Τι είναι το Matlab?

Το Matlab (MATrix LABoratory) είναι ένα interactive σύστημα για:

- αριθμητικούς υπολογισμούς, δίχως προγραμματισμό σε συμβατικές γλώσσες (Fortran, C)·
- γρήγορη ανάπτυξη και έλεγχο αλγορίθμων, (πλήθος έτοιμων συναρτήσεων και απλουστευμένη αλγοριθμική γλώσσα)·
- ανάλυση δεδομένων και γραφική παρουσίαση τους·
- εφαρμογές από διάφορες θεματικές περιοχές μέσω κατάλληλων *toolboxes* (στατιστική ανάλυση, θεωρία ελέγχου, επεξεργασία σήματος, βελτιστοποίηση, νευρωνικά δίκτυα, «συμβολικά» μαθηματικά, κ.π.α.)

Δημιουργήθηκε απο τον C. Moler, *αρχικά* σαν εργαλείο διαχείρισης των βιβλιοθηκών Fortran: LINPACK (γρ. άλγεβρα) και EISPACK (ιδιοτιμές και ιδιοδιανύσματα). *Εξελίχθηκε* σε σύνθετο πακέτο (γραμμένο σε C, C++) που αναπτύσσεται συνεχώς.

Βοήθεια στο Matlab

`help` λίστα με κατηγορίες βοήθειας.

`help` *θέμα* βοήθεια σε ένα συγκεκριμένο θέμα ή συνάρτηση.

`lookfor` *λέξη-κλειδί* ψάχνει σε όλες τις συναρτήσεις για τη *λέξη-κλειδί*

`helpdesk` «φορτώνει» στον Web browser αναλυτική τεκμηρίωση για το Matlab και τα toolboxes του.

`demo` επίδειξη δυνατοτήτων του Matlab.

Τα πάντα είναι πίνακες!

Βασικό αντικείμενο του Matlab είναι οι **πίνακες** (πραγματικοί ή μιγαδικοί).

Σε μερικές περιπτώσεις το Matlab ερμηνεύει:

- πίνακες 1×1 σαν βαθμωτά μεγέθη, και
- πίνακες με 1 γραμμή ή 1 στήλη σαν διανύσματα.

Η γλώσσα του Matlab είναι α-τυπη (δεν χρειάζεται δήλωση μεταβλητών).

Στο Matlab οι πράξεις κινητής υποδιαστολής γίνονται σύμφωνα με το standard της IEEE, συνήθως σε διπλή ακρίβεια.

Εισαγωγή πινάκων

1. Αμεσα από το χρήστη:

   ```
   A = [1 2 3; 4 5 6; 7 8 9]
   ```

   ή ισοδύναμα

   ```
   A = [ 1 2 3
         4 5 6
       7 8 9 ]
   ```

2. Από συναρτήσεις του Matlab:

   ```
   b = rand(1,5)
   ```

   δημιουργεί τυχαίο πίνακα 1×5 (διάνυσμα) με στοιχεία $\in [0, 1]$.

ΠΡΟΣΠΕΛΑΣΗ ΣΤΟΙΧΕΙΩΝ: `A(1,2)` για πίνακες, ή `b(A(2,2))` για διανύσματα. Οι δείκτες είναι θετικές ακέραιες σταθερές ή μεταβλητές.

Πράξεις πινάκων

`+`	Πρόσθεση	`help arith`
`-`	Αφαίρεση	`help arith`
`*`	Πολ/σμός	`help arith`
`^`	Υψωση σε δύναμη	`help arith`
`'`	Ανάστροφος πίνακας	`help punct`
`\`	**Αριστερή** διαίρεση	`help slash`
`/`	**Δεξιά** διαίρεση	`help slash`

Πράξεις πινάκων (συνεχ.)

Παρατηρήσεις

- Ισχύουν και για βαθμωτά μεγέθη (= πίνακες 1×1)
- Ασυμβατότητα διαστάσεων $\Longrightarrow$ *ΛΑΘΟΣ*.

