

Εισαγωγικές σημειώσεις στο Matlab

- **Τι είναι το Matlab;**

Το Matlab είναι ένα διαδραστικό πακέτο για αριθμητικούς υπολογισμούς που δημιουργήθηκε από τον Cleve Moler την δεκαετία του 70 και έχει εξελιχθεί σε επιτυχημένο εμπορικό πακέτο λογισμικού.

Κάνοντας χρήση των πολλών build in συναρτήσεων του, ο χρήστης έχει την δυνατότητα να κάνει περίπλοκους υπολογισμούς πολύ εύκολα. Επίσης έχει την δυνατότητα να δημιουργήσει τις δικές του συναρτήσεις.

Οι εφαρμογές του Matlab πραγματικά είναι πάρα πολλές και δεν έχει νόημα να αναφερθούν εδώ. Σίγουρα πάντως αποτελεί το πιο ισχυρό εργαλείο στα χέρια ενός φοιτητή που θέλει να ασχοληθεί με αριθμητική ανάλυση.

- **Το περιβάλλον του Matlab**

Το παράθυρό με το οποίο έχετε πρώτη φορά ο χρήστης όταν εκκινεί το πρόγραμμα, ονομάζεται **Matlab Desktop**. Μέσα σε αυτό υπάρχει το **Command Window** το οποίο είναι η βασική θέση αλληλεπίδρασης με το Matlab. Επιλέγοντας από την μπάρα του μενού του Matlab desktop **View** → **Desktop Layout** ο χρήστης μπορεί να επιλέξει τι θέλει να εμφανίζεται.

- **Command Window**

Το command window όπως αναφέρθηκε πιο πάνω είναι το βασικό σημείο που ο χρήστης μπορεί να αλληλεπιδράσει με το πρόγραμμα, καθώς είναι ο χώρος στον οποίο μπορεί να χρησιμοποιήσει τις εντολές και συναρτήσεις του Matlab.

- **Matlab Editor**

Για να μπορέσετε να γράψετε τα δικά σας προγράμματα και συναρτήσεις το Matlab έχει τον δικό του ενσωματωμένο editor, οποίος έχει και εργαλεία που βοηθάνε στο debugging. Για να έχουμε πρόσβαση αρκεί από την μπάρα του μενού να επιλέξετε **File** → **New** → **M-file** .

- **Ποιός είναι ο βασικός τύπος αρχείου στο Matlab;**

Τα **m-files** αποτελούν τον βασικό τύπο αρχείων που στηρίζεται όλο το Matlab. Είναι text files και περιέχουν κώδικά που υλοποιείται από την γλώσσα προγραμματισμού που υποστηρίζει το υπολογιστικό πακέτο.

Τα m-files χωρίζονται σε 2 μεγάλες κατηγορίες τα **script files** και τις **function**. Τα script files περιέχουν μια σειρά από εντολές Matlab, τις οποίες ο interpreter εκτελεί μια προς μία. Αν αλλάξουμε την τιμή μιας μεταβλητής σε αυτά , η τιμή αλλάζει και για το χώρο εργασίας μας και αντίστροφα.

Μια function είναι ένα πιο “δομημένο” m-file που όμως αλληλεπιδρά με τον χώρο εργασίας μόνο μέσω των μεταβλητών εισόδων που δέχεται και των μεταβλητών εξόδου που δημιουργεί.

Το Matlab παρέχει ένα σύνολο πρωτογενών δομών δεδομένων. Όπως floating point πίνακες και συμβολοσειρές. Αυτά είναι τα βασικά στοιχεία για να φτιάξουμε σύνθετες δομές με μέσα που μας παρέχει το πρόγραμμα.

Επειδή λοιπόν το βασικό δομικό στοιχείο του Matlab είναι ο πίνακας, οι σχεδιαστές του βελτιστοποίησαν τους υπολογισμούς με πίνακες κάνοντας το έργο των προγραμματιστών ευκολότερο.

Δήλωση Μεταβλητών

Πριν χρησιμοποιήσουμε μια μεταβλητή θα πρέπει πρώτα να την δηλώσουμε. Αυτό στο Matlab γίνεται πολύ απλά. Αρκεί να της αναθέσουμε μια τιμή. Για παράδειγμα αρκεί να πληκτρολογήσουμε στο command window το παρακάτω :

```
>> a = 1;
```

Το αποτέλεσμα αυτής της πράξης μας είναι η δημιουργία της μεταβλητής a η οποία έχει την τιμή 1. Η μεταβλητή a είναι ένας 1x1 πίνακας.

Με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε μεταβλητές που θα έχουν ως τιμές

- Κενή τιμή
>> b = []; (πίνακας 0x0)
- Συμβολοσειρά
>> c = 'yes';
- Διάνυσμα
>> d = [1 2 3];
- Διδιάστατο πίνακα
>> e = [1 0;0 1];

Όπως μπορούμε πολύ εύκολα να αντιληφθούμε για να κάνουμε ανάθεση τιμής σε μια μεταβλητή αρκεί να βάλουμε στην αριστερή πλευρά την μεταβλητή στην συνέχεια = και δεξιά την τιμή που θέλουμε.

