4.2. Μητρώα, Πίνακες & Διανύσματα

4.2.1. Ορισμός Πινάκων

Κάθε μεταβλητή του MatLab είναι ένα μητρώο το οποίο μπορεί να ξεκινά από μηδενικές διαστάσεις (0x0) και να φτάνει μέχρι N-διάστατα μητρώα (20 x 10 x 3 x 200 x … x N). Το MatLab έχει δημιουργηθεί γύρω από αυτή τη φιλοσοφία και γι' αυτό είναι το ιδανικό εργαλείο για τη μελέτη σύνθετων και πολυδιάστατων επιστημονικών προβλημάτων. Αυτό όμως σημαίνει ότι ο χρήστης θα πρέπει να χειρίζεται με ευχέρεια τα μητρώα, τους δείκτες και τα στοιχεία τους, τις πράξεις τους, αλλά και τη βασική γραμμική άλγεβρα και τις εφαρμογές της.

Για να **ορίσουμε ένα πίνακα** στο MatLab χρησιμοποιούμε τις αγκύλες **[ ]**, και χωρίζουμε τα στοιχεία με κενά ή κόμμα (**,**) και τις γραμμές με το ερωτηματικό (**;**). Π.χ., δίνοντας την επόμενη εντολή δημιουργούνται οι 4 πίνακες:

```
A = [10], B = [2 4], C = [1 ; 3 ; 5], D = [2 4 8 ; 3 6 9]
```

$$A=\begin{bmatrix}10\end{bmatrix} \quad B=\begin{bmatrix}2 & 4\end{bmatrix} \quad C=\begin{bmatrix}1\\3\\5\end{bmatrix} \quad D=\begin{bmatrix}2 & 4 & 8\\3 & 6 & 9\end{bmatrix}$$

Για να **αναφερθούμε σε ένα στοιχείο** ή ένα τμήμα ενός πίνακα γράφουμε, το όνομα του πίνακα και αμέσως μετά γράφουμε σε παρένθεση, χωρισμένους με κόμματα (**(, ,)**) τους δείκτες που ορίζουν το τμήμα που θέλουμε. Π.χ., για τους παραπάνω πίνακες θα έχουμε:

B(2)=4, C(3)=5 & D(2,2)=6, και, αν δώσουμε την επόμενη εντολή δημιουργούμε 2 νέα διανύσματα από τον D:

```
D1 = D( 1 , [1 2]), D2 = D( [1 2] , 3)
```

$$D1=\begin{bmatrix}2 & 4\end{bmatrix}, \qquad D2=\begin{bmatrix}8\\9\end{bmatrix}$$

Στην παραπάνω εντολή ο ένας δείκτης αντικαταστάθηκε από μια λίστα τιμών [1 2] ώστε να επιλεγούν περισσότερα του ενός στοιχεία και να δημιουργηθεί το διάνυσμα. Στη συνέχεια θα δούμε πιο αναλυτικά πως ορίζουμε, δίνουμε τιμές και γενικά χειριζόμαστε τα μητρώα.

4.2.2. Ο Τελεστής Άνω-Κάτω Τελεία (:)

Για να χειριστούμε σωστά τα μητρώα στο MatLab πρέπει να καταλάβουμε τη χρήση των δεικτών και ειδικότερα του τελεστή **άνω-κάτω τελεία (:) ή colon operator**.

4.2.2.1. Τρόποι Σύνταξης

Ο τελεστής άνω-κάτω τελεία (**:**) χρησιμοποιείται γενικά όπου χρειάζεται να δηλωθεί μια λίστα ή μια σειρά από αριθμούς του τύπου: **από : έως**. Υπάρχουν οι εξής τρεις (3) βασικοί τρόποι σύνταξης του τελεστή:

1. Με 3 αριθμούς (A : B : C) που σημαίνει: όλοι οι αριθμοί από το A έως το C με βήμα B.
2. Με 2 αριθμούς (A : C) που σημαίνει: όλοι οι αριθμοί από το A έως το C με βήμα +1.
3. Χωρίς αριθμούς (:) που σημαίνει: όλες οι στήλες ή όλες οι γραμμές.