  ΕΞΑΙΡΕΣΗ: πράξεις μεταξύ πινάκων και αριθμών, οπότε η πράξη εκτελείται μεταξύ του αριθμού και *κάθε* στοιχείου του πίνακα.
- Για τις διαιρέσεις: Αν ο πίνακας `A` είναι αντιστρέψιμος τότε:
  - `x = A\b` είναι η λύση του συστήματος: `A*x = b`.
  - `x = b/A` είναι η λύση του συστήματος: `x*A = b`.
- Οι πράξεις: `.*` `.^` `./` `.\` εκτελούνται μεταξύ των **στοιχείων** των πινάκων. Π.χ. `[1 2; 3 4].^2` δίνει `[1 4; 9 16]`.

Εντολές, Εκφράσεις, Μεταβλητές

- Το Matlab ερμηνεύει κάθε γραμμή στην είσοδο. Οι εντολές του έχουν τη μορφή:

  μεταβλητή=*έκφραση*

  ή απλά

  έκφραση

- Οι εκφράσεις είναι σύνθεση τελεστών, μεταβλητών και συναρτήσεων. Ο υπολογισμός τους παράγει ένα πίνακα που μπορεί να εμφανισθεί στην έξοδο ή να αποθηκευτεί σε μεταβλητή.

- Οι εντολές τερματίζονται με το τέλος της γραμμής. Συνέχεια σε περισσότερες από μία γραμμες αν η προηγούμενη γραμμή τελειώνει σε `...`

- Πολλές εντολές μεταξύ `,` ή `;` γράφονται σε μία γραμμή.

- Εντολή που τελειώνει σε `;` δεν παράγει output στην οθόνη.

Κατασκευή πινάκων

Μερικές συναρτήσεις (για σύνταξη: `help` *όνομα*)

`eye`	Μοναδιαίος πίνακας
`zeros`	Μηδενικός πίνακας
`ones`	Πίνακας με στοιχεία μονάδες
`diag`	Διαγώνιος πίνακας
`triu, tril`	Άνω, κάτω τριγωνικός πίνακας
`rand`	Πίνακας με 'τυχαία' στοιχεία
`magic`	Μαγικά τετράγωνα.

Παρατηρήσεις, παραδείγματα

- `zeros(m,n)`, $m \times n$ μηδενικός πίνακας, αλλά `zeros(n)` τετραγωνικός μηδενικός πίνακας.
- Για `x` διάνυσμα, `diag(x)` πίνακας με `x` στη διαγώνιο. Για πίνακα `A`, `diag(A)` διάνυσμα με τα διαγώνια στοιχεία του `A`.
  ΕΡΩΤΗΣΗ: Τι υπολογίζει η `diag(diag(A))`?
- Παραγωγή πινάκων από πίνακες: αν `A` είναι 3×3 τότε:

```
B = [A, zeros(3,2); ones(2,3), eye(2,2)]
```

  δίνει πίνακα 5×5.

Υποπίνακες

- Οι εκφράσεις: `1:5` και `0.2:0.2:1.2` είναι στην ουσία τα διανύσματα: `[1 2 3 4 5]` και `[0.2 0.4 0.6 0.8 1.0 1.2]`
- `A(1:4,3)` διάνυσμα με τα 4 πρώτα στοιχεία της 3ης στήλης του `A`.
- `A(:,3)` είναι ή 3η στήλη του `A`.
- `A(:,[2,4])` Οι στήλες 2 και 4 του `A`.
- `A(:,[2 4 5]) = B(:,1:3)` Αντικαθιστά τις στήλες 2, 4 και 5 του `A` με τις στήλες 1, 2, 3 του `B`.

