

# Επιστημονικός Υπολογισμός

Ε.Γαλλόπουλος

ΤΜΗΥΠ, Π. Πατρών

Διάλεξη 8: 24 Νοεμβρίου 2017

# Περιεχόμενα I

- 1 Θεμελιώδη προβλήματα της ΥΓΑ
- 2 Επίλυση γραμμικών συστημάτων

**Βασικές πράξεις γραμμικής άλγεβρας** Δίδονται  $A \in \mathbb{R}^{n_1 \times n_3}$ ,  $B \in \mathbb{R}^{n_3 \times n_2}$ ,  $C \in \mathbb{R}^{n_1 \times n_2}$ , να υπολογιστεί  $C + AB$ .

**Επίλυση γραμμικού συστήματος** Δίδονται  $A \in \mathbb{R}^{n \times n}$  και  $b \in \mathbb{R}^n$ . Να υπολογιστεί  $x \in \mathbb{R}^n$  τ.ώ.  $Ax = b$ .

**Γραμμικά ελάχιστα τετράγωνα** Δίδονται  $A \in \mathbb{R}^{m \times n}$  και  $b \in \mathbb{R}^m$ . Να υπολογιστεί  $x \in \mathbb{R}^n$  που επιλύει  $\arg \min_x \|b - Ax\|_2$ .

**Πρόβλημα ιδιοτιμών** Δίδεται  $A \in \mathbb{R}^{n \times n}$ . Να υπολογιστούν τα ιδιοζεύγη  $\lambda \in \mathbb{C}$  και  $x \in \mathbb{C}^n$ :  $Ax = \lambda x$ .

**Γενικευμένο πρόβλημα ιδιοτιμών** Δίδονται  $A, B \in \mathbb{R}^{n \times n}$ . Να υπολογιστούν  $\lambda \in \mathbb{C}$  και  $x \in \mathbb{C}^n$ :  $Ax = \lambda Bx$ .

**Διάσπαση ιδιαζουσών τιμών** Δίδεται  $A \in \mathbb{R}^{m \times n}$ . Να υπολογιστεί διαγώνιο  $\Sigma \in \mathbb{R}^{m \times n}$ , ορθογώνια  $U \in \mathbb{R}^{m \times m}$ ,  $V \in \mathbb{R}^{n \times n}$ :  $A = U\Sigma V^T$ .

**Συναρτήσεις μητρώων** Δίδονται  $A \in \mathbb{R}^{n \times n}$  και συνάρτηση  $f$ . Να υπολογιστεί το  $f(A)$  ή το  $f(A)B$  για  $B \in \mathbb{R}^{n \times m}$  ή το  $Cf(A)B$  για  $C \in \mathbb{R}^{k \times n}$ .

## Book Reviews



Review by B. Parlett, UB Berkeley

### Featured Review

**The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods.** By David S. Watkins. SIAM, Philadelphia, 2007. \$99.00. x+442 pp., softcover. ISBN 978-0-898716-41-2.

**Numerical Methods for General and Structured Eigenvalue Problems.** By Daniel Kressner. Springer, Berlin, 2005. \$99.00. xiv+258 pp., softcover. ISBN 978-3-540-24546-9.

Many years ago Dick Lau, a program director at ONR, said to me, “I support some of you who work in matrix computations and you only seem to tackle three problems: (a) solve  $Ax = b$ , (b) find various least squares solutions, and (c) solve  $Ax = Bx\lambda$ . Why can’t you solve them and go on to more advanced problems?” I did not have the wit to imitate Marvin Minsky<sup>1</sup> and respond, “Are there any other problems?” but replied lamely that while pure mathematicians could invoke military models of pushing back the frontiers of ignorance and had no need to keep reproving their theorems, we who worked in the service of computer users had to tackle the bottlenecks in their computations, and these seemed to fall under one of the three categories above. Although the categories may be the same, what goes on underneath has changed out of all recognition.

