



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

ΠΛΗΡΟΦΟΡΙΚΗ Ι

Ενότητα 5: Πίνακες [1/2] (Διανύσματα)

Μιχάλης Δρακόπουλος

Σχολή Θετικών επιστημών

Τμήμα Μαθηματικών

ΠΛΗΡΟΦΟΡΙΚΗ Ι (MATLAB)

Ενότητα 5

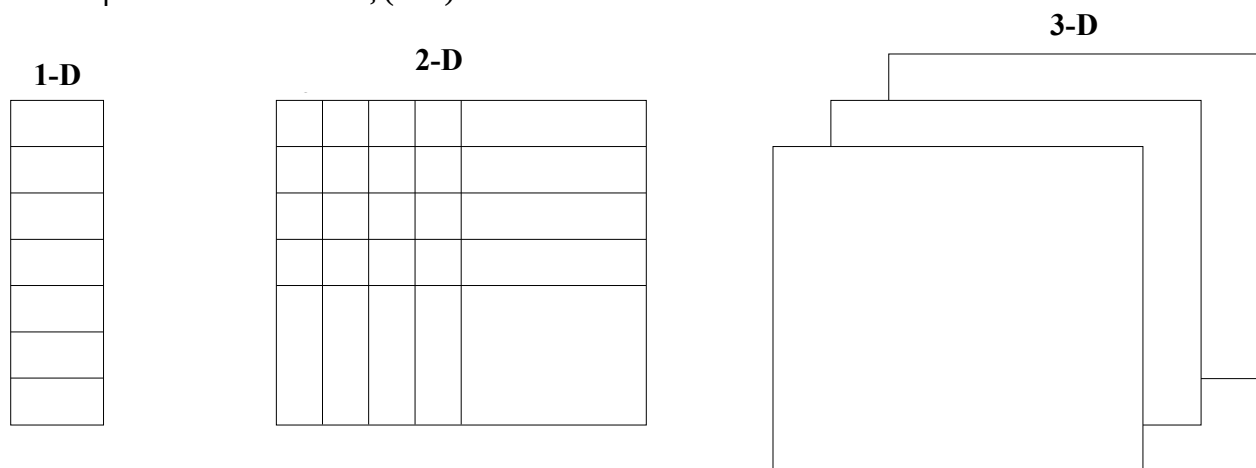
Σημειώσεις βασισμένες στο βιβλίο “Το MATLAB στην Υπολογιστική Επιστήμη και Τεχνολογία – Μια Εισαγωγή”

Πίνακες (Arrays) [1/2]

- Δομές δεδομένων για την αποθήκευση δεδομένων υπό ένα κοινό όνομα
- Η αποθήκευσή τους σε πίνακες βοηθάει κυρίως τη διαχείρισή τους

Διάκριση πινάκων:

- Μονοδιάστατοι πίνακες (1-D)
- Δισδιάστατοι πίνακες (2-D)
- Τρισδιάστατοι πίνακες (3-D) κτλ.



A) Μονοδιάστατοι πίνακες (Διανύσματα - Vectors):

Πρόκειται για διατεταγμένα στοιχεία ίδιου τύπου.
Έχουν **όνομα** και **μέγεθος** (πλήθος στοιχείων).

Π.χ.: Μέσες θερμοκρασίες ημέρας για μία εβδομάδα:

Δ	Τα	Τ	Π	Π	Σ	Κ
15	16.2	17.1	16	20.1	21.8	22

Αποθήκευση σε μονοδιάστατο πίνακα (διάνυσμα) T:

15	16.2	17.1	16	20.1	21.8	22
T(1)	T(2)	T(3)	T(4)	T(5)	T(6)	T(7)

- Οι θέσεις του διανύσματος είναι αριθμημένες (δείκτες)
- Οι δείκτες ξεκινούν από το 1.
- Το μέγεθος (μήκος ή πλήθος στοιχείων) δίνεται από την εντολή: `length(T)` (στην περίπτωσή μας `length(T) → 7`)