Η εντολή `for`

```
x = []; for i = 1:n, x=[x,i^2], end
```

ή

```
x = [];
for i = 1:n
  x=[x,i^2]
end
```

Στην πιο γενική της μορφή:

```
s = 0;          % Gia 2D pinaka A:
for c = A       % Diatrexei tic sthles tou A
  s = s + sum(c)
end
```

Η εντολή `while`

`while` *λογική σχέση*

εντολές

`end`

Παράδειγμα: υπολογισμός του $\lfloor \log_2 a \rfloor$

```
a = 256;
n = 0;
while 2^n < a
   n = n + 1;
end
n
```

και με χρήση συναρτήσεων: `n = floor(log2(a))`

Η εντολή `if`

`if` *λογική σχέση*

εντολές

`end`

Παράδειγμα:

```
if n < 0
  parity = 0;
elseif rem(n,2) == 0
  parity = 2;
else
  parity = 1;
end
```

Λογικές σχέσεις και τελεστές

`help relop`

Σχέσεις: `< > <= >= == =`

Τελεστές: `&` (σύζευξη), `|` (διάζευξη), `~` (άρνηση)

ΠΑΡΑΤΗΡΗΣΕΙΣ:

- Τιμές: Αληθής $\longrightarrow$ 1, Ψευδής $\longrightarrow$ 0.
- Οι λογικές σχέσεις μεταξύ πινάκων, εκτελούνται μεταξύ των στοιχείων τους και δίνουν πίνακα με 1 ή 0 στις αντίστοιχες θέσεις. Π.χ.
  `L = [1 2; 3 4] > [1 0; 10 0]` δίνει την τιμή
  `[0 1; 0 1]` στον `L`
- Οι `while` και `if` ερμηνεύουν μια σχέση μεταξύ πινάκων σαν αληθή όταν ο παραγόμενος πίνακας έχει όλα τα στοιχεία του $= 1$. Π.χ. για το προηγούμενο `L`, η `if L, disp('MATLAB'); end` ΔΕΝ ΕΚΤΕΛΕΙΤΑΙ

Βαθμωτές Συναρτήσεις

Ενεργούν σε βαθμωτά μεγέθη (δηλ. για πίνακα: σε κάθε στοιχείο του και παράγουν πίνακα με τα αποτελέσματα, ίδιας διαστασης με τον αρχικό)

```
sin   asin   exp   abs    round
cos   acos   log   sqrt   floor
tan   atan   rem   sign   ceil
```

ΠΑΡΑΔΕΙΓΜΑ

`sin([pi pi/2; 0 pi/4])` $\longrightarrow$ `[0.0 1.0; 0.0 0.707]`

Διανυσματικές συναρτήσεις

Ενεργούν σε διανύσματα (δηλ. για πίνακα: σε κάθε στήλη του και παράγουν πίνακα γραμμή με τα αποτελέσματα κάθε στήλης).

```
max    sum    median   any
min    prod   mean     all
sort          std
```

ΠΑΡΑΔΕΙΓΜΑΤΑ

`max([1 2 3])` $\longrightarrow$ `3`

`max([1; 2; 3])` $\longrightarrow$ `3`

`max([1 2 3; 3 2 1])` $\longrightarrow$ `[3 2 3]`

Συναρτήσεις πινάκων

`eig`	Ιδιοδιανύσματα, ιδιοτιμές
`chol`	Παραγοντοποίηση Choleski
`lu`	Παραγοντοποίηση Gauss
`expm`	e^A (συγκρ. `exp`)
`sqrtm`	Τετραγωνική ρίζα πίνακα (συγκρ. `sqrt`)
`det`	Ορίζουσα
`size`	Διαστάσεις
`norm`	Νόρμες
`rank`	Βαθμός.

m-αρχεία

Έχουν κατάληξη `.m`

1. **Scripts** Αρχεία που περιέχουν αλληλουχία εντολών του Matlab. Π.χ. εντολές στο αρχείο `mycommand.m` εκτελείται με `mycommand`.
2. **Συναρτήσεις** Δυνατότητα δημιουργίας νέων συναρτήσεων. Π.χ. η νέα εντολή `randint` ορίζεται στο αρχείο `randint.m`:

```
function a = randint(m, n)
%RANDINT Randomly generated integral matrix.
% RANDINT(M,N) M-by-N matrix.
% Elements between 0 and 9
a = floor(10*rand(m,n));
```

Τι εμφανίζει η: `help randint`?

Γραφικά - οι συνάρτησεις `plot` και `semilogy`

Η συνάρτηση `plot(x,y)` σχεδιάζει 2-διάστατες γραφικές παραστάσεις του διανύσματος `y` ως προς `x`.