## Γενική συνοπτική περιγραφή:

Ορίζονται  $m$  γραμμικές εξισώσεις ως προς  $n$  ανεξάρτητες μεταβλητές,  $x = (\xi_1, \dots, \xi_n)^T$ ,  
ή  $f(x) = b$ , ή

$$f_1(x) = \beta_1, \text{ δηλ. } \alpha_{1,1}\xi_1 + \dots + \alpha_{1,n}\xi_n = \beta_1$$

$$\dots \quad \vdots \quad \dots$$

$$f_m(x) = \beta_m, \text{ δηλ. } \alpha_{m,1}\xi_1 + \dots + \alpha_{m,n}\xi_n = \beta_m$$

και θέλουμε να υπολογίσουμε το  $x$  για το οποίο ικανοποιούνται οι εξισώσεις κατά τον καλύτερο δυνατό τρόπο.

## Γενική συνοπτική περιγραφή:

Ορίζονται  $m$  γραμμικές εξισώσεις ως προς  $n$  ανεξάρτητες μεταβλητές,  $x = (\xi_1, \dots, \xi_n)^\top$ ,  
ή  $f(x) = b$ , ή

$$f_1(x) = \beta_1, \text{ δηλ. } \alpha_{1,1}\xi_1 + \dots + \alpha_{1,n}\xi_n = \beta_1$$

$$\dots \quad \vdots \quad \dots$$

$$f_m(x) = \beta_m, \text{ δηλ. } \alpha_{m,1}\xi_1 + \dots + \alpha_{m,n}\xi_n = \beta_m$$

και θέλουμε να υπολογίσουμε το  $x$  για το οποίο ικανοποιούνται οι εξισώσεις κατά τον καλύτερο δυνατό τρόπο. Συνοψίζοντας τους συντελεστές με το μητρώο  $A$ , διακρίνουμε τις εξής περιπτώσεις:

$$m = n \text{ εύρεση } x \text{ τ.ώ. } Ax = b$$

## Γενική συνοπτική περιγραφή:

Ορίζονται  $m$  γραμμικές εξισώσεις ως προς  $n$  ανεξάρτητες μεταβλητές,  $x = (\xi_1, \dots, \xi_n)^T$ ,  
ή  $f(x) = b$ , ή

$$f_1(x) = \beta_1, \text{ δηλ. } \alpha_{1,1}\xi_1 + \dots + \alpha_{1,n}\xi_n = \beta_1$$

$$\dots \quad \vdots \quad \dots$$

$$f_m(x) = \beta_m, \text{ δηλ. } \alpha_{m,1}\xi_1 + \dots + \alpha_{m,n}\xi_n = \beta_m$$

και θέλουμε να υπολογίσουμε το  $x$  για το οποίο ικανοποιούνται οι εξισώσεις κατά τον καλύτερο δυνατό τρόπο. Συνοψίζοντας τους συντελεστές με το μητρώο  $A$ , διακρίνουμε τις εξής περιπτώσεις:

$m = n$  **εύρεση  $x$  τ.ώ.  $Ax = b \rightarrow$  Κεφ. 5, σημειώσεων ΕΓ. Κεφ. 3 (GV13)**

$m \neq n$  εύρεση/επιλογή βέλτιστου  $x \in \arg \min_{x \in \mathbb{R}^n} \|b - Ax\| \rightarrow$  Κεφ. 6 σημ. ΕΓ.  
Κεφ. 5 (GV13)

Στη συνέχεια θεωρούμε ως γνωστές τις μεθόδους και σχετική συζήτηση των κεφαλαίων του 2ου έτους στο  
“Αριθμητική Ανάλυση και Περιβάλλοντα Υλοποίησης”  
Δείτε τις διαφάνειες και τα σχετικά κεφάλαια από τις αναφορές: (QSG14, Ch. 5)(Moi04, Ch. 2).



