

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΟΜΕΑΣ ΘΕΩΡΗΤΙΚΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εισαγωγή στο MATLAB

Σημειώσεις

Φίλιππος Τζαφέρης

Επικ. Καθηγητής

ΑΘΗΝΑ, 2009

Περιεχόμενα

1	Εισαγωγή στο MATLAB	3
1.1	Βασικά στοιχεία του MATLAB	3
1.1.1	Περιβάλλον του MATLAB	3
1.2	Παράθυρο Εντολών (Command Window)	4
1.2.1	Αυτόματη δημιουργία ενός μονοδιάστατου πίνακα(διανύσματος στήλη)	4
1.2.2	Εισαγωγή ενός m × n πίνακα	4
1.2.3	Εντολές για Διάβασμα από αρχείο και Γράψιμο σε αρχείο	4
1.2.4	Άλλες εντολές εισόδου και εξόδου σε αρχεία	5
1.2.5	Εντολές help και demo	5
1.2.6	Οι πιο συχνά χρησιμοποιούμενες Πράξεις και Συναρτήσεις	6
1.2.7	Συναρτήσεις	8
1.2.8	Αριθμητικά Παραδείγματα	8
1.3	m -αρχεία(files)	11
1.3.1	Δημιουργία m -αρχείου	11
1.3.2	Εκτέλεση και επεξεργασία m -αρχείων	11
1.3.3	Βασικές Εντολές	12
1.4	Απλά Προγράμματα σε MATLAB	12

Κεφάλαιο 1

Εισαγωγή στο MATLAB

1.1 Βασικά στοιχεία του MATLAB

Η ονομασία MATLAB προέρχεται από τα αρχικά των λέξεων **MA**Trix **LAB**oratory (Εργαστήριο Πινάκων). Το MATLAB είναι ένα διαλογικό υπολογιστικό σύστημα σχεδιασμένο έτσι ώστε να διευκολύνει τους υπολογισμούς πινάκων που απαιτούνται για την επίλυση βασικών επιστημονικών και τεχνολογικών προβλημάτων.

Το MATLAB περιέχει προγράμματα για όλους τους βασικούς υπολογισμούς πινάκων, όπως είναι:

1. Η επίλυση γραμμικών συστημάτων
2. Σχετικά προβλήματα με την επίλυση ενός συστήματος
 - (i) η αντιστροφή ενός πίνακα
 - (ii) ο υπολογισμός της ορίζουσας ενός πίνακα
 - (iii) ο υπολογισμός της τάξης (rank) ενός πίνακα
3. Προβλήματα Ελαχίστων Τετραγώνων
4. Υπολογισμός Ιδιοτιμών και Ιδιοδιανυσμάτων ενός πίνακα

Το MATLAB είναι ένα εξαιρετικά χρήσιμο και πολύτιμο πακέτο λογισμικού για τον πειραματικό έλεγχο και την διδακτική εφαρμογή των αλγορίθμων της Αριθμητικής Γραμμικής Αλγεβρας και όχι μόνο. Ένα ακόμη αξιοσημείωτο χαρακτηριστικό της είναι οι δυνατότητες των γραφικών της στον δυδιάστατο και τρισδιάστατο χώρο.

1.1.1 Περιβάλλον του MATLAB

Το περιβάλλον MATLAB μπορεί να χρησιμοποιηθεί με δύο διαφορετικούς τρόπους:

1. Άμεσα στο παράθυρο εντολών **Command Window**(με αναγνωριστικό σύμβολο `>>`) με πληκτρολόγηση μίας ή περισσότερων εντολών του. Η κάθε μία εντολή εκτελείται χωριστά πατώντας το πλήκτρο **Enter** ή **Return**.
2. Έμμεσα όπως συμβαίνει σε μια οποιαδήποτε γλώσσα Προγραμματισμού(π.χ. στη C, C++) με τη δημιουργία και χρήση ενός ή περισσότερων αρχείων **m**-αρχείων που δηλώνονται με το όνομα τους και υποχρεωτικά με τη κατάληξη **.m**.

