



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Επιστημονικός Υπολογισμός Ι

Ενότητα 5 : Επίλυση Γραμμικών Συστημάτων

Ευστράτιος Γαλλόπουλος

Τμήμα Μηχανικών Η/Υ & Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



- 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.



- 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

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

- Επίλυση γραμμικών συστημάτων και εκμετάλλευση ιδιοτήτων του μητρώου
- Παραγοντοποίηση LU με μερική οδήγηση
- Επαναληπτική εκλέπτυνση
- Συμμετρικά θετικά ορισμένα μητρώα
- Παραγοντοποίηση Cholesky και μέθοδος Συζυγών Κλίσεων (CG)

- 1 Υπενθύμιση
- 2 Πρακτικά θέματα, μέθοδοι οδήγησης και διερεύνηση πίσω ευστάθειας της LU
- 3 Μέθοδοι παραγοντοποίησης ειδικών μητρώων
 - Παραγοντοποίηση Cholesky

- Διαδικασία LU και επίλυση συστημάτων με BLAS-3
- Δείκτης κατάστασης προβλήματος \rightarrow δ.κ. μητρώου
- Θεώρημα Rigal-Gaches, εκ των υστέρων πίσω σφάλμα
- πρακτικός υπολογισμός πίσω σφάλματος: Αν $\beta_N = \frac{\|r\|}{\|A\| \|\hat{x}\| + \|b\|}$, και ισχύει $\beta_N \kappa(A) < 1$, τότε

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{2\beta_N \kappa(A)}{1 - \kappa(A)\beta_N}$$

Επισήμανση

Σημαντικές αναφορές για τα παρακάτω είναι τα συγγράμματα των Golub και van Loan ([GV12](#)), του N. Higham ([Hig02](#)) του J. Demmel ([Dem97](#)), και τη μονογραφία των F. Chatelin και V. Frayssse ([CCF96](#)).

Παραδείγματα (υπενθύμιση)

Δεξί μέλος $b = A * \text{ones}(n, 1)$. Το `ferr2` είναι το παραπάνω φράγμα. Λύση μέσω `\` σε MATLAB 7.5.

Μητρώο	n	$\kappa_2(A)$	β_N	$\ x - \tilde{x}\ _2 / \ x\ _2$	<code>ferr2</code>
H	10	1.6e+13	5.0804e-017	2.7571e-004	0.0016
V	10	1.5e+07	3.6797e-017	3.3080e-010	1.1181e-009
R	100	1.7737e+003	2.2492e-016	5.2216e-014	7.9788e-013
Rn	100	684	5.0267e-016	1.3761e-014	6.8694e-013
D	100	1.0e+10	0	0	0
G	60	26.8	0.0156	0.4714	1.4317

H: `hilb(n)` - μεγάλο $\kappa(A)$, μικρό πίσω σφάλμα

V: `vander(linspace(0, 1, 10))` - μεγάλο $\kappa(A)$, μικρό πίσω σφάλμα

R: `rand(n)` - μέτριο $\kappa(A)$, μικρό πίσω σφάλμα

Rn: `randn(n)` - μέτριο $\kappa(A)$, μικρό πίσω σφάλμα

D: `diag([1e-10, ones(1, n-1)])`, μεγάλο $\kappa(A)$, 0 πίσω σφάλμα, κακή εκτίμηση σφάλματος

G: `gfpp(n)` - μικρό $\kappa(A)$, μεγάλο πίσω σφάλμα

Μερικές αδυναμίες των παραπάνω:

- Η δυσκολία άμεσου υπολογισμού του δείκτη κατάστασης ($\mathcal{O}(n^3)$), ενδεχομένως πιο ακριβός από την επίλυση του προβλήματος!
- ... ανάγκη για μεθόδους εκτίμησής του (π.χ. `condest` της MATLAB).
- Η χρήση διαφορετικών νορμών δεν επιφέρει ουσιαστικές αλλαγές στις τιμές (όλες οι νόρμες ισοδύναμες).
- Γενικά, μια αδυναμία των παραπάνω είναι ότι βασίζονται σε πολύ συνοπτικές πληροφορίες, όπως εκφράζονται από τις νόρμες.
- ... μπορούμε να έχουμε πολύ πιο λεπτομερή ανάλυση επεκτείνοντας τις μεθόδους «ανά στοιχείο» (componentwise).
- Σε λογισμικό υψηλής ποιότητας παρέχονται τέτοιες δυνατότητες με φράγματα που προέρχονται από «ανά στοιχείο» ανάλυση.
- Ο κλασικός δείκτης κατάστασης αντικαθίσταται με τον δείκτη κατάστασης κατά Skeel $\| \|A\|A^{-1}\| \|_{\infty}$.

Παρατηρήσεις για το δείκτη κατάστασης

- Η τιμή εξαρτάται από τη νόρμα που χρησιμοποιείται
- ... χονδρικά όμως παραμένει στην ίδια τάξη μεγέθους (Μαθηματική τεκμηρίωση: 'Όλες οι νόρμες είναι «ισοδύναμες»)
- Ο υπολογισμός του δείκτη κατάστασης μπορεί να είναι ακριβός
- ... χρησιμοποιούνται τεχνικές εκτίμησής του

Γενικά ο δείκτης κατάστασης του A ως προς την επίλυση συστήματος πληροφορεί για το πόσα ψηφία ακρίβειας θα μπορούσαν να καθούν κατά την επίλυση.

Χονδρικός κανόνας: Αν γνωρίζουμε το A με σχετική ακρίβεια t δεκαδικών ψηφίων, π.χ. στην αριθμητική διπλής ακρίβειας IEEE το μέγιστο t είναι περίπου 16 και $\kappa(A) \approx 10^p$, τότε η λύση του $Ax = b$ με «πίσω ευσταθή αλγόριθμο» αναμένεται να έχει περί τα $t - p$ σωστά ψηφία.

Προσοχή: Το φράγμα μπορεί να είναι και αρκετά χαλαρό (δηλαδή να καθούν λιγότερα ψηφία). Όμως θα πρέπει ο αλγόριθμος να είναι πίσω ευσταθής. Όταν αυτό δε τυχαίνει, ο κανόνας μπορεί να μην ισχύει και να καθούν πολύ περισσότερα ψηφία.

- Στα παραπάνω δεν λάβαμε καθόλου υπόψη τον αλγόριθμο επίλυσης.
- Πρώτα τρέχαμε τον υπό μελέτη αλγόριθμο και μετά, μέσω του β_N , εξετάζαμε αν ήταν πίσω ευσταθής για τα δεδομένα.
- Ιδανικά, θέλουμε να είμαστε βέβαιοι για την πίσω ευστάθεια ενός αλγορίθμου πριν τον τρέξουμε, δηλ. ανεξάρτητα των δεδομένων.
- Η LU δεν είναι πάντα πίσω ευσταθής αν και (ευτυχώς), σπάνια εμφανίζει μεγάλο πίσω σφάλμα.

Σημαντικό:

- Για να αναλύσουμε την LU πρέπει να προσδιορίσουμε ποιαν υλοποίηση εννοούμε,
- ... τα αποτελέσματα εξαρτώνται από τη μέθοδο οδήγησης!

Παραγοντοποίηση και οδήγηση

Αν το A αντιστρέψιμο τότε υπάρχουν μητρώα μετάθεσης P_1, P_2 , κάτω τριγωνικό L με τη μονάδα στη διαγώνιο και αντιστρέψιμο U ώστε $P_1AP_2 = LU$. Για να είναι εφικτή η παραγοντοποίηση, αρκεί μόνο με ένα από τα P_1, P_2 (τα L, U εξαρτώνται από την επιλογή που κάνουμε για τα P_j).

GE: απλή απαλοιφή χωρίς οδήγηση. Λέγεται επίσης Gaussian elimination with diagonal pivoting γιατί ο οδηγός είναι πάντα το διαγώνιο στοιχείο στην αντίστοιχη θέση. Χρησιμοποιείται μόνον σε ειδικές (όχι όμως σπάνιες) περιπτώσεις, π.χ. μητρώα που είναι ΣΘΟ ή ΔΚ.

GEPP: (Gaussian elimination with partial pivoting) Επιλέγουμε $P_2 = I$ και τον k παράγοντα του P_1 ώστε στο βήμα k να φέρνει το μέγιστο σε απόλυτη τιμή στοιχείο στις θέσεις $(k : n, k)$. Υλοποιείται στην DGETRF της LAPACK.

GECP: (Gaussian elimination with complete pivoting) Επιλέγουμε τους παράγοντες των P_2 και P_1 ώστε στο βήμα k να φέρνουν το μέγιστο σε απόλυτη τιμή στοιχείο στις θέσεις $(k : n, k : n)$.

Rook pivoting: Πιο πρόσφατη μέθοδος οδήγησης που επιτυγχάνει περιορισμένο σφάλμα με μικρότερο κόστος από την GECP.

Για ευκολία θεωρούμε ότι το A έχει ήδη μετασχηματιστεί με μεταθέσεις ώστε να μην χρειάζονται οι εναλλαγές της οδήγησης. Αποδεικνύεται τα υπολογισμένα \hat{L} , \hat{U} ικανοποιούν

$$A + E = \hat{L}\hat{U}$$

όπου

$$|E| \leq \gamma_n |\hat{L}| |\hat{U}|, \quad |||E||| \leq \gamma_n |||\hat{L}||| |||\hat{U}|||$$

Για ευκολία θεωρούμε ότι το A έχει ήδη μετασχηματιστεί με μεταθέσεις ώστε να μην χρειάζονται οι εναλλαγές της οδήγησης. Αποδεικνύεται τα υπολογισμένα \hat{L} , \hat{U} ικανοποιούν

$$A + E = \hat{L}\hat{U}$$

όπου

$$|E| \leq \gamma_n |\hat{L}| |\hat{U}|, \quad \|E\| \leq \gamma_n \|\hat{L}\| \|\hat{U}\|$$

ΠΡΟΣΞΕΤΕ

- 1 Η πίσω ευστάθεια της LU εξαρτάται από το $\frac{\|\Delta A\|}{\|A\|}$. Λέμε ότι έχουμε πίσω ευστάθεια αν $\frac{\|\Delta A\|}{\|A\|} \approx O(u)$.

Για ευκολία θεωρούμε ότι το A έχει ήδη μετασχηματιστεί με μεταθέσεις ώστε να μην χρειάζονται οι εναλλαγές της οδήγησης. Αποδεικνύεται τα υπολογισμένα \hat{L} , \hat{U} ικανοποιούν

$$A + E = \hat{L}\hat{U}$$

όπου

$$|E| \leq \gamma_n |\hat{L}||\hat{U}|, \quad \|E\| \leq \gamma_n \|\hat{L}\| \|\hat{U}\|$$

ΠΡΟΣΞΕΤΕ

- 1 Η πίσω ευστάθεια της LU εξαρτάται από το $\frac{\|\Delta A\|}{\|A\|}$. Λέμε ότι έχουμε πίσω ευστάθεια αν $\frac{\|\Delta A\|}{\|A\|} \approx O(u)$.
- 2 Αυτό ισχύει αν $3\gamma_n \|\hat{L}\| \|\hat{U}\| \approx O(u) \|A\|$
- 3 ... ελέγχεται εύκολα από τις **διαθέσιμες τιμές** \hat{L} , \hat{U} .
- 4 Αν οι τιμές είναι μεγάλες (συνήθως οφείλεται στο $|\hat{U}|$), η LU αστοχεί.

Θεώρημα (Wilkinson)

Έστω $A \in \mathbb{R}^{n \times n}$ και ότι υπολογίζουμε τη λύση \hat{x} του $Ax = b$ μέσω GEPP ή GECR. Τότε υπάρχει ΔA τ.ώ.

$$(A + \Delta A)\hat{x} = b, \quad \|\Delta A\|_{\infty} \leq \rho(n)\rho_n \|A\|_{\infty} \mathbf{u}.$$

όπου $\rho(n)$ είναι κυβικό πολυώνυμο και ο παράγοντας

$$\rho_n = \frac{\max_{i,j,k} |\alpha_{i,j}^{(k)}|}{\max_{i,j} |\alpha_{i,j}|}$$

ονομάζεται **συντελεστής αύξησης της μεθόδου**.

Συνεπώς **μεγάλος συντελεστής δηλώνει ενδεχόμενη μικρή (κακή) πίσω ευστάθεια.**

Ερώτημα Πόσο μεγάλος μπορεί να γίνει ο συντελεστής;
μερικής οδήγησης

$$\rho_n^{\text{ΜΟ}} \leq 2^{n-1}$$

πλήρους οδήγησης

$$\rho_n^{\text{ΠΟ}} < \sqrt{nn^{1+\frac{1}{2}+\dots+\frac{1}{n-1}}} = \mathcal{O}(n^{\log n})$$

- Παρόλα αυτά, η GEPP μπορεί να θεωρηθεί πρακτικά πίσω ευσταθής.
- Για ορισμένα μητρώα όμως, η GEPP δεν είναι πίσω ευσταθής.
- Οι αλγόριθμοι πρέπει να προειδοποιούν για την ύπαρξη τέτοιας δυσκολίας.

Παράδειγμα αποτυχίας της GEPP

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{pmatrix}$$

με GEPP:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 8 \end{pmatrix} \Rightarrow \rho^{\text{MO}} = 8.$$

Παράδειγμα αποτυχίας της GEPP

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{pmatrix}$$

με GEPP:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 8 \end{pmatrix} \Rightarrow \rho^{\text{MO}} = 8.$$

Με GECP:

$$PAQ = \begin{pmatrix} 1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & -2 \end{pmatrix} \\ \Rightarrow \rho^{\text{MO}} = 8.$$

Παρόμοια μορφή δίνουν το χειρότερο συντελεστή για μερική οδήγηση,

$$\rho_n^{MO} = 2^{n-1}.$$

'1.5' γραμμές MATLAB

```
e=[ones(n,1)]; en = [zeros(1,n-1),1]; ...
```

```
A=-tril(e*e',-1)+eye(n)+e*en-en'*en
```

Παρόμοια μητρώα παράγονται με τη συνάρτηση `gfpp` του **MC toolbox**:

```
1 n=64;a=gfpp(n);x=ones(n,1);b=a*x;
2 norm(x-a\b) ans = 3.1623
3 norm(b-a*(a\b)) ans = 177.4458
4 [l,u,rpp]=gePP(a);
5 cond(a) =
6 1.0775e+005
7 rpp = 1.8558e+016
8 condest(a) = 1200
9 [l,u,p,q,rcp]=geCP(a); rcp=1
```

Παρόμοια μορφή δίνουν το χειρότερο συντελεστή για μερική οδήγηση,

$$\rho_n^{\text{MO}} = 2^{n-1}.$$

'1.5' γραμμές MATLAB

```
e=[ones(n,1)]; en = [zeros(1,n-1),1]; ...
```

```
A=-tril(e*e',-1)+eye(n)+e*en-en'*en
```

Παρόμοια μητρώα παράγονται με τη συνάρτηση `gfpp` του **MC toolbox**:

```
1 n=64;a=gfpp(n);x=ones(n,1);b=a*x;
2 norm(x-a\b) ans = 3.1623
3 norm(b-a*(a\b)) ans = 177.4458
4 [l,u,rpp]=gePP(a);
5 cond(a) =
6 1.0775e+005
7 rpp = 1.8558e+016
8 condest(a) = 1200
9 [l,u,p,q,rcp]=geCP(a); rcp=1
```

Σχόλιο Wilkinson `` ... no matrix has yet been discovered for which $\rho_n^{\text{ΠΟ}} > n.$ ''

Ένα αντιπαράδειγμα βρέθηκε (δύσκολα) από τον Gould: Μητρώο 13×13 για το οποίο $\rho^{\text{ΠΟ}} = 13.0205$. Αυτό όμως ήταν μόνο μερικώς σωστό! Περισσότερα (για φανατικούς) διαβάστε στα άρθρα των [άρθρο A. Edelman](#) και των [Higham.](#))

Ειδικά μητρώα

1) Διαγώνια κυρίαρχα, 2) Συμμετρικά θετικά ορισμένα

Υπενθύμιση (Strang σελ. 416, 601)

Θυμηθείτε ορισμούς για ΔΚ και ΣΘΟ. Επίσης το θεώρημα Gerschgorin.

Για ορισμένες κατηγορίες μητρώων μπορούμε να αποφύγουμε οδήγηση. α) π.χ. επειδή το διαγώνιο στοιχείο συμβαίνει και είναι το μέγιστο της στήλης, ή β) γιατί χωρίς να χρησιμοποιήσουμε οδήγηση, ο 'συντελεστής αύξησης' είναι μικρός (π.χ. 1, 2).

Συμμετρικά θετικά ορισμένα (ΣΘΟ): υπάρχει παραγοντοποίηση $A = LL^T$ (Cholesky, δείτε σελ. 421, 430 στον Strang) και ο συντελεστής αύξησης είναι $\rho_n = 1$. ΠΡΟΣΟΧΗ: Το L δεν έχει κατ' ανάγκη 1 στη διαγώνιο.

Διαγώνια κυρίαρχα (ΔΚ): Θεώρημα (Wilkinson): $A \in \mathbb{R}^{n \times n}$ ΔΚ κατά στήλες ή γραμμές και αντιστρέψιμο, τότε υπάρχει $A = LU$ και ο συντελεστής αύξησης είναι $\rho_n \leq 2$. Επομένως, η απαλοιφή Gauss χωρίς οδήγηση είναι πίσω ευσταθής.

Συμμετρικά ΔΚ: υπάρχει παραγοντοποίηση $A = LU$ αν ισχύει αυστηρή ΔΚ.

Γιατί ενδιαφερόμαστε για ΔΚ και ΣΘΟ μητρώα;

Εμφανίζονται σε πάρα πολλές εφαρμογές: Στατιστική, οικονομετρία, σήματα (μητρώα συσχέτισης, μητρώα συμμεταβλητότητας), επίλυση διαφορικών εξισώσεων, κ.λπ.

Γράφουμε τις στήλες του $L = [l_1, l_2, l_3]$, λόγω ΣΘΟ τα διαγώνια στοιχεία του A είναι όλα θετικά.

$$\begin{aligned} A &= LL^T = l_1 l_1^T + l_2 l_2^T + l_3 l_3^T \\ &= \begin{pmatrix} \lambda_{11} \\ \lambda_{12} \\ \lambda_{13} \end{pmatrix} \begin{pmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 \\ \lambda_{22} \\ \lambda_{23} \end{pmatrix} \begin{pmatrix} 0 & \lambda_{22} & \lambda_{23} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \lambda_{33} \end{pmatrix} \begin{pmatrix} 0 & 0 & \lambda_{33} \end{pmatrix} \end{aligned}$$

Γράψαμε το A ως άθροισμα τριών συμμετρικών μητρώων τάξης 1. Προσέξτε τη δομή τους.

Υπολογίζουμε τις στήλες l_j με τη σειρά, ξεκινώντας πάντα από το διαγώνιο στοιχείο της καθεμιάς.

Μερικά βήματα:

- $\alpha_{11} = \lambda_{11}^2$: Επειδή το $\alpha_{11} > 0$ λόγω θετικής ορισμότητας, μπορούμε να επιλέξουμε θετικό $\lambda_{11} = \sqrt{\alpha_{11}}$. Επομένως γνωρίζουμε πλέον το λ_{11} . ΠΡΟΣΟΧΗ: Αν αποτύχει αυτό το βήμα, συμπεραίνουμε ότι το A δεν είναι θετικά ορισμένο και σταματάμε.
- Άμεσα προκύπτει ότι $[\lambda_{12}, \lambda_{13}] = [\alpha_{12}, \alpha_{13}]/\lambda_{11}$. Έχουμε πλέον υπολογίσει όλα τα στοιχεία του l_1 .
- Θέτουμε $A^{(1)} = A - l_1 l_1^T$. Στην πραγματικότητα, οι μόνες πράξεις που χρειάζονται πραγματικά είναι για να υπολογιστούν τα στοιχεία του $A^{(1)}$ στις θέσεις $(2, 2)$, $(2, 3)$, $(3, 3)$ λόγω συμμετρίας και εκ κατασκευής μηδενικής πρώτης στήλης και γραμμής του. Λόγω θεωρίας, το $A^{(1)}$ είναι ΣΘΟ και θα επαναλάβουμε αντίστοιχα βήματα για να υπολογίσουμε το l_2 .

- (συνέχ.) Προσοχή, τώρα το πρόβλημα είναι 2×2 , αφού η μορφή του $A^{(1)}$ είναι πλέον:

$$A^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \alpha_{22}^{(1)} & \alpha_{23}^{(1)} \\ 0 & \alpha_{23}^{(1)} & \alpha_{33}^{(1)} \end{pmatrix}$$

όπου

$$\alpha_{22}^{(1)} = \alpha_{22} - \lambda_{12}^2, \alpha_{23}^{(1)} = \alpha_{32} - \lambda_{13}\lambda_{12}, \alpha_{33}^{(1)} = \alpha_{33} - \lambda_{13}^2$$

- $\alpha_{22}^{(1)} = \lambda_{22}^2 \Rightarrow \lambda_{22} = \sqrt{\alpha_{22}^{(1)}}$, και πάλι αν το $\alpha_{22}^{(1)}$ δεν είναι θετικό, το αρχικό A δεν θα ήταν ΣΘΟ.
- $\lambda_{23} = [\alpha_{23}^{(1)}] / \lambda_{22}$. Έχουμε υπολογίσει όλα τα στοιχεία του l_2 .
- $A^{(2)} = A^{(1)} - l_2 l_2^T$. Στην πραγματικότητα, χρειάζεται να υπολογίσουμε μόνον το στοιχείο στη θέση $(3, 3)$, που είναι $\alpha_{33}^{(2)} = \alpha_{33}^{(1)} - \lambda_{23}^2$.
- $\alpha_{33}^{(2)} = \lambda_{33}^2 \Rightarrow \lambda_{33} = \sqrt{\alpha_{33}^{(2)}}$, και πάλι αν το $\alpha_{33}^{(2)}$ δεν είναι θετικό, το αρχικό A δεν θα ήταν ΣΘΟ.

Θεώρημα

Αν ένα μητρώο $A \in \mathbb{R}^{n \times n}$ και το γράψουμε σε μορφή πλοκάδων

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

τότε το A είναι ΣΘΟ αν και μόνον αν τα A_{11} και το $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ είναι αμφότερα ΣΘΟ.

Αν A ΣΘΟ τότε

- το A είναι αντιστρέψιμο
- κάθε κύριο υπομητρώο είναι ΣΘΟ \Rightarrow αντιστρέψιμο \Rightarrow υπάρχουν L, U ώστε $A = LU$.
- όλα τα στοιχεία της διαγωνίου είναι θετικά (γιατί;)
- Αν εφαρμόσουμε LU χωρίς οδήγηση σε ΣΘΟ μητρώο, ο συντελεστής αύξησης $\rho_n = 1$, επομένως **δεν χρειάζεται οδήγηση.**

Χρήσιμες και σημαντικές ιδιότητες

Εφαρμογή: Μέθοδος κανονικών εξισώσεων

- Αν το A συμμετρικό και ΔΚ και τα στοιχεία της διαγωνίου είναι όλα θετικά, τότε το A είναι ΣΘΟ
- Αν οι στήλες οποιουδήποτε $A \in \mathbb{R}^{m \times n}$ είναι γραμμικά ανεξάρτητες, τότε το μητρώο $A^T A \in \mathbb{R}^{m \times m}$ είναι ΣΘΟ.

Προσοχή: το 2ο αποτέλεσμα είναι κλειδί για μεθόδους επίλυσης **μη τετραγωνικών** συστημάτων με τη **μέθοδο των κανονικών εξισώσεων**:

Έστω ότι $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ όπου $m > n$ και ότι αναζητάμε το $x \in \mathbb{R}^n$ τ.ώ.

$$Ax = b$$

Τότε αν οι στήλες του A είναι γραμμικά ανεξάρτητες,

$$Ax = b \Leftrightarrow A^T Ax = A^T b$$

$$x = (A^T A)^{-1} A^T b$$

Θεώρημα

Έστω $A \in \mathbb{R}^{n \times n}$ ΣΘΟ. Τότε υπάρχει μοναδικό κάτω τριγωνικό L με θετικά διαγώνια στοιχεία, ώστε $A = LL^T$.

Ιδιότητες

- Το L ονομάζεται παράγοντας *Cholesky* του A
- η παραγοντοποίηση $A = R^T R = LL^T$ διάσπαση (ή παραγοντοποίηση) Cholesky.
- Στη MATLAB δείτε την εντολή `chol`.
- Η παραγοντοποίηση Cholesky μπορεί να υλοποιηθεί με πολλούς τρόπους.
- και κατά πλοκάδες με BLAS-3

```
1 chol    Cholesky factorization.
2 chol(A) uses only the diagonal and upper triangle of A.
3 The lower triangle is assumed to be the (complex conjugate)
4 transpose of the upper triangle. If A positive definite,
5 R = chol(A) produces an upper triangular R so that R'*R ...
    = A.
6 If A is not positive definite, an error message is printed.
7
8 L = chol(A, 'lower') uses only the diag. and lower triangle
9 of A to produce a lower triangular L so that L*L' = A. If
10 A is not positive definite, an error message is printed.
11 When A is sparse, this syntax of chol is typically faster.
```

Αλγόριθμος Cholesky

Υλοποίηση βασισμένη σε ανανεώσεις 1ης τάξης. Στο τέλος το κάτω τριγωνικό τμήμα του A περιέχει το L .

Στο 1ο βήμα:

$$\begin{pmatrix} \alpha_{1,1} & v^T \\ v & A_{2:n,2:n} \end{pmatrix} = \begin{pmatrix} \lambda_{1,1} & 0 \\ v/\lambda_{1,1} & I \end{pmatrix} \begin{pmatrix} \lambda_{1,1} & v^T/\lambda_{1,1} \\ 0 & A_{2:n,2:n} - w^T/\alpha_{1,1s} \end{pmatrix}$$

όπου $\lambda_{1,1} = \sqrt{\alpha_{1,1}}$. Συνεχίζουμε αναδρομικά με την παραγοντοποίηση Cholesky του S .

for $k = 1 : n$

$A(k, k) = \text{sqrt}(A(k, k))$

$A(k + 1 : n, k) = A(k + 1 : n, k) / A(k, k)$

for $j = k + 1 : n$

$A(j : n, j) = A(j : n, j) - A(j : n, k)A(j, k)$

end

end

$T_{\text{αρθ}} \approx \frac{n^3}{3}$, δηλ. περίπου το μισό της LU .

Έστω ότι τεμαχίζουμε ως εξής:

$$\begin{pmatrix} A_{1,1} & A_{2,1}^T \\ A_{2,1} & A_{2:n,2:n} \end{pmatrix} = \begin{pmatrix} L_{1,1} & 0 \\ A_{2,1}L_{1,1}^{-1} & I \end{pmatrix} \begin{pmatrix} L_{1,1}^T & L_{1,1}^{-T}A_{2,1}^T \\ 0 & \underbrace{A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{2,1}^T}_S \end{pmatrix}$$

Έστω ότι το $L_{1,1}$ περιέχει τις πρώτες β στήλες του $L_{1,1}$ και ότι έχει ήδη υπολογιστεί. Τότε τα στοιχεία $A_{2,1}L_{1,1}^{-T}$ υπολογίζονται με BLAS-3, το ίδιο και το S το οποίο θα είναι και αυτό ΣΘΟ. Επομένως, μπορούμε να εφαρμόσουμε την ίδια ιδέα αναδρομικά, με πράξεις σε πλοκάδες β στηλών κάθε φορά.

Παρατηρήσεις για την Cholesky

Προσέξτε ότι τα στοιχεία του L είναι φραγμένα σε σύγκριση με του A :

$$\alpha_{ii} = \sum_{j=1}^i \lambda_{ij} \lambda_{ij} = \sum_{j=1}^i \lambda_{ij}^2$$

επομένως $\alpha_{ii} \geq \lambda_{ij}^2$.

- Υπάρχει επίσης και η «παραγοντοποίηση Cholesky χωρίς τετραγωνικές ρίζες»:

$$A = LDL^T,$$

όπου D διαγώνιο και L κάτω τριγ. με μονάδα διαγώνιο.

- Η Cholesky είναι πίσω ευσταθής.

$$\begin{aligned} \|\Delta A\|_\infty &\leq 3\gamma_n \|L\|_\infty \|L^T\|_\infty \\ &\leq 3m\gamma_n \|A\|_\infty \end{aligned}$$

Μέθοδος κανονικών εξισώσεων

Επίλυση με Cholesky

- 1 Υπολογισμός του $C = A^T A$ και του $d = A^T b$. (ΠΡΟΣΟΧΗ: Λόγω συμμετρίας, ο υπολογισμός του C χρειάζεται $\Omega \approx mn^2$ αντί $\Omega \approx 2mn^2$.)
- 2 Παραγοντοποίηση Cholesky $C = LL^T$.
- 3 Επίλυση του $Ly = d$ και του $L^T x = y$.

Το αριθμητικό κόστος της μεθόδου είναι

$$\Omega = mn^2 + n^3/3 + O(n^2)$$

ΧΡΗΣΙΜΟ Θα δούμε σύντομα ότι το άνω φράγμα για το εμπρός σφάλμα της λύσης με κανονικές εξισώσεις καθορίζεται από το **τετράγωνο** του δ.κ. του A , δηλ. $(\kappa(A))^2$. Για βελτίωση της ακρίβειας του παραπάνω συνιστάται η χρήση επαναληπτικής βελτίωσης. Δείτε στο `help qr` στη MATLAB.



The BigChaos Solution to the Netflix (Prize)

Andreas Töschler and Michael Jähres

commendo research & consulting

Neuer Weg 23, A-8580 Köflach, Austria

{andreas.toeschler,michael.jaehres}@commendo.com

Robert M. Bell*

Congratulations!

The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

On September 21, 2009 we awarded the \$1M Grand Prize to team "BigChaos" Pragmatic Chaos. Read about their algorithm, checked team scores on the Leaderboard, and learn more about the Prize on the [FAQ](#).

We apply
request, or
connected:

Input: A matrix P with all previous probe predictions. P always includes a constant predictor column with ones).

- 1 Exclude the Probe ratings r from the training set.
- 2 Initialize the weights.
- 3 $RMSE_{best} = \infty$
- 4 $RMSE_{epoch} = 1000$
- 5 $epochs = 0$
- 6 **while** $RMSE_{epoch} \leq RMSE_{best}$ **do**
- 7 Train one epoch.
- 8 **if** $RMSE_{epoch} < RMSE_{best}$ **then**
- 9 $RMSE_{best} = RMSE_{epoch}$
- 10 Save the current weights.
- 11 **end**
- 12 Predict the probe set \hat{p} .
- 13 Merge current probe prediction \hat{p} and previous predictions: $X = [P \hat{p}]$
- 14 Calculate blending weights: $w = (X^T X)^{-1} X^T r$
- 15 Calculate prediction of the current blend: $p = X \cdot w$
- 16 Calculate the RMSE of the blend: $RMSE_{epoch} = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2}$; r_i is probe rating i ,
is #ratings in the probe set
- 17 $epochs = epochs + 1$
- 18 **end**
- 19 Load the weights.

Παρατηρήσεις για την επίλυση μέσω κανονικών εξισώσεων

Με βάση τα κριτήρια του ΕΥ:

- ταχεία μέθοδος
- ... αλλά μπορεί να εντείνει υπάρχοντα αριθμητικά προβλήματα

Βασική παρατήρηση: Πρόκειται για επίλυση του συστήματος $Bx = b$ όπου $B := A^T A$. Όμως ο δείκτης κατάστασης του προβλήματος «επίλυση κανονικών εξισώσεων» $\kappa_2(B) = \kappa_2^2(A)$.

Γιατί: $A = U\Sigma V^T \Rightarrow A^T A = V\Sigma^T \Sigma V^T$ οπότε $\kappa_2(B) = \left(\frac{\sigma_{\max}(A)}{\min_{\max}(A)} \right)^2$

Ακόμα και αν ο αλγόριθμος επίλυσης είναι πισω ευσταθής, το εμπρός σφάλμα μπορεί να καθοριστεί από το **τετράγωνο** του δείκτη κατάστασης του A .

Παράδειγμα: Αν $\kappa_2(A) = 1\text{e}8$ για $Ax = b$ και χρησιμοποιήσουμε MATLAB, θα «χάσουμε» το πολύ 8 ψηφία (τα μισά της διπλής ακρίβειας IEEE) ενώ στη λύση του $A^T Ax = A^T b$ μπορεί να τα χάσουμε όλα!

Αν ο δείκτης κατάστασης του A είναι μεγάλος, καλύτερα ΝΑ ΜΗ χρησιμοποιούμε κανονικές εξισώσεις αλλά μεθόδους που χρησιμοποιούν ορθογώνιους μετασχηματισμούς

ΣΘΟ μητρώα: Μια (ακόμα) σημαντική επιτυχία αποφοίτου του CEID

Approaching optimality for solving SDD linear systems*

Ioannis Koutis¹ Gary L. Miller² Richard Peng²
Computer Science Department
Carnegie Mellon University
{ioannis.koutis, glmiller, yangp}@cs.cmu.edu

August 4, 2010

Abstract

We present an algorithm that on input of an n -vertex m -edge weighted graph G and a value k , produces an *incremental sparsifier* \tilde{G} with $n - 1 + m/k$ edges, such that the condition number of G with \tilde{G} is bounded above by $\tilde{O}(k \log^2 n)$, with probability $1 - p$. The algorithm runs in time

$$\tilde{O}((m \log n + n \log^2 n) \log(1/p)).$$

As a result, we obtain an algorithm that on input of an $n \times n$ symmetric diagonally dominant matrix A with m non-zero entries and a vector b , computes a vector x satisfying $\|x - A^{-1}b\|_A < \epsilon \|A^{-1}b\|_A$, in expected time

$$\tilde{O}(m \log^2 n \log(1/\epsilon)).$$

The solver is based on repeated applications of the incremental sparsifier that produces a chain of graphs which is then used as input to a recursive preconditioned Chebyshev iteration.

Dr. Dob's
ARCHITECTURE & DESIGN
Now Targeting Multi-core CPUs, PG
ARCHITECTURE & DESIGN
Linear Equation Breakthrough
By Dr. Dob's Staff on October 23, 2010
From a Computer
Using graph theory, randomized algorithms, and linear algebra, researchers at Carnegie Mellon University have an innovative and elegant new algorithm that can solve systems of linear equations that are critical to our modern computer applications as image processing, energy, telecommunications and manufacturing that often may include millions, if not billions, of equations and variables.

CCC Computing Community Consortium
We support the computing research community in creating compelling research visions and their...
HOME YOUR VISION PLANS ACTIVITIES RESOURCES ABOUT CRA
COMPUTING RESEARCH HIGHLIGHT OF THE WEEK (November 19 - 23, 2010)
Speedy Algorithm for Linear Systems
Computer scientists at Carnegie Mellon University have devised an innovative and elegantly concise algorithm that can efficiently solve systems of linear equations that are critical to such important computer applications as image processing, logistics and scheduling problems, and recommendation systems.
The theoretical breakthrough by Professor Gary Miller, Systems Scientist Ioannis Koutis and Ph.D. student Richard Peng, all of Carnegie Mellon's Computer Science Department, has enormous practical potential. Linear systems are widely used to model real-world systems, such as transportation, energy, telecommunications and manufacturing that often may include millions, if not billions, of equations and variables.
The new algorithm, which applies to an important class of problems known as symmetric diagonally dominant (SDD) systems, employs a novel, new basis from graph theory, randomized algorithms and linear algebra. It is so efficient that it may soon be possible for a desktop workstation to solve systems with a billion variables in just a few seconds.
A number of SDD systems exist, but they tend not to work across the broad class of SDD problems and are prone to failure, while the randomized algorithm devised by Miller, Koutis and Peng applies across the spectrum of SDD systems. The team's approach is to first solve a simplified system that can be done rapidly



F. Chaitin-Chatelin and V. Frayssé.

Lectures on Finite Precision Computations.

SIAM, Philadelphia, 1996.



J.W. Demmel.

Applied Numerical Linear Algebra.

SIAM, Philadelphia, 1997.



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

Matrix Computations.

The Johns Hopkins University Press, Baltimore, 3d edition, 2012.



N.J. Higham.

Accuracy and Stability of Numerical Algorithms.

SIAM, Philadelphia, 2nd edition, 2002.



Ε. Γαλλόπουλος.

Επιστημονικός Υπολογισμός I.

Πανεπιστήμιο Πατρών, 2008.

- 1 MC toolbox (βλ. σελ 14-15)
- 2 http://www-math.mit.edu/~edelman/publications/complete_pivoting.pdf (βλ. σελ 15)
- 3 <http://epubs.siam.org/doi/abs/10.1137/0610012> (βλ. σελ 15)
- 4 Mathworks (βλ. σελ 24)
- 5 <http://www.netflixprize.com/> (βλ. σελ 29)
- 6 http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf (βλ. σελ 29)
- 7 <http://dl.acm.org/citation.cfm?id=1918388> (βλ. σελ 31)
- 8 <http://www.drdoobs.com/architecture-and-design/linear-equation-breakthrough/227900457>
(βλ. σελ 31)
- 9 <http://www.cra.org/ccs-old/rh-speedy.php> (βλ. σελ 31)

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



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