# “Μητέρα” των προβλημάτων της Υπολ. ΓΑ

**AGA.1 Δίνεται  $n \times n$  μητρώο  $A$ , διάνυσμα  $b$ ,  $n$  πμών. Υπολογίστε  $x$  τ.ώ.  $Ax = b$ .**

*Problem 3* The solution of simultaneous linear equations. In this problem we are likely to be limited by the storage capacity of the machine. If the coefficients in the equations are essentially random we shall need to be able to store the whole matrix of coefficients and probably also at least one subsidiary matrix. If we have a storage capacity of 6400 numbers we cannot expect to be able to solve equations in more than about 50 unknowns. In practice, however, the majority of problems have very degenerate matrices and we do not need to store anything like as much. For instance problem (2) above can be transformed into one requiring the solution of linear simultaneous equations if we replace the continuum by a lattice. The coefficients in these equations are very systematic and mostly zero. In this problem we should be limited not by the storage required for the matrix of coefficients, but by that required for the solution or for the approximate solutions.

*Problem 4* To calculate the radiation from the open end of a rectangular wave-guide. The complete polar diagram for the radiation could be calculated, together with the reflection coefficient for the end of the guide and interaction coefficients for the various modes. This would be one for any given wave.

A. M. TURING'S ACE REPORT OF 1946

*Problem 5* are polynomials multiplied together to give a polynomial of degree  $n$ .

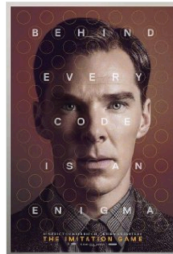
AND OTHER PAPERS

edited by

B. E. Carpenter and R. W. Doran

coefficients multiply the also having the design of

*Problem 6* Given a complicated electrical circuit and the characteristics of its components, the response to given input signals could be calculated. A standard code for the description of the components could easily be devised for this purpose, and also a code for describing connections. There is no



Contact the Filmmakers on IMDbPro »

## Το Παιχνίδι της Μίμησης

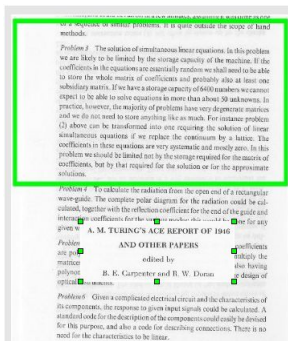
Το Παιχνίδι της Μίμησης είναι βρετανικό/αμερικανικό βιογραφικό, ιστορικό θρίλερ παραγωγής 2014. Η ταινία είναι βασισμένη στη ζωή του Βρετανού μαθηματικού, καθηγητή της λογικής, κρυπτογράφου και πρωτοπόρου επιστήμονα υπολογιστών Άλαν Τούρινγκ. **Βικιπαίδεια**

Ημερομηνία πρώτης κυκλοφορίας: 29 Αυγούστου



# “Μητέρα” των προβλημάτων της Υπολ. ΓΑ

**AGA.1 Δίνεται  $n \times n$  μητρώο  $A$ , διάνυσμα  $b$ ,  $n$  πινών. Υπολογίστε  $x$  τ.ώ.  $Ax = b$ .**



Από P.T. Bonoli, “Requirements for the Centre for Simulation of Wave Plasma Interactions”, 3/2013

**Wave solvers represent electric field in purely spectral (AORSA) or semi-spectral (TORIC) basis functions:**

- AORSA matrix is completely dense and complex with size  $\sim (3 \times N_x \times N_z)^2 \times 16$ , where typically  $(N_x, N_z) \sim (257, 513)$  for a size  $\sim 2.5$  TB.  $[N_x$  and  $N_z$  are the number of spectral modes, assuming axial ( $\phi$ ) symmetry.]
- TORIC & TORLH matrices are block tri-diagonal with dense, complex blocks of size  $\sim 3 \times (3 \times 2 \times N_m) \times N_y \times 16$ , where for  $N_m \sim 1023$  and  $N_y \sim 980$  the size is  $\sim 1.8$  TB.
- Solution is achieved through an LU factorization of the matrix with ScalAPACK, with inversion time scaling as  $(N_z)^3$  and  $N_y \times (N_m)^3$ .

Προσοχή: Ποτέ (σχεδόν) με Cramer. Στη συντριπτική πλειοψηφία των περιπτώσεων ΔΕΝ θέλουμε να υπολογίσουμε το αντίστροφο (ακόμα και αν το γράφουμε ή το ((αναφέρουμε)) ).

