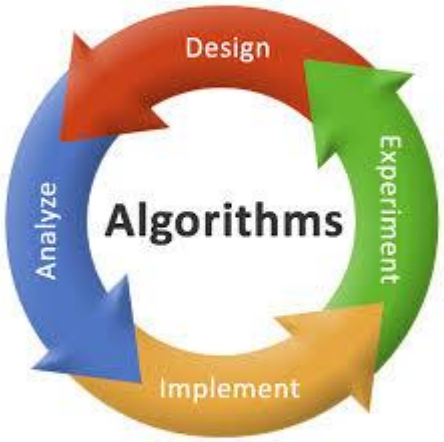


Γραμμική Άλγεβρα
(*Linear Algebra*)

ΑΓΓΕΛΟΣ ΣΙΦΑΛΕΡΑΣ

Καθηγητής

1η Διάλεξη (Θεωρία)

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΠΛΗΡΟΦΟΡΙΑΣ
Τμήμα Εφαρμοσμένης Πληροφορικής

CMOR Lab
Computational Methodologies and Operations Research

Ανακοινώσεις

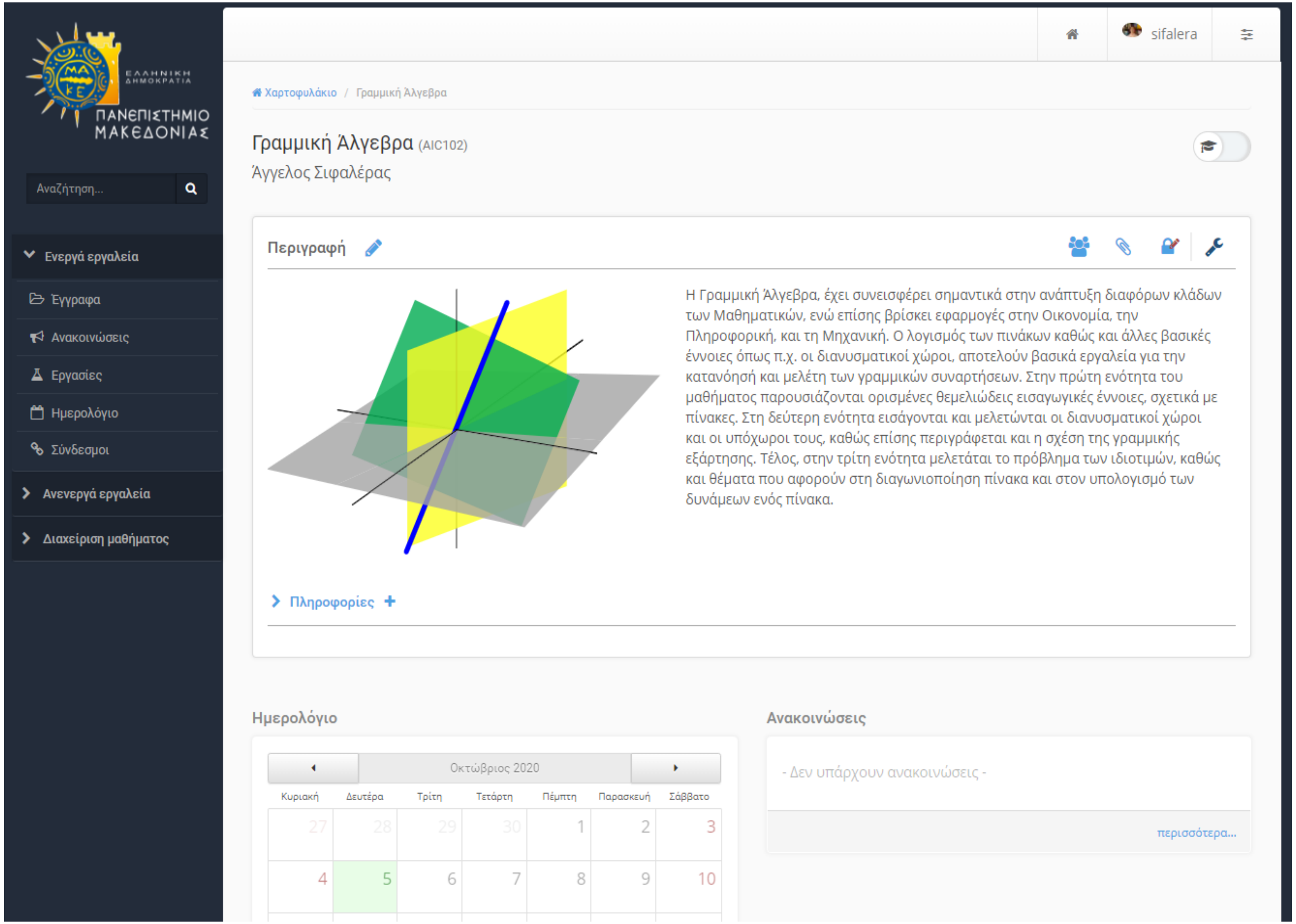
Μετά από κάθε μάθημα, θεωρία ή εργαστήριο, διαφάνειες, κώδικες και υποστηρικτικό υλικό θα ανακοινώνονται στην διεύθυνση

