



4 . 0 0 4

Σήματα και Συστήματα

Εβδομάδα 1 — Εισαγωγή στο GNU Octave

Γιάννης Στεφανής (ν2025-2026 με βάση τις σημειώσεις των προηγούμενων ετών)

Τμήμα Ηλεκτρολόγων Μηχανικών — ΕΛΜΕΠΑ

Εργαστήριο Σημάτων και Συστημάτων • Εαρινό Εξάμηνο 2025-26

Τι είναι η Επεξεργασία Σημάτων;



Ήχος & Μουσική

Φίλτρα, EQ,
επεξεργασία



Τηλεπικοινωνίες

WiFi, 5G, δορυφόροι



Ιατρική

ECG, MRI, υπέρηχοι



Εικόνα & Video

Συμπίεση, φίλτρα, AI



Αυτοματισμός

Αισθητήρες, PID,
έλεγχος



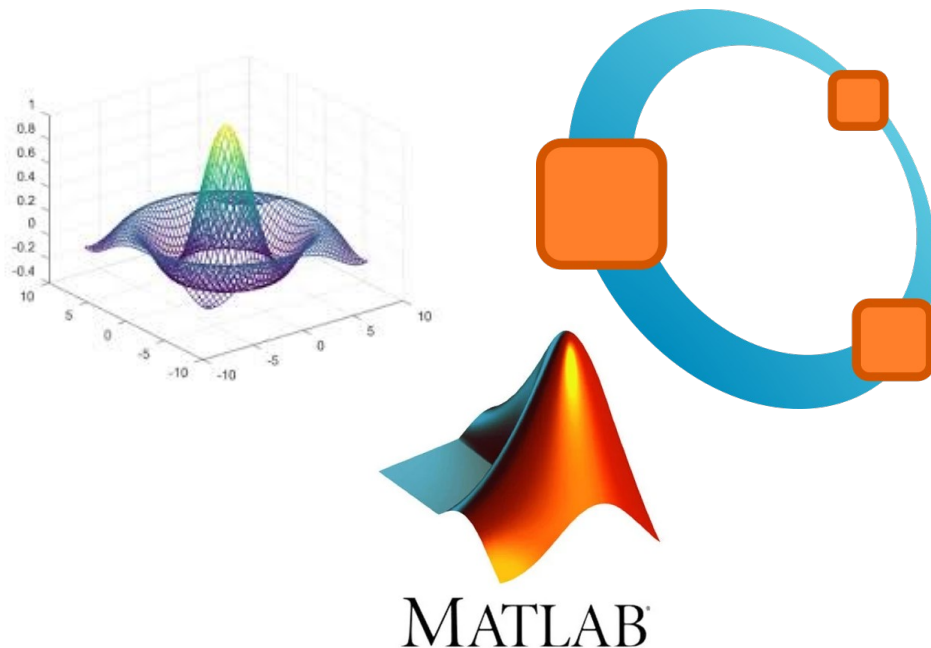
Ανάλυση Δεδομένων

Χρηματιστήριο, IoT,
radar



Σχεδόν κάθε ηλεκτρονική συσκευή που έχετε βασίζεται στην επεξεργασία σημάτων!

Γιατί GNU Octave;



- ✓ **Δωρεάν & Open Source**
Χωρίς κόστος άδειας χρήσης
- ✓ **MATLAB-Συμβατό**
Ίδια σύνταξη, εύκολη μετάβαση
- ✓ **Cross-Platform**
Windows, macOS, Linux
- ✓ **Πακέτα Signal/Control**
Εξειδικευμένα εργαλεία σημάτων



Η γνώση Octave μεταφέρεται απευθείας σε MATLAB — είναι συμβατά κατά 95%!

Τι θα μάθουμε σήμερα



Περιβάλλον
Octave

10'



Μεταβλητές
& Πράξεις

15'



Διανύσματα
& Πίνακες

20'



Γραφικές
Παραστάσεις

25'



Scripts
& Βοήθεια

10'

Συνολικός χρόνος: ~80 λεπτά

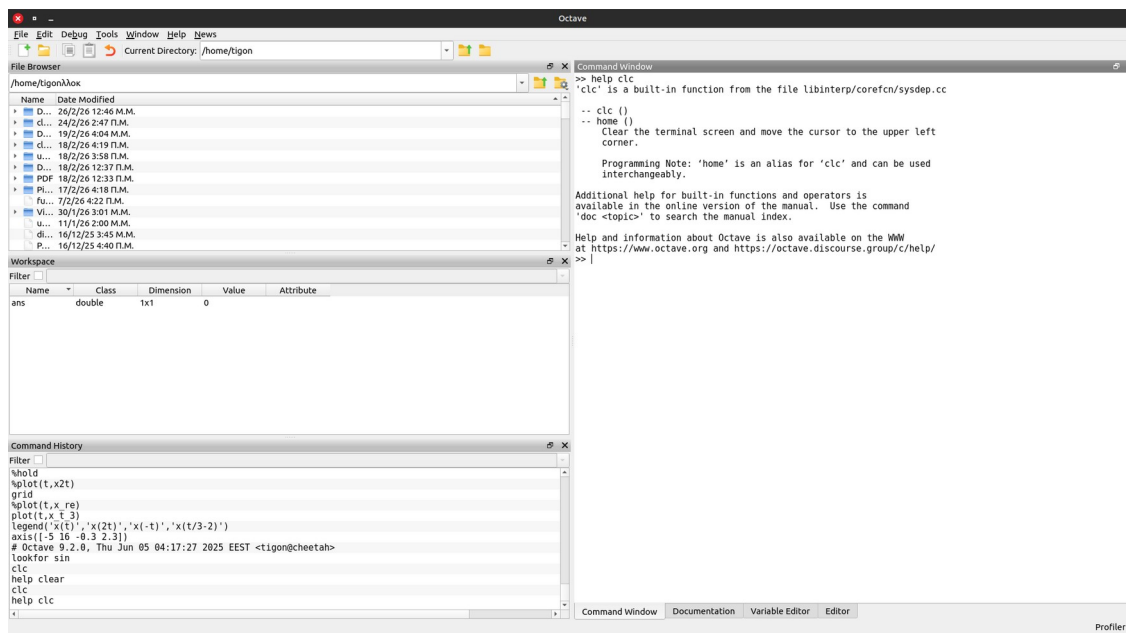
Στόχοι Μάθησης

Μετά το σημερινό εργαστήριο, θα μπορείτε να:

- ✓ Πλοηγηθείτε στο περιβάλλον του GNU Octave
- ✓ Δημιουργήσετε μεταβλητές, διανύσματα και πίνακες
- ✓ Εκτελέσετε βασικές μαθηματικές πράξεις
- ✓ Κατασκευάσετε γραφήματα με `plot()` και `stem()`
- ✓ Γράψετε και εκτελέσετε scripts (.m αρχεία)
- ✓ Χρησιμοποιήσετε τη βοήθεια (`help`, `doc`, `lookfor`)



