διάνυσμα ή πίνακας

```
>> vec = [44 3 2 9 11 6];
```

```
>> logv = vec > 6
```

```
logv =
```

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

- Το λογικό διάνυσμα μπορεί να χρησιμοποιηθεί για την δεικτοδότηση ενός διανύσματος ή πίνακα (μόνο αν το διάνυσμα δεικτών είναι τύπου logical)

```
>> vec(logv)
```

```
ans =
```

```
44   9  11
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

68

68

True/False (αληθές/ψευδές)

- false: ισοδύναμο με το logical(0)
- true: ισοδύναμο με το logical(1)
- Υπάρχουν και οι συναρτήσεις false και true, οι οποίες ορίζουν πίνακες που περιέχουν μόνο τιμές false ή true

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

69

69

Λογικές ενσωματωμένες συναρτήσεις

- Η any επιστρέφει την τιμή true αν οποιοδήποτε στοιχείο του ορίσματος εισόδου αναπαριστά την τιμή true
- Η all επιστρέφει την τιμή true μόνο αν όλα τα στοιχεία του ορίσματος εισόδου αναπαριστούν την τιμή true
- Η find βρίσκει θέσεις και επιστρέφει δείκτες


```
>> vec
vec =
    44    3    2    9   11    6
>> find(vec>6)
ans =
     1     4     5
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

70

70

Σύγκριση πινάκων

- Η συνάρτηση `isequal` συγκρίνει δύο πίνακες, και επιστρέφει την τιμή `logical true` αν οι πίνακες είναι ίσοι (δηλαδή αν όλα τα αντίστοιχα στοιχεία τους είναι ίσα) ή την τιμή `logical false` διαφορετικά

```
>> v1 = 1:4;
>> v2 = [1 0 3 4];
>> isequal(v1,v2)
ans =
    0
>> v1 == v2
ans =
     1     0     1     1
>> all(v1 == v2)
ans =
    0
```

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

71

71

Τελεστές που εφαρμόζονται ανά στοιχείο

- Οι τελεστές `|` και `&` χρησιμοποιούνται με πίνακες· συγκρίνουν ένα προς ένα τα στοιχεία και επιστρέφουν τιμές 1 ή 0 τύπου `logical`
- Οι τελεστές `||` και `&&` χρησιμοποιούνται με βαθμωτούς αριθμούς

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

72

72

Συνήθειες παγίδες

- Αποφεύγετε να ορίζετε πίνακες που δεν έχουν το ίδιο πλήθος στοιχείων σε κάθε γραμμή.
- Μην μπερδεύετε τον πολλαπλασιασμό πινάκων με τον πολλαπλασιασμό στοιχείο προς στοιχείο. Οι πράξεις πινάκων, συμπεριλαμβανομένου του πολλαπλασιασμού, της διαίρεσης και της ύψωσης σε δύναμη, που εκτελούνται ανά στοιχείο (συνεπώς, οι πίνακες πρέπει να έχουν το ίδιο μέγεθος)· έχουν τελεστές `.*`, `./`, `.\` και `.^`. Για να είναι εφικτός ο πολλαπλασιασμός πινάκων, οι εσωτερικές διαστάσεις πρέπει να συμφωνούν· ο αντίστοιχος τελεστής είναι ο `*`.
- Αποφεύγετε να χρησιμοποιείτε πίνακες με τιμές 1 και 0 τύπου `double` για δεικτοδότηση ενός άλλου πίνακα (υπενθυμίζεται ότι οι τιμές πρέπει να είναι τύπου `logical`).
- Αποφεύγετε να χρησιμοποιείτε τους τελεστές `||` και `&&` με πίνακες. Όταν δουλεύετε με πίνακες, πρέπει να χρησιμοποιείτε πάντα τους τελεστές `|` και `&`· οι τελεστές `||` και `&&` χρησιμοποιούνται μόνο με βαθμωτούς αριθμούς.

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

73

73

Οδηγίες σωστού προγραμματισμού

- Αν είναι εφικτό, προσπαθείτε να μην επεκτείνετε διανύσματα ή πίνακες, επειδή αυτό δεν είναι πολύ αποδοτικό.
- Μην χρησιμοποιείτε μόνο έναν δείκτη όταν αναφέρεστε στα στοιχεία ενός πίνακα· αντίθετα, πρέπει να χρησιμοποιείτε τους δείκτες γραμμής και στήλης (επιλέξτε την δεικτοδότηση με δείκτες, και όχι τη γραμμική δεικτοδότηση).
- Για να καλύπτετε όλες τις περιπτώσεις, μην υποθέτετε ποτέ ότι οι διαστάσεις οποιουδήποτε πίνακα (διανύσματος ή πίνακα) είναι γνωστές. Αντίθετα, πρέπει να χρησιμοποιείτε τις συναρτήσεις `length` ή `numel` για να προσδιορίζετε το πλήθος των στοιχείων ενός διανύσματος, και τη συνάρτηση `size` για να προσδιορίζετε τις διαστάσεις ενός πίνακα:


```
len = length(vec);
[r, c] = size(mat);
```
- Χρησιμοποιείτε τις τιμές `true` και `false` αντί για τις `logical(1)` και `logical(0)`, αντίστοιχα, ειδικά όταν ορίζετε διανύσματα ή πίνακες.

4-Apr-19

ΜΑΘ106 - Επιστημονικός Προγραμματισμός

74

74