

MATLAB - εισαγωγικά στοιχεία

Σωτήρης Καραβαρσάμης

sokar@aiia.csd.auth.gr

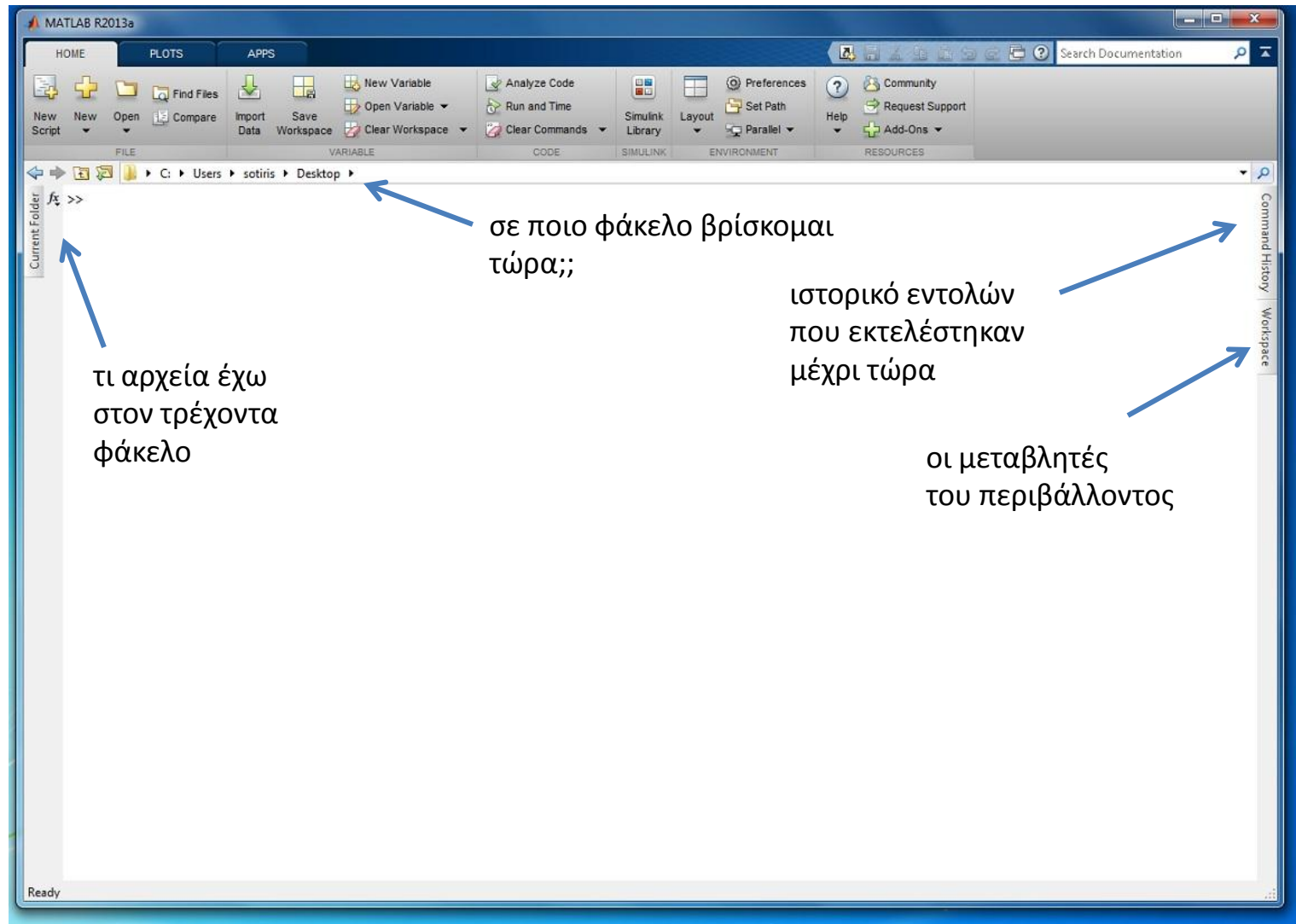
φροντιστήριο Αριθμητικής Ανάλυσης

Διδάσκων: Αναστάσιος Τέφας

Τι είναι το MATLAB;

- Είναι ένα rapid prototyping environment
 - επιτρέπει την «αποτύπωση» μιας ιδέας σε κώδικα «σπαταλώντας» το λιγότερο δυνατό χρόνο. **Πώς;**
 - διάθεση βιβλιοθηκών κάνουν την συγγραφή κώδικα περισσότερο παραγωγική! **Τι;**
 - standard library, open source κώδικας, εμπορικές βιβλιοθήκες, etc. **Nirvana!**
- Είναι μια απλοϊκή procedural γλώσσα (όπως η C) της οποίας οι στοιχειώδεις μεταβλητές είναι οι πίνακες (matrices).

Το οπτικό περιβάλλον



Μεταβλητές (1/7)

- Οι πίνακες (matrices) είναι στοιχειώδη αντικείμενα επι των οποίων κάνουμε υπολογισμούς
 - Ένας πίνακας έχει N γραμμές και M στήλες
 - Τα στοιχεία των πινάκων δεν είναι απαραίτητα αριθμητικά (π.χ. πραγματικοί αριθμοί)
 - Μπορεί, π.χ., ένας πίνακας να αποτελείται από χαρακτήρες
- Η λογική του κώδικά μας επιτελεί πράξεις/λειτουργίες επί πινάκων!

Μεταβλητές (2/7)

- Ποια η ορθή ονοματολογία μεταβλητών;
- Δήλωση μεταβλητής
 - Απλή αριθμητική τιμή, π.χ., `myvar = 10;`
 - Δήλωση ενός πίνακα 2x2, π.χ.
`myvar = [1 3; 3 1]`
 - Αντιστοιχεί στον 2x2 πίνακα που έχει στην πρώτη του γραμμή τα στοιχεία 1 και 3 και στη δεύτερη γραμμή τα στοιχεία 3 και 1
 - Αντίστοιχα γίνεται η δήλωση οποιουδήποτε πίνακα MxN (# γραμμών επί # στηλών).

Μεταβλητές (3/7)

- Χρήσιμες συναρτήσεις
 - Αρχικοποίηση NxM μηδενικού πίνακα, π.χ.,
`myvar = zeros(10, 30)`
 - Αρχικοποίηση μοναδιαίου πίνακα NxN,
`myvar = eye(3)`

Μεταβλητές (4/7)

- πράξεις πινάκων
 - Πολλαπλασιασμός πινάκων, π.χ.,
 - $C = A * B$
 - Γενικά, ο A είναι $M \times N$ και ο B είναι $N \times K$ πίνακας.
Αν αυτό δεν ισχύει, τότε λαμβάνουμε σφάλμα

```
>> A = eye(3);  
>> A
```

```
A =
```

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

```
>> B = eye(4);  
>> A*B
```

```
Error using *  
Inner matrix dimensions must agree.
```

Μεταβλητές (5/7)

- Πολλαπλασιασμός πινάκων στοιχείο προς στοιχείο, π.χ., `C = A .* B;`
 - Οι A και ο B έχουν ίσο πλήθος γραμμών/στηλών!

π.χ.,

```
A = eye(3); % create a 3x3 unitary matrix
```

```
B = randn(3,3); % create a random 3x3 matrix
```

```
C = A .* B; % multiply A and B element-wise
```


Μεταβλητές (6/7)

- Τελεστής επίλυσης γραμμικού συστήματος
 - Έστω το γραμμικό σύστημα $Ax=b$, με A $N \times N$ πίνακα και b ένα διάνυσμα $N \times 1$.
 - Με τον τελεστή “\” βρίσκω την λύση του συστήματος μέσω Gaussian elimination, π.χ.

`A = eye(3);`

`b = randn(3,1);`

`x = A \ b;`

Μεταβλητές (7/7)

- Ας δούμε ένα πιο ολοκληρωμένο παράδειγμα

```
A = zeros(100,100); % initialize zero matrix with 100 rows and 100 columns
```

```
for i=1:2:100 % iterate over the rows of A skipping one element at a time
    for j=1:2:100 % iterate over the cols of A skipping one element at a time
        if randn < 0.5 % make a random choice and place a zero or one in position (i,j)
            A(i, j) = 1;
        else
            A(i, j) = 0;
        end
    end
end
end
```

```
B = ones(100,100); % initialize 100x100 matrix whose elements are all unitary
```

```
C = B - A; % compute the “negative” of matrix A
```

Δομές επανάληψης (1/4)

- Οι δομές **for** / **while**
- Θα δούμε απλά παραδείγματα στην πράξη
- Χρειάζονται σχεδόν σε κάθε εργασία μας στο MATLAB (όσο αυτονόητες και αν φαίνονται)
- Η 1^η εργασία της Α. Α. απαιτεί χρήση αυτών των δομών
- Οι δομές επανάληψης περιγράφουν με πολύ απλό τρόπο επαναληπτικές μεθόδους

Δομές επανάληψης (2/4)

- keywords: δείκτης επανάληψης

```
% define a temporary variable
```

```
tmp = 0;
```

```
% iterate 10 times and increment the value of tmp by one
```

```
for i=1:10
```

```
    tmp = tmp + 1;
```

```
end
```

```
% print the value of tmp
```

```
fprintf(1, 'Value of tmp = %d.\n', tmp);
```

Δομές επανάληψης (3/4)

- keywords: βήμα επανάληψης

% define a temporary variable

```
tmp = 0;
```

% iterate 10 times and increment the value of tmp by one

```
for i=1:2:10
```

```
    tmp = tmp + 1;
```

```
end
```

% print the value of tmp

```
fprintf(1, 'Value of tmp = %d.\n', tmp);
```



start:step:end

Δομές επανάληψης (4/4)

- Η δομή επανάληψης **while**

```
% declare temporary variable
```

```
tmp = 0;
```

```
% declare iteration index
```

```
i = 0;
```

```
while i < 10
```

```
    % increment temporary variable
```

```
    tmp = tmp + 1;
```

```
    % print out some variables
```

```
    fprintf(1, 'value of i=%d, tmp =%d', i, tmp);
```

```
    i = i + 1; % increment iteration index
```

```
end
```

Μερικές take-home συμβουλές

- Η αρχή του παντός στο μάθημα είναι να «εγκληματιστεί» κάποιος γράφοντας κώδικα.
- Η (δι-)αίσθηση του τι κάνει μια εντολή πριν ξεκινήσετε το project σας σας υποβοηθά στο να φτάσετε σε μια λύση πιο γρήγορα.
- Μια καλή αρχή είναι να περιεργαστείτε μία-μία τις εντολές που είδαμε, εκτελώντας απλά παραδείγματα στην γραμμή εντολών του MATLAB.

Επικοινωνία

- Για απορίες
 - Email: natutor@aiia.csd.auth.gr
 - Φόρουμ μαθήματος στον ΠΗΛΕΑ
- Βοηθοί μαθήματος
 - Μαρία Βαβάμη
 - Σωτήρης Καραβαρσάμης