Δημιουργία λίστας ή διανύσματος. Μια πολύ συχνή χρήση του τελεστή είναι να δημιουργήσει ένα διάνυσμα με στοιχεία από μια λίστα αριθμών. Συντάσσεται με τους τρόπους 1 ή 2, ορίζοντας τα όρια των τιμών των αριθμών και προαιρετικά το βήμα τους. Π.χ.:

```
>> -3:3 ↵
ans =
    -3    -2    -1     0     1     2     3
>> x = -1:0.4:1 ↵
x =
   -1.0000   -0.6000   -0.2000    0.2000    0.6000    1.0000
>> y = [1:6 ; 2:7 ; 4:9] ↵
y =
     1     2     3     4     5     6
     2     3     4     5     6     7
     4     5     6     7     8     9
>> z = (9:-4:-9)*100+1 ↵
z =
   901   501   101  -299  -699
>> x = [0:0.1:pi]' ; y = sin(x); ↵
```

```
>> [x y]
ans =
         0         0
    0.1000    0.0998
    0.2000    0.1987
    0.3000    0.2955
    0.4000    0.3894
    0.5000    0.4794
    0.6000    0.5646
    0.7000    0.6442
    0.8000    0.7174
    0.9000    0.7833
    1.0000    0.8415
    1.1000    0.8912
    1.2000    0.9320
    1.3000    0.9636
    1.4000    0.9854
    1.5000    0.9975
    1.6000    0.9996
    1.7000    0.9917
    1.8000    0.9738
    1.9000    0.9463
    2.0000    0.9093
    2.1000    0.8632
    2.2000    0.8085
    2.3000    0.7457
    2.4000    0.6755
    2.5000    0.5985
    2.6000    0.5155
    2.7000    0.4274
    2.8000    0.3350
    2.9000    0.2392
    3.0000    0.1411
    3.1000    0.0416
```

4.2.2.2. Απόσπασμα ενός Πίνακα

Μια ακόμη συχνότερη χρήση του τελεστή είναι να ορίσει το τμήμα ενός πίνακα (υπο-πίνακα) που θα συμμετέχει σε μια πράξη ή ανάθεση. Ο τελεστής χρησιμοποιείται με οποιονδήποτε από τους τρεις τρόπους σύνταξης (1, 2 &3), για να δημιουργήσει τις λίστες των δεικτών του πίνακα που ορίζουν τον υπο-πίνακα που μας ενδιαφέρει.

Π.χ. δημιουργούμε τον παρακάτω πίνακα X (4x6) και επιλέγουμε διάφορα τμήματά του:

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


>> X(1:2:4, 2:2:6)
ans =
     2     4     6
     4     6     8
>> X(2,:) = X(1,:)+10
X =
     1     2     3     4     5     6
    11    12    13    14    15    16
     3     4     5     6     7     8
     4     5     6     7     8     9
```

4.2.2.3. Εντολές Επανάληψης

Η τρίτη βασική χρήση του τελεστή είναι η δημιουργία της λίστας για τον δείκτη της εντολής επανάληψης **FOR**. Η χρήση αυτή δεν διαφέρει από τις άλλες δυο σαν σύλληψη και συντάσσεται μόνο με τους τρόπους 1 ή 2. Παραδείγματα αυτής της χρήσης θα δούμε σε επόμενη παράγραφο για τις εντολές επανάληψης.

4.2.3. Αρχικοποίηση Πινάκων

Υπάρχουν πολλοί τρόποι για να ορίσουμε, να εισάγουμε ή να αρχικοποιήσουμε πίνακες στο MatLab:

1. Αναλυτικά, δίνοντας τη λίστα με τα στοιχεία του,
2. Παράγοντας τον από εσωτερικές συναρτήσεις του MatLab,
3. Δημιουργώντας τον σε ένα M-file,
4. Διαβάζοντας τον από ένα MAT-file,
5. Διαβάζοντας τον από ένα εξωτερικό αρχείο δεδομένων (ASCII), κλπ.

4.2.3.1 Αναλυτικά

Στο προηγούμενο κεφάλαιο είδαμε παραδείγματα από τους παραπάνω τρόπους σχηματισμού πινάκων. Εδώ θα δούμε πως τους χρησιμοποιούμε αποδοτικά για να δημιουργήσουμε πιο σύνθετους ή ειδικούς πίνακες που συναντάμε συχνά στα προβλήματά μας.

Ας ξεκινήσουμε δημιουργώντας ένα πίνακα Α(3x3) αναλυτικά από τα στοιχεία του. Τα δίνουμε γραμμή-γραμμή αρχίζοντας από τη πρώτη και βάζοντας (**;**) σε κάθε αλλαγή γραμμής.

Το (**;**) δεν χρειάζεται όταν γράψουμε κάθε γραμμή του πίνακα σε διαφορετική σειρά. Τα στοιχεία κάθε γραμμής του πίνακα τα χωρίζουμε, είτε με κενά είτε με κόμμα.

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

Όταν χρησιμοποιούμε πιο συχνά κάποιους πίνακες, τους γράφουμε σε ένα M-file και τους εισάγουμε από εκεί στην περιοχή εργασίας μας ή το πρόγραμμα μας. Στα M-file δεν χρειάζεται να συμπυκνώνουμε την εντολή δημιουργίας του πίνακα, αντίθετα, την κάνουμε πιο ευανάγνωστη και εύκολη στη διόρθωση. Το παρακάτω M-file **init1.m** δημιουργεί τον ίδιο πίνακα Α με την προηγούμενη εντολή στο CW.

M-file: **init1.m**

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

4.2.3.2 Συναρτήσεις Αρχικοποίησης

Όταν ο πίνακάς μας έχει ειδική μορφή τότε μπορούμε να γλιτώσουμε χρόνο, χώρο και λάθη, χρησιμοποιόντας τις δυνατότητες που μας δίνει το MatLab με τις κατάλληλες συναρτήσεις δημιουργίας πινάκων.

Οι συναρτήσεις **ones**, **zeros** & **eye** δημιουργούν πίνακες με μηδενικά και άσους σε όποια διάσταση επιθυμούμε, κομψά και αποδοτικά. Ο διαγώνιος μοναδιαίος πίνακας (100x100) που δημιουργεί η απλή εντολή eye(100) θα χρειαζόταν πάνω από 200 γραμμές κώδικα για να δηλωθεί αναλυτικά!

Συνάρτηση	Αποτέλεσμα
ones(m,n)	παράγει ένα (m x n) πίνακα με άσους
ones(n)	παράγει ένα (n x n) πίνακα με άσους
zeros(m,n)	παράγει ένα (m x n) πίνακα με μηδενικά
zeros(n)	παράγει ένα (n x n) πίνακα με μηδενικά
eye(n)	παράγει ένα (n x n) μοναδιαίο πίνακα, *I*

Την απλή χρήση των συναρτήσεων αυτών την έχουμε ήδη δει, αλλά υπάρχουν και άλλοι τρόποι χρήση τους που διευκολύνουν πολύ στο σχηματισμό μεγάλων πινάκων, όπως του πίνακα Β που ακολουθεί.

Ο πίνακας Β αποτελείται στην ουσία από 4 υπο-πίνακες: τον πίνακα Α (3x3), έναν Μηδενικό πίνακα (3x5), έναν Μηδενικό πίνακα (5x3), και έναν Μοναδιαίο πίνακα (5x5). Θα μπορούσαμε δηλαδή να δούμε τον Β σαν ένα πίνακα με 4 στοιχεία, 2 στη πρώτη γραμμή, τον Α και τον ένα Μηδενικό, και 2 στη δεύτερη γραμμή, τον άλλο Μηδενικό και τον Μοναδιαίο. Ο Β λοιπόν δημιουργείται στο MatLab απλούστατα με την παρακάτω εντολή:

$$B=\begin{bmatrix}1&2&3&0&0&0&0&0\\4&5&6&0&0&0&0&0\\7&8&9&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{bmatrix}=\begin{bmatrix}1&2&3&0&0&0&0&0\\4&5&6&0&0&0&0&0\\7&8&9&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{bmatrix}$$

```
>> B = [ A , zeros(3,5) ; zeros(5,3) , eye(5) ]
B =
     1     2     3     0     0     0     0     0
     4     5     6     0     0     0     0     0
     7     8     9     0     0     0     0     0
     0     0     0     1     0     0     0     0
     0     0     0     0     1     0     0     0
     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     1
```

Για block-διαγώνιους πίνακες όπως ο παραπάνω, το MatLab διαθέτει επίσης την ειδική συνάρτηση **blkdiag**() για τη δημιουργία τους. Ο πιο πάνω πίνακας B μπορεί να δημιουργηθεί με τη χρήση της συνάρτησης ως εξής:

```
>> B = blkdiag( A , eye(5) )
```

4.2.3.3 Ειδικοί Πίνακες

Το MatLab διαθέτει και συναρτήσεις για τη δημιουργία ειδικών πινάκων όπως είναι: η **pascal**(n) που δημιουργεί συμμετρικούς πίνακες Pascal, η **magic**(n) που δημιουργεί μαγικά τετράγωνα, ή, οι **rand**(n) και **randn**(n) που δημιουργούν πίνακες με τυχαίους αριθμούς. Π.χ.:

```
>> pascal(4)
ans =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
>> magic(4)
ans =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
>> rand(5)
ans =
    0.2286    0.9926    0.4222    0.1112    0.0269
    0.8620    0.3733    0.6604    0.5651    0.5195
    0.6566    0.5314    0.6737    0.9692    0.1923
    0.8912    0.1813    0.9573    0.0237    0.7157
    0.4881    0.5019    0.1919    0.8702    0.2507
```

Ο μοναδιαίος πίνακας δεν είναι ο μόνος διαγώνιος πίνακας που θα συναντήσουμε, απλά είναι ένας ειδικός πίνακας και γι' αυτό έχει τη δική του συνάρτηση. Για τη δημιουργία άλλων **διαγώνιων πινάκων**, χρησιμοποιούμε τη πιο γενική εντολή **diag**. Η diag μπορεί να τοποθετήσει ή να εξάγει οποιεσδήποτε τιμές σε οποιαδήποτε διαγώνιο ενός πίνακα και έχει δύο τύπους γραφής. Η σύνταξη 'πίνακας_X = diag(διάνυσμα_v, k)' δημιουργεί τον πίνακα X και τοποθετεί στη k διαγώνιο το διάνυσμα v. Η σύνταξη 'διάνυσμα_v = diag(πίνακας_X, k)' εξάγει από τον πίνακα X τη k διαγώνιο σα διάνυσμα v. Όταν το k έχει τη τιμή 0 μπορεί να παραληφθεί. Π.χ. :

```
>> v = [1 2 3];
>> X = diag(v)
X =
     1     0     0
     0     2     0
     0     0     3
>> X = diag(v,1)
X =
     0     1     0     0
     0     0     2     0
     0     0     0     3
     0     0     0     0
>> X = diag(v,-1)
X =
     0     0     0     0
     1     0     0     0
     0     2     0     0
     0     0     3     0
>> diag(X)
ans =
     1
     2
     3
```

Συνδυάζοντας την **diag** με τις άλλες συναρτήσεις ορισμού πινάκων δημιουργούμε εύκολα πιο σύνθετους πίνακες που θα απαιτούσαν αρκετές γραμμές εντολών. Η επόμενη εντολή δημιουργεί ένα **τρι-διαγώνιο πίνακα** (9x9) προσθέτοντας τρεις εντολές diag, μία για κάθε διαγώνιο:

```
>> m = 4;
>> diag(-m:m) + diag(ones(2*m,1),1) + diag(ones(2*m,1),-1)
ans =
    -4     1     0     0     0     0     0     0     0
     1    -3     1     0     0     0     0     0     0
     0     1    -2     1     0     0     0     0     0
     0     0     1    -1     1     0     0     0     0
     0     0     0     1     0     1     0     0     0
     0     0     0     0     1     1     1     0     0
     0     0     0     0     0     1     2     1     0
     0     0     0     0     0     0     1     3     1
     0     0     0     0     0     0     0     1     4
```

Για τη δημιουργία **τριγωνικών πινάκων**, το MatLab διαθέτει δυο συναρτήσεις τις **triu & tril** οι οποίες δημιουργούν άνω και κάτω τριγωνικούς πίνακες αντίστοιχα. Π.χ.:

```
>> triu(ones(4))
ans =
     1     1     1     1
     0     1     1     1
     0     0     1     1
     0     0     0     1
>> tril(ones(4),-1)
ans =
     0     0     0     0
     1     0     0     0
     1     1     0     0
     1     1     1     0
>> tril(ones(4),1)
ans =
     1     1     0     0
     1     1     1     0
     1     1     1     1
     1     1     1     1
```

Άλλοι ειδικού τύπου πίνακες είναι οι **ελλιπείς ή αραιοί (sparse) πίνακες**. Το MatLab διαθέτει ειδικέw εντολές-συναρτήσεις για το χειρισμό τους όπως είναι οι **sparse, spdiags, speye, sprand, spones, spconvert,** κλπ., τις οποίες θα δούμε σε ξεχωριστή παράγραφο.

Για τη δημιουργία διανυσμάτων με **στοιχεία που ισαπέχουν** μεταξύ τους χρησιμοποιούμε την **linspace**(). Η συνάρτηση συντάσσεται **linspace**(a,b.n), και δημιουργεί n ισαπέχοντα σημεία στο διάστημα [a, b] συμπεριλαμβανομένων των a & b. Τα διανύσματα αυτά είναι πολύ χρήσιμα στη σχεδίαση γραφικών παραστάσεων και κυρίως για τη δημιουργία τιμών για τις ανεξάρτητες μεταβλητές. Όταν η κλίμακα σχεδίασης δεν είναι γραμμική αλλά λογαριθμική τότε χρησιμοποιούμε την αντίστοιχη εντολή **logspace**() με ορίσματα τους εκθέτες του 10. Π.χ.:

```
>> linspace(1,9,5)
ans =
     1     3     5     7     9
>> logspace(1,5,5)
ans =
          10         100        1000       10000      100000
```

Τέλος, για τη δημιουργία ειδικών πινάκων, το MatLab διαθέτει μια μεγάλη συλλογή μέσω της συνάρτησης **gallery**() της οποίας ο πλήρης κατάλογος περιέχεται στα Παραρτήματα. Για παράδειγμα, η επόμενη εντολή επιστρέφει ένα πίνακα (5x5) του οποίου τα στοιχεία υπολογίζονται από τους δείκτες τους ως εξής: τιμή στοιχείου (i , j) = 1/(i+j).

```
>> gallery('cauchy',5)
ans =
    0.5000    0.3333    0.2500    0.2000    0.1667
    0.3333    0.2500    0.2000    0.1667    0.1429
    0.2500    0.2000    0.1667    0.1429    0.1250
    0.2000    0.1667    0.1429    0.1250    0.1111
    0.1667    0.1429    0.1250    0.1111    0.1000
```

4.2.4. Πράξεις Πινάκων

Οι πράξεις με πίνακες και διανύσματα είναι δυο ειδών: α) αυτές που εφαρμόζονται **στοιχείο-στοιχείο**, και, β) αυτές που εφαρμόζονται σε **ολόκληρα τα μητρώα**. Όταν μια πράξη, όπως π.χ., ο πολλαπλασιασμός, γίνεται και με τους δυο τρόπους οπότε για να διευκρινίσουμε ποιο αποτέλεσμα από τα δύο θέλουμε διαφοροποιούμε στο MatLab το σύμβολο της πράξης (*). Έτσι, με το κλασικό σύμβολο γίνεται η πράξη των μητρώων ενώ για να δηλώσουμε ότι η πράξη πρέπει να γίνει **στοιχείο-στοιχείο** βάζουμε την **τελεία** πριν

το σύμβολο της πράξης (π.χ.: **.***). Σημ.: οι πράξεις με **βαθμωτές** ποσότητες γίνονται πάντοτε στοιχείο-στοιχείο, είτε χρησιμοποιήσουμε τη τελεία είτε όχι.

Πράξη	Πράξη Στοιχείο-Στοιχείο	Πράξη Μητρώων
Πρόσθεση	+	
Αφαίρεση	-	
Πολλ/σμός	.*	*
Ύψωση σε Δύναμη	.^	^
Αριστ. Διαίρεση	./	/
Δεξιά Διαίρεση	.\	\
Αναστροφή		'
Αντιστροφή		**inv( )**
Ορίζουσα		**det( )**

4.2.4.1 Πρόσθεση & Αφαίρεση

Η **πρόσθεση** και η **αφαίρεση** πινάκων ή διανυσμάτων γίνεται πάντα στοιχείο-στοιχείο και πρέπει όλα τα μητρώα να έχουν τις ίδιες διαστάσεις. Στοιχείο-στοιχείο γίνεται και η πρόσθεση ή η αφαίρεση ενός πίνακα με μια βαθμωτή ποσότητα. Π.χ.:

```
>> A = [4 5 6];  ↵
>> B = [1 2 3];  ↵
>> C = A + B  ↵
C =
     5     7     9
>> D = A - B  ↵
D =
     3     3     3
>> ones(3) + 7  ↵
ans =
     8     8     8
     8     8     8
     8     8     8
>> zeros(2)-1  ↵
ans =
    -1    -1
    -1    -1
>> [1 2; 3 4] + [10 5; -1 0]  ↵
ans =
    11     7
     2     4
```

4.2.4.2 Πολλαπλασιασμός

Ο **πολλαπλασιασμός** δύο πινάκων είναι μια σύνθετη πράξη που απαιτεί να ταιριάζουν οι εσωτερικές διαστάσεις των δυο πινάκων. Συγκεκριμένα, η 2η διάσταση του 1ου πίνακα πρέπει να είναι ίση με τη 1η διάσταση του 2ου πίνακα, διαφορετικά θα προκύψει λάθος.

