Επιστημονικός Υπολογισμός Ι

Ενότητα **3** : Βασικές Πράξεις Αριθμητικής Γραμμικής Άλγεβρας

Ευστράτιος Γαλλόπουλος
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο **«Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών»** 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.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοπός Ενότητας

- Θεμελιώδη προβλήματα της αριθμητικής γραμμικής άλγεβρας.
- Η ιεραρχία BLAS.
- Πλοκαδοποίηση και βασικές πράξεις στο υπολογιστικό μοντέλο ιεραρχικής μνήμης.
- Υπερταχύς πολλαπλασιασμός Strassen και παραλλαγές.

Περιεχόμενα

Υπενθύμιση και πρόγραμμα διάλεξης

Στην προηγούμενη διάλεξη μιλήσαμε για

- τον πολλαπλασιασμό MM (πράξη BLAS-3)
- ονοματολογία και μερικές βιβλιοθήκες υλοποίησης των BLAS.
- θέματα μετάφρασης - ειδικότερα το loop unrolling.
- την αυτόματη επιλογή βέλτιστων παραμέτρων και αυτόματη παραγωγή `βέλτιστου κώδικα' (project ATLAS) για τα BLAS
- μεθόδους υπολογισμού $\mathtt{MM}$ μειωμένης πολυπλοκότητας πράξεων.

Υπενθύμιση και πρόγραμμα διάλεξης

Στην προηγούμενη διάλεξη μιλήσαμε για

- τον πολλαπλασιασμό MM (πράξη BLAS-3)
- ονοματολογία και μερικές βιβλιοθήκες υλοποίησης των BLAS.
- θέματα μετάφρασης - ειδικότερα το loop unrolling.
- την αυτόματη επιλογή βέλτιστων παραμέτρων και αυτόματη παραγωγή 'βέλτιστου κώδικα' (project ATLAS) για τα BLAS
- μεθόδους υπολογισμού MM μειωμένης πολυπλοκότητας πράξεων.

Σήμερα θα συζητήσουμε τα εξής:

- Αραιά μητρώα και μερικά ζητήματα που προκύπτουν στη διαχείρισή τους (μια εισαγωγή)
- Εισαγωγή στο μοντέλο αριθμητικής κινητής υποδιαστολής και στα σφάλματα στρογγύλευσης

Αραιά μητρώα

Ένα μητρώο $m \times n$ αποκαλείται **αραιό** (sparse) αν το πλήθος των μη μηδενικών του, nnz, είναι κατά πολύ μικρότερο του πλήθους των στοιχείων του. Συνήθως, αυτό σημαίνει ότι $\mathrm{nnz} = \mathcal{O}(\min\{m, n\})$.

- Η αραιότητα αξιοποιείται με (αραιές αναπαραστάσεις) σε ειδικές δομές δεδομένων ώστε τα αραιά μητρώα να αποθηκεύονται σε χώρο πολύ μικρότερο του $\mathcal{O}(mn)$.
- Προσοχή: Στην περίπτωση αυτή, οι κώδικες διαχείρισής τους πρέπει να λαμβάνουν υπόψη την αραιή δομή.
- Αν επιτελέσουμε μία πράξη μεταξύ δύο αραιών μητρώων, το αποτέλεσμα μπορεί να είναι αραιό ή όχι. Π.χ. αν $A \in \mathbb{R}^{n \times n}$ τριδιαγώνιο τότε το A^{n-1} θα είναι πυκνό.

Αραιά μητρώα

Η αραιότητα είναι η πιο συνηθισμένη δομική ιδιότητα των μητρώων που προκύπτουν στις εφαρμογές. Την αξιοποιούμε ως εξής:

- Κατά την αποθήκευση του μητρώου, με αραιές τεχνικές, ειδικές δομές δεδομένων, κλπ.
- Κατά την επιλογή μεθόδου χρήσης του μητρώου στα προβλήματα της υπολογιστικής γραμμικής άλγεβρας. Π.χ. στην επίλυση συστημάτων χρησιμοποιούνται:
  - άμεσες αραιές μέθοδοι (π.χ. βλ. (Dav06)), ή
  - επαναληπτικές μέθοδοι (π.χ. βλ. (Saa03))

Οι μέθοδοι επίλυσης, χρησιμοποιούν ειδικούς «υπολογιστικούς πυρήνες» για τις απλές πράξεις