1.2 Παράθυρο Εντολών (Command Window)

1.2.1 Αυτόματη δημιουργία ενός μονοδιάστατου πίνακα(διανύσματος στήλη)

Η χρήση της εντολής της μορφής $\mathbf{A}=[\text{αρχή} : \text{βήμα} : \text{τέλος}]$ έχει σαν αποτέλεσμα τη δημιουργία ενός μονοδιάστατου πίνακα \mathbf{A} .

Παράδειγμα

```
>> A=[0:10:50]
```

Αποτέλεσμα

```
A =    0   10   20   30   40   50
```

1.2.2 Εισαγωγή ενός $m \times n$ πίνακα

Η χρήση της εντολής της μορφής

$$\mathbf{A} = \begin{bmatrix} a(1,1) & a(1,2) & \cdots & a(1,n); \\ a(2,1) & a(2,2) & \cdots & a(2,n); \\ \vdots & \vdots & \cdots & \vdots; \\ a(m,1) & a(m,2) & \cdots & a(m,n) \end{bmatrix}$$

έχει σαν αποτέλεσμα την εισαγωγή και καταχώρηση ενός $m \times n$ πίνακα \mathbf{A} .

Παράδειγμα

```
>> A=[1 2 3; -2 0 8]
```

Αποτέλεσμα

```
A =
     1     2     3
    -2     0     8
```

1.2.3 Εντολές για Διάβασμα από αρχείο και Γράψιμο σε αρχείο

load : Διάβασμα πινάκων

Με την εντολή **load** διαβάζονται δεδομένα(στοιχεία) από ένα αρχείο με την προϋπόθεση ότι τα αυτά είναι είναι διαταγμένα σε μορφή πίνακα. Τα στοιχεία από το αρχείο αποθηκεύονται σε μια μεταβλητή της οποίας το όνομα είναι το ίδιο με το όνομα του αρχείου(χωρίς την προέκταση).

Παράδειγμα

```
>> pinakas.dat
```

```
>> A = pinakas
```

Αποτέλεσμα

```
A =
```

5 -1 2
2 6 3

imread : Διάβασμα εικόνων

Με την εντολή **imread** διαβάζονται εικόνες.

fopen : Άνοιγμα αρχείων

Με την εντολή **fopen** ανοίγεται ένα αρχείο με προκαθορισμένο σκοπό (ή δικαίωμα) χρήσης.

Παράδειγμα

1. >> fid = **fopen**('pinakas.dat', 'r')
2. >> fid1 = **fopen**('apotel1.dat', 'w')
3. >> fid2 = **fopen**('apotel2.dat', 'a')

Αποτέλεσμα

Με την 1η εντολή ανοίγεται το αρχείο pinakas μόνο για διάβασμα. Αν το άνοιγμα του αρχείου είναι επιτυχές τότε το fid είναι ένας ακέραιος που αντιστοιχεί στο συγκεκριμένο αρχείο και με τη βοήθεια του οποίου αναφερόμαστε στο αρχείο.

Με την 2η εντολή ανοίγεται το αρχείο pinakas για γράψιμο σε αυτό. Αν το αρχείο δεν υπάρχει τότε δημιουργείται αυτόματα.

Με την 3η εντολή προστίθενται τα νέα δεδομένα στο τέλος του αρχείου apotel2.

1.2.4 Άλλες εντολές εισόδου και εξόδου σε αρχεία

fread : Διάβασμα από αρχείο

sprintf : Γράψιμο κειμένου σε αρχείο

fprintf : Γράψιμο μορφοποιημένων δεδομένων σε αρχείο

print : Γράψιμο γραφήματος σε αρχείο

fread : Αποθήκευση τιμών μεταβλητών στο δίσκο

diary : Φύλαξη των εντολών μιας εργασίας στο MATLAB

1.2.5 Εντολές help και demo

Με την εντολή **help** εμφανίζεται ένας κατάλογος που περιέχει όλες τις συναρτήσεις καθώς επίσης και τις άλλες δυνατότητες που διαθέτει το MATLAB. Ειδικότερα αν δώσουμε την εντολή **help** ακολουθούμενη από το όνομα μιας συνάρτησης του καταλόγου της MATLAB θα πάρουμε πιο σαφείς πληροφορίες για τη συγκεκριμένη συνάρτηση που επιλέξαμε.

