

Σημειώσεις στα πλαίσια του μαθήματος:  
**Βελτιστοποίηση συστημάτων υδατικών πόρων –  
Υδροπληροφορική**

## Προγραμματισμός σε Matlab για προβλήματα βελτιστοποίησης

Ευάγγελος Ρόζος και Χρήστος Μακρόπουλος  
Τομέας Υδατικών Πόρων και Περιβάλλοντος  
Εθνικό Μετσόβιο Πολυτεχνείο, Φεβρουάριος 2011

# Μαθηματικά εργαλεία

Συμβολικά:

- Mathematica
- Mathcad

Αριθμητικά:

- Matlab
- Scilab
- Octave
- R

Περιβάλλοντα εκτέλεσης αριθμητικών υπολογισμών και γλώσσες προγραμματισμού τέταρτης γενιάς.

# Εισαγωγική οθόνη Matlab

The screenshot shows the MATLAB 7.10.0 (R2010a) desktop environment. The interface is divided into several panes:

- Current Folder:** Located on the left, it shows a file explorer view of the 'MATLAB' folder, containing files like 'FormUWOTGlobals.m' and 'hindex.m'. A red annotation 'Αρχεία καταλόγου' (File catalog) is placed over this pane.
- Command Window:** The central pane, containing a message about customizable keyboard shortcuts and a prompt 'f>>'. A red annotation 'Περιοχή εντολών' (Command area) is placed over this pane.
- Workspace:** Located on the right, it shows a table with columns 'Name' and 'Value'. A red circle highlights the 'Workspace' title bar, and a red annotation 'Μεταβλητές στη μνήμη' (Variables in memory) is placed over the workspace area.
- Command History:** Located at the bottom right, it shows a list of executed commands such as 'sum(a)', 'mean(a)', 'a', 'mean(a./a)', 'help NaN', 'help all', 'help any', 'a=rand(10,1)', 'b=rand(10,1)', 'a>b', 'sum(a>b)', 'sum(a>b)>0', 'b=rand(10,1)\*10', 'sum(a>b)>0', 'sum(a<b)>0', and 'edit'. A red annotation 'Ιστορικό εντολών' (Command history) is placed over this pane.

The MATLAB window title bar reads 'MATLAB 7.10.0 (R2010a)'. The menu bar includes 'File', 'Edit', 'Debug', 'Parallel', 'Desktop', 'Window', and 'Help'. The current folder path is 'C:\Documents and Settings\yozos\My Documents\MATLAB'. The status bar at the bottom shows 'Start Ready' and 'OVR'.

# Μητρώα, μητρώα, μητρώα ...

## Εισαγωγή στοιχείων μητρώων

```
>> [1,2;4,6]
>> A=[1,2;3,4]
>> B=[3;-3]
>> C= [A,B;B',0] (επέκταση)
```

## Παραγωγή συνήθων μητρώων

```
>> eye(3)
>> zeros(3,2)
>> ones(2,3)
>> diag([1 2])
>> diag(1:3)
>> rand(3,4)
```

## Αναφορά σε στοιχεία μητρώου

```
>> C(2,3)
>> C(2,3)=2.5
>> D=randn(4)
>> D(:,3:end)
>> D(:,3:end)=-1
>> D(:,end)=[]
```

## Άσκηση

Να φτιαχτεί διαγώνιο μητρώο 9x9 με τη διαγώνιο να παίρνει τιμές από -4 έως 4. Να αντικατασταθεί το άνω δεξιά 3x3 υπομητρώο με τυχαίους αριθμούς.

# Μητρώα, μητρώα, μητρώα ...

Πρόσθεση, αφαίρεση πολλαπλασιασμός πινάκων

```
>> A=eye(4)
>> B=rand(3,4)
>> B * A
>> A * B (μα...)
>> A + 2 (παραβιάζει μαθηματική σύμβαση, πως γράφεται για να είναι OK;)
>> A + [1 2] (μα...)
>> C=rand(3)
>> C * C
>> C^2
>> C.^2
>> C^3, C^-3, C^1.1 (παραβιάζει μαθηματική σύμβαση βλ. doc arithmeticoperators)
```