- **Δημιουργία διανυσμάτων:**

- 1) Άμεσα από τον προγραμματιστή:

α)

```
A = [10 20 30 40 50];
```

```
B = [3 2.5 4 0];
```

β) Με τον τελεστή : για πίνακες με ισαπέχουσες τιμές:

```
C = ΑρχικήΤιμή:Βήμα:ΤελικήΤιμή;
```

π.χ. C = 0:5:20; → C = [0 5 10 15 20];

D = 5:8; → D = [5 6 7 8];

- 2) Μέσω εσωτερικών συναρτήσεων του MATLAB:

x = rand(1,10) → διάνυσμα με 10 τυχαία στοιχεία

y = zeros(1,5) → διάνυσμα με 5 μηδενικά στοιχεία

z = ones(1,20) → διάνυσμα με 20 στοιχεία ίσα με 1

- **Πρόσβαση στα στοιχεία ενός διανύσματος:**

→ όνομα(δείκτης)

Ο δείκτης είναι πάντα **ακέραια** τιμή ή μεταβλητή ή αριθμητική έκφραση και δηλώνει τη θέση στο διάνυσμα (ξεκινώντας από το 1).

```
π.χ.: >> A = [2 5 3 0 1];
>> A(2) (Τυπώνει 5)
>> x = A(4);
>> disp(x); (Τυπώνει 0)
```

π.χ.: Εμφάνιση όλων των στοιχείων του A:

```
for i = 1:length(A)
    fprintf('Το %gο στοιχείο του A είναι το %g.',i,A(i));
end
```

Τυπώνει:

```
Το 1ο στοιχείο του A είναι το 2.
Το 2ο στοιχείο του A είναι το 5.
Το 3ο στοιχείο του A είναι το 3.
Το 4ο στοιχείο του A είναι το 0.
Το 5ο στοιχείο του A είναι το 1.
```

Είσοδος/Έξοδος διανυσμάτων:

- **Είσοδος (ανά στοιχείο):**

```
for i = 1:10
    message = sprintf('Δώσε το στοιχείο %g', i);
    A(i) = input(message);
end
```

Η συνάρτηση `sprintf` είναι ίδια με την `fprintf`, αλλά αποθηκεύει το αποτέλεσμα της σε μια συμβολοσειρά.

Μπορεί να χρησιμοποιηθεί για την εμφάνιση "μορφοποιημένου" κειμένου στην εντολή `input`, όπως παραπάνω.

- **Είσοδος ολόκληρου πίνακα:**

```
A = input('Δώσε τον πίνακα A');
```

Ο χρήστης δίνει: `>>[5 15 20 0 4]`

- **Έξοδος (ανά στοιχείο):**

```
for i = 1:10
    fprintf(A(%g) = %g\n', i, A(i));
    %ή πιο απλά: disp(A(i));
end
```

- **Έξοδος ολόκληρου πίνακα:** `disp(A)`;

Παραδείγματα διανυσμάτων:

- **Μέση τιμή θερμοκρασιών και θερμοκρασιακές διαφορές:**

```
T = input('Δώσε τον πίνακα θερμοκρασιών: ');
sum = 0;
for i = 1:length(T)
    sum = sum + T(i);
end
mT = sum/length(T);
fprintf('Η μέση θερμοκρασία είναι %g.\n', mT);
for i = 1:length(T)
    DT(i) = T(i) - mT;
end
disp('Θερμοκρασιακές διαφορές: ');
disp(DT);
```

- **Μέγιστο κι ελάχιστο διανύσματος:**

```
list = input('Δώσε το διάνυσμα: ');
small = list(1);
large = list(1);
for i = 2:length(list)
    if list(i)<small
        small = list(i);
    else
        if list(i)>large
            large = list(i);
        end
    end
end
end
```