https://openeclass.uom.gr/courses/DAI115

E-mail: sifalera@uom.gr

Ώρες γραφείου κάθε Τρίτη 13:00 – 14:00 & Πέμπτη 11:00 – 13:00

Ιστοχώρος μαθήματος



eClass

https://openeclass.uom.gr/courses/DAI115

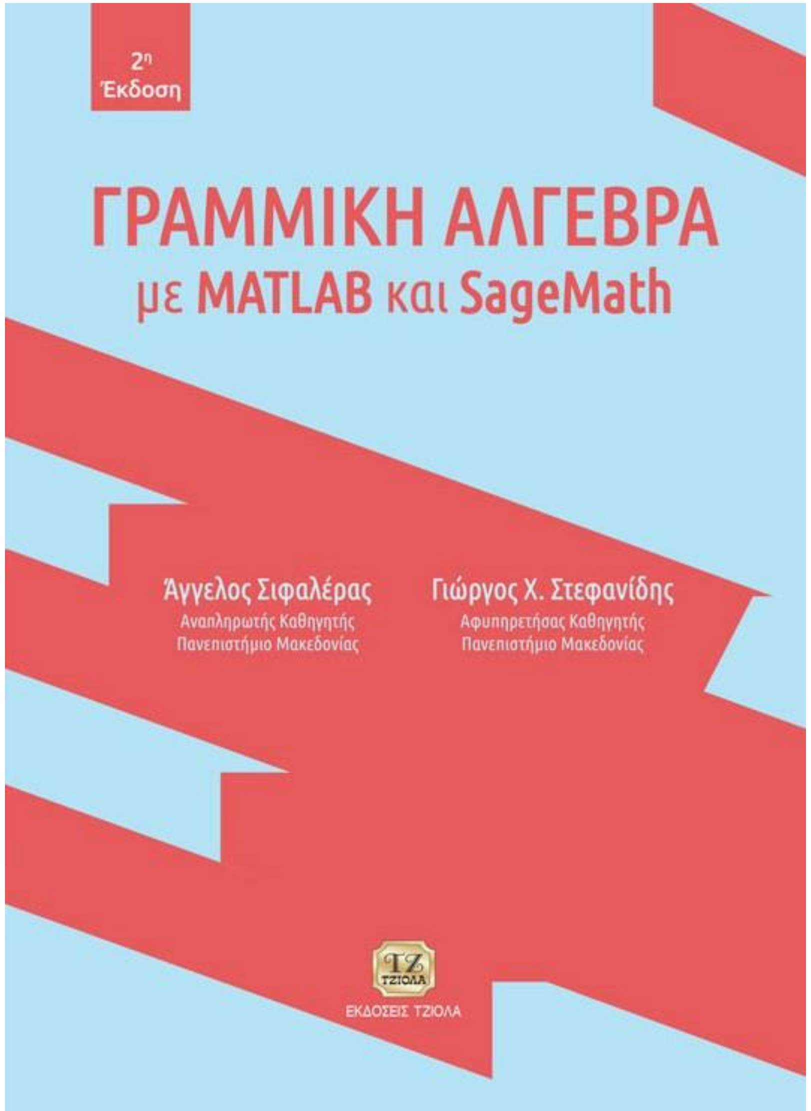
Μέχρι να σας χορηγηθούν ιδρυματικοί λογαριασμοί από το ΚΥΔ μπορείτε να χρησιμοποιείτε το

Demo account:

user id: tempuser
password: 20uom23

Σύγγραμμα

Σιφαλέρας Α. και Στεφανίδης Γ. Χ., (2021), *Γραμμική Άλγεβρα με MATLAB και SageMath*, 2η έκδοση, Εκδόσεις Τζιόλα.



Εισαγωγή

- Η Γραμμική Άλγεβρα, έχει συνεισφέρει σημαντικά στην ανάπτυξη διαφόρων κλάδων των Μαθηματικών, ενώ επίσης βρίσκει εφαρμογές στην Οικονομία, την Πληροφορική, και τη Μηχανική.

- Στην πρώτη ενότητα του μαθήματος παρουσιάζονται ορισμένες θεμελιώδεις εισαγωγικές έννοιες, σχετικά με πίνακες.

- Στη δεύτερη ενότητα εισάγονται και μελετώνται οι διανυσματικοί χώροι και οι υπό-χωροι τους, καθώς επίσης περιγράφεται και η σχέση της γραμμικής εξάρτησης.

- Τέλος, στην τρίτη ενότητα μελετάται το πρόβλημα των ιδιοτιμών, καθώς και θέματα που αφορούν στη διαγωνιοποίηση πίνακα και στον υπολογισμό των δυνάμεων ενός πίνακα.

Περιεχόμενο μαθήματος

Στα πλαίσια του μαθήματος, θα παρουσιαστούν τα ακόλουθα θέματα:

- Βασικά στοιχεία Γραμμικής Άλγεβρας, πίνακες (ορισμοί - ιδιότητες πράξεων)
- Γραμμικά συστήματα
- Διανυσματικοί χώροι – εφαρμογές
- Ιδιοτιμές – ιδιοδιανύσματα
- Εξοικείωση με το λογισμικό πακέτο *SageMath* (*free open-source mathematics software system* – *http://www.sagemath.org*)

Διεθνείς Ερευνητικές Κοινότητες σε Γραμμική Άλγεβρα

SIAM Activity Group on Linear Algebra

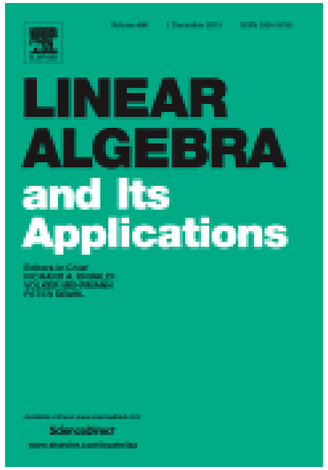
http://siags.siam.org/siagla

Διεθνή Συνέδρια σε Γραμμική Άλγεβρα

SIAM Conference on Applied Linear Algebra (SIAM-LA24)
May 13-17, 2024, Paris, France
https://www.siam.org/conferences/cm/conference/la24

25th Conference of the International Linear Algebra Society (ILAS)
July 12-16, 2023, Madrid, Spain
https://ilas2023.es

Διεθνή Περιοδικά σε Γραμμική Άλγεβρα



Linear Algebra and its Applications
http://www.journals.elsevier.com/linear-algebra-and-its-applications

Linear and Multilinear Algebra
http://www.tandfonline.com/loi/glma20

Εισαγωγή στις μήτρες (πίνακες)

- Μήτρα:

$$A = \left(a_{ij}\right) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in M_{m\times n}(F)$$

- (i, j) – στοιχείο της μήτρας: a_{ij}
- γραμμές: m
- στήλες: n
- μέγεθος: $m\times n$

- **Τανυστής** (*tensor*): *n*-διάστατος πίνακας (*array*) / διάταξη αριθμών
- **Μήτρα** (*matrix*): δισδιάστατος <u>ορθογώνιος</u> πίνακας (*array*) / διάταξη αριθμών, διάστασης ή μεγέθους *m x n* ή τανυστής 2ης τάξεως (*second order tensor*)
- **Διάνυσμα** (*vector*): μονοδιάστατος πίνακας (*array*) ή τανυστής 1ης τάξεως (*first order tensor*)

Εισαγωγή στους πίνακες

- Ίσοι πίνακες:

$$\text{Έστω}\ \ A=[a_{ij}]_{m\times n},\quad B=[b_{ij}]_{m\times n}$$

$$\text{Τότε}\ A=B \Leftrightarrow \forall\ (i,\,j),\ a_{ij}=b_{ij}$$

- Παράδειγμα: (ισότητα πινάκων)

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\qquad\qquad B=\begin{bmatrix}a & b\\ c & d\end{bmatrix}$$

$$\text{Αν}\ \ A=B$$

$$\text{Τότε}\quad a=1,\quad b=2,\quad c=3,\quad d=4$$

Εισαγωγή στους πίνακες

- Μηδενικός πίνακας ($O_{m,n}$ ή O):

Αν $A \in M_{m,n}$(R) και $\alpha_{ij} = 0$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, τότε ο A λέγεται **μηδενικός πίνακας**, (συμβολικά $\mathrm{O}_{m,n}$ ή O, αν τα m, n εννοούύνται)

$$O_{2,3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Εισαγωγή στους πίνακες

- Ανάστροφος (*transpose*) ενός πίνακα A

$$\text{Αν } A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in M_{m\times n}$$

$$\text{Τότε } A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} \in M_{n\times m}$$

π.χ., ο πίνακας $\begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 1 \end{bmatrix}$ είναι ο ανάστροφος πίνακας του $\begin{bmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 1 \end{bmatrix}$

Εισαγωγή στους πίνακες

- Παραδείγματα: (αναστροφή πινάκων)

$$(a)\quad A=\begin{bmatrix}2\\8\end{bmatrix}\qquad \Rightarrow A^T=\begin{bmatrix}2 & 8\end{bmatrix}$$

$$(b)\quad A=\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}\qquad \Rightarrow A^T=\begin{bmatrix}1 & 4 & 7\\2 & 5 & 8\\3 & 6 & 9\end{bmatrix}$$

$$(c)\quad A=\begin{bmatrix}0 & 1\\2 & 4\\1 & -1\end{bmatrix}\qquad \Rightarrow A^T=\begin{bmatrix}0 & 2 & 1\\1 & 4 & -1\end{bmatrix}$$

- Ιδιότητες ανάστροφων πινάκων:

$$(1)\ \ (A^T)^T = A$$

$$(2)\ \ (A+B)^T = A^T + B^T$$

$$(3)\ \ (cA)^T = c(A^T)$$

$$(4)\ \ (AB)^T = B^T A^T$$

Εισαγωγή στους πίνακες

- Τραπεζοειδής άνω & κάτω πίνακας:

Τραπεζοειδής άνω, όταν $a_{ij} = 0 \;\; \forall\; i > j$.

π.χ.:

$$\begin{bmatrix} 2 & 1 & 4 & 0 & 3 \\ 0 & 6 & 0 & 1 & 1 \\ 0 & 0 & 4 & 3 & 0 \\ 0 & 0 & 0 & 3 & 2 \end{bmatrix}, \quad \begin{bmatrix} 2 & -3 & 1 \\ 0 & -1 & 2 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}$$

Τραπεζοειδής κάτω, όταν $a_{ij} = 0 \;\; \forall\; i < j$.

π.χ.:

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 5 & 0 & 0 \\ 0 & 3 & 6 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 \\ 1 & 3 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Εισαγωγή στους πίνακες

- Τριγωνικός άνω & κάτω πίνακας:

Ένας <u>τετραγωνικός</u> πίνακας A λέγεται **τριγωνικός άνω** ή **κάτω**, όταν είναι τραπεζοειδής άνω ή κάτω, αντίστοιχα.

π.χ.:

$$\begin{bmatrix} 2 & 1 & 4 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 0 & 2 & 0 \\ 1 & 5 & 2 & 6 \end{bmatrix}$$

Εισαγωγή στους πίνακες

- Συμμετρικός & αντισυμμετρικός πίνακας:

Ένας τετραγωνικός πίνακας A λέγεται **συμμετρικός**, όταν $a_{ij} = a_{ji}\ \forall\ (i, j)$, δηλ. όταν $A = A^{\mathrm{T}}$.

π.χ.:

$$\begin{bmatrix} 2 & 1 & 0 & 4 \\ 1 & 5 & 2 & 0 \\ 0 & 2 & 1 & 7 \\ 4 & 0 & 7 & 2 \end{bmatrix}$$

Ένας τετραγωνικός πίνακας A λέγεται **αντισυμμετρικός**, όταν $a_{ij} = -a_{ji}\ \forall\ (i, j)$, δηλ. όταν $A = -A^{\mathrm{T}}$.

π.χ.:

$$\begin{bmatrix} 0 & 1 & 0 & 4 \\ -1 & 0 & -2 & 3 \\ 0 & 2 & 0 & 7 \\ -4 & -3 & -7 & 0 \end{bmatrix}$$

τα στοιχεία της κυρίας διαγωνίου
$(a_{11}, a_{22}, \ldots, a_{nn}) = 0$

Εισαγωγή στους πίνακες

- Διαγώνιος πίνακας:

όταν $a_{ij} = 0 \;\; \forall \; i \neq j$ και γράφουμε συνήθως $A = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$

Παραδείγματα:

$$\mathrm{diag}(2,4) = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}, \quad \mathrm{diag}(1,0,3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Κάθε διαγώνιος πίνακας είναι και συμμετρικός.

- Μοναδιαίος πίνακας:

Ο n-τετραγωνικός πίνακας $I_n = \mathrm{diag}(1, 1, \ldots, 1) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$

λέγεται ότι είναι ο n-τετραγωνικός ταυτοτικός ή μοναδιαίος πίνακας.

Εισαγωγή στους πίνακες

- Διαγώνιος ζώνης (*banded matrix*):