Το Περιβάλλον του GNU Octave



Menu & Toolbar

Command Window

Workspace

Status Bar



Command Window — Πρώτες Εντολές

```
>> 2 + 3  
ans = 5
```

```
>> sin(pi/2)  
ans = 1
```

```
>> sqrt(144)  
ans = 12
```

```
>> format long  
>> pi  
ans = 3.14159265358979
```

>>

Prompt — εδώ γράφουμε εντολές

ans

Αυτόματη μεταβλητή αποτελέσματος

sin(), cos()

Τριγωνομετρικές (σε ακτίνια)

sqrt(), abs()

Ρίζα, απόλυτη τιμή

pi, e, i, j

Ενσωματωμένες σταθερές

format long

Περισσότερα δεκαδικά ψηφία



Τελειώστε μια εντολή με ; για να μη δείξει αποτέλεσμα



Μεταβλητές και Τύποι Δεδομένων

```
>> x = 5  
x = 5
```

```
>> name = 'Σήματα'  
name = Σήματα
```

```
>> z = 3 + 4i  
z = 3 + 4i
```

```
>> flag = true  
flag = 1
```

double Πραγματικοί αριθμοί (default)
x = 3.14

complex Μιγαδικοί αριθμοί
z = 3 + 4i

char Κείμενο (strings)
s = 'hello'

logical Λογικές τιμές (0/1)
b = true

Κανόνες ονομασίας: Ξεκινά με γράμμα • Case-sensitive (x ≠ X) • Χωρίς κενά ή ειδικούς χαρακτήρες



Χρησιμοποιήστε *whos* για να δείτε όλες τις μεταβλητές στο *Workspace*



Αριθμητικές Πράξεις

Τελεστής	Περιγραφή
+ - * /	Πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση
^	Δύναμη ($2^3 = 8$)
.* ./ .^	Στοιχείο-προς-στοιχείο (element-wise)
'	Ανάστροφος πίνακα (transpose)
\	Επίλυση γραμμικού συστήματος ($A \setminus b$)

```
>> a = 10; b = 3;
```

```
>> a + b
```

```
ans = 13
```

```
>> a^2
```

```
ans = 100
```

```
>> mod(a, b)
```

```
ans = 1
```

```
>> round(3.7)
```

```
ans = 4
```



Ο τελεστής .* είναι κρίσιμος — τον χρησιμοποιούμε παντού στα σήματα!



Διανύσματα — Δημιουργία

① Απευθείας ορισμός

```
>> v = [1 3 5 7 9]
v = 1 3 5 7 9
```

② Τελεστής : (colon)

```
>> v = 1:2:9
v = 1 3 5 7 9

>> t = 0:0.1:1
t = 0 0.1 0.2 ... 1.0
```

③ linspace(start, stop, N)

```
>> t = linspace(0, 1, 100);
>> length(t)
ans = 100
```

④ Ειδικά διανύσματα

```
>> zeros(1, 5)
ans = 0 0 0 0 0

>> ones(1, 3)
ans = 1 1 1
```

```
>> rand(1, 4)
```



linspace είναι η πιο συχνή επιλογή για δημιουργία σημάτων — ελέγχουμε ακριβώς πόσα σημεία θέλουμε!



Πράξεις σε Διανύσματα

Element-wise (στοιχείο-προς-στοιχείο)

```
>> x = [1 2 3 4];  
>> y = [10 20 30 40];  
  
>> x + y  
ans = 11 22 33 44  
  
>> x .* y  
ans = 10 40 90 160  
  
>> x .^ 2  
ans = 1 4 9 16
```

Χρήσιμες συναρτήσεις

```
>> length(x)  
ans = 4  
  
>> sum(x)  
ans = 10  
  
>> max(x)  
ans = 4  
  
>> sort([3 1 4 1 5])  
ans = 1 1 3 4 5
```

Παράδειγμα σημάτων: `t = linspace(0, 1, 1000); x = sin(2*pi*5*t);` — δημιουργεί ημιτονοειδές 5 Hz!

Πίνακες (Matrices) – Δημιουργία & Ευρετηρίαση

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

i=1	1	2	3
i=2	4	5	6
i=3	7	8	9
	j=1	j=2	j=3

A(2,3)=6

A(i,j)	στοιχείο γραμμή i, στήλη j
A(1,:)	πρώτη γραμμή (όλες στήλες)
A(:,2)	δεύτερη στήλη (όλες γραμμές)
A(2:3,1:2)	υποπίνακας 2x2
size(A)	→ [3 3]
A'	ανάστροφος (transpose)
zeros(m,n)	μηδενικός m×n πίνακας
eye(n)	μοναδιαίος n×n

Ευρετηρίαση & Εντολές

⚠ **1-based: A(1,1) είναι πάνω-αριστερά — ΟΧΙ A(0,0)**

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

```
% Ευρετηρίαση (1-based)
A(2,3) % → 6
A(1,:) % → [1 2 3]
A(:,2) % → [2;5;8]
```

```
% Μέγεθος
size(A) % → [3 3]
[r,c] = size(A)
```

```
% Ειδικοί
zeros(3); ones(2,4)
eye(3); rand(3)
```

- 1-based: A(1,1) = πάνω-αριστερά
- ΟΧΙ A(0,0) — αυτό είναι Python/C!
- A(i,:) = γραμμή | A(:,j) = στήλη
- A(:) = όλα ως στήλη-διάνυσμα



Πράξεις Πινάκων

Πράξεις πινάκων (matrix)

```
>> A = [1 2; 3 4];  
>> B = [5 6; 7 8];  
  
>> A * B           % πολλαπλασιασμός  
ans =  
    19    22  
    43    50  
  
>> A'              % ανάστροφος  
ans =  
     1     3  
     2     4  
  
>> det(A)          % ορίζουσα  
ans = -2
```

Επίλυση γραμμικού συστήματος

$$Ax = b \rightarrow x = A \setminus b$$

```
>> A = [2 1; 5 3];  
>> b = [4; 7];  
>> x = A \ b  
x =  
     5  
    -6
```

⚠ Προσοχή!

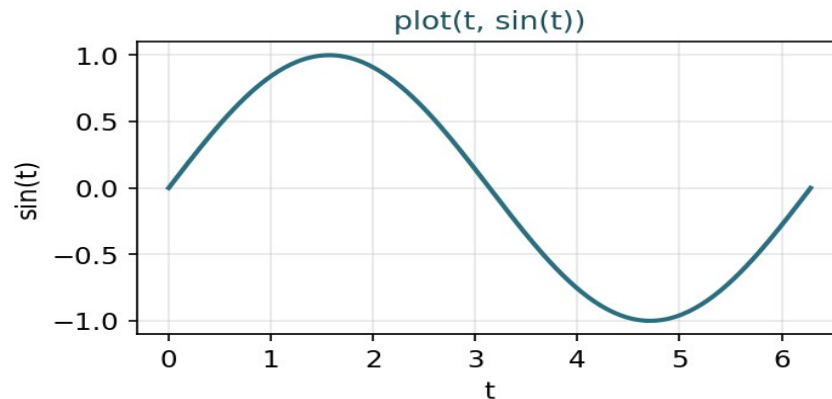
$A * B \rightarrow$ πολλαπλασιασμός πινάκων