ΠΡΟΣΟΧΗ : Όταν θέλουμε να εισάγουμε χαρακτήρα ή συμβολοσειρά σαν τιμή σε μια μεταβλητή πρέπει να κάνουμε χρήση μονών εισαγωγικών, γιατί αλλιώς ο interpreter θα νομίζει ότι πάμε να κάνουμε εκχώρηση της τιμής μιας μεταβλητής στην άλλη.

Κανόνες Ονομασίας Μεταβλητών

1. Στα ονόματα των μεταβλητών γίνεται διάκριση πεζών/κεφαλαίων.
2. Τα ονόματα των μεταβλητών μπορούν να περιέχουν μέχρι 63 χαρακτήρες. Παραπάνω χαρακτήρες αγνοούνται.
3. Τα σημεία στίξης δεν επιτρέπονται.
4. Τα ονόματα πρέπει να ξεκινάνε με γράμμα.
5. Δεν επιτρέπεται να ονομάζουμε μεταβλητές χρησιμοποιώντας δεσμευμένες λέξεις . Για να δούμε ποιες είναι αυτές αρκεί να πληκτρολογήσουμε στο

command window την εντολή **iskeyword**.

Αν δεν είμαστε σίγουροι για το αν επιτρέπεται να χρησιμοποιήσουμε κάποιο συγκεκριμένο όνομα για μεταβλητή, μπορούμε να κάνουμε τον παρακάτω έλεγχο.

Έστω *S* το όνομα που θέλουμε να χρησιμοποιήσουμε. Πληκτρολογούμε :

isvarname('S') και αν το αποτέλεσμα είναι 1 τότε μπορούμε να ονομάσουμε μια μεταβλητή με το αντίστοιχο string *S*.

Ειδικές Μεταβλητές

Το Matlab διαθέτει έναν αριθμό ειδικών μεταβλητών

1. **ans** προεπιλεγμένο όνομα μεταβλητής που χρησιμοποιείται για αποτελέσματα.
2. **pi** το γνωστό μας π
3. **eps** Το eps είναι ο μικρότερος $\epsilon > 0$ για τον οποίο $1 + \epsilon > 1$ για τον υπολογιστή.
4. **inf** είναι το άπειρο, πχ $1/0$
5. **NaN** μη αριθμός πχ $0/0$
6. **i** ή **j** είναι το $\sqrt{-1}$

Εισαγωγή Σχολίων

Για να εισάγουμε σχόλια σε κώδικα Matlab χρησιμοποιούμε το σύμβολο % , ότι και να γράψουμε αριστερά από το σύμβολο του ποσοστού θεωρείται σχόλιο και αγνοείται από τον interpreter κατά την εκτέλεση του.

I/O (input/output)

Για να δώσουμε στον χρήστη ενός προγράμματος την δυνατότητα να εισάγει δεδομένα όταν χρειάζεται από το πληκτρολόγιο, χρησιμοποιούμε την εντολή **input** .

Η σύνταξη της **input** είναι

Μεταβλητή = input('Μήνυμα προς τον χρήστη')

Όταν εκτελεστεί η συγκεκριμένη εντολή βλέπουμε στο command window το μήνυμα που έχουμε μέσα στην παρένθεση και περιμένει το πρόγραμμα να εισάγει ο χρήστης δεδομένα και να πατήσει enter.

Για την έξοδο δεδομένων έχουμε πιο πολλές επιλογές.

Αρχικά ας μιλήσουμε για την χρήση του ελληνικού ερωτηματικού (;) στο τέλος των πράξεων και των εντολών. Αν δεν βάλουμε ; στο τέλος των εντολών αυτό έχει σαν αποτέλεσμα την εκτύπωση των δεδομένων μας στην οθόνη. Ακόμα και όταν κάνουμε απλά μια ανάθεση μεταβλητών. Για αυτό καλό είναι να βάζουμε πάντα ερωτηματικό και αν θέλουμε να εμφανιστεί κάτι στην οθόνη να χρησιμοποιήσουμε κάποια από τις 2 εντολές που ακολουθούν.

disp και **fprintf**

Σύνταξη της **disp** : `disp('Μήνυμα')` ή `disp(μεταβλητή)`

Στην πρώτη περίπτωση εμφανίζεται το μήνυμα που βρίσκεται μέσα στην παρένθεση της `disp`. Στην δεύτερη εμφανίζεται η τιμή που έχει η μεταβλητή.

Σύνταξη της **fprintf** : `fprintf('φόρμα ελέγχου',έκφραση1, έκφραση2,...)`

Η φόρμα ελέγχου καθορίζει αριθμό ορισμάτων και εκτύπωσης. Περιέχει 3 τύπους αντικειμένων.