Με την εντολή **demo** μαθαίνεις πως να χρησιμοποιείς τις συναρτήσεις του MATLAB καθώς επίσης πως να εισάγεις τιμές σε ένα πίνακα, πως να βρίσκεις τον ανάστροφό του, την τάξη του και άλλα.

Παράδειγμα 1

>> **help norm**

NORM Matrix or vector norm

For matrices..

NORM(X) is the largest singular value of **X**, **max(svd(X))**.

NORM(X,2) is the same as **NORM(X)**.

NORM(X,1) is the 1-norm of **X**, the largest column sum, = **max(sum(abs((X))))**.

NORM(X,inf) is the infinity of **X**, the largest row sum, = **max(sum(abs((X'))))**.

NORM(X,'inf') is same as **NORM(X,inf)**.

NORM(X,'fro') is the F-norm, = **sqrt(sum(diag(X' * X)))**.

NORM(X,P) is available for matrix **X** only if **P** is 1, 2, inf or '*fro*'.

For vectors..

NORM(V,P) = sum(abs(V) \hat{P}) \hat{P} (1/P) .

NORM(V) = norm(V,2).

NORM(V, inf) = max(abs(V)).

NORM(V, -inf) = min(abs(V)).

Παράδειγμα 2

>> **help lu**

LU Factors from Gaussian Elimination.

[L,U] = LU(X) stores an upper triangular matrix in U and a "psychologically lower triangular matrix", i.e., a product of lower triangular and permutation matrices, in L, so that **X = L*U**.

[L,U,P] = LU(X) returns lower triangular matrix L, upper triangular matrix U, and permutation matrix P so that **P*X=L*U**.

1.2.6 Οι πιο συχνά χρησιμοποιούμενες Πράξεις και Συναρτήσεις Βασικές Πράξεις Πινάκων

Σύμβολο	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
*	Πολ/σμός διανύσματος
^	Δύναμη
\	Αριστερή Διαίρεση ($A \setminus B = A^{-1} * B$)
/	Δεξιά Διαίρεση ($A / B = A * B^{-1}$)
sqrt	Τετραγωνική ρίζα
abs	Απόλυτη τιμή
&	Λογικό ΚΑΙ
	Λογικό Η
~	Λογικό ΟΧΙ

Συναρτήσεις Πίνακα : Αριθμητική Γραμμική Άλγεβρα

Ανάλυση Πίνακα

Συνάρτηση	Αποτέλεσμα
cond	αριθμός συνθήκης πίνακα
norm	norm πίνακα ή διανύσματος
norm(A,1)	1-norm του πίνακα A
norm(A,2)	2-norm του πίνακα A
size(A)	η διάσταση του πίνακα A
size(A,1)	το πλήθος των γραμμών του πίνακα A
size(A,2)	το πλήθος των στηλών του πίνακα A
diag	το διαγώνιο τμήμα πίνακα
triu	το άνω τριγωνικό τμήμα πίνακα
tril	το κάτω τριγωνικό τμήμα πίνακα
rank	το πλήθος των γραμμικώς ανεξαρτήτων γραμμών ή στηλών (τάξη πίνακα)
det	Ορίζουσα
trace	άθροισμα των διαγωνίων στοιχείων (ή ίχνος πίνακα)
null	μηδενικός χώρος
orth	ορθογωνιοποίηση
rref	αναγωγή σε κλιμακωτό κατά γραμμές
zeros(n)	δημιουργία ενός $n \times n$ μηδενικού πίνακα
eyes(n)	δημιουργία ενός $n \times n$ μοναδιαίου πίνακα
ones(n)	δημιουργία ενός $n \times n$ πίνακα με όλα τα στοιχεία = 1
rand(n)	δημιουργία ενός τυχαίου $n \times n$ πίνακα

Γραμμικές Εξισώσεις

Συνάρτηση	Αποτέλεσμα
\ and /	Λύση γραμμικής εξίσωσης (βλέπε 'help slash')
chol	Παραγοντοποίηση(ή διάσπαση) Cholesky
lu	Παράγοντες από την απαλοιφή του Gauss
inv	Αντίστροφος πίνακα
qr	Ορθογώνια-τριγωνική διάσπαση
qrdelete	Διαγραφή μιας στήλης από την QR παραγοντοποίηση
qrinsert	Εισαγωγή μιας στήλης στην QR παραγοντοποίηση