- π.χ. sparse BLAS, όπως «πολλαπλασιασμός αραιού μητρώου με διάνυσμα».
- «κρυφός πολλαπλασιασμός» μέσω υπορουτίνας «μαύρου κουτιού»

Είδη μητρώων στα BLAS, αναπαράσταση στη μνήμη και κώδικες

Παραδείγματα χρήσης στους συνδέσμους dtrmv.c, dtpmv.c, zhymv.c, dgbmv.c

UPLO	Triangular matrix A	Storage in array A
'U'	$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{pmatrix}$	$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} \\ * & a_{22} & a_{23} & a_{24} \\ * & * & a_{33} & a_{34} \\ * & * & * & a_{44} \end{matrix}$
'L'	$\begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$	$\begin{matrix} a_{11} & * & * & * \\ a_{21} & a_{22} & * & * \\ a_{31} & a_{32} & a_{33} & * \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix}$

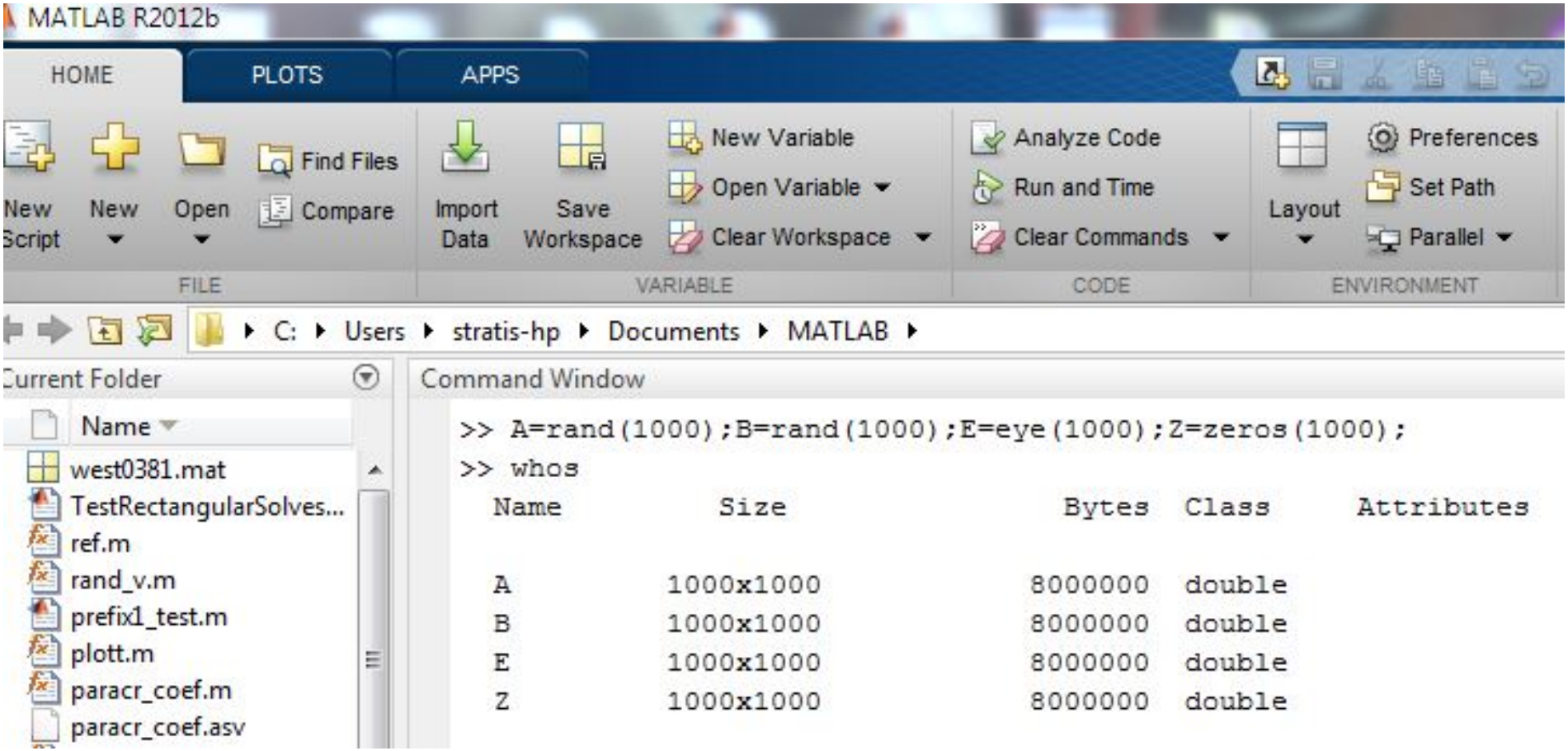
UPLO	Triangular matrix A	Packed storage in array AP
'U'	$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & a_{22} & a_{23} & a_{24} \\ & & a_{33} & a_{34} \\ & & & a_{44} \end{pmatrix}$	$a_{11}\ \underbrace{a_{12}\ a_{22}}\ \underbrace{a_{13}\ a_{23}\ a_{33}}\ \underbrace{a_{14}\ a_{24}\ a_{34}\ a_{44}}$
'L'	$\begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$	$\underbrace{a_{11}\ a_{21}\ a_{31}\ a_{41}}\ \underbrace{a_{22}\ a_{32}\ a_{42}}\ \underbrace{a_{33}\ a_{43}}\ a_{44}$