- Χαρακτήρες που σχηματίζουν μηνύματα που θέλουμε να εμφανιστούν.
- Κώδικες μετατροπής/εκτύπωσης που εκτυπώνουν κατάλληλα την επόμενη διαθέσιμη έκφραση. Εισάγονται με % (είναι η μόνη περίπτωση που ότι ακολουθεί το % δεν θεωρείται σχόλιο)
 - Συνηθέστεροι οι %f (κανονική μορφή), %e (εκθετική μορφή)
 - Γενικά με %a.%bf γίνεται η εκτύπωση σε α θέσεις με β δεκαδικά ψηφία
- Τον χαρακτήρα αλλαγής γραμμής \n

Παράδειγμα : `fprintf(' Το αποτέλεσμα της πράξης 4+1 είναι %2.1f \n',4+1)`

Τελεστές

Οι συνήθεις πράξεις που γνωρίζουμε από τα μαθηματικά ισχύουν κανονικά και στο Matlab.

- + πρόσθεση
- - αφαίρεση
- * πολλαπλασιασμός
- / διαίρεση

Στο Matlab συναντάμε και τον τελεστή άνω και κάτω τελεία (:). Είναι πολύ σημαντικός στην δημιουργία ακολουθιών και διανυσμάτων.

Αρχική τιμή : Βήμα : Τελική τιμή

Στην περίπτωση που το βήμα παραλείπεται , δηλαδή **Αρχική τιμή : Τελική τιμή** , για βήμα θεωρούμε την μονάδα.

Αν θέλουμε να υψώσουμε σε δύναμη χρησιμοποιούμε τον τελεστή ^ ,πχ $2^3 = 8$

Ενώ για την τετραγωνική ρίζα την συνάρτηση `sqrt` , πχ `sqrt(4) = 2`

Για να συγκρίνουμε δεδομένα χρησιμοποιούμε τους παρακάτω τελεστές

- μικρότερο από <
- μικρότερο ή ίσο από <=
- μεγαλύτερο από >
- μεγαλύτερο ή ίσο από >=
- ίσο με ==
- διάφορο από ~=

Λογικοί Τελεστές

Στο Matlab χρησιμοποιούνται σαν λογικοί τελεστές τα ακόλουθα σύμβολα

& Λογικό και
| Λογικό ή
~ Λογική άρνηση

Το Matlab δεν έχει τελεστή για την αποκλειστική διάζευξη αλλά χρησιμοποιούμε την συνάρτηση **xor**.

Το αληθές στο Matlab παριστάνεται με το 1 (ή οποιαδήποτε μη μηδενική τιμή), ενώ το ψευδές με 0. Μπορείτε να χρησιμοποιήσετε και το true, false αντίστοιχα αλλά πρέπει να είστε σίγουροι ότι δεν έχουν χρησιμοποιηθεί σαν μεταβλητές προηγουμένως και έχουν πάρει κάποια τιμή, γιατί τότε χάνουν την “λογική ιδιότητα” που ξέρουμε.

Επαναληπτικές Μέθοδοι

Οποιοσδήποτε έχει ασχοληθεί με μια γλώσσα προγραμματισμού γνωρίζει ότι 2 είναι οι επαναληπτικές μέθοδοι που χρησιμοποιούμε.

For όταν γνωρίζουμε τον αριθμό των επαναλήψεων που θέλουμε να πραγματοποιήσουμε και **While** όταν το πλήθος των επαναλήψεων μας είναι άγνωστο αλλά εξαρτάται από μια συνθήκη. Για να δούμε λοιπόν πως συντάσσονται αυτές οι δύο στο περιβάλλον του Matlab.

for

for μεταβλητή = διάνυσμα
 εντολές
end

Η μεταβλητή μας είναι ουσιαστικά ο μετρητής μας.

Γιατί όμως λέμε διάνυσμα; Συνήθως θα δούμε την for να συντάσσεται

```
for i = start_value:step:end_value  
    ....  
    ....  
end
```

Αυτό όμως είναι το ίδιο πράγμα με το να πάρουμε το διάνυσμα

```
i = [ start_value, start_value +step, start_value +2step,..,end_value*]
```

ΠΡΟΣΟΧΗ : Λέμε end_value* γιατί το αν θα πάρει το i την end_value σαν τιμή εξαρτάται από το βήμα (step) που έχουμε επιλέξει. Πάντα όμως το end_value* θα είναι μικρότερο ή ίσο του end_value και end_value* + step θα είναι μεγαλύτερο του end_value

για αυτό μπορούμε και να πούμε

```
for i = [4 6 3 6 2]
```

Σε αυτή την περίπτωση το i θα πάρει με την σειρά τις τιμές 4,6,3,6 και 2. Ανάλογα σε ποια φορά της επανάληψης βρισκόμαστε.

while

```
while λογική συνθήκη  
    εντολές  
end
```

Όσο η λογική συνθήκη μας παραμένει αληθής οι εντολές που βρίσκονται μέσα στο “σώμα” της while θα εκτελούνται. (Να φροντίζουμε πάντα όταν χρησιμοποιούμε την while να μην οδηγηθούμε σε ατέρμονη επανάληψη)

Παράδειγμα

```
e = 1;  
while 1+e>1  
    e=e/2;  
end  
disp(2*e)
```

Αν εκτελέσουμε τον παραπάνω κώδικα θα πάρουμε σαν αποτέλεσμα την τιμή του eps που αναφέρθηκε στις ειδικές μεταβλητές.

Μέθοδοι Ελέγχου

Ενώ υπάρχει και η switch-case εμείς θα αναφερθούμε μόνο στην **if – elseif – else**

if

```
if    λογική συνθήκη  
        εντολές  
elseif λογική συνθήκη  
        εντολές  
else  
        εντολές  
end
```

Παράδειγμα

```
n=input('Give a number');  
if n > 0  
    disp('positive')  
elseif n < 0  
    disp('negative')  
else  
    disp('zero')  
end
```

ΠΡΟΣΟΧΗ!!!

Είναι άλλο το **elseif** και άλλο το **else if**

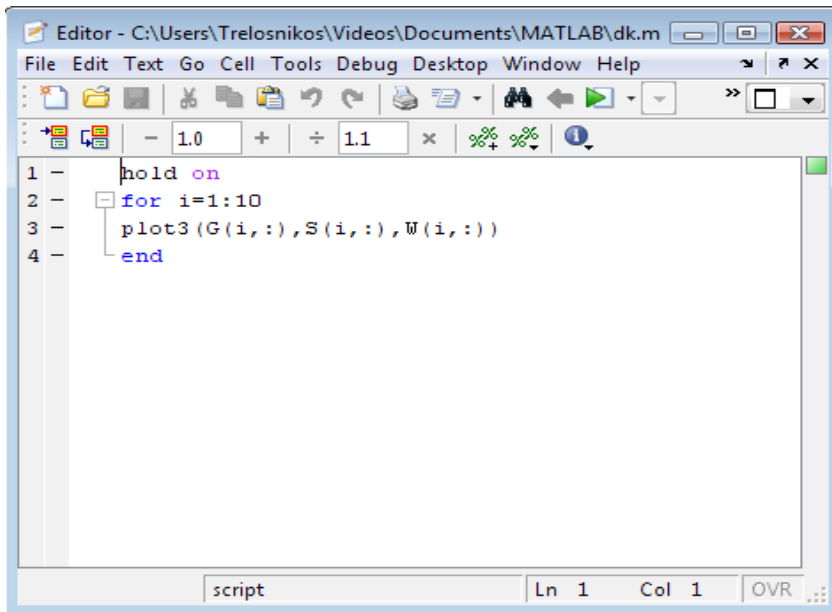
Το παράδειγμα πιο πάνω στην περίπτωση που αντί για elseif βάλουμε else if γίνεται

```
n=input('Give a number');  
if n > 0  
    disp('positive')  
else if n < 0  
    disp('negative')  
else  
    disp('zero')  
end  
end
```

Script Files

Ενώ μπορούμε να εκτελούμε τις εντολές που θέλουμε στο command window, για περιπτώσεις που έχουμε μεγάλο πλήθος εντολών ή στην περίπτωση που θέλουμε να αλλάξουμε κάποια δεδομένα και να τρέξουμε ξανά τον κώδικά μας το Matlab προσφέρει την δυνατότητα να γράψουμε το σύνολο των εντολών μας σε ένα αρχείο κειμένου και αυτό να το διαβάσει και να τις εκτελέσει με την σειρά που θα τις εκτελούσαμε και εμείς αν τις βάζαμε μια μία στο command window. Αυτά τα αρχεία είναι τα λεγόμενα script files. Και έχουν κατάληξη .m

Παράδειγμα



```
1 - hold on
2 - for i=1:10
3 -     plot3(G(i,:),S(i,:),W(i,:))
4 - end
```

Για να εκτελέσουμε ένα script file αρκεί στην γραμμή εντολών να γράψουμε το όνομά του χωρίς την κατάληξη .m και να πατήσουμε enter

ΠΡΟΣΟΧΗ :

Σε περίπτωση που υπάρχει μεταβλητή με το ίδιο όνομα με το αρχείο μας ο interpreter θα καλέσει την μεταβλητή και όχι το αρχείο. Το ίδιο ισχύει και στην περίπτωση που το όνομα που χρησιμοποιήσαμε είναι ενσωματωμένη συνάρτηση.

Απαραίτητη προϋπόθεση να βρίσκεται στο τρέχον φάκελο του Matlab

Οι μεταβλητές που χρησιμοποιούμε σε ένα script file αλληλεπιδρούν με το “χώρο εργασίας” (workspace) και είναι global μεταβλητές.

Συναρτήσεις

● Διαφορές συναρτήσεων με script files

Μια συνάρτηση (function) στο Matlab διαφέρει με ένα script file, γιατί μπορεί να έχει παραμέτρους εισόδου και εξόδου καθώς επίσης μπορεί να κάνει χρήση τοπικών μεταβλητών. Επίσης έχει μια καλύτερα καθορισμένη δομή.

