Επιστημονικός Υπολογισμός Ι

Ενότητα **7** : Διαχείρηση Μητρώων Ειδικής Δομής

Ευστράτιος Γαλλόπουλος
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

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

- Μητρώα ζώνης
- Αραιά μητρώα
- Βιβλιοθήκη LAPACK
- Στοιχεία επαναληπτικών μεθόδων για την επίλυση γραμμικών συστημάτων

Περιεχόμενα

Επανάληψη

Διάλεξη 9/12

- Μέθοδοι ελαχίστων τετραγώνων για την επίλυση γραμμικών συστημάτων
- Επίλυση συστημάτων μητρώα ζώνης και τριδιαγώνια μητρώα
- Επίδραση της οδήγησης
- Βιβλιοθήκη LAPACK

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»

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`nnz` θέσεις μνήμης
- πολυπλοκότερος προγραμματισμός

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

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)

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

	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 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`

’Αλλες 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

Χρησιμοποιώντας ορολογία MATLAB για τη δομή CSR

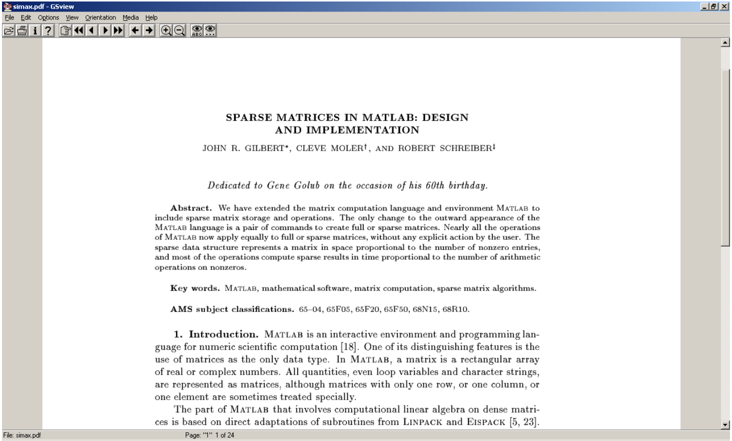
```
function [y] = spmv_csr(VAL,IA,JA,y,x)
% pollaplasiasmos y <- y + A*x opou to A einai se morfh CSR
for i=1:n
    k1=IA(i); k2=IA(i+1)-1;
    y(i) = y(i)+ dot(VAL(k1:k2),x(JA(k1:k2)));
end
```

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
```

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».

Δεν υπάρχει υποδομή για όλα: παραδείγματα και ελλείψεις

```
>>whos A
  Name   Size      Bytes  Class    Attributes
  A      100x100   1592   double   sparse
>>norm(A)
??? Error using ==> norm
Sparse norm(S,2) is not available.
```

```
A=toeplitz([2,-1,zeros(1,998)]); As=sparse(A);
whos A As
  Name      Size           Bytes     Class      Attributes
  A         1000x1000      8000000   double
  As        1000x1000      39980     double     spars
single(As)
??? Error using ==> single
Attempt to convert to unimplemented sparse ty
```

Πολλά ενδιαφέροντα προβλήματα αντιστοιχούν σε μητρώα που είναι

- ... πολύ μεγάλα
- ... ειδικής δομής
- δείτε Sparse Matrix Florida Collection

οι άμεσες μέθοδοι (π.χ. LU, QR) είναι απαγορευτικές

Βασικές περιοχές που ενδιαφέρουν άμεσα στο ΤΜΗΥΠ:

- Μεγάλες προσομοιώσεις, π.χ. κυκλωμάτων
- Ανάκτηση πληροφορίας, π.χ. ειδικές παραγοντοποιήσεις
- Αλγοριθμικά θέματα στο διαδίκτυο, π.χ. βαθμολόγηση ιστοσελίδων (PageRank)
- Μεγάλα προβλήματα δικτύων

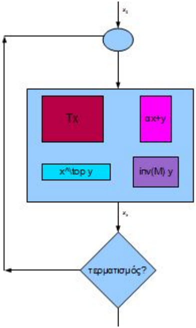
Επαναληπτικές μέθοδοι

Γενικές μέθοδοι επίλυσης στις οποίες επαναλαμβάνεται μία συγκεκριμένη διαδικασία μέχρι να επιτευχθεί ικανοποιητική προσέγγιση της λύσης.

$$x^{(k+1)} = \Phi(k, A, b, x^{(k)}, x^{(k-1)}, \ldots), \text{ για } k = 0, \ldots \text{ δοθέντος αρχικού } x^{(0)}.$$

<ins>Ζητούμενα</ins>: σχεδίαση και υλοποίηση της Φ με τις εξής ιδιότητες

- χαμηλό κόστος (πράξεις, μνήμη) ανά βήμα
- ταχύ ρυθμό προσέγγιση λύσης



Επαναληπτικές μέθοδοι: Χονδρική κατηγοριοποίηση

Κλασικές μέθοδοι: σε κάθε βήμα επιτελείται 1) "γρήγορη επίλυση" με "εύκολο μητρώο" που προέρχεται από το A και 2) διόρθωση της τρέχουσας λύσης, μέχρις ότου ικανοποιηθεί κάποιο κριτήριο τερματισμού.

Μέθοδοι προβολής: σε κάθε βήμα επιχειρείται 1) προσέγγιση της λύσης από κάποιον υπόχωρο του $\mathbb{R}^n$, 2) αλλαγή ή επαύξηση του υπόχωρου, μέχρις ότου ικανοποιηθεί κάποιο κριτήριο τερματισμού.

Κατασκευή κλασικών επαναληπτικών μεθόδων

Για να λύσουμε $Ax = b$

Διαχωρισμός $A = M - N$ (splitting) επιλέγοντας «εύκολα αντιστρέψιμο» M

Επαναλήψεις $Mx^{(k+1)} = Nx^{(k)} + b$ με δοθέν $x^{(0)}$

Κριτήριο τερματισμού με βάση τις τιμές $\|b - Ax^{(k)}\|$, $\|x^{(k+1)} - x^{(k)}\|$, k

Παραδείγματα

Jacobi $A = D - (L + U)$, $D =$ `diag(diag(A))`

Gauss-Seidel $A = (D - L) - U$

Επαναληπτική εκλέπτυνση $A = \tilde{L}\tilde{U} + (A - \tilde{L}\tilde{U})$

Ερωτήματα

- συγκλίνει; ($x^{(k)} \to x$)
- πόσο γρήγορα;
- πώς διαπιστώνουμε τη σύγλιση;
- ποιό είναι το κόστος επανάληψης;
- πως υλοποιείται αποτελεσματικά;

Μερικές φορές η διόρθωση είναι απλή!

Τύπος Sherman-Morrison Αν $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^{n\times 1}$, $v \in \mathbb{R}^{n\times 1}$ τότε

$$(A + uv^\top)^{-1} = A^{-1} - A^{-1}u(1 + v^\top A^{-1}u)^{-1}v^\top A^{-1}$$

Στην επίλυση συστημάτων

$$(A + uv^\top)^{-1}b = A^{-1}b - A^{-1}u(1 + v^\top A^{-1}u)^{-1}v^\top A^{-1}b$$

- Το αντίστροφο της ανανέωσης τάξης-1 ενός μητρώου A μπορεί να γραφτεί συναρτήσει του αντιστρόφου του A.
- Το χρησιμοποιούμε όταν η διαχείριση του A είναι πολύ πιο εύκολη από τη διαχείριση του $B = A + uv^\top$ αν βέβαια γνωρίζουμε τα u, v.
- Σκεφτείτε αν το A ήταν διαγώνιο! ή τριδιαγώνιο και τα u, v τυχαία μητρώα!!!
- Επεκτείνεται και για $k > 1$, όμως καθώς μεγαλώνει το k μεγαλώνει και η επιβάρυνση του υπολογισμού της διόρθωσης ....

Θεώρημα Cayley-Hamilton Αν $p(z) = \det(A - zI) = z^n + \sum_{j=0}^{n-1} \gamma_j z^j$ τότε

$$p(A) \quad = \quad 0 \Rightarrow A^{-1} = -\frac{1}{\gamma_0}\left(A^{n-1} + \gamma_{n-1}A^{n-2} + \cdots \gamma_1\right)$$

Επαναληπτικές μέθοδοι Krylov

Θεώρημα Cayley-Hamilton Αν $p(z) = \det(A - zI) = z^n + \sum_{j=0}^{n-1} \gamma_j z^j$ τότε

$$
\begin{aligned}
p(A) &= 0 \Rightarrow A^{-1} = -\frac{1}{\gamma_0}\left(A^{n-1} + \gamma_{n-1}A^{n-2} + \cdots \gamma_1\right) \\
A^{-1}b &= [b, Ab, \cdots, A^{n-1}b]c \quad \text{για } c = -\frac{1}{\gamma_0}[\gamma_1, \cdots, \gamma_{n-1}, 1]^\top, \text{ μη πρακτικό}
\end{aligned}
$$

ΜΕΘΟΔΟΙ Krylov Αναζητούμε προσέγγιση

$$
\begin{aligned}
x^{(m)} &\in K_m(A;b) = \overbrace{\mathrm{span}\{b, Ab, \ldots, A^{m-1}b\}}^{\text{υπόχωρος Krylov}} \\
&= V_m y^{(m)} \text{ όπου } V_m \text{ ΟΚ βάση του } K_m(A;b)
\end{aligned}
$$

Επαναληπτικές μέθοδοι Krylov

Θεώρημα Cayley-Hamilton Αν $p(z) = \det(A - zI) = z^n + \sum_{j=0}^{n-1} \gamma_j z^j$ τότε

$$\begin{aligned} p(A) &= 0 \Rightarrow A^{-1} = -\frac{1}{\gamma_0}\left(A^{n-1} + \gamma_{n-1}A^{n-2} + \cdots \gamma_1\right) \\ A^{-1}b &= [b, Ab, \cdots, A^{n-1}b]c \quad \text{για } c = -\frac{1}{\gamma_0}[\gamma_1, \cdots, \gamma_{n-1}, 1]^\top, \text{ μη πρακτικό} \end{aligned}$$

ΜΕΘΟΔΟΙ Krylov Αναζητούμε προσέγγιση

$$\begin{aligned} x^{(m)} &\in K_m(A; b) = \overbrace{\operatorname{span}\{b, Ab, \ldots, A^{m-1}b\}}^{\text{υπόχωρος Krylov}} \\ &= V_m y^{(m)} \text{ όπου } V_m \text{ ΟΚ βάση του } K_m(A; b) \end{aligned}$$

Για πλήρη καθορισμό του $x^{(m)}$ περιορίζουμε ώστε

$$r^{(m)} = b - Ax^{(m)} \perp K_m(A; b) \Rightarrow 0 = V_m^\top (b - AV_m y^{(m)})$$

Επαναληπτικές μέθοδοι Krylov

Θεώρημα Cayley-Hamilton Αν $p(z) = \det(A - zI) = z^n + \sum_{j=0}^{n-1} \gamma_j z^j$ τότε

$$\begin{aligned} p(A) &= 0 \Rightarrow A^{-1} = -\frac{1}{\gamma_0}\left(A^{n-1} + \gamma_{n-1}A^{n-2} + \cdots \gamma_1\right) \\ A^{-1}b &= [b, Ab, \cdots, A^{n-1}b]c \quad \text{για } c = -\frac{1}{\gamma_0}[\gamma_1, \cdots, \gamma_{n-1}, 1]^\top, \text{ μη πρακτικό} \end{aligned}$$

ΜΕΘΟΔΟΙ Krylov Αναζητούμε προσέγγιση

$$\begin{aligned} x^{(m)} &\in K_m(A; b) = \overbrace{\operatorname{span}\{b, Ab, \ldots, A^{m-1}b\}}^{\text{υπόχωρος Krylov}} \\ &= V_m y^{(m)} \text{ όπου } V_m \text{ ΟΚ βάση του } K_m(A; b) \end{aligned}$$

Για πλήρη καθορισμό του $x^{(m)}$ περιορίζουμε ώστε

$$r^{(m)} = b - Ax^{(m)} \perp K_m(A; b) \Rightarrow 0 = V_m^\top (b - AV_m y^{(m)})$$

Το αρχικό πρόβλημα $Ax = b$ προσεγγίζεται μέσω επίλυσης συστήματος $m \times m$ (αν αντιστρέψιμο):

$$\boxed{(V_m^\top A V_m) y^{(m)} = V_m^\top b \Rightarrow x^{(m)} = V_m y^{(m)}} \rightarrowtail \text{ ενδιαφέροντα θέματα, βλ. (Saa03)}$$

Αποθήκευση αραιό μητρώο $\rightarrow$ φθηνή αποθήκευση

Δυσκολία προγραμματισμού χρειάζεται ειδικός που εξαρτάται από την αραιή δομή 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.

Μειωμένη απόδοση (MV) με αραιό $A \in \mathbb{R}^{n\times n}$ με 5 περίπου στοιχεία ανά γραμμή (π.χ. `gallery('poisson',N)` όπου $n = \mathtt{N}^2$). Τότε

$$\mu_{\min}^{\text{sparse}} = 4/5$$

Αν ήταν πυκνό, η τιμή για MV θα ήταν $\mu_{\min} = (n^2 + 3n)/2n^2 \approx 1/2$. Δηλαδή, η τοπικότητα είναι μικρότερη λόγω αραιότητας.

Κόστος MV $O(\mathtt{nnz}(A))$

Βιβλιογραφία I

T.A. Davis.
Direct Methods for Sparse Linear Systems.
SIAM, Philadelphia, 2006.

Y. Saad.
Iterative Methods for Sparse Linear Systems.
SIAM, Philadelphia, 2003.

E. Γαλλόπουλος.
Επιστημονικός Υπολογισμός I.
Πανεπιστήμιο Πατρών, 2008.

Χρήση Έργου Τρίτων I

1. https://www.mathworks.com/help/pdf_doc/otherdocs/simax.pdf (βλ. σελ 11)
2. Sparse Matrix Florida Collection (βλ. σελ 13)

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

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