“Διαίρεση”, αντίστροφος, γραμμικές εξισώσεις, tic-toc

```
>> B=rand(3)
>> A=eye(3); A(3,1)=0.3
>> tic; A^-1; toc
>> tic; pinv(A); toc (βλ. doc pinv)
>> tic; B*A^-1; toc
>> tic; B/A; toc
>> A^-1 * B
>> A\B (επιλύει την εξίσωση  $A X = B$ )
>> A/A
>> A./A
```

# Μητρώα, μητρώα, μητρώα ...

## Τελεστές μητρώων

```
>> min(A)
>> mean(A)
>> mean(mean(A))
>> mean(A(:))
>> reshape(A,2,3) (μα...)
>> reshape(A,9,1)
>> [L,I]=sort(B(:))
>> A=[1 4; 2 5; 3 6]
>> flipdim(A,1)
>> flipdim(A,2)
>> A=rand(2)
>> repmat(A,2,1)
```

## Αναζήτηση στοιχείων υπό συνθήκη

```
>> A=randn(1,7)
>> A<1 & A>-1
>> A(A<1 & A>-1)
>> find(A<1 & A>-1)
```

## Άσκηση

Να φτιαχτεί διάνυσμα με 1000 στοιχεία που ακολουθούν κανονική κατανομή. Να φτιαχτεί η εμπειρική συνάρτηση κατανομής δηλαδή η θέση σχεδίασης Weibull για τα 1000 σημεία (σελ. 116, <http://itia.ntua.gr/getfile/122/6/documents/1997StatHydrolChap5.pdf>).

# Γραφήματα

## Απλά γραφήματα

```
>> x=-pi:0.1:pi
>> plot(x,sin(x),x,cos(x),'-.')
>> plot(x,1./x,'k')
>> ylabel('x^{-1}')
```

## Άσκηση

Να σχεδιαστεί η πιθανότητα μη υπέρβασης της εμπειρική συνάρτηση κατανομής της προηγούμενης άσκησης.

## Ιστογράμματα

```
>> hist(randn(1000,1),50)
```

## Γραφήματα X-Y

```
>> scatter(cos(x),sin(x)) (πως γίνεται ευθεία χωρίς να αλλάξουν οι συναρτήσεις;)
```

## Συνοπτική παράσταση δείγματος (box plot)

```
>> boxplot(randn(1000,1))
```

## Γρήγορα γραφήματα

```
>> clear (προσοχή! σβήνει τα πάντα)
>> ezplot('sin(x)*cos(x^2)')
```

# Γραφήματα

## 3D γραφήματα

```
>> [X Y]=meshgrid(-3:3)
```

Επιστρέφει δύο μητρώα. Το πρώτο περιέχει τις τετμημένες και το δεύτερο τις τεταγμένες των κυττάρων ενός καννάβου.

```
>> [X Y]=meshgrid(-3:0.1:3);
```

```
>> Z=X.*exp(-X.^2-Y.^2);
```

```
>> mesh(X,Y,Z)
```

```
>> surf(X,Y,Z)
```

## Άσκηση

Να χρησιμοποιηθούν τα εργαλεία του παραθύρου γραφημάτων για να γίνει μεγέθυνση, μετακίνηση, περιστροφή του γραφήματος καθώς και αναγνώριση τιμών σε διάφορα σημεία του.

## Επιδείξεις (demos)

```
>> hndlaxis (βλ. help gca)
```

```
>> hndlgraf
```

```
>> graf3d
```



# Προγραμματισμός

## Συναρτήσεις - δέσμη εντολών (scripts)

Οι συναρτήσεις χρησιμοποιούν αποκλειστική περιοχή μνήμης (workspace) για να αποθηκεύουν τις μεταβλητές τους. Ανταλλάσσουν δεδομένα με άλλες συναρτήσεις ή τον χρήστη μέσω των ορισμάτων τους και των τιμών που επιστρέφουν.

Οι δέσμες εντολών χρησιμοποιούν την ίδια περιοχή μνήμης με τη γραμμή εντολών.

Χρησιμοποιούνται για την αυτοματοποίηση λειτουργιών μιας συγκεκριμένης εφαρμογής.