Παραδειγμα:

```
x = -4:.01:4;
y = sin(x);
plot(x,y);
title('MATLAB plot');
xlabel('x'); ylabel('y')
```

Για λογαριθμική κλίμακα στον άξονα των y: `semilogy`

Παράδειγμα:

```
semilogy(x,y);
```

Παραδείγματα

Για: `A=rand(3,5)`, `B` οποιοσδήποτε 3×5 πίνακας, `x=rand(5,1)`, `y` ένα διάνυσμα 5×1 και `C=[1 2 3; 4 5 6; 7 8 9]`:

1. Πραξεις:

   ```
   A+B, A+2, C^2, C.^2, C.^C
   ```

2. Δημιουργία πινάκων:

   ```
   diag(C), diag(y), diag(diag(C)), triu(C)
   eye(3)-2, -eye(3)
   ```

3. Σύνθετοι πίνακες:

   ```
   D = [A; zeros(2,3) ones(2)]
   ```

4. For:

   ```
   s = 0; for i = 1:5, s = s + x(i); end; s, sum(x)
   ```

5. Λογικές σχέσεις:

```
D = triu(C), C == D
```

6. Συναρτήσεις:

```
sin(B), max(y), rank(B), det(C), size(x)
help eig
eig(C)
[V,L]=eig(C)
```

7. Υποπίνακες:

```
B(1:2,3), B(:,2), B(2,:), B(:, [1 5])
M = B, M(:,1:3)=eye(3)
```

8. Γραφικά:

```
x=-4:.01:4; y = sin(x); plot(x,y);
title('MATLAB plot'); xlabel('x'); ylabel('sin')
```

Χρήσιμες συναρτήσεις

`rand(m,n)` Τυχαίος πίνακας $\mathtt{m}\times\mathtt{n}$.

`rand(n)` Τυχαίος πίνακας $\mathtt{n}\times\mathtt{n}$.

`ones(m,n)` Πίνακας $\mathtt{m}\times\mathtt{n}$ με στοιχεία 1.

`zeros(m,n)` Πίνακας $\mathtt{m}\times\mathtt{n}$ με στοιχεία 0.

`diag(v)` Διαγώνιος πίνακας με το διάνυσμα $\mathtt{v}$ στην κύρια διαγώνιο.

`diag(v,k)` Πίνακας με το διάνυσμα $\mathtt{v}$ στην $\mathtt{k}$ διαγώνιο ($\mathtt{k}=0$ κύρια διαγώνιος, $\mathtt{k}>0/\mathtt{k}<0$ πάνω/κάτω από κύρια διαγώνιο.

Παράδειγμα: δημιουργία τριδιαγώνιου πίνακα

Η εντολή:

```
10*eye(3) + diag(ones(2,1),1) + diag(ones(2,1),-1)
```

δίνει τον τριδιαγώνιο πίνακα:

$$\begin{pmatrix} 10 & 1 & 0 \\ 1 & 10 & 1 \\ 0 & 1 & 10 \end{pmatrix} = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Κόστος Αλγορίθμων

Για οποιαδήποτε συνάρτηση `fun(...)`:

1. Αριθμός πράξεων flops (δεν υποστηρίζεται σε νεώτερες εκδόσεις του Matlab)

   ```
   flops(0); fun(...); flops
   ```

   Προσθαφαιρέσεις: 1 flop για πραγματικούς, 2 flops για μιγαδικούς.

   Πολ/σμοί, διαιρέσεις: 1 flop για πραγματικούς, 6 flops για μιγαδικούς.

   Βασικές συναρτήσεις: 1 flop για πραγματικούς, περισσότερα για μιγαδικούς.

