

# Εισαγωγή στη Matlab 2

Εισαγωγή στην Αριθμητική Ανάλυση

Διδάσκων: Γεώργιος Ακρίβης

Βοηθός: Δημήτριος Ζαβαντής

email: dzavanti@cs.uoi.gr



# Περιεχόμενα

- Ορισμοί
- Λογικοί τελεστές
- for loops
- while loops
- if – else statement
- switch, case, otherwise statement
- Μορφή της συνάρτησης



# Ορισμοί

## M-files

Η Matlab παρέχει τη δυνατότητα κλήσης συναρτήσεων και αρχείων εντολών που κατασκευάζονται από το χρήστη. Τα αρχεία αυτά (script M-files και function M-files) είναι αρχεία κειμένου που περιέχουν κώδικα Matlab.

## Script M-Files (ή command files)

Δεν έχουν ορίσματα (μεταβλητές εισόδου και εξόδου), χρησιμοποιούνται για την αυτόματη εκτέλεση εργασιών και λειτουργούν σε μεταβλητές του χώρου εργασίας της Matlab ή φτιάχνουν δικές τους μεταβλητές οι οποίες παραμένουν ενεργές στο τρέχον Workspace και μετά την εκτέλεση τους.

## Function M-Files

Περιέχουν μία γραμμή καθορισμού μίας συνάρτησης και μπορούν να δεχτούν ορίσματα (μεταβλητές εισόδου) και επιστρέφουν ορίσματα (έξοδος).



# Λογικοί τελεστές

- Statements are definitely True (=1) or False (=0)
- <, >, <=, >=, == (logical equals)
- ~= (not-equal)
- | (logical OR)
- & (logical AND)
- ~ (negation)



# 'for' loops

- Χρησιμοποιείται όταν θέλουμε να επαναλάβουμε κάποια εντολή ή μια ομάδα από εντολές όπως είναι γνωστό.

Τρόπος σύνταξης:

```
for i = m:k:n  
    statement 1;  
    statement 2;  
end
```

Οι εντολές που εσωκλείονται στον βρόγχο εκτελούνται για όλες τις τιμές του  $i$  από την τιμή  $m$  ως την τιμή  $n$  αυξανόμενο κατά  $k$  κάθε φορά. (Αν το  $k$  απουσιάζει, τότε η τιμή του  $i$  αυξάνεται κατά ένα κάθε φορά).



# Παράδειγμα στο for

```
prompt = 'Dwse timh gia to k: ';  
k = input(prompt);  
  
a = zeros(k,k); % Pre - allocate matrix  
  
for m = 1:k  
    for n = 1:k  
         $a(m,n) = \frac{1}{(m+n-1)}$ ;  
    end  
end
```

Για  $k = 3$

$a =$

1	0.5000	0.3333
0.5000	0.3333	0.2500
0.3333	0.2500	0.2000



# 'while' loops

- Χρησιμοποιείται όταν επιθυμούμε να εκτελέσουμε μια εντολή ή μια ομάδα από εντολές ένα απεριόριστο αριθμό φορές, μέχρις ότου η συνθήκη η οποία αναφέρεται αμέσως μετά την λέξη 'while' πάψει να ισχύει.

## Τρόπος σύνταξης:

*while* (λογική έκφραση)

*statement;*

*end*



# Παράδειγμα στο while

```
value = input ('Please Enter a Number between 1 and 10 (1 – 10)');
```

```
while ( value < 1 || value > 10)
```

```
    fprintf('Incorrect input, please try again. \n');
```

```
    value = input ('Enter a Number between 1 and 10 (1 – 10)');
```

```
end
```





# Παράδειγμα στο while

Result:

*Please Enter a Number between 1 and 10 (1 – 10): 18*

*Incorrect input, please try again.*

*Enter a Number between 1 and 10 (1 – 10): -4*

*Incorrect input, please try again.*

*Enter a Number between 1 and 10 (1 – 10): 2015*

*Incorrect input, please try again.*

*Enter a Number between 1 and 10 (1 – 10): -13*

*Incorrect input, please try again.*

*Enter a Number between 1 and 10 (1 – 10): 5*



# 'if' statement

- Χρησιμοποιείται όταν επιθυμούμε να εκτελέσουμε κάποια εντολή ή κάποια ομάδα εντολών μόνο αν κάποια έκφραση είναι αληθής.
- Μπορούν να γίνουν διάφοροι συνδυασμοί χρησιμοποιώντας και τις κωδικές λέξεις 'else' και 'elseif'.



# 'if' statement

```
if (expression1)  
    statement1;  
εκτελείται μόνο αν  
το expression1 είναι  
αληθές  
end
```

```
if (expression1)  
    statement1;  
εκτελείται μόνο αν  
το expression1 είναι  
αληθές  
else  
    statement2;  
εκτελείται μόνο αν  
το expression1 δεν  
είναι αληθές  
end
```

```
if (expression1)  
    statement1;  
εκτελείται μόνο αν  
το expression1 είναι  
αληθές  
elseif (expression2)  
    statement2;  
εκτελείται μόνο αν το  
expression1 δεν είναι αληθές  
και το expression2 είναι αληθές  
else  
    statement3;  
εκτελείται μόνο αν κανένα  
expression δεν είναι αληθές  
end
```



# Παράδειγμα στο if

```
for m = 1:k
  for n = 1:k

    if m == n

      a(m,n) = 2;

    elseif abs(m - n) == 2

      a(m,n) = 1;

    else a(m,n) = 0;

    end
  end
end
```

Για  $k = 5$   
 $a =$

2	0	1	0	0
0	2	0	1	0
1	0	2	0	1
0	1	0	2	0
0	0	1	0	2



# Χρήσιμες εντολές

- **break:**
  - Τερματίζει την εκτέλεση ενός βρόγχου 'for' ή 'while'.
  - Οι εντολές που ακολουθούν στον βρόγχο δεν εκτελούνται.
  - Στους βρόγχους μέσα σε βρόγχο (nested loops) η εντολή break τερματίζει μόνο τον βρόγχο μέσα στον οποίο βρίσκεται. Οι εξωτερικοί βρόγχοι δεν επηρεάζονται.
- **continue:**
  - Συνεχίζει με την επόμενη επανάληψη του βρόγχου 'for' ή 'while' και παραλείπει τις εντολές που ακολουθούν μέσα στο βρόγχο που βρίσκεται.



# Switch, case, otherwise statement

- Εκτελεί μία από τις ομάδες που δηλώνουμε

```
switch switch_expression
```

```
case case_expression
```

```
statements
```

```
case case_expression
```

```
statements
```

```
...
```

```
otherwise statements
```

```
end
```



# Παράδειγμα στο switch

```
method = 'Bilinear';
```

```
switch lower(method)
```

```
case {'linear', 'bilinear'}
```

```
disp('Method is linear')
```

```
case 'cubic'
```

```
disp('Method is cubic')
```

```
case 'nearest'
```

```
disp('Method is nearest')
```

```
otherwise
```

```
disp('Unknown method.')
```

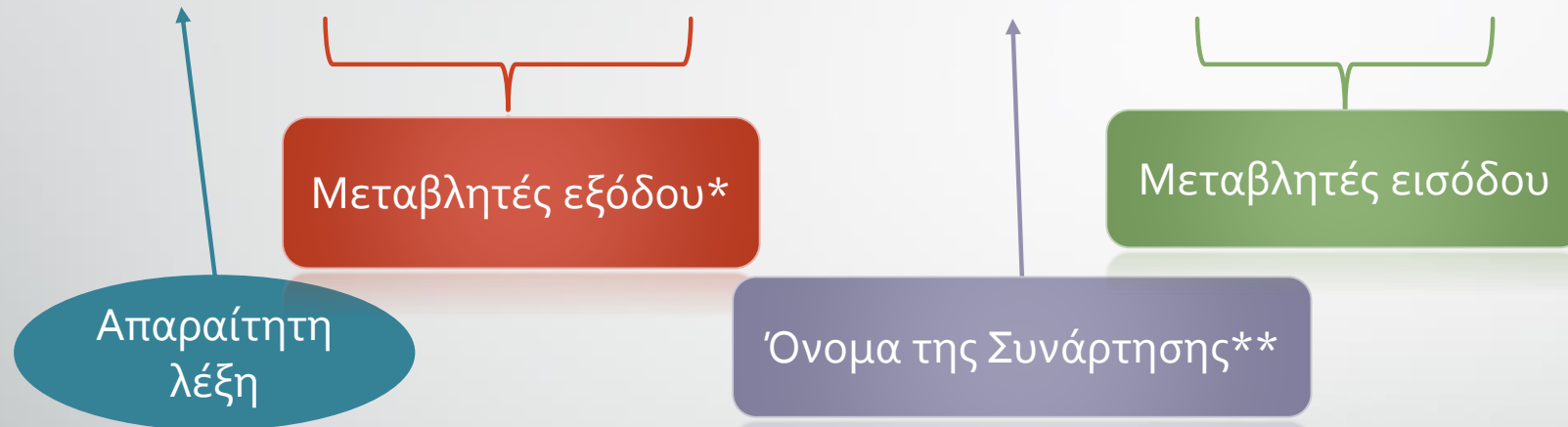
```
end
```

**Method is linear**



# Μορφή σύνταξης συνάρτησης

**function** [out1, out2, ...] = function\_name(in1, in2, ...);



\* αν δεν υπάρχουν, τότε μπορεί να μην χρησιμοποιηθούν οι αγκύλες

\*\* πρέπει απαραίτητως το αρχείο να ονομαστεί με το ίδιο ακριβώς όνομα





# Παράδειγμα συνάρτησης

```
function y = maxabsval(A)  
    y = max(max(abs(A)));  
end
```

Κλήση της συνάρτησης

```
>> maxabsval(-8:6)
```

```
ans =
```

```
8
```



# Παράδειγμα συνάρτησης

```
function y = fibonacci(a)
```

```
f = zeros(1,a);
```

```
f(1) = 0;
```

```
f(2) = 1;
```

```
for i = 3:a
```

```
    f(i) = f(i - 1) + f(i - 2);
```

```
end
```

```
end
```

```
f = fibonacci(15)
```

```
f =
```

```
0  1  1  2  3  5  8  13  21  34  55  89  144  233  377
```



Ευχαριστώ!!!  
Ευχαριστώ!!!