## Έλεγχος ροής

```
if expression
```

```
    ...
```

```
elseif expression2
```

```
    ...
```

```
else
```

```
    ...
```

```
end
```

Παράδειγμα

```
function retval=thiem(q, t, rw, roo)
```

```
if (rw==0)
```

```
    rw=0.1;
```

```
elseif (rw>roo)
```

```
    rw=roo;
```

```
end;
```

```
retval=(q/(2*pi*t)*log(roo/rw));
```

```
end
```

# Προγραμματισμός

## Σχεσιακοί τελεστές

```
>> A=[1 2]
>> B=[1 0]
>> A < B
>> A > B
>> A <= B
>> A >= B
>> A == B (προσοχή! να μην συγχέεται με το "=")
>> A ~= B
```

## Λογικοί τελεστές

A & B σύζευξη

A | B διάζευξη

xor(A,B) αποκλειστική διάζευξη

~A άρνηση

```
>> expr1=true; expr2=true; expr3=true; expr4=false;
>> (expr1 & expr2) | (expr3 & expr4)
>> expr1 & (expr2 | expr3) & expr4
>> A & B
>> A && B (μα...)
>> A | B
>> A || B (μα...)
>> ~B
```

# Προγραμματισμός

## Βρόχοι

```
for iterator = vector  
...  
end
```

## Παράδειγμα

```
...  
for i = 1:n  
    for j = 1:n  
        SFLD_ = ASS .* V(i:2*q+i, j:2*q+j);  
        SFLD(i, j) = sum(sum(SFLD_));  
    end  
end  
...
```

## Τυπική δομή συνάρτησης

```
function [out1, out2, ...] = myfun(in1, in2, ...)  
out1 = ...in1...in2...;  
out2 = ...in1...in2...;
```

# Προγραμματισμός

Παράδειγμα δημιουργίας συνάρτησης

```
>> edit stat.m
```

Εισάγετε

```
function [mean,stdev] = stat(x)
n = length(x);
mean = sum(x)/n;
stdev = sqrt(sum((x-mean).^2/n));
```

Σώστε και κλείστε ΤΟΝ ΕΠΕΞΕΡΓΑΣΤΗ

```
>> A=rand(2)
```

```
>> stat(A) (μα... έλεγχος ορισμάτων)
```

```
>> A=rand(1,10);
```

```
>> stat(A)
```

```
>> [a b]=stat(A)
```

Δεύτερο παράδειγμα

```
>> edit stat2.m
```

Εισάγετε

```
function [mean,stdev] = stat(x)
n = length(x);
mean = avg(x)/n;
stdev = sqrt(sum((x-mean).^2/n));
```

```
function mean = avg(x,n)
```

```
mean = sum(x)/n;
```

Σώστε, κλείστε ΤΟΝ ΕΠΕΞΕΡΓΑΣΤΗ και δοκιμάστε (μα...)

# Προγραμματισμός

## Πειράματα με ταχύτητα

```
>> edit slow.m
```

Εισάγετε

```
function [ret] = slow(x)
for ii=1:length(x)
    s(ii)=sqrt(x(ii));
end;
```

```
ret=sum(s);
```

Σώστε και κλείστε τον επεξεργαστή

```
>> x=-10000:0.5:10000;
```

```
>> tic; slow(x); toc;
```

```
>> edit slow.m
```

Εισάγετε στη δεύτερη γραμμή του κειμένου το ακόλουθο

```
s=zeros(1,length(x));
```

Σώστε και κλείστε τον επεξεργαστή

```
>> tic; slow(x); toc;
```

Τέλος δώστε

```
>> tic; s=sum(sqrt(x)); toc
```

## Συμβουλές για ταχύτητα

1. Εκχώρηση μνήμης άπαξ
2. Χρήση διανυσματικών πράξεων (αποφυγή for-loop)
3. Χρήση κατάλληλου τελεστή (tic-toc)

# Δομές δεδομένων

## Ορισμός

Οι δομές (structures) είναι ένας τύπος δεδομένων που επιτρέπει την αποθήκευση οργανωμένης πληροφορίας που συντίθεται από διαφορετικούς απλούς τύπους δεδομένων (αριθμοί, αλφαριθμητικά, μητρώα,...).