● Παράμετροι Συναρτήσεων

Το Matlab αυστηρά ορίζει τον μηχανισμό μέσα από τον οποίο μπορεί να μπει ή να βγει μια πληροφορία από μια συνάρτηση. Ονομάζει την πληροφορία που μπαίνει σε μια συνάρτηση “Παράμετρο Εισόδου της Συνάρτησης” και αντίστοιχα την πληροφορία που επιστρέφει η συνάρτηση την ονομάζει “Παράμετρο Εξόδου της Συνάρτησης”.

● Τοπικές Μεταβλητές

Ως τοπικές μεταβλητές ορίζουμε τις μεταβλητές που βρίσκονται εντός της συνάρτησης. Κάθε συνάρτηση έχει το δικό της τοπικό name-space. Αυτό σημαίνει ότι οι μεταβλητές που ορίζουμε μέσα στην συνάρτηση δεν μπορούμε να τις δούμε και να τις επεξεργαστούμε εκτός αυτής. Αυτό έχει σαν αποτέλεσμα, να μπορούμε να δημιουργήσουμε μια τοπική μεταβλητή αλλά και μια μεταβλητή του workspace του Matlab, που θα έχουν και οι δύο το ίδιο όνομα.

Δομή Συνάρτησης

Μια συνάρτηση αποτελείται από 3 βασικά μέρη

1. Την γραμμή ορισμού συνάρτησης
2. Την H1 γραμμή και το τμήμα βοήθειας
3. το κύριο σώμα της συνάρτησης

1. Γραμμή ορισμού συνάρτησης

Η πρώτη γραμμή που δεν είναι σχόλιο μιας συνάρτησης καλείται γραμμή ορισμού της. Αποτελείται από

`function [out1, ... ,outm] = όνομα συνάρτησης (inp1, ... ,inpk)`

- Την δεσμευμένη λέξη `function`
- Τις παραμέτρους εξόδου
- Το όνομα της συνάρτησης
- Τις παραμέτρους εισόδου

Το όνομα με το οποίο καλούμε την συνάρτηση καθορίζεται από το όνομα του m-file. Συνηθίζεται να δίνουμε στο m-file το ίδιο όνομα με το όνομα που χρησιμοποιούμε στην γραμμή ορισμού.

2. H1 γραμμή και τμήμα βοήθειας

Η πρώτη γραμμή σχολίων μετά την γραμμή ορισμού καλείται `HELP1` ή `H1`. Και είναι η περίληψη της συνάρτησης. Όλα τα σχόλια από την `H1` γραμμή

μέχρι την πρώτη γραμμή χωρίς σχόλια είναι το τμήμα βοήθειας. Και εμφανίζονται στην περίπτωση που πληκτρολογήσουμε στο command window help όνομα συνάρτησης

3. Κύριο σώμα συνάρτησης

Το κομμάτι της συνάρτησης που περιέχει τον κώδικα με τον αλγόριθμό που υλοποιεί τη συνάρτηση μας είναι το κύριο σώμα της.

Παράδειγμα

```
function [riza,k] = NewtonRaphson(x0,tol,n,fstr)
% NEWTONRAPHSON Συνάρτηση που υλοποιεί την μέθοδο Newton - Raphson
% Δέχεται σαν είσοδο
% x0 = Αρχική τιμή
% tol = Απόκλιση
% n = Μέγιστος αριθμός προσεγγίσεων
% fsrt = Συνάρτηση της οποίας ψάχνουμε ρίζα
% Σας έξοδο μας δίνει
% riza = Ρίζα που ψάχνουμε
% k = Αριθμός επαναλήψεων που χρειάστηκε
f = inline(fstr);
df = inline(diff(sym(fstr)));
k = 0;
x = x0-(f(x0)/df(x0));
while abs(f(x)) > tol && k <=n
    k = k + 1;
    x0 = x;
    x = x0-(f(x0)/df(x0));
end
riza = x;
```

- Ένα αρχείο συνάρτησης τερματίζεται με την εκτέλεση της τελευταίας γραμμής του αρχείου ή αν συναντήσουμε την εντολή **return**.
- Μια συνάρτηση μπορεί να “καλέσει” ένα script file αλλά οι υπολογισμοί του, γίνονται στο χώρο εργασίας της συνάρτησης και όχι σε αυτό του Matlab.
- Σε μια συνάρτηση μπορούμε να έχουμε και πρόσθετες συναρτήσεις (subfunctions) οι οποίες βρίσκονται στο τέλος της κύριας συνάρτησης και ακολουθούν όλους τους κανόνες των συναρτήσεων.
- Οι δευτερεύουσες συναρτήσεις μπορούν να κληθούν είτε από την κύρια συνάρτηση είτε από άλλη δευτερεύουσα συνάρτηση του ίδιου αρχείου αλλά όχι από άλλο m-file.