```
>> A = [2 4 1; 0 3 -2]  ↵
A =
     2     4     1
     0     3    -2
>> B = [2 0; -1 3; 5 2]  ↵
B =
     2     0
    -1     3
     5     2
>> A * B  ↵
ans =
     5    14
   -13     5

>> B * A  ↵
ans =
     4     8     2
    -2     5    -7
    10    26     1
>> C = [ 1 2 3 ]  ↵
C =
     1     2     3
>> D = [ 1; 2; 3]  ↵
D =
     1
     2
     3
>> C * D  ↵
ans =
    14
>> D * C  ↵
ans =
     1     2     3
     2     4     6
     3     6     9
```

Αν ζητείται ο **πολλαπλασιασμός δυο πινάκων στοιχείο-στοιχείο** τότε χρησιμοποιείται το (**.***) και οι δυο πίνακες πρέπει να έχουν τις ίδιες διαστάσεις. Η διαφορά του αποτελέσματος των δυο ειδών πολλαπλασιασμού φαίνεται στο επόμενο παράδειγμα.

```
>> [1 2; 3 4] .* [10 5; -1 0]  ↵
ans =
    10    10
    -3     0
>> [1 2; 3 4] * [10 5; -1 0]  ↵
ans =
     8     5
    26    15
```

Η **ύψωση σε δύναμη** γίνεται κατά κανόνα στοιχείο-στοιχείο και χρησιμοποιούμε το τελεστή (**.^**). Όταν ο πίνακας δεν είναι τετραγωνικός η πράξη πρέπει να γίνεται πάντοτε στοιχείο-στοιχείο.

```
>> A = [1 2 3; 4 5 6]  ↵
A =
     1     2     3
     4     5     6
>> A .^2  ↵
ans =
     1     4     9
    16    25    36
>> 2 .^A  ↵
ans =
     2     4     8
    16    32    64
```

Όταν ο πίνακας είναι τετραγωνικός η ύψωση σε δύναμη μπορεί να είναι είτε πράξη μητρώων είτε στοιχείο-στοιχείο.

```
>> B = [1 2; 3 4]  ↵
B =
     1     2
     3     4
>> B .^2  ↵
ans =
     1     4
     9    16
>> B ^2  ↵
ans =
     7    10
    15    22
>> B ^2.5  ↵
ans =
  15.9802 + 0.0644i  23.2900 - 0.0294i
  34.9350 - 0.0442i  50.9153 + 0.0202i
>> 2 .^B  ↵
ans =
     2     4
     8    16
>> 2 ^B  ↵
ans =
   10.4827   14.1519
   21.2278   31.7106
```

Όταν έχουμε συντάξει μια παράσταση για βαθμωτές μεταβλητές και θέλουμε να την εφαρμόσουμε σε στοιχεία πινάκων, τότε χρησιμοποιούμε τη συνάρτηση **vectorize**() η οποία μετατρέπει τις πράξεις (*****, **/**, **^**) σε πράξεις (**.***, **./**, **.^**) ώστε να εκτελούνται στοιχείο-στοιχείο.

```
>> vectorize('a*b+c/d-e^f*2')  ↵
ans =
a.*b+c./d-e.^f.*2
```

4.2.4.3 Διαίρεση

Στο MatLab έχουμε δυο διαιρέσεις,

την **κανονική (forward) ή αριστερή διαίρεση** (διαιρεταίος **/** διαιρέτης), και,

την **ανάποδη (backward) ή δεξιά διαίρεση** (διαιρέτης **** διαιρεταίος)

Για τις βαθμωτές ποσότητες οι δύο πράξεις είναι ταυτόσημες.