## Παραδείγματα δομών

```
>> strucdem_helper(1);
```

## Μητρώο δομών

```
>> patient(2).name = 'Ann Lane';
```

```
>> patient(2).billing = 28.50;
```

```
>> patient(2).test = [68 70 68; 118 118 119; 172 170 169];
```

Δοκιμάστε τώρα

```
>> strucdem_helper(2);
```

## Όλα σε μία γραμμή

```
>> simplestrct=struct('FirstName', 'Nikos', 'LastName',  
'Foufotos', 'Phone', 2107722841, 'Vector', [1 2 3])
```

Το τηλέφωνο σας παρακαλώ

```
>> fprintf('phone is %d\n', simplestrct.Phone)
```

# Αποθήκευση ανάκτηση δεδομένων

## Εντολές υψηλού επιπέδου

```
>> save('Project.mat')
>> clear
>> load Project.mat
>> who
>> save('A.txt', 'A', '-ascii')
>> clear A
>> load A.txt
>> A
```

## Εντολές χαμηλού επιπέδου

```
...
for i=1:nruns
    name=sprintf('rain%d.txt',i);
    rainfile=fopen(name, 'w');
    for j=1:simlen
        text= sprintf('%3.1f\t%3.2f\t%3.2f', Anw((i-1)*nruns+j), ...
                                                                Mesos((i-1)*nruns+j), ...
                                                                Katw((i-1)*nruns+j) );
        fprintf(rainfile, '%s\n', text);
    end
    fclose(rainfile);
end
...
end
...
```

# Μέθοδοι βελτιστοποίησης

Βελτιστοποίηση πολυμεταβλητής συνάρτησης με περιορισμούς

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

## Optimization toolbox

`x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub)`

Η `fmincon` βασίζεται στην εύρεση Εσσιανού μητρώου της Lagrangian συνάρτησης.

## Global optimization toolbox

`x = ga(problem)`

όπου

`problem=struct('fitnessfcn', <στοχική συνάρτηση>, 'nvars', <αριθμός παραμέτρων>, 'Aineq', <A>, 'Bineq', <b>, 'Aeq', <Aeq>, 'Beq', <Beq>, 'lb', <lb>, 'ub', <ub>, 'nonlcon', <συνάρτηση μη γραμμικών περιορισμών>, 'solver', 'ga', 'options', options_structure)`

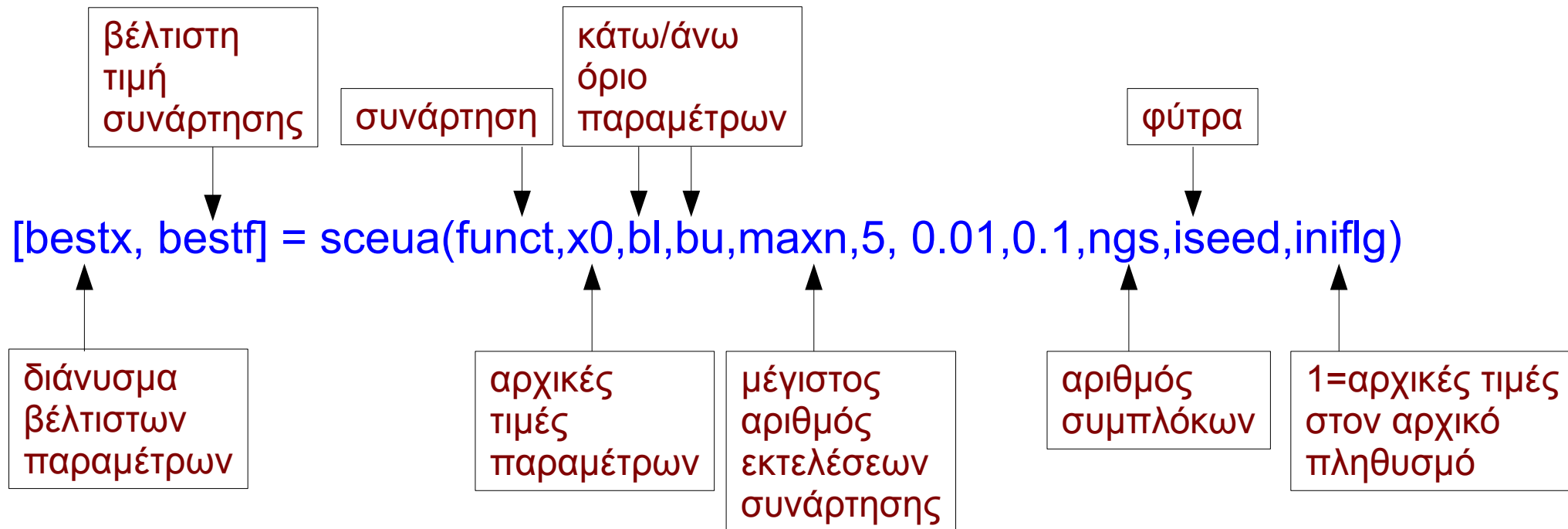
Το `options_structure` δημιουργείται με την εντολή `gaoptimset` και περιέχει τις παραμέτρους του γενετικού αλγορίθμου (βλ. `doc ga`).



Δώστε αυτή την εντολή τώρα  
`!cp /mnt/home/gkarano/MATLAB/*.m .`

# Μέθοδοι βελτιστοποίησης

## Μέθοδος SCE



## Παράδειγμα

```
>> f=@(x) sin(x);  
>> sceua(f,1,-pi,pi, 200, 5, 0.01, 0.001, 3, -1,0)
```

## Άσκηση

Βρείτε το ελάχιστο της `besselj(sqrt(x^2+y^2)+1, 5)` και επαληθεύστε το γραφικά.

# Εργασία

Γράψτε συνάρτηση που να υπολογίζει τη διόδευση πλημμυρικού κύματος από κλάδο υδρογραφικού δικτύου χρησιμοποιώντας τη μέθοδο Muskingum. Τα ορίσματα της συνάρτησης θα είναι το διάνυσμα με την παροχή στην είσοδο του κλάδου και οι δύο παράμετροι της μεθόδου. Επιλέξτε κάποιες τιμές για αυτές τις παραμέτρους (π.χ.  $x=0.24$ ,  $K=4\text{hr}$ ) και εφαρμόστε τη μέθοδο (με ιστορικές χρονοσειρές εισόδου που θα βρείτε από κάποια πραγματική εφαρμογή, π.χ. εφαρμογή στο τέλος του κεφαλαίου 6 του βιβλίου του Ξανθόπουλου (1990)) για να πάρετε την προσομοιωμένη παροχή στην έξοδο. Θεωρήστε αυτή την παροχή σαν “μετρήσεις”. Ακολουθώντας χρησιμοποιήστε εναλλακτικές μεθόδους βελτιστοποίησης (π.χ. SCE, GA, fmincon ...) και βάσει των “μετρήσεων” εκτιμήστε τις δύο παραμέτρους της μεθόδου. Συγκρίνετε τα αποτελέσματα ως προς ταχύτητα και ακρίβεια (δηλαδή διαφορά εκτιμημένων τιμών από βελτιστοποίηση ως προς αυτές που είχατε θεωρήσει για να παράξετε τις “μετρήσεις”).

# Εργασία

Να φτιαχτεί μια συνάρτηση που να προσομοιώνει έναν ορθογωνικό υδροφορέα στον οποίο επενεργούν συστοιχία γεωτρήσεων. Η συνάρτηση να έχει για ορίσματα το μητρώο με τις μεταφορικότητες των κυττάρων ( $m^2/s$ ), ένα μητρώο δομών που να περιέχει τη θέση της κάθε γεώτρησης και την παροχή της (π.χ. `wellstr.i`, `wellstr.j`, `wellstr.q`), τα  $dx$  και  $dy$  των κυττάρων, ένα μητρώο με τα υδραυλικά ύψη μόνιμης ροής χωρίς τις αντλήσεις. Η συνάρτηση επιστρέφει τα υδραυλικά ύψη που αντιστοιχούν στην αποκατάσταση μόνιμων συνθηκών ροής. Το δεξί άκρο του υδροφορέα γειτονεύει με θάλασσα. Για την αποφυγή υφαλμύρισης πρέπει το ελάχιστο υδραυλικό ύψος της πλέον δεξιά στήλης κυττάρων να είναι πάνω από 1 μέτρο. Να

Βλέπε κεφάλαιο 6 του βιβλίου των Κουτσογιάννη & Ξανθόπουλου (1999)  
<http://itia.ntua.gr/getfile/115/7/documents/1999EngHydroChap6.pdf>