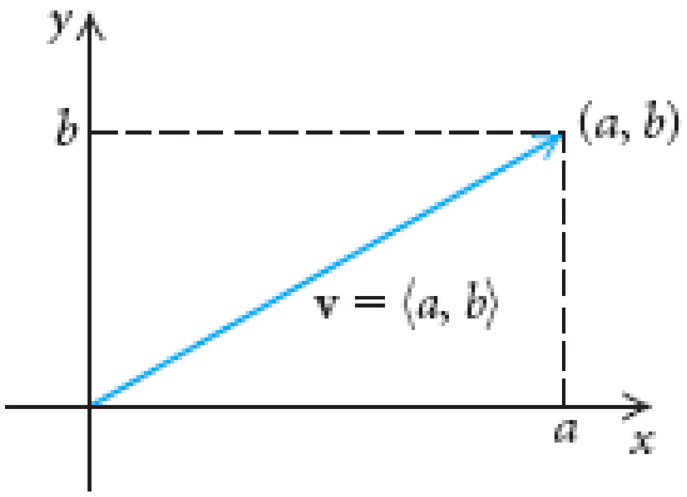
Ένας τετραγωνικός πίνακας A λέγεται **διαγώνιος ζώνης (***banded matrix***)**, όταν τα στοιχεία a_{ij} είναι μηδέν εκτός απ' αυτά που βρίσκονται στην κύρια διαγώνιο και ορισμένα που βρίσκονται κοντά και παράλληλα προς αυτή.

π.χ.:

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 4 & 5 & 2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 7 & 2 \end{bmatrix}$$

Διανύσματα

(γεωμετρική περιγραφή δ/των)



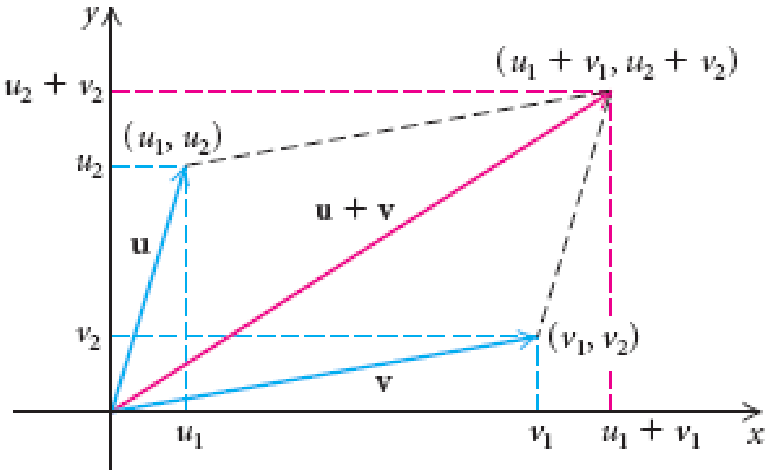
- i-οστό διάνυσμα γραμμή:

$$r_i = \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{bmatrix}$$ πίνακας γραμμή *(1 × n)*

- j-οστό διάνυσμα στήλη:

$$c_j = \left.\begin{bmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{mj} \end{bmatrix}\right\}$$ συνιστώσες ή συντεταγμένες δ/τος

πίνακας στήλη *(m × 1)*



- τετραγωνικός πίνακας: $m = n$

Μέτρο δ/τος

- Η **στάθμη**, ή **νόρμα** (*norm*), ή **μέτρο** (μήκος) διανύσματος ορίζεται ως: $\|\boldsymbol{v}\| = \sqrt{\boldsymbol{v}\cdot\boldsymbol{v}} = \sqrt{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} = \sqrt{\sum_{i=1}^{n} v_i^{\,2}}$

- Το διάνυσμα $\boldsymbol{e}_k$, είναι ένα διάνυσμα του οποίου όλες οι συνιστώσες είναι 0, πλην της k συνιστώσας που είναι 1, δηλ.:

$$\boldsymbol{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 1_k \\ \vdots \\ 0 \end{bmatrix}$$

- Ισχύει: $\|\boldsymbol{e}_k\| = 1$

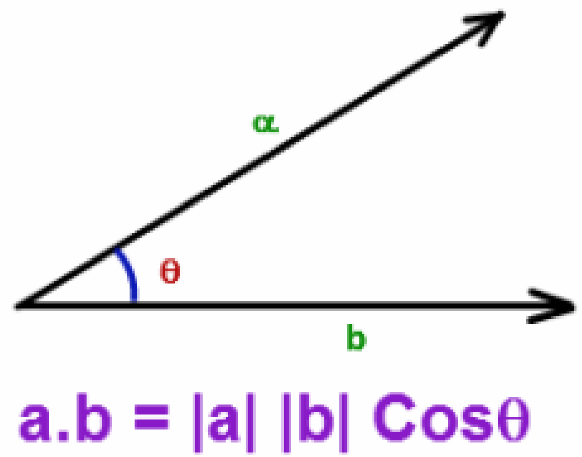
Εσωτερικό γινόμενο δ/των

- <u>Αλγεβρική περιγραφή</u>

Έστω ***u*** και ***v*** είναι δυο *n*-διάστατα διανύσματα. Τότε το **εσωτερικό τους γινόμενο** (*dot product* ή *inner product*) ορίζεται ως ο αριθμός:

$$u \cdot v = u^T v = u_1 v_1 + u_2 v_2 + \ldots + u_n v_n = \sum_{i=1}^{n} u_i v_i$$

- <u>Γεωμετρική περιγραφή</u>



Παραδείγματα υπολογισμού εσωτερικού γινομένου

Έστω τα δυο δ/τα: $A = \begin{bmatrix} 1 & 3 & -5 \end{bmatrix}$ και $B = \begin{bmatrix} 4 & -2 & -1 \end{bmatrix}$

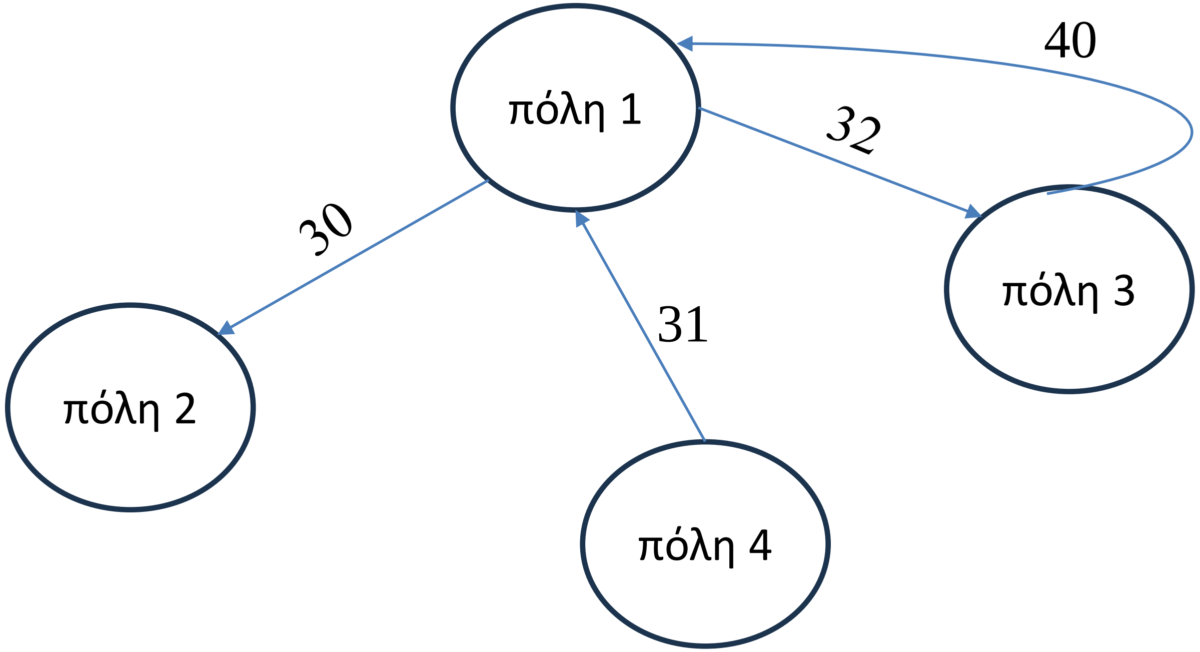
Να βρεθεί το εσωτερικό γινόμενο Α·Β

τότε:
$$\begin{aligned} A \cdot B &= \begin{bmatrix} 1 & 3 & -5 \end{bmatrix} \begin{bmatrix} 4 & -2 & -1 \end{bmatrix} = (1)(4) + (3)(-2) + (-5)(-1) \\ &= 4 - 6 + 5 \\ &= 3 \end{aligned}$$

Εφαρμογές Γραμμικής Άλγεβρας

- Θεωρία γραφημάτων
- Επεξεργασία εικόνας
- …

Αναπαράσταση γραφημάτων με χρήση πινάκων



Γράφημα με χρήση βαρών (δίκτυο), κατευθυνόμενο γράφημα, …

Μήτρα γειτονιάς (γειτνίασης) : $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

Εφαρμογές σε *logistics optimization, social media analysis*, κλπ…

Επεξεργασία εικόνας με χρήση πινάκων

- Μία εικόνα μπορεί να οριστεί ως μία συνάρτηση δύο διαστάσεων *f(x,y)*, όπου *x* και *y* οι συντεταγμένες, και η τιμή της *f* για κάθε ζευγάρι συντεταγμένων *(x,y)* καλείται ως φωτεινότητα της εικόνας στο σημείο αυτό.

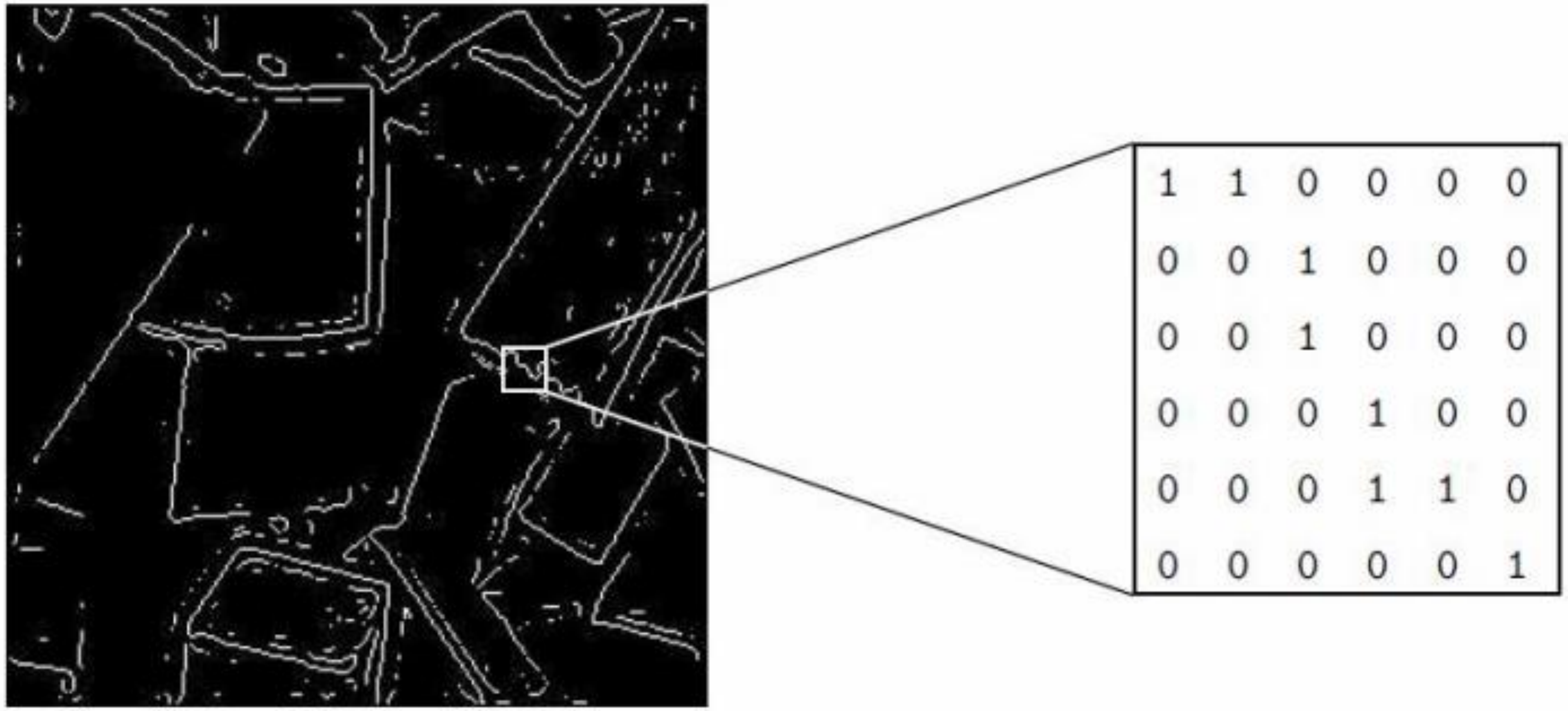
- Στις μονοχρωματικές εικόνες συνηθίζεται να χρησιμοποιείται ο όρος “επίπεδο γκρι”.

- Μια ψηφιακή εικόνα αποτελείται από ένα δισδιάστατο ή τρισδιάστατο πίνακα εικονοστοιχείων (*pixels*). Κάθε *pixel* περιέχει έναν αριθμό ή περισσότερους αριθμούς που μας δείχνει μία τιμή στην κλίμακα του γκρι ή σε έγχρωμη κλίμακα που έχει εκχωρηθεί σε αυτό.

Επεξεργασία εικόνας με χρήση πινάκων

- ***Binary***. Κάθε *pixel* είναι μαύρο ή άσπρο.

- Εφόσον υπάρχουν δύο μόνο τιμές που μπορεί να πάρει ένα *pixel* χρειαζόμαστε μόνο ένα *bit* για κάθε *pixel*.



Επεξεργασία εικόνας με χρήση πινάκων

- ***Greyscale***: Κάθε *pixel* παίρνει μία τιμή γκρίζου , από το 0 (μαύρο) μέχρι το 255 (άσπρο) όπως φαίνεται στη διπλανή εικόνα:

230	229	232	234	235	232	148
237	236	236	234	233	234	152
255	255	255	251	230	236	161
99	90	67	37	94	247	130
222	152	255	129	129	246	132
154	199	255	150	189	241	147
216	132	162	163	170	239	122

Επεξεργασία εικόνας με χρήση πινάκων

- ***True colour*** ή ***RGB***: Κάθε *pixel* έχει μία τιμή χρώματος και το χρώμα περιγράφεται με βάση το ποσοστό που περιέχει σε κόκκινο (*Red*), μπλε (*Blue*), και πράσινο (*Green*).