Γιατί (1)? Αυτό που ενδιαφέρει είναι να υπολογίσουμε το  $x = A^{-1}B$  ή  $x = CA^{-1}$  ή γενικότερα  $CA^{-1}B$  για δεδομένα  $B, C$ , όπου  $B$  έχει  $n$  γραμμές και το  $C$  έχει  $n$  στήλες.

Γιατί (2)? Όπως αν θέλουμε να υπολογίσουμε το  $\beta/\alpha$  για βαθμωτούς, κάνουμε διαίρεση και ΔΕΝ υπολογίζουμε πρώτα  $\gamma := 1/\alpha$  και μετά  $\gamma\beta$ .

Γιατί (3)? Το  $A^{-1}$  συνήθως δεν έχει κάποια ειδική δομή. Π.χ. αν το τριδιαγώνιο (άρα αποθήκευση σε  $\approx 3n$  θέσεις), το  $A^{-1}$  πυκνό (άρα αποθήκευση σε  $\approx n^2$  θέσεις).

Γιατί (4)? Ο υπολογισμός του  $A^{-1}$  στοιχίζει λίγο περισσότερο από το να λύσουμε ένα γραμμικό σύστημα.

Για  $X = A^{-1}B$  , αν  $B \in \mathbb{R}^{n \times k}$  αρκεί να λύσουμε  $AX = B$ , δηλ. να λύσουμε  $k$  γραμμικά συστήματα με το ίδιο μητρώο συντελεστών οπότε θα αρκέσει μια μόνο παραγοντοποίηση, π.χ.  $LU$ .

Για  $Y = CA^{-1}$  , αν  $C \in \mathbb{R}^{m \times n}$  έχουμε  $YA = C \Rightarrow A^T Y^T = C^T$ , επομένως λύνουμε ως προς  $Y^T$  τα  $m$  γραμμικά συστήματα με το ίδιο μητρώο συντελεστών,  $A^T$ , οπότε πάλι αρκεί μια μόνο παραγοντοποίηση.

Για  $Y = CA^{-1}$  , θα μπορούσαμε να υπολογίσουμε  $Y = C(A^{-1}) = CX$  ή  $Y = (A^{-T} C^T) = Y^T B$ . Η σειρά υπολογισμών που ελαχιστοποιεί τις πράξεις εξαρτάται από τις σχέσεις μεταξύ των  $m, k, n$ .

Προσοχή όμως: Σήμερα υπάρχουν εφαρμογές (στατιστική, ποσοτικοποίηση αβεβαιότητας, δίκτυα) όπου ενδιαφέρουν τα στοιχεία του αντιστρόφου ενός μητρώου, π.χ. όλη η διαγώνιος ή το ίχνος του αντιστρόφου, ή στοιχεία του εκθετικού  $e^A$ .

Για  $X = A^{-1}B$ , αν  $B \in \mathbb{R}^{n \times k}$  αρκεί να λύσουμε  $AX = B$ , δηλ. να λύσουμε  $k$  γραμμικά συστήματα με το ίδιο μητρώο συντελεστών οπότε θα αρκέσει μια μόνο παραγοντοποίηση, π.χ.  $LU$ .

Για  $Y = CA^{-1}$ , αν  $C \in \mathbb{R}^{m \times n}$  έχουμε  $YA = C \Rightarrow A^T Y^T = C^T$ , επομένως λύνουμε ως προς  $Y^T$  τα  $m$  γραμμικά συστήματα με το ίδιο μητρώο συντελεστών,  $A^T$ , οπότε πάλι αρκεί μια μόνο παραγοντοποίηση.

Για  $Y = CA^{-1}$ , θα μπορούσαμε να υπολογίσουμε  $Y = C(A^{-1}) = CX$  ή  $Y = (A^{-T} C^T)^T = Y^T B$ . Η σειρά υπολογισμών που ελαχιστοποιεί τις πράξεις εξαρτάται από τις σχέσεις μεταξύ των  $m, k, n$ .

Προσοχή όμως: Σήμερα υπάρχουν εφαρμογές (στατιστική, ποσοτικοποίηση αβεβαιότητας, δίκτυα) όπου ενδιαφέρουν τα στοιχεία του αντιστρόφου ενός μητρώου, π.χ. όλη η διαγώνιος ή το ίχνος του αντιστρόφου, ή στοιχεία του εκθετικού  $e^A$ .

Παράδειγμα: Στην ανάλυση δικτύων (π.χ. Social Networks) πολλές φορές θέλουμε να υπολογίσουμε το λεγόμενο **σکور του Katz** για οποιουσδήποτε δύο κόμβους  $i, j$  του δικτύου. Αυτό ορίζεται ως

$$k_{i,j} := e_i^T (I - \alpha A)^{-1} e_j$$

όπου  $A$  είναι το μητρώο γεινίασης του γραφήματος (δικτύου) και  $\alpha$  μία παράμετρος (επιλεγμένη έτσι ώστε το  $I - \alpha A$  να είναι αντιστρέψιμο).

# A Domain-Specific Compiler for Linear Algebra Operations

## A Domain-Specific Compiler for Linear Algebra Operations

Diego Fabregat-Traver and Paolo Bientinesi

AICES, RWTH Aachen, Germany  
{fabregat,pauldj}@aices.rwth-aachen.de

**Abstract.** We present a prototypical linear algebra compiler that automatically exploits domain-specific knowledge to generate high-performance algorithms. The input to the compiler is a target equation together with knowledge of both the structure of the problem and the properties of the operands. The output is a variety of high-performance algorithms, and the corresponding source code, to solve the target equation. Our approach consists in the decomposition of the input equation into a sequence of library-supported kernels. Since in general such a decomposition is not unique, our compiler returns not one but a number of algorithms. The potential of the compiler is shown by means of its application to a challenging equation arising within the *genome-wide association study*. As a result, the compiler produces multiple “best” algorithms that outperform the best existing libraries.

We show the potential of the compiler by means of a challenging operation arising in computational biology: the *genome-wide association study* (GWAS), an ubiquitous tool in the fields of genomics and medical genetics [2,3,4]. As part of GWAS, one has to solve the following equation

$$\begin{cases} b_{ij} := (X_i^T M_j^{-1} X_i)^{-1} X_i^T M_j^{-1} y_j & \text{with } 1 \leq i \leq m \\ M_j := h_j \Phi + (1 - h_j) I & \text{and } 1 \leq j \leq t, \end{cases} \quad (1)$$

where  $X_i$ ,  $M_j$ , and  $y_j$  are known quantities, and  $b_{ij}$  is sought after. The size and properties of the operands are as follows:  $b_{ij} \in \mathcal{R}^p$ ,  $X_i \in \mathcal{R}^{n \times p}$  is full rank,  $M_j \in \mathcal{R}^{n \times n}$  is symmetric positive definite (SPD),  $y_j \in \mathcal{R}^n$ ,  $\Phi \in \mathcal{R}^{n \times n}$ , and  $h_j \in \mathcal{R}$ ;  $10^3 \leq n \leq 10^4$ ,  $1 \leq p \leq 20$ ,  $10^6 \leq m \leq 10^7$ , and  $t$  is either 1 or of the order of  $10^5$ .

At the core of GWAS lays a linear regression analysis with non-independent outcomes, carried out through the solution of a two-dimensional sequence of the Generalized Least-Squares problem (GLS)

$$b := (X^T M^{-1} X)^{-1} X^T M^{-1} y. \quad (2)$$

While GLS may be directly solved, for instance, by MATLAB, or may be reduced to a form accepted by LAPACK [5], none of these solutions can exploit the specific structure pertaining to GWAS. The nature of the problem, a sequence of correlated GLSs, allows multiple ways to reuse computation. Also, different sizes of the input operands demand different algorithms to attain high performance in all possible scenarios. The application of our compiler to GWAS, Eq. 1, results in the automatic generation of dozens of algorithms, many of which outperform the current state of the art by a factor of four or more.

# Πολυπλοκότητα επίλυσης γραμμικών συστημάτων

κατά Strassen - ΚΑΙ ΜΕΡΙΚΟΙ ΣΗΜΑΝΤΙΚΟΙ ΤΥΠΟΙ

Παρακάτω υποθέτουμε ότι όποτε απαιτείται αντιστροφή, τα μητρώα που εμπλέκονται είναι αντιστρέψιμα. Έστω η πλοκαδοποίηση του μητρώου με πλοκάδες μεγέθους  $n/2 \times n/2$  και η παραγοντοποίηση block LU:

$$\begin{aligned} A &= \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \\ &= \begin{pmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ 0 & \underbrace{A_{22} - A_{21}A_{11}^{-1}A_{12}}_{\text{συμπλήρωμα Schur}} \end{pmatrix} \end{aligned}$$

$$\begin{aligned} A^{-1} &= \begin{pmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \\ &= \begin{pmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}S^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}S^{-1} \\ -S^{-1}A_{21}A_{11}^{-1} & S^{-1} \end{pmatrix} \end{aligned}$$

Για το αντίστροφο χρειάζονται:

- 1 Το  $A_{11}^{-1}$ ,
- 2 τα  $A_{11}^{-1}A_{12}$  και  $A_{21}A_{11}^{-1}$ ,
- 3 το  $S^{-1} = (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}$ ,
- 4 τα  $-S^{-1}(A_{21}A_{11}^{-1})$ ,  $(A_{11}^{-1}A_{12})S^{-1}$ ,
- 5 και  $A_{11}^{-1} + (A_{11}^{-1}A_{12})S^{-1}(A_{21}A_{11}^{-1})$

Συνολικά

- 2 αντιστροφές, 6 πολλαπλασιασμοί και 2 προσθέσεις, μητρώων μεγέθους  $n/2 \times n/2$ . και
- 6

Επομένως  $T_{\text{INV}}(n) = 2T_{\text{INV}}(n/2) + 6T_{\text{MUL}}(n/2) + 2T_{\text{ADD}}(n/2)$  και έπεται ότι

## Θεώρημα

Αν  $A \in R^{m \times n \times n \times n}$  είναι αντιστρέψιμο, τότε αντιστρέφεται με  $\Omega < 5.64n^{\log 7}$ .

**ΣΥΜΠΕΡΑΣΜΑ:** Αν διαθέτουμε έναν υπερταχύ ( $\Omega = O(n^{2+\mu})$  για  $\mu < 1$ ), αλγόριθμο πολλαπλασιασμού μητρώων, μπορούμε να τον χρησιμοποιήσουμε άμεσα και να κατασκευάσουμε **αλγόριθμο αντιστροφής με ίδια ασυμπτωτική πολυπλοκότητα!**

♣ ΕΝΑΣ ΛΟΓΟΣ ΑΚΟΜΑ ΝΑ ΜΑΣ ΕΝΔΙΑΦΕΡΟΥΝ ΤΑΧΕΙΣ ΑΛΓΟΡΙΘΜΟΙ MM



## Αλγεβρικές ιδιότητες:

- γενικό μητρώο (δηλ. καμία ιδιότητα)
- συμμετρικό ή ερμιτιανό ( $A = A^*$ ), όπου  $A^*$  συμβολίζει το ((συζυγές ανάστροφο)) του  $A$  (αν  $A \in \mathbb{R}^{n \times n}$  προφανώς  $A^T = A^*$ ).
- κανονικό ( $AA^* = A^*A$ )
- ΣΘΟ ( $\forall x \neq 0, x^*Ax > 0$ ).
- πραγματικό ορθογώνιο, μιγαδικό ορθομοναδιαίο ( $A^*A = I$ )

## Δομικές ιδιότητες

- **trigwnik'h, ζώνης, Hessenberg**
- συμμετρική ( $A = A^T$ ), ερμιτιανή ( $A = A^*$ )
- **Toeplitz, Hankel, Vandermonde**, (low displacement rank matrices), ...

### Βάσει του πλήθους μηδενικών/αραιότητας

- πυκνό
- αραιό:  $O(n)$  το πλήθος των μη μηδενικών τιμών για  $n \times n$  μητρώο.

Προσοχή: τη δομή του μητρώου πρέπει να την αξιοποιούμε

- 1 στη μέθοδο επίλυσης,
- 2 στον τρόπο αποθήκευσης (επιλογή δομής δεδομένων),
- 3 στην υλοποίηση της μεθόδου: το πρόγραμμα πρέπει να ανταποκρίνεται στη δομή.

Παρατήρηση: Ορισμένες φορές αν το  $MV$  με το  $A$  μπορεί να εκτελεστεί με πράξεις λιγότερες από τετραγωνικές, συμβαίνει και η λύση του  $Ax = b$  να μπορεί να υπολογιστεί ταχύτερα από το αναμενόμενο.

Παράδειγμα: Μητρώα FFT, circulant, Toeplitz επιδέχονται ταχύτερης διαχείρισης

# ((Νέα)) θέματα !!

## Υπολογιστικά

- 1 Συμπεριφορά ((κλασικών)) αλγορίθμων στο μοντέλο ιεραρχικής μνήμης?
- 2 Τροποποιήσεις για καλύτερη επί/απόδοση υπό το νέο μοντέλο.
- 3 Υλοποιήσεις και Λογισμικό

## Αριθμητικά

- 1 Συμπεριφορά ((κλασικών)) αλγορίθμων στο μοντέλο αριθμητικής.
- 2 Συμπεριφορά ((νέων)) αλγορίθμων στο μοντέλο αριθμητικής.

# Επίλυση γενικών γραμμικών συστημάτων: Αριθμητικές πράξεις και Μεταφορές

Ισχύει ότι:

Μεταφορές:  $\phi_{\min} \approx n^2 + 2n$

Πράξεις:  $\Omega = O(n^{3-\delta})$  όπου συνήθως  $\delta = 0$  ή πολύ μικρό (βλ. ανάλυση πολυπλοκότητας)

$$\Rightarrow \mu_{\min} = \frac{\phi_{\min}}{\Omega} \approx O\left(\frac{1}{n}\right)$$

Εγγενής τοπικότητα  $\Rightarrow$  να υλοποιήσουμε μεθόδους που αξιοποιούν την cache

Κίνητρο:

- $\Omega = \frac{2}{3}n^3 + O(n^2)$
- $\Phi_{\min} = n^2 + n, \mu_{\min} \approx \frac{3}{2n}$
- $\Rightarrow$  ένδειξη τοπικότητας. Πώς την αναδεικνύουμε?

Έστω  $s'$  ummorfoς temaqism'oc tou  $A$  kai twν parag'ontwn  $L, U$

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{pmatrix}, \quad (1)$$

όπου  $A_{11} \in \mathbb{R}^{(k-1)\beta \times (k-1)\beta}$ ,  $A_{22} \in \mathbb{R}^{\beta \times \beta}$  και  $A_{33} \in \mathbb{R}^{(n-k\beta) \times (n-k\beta)}$  ενώ τα  $L, U$  είναι τεμαχισμένα αντιστοίχως. Τότε

$$\begin{pmatrix} L_{11}U_{11} & L_{11}U_{12} & L_{11}U_{13} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} & L_{21}U_{13} + L_{22}U_{23} \\ L_{31}U_{11} & L_{31}U_{12} + L_{32}U_{22} & L_{31}U_{13} + L_{32}U_{23} + L_{33}U_{33} \end{pmatrix},$$

- Αν έχουμε υπολογίσει τις πρώτες  $(k - 1)\beta$  στήλες του  $L$  και τις  $(k - 1)\beta$  πρώτες γραμμές του  $U$ , δηλ. τα  $L_{11}, L_{21}, L_{31}$  και  $U_{11}, U_{12}, U_{13}$ .
- Πως μπορούμε να υπολογίσουμε τις επόμενες  $\beta$  στήλες του  $L$  και  $\beta$  γραμμές του  $U$ , δηλ. τα  $L_{22}, L_{32}$  και  $U_{22}, U_{23}$ .

$$\begin{pmatrix} L_{11}U_{11} & L_{11}U_{12} & L_{11}U_{13} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} & L_{21}U_{13} + L_{22}U_{23} \\ L_{31}U_{11} & L_{31}U_{12} + L_{32}U_{22} & L_{31}U_{13} + L_{32}U_{23} + L_{33}U_{33} \end{pmatrix},$$

Από τον τεμαχισμό ισχύουν τα παρακάτω:

$$A_{22} = L_{21}U_{12} + \boxed{L_{22}U_{22}}$$

$$A_{32} = L_{31}U_{12} + \boxed{L_{32}}U_{22}$$

$$A_{23} = L_{21}U_{13} + L_{22}\boxed{U_{23}}$$

$$A_{33} = L_{31}U_{13} + L_{32}U_{23} + \boxed{L_{33}U_{33}}$$

όπου εντός πλαισίου είναι οι παράγοντες που πρέπει να υπολογιστούν στο παρόν βήμα.

Συγκεντρώνουμε τις εξισώσεις για  $A_{22}$ ,  $A_{32}$  σε μια ομάδα

$$\begin{bmatrix} A_{22} \\ A_{32} \end{bmatrix} = \begin{bmatrix} L_{21} \\ L_{31} \end{bmatrix} U_{12} + \begin{bmatrix} \boxed{L_{22}} \\ \boxed{L_{32}} \end{bmatrix} U_{22}$$

$$A_{23} = L_{21}U_{13} + L_{22}\boxed{U_{23}}$$

$$A_{33} = L_{31}U_{13} + L_{32}U_{23} + \boxed{L_{33}U_{33}}$$

Μπορούμε τώρα να υλοποιήσουμε τις πράξεις χρησιμοποιώντας μόνον BLAS-3 και παραγοντοποίηση LU μητρώων μεγέθους  $\beta \times \beta$ .

$$1) \begin{pmatrix} A_{22} \\ A_{32} \end{pmatrix} - \begin{pmatrix} L_{21} \\ L_{31} \end{pmatrix} U_{12} = \begin{pmatrix} L_{22} \\ L_{32} \end{pmatrix} U_{22} \quad \text{με sGEMM και sGETF2}$$

$$2) A_{23} - L_{21} U_{13} = L_{22} U_{23} \quad \text{με sGEMM και sGETRSM}$$

$$3) \text{ υπολ. } \hat{A}_{33} = A_{33} - L_{31} U_{13} - L_{32} U_{23} \quad \text{με sGEMM}$$

4) LU παραγοντοποίηση του  $\hat{A}_{33}$  επανάληψη των βημάτων 1 ως 4 (αναδρομή)

- sGETF2 Προσεκτική υλοποίηση LU μεγέθους  $\beta \times \beta$  με μερική οδήγηση (χωρίς BLAS-3) στη βιβλιοθήκη LAPACK (επόμενη διάλεξη). Το  $\beta$  επιλέγεται βάσει του μεγέθους της cache.
- ... η sGETF2 αποτελεί βασική ψηφίδα για τη subroutine sGETRF της βιβλιοθήκης LAPACK.

Right-looking και Left-looking αλγόριθμοι: τα παραπάνω εκτελούνται υπολογίζοντας τις επόμενες  $\beta$  στήλες και γραμμές κάθε φορά και ανανεώνοντας το υπομητρώο που βρίσκεται ((κάτω δεξιά)) (right-looking). Εναλλακτικά μπορούμε να οργανώσουμε τον αλγόριθμο να ανανεώνει  $\beta$  στήλες κάθε φορά (left-looking).





G.H. Golub and C.F. Van Loan.

*Matrix Computations.*

Johns Hopkins, 4th edition, 2013.



Cleve B. Moler.

*Numerical computing with MATLAB.*

SIAM, Philadelphia, 2004.



A. Quarteroni, F. Saleri, and P. Gervasio.

*Numerical Methods using MATLAB and Octave.*

Springer, 4th edition, 2014.



G.W. Stewart.

Modifying pivot elements in Gaussian elimination.

*Math.Comp.*, 28(126):537–542, 1974.