UPLO	Hermitian matrix A	Storage in array A
'U'	$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ \bar{a}_{12} & a_{22} & a_{23} & a_{24} \\ \bar{a}_{13} & \bar{a}_{23} & a_{33} & a_{34} \\ \bar{a}_{14} & \bar{a}_{24} & \bar{a}_{34} & a_{44} \end{pmatrix}$	$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} \\ * & a_{22} & a_{23} & a_{24} \\ * & * & a_{33} & a_{34} \\ * & * & * & a_{44} \end{matrix}$
'L'	$\begin{pmatrix} a_{11} & \bar{a}_{21} & \bar{a}_{31} & \bar{a}_{41} \\ a_{21} & a_{22} & \bar{a}_{32} & \bar{a}_{42} \\ a_{31} & a_{32} & a_{33} & \bar{a}_{43} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$	$\begin{matrix} a_{11} & * & * & * \\ a_{21} & a_{22} & * & * \\ a_{31} & a_{32} & a_{33} & * \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix}$

Band matrix A	Band storage in array AB
$\begin{pmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ a_{31} & a_{32} & a_{33} & a_{34} & \\ & a_{42} & a_{43} & a_{44} & a_{45} \\ & & a_{53} & a_{54} & a_{55} \end{pmatrix}$	$\begin{matrix} * & a_{12} & a_{23} & a_{34} & a_{45} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} \\ a_{21} & a_{32} & a_{43} & a_{54} & * \\ a_{31} & a_{42} & a_{53} & * & * \end{matrix}$

Υπόδειξη: Δείτε μερικούς από τους κώδικες μέσω των συνδέσμων.

Διαγώνια μητρώα: MATLAB vs. Octave I



Διαγώνια μητρώα: MATLAB vs. Octave II

```
>>ver
GNU Octave Version 3.8.1
GNU Octave License: GNU General Public License
Operating System: Linux 3.13.0-37-generic #64~precise1-Ubuntu SMP Wed Sep 24 21:37:11 UTC 2014 x86_64
Package Name    | Version | Installation directory
----------------+---------+-----------------------
miscellaneous   |   1.2.1 | /home/gallop/octave/miscellaneous-1.2.1
>> A=rand(1000);B=rand(1000);E=eye(1000);Z=zeros(1000);
>> whos
Variables in the current scope:
Attr Name        Size                     Bytes  Class
==== ====        ====                     =====  =====
A        1000x1000                 8000000  double
B        1000x1000                 8000000  double
E        1000x1000                    8000  double
Z        1000x1000                 8000000  double
Total is 4000000 elements using 24008000 bytes
```

Προσοχή: Δείτε για διαγώνια μητρώα στην Octave

Σχετικά με τα διαγώνια μητρώα στην Octave

Από το εγχειρίδιο του John W. Eaton, GNU Octave

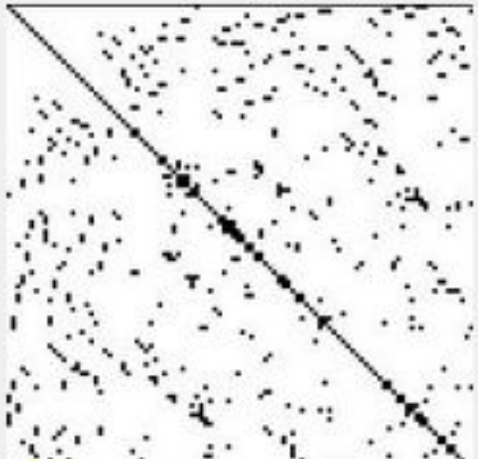
21.1.3 Explicit and Implicit Conversions