Γιατί δεν χρησιμοποιήθηκαν
2 ανεξάρτητα if;

- **Εσωτερικό γινόμενο διανυσμάτων:**

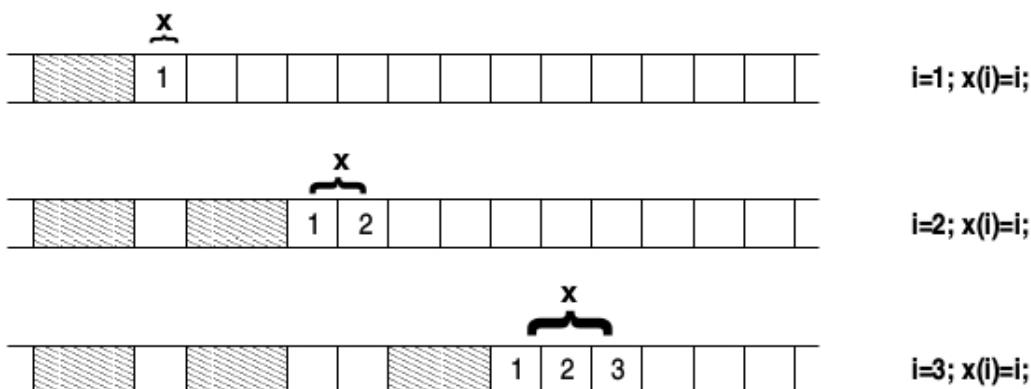
```
% έστω διανύσματα x, y ίδιου μήκους
p = 0;
for i = 1:length(x)
    p = p + x(i)*y(i);
end
```

- **Αρχικοποίηση διανυσμάτων:**

α)

```
for i = 1:100
    x(i) = i;
    %εντολές που δεσμεύουν μνήμη
end
```

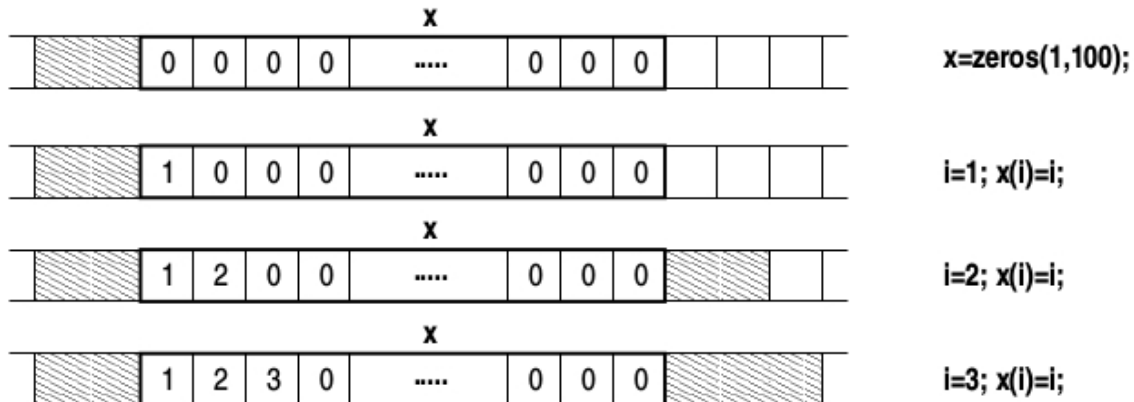
Εικόνα της μνήμης:



β) (με δημιουργία ολόκληρου του διανύσματος πριν την αρχικοποίηση)

```
x = zeros(1,100);
for i = 1:100
    x(i) = i;
    %εντολές που δεσμεύουν μνήμη
end
```

Εικόνα της μνήμης:



● **Ταξινόμηση διανύσματος με τον αλγόριθμο της επιλογής (selection sort):**

1. Ξεκίνα από την πρώτη θέση.
2. Βρες το ελάχιστο στοιχείο από τη θέση αυτή και μετά.
3. Ενάλλαξε (swap) τα στοιχεία της θέσης που ξεκίνησες και της θέσης του ελάχιστου.
4. Πήγαινε στην επόμενη θέση και επανάλαβε το “Βήμα 2”, μέχρι να φτάσεις στην προτελευταία θέση.

Υλοποίηση με συνάρτηση:

```
function v = selection(v)
for i = 1:length(v)-1
    minIndex = i; % έστω ότι το στοιχείο αυτό είναι το ελάχιστο
    %αναζήτηση του ελάχιστου:
    for j = i+1:length(v)
        if v(j)<v(minIndex)
            minIndex = j; % η θέση του ελάχιστου στοιχείου
        end
    end
    if minIndex ~= i; % αν βρέθηκε κάποιο μικρότερο στοιχείο
        temp = v(i);
        v(i) = v(minIndex);
        v(minIndex) = temp;
    end
end
end
```

- **Ταξινόμηση διανύσματος με τον αλγόριθμο εισαγωγής (insertion sort):**

1. Ξεκίνα από το δεύτερο στοιχείο.
2. Σύγκρινε το στοιχείο με ένα-ένα τα στοιχεία που βρίσκονται αριστερά του **μέχρι να βρεις** (άρα while-loop) κάποιο μικρότερό το κι όσο βρίσκεις κάποιο μεγαλύτερό του, **ελλάσσέ** τα.
3. Πήγαινε στο επόμενο στοιχείο και επανάλαβε το “Βήμα 2”.

Υλοποίηση με συνάρτηση:

```
function v = insertion(v)
for i = 2:length(v)
    j = i;
    while j>1 && v(j-1)>v(j)
        temp = v(i);
        v(j) = v(j-1);
        v(j-1) = temp;
        j = j - 1;
    end
end
end
```

- **Γραμμική αναζήτηση (linear search):**

→ Ξεκινώντας από το πρώτο στοιχείο, σύγκρινε ένα-ένα τα στοιχεία του διανύσματος με το προς αναζήτηση στοιχείο (**key**), μέχρι να το βρεις.

Υλοποίηση με συνάρτηση:

```
function pos = linearSearch(v, key)
loc = 1;      % η τρέχουσα θέση αναζήτησης
pos = -1;    % η θέση του στοιχείου αναζήτησης
             % Θα επιστρέψει -1 αν δε βρεθεί
hit = false; % flag για το αν έχει βρεθεί το προς αναζήτηση στοιχείο
             % (δεν είναι απαραίτητο)
while loc<=length(v) && ~hit
    if v(loc)==key
        pos = loc;
        hit = true;
    else
        loc = loc + 1;
    end
end
end
```


● Δυαδική αναζήτηση (binary search):

→ Προϋπόθεση: Η λίστα πρέπει να είναι ταξινομημένη!

1. Σύγκρινε το αναζητούμενο στοιχείο με το μεσαίο στοιχείο της λίστας
2. Όσο αυτά διαφέρουν και υπάρχουν ακόμα στοιχεία στη λίστα:
 - αν είναι μικρότερο του στοιχείου, επανάλαβε το “Βήμα 1” για το αριστερό τμήμα της λίστας.
 - αν είναι μεγαλύτερο του στοιχείου, επανάλαβε το “Βήμα 1” για το δεξί τμήμα της λίστας.

Υλοποίηση με συνάρτηση:

```
function pos = binarySearch(v, key)
left = 1; % το αριστερό άκρο της τρέχουσας λίστας
right = length(v); % το δεξί άκρο της τρέχουσας λίστας
pos = -1;
hit = false;
while left <= right && ~hit
    mid = fix((left+right)/2);
    if v(mid) == key
        pos = mid;
        hit = true;
    elseif key < v(mid)
        right = mid - 1;
    else
        left = mid + 1;
    end
end
```

● Το κόσκινο του Ερατοσθένη (3^{ος} αιώνας π.Χ.):

Αλγόριθμος για τη εύρεση των πρώτων αριθμών από το 2 έως το N.

1. Γράψε όλους τους αριθμούς από το 2 έως το N.
2. Κύκλωσε τον πρώτο διαθέσιμο αριθμό (θα είναι πρώτος).
3. Διέγραψε τα πολλαπλάσιά του.
4. Πήγαινε στο “Βήμα 2”.

π.χ.: για N=10 έχουμε:

2	3	4	5	6	7	8	9	10
Ⓜ	3	4	5	6	7	8	9	10
Ⓜ	Ⓜ	4	5	6	7	8	9	10
Ⓜ	Ⓜ	4	Ⓜ	6	7	8	9	10
Ⓜ	Ⓜ	4	Ⓜ	6	Ⓜ	8	9	10

Υλοποίηση με συνάρτηση:

```

function primes = eratosthenis(N)
P = 1:N % η λίστα των αριθμών
P(1) = 0; % Όσοι διαγράφονται θα γίνονται 0. Το 1 ΔΕΝ είναι πρώτος.
for i = 2:sqrt(N)
    if P(i) % αν το i δεν έχει διαγραφεί
        for j = 2*i:i:N % πήγαινε σε κάθε πολλαπλάσιο του i
            P(j) = 0; % και διέγραψε το
        end
    end
end
end
% Μέτρηση των πρώτων αριθμών που βρέθηκαν
NP = 0;
for i = 2:N
    if P(i)
        NP = NP + 1;
    end
end
primes = zeros(1,NP);
k = 0;
for i = 2:N
    if P(i)
        k = k + 1;
        primes(k) = i;
    end
end
end

```

- **Δημιουργία διανύσματος με τη συνάρτηση `linspace`**

`linspace(Αρχική_Τιμή, Τελική_Τιμή, πλήθος_στοιχείων)`

Π.χ., `x = linspace(0, 2*pi, 9);`

→ [0 0.7854 1.5708 2.3562 3.1416 3.9270 4.7124 5.4978 6.2832]

- **Πράξεις με διανύσματα:**

Έστω τα διανύσματα:

```

a = [10 8 -5]
b = [2 4 1]

```

```

c = 5*a → c: [50 40 -25]
c = a/2 → c: [5 4 -2.5]

```

```

c = -a      →      c: [-10 -8 5]
c = 1./a    →      c: [.100 .125 -.200]
c = a+5     →      c: [15 13 0]
c = a.^2    →      c: [100 64 25]
c = a+b     →      c: [12 12 -4]
c = a-b     →      c: [8 4 -6]
c = a.*b    →      c: [20 32 -5]
c = a./b    →      c: [5 2 -5]

```

Έστω το διάνυσμα $v = [2 \ 5 \ 7]$

```

v(4) = 8;      →      v: [2 5 7 8]
v(6) = 10;     →      v: [2 5 7 8 0 10]
v(2:4)         →      [5 7 8]
v([4 6])       →      [8 10]
[v(1:2) v([4 6])] → [2 5 8 10]

```

● Διανύσματα και γραφικές παραστάσεις:

Να γραφεί πρόγραμμα που να εμφανίζει τη γραφική παράσταση της συνάρτησης

$$f(x) = \frac{\eta\mu(5x) \cdot e^{-\frac{x}{2}}}{1+x^2} \text{ στο διάστημα } [-2, 3].$$

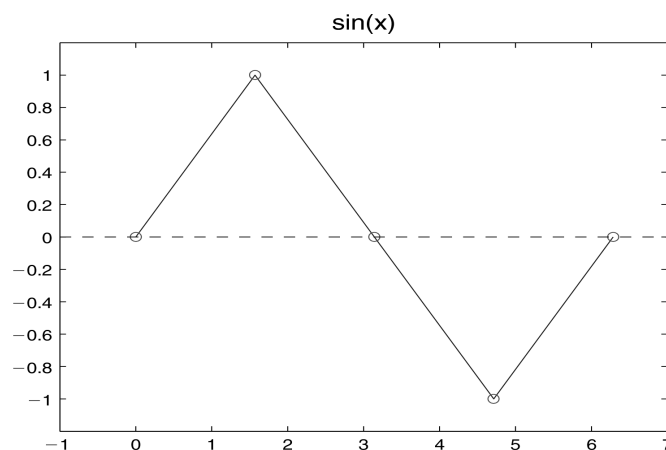
→ Ένα πιο απλό πρόβλημα: σχεδίαση του $\eta\mu(x)$ στο $[0, 2\pi]$.

“Με το χέρι”, θα παράγαμε έναν πίνακα τιμών, π.χ.:

x	0.000	1.571	3.142	4.712	6.283
$\sin(x)$	0.000	1.000	0.000	-1.000	0.000

Στη συνέχεια θα συνδέαμε τα αντίστοιχα 5 σημεία:

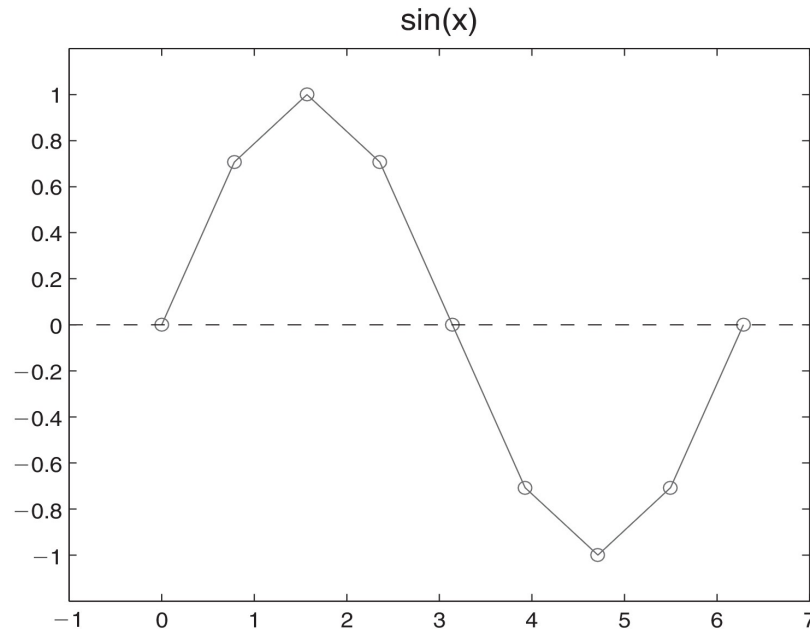
$(0, 0)$, $(1.571, 1)$, $(3.142, 0)$, $(4.712, -1)$, $(6.283, 0)$



Αν δημιουργήσουμε 9 δειγματοληπτικά σημεία αντί για 5:

x	0.000	0.785	1.571	2.356	3.142	3.927	4.712	5.498	6.283
$\sin(x)$	0.000	0.707	1.000	0.707	0.000	-0.707	-1.000	-0.707	-0.000

θα προκύψει το εξής:



Για τη γραφική αποτύπωση μιας συνάρτησης $y = f(x)$ σε ένα διάστημα $[L, R]$:

1. Δημιουργία πίνακα με τις τιμές του x από το δεδομένο διάστημα.
2. Δημιουργία πίνακα με τις τιμές του y που αντιστοιχούν (μέσω της f) στις τιμές του x .
3. Συνένωση των σημείων που ορίζονται από τα ζεύγη xy και εμφάνιση της γραμμής που προκύπτει.

```
x = linspace(0, 2*pi, 9);
y = sin(x);
plot(x, y);
```

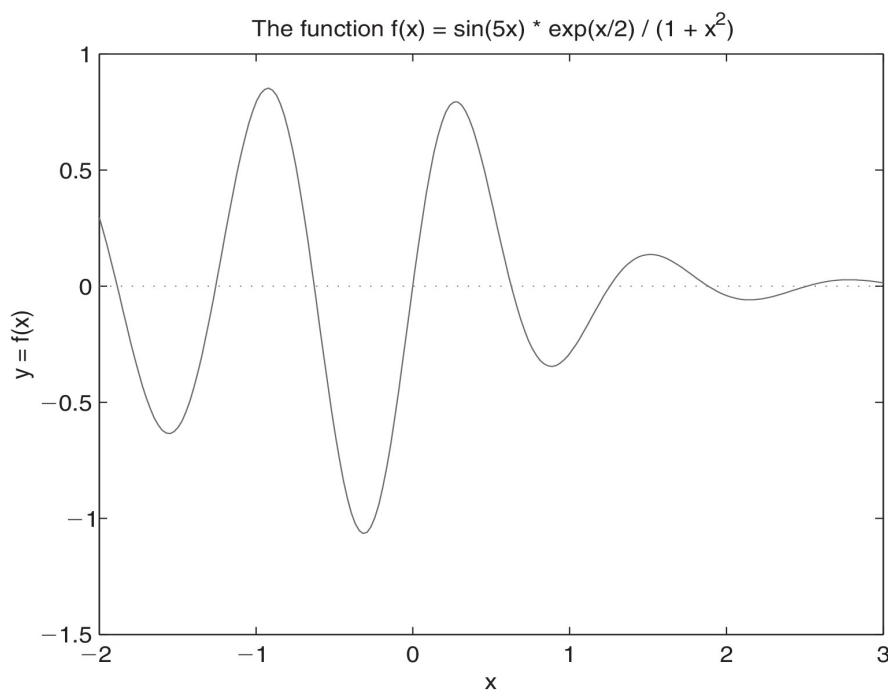
Π.χ., το ακόλουθο script σχεδιάζει το $\sin(x)$ για διάφορα πλήθη δειγματοληπτικών σημείων:

```
for n = 25:25:500
    x = linspace(0, 2*pi, n);
    y = sin(x);
    plot(x, y);
    title(sprintf('n = %3d', n));
    pause;
end
```

→ Script για τη σχεδίαση της αρχικής συνάρτησης:

```
x = linspace(2,3,100);
y1 = 5*x;
y2 = sin(y1);
y3 = -x;
y4 = y3/2;
y5 = exp(y4);
y6 = y2.*y5;
y7 = x.^2;
y8 = 1+y7;
y = y6./y8;
plot(x,y);
```

} $y = \sin(5x) \cdot \exp(-x/2) / (1+x^2);$



Πολλαπλά γραφήματα: `plot(x, y1, x, y2, x, y3, ...)`

Η συνάρτηση `plot` με παραμέτρους μορφοποίησης:

`plot(x, y, 'XYZ')`

- X: τύπος γραμμής
- Y: τύπος σημείων
- Z: χρώμα σημείων και γραμμής

Επιλογές:

X: - : -. --

Y: . o + x *

Z: b (μπλε) w (άσπρο) c (γαλάζιο) m (μωβ) r (κόκκινο) y (κίτρινο) g (πράσινο) k (μαύρο)

π.χ., `plot(x, y, '-or')`; → κόκκινη συνεχής γραμμή με σημεία-κύκλους

Βασικές εντολές γραφήματος:

Τίτλος γραφήματος: `title('.....');`

Τίτλος άξονα x: `xlabel('.....');`

Τίτλος άξονα y: `ylabel('.....');`

Σημειώματα

Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Μιχάλης Δρακόπουλος, 2014.
Μιχάλης Δρακόπουλος. «Πληροφορική Ι. Ενότητα 5: Πίνακες [1/2] (Διανύσματα)». Έκδοση: 1.0.
Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://opencourses.uoa.gr/modules/document/?course=MATH105>.

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