$A .* B \rightarrow$ στοιχείο-προς-στοιχείο



Γραφική παράσταση συνεχούς σήματος – plot()

```
>> t = linspace(0, 2*pi, 200);  
>> y = sin(t);  
>> plot(t, y)  
>> title('Ημίτονο')  
>> xlabel('t (rad)')  
>> ylabel('sin(t)')  
>> grid on
```



Επιλογές μορφοποίησης:

'r' Κόκκινη γραμμή
'b--' Μπλε διακεκομμένη
'g*' Πράσινα αστέρια

'ko-' Μαύρο, κύκλοι + γραμμή
'LineWidth', 2 Πάχος γραμμής



Πολλαπλά Γραφήματα — hold on

```
>> t = linspace(0, 2*pi, 200);
>> plot(t, sin(t), 'b', 'LineWidth', 2)
>> hold on
>> plot(t, cos(t), 'r--', 'LineWidth', 2)
>> plot(t, sin(2*t), 'g', 'LineWidth', 1.5)
>> hold off
>> legend('sin(t)', 'cos(t)', 'sin(2t)')
>> title('Πολλαπλά σήματα')
>> grid on
```

hold on / off

Κρατά ή καθαρίζει το γράφημα

legend()

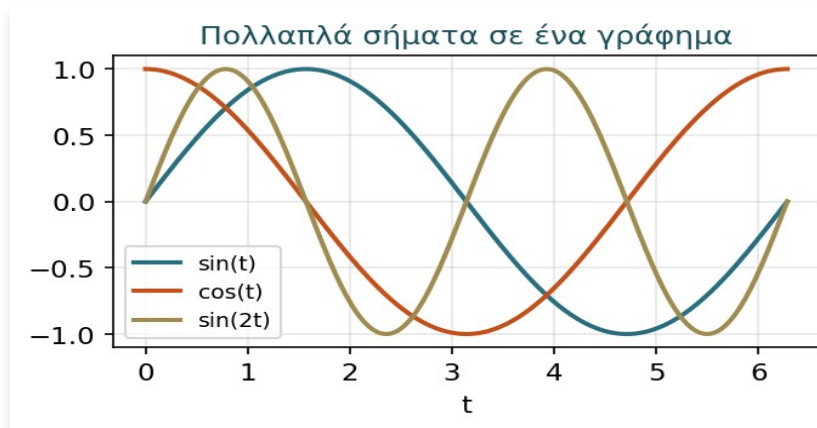
Υπόμνημα για κάθε σήμα

title(), xlabel()

Τίτλος και ετικέτες αξόνων

grid on

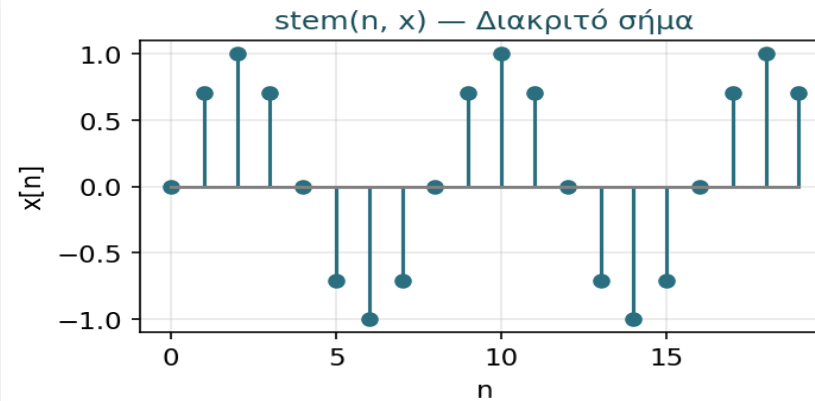
Εμφανίζει πλέγμα





stem() — Διακριτά Σήματα

```
>> n = 0:19;  
>> x = sin(2*pi*n/8);  
>> stem(n, x)  
>> title('Διακριτό σήμα x[n]')  
>> xlabel('n')  
>> ylabel('x[n]')
```



plot() vs stem() — Πότε χρησιμοποιούμε τι;

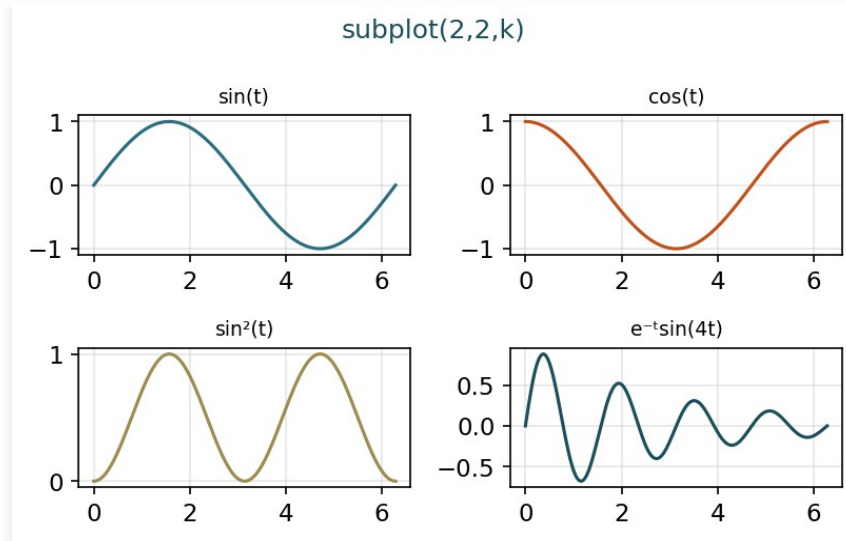
plot(t, x) → Συνεχή σήματα (continuous-time). Συνδέει σημεία με γραμμές.

stem(n, x) → Διακριτά σήματα (discrete-time). Κάθετες γραμμές σε ακέραια n.



subplot() — Πολλαπλά Γραφήματα

```
>> t = linspace(0, 2*pi, 200);  
  
>> subplot(2,2,1)  
>> plot(t, sin(t)); title('sin(t)')  
  
>> subplot(2,2,2)  
>> plot(t, cos(t)); title('cos(t)')  
  
>> subplot(2,2,3)  
>> plot(t, sin(t).^2); title('sin^2(t)')  
  
>> subplot(2,2,4)  
>> plot(t, exp(-t/3).*sin(4*t))  
>> title('e^{-t}sin(4t)')
```



subplot(rows, cols, index) — χωρίζει το figure σε πλέγμα γραφημάτων



Scripts — Αρχεία .m

my_signal.m

```
% my_signal.m – Πρώτο script!  
% Εβδομάδα 1, Εργαστήριο ΣΚΣ  
  
clc; clear; close all;  
  
% Παράμετροι  
f = 5;           % Συχνότητα (Hz)  
A = 2;          % Πλάτος  
fs = 1000;      % Δειγματοληψία  
T = 1;          % Διάρκεια (sec)  
  
% Δημιουργία σήματος  
t = linspace(0, T, fs*T);  
x = A * sin(2*pi*f*t);  
  
% Γράφημα  
plot(t, x, 'b', 'LineWidth', 1.5)  
title('Ημίτονο 5 Hz')  
xlabel('Χρόνος (s)')  
ylabel('x(t)')  
grid on
```