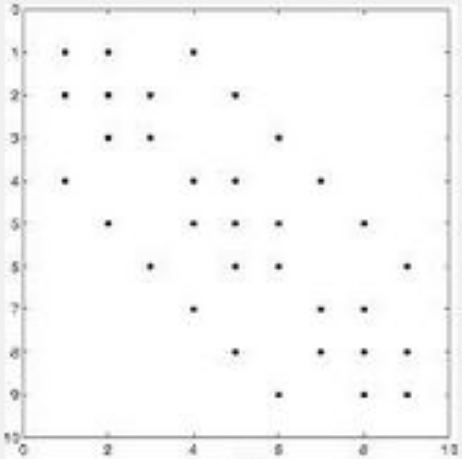
The diagonal and permutation matrices are special objects in their own right. A number of operations and built-in functions are defined for these matrices to use special, more efficient code than would be used for a full matrix in the same place. ...

To facilitate smooth mixing with full matrices, backward compatibility, and compatibility with MATLAB, the diagonal and permutation matrices should allow any operation that works on full matrices, and will either treat it specially, or implicitly convert themselves to full matrices.

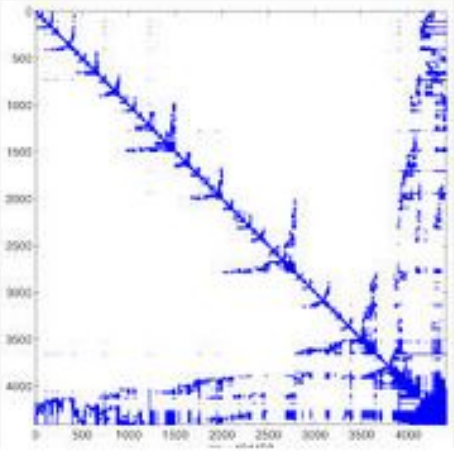
Instances include matrix indexing, except for extracting a single element or a leading submatrix, indexed assignment, or applying most mapper functions, such as exp.

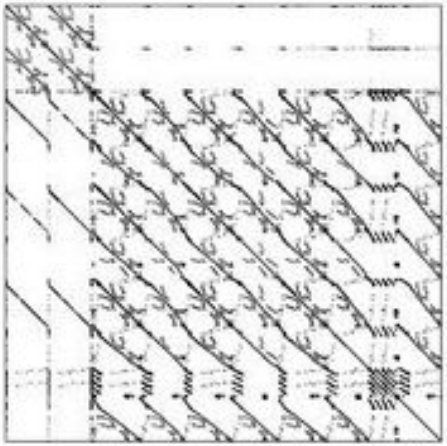
Αραιά μητρώα (δείτε τη συλλογή Davis-Florida) (DH11)

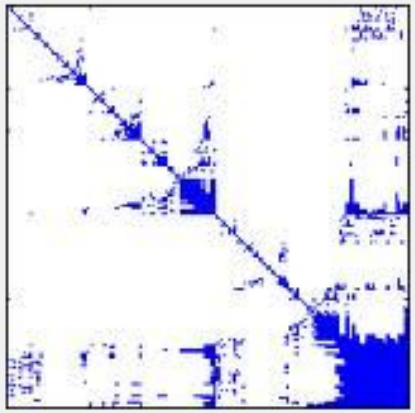




$$\begin{pmatrix} 1.0 & 0 & 5.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.0 & 0 & 0 & 0 & 0 & 11.0 & 0 \\ 0 & 0 & 0 & 0 & 9.0 & 0 & 0 & 0 \\ 0 & 0 & 6.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.0 & 0 & 0 & 0 & 0 \\ 2.0 & 0 & 0 & 0 & 0 & 10.0 & 0 & 0 \\ 0 & 0 & 0 & 8.0 & 0 & 0 & 0 & 0 \\ 0 & 4.0 & 0 & 0 & 0 & 0 & 0 & 12.0 \end{pmatrix}$$







Δυσκολίες/αντιξοότητες - προκλήσεις

Οι υπολογισμοί με αραιά μητρώα παρουσιάζουν ορισμένα σημαντικά χαρακτηριστικά:

Θέματα τοπικότητας: μειωμένη στην $\mathtt{MV}$, όπως εκφράζεται από το αναμενόμενο μ σε σχέση με εκείνο των υπολογισμών με πυκνά μητρώα.