Παράμετροι Εισόδου και Εξόδου

- Μια συνάρτηση μπορεί να μην έχει καθόλου παραμέτρους εισόδου και εξόδου.
- Μπορούμε να καλέσουμε μια συνάρτηση με λιγότερες παραμέτρους εισόδου και εξόδου από αυτά που καθορίζονται στην γραμμή ορισμού της συνάρτησης. Αλλά δεν μπορούμε να την καλέσουμε με περισσότερα.
- Αν έχω πάνω από μία παραμέτρους εξόδου πρέπει τις παραμέτρους αυτές να τις έχω μέσα σε αγκύλες.

Χρήσιμες Συναρτήσεις που έχει έτοιμες το Matlab

- **abs(x)** : υπολογίζει την απόλυτη τιμή του x
- **sqrt(x)** : υπολογίζει την τετραγωνική ρίζα του x
- **exp(x)** : υπολογίζει την τιμή της $f = e^x$ για το x
- **sin(x)** : υπολογίζει το ημίτονο του x
- **cos(x)** : υπολογίζει το συνημίτονο του x
- **tan(x)** : υπολογίζει την εφαπτομένη του x
- **rand** : επιστρέφει τυχαίο αριθμό στο διάστημα [0,1] ακολουθώντας την ομοιόμορφη κατανομή
- **fix(x)** : επιστρέφει το ακέραιο μέρος του x
- **floor(x)** : επιστρέφει τον κοντινότερο ακέραιο που είναι μικρότερος από x
- **round(x)** : στρογγυλοποιεί τον x
- **rem(x,y)** : υπολογίζει το υπόλοιπο της διαίρεσης x/y
- **mod(x,y)** : υπολογίζει το xmody
- **log(x)** : υπολογίζει τον φυσικό λογάριθμο του x
- **gcd(x,y)** : υπολογίζει τον μέγιστο κοινό διαιρέτη των x και y

ΠΙΝΑΚΕΣ

Όπως αναφέραμε και προηγουμένως οι πίνακες είναι το βασικό στοιχείο δεδομένων του Matlab. Τα βαθμωτά μεγέθη είναι πίνακες 1x1, δηλαδή έχουν μια γραμμή και μια στήλη. Τα διανύσματα είναι μιας διάστασης πίνακες, είτε mx1 διάνυσμα στήλη (με m γραμμές) είτε 1xn διάνυσμα γραμμή (με n στήλες). Γενικότερα το Matlab υποστηρίζει πολυδιάστατους πίνακες.

Τα στοιχεία ενός πίνακα μπορεί να είναι αριθμοί, χαρακτήρες, συμβολοσειρές αλλά ακόμα και άλλοι πίνακες ή δομές. Μόνη προϋπόθεση να είναι τα ίδια σε κάθε πίνακα. (Ειδική περίπτωση οι cell arrays αλλά δεν θα αναφερθούμε σε αυτούς).

Οι πίνακες στο Matlab έχουν την δυνατότητα να αυτοπροσδιορίζουν την διάστασή τους. Έτσι όταν αρχικοποιούμε έναν πίνακα βασιζόμενοι σε ένα σύνολο τιμών το πρόγραμμα από μόνο του καθορίζει τον απαραίτητο χώρο στην μνήμη του συστήματος. Επίσης αν εισάγουμε εκ των υστέρων ένα στοιχείο το Matlab αν χρειαστεί αλλάζει τις διαστάσεις του πίνακα έτσι ώστε να μπορεί να το συμπεριλάβει και αυτό.

Εισαγωγή Πινάκων

Μπορούμε να δημιουργήσουμε πίνακες με πολλούς τρόπους. Ας δούμε τους βασικότερους.

1) Εισαγωγή μέσω command line

Αν θέλουμε για παράδειγμα να εισάγουμε το διάνυσμα γραμμή

$v = [1 \ 2 \ 4 \ 7 \ 3]$ αρκεί να γράψουμε

```
>> v = [1 2 4 7 3]
```

Τα στοιχεία μπορούμε να τα γράψουμε είτε αφήνοντας κενό είτε βάζοντας (,)

ανάμεσα τους.

Αν θέλουμε να εισάγουμε ένα διάνυσμα στήλη αρκεί να χωρίζουμε τους αριθμούς με ελληνικό ερωτηματικό. πχ

```
>> w = [4;6;2;7;9]
```

Τους διδιάστατους πίνακες τους εισάγουμε με τον ακόλουθο τρόπο.

Έστω ότι θέλουμε να εισάγουμε τον A=

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

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

Όπως παρατηρούμε εισάγουμε ανά γραμμή τα στοιχεία και όταν θέλουμε να αλλάξουμε βάζουμε (;). Κάθε γραμμή πρέπει να έχει τον ίδιο αριθμό στοιχείων.

2) Εισαγωγή μέσω m-file

Σε ένα script file γράφουμε όπως θα γράφαμε στο command line έναν πίνακα. Και απλά καλούμε το αρχείο.