- Κάθε ένα από αυτά τα *RGB* στοιχεία μπορεί να πάρει τιμή 0-255 (αυτό μας δίνει 255^3 = 16,777,216 διαφορετικά πιθανά χρώματα στην εικόνα).

49	55	56	57	52	53
58	60	60	58	55	57
58	58	54	53	55	56
83	78	72	69	68	69
88	91	91	84	83	82
69	76	83	78	76	75
61	69	73	78	76	76

Red

64	76	82	79	78	78
93	93	91	91	86	86
88	82	88	90	88	89
125	119	113	108	111	110
137	136	132	128	126	120
105	108	114	114	118	113
96	103	112	108	111	107

Green

66	80	77	80	87	77
81	93	96	99	86	85
83	83	91	94	92	88
135	128	126	112	107	106
141	129	129	117	115	101
95	99	109	108	112	109
84	93	107	101	105	102

Blue

Επεξεργασία εικόνας με χρήση πινάκων

In 1

```python
# For this demonstration I'll use the python imaging library (PIL) and a function to display images in the
# Jupyter notebook
from PIL import Image
from IPython.display import display

# my photo...
im = Image.open('photo.png')
display(im)
```

Executed at 2023.10.02 22:15:37 in 53ms

Επεξεργασία εικόνας με χρήση πινάκων

In 2

```python
import numpy as np

# Now, we can convert this PIL image to a numpy array
array=np.array(im)
print(array.shape)
array
```

Executed at 2023.10.02 22:15:54 in 73ms

```
(346, 300)
```

Out 2

1-10 346 rows × 300 columns np.ndarray CSV

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	175	176	176	177	177	178	178	179	177	177	177	177	177	177	177	177	178	177	176
1	176	176	177	177	177	177	178	178	177	177	177	177	177	177	177	177	178	177	176
2	178	178	177	177	177	177	176	176	177	177	177	177	177	177	177	177	178	177	176
3	179	178	178	177	177	176	176	175	177	177	177	177	177	177	177	177	178	177	176
4	179	178	178	177	177	176	176	175	177	177	177	177	177	177	177	177	178	177	176
5	178	178	177	177	177	177	176	176	177	177	177	177	177	177	177	177	178	177	176
6	176	176	177	177	177	177	178	178	177	177	177	177	177	177	177	177	178	177	176
7	175	176	176	177	177	178	178	179	177	177	177	177	177	177	177	177	178	177	176

Επεξεργασία εικόνας με χρήση πινάκων

In 3

```
# Here we see that we have a 346x300 array and that the values are all uint8. The uint means that they are
# unsigned integers (so no negative numbers) and the 8 means 8 bits per byte. This means that each value can
# be up to 2*2*2*2*2*2*2*2=256 in size (well, actually 255, because we start at zero). For black and white
# images black is stored as 0 and white is stored as 255. So if we just wanted to invert this image we could
# use the numpy array to do so

# Let's create an array the same shape
mask=np.full(array.shape,255)
mask
```

Executed at 2023.10.02 22:16:03 in 12ms

Out 3

1-10 346 rows × 300 columns np.ndarray CSV

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
1	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
2	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
3	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
4	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
5	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
6	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
7	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Επεξεργασία εικόνας με χρήση πινάκων

In 4

```python
# Now let's subtract the modified array from the mask
modified_array = mask - array

# Now let's subtract that from the modified array
# modified_array=array-mask

# And as a last step, let's tell numpy to set the value of the datatype correctly
modified_array = modified_array.astype(np.uint8)

modified_array
```

Executed at 2023.10.02 22:16:08 in 9ms

Out 4

1-10 346 rows × 300 columns np.ndarray CSV

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	80	79	79	78	78	77	77	76	78	78	78	78	78	78	78	78	77	78	79
1	79	79	78	78	78	78	77	77	78	78	78	78	78	78	78	78	77	78	79
2	77	77	78	78	78	78	79	79	78	78	78	78	78	78	78	78	77	78	79
3	76	77	77	78	78	79	79	80	78	78	78	78	78	78	78	78	77	78	79
4	76	77	77	78	78	79	79	80	78	78	78	78	78	78	78	78	77	78	79
5	77	77	78	78	78	78	79	79	78	78	78	78	78	78	78	78	77	78	79
6	79	79	78	78	78	78	77	77	78	78	78	78	78	78	78	78	77	78	79
7	80	79	79	78	78	77	77	76	78	78	78	78	78	78	78	78	77	78	79

Επεξεργασία εικόνας με χρήση πινάκων

In 5

```python
# And lastly, lets display this new array. We do this by using the fromarray() function in the python
# imaging library to convert the numpy array into an object jupyter can render
display(Image.fromarray(modified_array))
```

Executed at 2023.10.02 22:16:14 in 30ms