

Στη συγκεκριμένη παρουσίαση στο Matlab θα βλέπουμε τα διανύσματα (όπως και τους πίνακες) ως εργαλεία αποθήκευσης όγκου πληροφοριών.

Διανύσματα:

- **Διανύσματα γραμμή:** Ορίζουμε στο Matlab διανύσματα με τον ακόλουθο τρόπο:

```
v=[1,2,3,4,6,3,1,2,10,0,102]
v=
     1  2  3  4  6  1  2 10  0 102
```

(την πρώτη γραμμή γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

- **Διανύσματα στήλη:** Ορίζουμε στο Matlab διανύσματα με τον ακόλουθο τρόπο:

```
v=[1,4,3]
v=
     1
     4
     3
```

(την πρώτη γραμμή γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

- **Ειδικές κατηγορίες διανυσμάτων:** Διανύσματα των οποίων τα στοιχεία βρίσκονται «με βήμα» έχουμε ήδη δει, σε άλλο πλαίσιο.

```
v=start:step:end
```

Για παράδειγμα:

```
v=1:2:9
v=
     1  3  5  7  9
```

(την πρώτη γραμμή γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

- **Ανάκτηση στοιχείου από διάνυσμα:** Δεδομένου ενός διανύσματος v , μπορεί να ανακτηθεί το στοιχείο του στην i -θέση:

```
v=[1,2,4];
v(3)
ans = 4
```

(τις πρώτες δύο γραμμές γράφουμε εμείς και την άλλη εμφανίζει το Matlab).

Πίνακες:

- **Πίνακες:** Τους πίνακες θα τους φανταζόμαστε ως διανύσματα στοιβαγμένα, ή ως διδιάστατα διανύσματα.

```
A=[1,2,3;0,0,0;4,5,6]
```

```
A=
```

```
1 2 3
0 0 0
4 5 6
```

(την πρώτη γραμμή γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

Σχόλιο: Προσέξτε ότι τα στοιχεία της γραμμής διαχωρίζονται με κόμμα, ενώ η αλλαγή στήλης σημαίνεται με το ερωτηματικό.

- **Ειδικές κατηγορίες πινάκων:** Αρχικά, υπάρχουν οι μηδενικοί πίνακες:

```
A=zeros(m,n)
```

Η παραπάνω εντολή κατασκευάζει έναν $m \times n$ μηδενικό πίνακα. Επίσης υπάρχουν οι ταυτοτικοί πίνακες:

```
A=eye(m,n)
```

Η παραπάνω εντολή κατασκευάζει έναν $m \times n$ ταυτοτικό πίνακα. Για παράδειγμα (γιατί δεν είναι όλοι οι ταυτοτικοί πίνακες τετραγωνικοί):

```
A=eye(2,3)
```

```
A=
```

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

- **Ανάκτηση στοιχείου από διάνυσμα:** Δεδομένου ενός πίνακα, μπορούμε να εξάγουμε το στοιχείο του στην (i, j) -θέση.\

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

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

```
ans = 3
```

(τις πρώτες δύο γραμμές γράφουμε εμείς και την άλλη εμφανίζει το Matlab).

Πράξεις με πίνακες και συναρτήσεις σε πίνακες:

Οι βασικές πράξεις με πίνακες (δείτε τις σημειώσεις της κ. Μητρούλη), της πρόσθεσης και του πολλαπλασιασμού, εκτελούνται με τα ίδια σύμβολα που χρησιμοποιούνται και στους πραγματικούς.

- **Πρόσθεση πινάκων:** (ανάλογα η αφαίρεση).

```
A=[1,2;3,4];
B=[0,1;1,2];
A+B
ans =
     1     3
     4     6
```

(τις πρώτες τρεις γραμμές γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

Η πρόσθεση πινάκων γίνεται κατά συντεταγμένη:

$$(a_{i,j})+(b_{i,j})=(a_{i,j}+b_{i,j})$$

- **Πολλαπλασιασμός πινάκων:**

```
A=[1,2;3,4;5,6];
B=[0,1,1;1,2,0];
A*B
ans =
     2     5     1
     4    11     3
     6    17     5
```

(τις πρώτες τρεις γραμμές γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

Ο πολλαπλασιασμός των πινάκων γίνεται ίσως με ανορθόδοξο τρόπο (κι αυτό για να μεταφράζεται η σύνθεση γραμμικών συναρτήσεων σε πολλαπλασιασμό πινάκων – περιμένετε μέχρι την γραμμική άλγεβρα I).

$$(a_{i,j})\cdot(b_{i,j})=(\sum_{k=1}^n a_{i,k}b_{k,j})$$

Δηλαδή το στοιχείο στη θέση (i,j) το βρίσκουμε πολλαπλασιάζοντας την i -γραμμή του πρώτου πίνακα με τη j -στήλη του δεύτερου.

Σχόλιο: Ο πολλαπλασιασμός $m \times n$ πίνακα με $n \times p$ πίνακα δίνει $m \times p$ πίνακα.

Επίσης ορίζεται η δύναμη ενός πίνακα, με A^n .

- **Κατά σημείο πράξεις:** Όσες πράξεις δεν είναι ήδη κατά σημείο, μπορούν να γίνουν κατά σημείο προθέτοντας μία τελεία. Για παράδειγμα:

```
A=[1,2;3,4];  
B=[1,0;0,2];  
A.*B  
ans =  
     1 0  
     0 8
```

(τις πρώτες τρεις γραμμές γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

- **Μέγεθος πίνακα:** Με τη συνάρτηση `size` μπορούμε να πάρουμε το μέγεθος ενός πίνακα.

```
A=[1,2;3,4;5,6];  
size(A)  
ans =  
     3 2
```

(τις πρώτες δύο γραμμές γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

Σχόλιο: Η `size` επιστρέφει πίνακα. Μπορεί να χρησιμοποιηθεί και σε διανύσματα (και δίνει πίνακα $[1, \text{μήκος διαν/τος}]$).

- **Συναρτήσεις μεγίστου/ελαχίστου:** Με τη συνάρτηση `max` (σε πίνακες) μπορεί να εξαχθεί το μέγιστο στοιχείο κάθε **στήλης** του πίνακα. Η εκχώρηση γίνεται σε διάνυσμα.

```
A=[1,2;3,4];  
max(A)  
ans =  
     3 4
```

(τις πρώτες δύο γραμμές γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

Αντίστοιχα, υπάρχει και η συνάρτηση του ελαχίστου `min`.

```
A=[1,2;3,4];  
min(A)  
ans=  
    1 2
```

(τις πρώτες δύο γραμμές γράφουμε εμείς και τις υπόλοιπες εμφανίζει το Matlab).

Για να βρούμε λοιπόν το μέγιστο στοιχείο ενός πίνακα πρέπει να γράψουμε:

```
max(max(A))
```

κι αντίστοιχα για το ελάχιστο:

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

Φυσικά καταλαβαίνετε ότι το Matlab (ως Matrix-Lab) εξειδικεύεται σε πίνακες κι έχει ακόμη πολλές δυνατότητες που δεν εξερευνήσαμε. Προς το παρόν, θα σταθούμε εδώ.

Σημείωση: Για τις επόμενες σημειώσεις, στη LaTeX, σκανάρετε το παρακάτω:

