

Εισαγωγικά

Το MATLAB μια interpreted γλώσσα προγραμματισμού, που βολεύει ιδιαίτερα για πράξεις με πίνακες και διανύσματα. Όλες οι μεταβλητές στο MATLAB είναι πίνακες, ακόμη και τα διανύσματα και οι βαθμωτές ποσότητες. Η προκαθορισμένη ακρίβεια των στοιχείων των πινάκων είναι double και γενικά οι πράξεις γίνονται σε double arithmetic.

Παράδειγμα

```
>> a = 1; % enas 1x1 pinakas pou parista mia ba8mwth metablthh
>> a
a =
    1
>> b = [ 1 2 3 ] % enas pinakas 1x3 pou parista ena dianysma
b =
    1    2    3
>> c = [ 1 2 3; 4 5 6; 7 8 9 ] % enas 3x3 pinakas pou parista enan
                                % 2d pinaka.
c =
    1    2    3
    4    5    6
    7    8    9
>> d = [1:15:50] % dianysma me prwto stoixeio to 1, bhma 15 anw
                                % fragma to 50
d =
    1    16    31    46
>> e = [1:10; 3:3:30]
e =
    1    2    3    4    5    6    7    8    9    10
    3    6    9    12   15   18   21   24   27   30
```

Χειρισμός πινάκων

Στο MATLAB μπορούμε να χειριστούμε τα στοιχεία ενός πίνακα, ως εξής:

Παράδειγμα

Μπορούμε και έχουμε πρόσβαση σε κάθε στοιχείο του πίνακα ως εξής:

```
>> b(2)
ans =
    2
>> c(1, 2) % prwth grammh deyterh sthllh
ans =
    2
```

```
>> c(:, 2) % epistrefei to dianysma ths defterhs sthlhs toy c.
ans =
     2
     5
     8
```

Σημείωση 1: Η μεταβλητή ans περιέχει πάντα την απάντηση (!) στην τελευταία εντολή που εκτελέστηκε χωρίς ανάθεση...

Σημείωση 2: -ΠΡΟΣΟΧΗ!- Το MATLAB ξεκινά την αρίθμηση των στοιχείων των πινάκων από το 1 –Σε αντίθεση με τη C που ξεκινά από το 0...

```
>> c(3, :) % epistrefei to dianysma ths triths grammhs toy c.
ans =
     7     8     9
```

Κατ'αυτόν τον τρόπο μπορούμε να κάνουμε extract έναν υπο-πίνακα:

```
>> c(1:2, 2:3) % epistrefei ton ypopinaka c[1..2, 2..3]
ans =
     2     3
     5     6
```

Κατ'αυτόν τον τρόπο μπορούμε επίσης και να αναθέσουμε έναν υπο-πίνακα:

```
>> c(1:2, 3) = [10; 11]
c =
     1     2    10
     4     5    11
     7     8     9
```

Επισυνάψεις (concatenation)

Μπορούμε να επισυνάψουμε δύο πίνακες ως εξής:

```
>> a = [1 2 3]; b = [5 6 7];
```

A. Οριζόντια επισύναψη:

```
>> c = [a b]
c =
     1     2     3     5     6     7
```

B. Κατακόρυφη επισύναψη:

```
>> c = [a ; b]
c =
     1     2     3
     5     6     7
```

Μιγαδικοί αριθμοί

Είναι πολύ εύκολο να χρησιμοποιήσουμε μιγαδικούς στο MATLAB. Για να δώσουμε το φανταστικό μέρος αρκεί να πολλαπλασιάσουμε έναν πραγματικό με τη φανταστική μονάδα:

$$i = \sqrt{-1}$$

Παραδειγμα:

```
>> z = [1 + 2*i; 3 + 4*i; 2 + 5*i];  
z =  
    1.0000 + 2.0000i  
    3.0000 + 4.0000i  
    2.0000 + 5.0000i
```

Σημείωση: Το MATLAB έχει προκαθορίσει τις καθολικές μεταβλητές *i* ΚΑΙ *j* να είναι ίσες με τη φανταστική μονάδα ως εξής:

```
i = j = sqrt(-1);
```

Προφανώς μπορούμε να χρησιμοποιήσουμε τις μεταβλητές *i*, *j* αναθέτοντας τους μια τιμές (...ή πίνακες) με αποτέλεσμα όταν θα χρειαστούμε την φανταστική μονάδα να πρέπει να την επανορίσουμε, όπως παραπάνω.

Πράξεις

A+B	Πρόσθεση πινάκων
A-B	Αφαίρεση πινάκων
A*B	Πολλ/σμος πινάκων
A/B	Επίλυση συστήματος. Ισοδύναμο με: <code>A*inv(B)</code> ; %οπου <code>inv(B)</code> ο <i>antistrofos</i> του <i>B</i> -an yparxei...
A.*B	Στοιχείο-προς-στοιχείο πολλαπλασιασμός
A./B	Στοιχείο-προς-στοιχείο διάιρεση
A^2	<code>ans = A*A</code>
A.^2	Στοιχείο-προς-στοιχείο ύψωση σε δύναμη
sqrt(A)	Πίνακας με στοιχεία τις τετραγωνικές ρίζες των στοιχείων του A
exp(A)	Πίνακας με στοιχεία τα εκθετικά των στοιχείων του A
A.'	Ο ανάστροφος του A
A'	Ο συζυγής ανάστροφος του A (για μιγαδικούς...)

Για περισσότερες πληροφορίες:

```
>>help ops  
>>help elfun
```

Διανυσματικές Πραξεις

Όπως έχουμε δει οι συναρτησεις στο MATLAB είναι vectorized δηλαδή οι πράξεις αφορούν ΚΑΘΕ στοιχείο του πίνακα, όπως πχ στο $C = \text{sqrt}(A)$ όπου το καθε στοιχείο c_{ij} του πίνακα C είναι $c_{ij} = \sqrt{a_{ij}}$, όπου a_{ij} το αντίστοιχο στοιχείο του πίνακα A. Αυτό συμφέρει υπολογιστικά καθώς το MATLAB είναι μια interpreted γλώσσα και τα loop της είναι πολύ αργά.

Ροή Ελέγχου (flow control)

1) if

IF expression	<code>if I == J</code>
statements	<code> A(I,J) = 2;</code>
ELSEIF expression	<code>elseif abs(I-J) == 1</code>
statements	<code> A(I,J) = -1;</code>
ELSE	<code>else</code>
statements	<code> A(I,J) = 0;</code>
END	<code>end</code>

2) switch

SWITCH switch_expr	<code>method = 'Bilinear';</code>
CASE case_expr,	<code>switch lower(METHOD)</code>
statement, ..., statement	<code> case {'linear','bilinear'}</code>
CASE {case_expr1, case_expr2, ... }	<code> fprintf (1, 'Method is linear\n')</code>
statement, ..., statement	<code> case 'cubic'</code>
...	<code> fprintf (1, 'Method is cubic\n')</code>
OTHERWISE,	<code> case 'nearest'</code>
statement, ..., statement	<code> fprintf (1, 'Method is nearest\n')</code>
END	<code> otherwise</code>
	<code> fprintf (1, 'Unknown method.\n')</code>
	<code> end</code>

Σημείωση: Στο MATLAB τα strings είναι απλά ένας πίνακας από χαρακτήρες.

3) while

WHILE expression	<code>E = 0*A;</code>
statements	<code> F = E + eye(size(E));</code>
END	<code> N = 1;</code>
	<code> while norm(E+F-E,1) > 0</code>
	<code> E = E + F;</code>
	<code> F = A*F/N;</code>
	<code> N = N + 1;</code>
	<code> end</code>

4) for

```
FOR variable = expr,
    statement, ..., statement
END
                                for I = 1:N,
                                    for J = 1:2:N,
                                        A(I,J) = 1/(I+J-1);
                                    end
                                end
                                end
```

Σημείωση: Η πρώτη for αυξάνει το I με (default) βήμα ίσο με 1, ενώ η δεύτερη το J με (ρητά ορισμένο) βήμα ίσο με 2. Μη ακέραια βήματα και αρχικές και τελικές τιμές μπορούν επίσης να δοθούν. Π.χ.: `a = 0.2:0.01:0.6`; Επίσης, δεν είναι ανάγκη να συμπίπτει το βήμα με την τελική τιμή. Π.χ. στο προηγούμενο παράδειγμα το `A=0.2:0.01:0.60002`; δίνει τα ίδια αποτελέσματα.

5) break

Όπως ακριβώς στη C χρησιμοποιείται για έξοδο από for ή while.

Περισσότερη βοήθεια μπορείτε να βρείτε στο help του MATLAB με την εξής διαδρομή:

Help → MATLAB → Using Matlab → M-File Programming → Flow Control.

Ή με την shell εντολή:

```
>>help lang
```

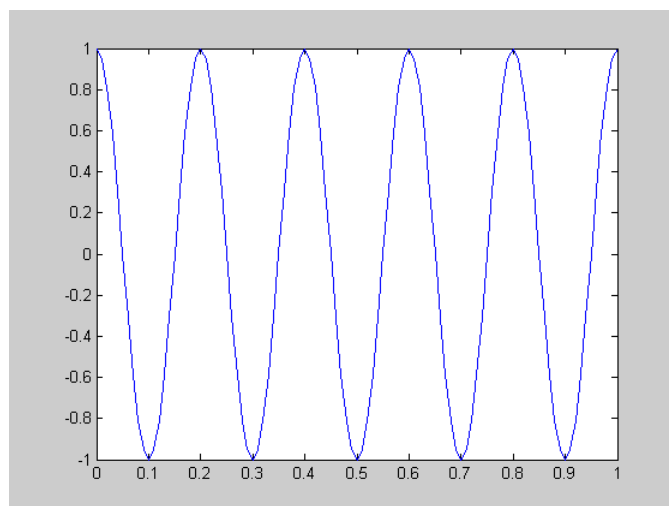
```
ή (help if) (help else) (help switch)
```

Plots

Το MATLAB υποστηρίζει την εύκολη δημιουργία διαγραμμάτων.

Παράδειγμα

```
>>t = [0:0.01:1]; % ena dianysma akolouθias xronikwn stigmwn
>>f0 = 5;          % syxnothta se Hz
>>plot ( t, cos ( 2 * pi * f0 * t ) );
```



Συναρτήσεις

Στο MATLAB μπορούμε να ορίσουμε συναρτήσεις ως εξής:

```
function [y1, y2, y3] = myfunction(x1, x2, x3)
/// blah blah-code
if something...
    return
else
    /// blah blah-code
end
/// blah blah-code
return;
```

Οι συναρτήσεις παίρνουν πολλά ορίσματα και μπορούν να επιστρέψουν περισσότερες από μια μεταβλητές. Δεν υπάρχει type-checking, αφού όλα τα ορίσματα είναι πίνακες, επομένως αν χρειάζεται να γίνει κάτι τέτοιο, πρέπει να γίνει manually μέσα στην συνάρτηση. Αν θέλουμε την συνάρτηση μας να την βλέπουμε από το περιβάλλον (γραμμή εντολών) του MATLAB πρέπει να την βάλουμε μόνη της μέσα σε ένα αρχείο με το όνομα της και κατάληξη .m. Δηλαδή η myfunction θα πρέπει να μπει στο αρχείο Myfunction.m. Αυτό το αρχείο θα πρέπει να βρίσκεται στο current directory του περιβάλλοντος του MATLAB, ή σε ένα directory στο path του MATLAB (MATLAB->File->Set Path...). Το MATLAB πρώτα κοιτά το current directory και μετά ψάχνει με την σειρά στο path του να βρεί το αντίστοιχο .m αρχείο της οποιασδήποτε συνάρτησης κληθηκε.

Παράδειγμα

```
function y = average(x)
% AVERAGE Mean of vector elements.
% AVERAGE(X), where X is a vector, is the mean of vector elements.
% Nonvector input results in an error.
[m,n] = size(x);
if ~(m == 1) | (n == 1) | (m == 1 & n == 1)
    error('Input must be a vector')
end
y = sum(x)/length(x);      % Actual computation
```

Εάν η συνάρτηση είναι μόνη της στο αρχείο της, δεν απαιτείται να μπει "return".