Αναπαράσταση και επιλογή δομής.

Περίπλοκος προγραμματισμός για την ορθή διαχείριση της αναπαράστασης.

Αξιοσημείωτο: ο σχεδιασμός και η υλοποίηση αποδοτικών αλγορίθμων $\mathtt{MV}$ για αραιά μητρώα ($\mathtt{SpMV}$) είναι πολύ μεγάλης σημασίας σε πάρα πολλές εφαρμογές.

Η αραιότητα μειώνει την τοπικότητα

Διερευνούμε το αναμενόμενο μ Υπολογίζουμε Ω, $\Phi_{\min}$, $\mu_{\min}$ για $\mathtt{SpMV}$ και $\mathtt{SpMM}$ για τις πράξεις Ax, AB. Υποθέτουμε ότι το μητρώο $A \in \mathbb{R}^{n\times n}$ είναι αραιό, με $\mathtt{nnz}$ μη μηδενικά στοιχεία και ότι έχει αποθηκευτεί με αυτόν τον τρόπο.

- $\Phi_{\min} = \mathtt{nnz} + 2n$
- $\Omega = \sum_{j=1}^{n} 2n_j = 2\mathtt{nnz}$ όπου n_j το πλήθος των μη μηδενικών ανά γραμμή.
- $\mu_{\min} = (\mathtt{nnz} + 2n)/(2\mathtt{nnz}) \approx \frac{1}{2} + \frac{n}{\mathtt{nnz}}$

Πολλαπλασιασμός τετραγωνικών μητρώων, AB, όπου το A αραιό:

- $\Phi_{\min} = n^2 + \mathtt{nnz} + 2n$
- $\Omega = n\sum_{j=1}^{n} 2n_j = 2n\mathtt{nnz}$.
- $\mu_{\min} = (n^2 + \mathtt{nnz} + 2n)/(2n\mathtt{nnz}) \approx \frac{n}{2\mathtt{nnz}} + \frac{1}{\mathtt{nnz}}$

Προσέξτε ότι αναμένουμε ότι $n \leq \mathtt{nnz} \leq n^2$. Για μητρώα που είναι αρκετά αραιά, $\mathtt{nnz} = \gamma n$ για μικρό ακέραιο $\gamma \ll n$, π.χ. $\gamma = 10$

Συμπέρασμα: Η δυνάμει τοπικότητα της πραξης $\mathtt{MV}$ με αραιό μητρώο μειώνεται εξαιτίας της αραιότητας.

Αναπαράσταση και αποθήκευση

Πυκνή αποθήκευση

- σε πίνακα $m \times n$ θέσεων (Δομή Δεδομένων $\rightarrow$ Λογισμικό)
- στη μνήμη εγγράφεται σε διαδοχικές διευθύνσεις (κατά γραμμές ή στήλες) (Λογισμικό $\rightarrow$ Υλικό)
- χρειάζονται mn θέσεις μνήμης

Αραιή αποθήκευση

- Αποθηκεύονται μόνον τα μη μηδενικά καθώς και δείκτες για πρόσβαση
- Συχνά 3 μονοδιάστατοι πίνακες αρκούν
- $\approx 3\texttt{nnz}$ θέσεις μνήμης
- πολυπλοκότερος προγραμματισμός

Παράδειγμα

Αποθήκευση COO και CSR

$$\begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 \\ -2.0 & 0 & 0 & 0 & 3.0 \\ 0 & 0 & 0 & -1.0 & 0 \\ 0 & 4.0 & 0 & 0 & 0 \\ -9.0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

25 θέσεις

coordinate

row	col	val
1	1	1.
2	1	-2.
2	5	3.
3	4	-1.
4	2	4.
5	1	-9.

CSR

r_pt	col	val
1	1	1.
2	1	-2.
4	5	3.
5	4	-1.
6	2	4.
	1	-9.

COO: 18 θέσεις. CSR: 17 θέσεις.

