Matlab

Μάθημα 2

Νέο υλικό

- www.cs.uoi.gr/~develeg
  - Matlab2.pdf - Παρουσίαση μαθήματος 2.
  - Matlab-reference.pdf – Σημειώσεις matlab στα ελληνικά (13 σελίδες).

Επαναληπτικές δομές

- **Όταν εκτελείται μια πράξη σε ένα διάνυσμα, όπως ο υπολογισμός του τετραγώνου των στοιχείων του, η Matlab στην ουσία, διατρέχει ένα-ένα τα στοιχεία του διανύσματος, και αντικαθιστά με το τετράγωνο του αριθμού σε κάθε θέση. Αυτή η διαδικασία είναι επαναληπτική.**
  - A = [1 2 3 4];
  - B = A.^2
  - Έξοδος: B = 1 4 9 16
- Στη γλώσσα της matlab υπάρχουν δυο επαναληπτικές δομές: for και while.

for

```
>>for i=1:10
    i^2
end
```

```
ans =

     1

ans =

     4

ans =

     9

ans =

    16

...

ans =

   100
```

For – δείκτης σε διάνυσμα

```
>> a=[3,40,1,8,0]
b=[-7,5,-.4,16,1]
c=[ ]
for i=1:5
  c(i) = a(i) + b(i)
end

ή

for i=1:length(a)
  c(i) = a(i) + b(i)
end
```

```
a =
     3    40     1     8     0
b =
   -7.0000    5.0000   -0.4000   16.0000    1.0000
c =
     []
c =
    -4
c =
    -4    45
c =
   -4.0000   45.0000    0.6000
c =
   -4.0000   45.0000    0.6000   24.0000
c =
   -4.0000   45.0000    0.6000   24.0000    1.0000
```

For – δείκτες σε πίνακα

```
m=[1:4;5:8;9:12]

for i=1:3
  for j=1:4
    m(i,j) = m(i,j) * 2;
  end
end
m
```

```
m =

     1     2     3     4
     5     6     7     8
     9    10    11    12

m =

     2     4     6     8
    10    12    14    16
    18    20    22    24
```

Το ίδιο αποτέλεσμα με την πράξη στην matlab: **m=m.*2**

For – δείκτες σε πίνακα

```
m=[1:4;5:8;9:12]
l = size(m)
for i=1:l(1)
  for j=1:l(2)
    m(i,j) = m(i,j) * 2;
  end
end
m
```

```
m =
    1     2     3     4
    5     6     7     8
    9    10    11    12

l =
    3  4

m =
    2     4     6     8
   10    12    14    16
   18    20    22    24
```

l = size(m) επιστρέφει διάνυσμα με αριθμό στοιχείων όσες είναι οι διαστάσεις του πίνακα m, το πρώτο στοιχείο είναι ο αριθμός των γραμμών και δεύτερο στοιχείο ο αριθμός των στηλών κτλ.

while

```
>> a=[3,40,1,8,0]
b=[-7,5,-.4,16,1]
c=[ ]
while (i <= 5)
  c(i) = a(i) + b(i)
  i = i + 1;
end
```

```
a =
     3    40     1     8     0
b =
   -7.0000    5.0000   -0.4000   16.0000    1.0000
c =
     [ ]
c =
    -4
c =
    -4    45
c =
   -4.0000   45.0000    0.6000
c =
   -4.0000   45.0000    0.6000   24.0000
c =
   -4.0000   45.0000    0.6000   24.0000    1.0000
```

Οι αριθμοί Fibonacci

- Τοποθετούμε ένα ζεύγος κουνελιών σε κλειστό χώρο. Πόσα ζεύγη κουνελιών θα παραχθούν σε ένα χρόνο, με την προϋπόθεση ότι κάθε μήνα κάθε ζεύγος παράγει ένα νέο ζεύγος, το οποίο γίνεται παραγωγικό από τον δεύτερο μήνα και μετά;

Οι αριθμοί Fibonacci

- f(n) αριθμός ζευγών μετά από n μήνες.
- f(1) = 1
- f(2) = 2
- f(n) = (*αριθμός ζευγών κατά την έναρξη του μήνα n, συν τα ζεύγη που παράγονται από τα γόνιμα ζεύγη τον μήνα n*)
  = f(n-1) + f(n-2)