Ιδιοτιμές και Ιδιάζουσες τιμές

Συνάρτηση	Αποτέλεσμα
eig	Ιδιοτιμές και Ιδιοδιανύσματα ($\mathbf{E} = \mathbf{eig}(\mathbf{A})$ ή $[\mathbf{V}, \mathbf{D}] = \mathbf{eig}(\mathbf{A})$)
poly	Χαρακτηριστικό Πολυώνυμο
hess	Μορφή Hessenberg ($[\mathbf{P}, \mathbf{H}] = \mathbf{hess}(\mathbf{A}) \Leftrightarrow \mathbf{A} = \mathbf{P} * \mathbf{H} * \mathbf{P}^T$ και $\mathbf{P}^T * \mathbf{P} = \mathbf{I}$)
svd	Διάσπαση σε ιδιάζουσες τιμές ($[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathbf{svd}(\mathbf{X}) \Leftrightarrow \mathbf{X} = \mathbf{U} * \mathbf{S} * \mathbf{V}^T$, \mathbf{S} διαγώνιος, \mathbf{U}, \mathbf{V} μοναδιαίοι(unitary), $\mathbf{U} * \mathbf{V}^T = \mathbf{I}$)
qz	Γενικευμένες Ιδιοτιμές
rsf2csf	Πραγματική block διαγώνια μορφή σε μιγαδική διαγώνια μορφή
cdf2rdf	Μιγαδική διαγώνια μορφή σε πραγματική block διαγώνια μορφή
schur	Διάσπαση Schur ($[\mathbf{U}, \mathbf{T}] = \mathbf{schur}(\mathbf{X}) \Leftrightarrow \mathbf{X} = \mathbf{U} * \mathbf{T} * \mathbf{U}^T$ και $\mathbf{U}^T * \mathbf{U} = \mathbf{I}$)
balance	Διαγώνιο scaling για βελτίωση της ακρίβειας της ιδιοτιμής ($[\mathbf{T}, \mathbf{B}] = \mathbf{balance}(\mathbf{A}) \Leftrightarrow \mathbf{B} = \mathbf{T}^{-1} * \mathbf{A} * \mathbf{T}$)

1.2.7 Συναρτήσεις

Στο περιβάλλον MATLAB παρέχεται η δυνατότητα στον χρήστη είτε να εφαρμόσει τις έτοιμες συναρτήσεις βιβλιοθήκης, είτε να κατασκευάσει νέες δικές του συναρτήσεις.

Μία συνάρτηση ορίζεται γενικά ως εξής :

function[output_1, output_2, ... , output_m] = fun_name(input_1, input_2, ... , input_n).

1.2.8 Αριθμητικά Παραδείγματα

1. Υπολογισμός του αριθμού $eps = \epsilon_{μηχανής}$

```
>> eps
```

Αποτέλεσμα: $2.2204e - 16$

2. Εισαγωγή ενός πίνακα και σχηματισμός υποπινάκων

```
% Εισαγωγή ενός πίνακα A
```

```
A = [1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15; 16 17 18 19 20]
```

```
% Επιλογή γραμμών του A
```

```
L1 = A(1, :)
```

```
L2 = A(2, :)
```

```
L3 = A(3, :)
```

```
L4 = A(4, :)
```

```
% Επιλογή στηλών του A
```

```
C1 = A(:, 1)
```

```
C3 = A(:, 3)
```

```
% Επιλογή στοιχείων από μια γραμμή του A
```

```
E = A(2, 1 : 2 : 5)
```

```
% Σχηματισμός Υποπινάκων του A
```


$$B1=A([2\ 4], :)$$

$$B2=A([3\ 2], [5\ 4])$$

$$B3=A(4:-1:1, 5:-1:1)$$

Αποτέλεσμα:

$$A = \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \end{array}$$

$$L1 = \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array}$$

$$L2 = \begin{array}{ccccc} 6 & 7 & 8 & 9 & 10 \end{array}$$

$$L3 = \begin{array}{ccccc} 11 & 12 & 13 & 14 & 15 \end{array}$$

$$L4 = \begin{array}{ccccc} 16 & 17 & 18 & 19 & 20 \end{array}$$

$$C1 = \begin{array}{c} 1 \\ 6 \\ 11 \\ 16 \end{array}$$

$$C3 = \begin{array}{c} 3 \\ 8 \\ 13 \\ 18 \end{array}$$

$$E = \begin{array}{ccc} 6 & 8 & 10 \end{array}$$

$$B1 = \begin{array}{ccccc} 6 & 7 & 8 & 9 & 10 \\ 16 & 17 & 18 & 19 & 20 \end{array}$$

$$B2 = \begin{array}{cc} 15 & 14 \\ 10 & 9 \end{array}$$

$$B3 = \begin{array}{ccccc} 20 & 19 & 18 & 17 & 16 \\ 15 & 14 & 13 & 12 & 11 \\ 10 & 9 & 8 & 7 & 6 \\ 5 & 4 & 3 & 2 & 1 \end{array}$$

3. Εισαγωγή ενός διανύσματος

```
>> b = [1 0 0 0]'
```

Αποτέλεσμα:

```
b =  
    1  
    0  
    0  
    0
```

4. Επίλυση του γραμμικού συστήματος $Ax=b$

```
A =  
    5    7    6    5  
    7   10    8    7  
    6    8   10    9  
    5    7    9   10
```

```
>> x = A\b
```

Αποτέλεσμα:

```
x =  
    68  
   -41  
   -17  
    10
```

5. Εύρεση του αντιστρόφου του πίνακα A

```
>> inv(A)
```

Αποτέλεσμα:

```
ans =  
    68   -41   -17    10  
   -41    25    10    -6  
   -17    10     5    -3  
    10    -6    -3     2
```

6. Υπολογισμός των Ιδιοτιμών του πίνακα A

```
>> eig(A)
```

Αποτέλεσμα:

```
ans =  
    0.0102  
    0.8431  
    3.8581  
   30.2887
```

7. LU παραγοντοποίηση του πίνακα A με μερική οδήγηση (PA = LU)

```
>> [L,U,P] = lu(A)
```

Αποτέλεσμα:

```
L =  
 1.0000    0    0    0  
 0.8571  1.0000    0    0  
 0.7143  0.2500  1.0000    0  
 0.7143  0.2500 -0.2000  1.0000
```

```
U =  
 7.0000  10.0000  8.0000  7.0000  
    0 -0.5714  3.1429  3.0000  
    0    0  2.5000  4.2500  
    0    0    0  0.1000
```

```
P =  
 0  1  0  0  
 0  0  1  0  
 0  0  0  1  
 1  0  0  0
```

1.3 m-αρχεία(files)

1.3.1 Δημιουργία m-αρχείου

Στο περιβάλλον Matlab μπορούμε να δημιουργήσουμε ένα **m-αρχείο**(ή πηγαίο πρόγραμμα) ακολουθώντας την διαδρομή επιλογών **File** → **New** → **M-File** και δίνοντας ένα όνομα υποχρεωτικά με προέκταση **.m**. Για να εκτελέσετε ένα πρόγραμμα πληκτρολογείτε το όνομά του χωρίς την προέκταση **.m** δηλαδή **>> όνομα** .

1.3.2 Εκτέλεση και επεξεργασία m-αρχείων

Ένα πρόγραμμα για να εκτελεστεί μπορεί να χρειάζεται να αποκτήσει πληροφορίες από περισσότερα του ενός **m-αρχεία**. Συνεπώς, όλα τα αρχεία που σχετίζονται με το πρόγραμμα που θέλουμε να εκτελέσουμε πρέπει να τοποθετηθούν στον ίδιο φάκελο.

