Προγραμματισμός & Εφαρμογές Η/Υ

Επαναλήψεις
Εισαγωγή στις Γραφικές Παραστάσεις
Εισαγωγή και εξαγωγή δεδομένων

Π. Οικονόμου

Τμήμα Πολιτικών Μηχανικών

2023–2024

MATLAB®

Εντολές επαναλήψεων

for – while

- **Συντακτικό for**

```
for n = start:finish
    εντολές
end
```

```
for n = start:step:finish
    εντολές
end
```

- **Συντακτικό while**

```
while συνθήκη
    εντολές
end
```

Παράδειγμα 1

Δοθέντος του διανύσματος $len = [-5, -1 : 0.5 : 2, -3, 2]$, να φτιάξετε ένα διάνυσμα που να έχει

- την τιμή -1 στις θέσεις που το *len* έχει αρνητικό αριθμό
- την τιμή 0 στις θέσεις που το *len* έχει μηδέν και
- την τιμή 1 στις θέσεις που το *len* έχει θετικό αριθμό.

Παραδείγματα 2α και 2β

- Υπολογίστε και αποθηκεύστε το άθροισμα $\sum_{i=1}^{n} i$ για $\forall n \in \{1, 2, \ldots, 10000\}$.

- Σχολιάστε τι κάνουν οι επόμενες εντολές

```
s = 0;
power = 0;
for  n = 1:2:100000
        s = s+(-1)^power*1/n;
        power = power+1;
end
4*s
```

Οι εντολές continue - break

continue

Παρατηρήστε τις διαφορές στα αποτελέσματα των δυο επόμενων group εντολών

```
for k = -10:1:10
    if (k^2-50 < 0)
        continue;
    end
    val = k^2-50;
    disp([k, val])
end

   -10    50
    -9    31
    -8    14
     8    14
     9    31
    10    50
```

```
for k = -10:1:10
    if (k^2-50 < 0)
        %continue;
    end
    val = k^2-50;
    disp([k, val])
end

   -10    50
    -9    31
    -8    14
    -7    -1
    -6   -14
    -5   -25
    -4   -34
....
```

Οι εντολές continue - break

break

Παρατηρήστε τις διαφορές στα αποτελέσματα των δυο επόμενων group εντολών

```
x = [1,2,3,4,5,6,-21,10];
sum = 0;
for i = 1:length(x)
   % check for negative value
    if x(i) < 0
       % terminate loop execution
       break
    end
   % update the sum
   sum = sum+x(i);
end
sum

sum =
    21
```

```
x = [1,2,3,4,5,6,-21,10];
sum = 0;
for i = 1:length(x)
   % check for negative value
    if x(i) < 0
       % terminate loop execution
       %break
    end
   % update the sum
   sum = sum+x(i);
end
sum

sum =
    10
```

Εξετάσεις 2/2014

Περιγράψτε εν συντομία τι κάνουν οι παρακάτω κώδικες

```
s = 1;
n = 1;
while n^(sqrt(n)) < 10^4
      s = s+n^(sqrt(n));
      n = n+1;
end
[n,s]
```

```
s1 = 0;
s2 = s1;
for k = 1:100
    s1 = s1+(-1)^k*k^2;
    s2 = s2+(-1)^k*k^3;
end
```

Εξετάσεις 2/2014

- Να υπολογιστεί το άθροισμα

$$\sum_{n=1}^{K-1} \eta\mu(\sigma\upsilon\nu^n(6))$$

όπου το K είναι ο πρώτος (ελάχιστος) φυσικός αριθμός για τον οποίο δεν ικανοποιείται η σχέση $\eta\mu(\sigma\upsilon\nu^K(6)) > 0.3$. Μεταφέρετε τον κώδικα καθώς και το αποτέλεσμά σας στην κόλλα σας.

- Να υπολογιστεί το άθροισμα των AM πρώτων όρων της ακολουθίας

$$a(n) = (1 - (1/AM)^n)/(1 - 1/AM)$$

Μεταφέρετε τον κώδικα καθώς και το αποτέλεσμά σας στην κόλλα σας.