```
>> 5/2  ↵
ans =
    2.5000
>> 2\5  ↵
ans =
    2.5000
```

Αντίθετα, στην περίπτωση διαίρεσης πινάκων οι δύο διαιρέσεις διαφέρουν. Οι διαιρέσεις αυτές εφαρμόζονται συχνά στο πρόβλημα της επίλυσης **γραμμικών εξισώσεων** όπου διακρίνουμε δυο περιπτώσεις. Όπως θα δούμε και σε επόμενη παράγραφο, ανάλογα με η θέση

του πίνακα αγνώστων X, δηλ., αριστερά ή δεξιά από το πίνακα των συντελεστών A, η λύση θα δίνεται αντίστοιχα με την αριστερή ή με τη δεξιά διαίρεση. Οι δύο πράξεις είναι διαφορετικές αλλά συνδέεται η μία με την άλλη με την παρακάτω ισοδυναμία:

A / B = (B' \ A')'

4.2.4.4 Γινόμενο

Το **εσωτερικό γινόμενο** δυο διανυσμάτων μπορεί να υπολογιστεί με δυο τρόπους χρησιμοποιώντας τις πράξεις πινάκων που αναφέραμε πιο πάνω:

1. Πολλαπλασιάζουμε τα διανύσματα στοιχείο-στοιχείο και παίρνουμε το άθροισμα όλων των γινομένων,
2. Τα πολλαπλασιάζουμε σαν μητρώα (1xn)x(nx1) και το αποτέλεσμα είναι (1x1) βαθμωτό.
3. Χρησιμοποιούμε τη συνάρτηση **dot**(A,B).

```
>> A = [4 -2 3], B = [1 3 -1]
A =
     4    -2     3
B =
     1     3    -1
>> sum( A .* B)
ans =
    -5
>> sum( A' .* B')
ans =
    -5
>> A * B'
ans =
    -5
>> dot(A,B)
ans =
    -5
```

Το **εξωτερικό γινόμενο** δυο διανυσμάτων υλοποιείται τη συνάρτηση **cross**(A,B).

```
>> cross(A,B)
ans =
    -7     7    14
```

4.2.4.5 Αναστροφή

Η **αναστροφή πίνακα (A^T)** μετατρέπει τις γραμμές σε στήλες και τις στήλες σε γραμμές, περιστρέφοντάς τον γύρω από την κύρια διαγώνιο, και πραγματοποιείται με τον τελεστή (**'**) που ακολουθεί το όνομα του πίνακα στη θέση του (^{T}) δηλαδή (A').

```
>> A = [1 2 0; 3 4 2; -1 3 9]
A =
     1     2     0
     3     4     2
    -1     3     9
>> A'
ans =
     1     3    -1
     2     4     3
     0     2     9



>> B = [2 3 -1];
>> B'
ans =
     2
     3
    -1
>> C = [5; 6; 7];
>> C'
ans =
     5     6     7
```

4.2.5. Άλλοι Υπολογισμοί & Μετασχηματισμοί Πινάκων

4.2.5.1 Αντιστροφή

Η **αντιστροφή πίνακα (A^{-1})** δεν είναι απλή πράξη και δεν γίνεται για όλους τους πίνακες. Το MatLab έχει ειδική συνάρτηση αντιστροφής την **inv**() η οποία υπολογίζει τον αντίστροφο ή επιστρέφει τους λόγους της αποτυχίας να υπολογιστεί. Εάν ο υπολογισμός είναι σωστός, το γινόμενο $A*A^{-1}$ θα δίνει τον μοναδιαίο I.

```
>> A = [1 2 ; 3 4]
A =
     1     2
     3     4
>> Ainv = inv(A)
Ainv =
   -2.0000    1.0000
    1.5000   -0.5000
```

```
>> A * Ainv  ↵
ans =
    1.0000         0
    0.0000    1.0000

>> A = [100 0 0 ; 2 0 3; 1000 0 0]  ↵
A =
         100           0           0
           2           0           3
        1000           0           0
>> inv(A)  ↵
Warning: Matrix is singular to working precision.
ans =
   Inf   Inf   Inf
   Inf   Inf   Inf
   Inf   Inf   Inf
```

4.2.5.2 Ορίζουσα

Η **ορίζουσα πίνακα (|A|)**, υπολογίζεται με τη συνάρτηση **det**() του MatLab. Ορίζουσα όπως και αντίστροφο έχουν μόνο οι τετραγωνικοί πίνακες.

```
>> A = [1 2 ; 3 4]  ↵
A =
     1     2
     3     4
>> det(A)
ans =
    -2
```

Όπως θα δούμε παρακάτω στην επίλυση προβλημάτων με το MatLab, μια από τις συχνές χρήσεις των πινάκων είναι να περιγράφουν **συστήματα γραμμικών εξισώσεων**. Έστω ότι, Α είναι ο πίνακας των συντελεστών ενός συστήματος γραμμικών εξισώσεων, Χ είναι το διάνυσμα των αγνώστων, και, Β το διάνυσμα των σταθερών όρων στο δεξί σκέλος των εξισώσεων. Τότε το σύστημα των γραμμικών εξισώσεων και η λύση του με τη μορφή μητρώων θα γράφεται με ένα από τους δύο τρόπους:

α) $\mathbf{A \cdot X = B}$ => $\mathbf{X = A^{-1} \cdot B}$ =>(MatLab): **X = inv(A)·B** ή **X = A\B**

β) $\mathbf{X \cdot A = B}$ => $\mathbf{X = B \cdot A^{-1}}$ =>(MatLab): **X = B·inv(A)** ή **X = B/A**

4.2.5.3 Μετασχηματισμοί

Το MatLab διαθέτει επίσης πληθώρα από **συναρτήσεις μετασχηματισμού** των πινάκων. Τέτοιες είναι οι:

- **rot90(A)** η οποία περιστρέφει αριστερά 90^o ένα μητρώο,
- **flipud(A),** η οποία περιστρέφει κάθετα (επάνω-κάτω) ένα μητρώο,
- **fliplr(A),** η οποία περιστρέφει οριζόντια (δεξιά-αριστερά) ένα μητρώο,
- **flipdim(A,dim),** η οποία περιστρέφει ένα μητρώο ως προς τα διάσταση dim.

Π.χ.:

```
>> a = [1 2 3; 4 5 6]  ↵
a =
     1     2     3
     4     5     6
>> rot90(a)  ↵
ans =
     3     6
     2     5
     1     4
>> flipud(a)  ↵
ans =
     4     5     6
     1     2     3
>> fliplr(a)  ↵
ans =
     3     2     1
     6     5     4
```

- **reshape(A,m,n,…)** η οποία αλλάζει τις διαστάσεις ενός πίνακα σε (mxnx…).

```
>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12]  ↵
A =
     1     2     3     4
     5     6     7     8
     9    10    11    12
>> reshape(A,2,6)  ↵
ans =
     1     9     6     3    11     8
     5     2    10     7     4    12
```

- **permute(A,order),** η οποία αναδιατάσσει τη σειρά των διαστάσεων ενός πίνακα.

```
>> X = fix(10*rand(3, 4, 2))  ↵
X(:,:,1) =
     7     7     2     9
     9     4     6     7
```

```
     9     4     3     4
X(:,:,2) =
     7     9     8     2
     2     6     6     6
     4     2     1     6
>> permute(X,[3 1 2])
ans(:,:,1) =
     7     9     9
     7     2     4
ans(:,:,2) =
     7     4     4
     9     6     2
ans(:,:,3) =
     2     6     3
     8     6     1
ans(:,:,4) =
     9     7     4
     2     6     6
```

- **squeeze(A)** η οποία εξαλείφει καταχρηστικές διαστάσεις (=1) ενός πίνακα,
- **sort(A)** η οποία ταξινομεί τα στοιχεία ενός πίνακα σε αύξουσα σειρά,
- κλπ.

4.2.5.4 Χαρακτηριστικά Πινάκων

Η νόρμα ενός διανύσματος Α υπολογίζεται με τη συνάρτηση **norm**(A,p), όπου p δηλώνει το είδος της νόρμας που θέλουμε να υπολογίσουμε. Αν το p απουσιάζει τότε υπολογίζεται η Ευκλείδεια νόρμα του διανύσματος. Π.χ.:

```
>> A = [1 2 3 4];
>> norm(A)
ans =
    5.4772
>> sqrt(sum(A.^2))
ans =
    5.4772
```

Ο δείκτης κατάστασης ενός πίνακα δίνεται από τη συνάρτηση **cond**(X). Αν επιστραφεί μικρή τιμή, ο πίνακας είναι καλής κατάστασης ενώ μια μεγάλη τιμή σημαίνει ότι ο πίνακας πλησιάζει στο να μην είναι ομαλός.

```
>> cond([ 1 2 ; 3 4])
ans =
   14.9330
>> cond([ 100  2 ; 3000  4])
ans =
  1.6089e+003
```

Άλλα χαρακτηριστικά πινάκων που μπορούμε να εξάγουμε με τη βοήθεια συναρτήσεων είναι το μέγεθός του με τις **size**() & **length**() και το πλήθος των στοιχείων του με την **numel**(). Οι συναρτήσεις αυτές είναι ιδιαίτερα χρήσιμες προγράμματα που δέχονται στην είσοδο πίνακες και πρέπει να γνωρίζουν τις διαστάσεις τους ώστε να αποφεύγονται υπολογιστικά και αλγοριθμικά λάθη.