Εκτέλεση:

- 1 Αποθηκεύουμε ως my_signal.m
- 2 Πλοηγούμαστε στον φάκελο (cd)
- 3 Γράφουμε: >> my_signal
- 4 Ή πατάμε ► Run στον editor

Καλές πρακτικές:

- clc; clear; close all; στην αρχή
- Σχόλια με %
- Περιγραφικά ονόματα μεταβλητών



Βοήθεια — help, doc, lookfor

help <εντολή>

Σύντομη βοήθεια στο Command Window

```
>> help sin
```

```
sin is a built-in function  
Y = sin(X)  
Compute the sine for each element of X.
```

doc <εντολή>

Αναλυτική τεκμηρίωση σε ξεχωριστό παράθυρο

```
>> doc plot
```

```
(Ανοίγει παράθυρο Documentation)
```

lookfor <λέξη>

Αναζήτηση εντολής βάσει λέξης-κλειδιού

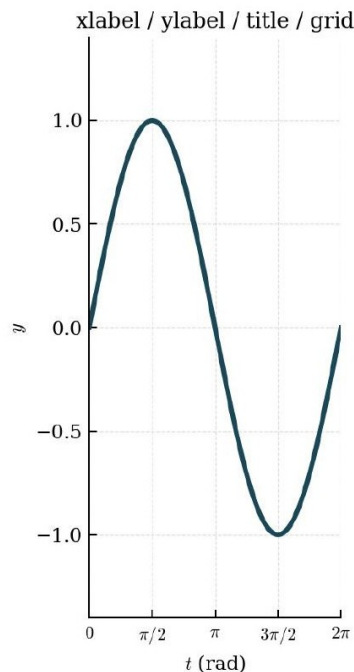
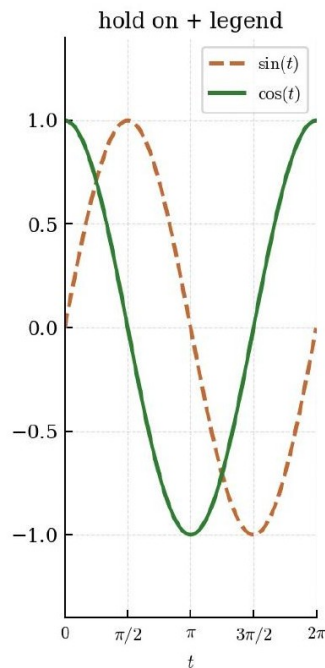
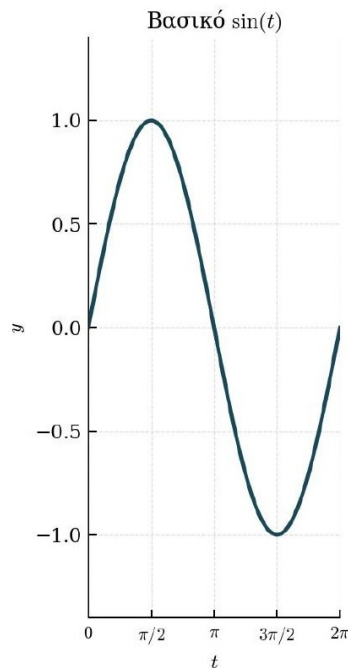
```
>> lookfor 'frequency'
```

```
fft      Discrete Fourier transform  
ifft     Inverse discrete Fourier  
freqz    Frequency response...
```



Αν δεν θυμάστε το όνομα μιας εντολής, χρησιμοποιήστε lookfor!

Μορφοποίηση Γραφήματος — xlabel, ylabel, title, grid



```
t = linspace(0, 2*pi, 200);
```

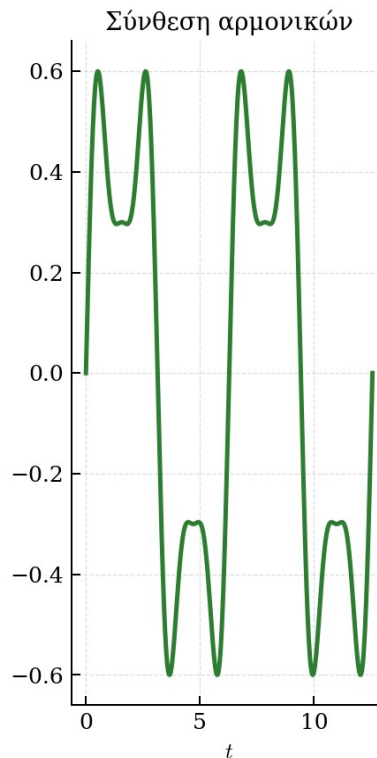
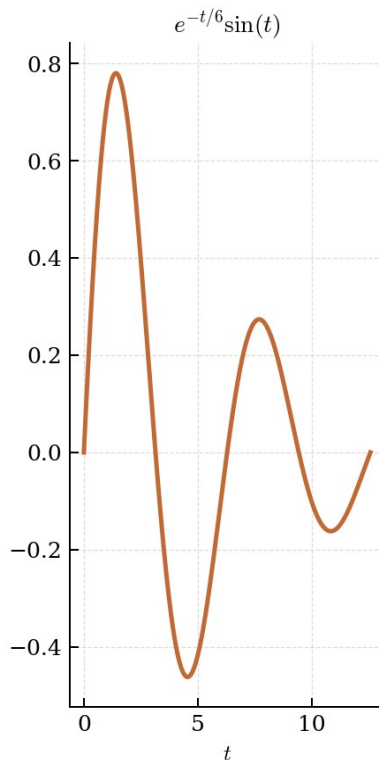
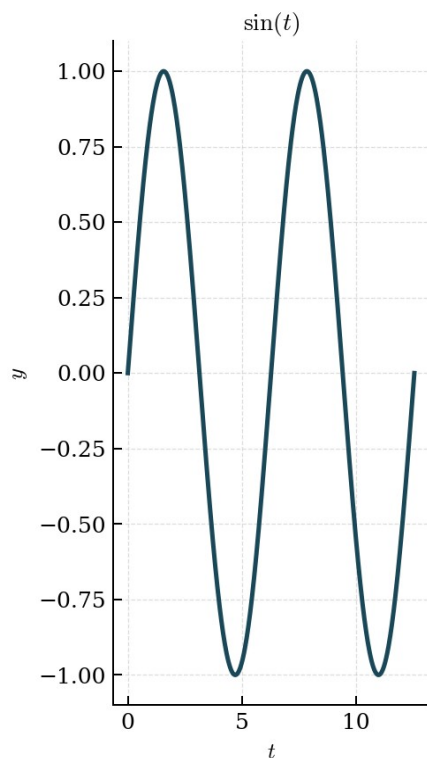
```
% --- Panel 1: Βασικό sin(t) ---
plot(t, sin(t), 'b', 'LineWidth', 2)
title('Βασικό sin(t)')
xlabel('t'); ylabel('y')
grid on
```

```
% --- Panel 2: hold on + legend ---
plot(t, sin(t), '--', 'Color', [0.76 0.42 0.21], 'LineWidth', 2)
hold on
plot(t, cos(t), 'Color', [0.18 0.49 0.20], 'LineWidth', 2)
legend('sin(t)', 'cos(t)')
title('hold on + legend')
xlabel('t')
grid on
```

```
% --- Panel 3: xlabel / ylabel / title / grid ---
plot(t, sin(t), 'b', 'LineWidth', 2.5)
title('Ημίτονο sin(t)')
xlabel('t (rad)')
ylabel('y')
grid on
xlim([0 2*pi])
ylim([-1.3 1.3])
```