Εξετάσεις 6/2014

Περιγράψτε εν συντομία τι κάνουν οι παρακάτω κώδικες

```

count = 0;
total = 1;
while(total <= 1000000)
      total = total*2;
      count = count+1;
end
count
clear total
```

Παράδειγμα 3

Η βασική ιδέα της μεθόδου του Ήρων (Heron's formula) για την αριθμητική προσέγγιση της τετραγωνικής ρίζας ενός αριθμού α είναι ότι αν το x είναι μεγαλύτερο της $\sqrt{\alpha}$ τότε το α/x είναι μικρότερο της $\sqrt{\alpha}$ και έτσι ο μέσος όρος αυτών των δύο αριθμών μπορεί να θεωρηθεί ότι προσφέρει μια καλύτερη προσέγγιση της $\sqrt{\alpha}$.
Η μέθοδος του Ήρων ξεκινά με μια αυθαίρετη εκτίμηση x της $\sqrt{\alpha}$, η οποία ανανεώνεται από την

$$x := \frac{x + \frac{\alpha}{x}}{2}.$$

Η διαδικασία αυτή επαναλαμβάνεται συνεχώς μέχρις ότου η απόλυτη διαφορά *tol* της νέας εκτίμησης της $\sqrt{\alpha}$ από την προηγούμενη να είναι μικρότερη από μια προκαθορισμένη μικρή ποσότητα.

Να εφαρμόσετε τη μέθοδο του Ήρων για την εύρεση της $\sqrt{154563}$ χρησιμοποιώντας

- την τιμή 1.8 ως αρχική εκτίμηση της $\sqrt{154563}$ και
- την τιμή 10^{-3} για το κριτήριο σύγκλισης *tol*.

Παράδειγμα 3 - Λύση

```
%uses the Heron's formula to determine the square root of a positive number

a = input('Enter a: ');

xinitial = input('Enter initial guess: ');

tol = 10^(-3);
xnew = (xinitial+a/xinitial)/2;
dif = abs(xinitial-xnew);

loop = 0;

while dif >= tol
        xinitial = xnew;
        xnew = (xinitial+a/xinitial)/2;
        loop = loop+1;
        dif = abs(xinitial-xnew);
end

disp(['You have found the square root of ',num2str(a), ' after ', num2str(loop), ' loops.'])
disp(['The square root of ',num2str(a), ' is ', num2str(xnew)])
```

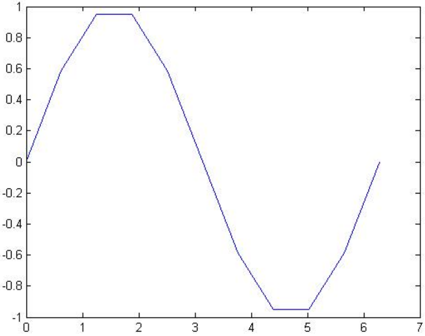
Παράδειγμα 4

```
% prime factorization of positive integers until the first next prime

start = input('enter the starting value: ');
x = start;

if isprime(x)

    disp([num2str(x), ' is  prime'])

else while ~isprime(x)

        disp(['The prime factorization of ', num2str(x), ' is: ', num2str(factor(x))])
        x = x+1;

     end

     disp(['The prime factorization of ', num2str(x), ' is: ', num2str(factor(x))])
     disp([num2str(x),' is the first prime larger than ',num2str(start)])

end
```

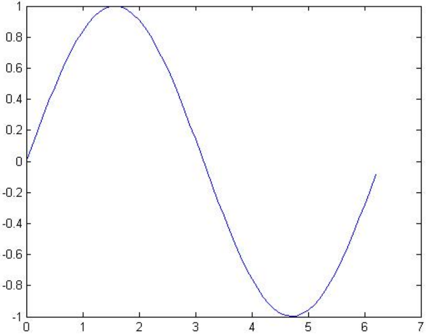
Γραφική παράσταση συνάρτησης

plot(x,y)

```
>> x=0:2*pi/10:2*pi;
>> y=sin(x);
>> plot(x,y) % plot(x,sin(x))
```




```
>> x=0:2*pi/100:2*pi;
>> y=sin(x);
>> plot(x,y) % plot(x,sin(x))
```

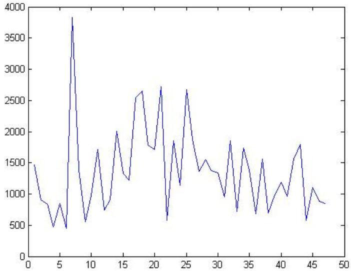


Γράφημα χρονοσειράς

plot(y)

Αν καλέσουμε την εντολή plot(y) με ένα όρισμα (διάνυσμα) τότε απεικονίζει τα στοιχεία του y ως προς τον αύξοντα αριθμό τους (γράφημα χρονοσειράς)

```
>> rainfall=importdata('rainfall.dat');
>> plot(rainfall)
```



Σχόλιο 2: η εντολή *importdata* ..... (βλ. και διαφάνεια 23)

Παραδείγματα

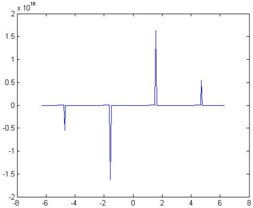
1. Κατασκευάστε το ευθύγραμμο τμήμα που ενώνει τα σημεία (2,3) και (10,5).

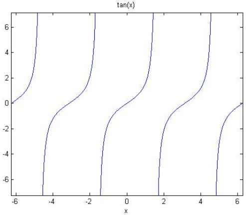
```
>>plot([2,10],[3,5])
```

2. Κατασκευάστε τη γραφική παράσταση της εφαπτομένης στο διάστημα (-2π,2π).

```
>> x=-2*pi:4*pi/200:2*pi;
>> y=tan(x);
>> plot(x,y)
```

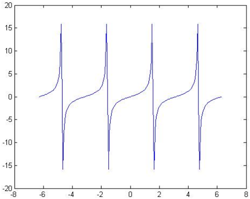
```
>> ezplot('tan(x)')
```





Παραδείγματα – Συνέχεια

```
>> x=-2*pi:4*pi/200:2*pi;
>> y=tan(x);
>> y=y.*(abs(y)<1e5);
>> plot(x,y)
```



Σχόλιο 1: Η πράξη $(.*)$ πολλαπλασιάζει τα στοιχεία των διανυσμάτων (ή των πινάκων) στοιχείο προς στοιχείο. Δηλαδή αν $x1 = [2, 3]; x2 = [4, 0];$ τότε $x1. * x2 = [8, 0]$.
Αντίθετα, η πράξη $(*)$ ανάμεσα σε διανύσματα εκτελεί το αποκαλούμενο εσωτερικό γινόμενό τους, ενω ανάμεσα σε πίνακες εκτελεί τον κλασικό πολλαπλασιασμό πινάκων.
Περισσότερα σε επόμενα μαθήματα.
Σχόλιο 2: η εντολή $(abs(y) < 1e5)$ .....

Μορφοποίηση γραμμών (σημάδια-χρώμα)

```
>> x=-2*pi:4*pi/200:2*pi;
>> y=tan(x);
>> y=y.*(abs(y)<1e5);
>> plot(x,y,'-k.')
```

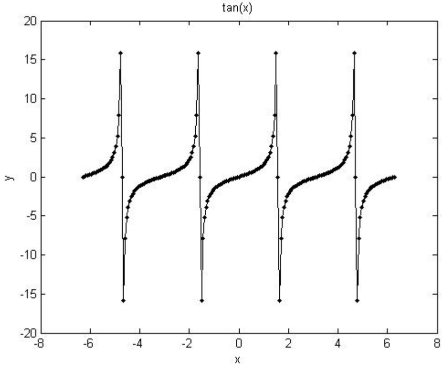
options

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Labels

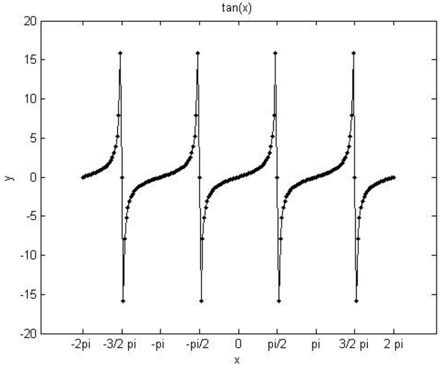
title, xlabel, ylabel

```
>> x=-2*pi:4*pi/200:2*pi;
>> y=tan(x);
>> y=y.*(abs(y)<1e5);
>> plot(x,y,'-k.')
>> title('tan(x)'); xlabel('x'); ylabel('y')
```



Μορφοποίηση αξόνων

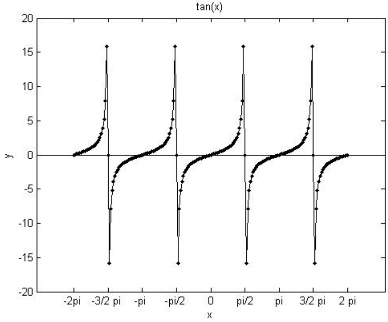
```
>>set(gca,'XTick',-2*pi:pi/2:2*pi)
>>set(gca,'XTickLabel',{'-2pi','-3/2 pi','-pi','-pi/2','0','pi/2','pi','3/2 pi','2 pi'})
```

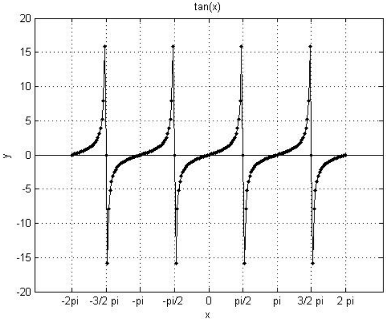


gca returns the handle to the current axis in the current figure

Εισαγωγή γραμμών αναφοράς – πλέγματος

```
>>hline=refline(0,0)
>>set(hline,'Color','k')
>>grid
```

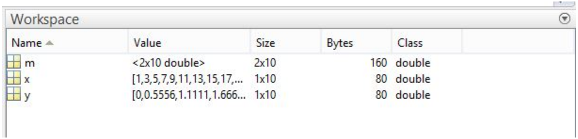




εξαγωγή και εισαγωγή μεταβλητών

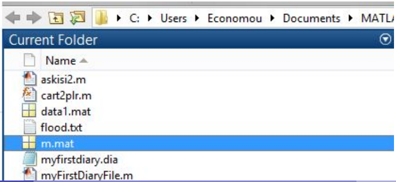
save – load (βλ. και άσκηση 3 Εργαστήριο 1)

```
>> x=1:2:20;
>> y=linspace(0,5,10);
>> m=[x;y];
```



Name	Value	Size	Bytes	Class
m	<2x10 double>	2x10	160	double
x	[1,3,5,7,9,11,13,15,17,...	1x10	80	double
y	[0,0.5556,1.1111,1.666...	1x10	80	double

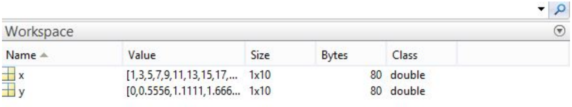
```
>> save m
```



εξαγωγή και εισαγωγή μεταβλητών

save – load

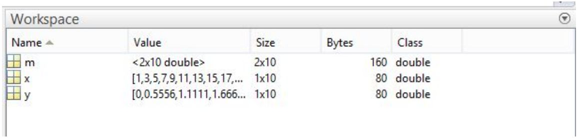
```
>> clear m
```



Workspace

Name	Value	Size	Bytes	Class
x	[1,3,5,7,9,11,13,15,17,...	1x10	80	double
y	[0,0.5556,1.1111,1.666...	1x10	80	double

```
>> load m
```

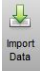


Workspace

Name	Value	Size	Bytes	Class
m	<2x10 double>	2x10	160	double
x	[1,3,5,7,9,11,13,15,17,...	1x10	80	double
y	[0,0.5556,1.1111,1.666...	1x10	80	double

εισαγωγή και εξαγωγή αρχείων

Συχνά θέλουμε να εισάγουμε μεγάλο όγκο δεδομένων από αρχεία που έχουν δημιουργηθεί από άλλα προγράμματα.

Χρήσιμα εργαλεία, για να το πετύχουμε αυτό, είναι το import tool 

καθώς και οι εντολές

- load

  ```
  >>load filename
  ```

- importdata

  ```
  >>datain=importdata('filename')
  ```

* fprintf – Παράδειγμα

```
fprintf('The minimum temperature today\n')
fprintf('has been %.1f degree C. (%g degree F.)\n',3.9,3.9*9/5 + 32)
fprintf('The maximum temperature today\n')
fprintf('has been %1.2f degree C. (%7.3f degree F.)\n',23.1,23.1*9/5 + 32)

The minimum temperature today
has been 3.9 degree Celsius (39.02 degree Fahrenheit)
The maximum temperature today
has been 23.10 degree Celsius ( 73.580 degree Fahrenheit)
```

Για περισσότερες πληροφορίες για την εντολή fprintf μπορείτε να επισκεφθείτε τις σελίδες
http://nf.nci.org.au/facilities/software/Matlab/techdoc/ref/fprintf.html
http://web.ics.purdue.edu/ mverleg1/ENGR126/Guides/fprintf.htm