Compressed Sparse Row (CSR / or CRS)

$$\begin{pmatrix} 1 & 0 & 0 & -1 & 0 \\ 2 & 0 & -2 & 0 & 3 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 4 & 0 & -4 & 0 \\ 5 & 0 & -5 & 0 & 6 \end{pmatrix}$$

	1	2	3	4	5	6	7	8	9	10	11
VAL	1	-1	2	-2	3	-3	4	-4	5	-5	6
JA	1	4	1	3	5	2	2	4	1	3	5
IA	1	3	6	7	9	12					

`VAL` πίνακας `nnz` α.κ.υ. Περιέχει τις μη μηδενικές τιμές α.κ.υ. διατεταγμένες (με αυξανόμενο δείκτη στήλης) ανά γραμμή

`JA` πίνακας `nnz` ακεραίων. Το `JA(i)` είναι η στήλη που περιέχει το `VAL(i)`.

`IA` πίνακας $n+1$ ακεραίων. Το `IA(i)` είναι η θέση του 1ου στοιχείου της *i*-οστής γραμμής *i* στα `JA`, `VAL`. Αν IA(i)=IA(i+1) η γραμμή είναι μηδενική. Συνήθως για ένδειξη τέλους `IA(n+1) = IA(1) +nnz`

Άλλες μέθοδοι αραιής αποθήκευσης / αραιές δομές

- Πολλές προτάσεις στη βιβλιογραφία (compressed sparse column, modified sparse row, diagonal, skyline, jagged format, blocked CSR, ...)
- Πρέπει να προσαρμόζεται ο κώδικας ανάλογα με τον τρόπο αποθήκευσης ....
- Η MATLAB απλουστεύει γιατί αναφερόμαστε σε αραιά μητρώα ως πυκνά. π.χ. το `A(i,j)` εξακολουθεί να σημαίνει το στοιχείο στη θέση (i, j) του μητρώου A αν και εσωτερικά, υλοποιείται εντελώς διαφορετικά.

Πολλαπλασιασμός SpMV

Μητρώο σε αραιή αποθήκευση

Χρησιμοποιώντας ορολογία MATLAB για τη δομή CSR

```
function [y] = spmv_csr(VAL,IA,JA,y,x)
% pollaplasiasmos y <- y + A*x opou to A einai se morfh CSR
for i=1:n
    k1=IA(i); k2=IA(i+1)-1;
    y(i) = y(i)+ dot(VAL(k1:k2),x(JA(k1:k2)));
end
```