2. Χρονική διάρκεια

   ```
   tic; fun(...); toc
   ```

Παράδειγμα: Linpack benchmark

Μέτρηση υπολογιστικής ταχύτητας σε MFlopS (= MegaFlop/sec) με βάση το χρόνο που απαιτείται για την επίλυση με απαλοιφή Gauss γραμμικού συστήματος 100×100:

```
n = 100;
A = rand(n);
b = rand(n,1);
flops(0);
tic;
x = A\b;
t = toc;
megaflops = flops/t/1.e6
```

Αποτελεσματικότητα συναρτήσεων

Πίνακας A, $n \times n$ με στοιχεία: $a_{ij} = 1/(i+j)$ς

1ος τρόπος, χρόνος εκτέλεσης για $n = 500$ σε Pentium 1 **146.53s**

```
function A = slower(n)
%SLOWER(N) pinakas A, NxN me A(I,J) = 1/(I+J)
for i=1:n
   for j=1:n
      A(i,j) = 1 / (i+j);
   end
end
```

Αποτελεσματικότητα συναρτήσεων (συνεχ.)

Πίνακας A, $n \times n$ με στοιχεία: $a_{ij} = 1/(i+j)$

2ος τρόπος, χρόνος εκτέλεσης για $n = 500$ σε Pentium 1 | 120.01s |

```
function A = slow(n)
%SLOW(N) pinakas A, NxN me A(I,J) = 1/(I+J)
A = zeros(n);
for i=1:n
   for j=1:n
      A(i,j) = 1 / (i+j);
   end
end
```

Αποτελεσματικότητα συναρτήσεων (συνεχ.)

Πίνακας A, $n \times n$ με στοιχεία: $a_{ij} = 1/(i+j)$

3ος τρόπος, χρόνος εκτέλεσης για $n = 500$ σε Pentium 1 | 1.38s |

```
function A = fast(n)
%FAST(N) pinakas A, NxN me A(I,J) = 1/(I+J)
Y = zeros(n);
for i=1:n
   Y(i,:) = 1:n;
end
A = 1./(Y + Y');
```

Αποτελεσματικότητα συναρτήσεων (συνεχ.)

Πίνακας A, $n \times n$ με στοιχεία: $a_{ij} = 1/(i+j)$

4ος τρόπος, χρόνος εκτέλεσης για $n = 500$ σε Pentium 1 **0.94s**

```
function A = faster(n)
%FASTER(N) pinakas A, NxN me A(I,J) = 1/(I+J)
A = zeros(n);
tmp = 1:n;
for i=1:n
   A(i,:) = 1./(tmp + i);
end
```

Αποτελεσματικότητα συναρτήσεων (συνεχ.)

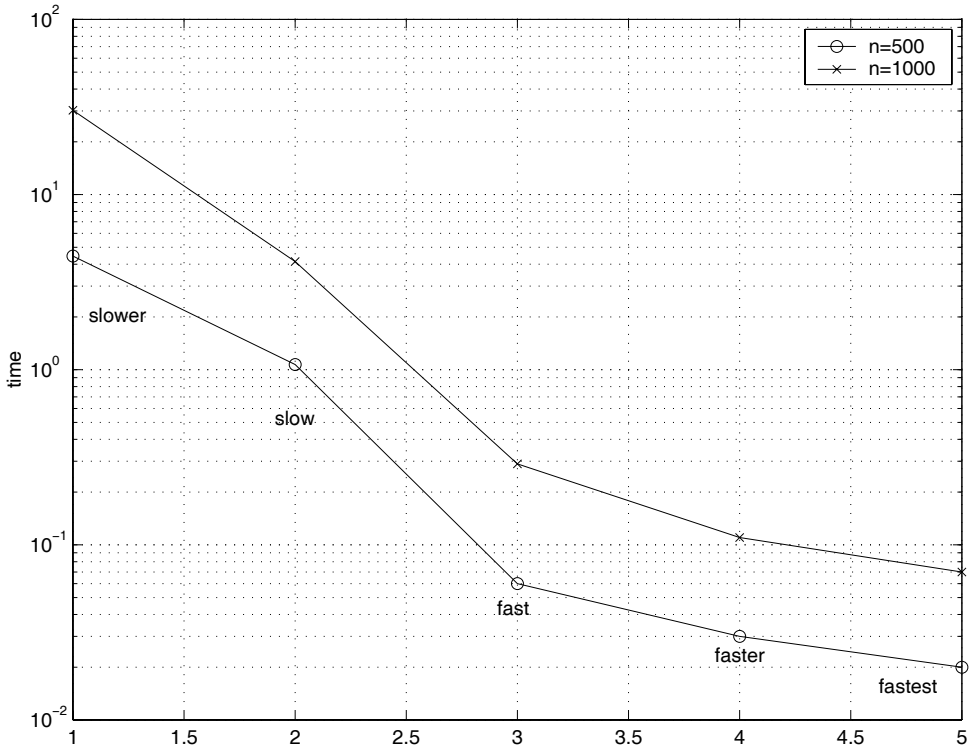
Πίνακας A, $n \times n$ με στοιχεία: $a_{ij} = 1/(i+j)$