Ένα **m-αρχείο** μπορεί να περιέχει είτε μια ομάδα εντολών που θα πραγματοποιούν μια συγκεκριμένη εργασία (**script** αρχείο) ή μια συνάρτηση(υποπρόγραμμα) (function αρχείο) με κατάλληλο όνομα και αντίστοιχες τυπικές μεταβλητές εισόδου και εξόδου. Η βασική διαφορά στους δύο τύπους **m-αρχείων** είναι ότι σε ένα **script** αρχείο όλες οι μεταβλητές είναι καθολικές ενώ σε ένα function αρχείο είναι όλες τοπικές. Για να εκτελέσουμε ένα **m-αρχείο** που περιέχει μόνο εντολές, πληκτρολογούμε στο παράθυρο εντολών(**Command Window**) του MATLAB μόνο το όνομα του αρχείου, δηλ. για το **m-αρχείο mywork.m** πληκτρολογούμε τη λέξη **mywork**. Για να εκτελέσουμε όμως ένα **m-αρχείο** στο οποίο περιέχεται μία συνάρτηση (function) πρέπει να πληκτρολογήσουμε το όνομα του m-αρχείου και υποχρεωτικά τα αντίστοιχα πραγματικά ορίσματα, όπως αυτά ορίζονται στη function. Η γενική μορφή ενός function αρχείου είναι

```
function [output1, output2, ..., outputn] = program_name (input1, input2, ..., inputm)
```

Π.χ. για να εκτελεστεί το **m-αρχείο mywork.m** όταν μέσα σε αυτό περιέχεται το υποπρόγραμμα **function = fun(x,y)**, πληκτρολογούμε **mywork(3,5)** για να υπολογιστεί η τιμή **fun(2,3)** της συνάρτησης για $x = 2$, $y = 3$.

Όταν θέλουμε να επεξεργαστούμε ένα **m**-αρχείο, αλλά δεν έχουμε εγκαταστήσει το πρόγραμμα του MATLAB, δηλαδή δεν έχουμε στη διάθεσή μας τον **m-file editor**, τότε μπορούμε να χρησιμοποιήσουμε έναν οποιοδήποτε άλλο editor, όπως οι **WinEdt**, **WordPad**, **NotePad** για να επεξεργαστούμε τον πηγαίο κώδικα.

1.3.3 Βασικές Εντολές

1. Η εντολή **for**

```
for μεταβλητή = αρχική τιμή μεταβλητής : τελική τιμή μεταβλητής
    εντολές
    :
end;
```

2. Η εντολή **while**

```
while συνθήκη
    εντολές
    :
end;
```

3. Η εντολή **if**

```
if συνθήκη
    εντολές
    :
else
    εντολές
    :
end ;
```

1.4 Απλά Προγράμματα σε MATLAB

Πρόγραμμα 1.

Το ακόλουθο πρόγραμμα υπολογίζει τον αριθμό $e = \epsilon_{μηχανής}$.

```
% eps.m : Υπολογισμός του  $\epsilon_{μηχανής}$ .
e = 1.0;
while 1.0 + e > 1.0
    e = e/2.0
end
```

$$e = 2.0 * e$$

Πρόγραμμα 2.

Το ακόλουθο πρόγραμμα δημιουργεί ένα τυχαίο 4×4 πίνακα A και μηδενίζει όλα τα στοιχεία του κάτω από τη κύρια διαγώνιο.

```
% tyx_pin.m : Δημιουργία τυχαίου πίνακα .  
A=rand(4,4)  
for i = 1:4  
    for j=1:4  
        if i > j  
            A(i,j)=0;  
        end;  
    end;  
end;
```

Πρόγραμμα 3.

Το ακόλουθο πρόγραμμα είναι μια συνάρτηση(function) που δέχεται ως είσοδο δύο άνω τριγωνικούς πίνακες U και V και δίνει ως έξοδο το γινόμενό τους $C = U * V$

```
function C = matmat(U,V)  
[n,m] = size(U)  
C = zeros(n,n)  
for i = 1:n  
    for j = i:n  
        for k = i:j  
            C(i,j) = C(i,j) + U(i,k) * V(k,j);  
        end;  
    end;  
end;  
end;
```

Πρόγραμμα 4.

Το ακόλουθο πρόγραμμα είναι μια συνάρτηση(function) που δέχεται ως είσοδο ένα διάνυσμα x και δίνει ως έξοδο τη **2-norm** του x .

```
% norm_2.m : Υπολογισμός της 2-νορμ ενός διανύσματος x  
function nrm = twonorm(x)  
[n,m] = size(x);  
d=max(abs(x));  
y = x/d;  
s = 0;  
for i=1:n  
    s = s + y(i)^2;  
end;  
nrm=d*s^0.5;  
end;
```