- title/xlabel/ylabel: ετικέτες
- legend: υπόμνημα καμπύλης
- grid on/off: πλέγμα
- xlim/ylim: όρια αξόνων
- 'LineWidth', 2: πάχος γραμμής

Διανυσματοποίηση — Element-wise vs Matrix



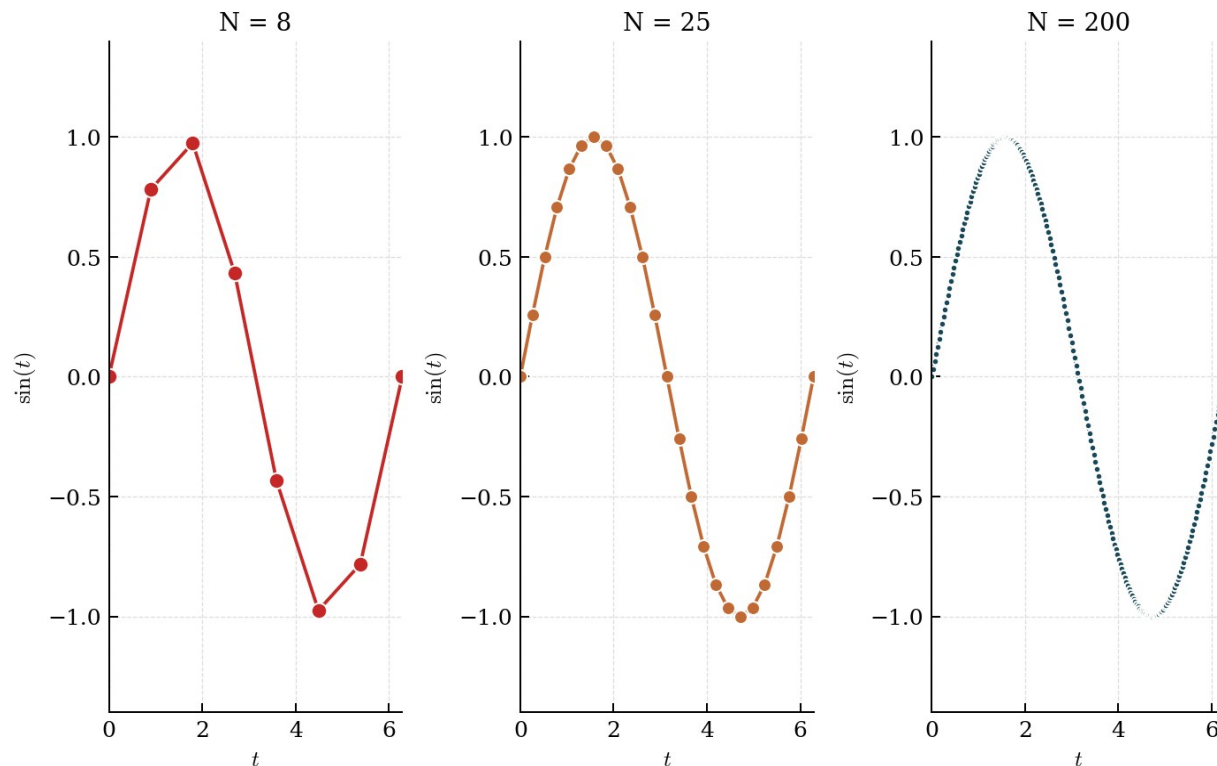
```
t = linspace(0, 4*pi, 500);
% Element-wise (τελεστές με .)
y1 = sin(t); % αυτόματα element-wise
y2 = sin(t) .* exp(-t/6); % .* σημαντικό!

% Σύνθεση αρμονικών
y3 = 0.5*sin(t) + ...
     0.3*sin(3*t) + ...
     0.1*sin(5*t);

% ΛΑΘΟΣ: sin(t) * exp(-t) % matrix mult!
% ΣΩΣΤΟ: sin(t) .* exp(-t)
```

- `./ .^` : element-wise πράξεις
- `* / ^` : πράξεις πινάκων
- Διανύσματα: ΠΑΝΤΑ element-wise
- Σφάλμα 'nonconformant': λάθος διαστάσεις
- `sum()`, `mean()`, `max()`: αριθμητική στατιστική

linspace vs Δίκτυο Δειγματοληψίας



```
% linspace: N σημεία από a ως b
t = linspace(0, 2*pi, 200);
% t(1)=0, t(200)=2π, Δt=(2π)/199
```

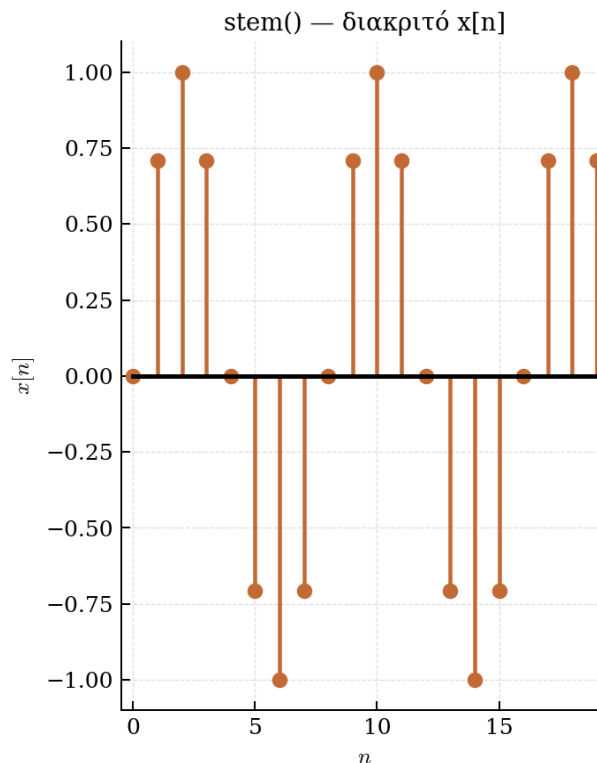
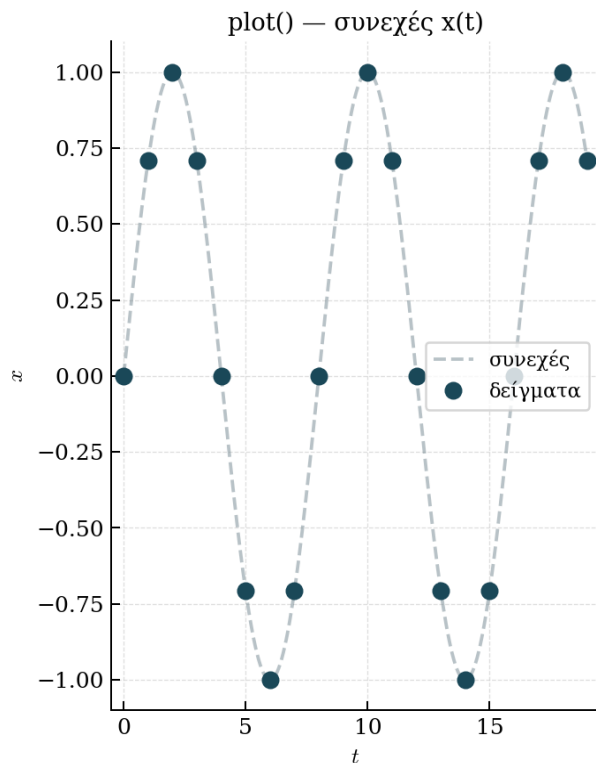
```
% Δίκτυο με βήμα dt
dt = 0.01;
t2 = 0 : dt : 2*pi;
% t2(1)=0, βήμα=0.01
```

```
% Διακριτός χρόνος (δείκτης n)
n = 0 : 19; % n = 0, 1, ..., 19
```

```
% Μετατροπή: Fs=δειγματ./sec
Fs = 1000; % 1000 Hz
t3 = (0:N-1) / Fs;
```