Παραγωγή αριθμών Fibonacci

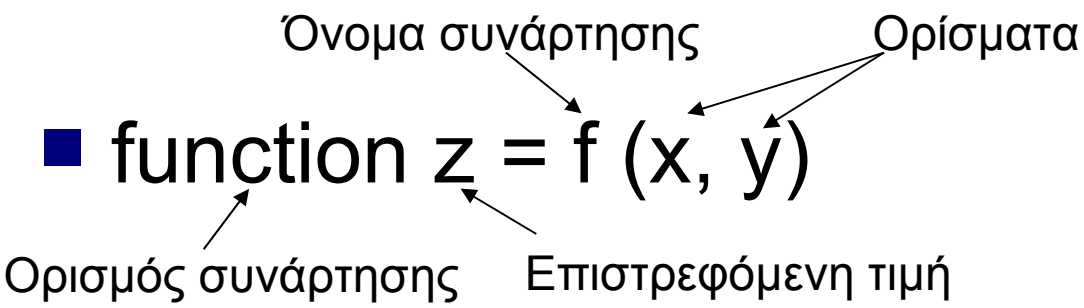
- f(n) = f(n-1) + f(n-2)
- Αρχικές συνθήκες f(1) = 1, f(2) = 2

```
f = zeros(12,1);
f(1)=1;
f(2) = 2;
for i=3:length(f)
   f(i) = f(i - 1) + f(i - 2);
end
f
```

```
f =

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

Συνάρτηση

- Για να φτιάξουμε μια συνάρτηση στην matlab, δημιουργούμε ένα m-file με



- Στην συνέχεια γραφούμε τις εντολές της συνάρτησης (θα πρέπει να εκχωρείται κάποια τιμή στην μεταβλητή επιστροφής z).

Συνάρτηση fibonacci.m

```
function f = fibonacci(n)
% f = fibonacci(n) παράγει τους πρώτους n αριθμούς της ακολουθίας Fibonacci
f = zeros(n,1);
f(1) = 1;
f(2) = 2;
for i = 3:n
    f(i) = f(i - 1) + f(i - 2);
end
```

```
>> fibonacci(12)

ans =

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

Η κλήση της συνάρτησης στο Command window γίνεται με το όνομα του m-file. Καλή τακτική το m-file και η συνάρτηση να έχουν το ίδιο όνομα.

Συνάρτηση fibonacci.m

```
function f = fibonacci(n)
% f = fibonacci(n) παράγει τους πρώτους n αριθμούς της ακολουθίας Fibonacci
f = zeros(n,1);
f(1) = 1;
f(2) = 2;
for i = 3:n
    f(i) = f(i - 1) + f(i - 2);
end
```

```
>> help fibonacci
 f = fibonacci(n) παράγει τους πρώτους
n αριθμούς της ακολουθίας Fibonacci
```

Με το σύμβολο % βάζουμε σχόλια. Τα σχόλια αμέσως μετά τον ορισμό της συνάρτησης εμφανίζονται όταν εκτελούμε help ονομα_συνάρτησης, για αυτό καλό είναι να δίνουμε σχόλια για το τι κάνει η συνάρτηση και τον τρόπο κλίσης της.

Επιλογή

```
if (λογική συνθήκη)
    Ομάδα εντολών
end
```

```
if (λογική συνθήκη)
    Ομάδα εντολών 1
else
    Ομάδα εντολών 2
end
```

```
if (λογική συνθήκη 1)
    ομάδα εντολών 1
elseif (λογική συνθήκη 2)
    ομάδα εντολών 2
else
    ομάδα εντολών 3
end
```

Λογικές Συνθήκες

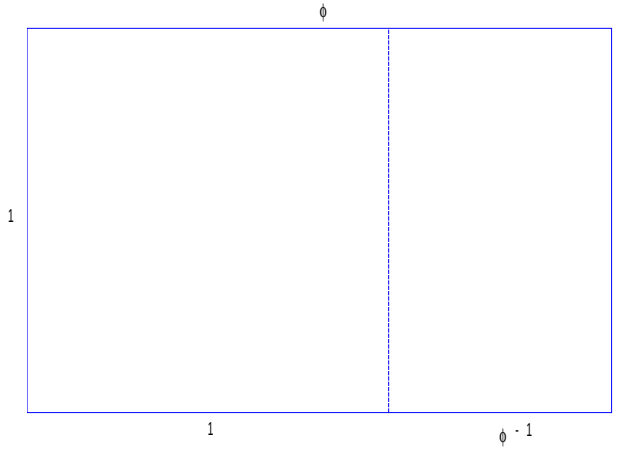
- Matlab αναπαριστά τις τιμές true και false με τους ακεραίους 0 και 1.
  - true = 1
  - false = 0
- Έστω ότι στη μεταβλητή x, έχει εκχωρηθεί κάποιο βαθμωτό μέγεθος, μπορούν να ελεγχθούν οι παρακάτω λογικές συνθήκες:
  - x == 2 είναι το x ίσο με 2;
  - x ~= 2 είναι το x διάφορο του 2;
  - x > 2 είναι το x μεγαλύτερο του 2;
  - x < 2 είναι το x μικρότερο του 2;
  - x >= 2 είναι το x μεγαλύτερο ή ίσο του 2;
  - x <= 2 είναι το x μικρότερο ή ίσο του 2;
- Σύνθετες λογικές συνθήκες
  - x > 0 && x <=2 είναι το x μεγαλύτερο του 0 **και** μικρότερο ή ίσο του 2;
  - x == 1 || x ==2 είναι το x ίσο με 1 **ή** ίσο με 2;

```
function A = area_ellipse(a, b)
% A = area_ellipse(a, b)
% υπολογίζει το εμβαδόν μίας έλλειψης όπου τα ορίσματα a και b
% αντιστοιχούν στο μισό του μικρού και του μεγάλου άξονα.
% Για κλήσεις με ένα όρισμα a = b (κύκλος)

error(nargchk(1, 2, nargin))

if (nargin == 1)
  b=a;
end

if ((a < 0) || (b < 0))
  A=0;
else
  A=pi*a*b;
end
```

```
>>area_ellipse
??? Error using ==> area_ellipse
Not enough input arguments.

>> area_ellipse(4,5,6)
??? Error using ==> area_ellipse
Too many input arguments.

>> area_ellipse(4,5)
ans =

   62.8319

>> area_ellipse(-4,5)
ans =

     0
```

Ο Χρυσός Λόγος (φ)

- Χρυσό ορθογώνιο
- Με την αφαίρεση ενός τετραγώνου προκύπτει ορθογώνιο με τον ίδιο λόγο πλευρών.



$$\frac{1}{\phi} = \frac{\phi - 1}{1} \Rightarrow \varphi^2 - \varphi - 1 = 0 \Rightarrow \varphi = \frac{1 \pm \sqrt{5}}{2}$$

Ο Χρυσός Λόγος (φ)

```
>> phi=(1+sqrt(5))/2

phi =

    1.6180

>> format long
>> phi

phi =

   1.61803398874989
```

Υπολογισμός ριζών πολυωνύμου

- Το διάνυσμα p=[1 -1 -1] αναπαριστά το πολυώνυμο $p(x)=x^2 - x - 1$.
- Η συνάρτηση roots(p) υπολογίζει τις ρίζες του.

```
>> p=[1 -1 -1]

p =

     1    -1    -1

>> r=roots(p)

r =

   1.61803398874989
  -0.61803398874989
```

Συνάρτηση inline('string')

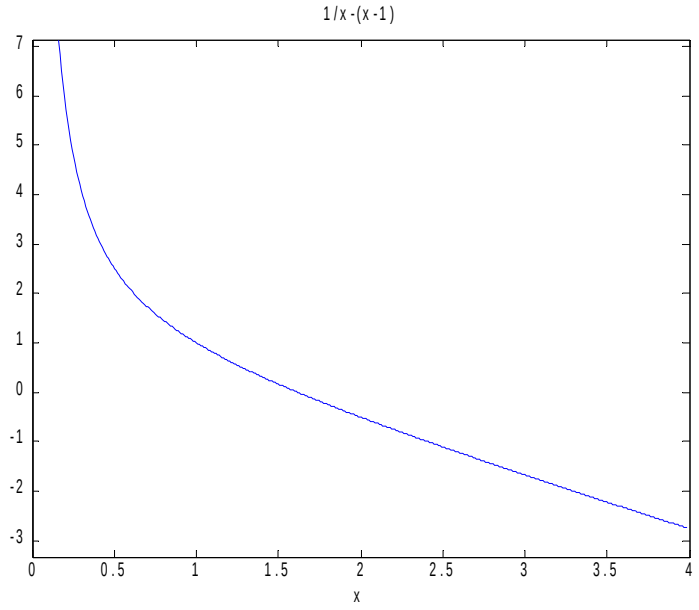
- Μετατρέπει συμβολοσειρές εκφράσεων σε αντικείμενα που μπορούν να χρησιμοποιηθούν ως ορίσματα σε άλλες συναρτήσεις.

```
>> f=inline('1/x-(x-1)')

f =

     Inline function:
     f(x) = 1/x-(x-1)
```

Συνάρτηση ezplot(f,a,b)

- Κατασκευάζει το γράφημα της συνάρτησης f = f(x) στο διάστημα a < x < b.