5ος τρόπος, χρόνος εκτέλεσης για $n = 500$ σε Pentium 1 **0.69s**

```
function A = fastest(n)
%FASTEST(N) pinakas A, NxN me A(I,J) = 1/(I+J)
A = zeros(n);
tmp = (1:n)';
for i=1:n
   A(:,i) = 1./(tmp + i);
end
}
```

Χρόνοι εκτέλεσης σε Pentium 4

n	slower	slow	fast	faster	fastest
500	4.45	1.07	0.06	0.03	0.02
1000	30.26	4.14	0.29	0.11	0.07



Οδηγίες για γρηγορότερα προγράμματα Matlab

- Δέσμευσε εκ των προτέρων χώρο για πίνακες (π.χ. με την `zeros(m,n)`)
- Χρησιμοποίησε πράξεις σε ολόκληρα διανύσματα ή πίνακες αντί για βρόγχους επανάληψης με βαθμωτά μεγέθη.
- Χρησιμοποίησε *εσωτερικές* συναρτήσεις του Matlab όπου είναι δυνατόν.

Επίλυση γραμμικών συστημάτων

Αν `A` πίνακας `n`×`n` και `b`, `x` διανύσματα στήλες, τότε:

```
x = A\b
```

υπολογίζει τη λύση του συστήματος $Ax = b$ με κάποια μορφή απαλοιφής Gauss.

Αλγόριθμοι

1. `A` συμμετρικός: παραγοντοποίηση Choleski.
2. `A` μη συμμετρικός: απαλοιφή Gauss με μερική οδήγηση.

Στη γενικευμένη μορφή του ο τελεστής `\` επιλύει συστήματα πινάκων $AX = B$ με `A` διαστάσεων `m`×`n`. (Οταν `m`≠`n` γίνεται επίλυση ελαχίστων τετραγώνων).

Αντιστροφή πίνακα

Αντίστροφος του *τετραγωνικού* πίνακα `X`:

```
Y = inv(X)
```

- Στην πράξη η αντιστροφή χρησιμοποιείται *πολύ σπάνια*.
- Η συνηθέστερη κατάχρηση της για επίλυση γραμμικών συστημάτων. Η απαλοιφή Gauss (τελεστής `\`) υπερέχει σε *ακρίβεια* και *υπολογιστικό χρόνο*.

Νόρμες και δείκτης κατάστασης

`norm(X), norm(X,2)` υπολογίζει: $\|\cdot\|_2$

`norm(X,1)` υπολογίζει: $\|\cdot\|_1$

`norm(X,inf), norm(X,Inf)` υπολογίζει: $\|\cdot\|_\infty$

Για το δείκτη κατάστασης του πίνακα `X` στην *Ευκλείδεια* νόρμα:

```
kappa = cond(X)
```

Πίνακες Hilbert

Πίνακας Hilbert τάξης `n`:

```
H = hilb(n)
```

Κλασσικό παράδειγμα πινάκων με *υψηλό* δείκτη κατάστασης.

Ο αντίστροφος πίνακα Hilbert υπολογίζεται με:

```
H = invhilb(n)
```

ακριβώς (χωρίς σφάλματα περικοπής) για $n \leq 13$, και *προσεγγιστικά* για μεγαλύτερα `n`.

Η `invhilb(n)` σε σχέση με `inv(hilb(n))` αντιμετωπίζει καλύτερα τα σφάλματα περικοπής:

- στην παράσταση των `hilb(n)` και `invhilb(n)`
- στη διαδικασία αντιστροφής.