Πολλαπλασιασμός SpMV

Μητρώο σε αραιή αποθήκευση

Χρησιμοποιώντας ορολογία MATLAB για τη δομή CSR

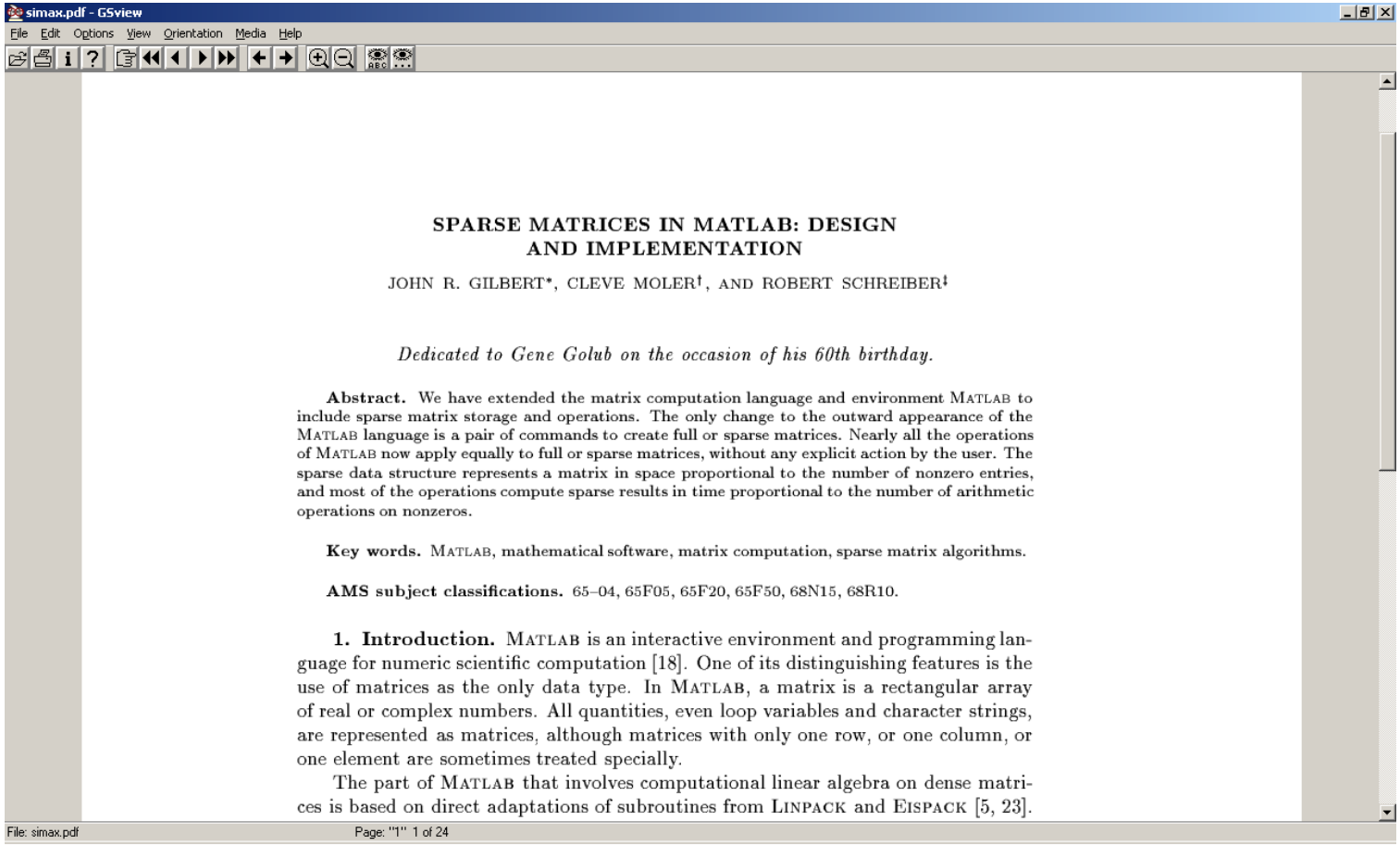
```
function [y] = spmv_csr(VAL,IA,JA,y,x)
% pollaplasiasmos y <- y + A*x opou to A einai se morfh CSR
for i=1:n
    k1=IA(i); k2=IA(i+1)-1;
    y(i) = y(i)+ dot(VAL(k1:k2),x(JA(k1:k2)));
end
```

```
function [y] = spmv_csc(VAL,IA,JA,y,x)
% pollaplasiasmos y <- y + A*x opou to A se morfh CSC
for j=1:n
    k1=IA(j); k2=IA(j+1)-1;
    y(JA(k1:k2)) = y(JA(k1:k2) + x(j)*VAL(k1:k2);
end
```

Υποδομή για αραιά μητρώα στη MATLAB

```
function [y] = spmv_csc(VAL,IA,JA,y,x)
% pollaplasiasmos y <- y + A*x opou to A se morfh CSC
for j=1:n
    k1=IA(j); k2=IA(j+1)-1;
    y(JA(k1:k2)) = y(JA(k1:k2) + x(j)*VAL(k1:k2);
end
```

Αραιά μητρώα στη MATLAB



SPARSE MATRICES IN MATLAB: DESIGN AND IMPLEMENTATION

JOHN R. GILBERT*, CLEVE MOLER†, AND ROBERT SCHREIBER‡

Dedicated to Gene Golub on the occasion of his 60th birthday.

Abstract. We have extended the matrix computation language and environment MATLAB to include sparse matrix storage and operations. The only change to the outward appearance of the MATLAB language is a pair of commands to create full or sparse matrices. Nearly all the operations of MATLAB now apply equally to full or sparse matrices, without any explicit action by the user. The sparse data structure represents a matrix in space proportional to the number of nonzero entries, and most of the operations compute sparse results in time proportional to the number of arithmetic operations on nonzeros.

Key words. MATLAB, mathematical software, matrix computation, sparse matrix algorithms.

AMS subject classifications. 65–04, 65F05, 65F20, 65F50, 68N15, 68R10.

1. Introduction. MATLAB is an interactive environment and programming language for numeric scientific computation [18]. One of its distinguishing features is the use of matrices as the only data type. In MATLAB, a matrix is a rectangular array of real or complex numbers. All quantities, even loop variables and character strings, are represented as matrices, although matrices with only one row, or one column, or one element are sometimes treated specially.