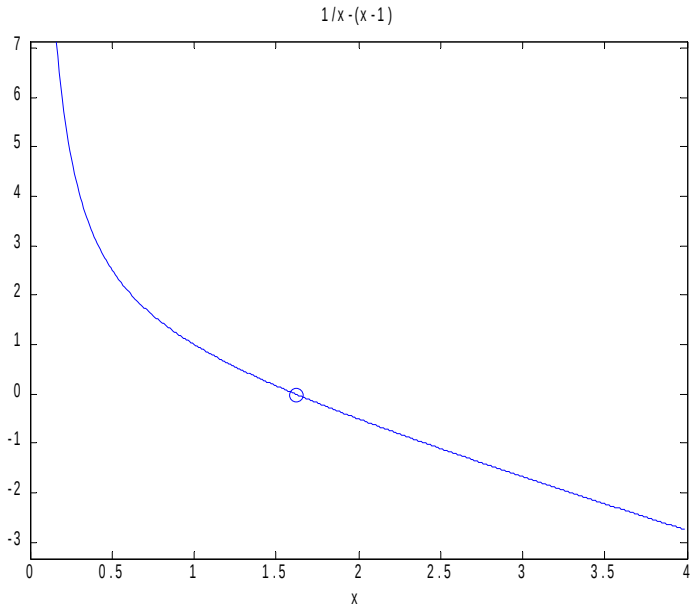
```
>>ezplot('1/x-(x-1)',0,4)
ή
>>ezplot(f,0,4)
```



Συνάρτηση fzero(f,xo)

- Ψάχνει για f(x) = 0 κοντά στο xo

```
>> f=inline('1/x-(x-1)')

f =

     Inline function:
     f(x) = 1/x-(x-1)

>>ezplot(f,0,4)

>>phi=fzero(f,1)

phi =

   1.61803398874990

>>hold on

>>plot(phi, 0,'o')
```



Σχέση ακολουθίας Fibonacci και φ

$$\lim_{n \to \infty} \frac{f_{n+1}}{f_n} = \phi$$

```
>> f=fibonacci(n);
>> f(2:n)./f(1:n-1)

ans =

   2.00000000000000
   1.50000000000000
   1.66666666666667
   1.60000000000000
   1.62500000000000
...
   1.61803398874989
   1.61803398874989

>> f(n)/f(n-1)- phi

ans =

     0
```

Εύρεση λύσης closed-form

- $f_n = c\rho^n$ για σταθερές c και ρ.
- $f_n = f_{n-1} + f_{n-2} \rightarrow \rho^2 = \rho + 1$ με ρίζες ϕ και $1-\phi$
- Γενική λύση: $f_n = c_1\phi^n + c_2(1-\phi)^n$
- Από τις αρχικές συνθήκες:

$$f_0 = c_1 + c_2 = 1$$
$$f_1 = c_1\phi + c_2(1-\phi) = 1$$

$$\Rightarrow \quad c_1 = \frac{\phi}{2\phi - 1}, \; c_2 = -\frac{(1-\phi)}{2\phi - 1}$$

$$f_n = \frac{1}{2\varphi - 1}\left(\varphi^{n+1} - (1-\varphi)^{n+1}\right)$$

```
>> format long e
>> n=(1:40)';
>> f=(phi.^(n+1) - (1-phi).^(n+1))/(2*phi-1)
f =
    1.000000000000000e+000
    2.000000000000000e+000
    3.000000000000000e+000
    5.000000000000000e+000
    8.000000000000000e+000
    1.300000000000000e+001
    2.100000000000000e+001
...
    1.023341550000001e+008
    1.655801410000002e+008
>> f = round(f)
f =
           1
           2
           3
...
   102334155
   165580141
```