- `linspace(a,b,N)`: N σημεία, ακριβές b
- `a:dt:b`: βήμα dt, b \approx τελευταίο
- Μεγάλο N \rightarrow ομαλότερη καμπύλη
- Διακριτός: $n=0:N-1$ (ακέραιος δείκτης)
- $fs = 1/dt$: συχνότητα δειγματοληψίας

stem() vs plot() — Συνεχές και Διακριτό



```
% Συνεχές σήμα — plot()
t = linspace(0, 19, 200);
plot(t, sin(2*pi*t/8), 'LineWidth', 2)
xlabel('t'); title('Συνεχές')
```

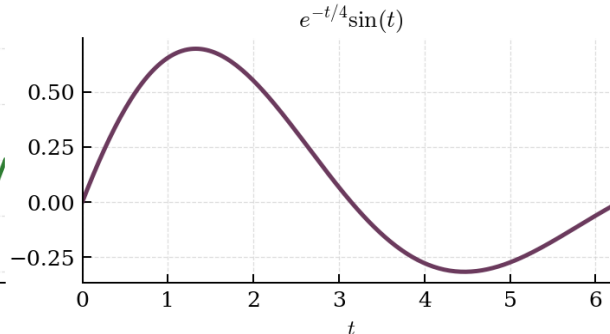
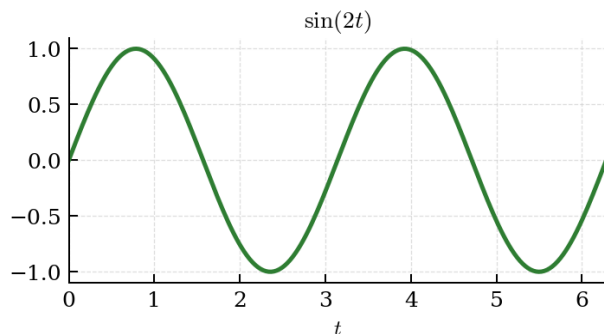
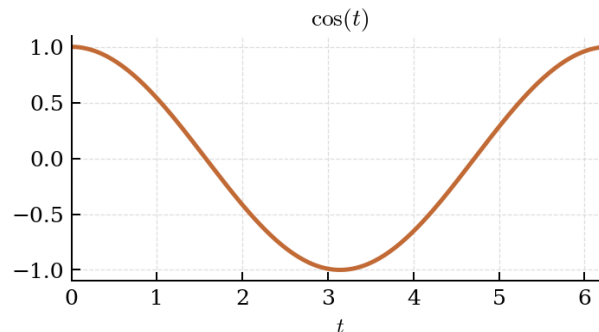
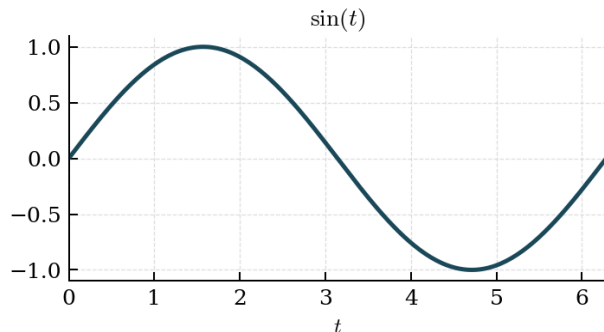
```
% Διακριτό σήμα — stem()
n = 0:19;
x = sin(2*pi*n/8);
stem(n, x)
xlabel('n'); title('Διακριτό')
```

```
% subplot για σύγκριση
subplot(1,2,1); plot(t, sin(2*pi*t/8))
subplot(1,2,2); stem(n, x)
```

- plot(): συνεχές $x(t)$ — γραμμή
- stem(): διακριτό $x[n]$ — ράβδοι
- Ποτέ stem() για συνεχές σήμα
- Ποτέ plot() για διακριτό (παρπλανητικό)
- Το Octave χρησιμοποιεί [] για διακριτό

subplot() — Οργάνωση Γραφημάτων

subplot(2,2,n)



```
t = linspace(0, 2*pi, 200);
```

```
% Πλέγμα 2x2
subplot(2,2,1)
plot(t, sin(t)); title('sin(t)')
```

```
subplot(2,2,2)
plot(t, cos(t)); title('cos(t)')
```

```
subplot(2,2,3)
plot(t, sin(2*t)); title('sin(2t)')
```

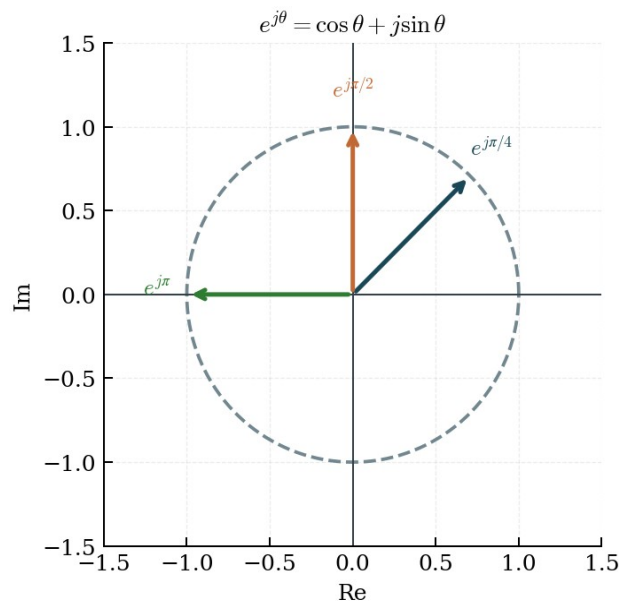
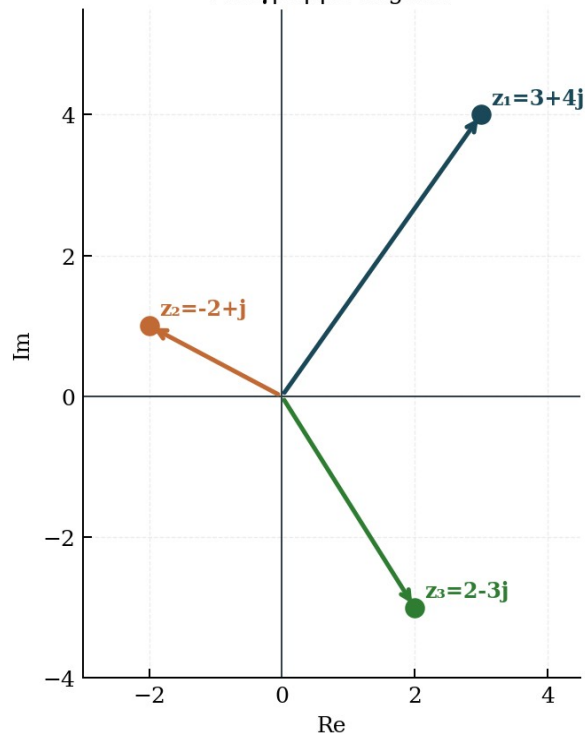
```
subplot(2,2,4)
plot(t, sin(t).*exp(-t/4))
title('αποσβ. ταλάντωση')
```

```
% subplot(rows, cols, index)
```

- subplot(r,c,n): r×c πλέγμα, n-οστό
- n=1: πάνω αριστερά, by row
- figure: νέο παράθυρο γραφήματος
- clf: καθαρισμός τρέχοντος figure
- close all: κλείσιμο όλων

Μιγαδικοί Αριθμοί στο Octave – Βάση για Fourier

Διάγραμμα Argand



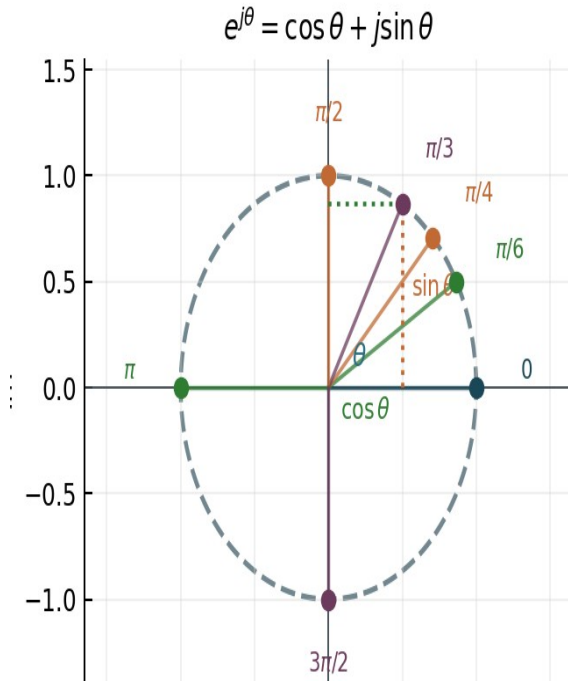
```
% Μιγαδικοί αριθμοί
z1 = 3 + 4i;
z2 = 2*exp(1j*pi/4); % πολική
```

```
% Στοιχειώδεις πράξεις
real(z1) % → 3
imag(z1) % → 4
abs(z1) % → 5 (μέτρο)
angle(z1) % → 0.9273 rad
conj(z1) % → 3-4i
```

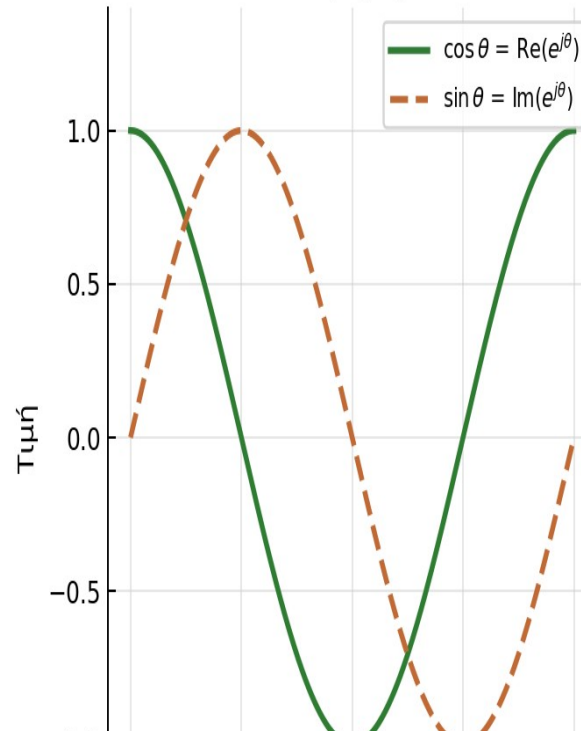
```
% Τύπος Euler
exp(1j*pi) % → -1 (!)
```

- $z = a + jb$ | $|z| = \sqrt{a^2+b^2}$
- $z = |z| \cdot e^{j\theta}$, $\theta = \text{angle}(z)$
- Τύπος Euler: $e^{j\theta} = \cos \theta + j \cdot \sin \theta$
- Χρήση j ή i (αν δεν είναι μεταβλητή)
- Fourier: σήματα ως $e^{j\omega t}$ (Εβδ. 7)
- Φάσμα: $|X(f)|$ πλάτος, $\text{angle}(X)$ φάση

Μοναδιαίος Κύκλος — Βασικές Γωνίες & Περιστροφή



Re και Im μέρος του $e^{j\theta}$



```
% Μοναδιαίος κύκλος: |e^(jθ)| = 1
theta = linspace(0, 2*pi, 100);
z = exp(1j*theta);
plot(real(z), imag(z))
axis equal; grid on
```

```
% Βασικές γωνίες
exp(1j*0) % → 1 (0°)
exp(1j*pi/2) % → j (90°)
exp(1j*pi) % → -1 (180°)
exp(1j*3*pi/2) % → -j (270°)
exp(1j*2*pi) % → 1 (360°)
```

```
% Σύνδεση με sin/cos
real(exp(1j*pi/3)) % = cos(pi/3) = 0.5
imag(exp(1j*pi/3)) % = sin(pi/3) = 0.866
```

- $|e^{j\theta}| = 1$ για κάθε θ : ΠΑΝΤΑ στον κύκλο
- $\text{Re}(e^{j\theta}) = \cos \theta$, $\text{Im}(e^{j\theta}) = \sin \theta$
- $e^{j\pi} = -1$ (ταυτότητα Euler)
- $e^{j2\pi} = 1$ (μια πλήρης περιστροφή)
- Fourier: $x(t) = \sum c_n \cdot e^{j\omega_0 t}$ (Εβδ.7)
- $|c_n|$ = πλάτος, $\angle c_n$ = φάση αρμονικής

Element-wise vs Matrix — .* ./ .^ vs * / ^

Πίνακες εισόδου

1	2	3	0
3	4	?	3

$$A = [1 \ 2; \ 3 \ 4]$$

$$B = [2 \ 0; \ 0 \ 3]$$

A .* B (element-wise)

2	0
0	12

κάθε (i,j) × (i,j)

$$1 \cdot 2 = 2 \qquad 2 \cdot 0 = 0$$

$$3 \cdot 0 = 0 \qquad 4 \cdot 3 = 12$$

A * B (γραμμή × στήλη)

2	6
6	12

Σ γραμμή_i · στήλη_j

$$1 \cdot 2 + 2 \cdot 0 = 2$$

$$1 \cdot 0 + 2 \cdot 3 = 6$$

$$3 \cdot 2 + 4 \cdot 0 = 6$$

```
A = [1 2; 3 4]; B = [2 0; 0 3];
```

```
% Element-wise (στοιχείο-με-στοιχείο)
```

```
A .* B % → [2 0; 0 12]
```

```
A ./ B % → [0.5 Inf; Inf 1.33]
```

```
A .^ 2 % → [1 4; 9 16]
```

```
% Matrix (γραμμική άλγεβρα)
```

```
A * B % → [2 6; 6 12]
```

```
A ^ 2 % → A*A = [7 10; 15 22]
```

```
% Σήματα: ΠΑΝΤΑ element-wise!
```

```
t = linspace(0, 1, 100);
```

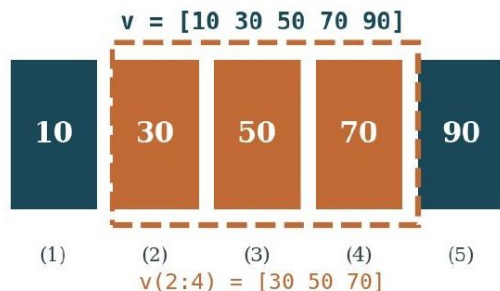
```
y = sin(2*pi*t) .* exp(-t); % ✓
```

```
% y = sin(2*pi*t) * exp(-t); % ✗
```

- .* ./ .^ : κάθε (i,j) χωριστά
- * / ^ : γραμμικής άλγεβρας (Σ γραμμή×στήλη)
- Σφάλμα "nonconformant": λάθος διαστάσεις
- Ενσωματωμένες (sin, exp, abs) → αυτόματα .*
- * χρησιμοποιείται στη συνέλιξη (Εβδ. 4)
- Κανόνας: αν έχεις t[] → πάντα .* ./ .^

Ευρετηρίαση (Indexing) — Διανύσματα & Πίνακες

Διανύσματα — Ευρετηρίαση



$v(1)$ → 10 (πρώτο)
 $v(\text{end})$ → 90 (τελευταίο)
 $v(2:4)$ → [30 50 70]
 $v([1 \ 3 \ 5])$ → [10 50 90]
 $v(\text{end}-1:\text{end})$ → [70 90]
 $\text{length}(v)$ → 5

Πίνακες — Ευρετηρίαση

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

$A(2,3)$ → 7 (γραμμή 2, στήλη 3)
 $A(1,:)$ → [1 2 3 4] (γραμμή 1)
 $A(:,2)$ → [2;6;10] (στήλη 2)
 $A(1:2,2:3)$ → υποπίνακας 2x2

```
% Διανύσματα
v = [10 30 50 70 90];
v(1)      % → 10
v(end)    % → 90
v(2:4)    % → [30 50 70]
v([1 3 5]) % → [10 50 90]
```

```
% Πίνακες
A = magic(4);
A(2,3) % στοιχείο
A(1,:) % γραμμή 1
A(:,2) % στήλη 2
A(1:2,1:2) % υποπίνακας
```

- 1-based: $v(1)$ = πρώτο (ΟΧΙ $v(0)$)
- $v(\text{end})$ = τελευταίο στοιχείο
- Slicing: $v(a:b)$, $v(a:\text{step}:b)$
- Logical index: $v(v > 3)$
- $A(:)$ = reshape σε στήλη
- $\text{find}(v > 50)$ → δείκτες που ισχύει

Ανακεφαλαίωση



Περιβάλλον

Command Window, Workspace, Editor



Μεταβλητές

double, complex, char, logical



Διανύσματα & Πίνακες

linspace, colon, element-wise ops



Γραφικές

plot, stem, subplot, hold on



Scripts

.m αρχεία, clc/clear/close









Βοήθεια

help, doc, lookfor

Αυτά τα εργαλεία αποτελούν τη βάση για ΟΛΟ το εξάμηνο!

Ασκήσεις Εργαστηρίου

-  **Βασικό** Δημιουργήστε ημίτονο 10 Hz, σχεδιάστε με plot()
-  **Βασικό** Φτιάξτε πίνακα 3×3 και βρείτε ορίζουσα & αντίστροφο
-  **Μέτριο** Σχεδιάστε sin, cos, tan σε subplot(3,1,k) για $t \in [0, 2\pi]$
-  **Μέτριο** Γράψτε script: δημιουργία σήματος $x(t) = Ae^{(-at)}\sin(2\pi ft)$
-  **Προχωρ.** Δημιουργήστε 5 ημίτονα διαφορετικής f, σχεδιάστε και αθροίστε

 Ξεκινήστε από τα Βασικά, προχωρήστε στα Μέτρια, δοκιμάστε τα Προχωρημένα όταν κατανοήσετε τα προηγούμενα!

Πόροι & Χρήσιμοι Σύνδεσμοι

GNU Octave Documentation

octave.org/doc

Πλήρης τεκμηρίωση

Octave Wiki

wiki.octave.org

Παραδείγματα & tutorials

Octave Packages

packages.octave.org

Signal, Control, Symbolic

Jupyter Notebooks

jupyter.org

Interactive environment

Octave Online

octave-online.net

Octave στο browser (χωρίς εγκατάσταση)

Προτεινόμενα βιβλία:

Jesper Schmidt Hansen — *GNU Octave Beginner's Guide*

• John W. Eaton et al. —

GNU Octave Manual (free online)

Εβδομάδα 2 — Βασικά Σήματα

Τι θα δούμε την επόμενη εβδομάδα:

- 1 Ημιτονοειδή σήματα — πλάτος, συχνότητα, φάση
- 2 Εκθετικά σήματα — πραγματικά & μιγαδικά
- 3 Βηματική & κρουστική συνάρτηση (step, impulse)
- 4 Η συνάρτηση sinc — ζωτική στην επεξεργασία σημάτων
- 5 Ενέργεια & ισχύς σημάτων

Προετοιμασία: Ολοκληρώστε τις ασκήσεις Βασικό/Μέτριο και εξοικειωθείτε με τα scripts!



Ερωτήσεις;

Σήματα και Συστήματα — 4.004

octave.org • jupyter.org • packages.octave.org