3) Με χρήση του variable editor

Με την βοήθεια του variable editor, που τον βρίσκουμε στο workspace, μπορούμε να εισάγουμε πίνακες.

4) Χρησιμοποιώντας έτοιμες συναρτήσεις γεννήτορες του Matlab

Για συγκεκριμένους πίνακες μπορούμε να χρησιμοποιήσουμε συναρτήσεις γεννήτορες πινάκων που έχει ήδη το Matlab. Θα αναφερθούμε αναλυτικά σε αυτές αργότερα.

Διαχείριση Στοιχείων Πίνακα

• Ανάκτηση στοιχείων πίνακα

Αν θέλουμε να δούμε την τιμή που έχει ένα συγκεκριμένο στοιχείο ενός διανύσματος ή ενός πίνακα αρκεί να γράψουμε το όνομα της μεταβλητής μας και μέσα σε παρένθεση την θέση του στοιχείου που μας ενδιαφέρει.

Για παράδειγμα στο διάνυσμα $v = [1 \ 2 \ 5 \ 3 \ 7 \ 2]$ αν θέλουμε να δούμε τι τιμή έχει στην θέση 4 αρκεί να γράψουμε :

```
>> v(4) και να πατήσουμε enter.
```

Για διδιάστατο πίνακα, έστω τον A που εισάγαμε πιο πάνω, αν θέλουμε να δούμε την τιμή του στοιχείου στην θέση (2,3) αρκεί να γράψουμε

```
>> A(2,3)
```

• Ανάθεση τιμής στοιχείου πίνακα σε μεταβλητή

Αρκεί να κάνουμε το παρακάτω

```
>> w = A(3,1);
```

Και η τιμή του στοιχείου του πίνακα A που βρίσκεται στην 3η γραμμή της πρώτης στήλης θα γίνει τιμή και της μεταβλητής w.

• Επιλογή υποπίνακα από δοσμένο αρχικό πίνακα

Έστω ότι έχουμε εισάγει τον πίνακα

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

Και θέλουμε να δημιουργήσουμε τον υποπίνακα A11,

που είναι ο πίνακας
$$\begin{bmatrix} 3 & 4 & 8 \\ 2 & 5 & 1 \\ 9 & 3 & 2 \end{bmatrix}$$

Αρκεί να γράψουμε : `>> A11 = A(1:3,1:3)`

Αυτό σημαίνει ότι από τον πίνακα A πάρε και κατασκεύασε έναν νέο πίνακα που θα έχει τις γραμμές από την 1 έως την 3 και το ίδιο για τις στήλες.

Αν θέλαμε να επιλέξουμε αντί για τις 3 πρώτες στήλες, τις 3 τελευταίες θα βάζαμε

`>>A11 = A(1:3,2:4)`. Επίσης αν θέλαμε να πάρουμε όλες τις γραμμές αρκούσε να γράφαμε `>> A11 = A(:,1:3)`

Σημείωση!!!

Αν θέλουμε μπορούμε να επιλέξουμε αντί για συνεχόμενο διάνυσμα για τις γραμμές και τις στήλες, (αυτό κάνουμε όταν χρησιμοποιούμε τον τελεστή :), να δώσουμε διάνυσμα που θα είχε σαν τιμές ακεραίους που δεν υπερβαίνουν τις αντίστοιχες διαστάσεις του πίνακα.

Στο παράδειγμα μας ο πίνακας A είναι 4x4. Αν θέλαμε να φτιάξουμε έναν πίνακα που θα είχε μόνο στοιχεία από την 2η και 4η γραμμή και στήλη θα μπορούσαμε να γράψουμε απλά

`>> B = A([2 4],[2 4])`

- **Επέκταση διαστάσεων πίνακα**

Μπορείτε δυναμικά να επεκτείνετε έναν πίνακα απλά προσθέτοντας στοιχεία. Για παράδειγμα .

```
>> c = [3 5 7]
```

```
c =
```

```
3     5     7
```

```
>> c = [c 9]
```

```
c =
```

```
3     5     7     9
```

- **Διαγραφή γραμμών ή στηλών**

Αν θέλουμε να αφαιρέσουμε τελείως μια γραμμή ή στήλη ενός πίνακα αρκεί να κάνουμε το παρακάτω.

`A(3,:) = []`

- **Εναλλαγή θέσης γραμμών ή στηλών**

Στο Matlab είναι πολύ εύκολο να κάνει κάποιος εναλλαγή στοιχείων πίνακα. Το μόνο που πρέπει να προσέχει είναι μην βγαίνει εκτός ορίων διαστάσεων.

Έστω ότι θέλουμε να αλλάξουμε θέση στην 1η και την 4η γραμμή του A.

```
>> A([1 4],:)=A([4 1],:)
```

- **Εύρεση ανάστροφου πίνακα**

Από τις πιο εύκολες διαδικασίες είναι στο Matlab να βρούμε τον ανάστροφο ενός πίνακα. Εκεί που σε άλλες γλώσσες προγραμματισμού θα χρειαζόντουσαν πολλές γραμμές κώδικα, στο Matlab αρκεί μόνο ένας τελεστής (')

Ο ανάστροφος του πίνακα A υπολογίζεται απλά με την παρακάτω γραμμή

```
>> A'
```

Πράξεις Πινάκων

1) Πράξεις βαθμωτού με πίνακα

Αν σε οποιοδήποτε πίνακα προσθέσετε, αφαιρέσετε, πολλαπλασιάσετε ή διαιρέσετε έναν αριθμό το Matlab θα πάει και θα εφαρμόσει την επιθυμητή πράξη σε κάθε στοιχείο του πίνακα. Για να καταλάβετε τι εννοούμε απλά για τον γνωστό πίνακα A κάντε το παρακάτω

```
>> A+5
```

2) Πράξεις στοιχείο ανά στοιχείο

Έστω οι πίνακες B και C που και οι 2 είναι mxn τότε αν κάνουμε την παρακάτω πράξη το αποτέλεσμα θα είναι ένας νέος πίνακας που το στοιχείο του στην θέση (i,j) είναι το γινόμενο των αντίστοιχων στοιχείων των B και C

```
>> B.*C
```

Αντίστοιχα το >> B./C διαιρεί κάθε στοιχείο του B με το αντίστοιχο στοιχείο του C.

3) Πράξεις μεταξύ πινάκων

Αν ισχύουν οι περιορισμοί που μας βάζουν τα μαθηματικά τότε μπορούμε να κάνουμε πράξεις μεταξύ πινάκων όπως θα γράφαμε στο Matlab και πράξεις μεταξύ αριθμών. Για παράδειγμα το >>A * B είναι ο πίνακας που προκύπτει αν πολλαπλασιάσουμε τους A και B

Χρήσιμες Συναρτήσεις Πινάκων

- **size**

```
>>[m,n] = size(A)
```

Επιστρέφει σαν m τις γραμμές και n τις στήλες του πίνακα A

- **diag**

```
>> v = diag(A)
```

Επιστρέφει σαν διάνυσμα την διαγώνιο του πίνακα A

```
>> B = diag(diag(A))
```

Κατασκευάζει έναν πίνακα με μηδενικά εκτός από την κύρια διαγώνιο που είναι ίδια με αυτή του πίνακα A

- **sort**

```
>> [ x y] = sort(A)
```

Αν ο A είναι διάνυσμα ταξινομεί τα στοιχεία, από το μικρότερο στο μεγαλύτερο και αυτό είναι το διάνυσμα x, ενώ το y διάνυσμα είναι σε πια θέση του A βρισκόταν το κάθε στοιχείο του x που βρίσκεται στην αντίστοιχη θέση.

Αν ο A είναι πίνακας ταξινομεί κάθε στήλη του από το μικρότερο που τοποθετείτε στην πρώτη γραμμή της στήλης που βρίσκεται στο μεγαλύτερο που τοποθετείτε στην τελευταία γραμμή. Και αυτός ο νέος πίνακας είναι ο x. Ο y είναι οι θέσεις που είχε κάθε στοιχείο του αρχικού πίνακα στην αντίστοιχη στήλη.

- **max , min**

```
>> [x y] = max(A)
```

Αν το A είναι διάνυσμα τότε το x είναι το μέγιστο στοιχείο και το y σε ποια

θέση του A βρίσκεται. Αν το A είναι πίνακας τότε το x είναι ένα διάνυσμα που το $x(i)$ είναι το μέγιστο στοιχείο της i -στήλης και το $y(i)$ είναι το νούμερο της γραμμής που βρίσκεται το $x(i)$.

Παρατήρηση

Με τον παρακάτω κώδικα μπορούμε να εντοπίσουμε το μέγιστο στοιχείο ενός πίνακα A καθώς και την θέση του.

```
>>[x r] = max(A);
```

```
>>[y c] = max(x);
```

Το y είναι το μέγιστο στοιχείο το c είναι η στήλη του A στην οποία βρίσκεται και $r(c)$ είναι η γραμμή.

- **zeros, ones**
>> zeros(n)
Δημιουργεί έναν nxn μηδενικό πίνακα
>> zeros(m,n)
Δημιουργεί έναν mxn μηδενικό πίνακα
Αντίστοιχα ο ones δημιουργεί πίνακες με όλα τα στοιχεία να είναι 1
- **eye**
>>eye(n)
Είναι ο nxn ταυτοτικός
- **inv**
>>inv(A)
Είναι ο αντίστροφος του A αν υπάρχει
- **det**
>>det(A)
Υπολογίζει την ορίζουσα του A
- **rand**
>>rand(m,n)
Δημιουργεί έναν mxn τυχαίο πίνακα με τιμές μεταξύ 0 και 1 ακολουθώντας την ομοιόμορφη κατανομή
- **triu, tril**
>>triu(A) , tril(A)
Αντίστοιχα οι άνω και κάτω τριγωνικοί πίνακες με βάση τα στοιχεία του A