The part of MATLAB that involves computational linear algebra on dense matrices is based on direct adaptations of subroutines from LINPACK and EISPACK [5, 23].

- Σημαντική διευκόλυνση: τα μητρώα αποθηκεύονται ως CSR, αναφερόμαστε σ' αυτά με τον κλασικό τρόπο, `A(i,j)` είναι το στοιχείο στη θέση (i, j) του μητρώου A.
- Οι σημαντικότερες πράξεις (όπως η LU) υλοποιούνται και ως μέθοδοι για μητρώα που ανήκουν στην «αραιή κλάση».

Αραιά μητρώα στη Octave

Για ενδιαφερόμενους Πλήρης παρουσίαση Sparse Matrix Implementation in Octave.

Αραιά μητρώα στην Octave (manual)

- Τα πυκνά μητρώα αποθηκεύονται με διάταξη column major και κατ' αντιστοιχεία, τα αραιά μητρώα σε Compressed Sparse Column format.
- Τα στοιχεία των γραμμών αποθηκεύονται σε αύξουσα σειρά ως προς το δείκτη των γραμμών.
- Ο δείκτης στηλών περιέχει ένα στοιχείο επιπλέον του πλήθους των στηλών. Το πρώτο στοιχείο είναι πάντα 0. Με αυτόν τον τρόπο δεν χρειάζεται ειδική μεταχείριση για την πρώτη ή τελευταία στήλη.

Παράδειγμα Το μητρώο

$$\begin{pmatrix} -1.0 & -2.0 & 0 & 0 \\ 0 & 0 & 0 & -3.0 \\ 0 & 0 & 0 & -4.0 \end{pmatrix}$$

αποθηκεύεται στους πίνακες

cidx = (0, 1, 2, 2, 4); ridx = (0, 0, 1, 2); data = (-1.0, -2.0, -3.0, -4.0).

```
A=rand(1000);B=rand(1000);E=eye(1000);
Z=zeros(1000);Es=speye(1000);Zs=sparse(zeros(1000));
>> whos
Variables in the current scope:

   Attr Name        Size                    Bytes  Class
   ==== ====        ====                    =====  =====
        A        1000x1000                8000000  double
        B        1000x1000                8000000  double
        E        1000x1000                   8000  double
        Es       1000x1000                  16004  double
        Z        1000x1000                8000000  double
        Zs       1000x1000                   4004  double

Total is 4001000 elements using 24028008 bytes
```

Βιβλιογραφία I

T.A. Davis.

Direct Methods for Sparse Linear Systems.

SIAM, Philadelphia, 2006.

T.A. Davis and Y. Hu.

The University of Florida Sparse Matrix Collection.

ACM Trans. Math. Softw., 38(1):1:1--1:25, December 2011.

Y. Saad.

Iterative Methods for Sparse Linear Systems.

SIAM, Philadelphia, 2003.

Ε. Γαλλόπουλος.

Επιστημονικός Υπολογισμός Ι.

Πανεπιστήμιο Πατρών, 2008.

Χρήση Έργου Τρίτων I

1. http://www.netlib.org/clapack/cblas/ (βλ. σελ 9)
2. http://www.mathworks.com/products/matlab/ (βλ. σελ 10)
3. https://www.gnu.org/software/octave/ (βλ. σελ 11)
4. https://www.gnu.org/software/octave/doc/interpreter/index.html#SEC_Contents (βλ. σελ 12)
5. http://www.cise.ufl.edu/research/sparse/matrices/ (βλ. σελ 13)
6. https://www.mathworks.com/help/pdf_doc/otherdocs/simax.pdf (βλ. σελ 21)
7. http://www.sce.carleton.ca/faculty/adler/talks/2006/bateman-adler-octave2006.pdf (βλ. σελ 22)
8. https://www.gnu.org/software/octave/doc/interpreter/Storage-of-Sparse-Matrices.html (βλ. σελ 23)

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών - Ευστράτιος Γαλλόπουλος 2015

``Επιστημονικός Υπολογισμός Ι'', Έκδοση: 1.0, Πάτρα 2013-2014.

Διαθέσιμο από τη δικτυακή διεύθυνση: https://eclass.upatras.gr/courses/CEID1096/

Τέλος Ενότητας

BY NC ND

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης