Διερεύνηση μεθόδων για την άμεση επίλυση αραιών γραμμικών συστημάτων

ΜΑΡΗΣ ΓΕΩΡΓΙΟΣ

(ΠΜΣΕ:0443)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων : ΣΑΜΑΡΑΣ ΝΙΚΟΛΑΟΣ, Λέκτορας

Εξεταστής : ΕΥΑΓΓΕΛΙΔΗΣ ΓΕΩΡΓΙΟΣ, Επικ. Καθηγητής

Τμήμα Εφαρμοσμένης Πληροφορικής

Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη

ΣΕΠΤΕΜΒΡΙΟΣ 2005

Πίνακας περιεχομένων

1 Περίληψη

Στα πλαίσια αυτής της διπλωματικής θα πραγματοποιήσουμε μια επισκόπηση και διερεύνηση αλγορίθμων για την άμεση επίλυση αραιών γραμμικών συστημάτων της μορφής $\mathbf{Ax} = \mathbf{b}$. Η χρήση άμεσων μεθόδων έχει ως αποτέλεσμα την επίλυση γραμμικών συστημάτων σε προκαθορισμένο cpu χρόνο και χώρο στη μνήμη και με υψηλή ακρίβεια.

Οι μέθοδοι άμεσης επίλυσης εξετάζονται ως προς ορισμένα χαρακτηριστικά, όπως είναι το fill-in, διατάξεις αραιότητας και πολυπλοκότητα χρόνου. Ιδιαίτερη έμφαση θα δοθεί στην υπολογιστική συμπεριφορά αυτών των μεθόδων καθώς και σε αποτελεσματικούς τρόπους υλοποίησης τους σε Η/Υ. Επίσης, παρουσιάζονται αποτελέσματα σύγκρισης αυτών των μεθόδων σε αραιά και πυκνά γραμμικά προβλήματα.

Summary

In this paper we will review in depth current algorithms for the direct solution of sparse linear systems $\mathbf{Ax} = \mathbf{b}$. The use of direct methods has the advantage of solving linear systems in linear CPU time and memory storage, and increased precision.

The methods for the direct solution are examined regarding certain attributes, such as fill-in, sparsity orderings, and time complexity. Care is given to the computational behavior of these methods and to efficient ways of implementing them in computer systems. The results of the comparison of such methods in dense and sparse problems are also presented.

2 Εισαγωγή

Σε αυτή την εργασία παρουσιάζεται η άμεση επίλυση των γραμμικών συστημάτων

$$\mathbf{Ax} = \mathbf{b} \qquad \text{(Σφάλμα! Δεν έχει οριστεί στυλ.}.1)$$

όπου ο πίνακας συντελεστών **A** είναι μεγάλος και αραιός. Αραιά συστήματα συναντάει κανείς σε πολλά πεδία εφαρμογών. Στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1 απαριθμούνται μερικά από αυτά τα πεδία

Ακουστική	4	Δομική μηχανική	95	Οικονομία	11
Αριθμητική ανάλυση	4	Ελεγχος εναέριας κυκλοφορίας	1	Οκεανογραφία	4
Αστροφυσική	2	Εξομοίωση υπολογιστών	7	Οπτική	1
Βιοχημεία	2	Ηλεκτρική ενέργεια	18	Πεπερασμένα στοιχεία	50
Γραμμικός προγραμματισμός	16	Μηχανική ρευστών	6	Στατιστική	1
Δεδομένα δημοσκοπήσεων	11	Μηχανική του πετρελαίου	19	Χημεία	16
Δημογραφία	3	Μοντελοποίηση αντιδραστήρων	3	Χημική κινητική	14
Δίκτυα Η/Υ	1				

Πίνακας 2.1: Λίστα πεδίων εφαρμογής για τους αραιούς πίνακες Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1

Αυτός ο πίνακας, που προέρχεται από τους Duff, Grimes και Lewis (1989), δείχνει τον αριθμό των πινάκων από κάθε πεδίο εφαρμογής της συλλογής Harwell-Boeing Sparse Matrix Collection.

Ο χαρακτηρισμός ενός αραιού πίνακα ως μεγάλου είναι θέμα που επιδέχεται πολλές απόψεις. Είναι αρκετό να ειπωθεί πως αντιλαμβανόμαστε ένα πίνακα ως μεγάλο

αν δεν μπορεί να παραγοντοποιηθεί αποτελεσματικά χρησιμοποιώντας ένα κώδικα για την επίλυση γενικών γραμμικών συστημάτων όπως ο LAPACK (Anderson et al. 1995). Η τάξη του πίνακα που θεωρείται μεγάλη είναι επομένως μία συνάρτηση του χρόνου και εξαρτάται από την εξέλιξη των κωδίκων επίλυσης αραιών και πυκνών συστημάτων και από την πρόοδο της αρχιτεκτονικής πληροφοριακών συστημάτων. Στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.2 εκθέτεται η τάξη των γενικών μη δομημένων πινάκων που επιλύθηκαν με την χρήση αλγορίθμων επίλυσης αραιών συστημάτων ως συνάρτηση του έτους που επιλύθηκαν.

Έτος	Τάξη
1970	200
1975	1.000
1980	10.000
1985	100.000
1990	250.000
1995	1.000.000

Πίνακας 2.2: Τάξη αραιών πινάκων που μπορούν να επιλυθούν Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.2

Είναι εύκολο να διαπιστώσει κανείς τη πρόοδο που συντελέστηκε τα τελευταία 25 χρόνια στην επίλυση αραιών συστημάτων.

3 Άμεση επίλυση αραιών γραμμικών συστημάτων

3.1 Επίλυση γραμμικού συστήματος εξισώσεων

Πρόκειται για διευκόλυνση συμβολισμού το να αναπαριστά κανείς τον αντίστροφο του πίνακα **A** ως $\mathbf{A}^{-1}$ έτσι ώστε η λύση του συστήματος (**Σφάλμα! Δεν έχει οριστεί στυλ.**.1) να δίνεται από την $\mathbf{A}^{-1}\mathbf{b}$. Ωστόσο δεν υπάρχει σχεδόν καμία περίπτωση όπου είναι χρήσιμο να υπολογισθεί ο αντίστροφος προκειμένου να επιλυθεί το σύστημα. Ακόμα και αν συγκεκριμένες καταχωρήσεις του αντιστρόφου χρειάζονται, παραδείγματος χάρη για ανάλυση ευαισθησίας, υπάρχουν συνήθως υπολογιστικά πολύ πιο αποδοτικές μέθοδοι για να επιτευχθεί αυτό από το να υπολογισθεί ο αντίστροφος.

Οι μέθοδοι που θα παρουσιαστούν μπορεί να φαίνονται σύνθετες, αλλά είναι σημαντικό να θυμάται κανείς πως έχουμε να κάνουμε με μεθόδους που βασίζονται στην απαλοιφή κατά Gauss.

Συγκεκριμένα, οι άμεσες μέθοδοι χρησιμοποιούν μία παραγοντοποίηση του πίνακα συντελεστών για να διευκολύνουν την επίλυση. Η πιο κοινή παραγοντοποίηση για μη συμμετρικά συστήματα είναι η **LU** παραγοντοποίηση όπου ο πίνακας **A** (η καλύτερα μία αντιμετάθεση αυτού) εκφράζεται ως το γινόμενο ενός κάτω τριγωνικού πίνακα **L** και ενός άνω τριγωνικού πίνακα **U**. Έτσι

$$\mathbf{PAQ} = \mathbf{LU} \qquad \text{(Σφάλμα! Δεν έχει οριστεί στυλ.}.1)$$

όπου **P** και **Q** είναι οι πίνακες αντιμετάθεσης. Η παραγοντοποίηση μπορεί να χρησιμοποιηθεί για την επίλυση του συστήματος μέσω των δύο βημάτων

$$\mathbf{Ly} = \mathbf{Pb} \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.2)$$

και

$$\mathbf{Uz} = \mathbf{y} \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.3)$$

οπότε η λύση **x** είναι απλά μία αντιμετάθεση του διανύσματος **z** δηλαδή

$$\mathbf{x} = \mathbf{Qz} \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.4)$$

Όταν ο **A** είναι συμμετρικός, το γεγονός εμφανίζεται στους παράγοντες, και η ανάλυση γίνεται $\mathbf{PAP^T} = \mathbf{LL^T}$ (παραγοντοποίηση Cholesky).

3.2 Ιδιαιτερότητες του προβλήματος

Είναι γνωστό από τη βιβλιογραφία πως η επίλυση της (**Σφάλμα! Δεν έχει οριστεί στυλ.**.1) όπου ο **A**, πίνακας τάξης n, αντιμετωπίζεται ως πυκνός πίνακας απαιτεί $O(n^2)$ αποθηκευτικό χώρο και $O(n^3)$ πράξεις κινητής υποδιαστολής. Εφόσον στην τυπική

περίπτωση τα προβλήματα προς επίλυση είναι τάξεως αρκετών χιλιάδων, μερικές φορές δεκάδων ή και εκατοντάδων χιλιάδων, η χρήση των πυκνών αλγορίθμων γίνεται πολύ γρήγορα αδύνατη λόγω και αποθηκευτικού χώρου και χρόνου επεξεργασίας.

Ο στόχος των αραιών αλγορίθμων είναι η επίλυση των εξισώσεων της μορφής (**Σφάλμα! Δεν έχει οριστεί στυλ.**.1) σε χρόνο και χώρο ανάλογο με $O(n)+O(\tau)$, για ένα πίνακα τάξης *n* με *τ* μη-μηδενικά. Είναι για αυτόν τον λόγο που οι αραιοί κώδικες μπορούν να γίνουν πολύ σύνθετοι. Αν και υπάρχουν περιπτώσεις που αυτός ο στόχος δεν μπορεί να επιτευχθεί, η αποδοτικότητα των αραιών κωδίκων είναι πολύ καλύτερη από αυτή των πυκνών.

Η μελέτη των άμεσων μεθόδων επίλυσης αραιών συστημάτων εμπεριέχει πολλούς προβληματισμούς που εμφανίζονται ευρέως στην υπολογιστική επιστήμη και που δεν είναι τόσο εμφανείς σε περιπτώσεις πραγματικά μεγάλων επιστημονικών κωδίκων.

Οι σημαντικότεροι προβληματισμοί μπορούν να συνοψιστούν στους εξής

1. Μεγάλο μέρος των υπολογισμών είναι αριθμητική ακεραίων
2. Το πρόβλημα διαχείρισης των δεδομένων είναι σημαντικό
3. Ο αποθηκευτικός χώρος είναι συχνά περιοριστικός παράγοντας και πολλές φορές απαιτείται συμπληρωματικός χώρος
4. Ενώ οι εσωτερικοί βρόγχοι είναι συνήθως καλά οριοθετημένοι, ένα σημαντικό μέρος του χρόνου ξοδεύεται σε υπολογισμούς έξω από αυτούς.
5. Οι εσωτερικοί βρόγχοι είναι μερικές φορές εξαιρετικά περίπλοκοι.

Οι προβληματισμοί 1 έως 3 σχετίζονται με τον χειρισμό των διατάξεων αραιότητας. Η αποδοτική υλοποίηση τεχνικών για τον χειρισμό τους είναι κρίσιμης σημασίας στην επίλυση αραιών πινάκων.

Οι προβληματισμοί 2 και 4 δείχνουν την έντονη αντίθεση μεταξύ της αραιής και μη αραιής γραμμικής άλγεβρας. Σε κώδικες μεγάλων πυκνών συστημάτων, πάνω από 98% του χρόνου καταναλώνεται στους εσωτερικούς βρόγχους, αντίθετα με τους αραιούς κώδικες όπου συναντάει κανείς πολύ μικρότερο ποσοστό. Πιο συγκεκριμένα, ο χειρισμός των δεδομένων σχεδόν πάντα απαιτεί έμμεση διευθυνσιοδότηση. Αυτό έχει σαφής επιπτώσεις στην απόδοση, ιδιαίτερα στις παράλληλες αρχιτεκτονικές.

3.3 Φάσεις επίλυσης

Η λύση της (**Σφάλμα! Δεν έχει οριστεί στυλ.**.1) μπορεί να διαιρεθεί σε αρκετές φάσεις. Αν και ο ακριβής διαχωρισμός σε φάσεις εξαρτάται σαφώς από τον αλγόριθμο και την υλοποίησή του που χρησιμοποιείται, μία κοινή διαίρεση είναι η εξής:

1. Μία φάση ανακατάταξης που εκμεταλλεύεται την δομή των δεδομένων, για παράδειγμα, μία ανακατάταξη στην block τριγωνική μορφή (Duff, Erismann και Reid (1986)).
2. Μία φάση ανάλυσης όπου η δομή του πίνακα αναλύεται για να παραχθούν μία κατάλληλη ταξινόμηση και οι δομές δεδομένων για αποδοτική παραγοντοποίηση.
3. Μία φάση παραγοντοποίησης όπου η αριθμητική παραγοντοποίηση εκτελείται.
4. Μία φάση επίλυσης όπου οι παράγοντες χρησιμοποιούνται για να επιλύσουν το σύστημα χρησιμοποιώντας προς τα εμπρός και προς τα πίσω αντικατάσταση.

Κάποιοι κώδικες συνδυάζουν τις φάσεις 2 και 3 έτσι ώστε οι αριθμητικές τιμές να είναι διαθέσιμες όταν η ταξινόμηση ολοκληρώνεται. Η φάση 3 (ή ο συνδυασμός των

φάσεων 2 και 3) συνήθως απαιτεί τον μεγαλύτερο υπολογιστικό χρόνο, ενώ η φάση επίλυσης είναι μία τάξη μεγέθους ταχύτερη.

Αξίζει να σημειωθεί ότι η έννοια μίας ξεχωριστής φάσης παραγοντοποίησης, η οποία μπορεί και να εφαρμόζεται σε διαφορετικό πίνακα από τον αρχικά αναλυθέντα, είναι καινούργια για τα αραιά συστήματα. Στην περίπτωση των πυκνών πινάκων, μόνο η συνδυασμένη ανάλυση και παραγοντοποίηση υπάρχει, ενώ η φάση 1 δεν υπάρχει.

3.4 Δομικά στοιχεία

Ένα κοινό χαρακτηριστικό των σύγχρονων υπολογιστών υψηλών επιδόσεων είναι πως το κύριο εμπόδιο στο να αποκτηθεί η υψηλή επίδοση είναι η καθυστέρηση στο να ληφθούν τα δεδομένα από την κύρια μνήμη στις λειτουργικές μονάδες. Αυτό είναι αληθές είτε αυτές είναι ειδικά κατασκευασμένα ολοκληρωμένα κυκλώματα είτε πρόκειται για RISC επεξεργαστές, PCs βασισμένα στην αρχιτεκτονική x86, vector επεξεργαστές ή παράλληλοι υπολογιστές διαμοιραζόμενης ή κατανεμημένης μνήμης. Οι περισσότεροι υπολογιστές χρησιμοποιούν λανθάνουσα μνήμη ως χώρο υπολογισμών. Τα δεδομένα σε αυτήν την λανθάνουσα μνήμη (πολλοί υπολογιστές μπορούν να έχουν πολλαπλές λανθάνουσες μνήμες οργανωμένες ιεραρχικά, αλλά εδώ η αναφορά γίνεται για το ανώτερο επίπεδο) μπορούν να μεταφερθούν με μικρή υστέρηση και υψηλή διαμεταγωγή στις λειτουργικές μονάδες, αλλά η ποσότητα των δεδομένων που χωράει σε αυτήν είναι αρκετά περιορισμένη (συνήθως κάτω του ενός Megabyte).

Αυτό σημαίνει πως αν πρόκειται να αποκτηθεί κορυφαία επίδοση σχετικά με το θεωρητικό μέγιστο του υπολογιστικού συστήματος, είναι αναγκαίο να επαναχρησιμοποιηθούν τα δεδομένα στην λανθάνουσα μνήμη όσο το δυνατόν περισσότερο για να αμβληθεί το κόστος της μεταφοράς τους από την κύρια μνήμη. Οι

πιο κατάλληλοι και διαδεδομένοι πυρήνες για αυτήν την δουλειά είναι οι BLAS επιπέδου 3 για $O(n^3)$ πράξεις σε πίνακες τάξης n. Υπάρχουν εννέα πυρήνες BLAS επιπέδου 3, αλλά μόνο δύο χρησιμοποιούνται περισσότερο σε ρουτίνες βασισμένες στην LU παραγοντοποίηση: η ρουτίνα πολλαπλασιασμού πίνακα με πίνακα _GEMM και η λύση ενός τμήματος των δεξιών πλευρών ενός τριγωνικού συστήματος, _TSRM, αν και η ρουτίνα συμμετρικής ενημέρωσης, _SYRK, μπορεί να χρησιμοποιηθεί στην συμμετρική παραγοντοποίηση.

Υπολογιστής	Μέγιστο	_GEMM
Meiko CS2-HA	100	88
IBM SP2	266	232
Intel PARAGON	75	68
DEC Turbo Laser	600	450
CRAY C90	952	900
CRAY T3D	150	102

Πίνακας 3.1: Απόδοση του πυρήνα _GEMM σε Mflop/s σε ένα εύρος υπολογιστών. Πίνακες τάξης 500

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1

Στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1 φαίνονται οι επιδόσεις του πυρήνα BLAS επιπέδου 3 _GEMM σε ένα εύρος υπολογιστών με ποικίλες δομές μνήμης και μικροεπεξεργαστές. Σε πολλές περιπτώσεις ο πυρήνας καταφέρνει περίπου 90% ή και περισσότερο από την μέγιστη επίδοση του μικροεπεξεργαστή, και, σε κάθε περίπτωση, πάνω από τα 2/3 της επίδοσης επιτυγχάνεται.

3.5 Παραγοντοποίηση πυκνών πινάκων

Σε κάποιο βαθμό, η ανάπτυξη αλγορίθμων και κώδικα για την αριθμητική άλγεβρα υπήρξε πάντα καθοδηγούμενη από τις εξελίξεις στις αρχιτεκτονικές των υπολογιστών. Η

πρώτη πραγματική βιβλιοθήκη υπορουτίνων για γραμμική άλγεβρα σε πυκνούς πίνακες αναπτύχθηκε στην γλώσσα Algol από τον Wilkinson και Reinsch. Αυτή υπήρξε η βάση για το πρόγραμμα LINPACK όπου ένα μεγάλο εύρος λογισμικού για την επίλυση πυκνών συστημάτων εξισώσεων αναπτύχθηκε σε Fortran και περιγράφεται στο βιβλίο του LINPACK. Ο κώδικας LU παραγοντοποίησης χρησιμοποιήθηκε ως βάση για την βαθμολόγηση των επιδόσεων των υπολογιστών. Οι κώδικες του LINPACK χρησιμοποιούσαν την BLAS επιπέδου 1 και ήταν φορητοί σε ένα μεγάλο εύρος υπολογιστών. Ενώ η BLAS επιπέδου 1 απευθύνεται σε υπολογισμούς με διανύσματα, οι κώδικες LINPACK δεν απέδιδαν ικανοποιητικά σε υπολογιστές με λανθάνουσα μνήμη. Αυτό αντιμετωπίστηκε με την ανάπτυξη του LAPACK. Οι κώδικες σε αυτό το πακέτο χρησιμοποιούσαν BLAS επιπέδου 2 και 3 με πολύ καλύτερα αποτελέσματα στους μοντέρνους υπολογιστές. Ήδη πολλοί κατασκευαστές παράλληλων αρχιτεκτονικών διανείμαν παράλληλες εκδόσεις των κωδίκων BLAS και έτσι, η παράλληλη λειτουργία ήταν εύκολη υπόθεση. Ωστόσο, ο LAPACK δεν σχεδιάστηκε για παράλληλες αρχιτεκτονικές, ιδιαίτερα αυτή της κατανεμημένης μνήμης που χρησιμοποιεί διαβίβαση μηνυμάτων για την επικοινωνία των δεδομένων. Αυτή η τελευταία κατηγορία υπολογιστών βρίσκεται στο επίκεντρο της συνεχιζόμενης προσπάθειας με όνομα ScaLAPACK που υποστηρίζει κατανεμημένη υπολογιστική με εργαλεία όπως το BLACS (*basic linear algebra communications routines*).

3.6 Παραγοντοποίηση αραιών πινάκων

Δύο συγγράμματα εξηγούν την άμεση λύση αραιών γραμμικών συστημάτων εξισώσεων, αυτό των George και Liu (1981) και αυτό των Duff, Erisman και Reid (1987). Το πρώτο περιορίζεται στην περιγραφή των συμμετρικών συστημάτων θετικής διακρίνουσας και

δίνει έμφαση στην θεωρία των γράφων, ενώ το δεύτερο ασχολείται με συμμετρικά και μη συμμετρικά συστήματα και περιέχει σχόλια γύρω από τους αλγόριθμους της Harwell Subroutine Library (HSL). Η HSL έχει σίγουρα τον μεγαλύτερο αριθμό άμεσων αραιών κωδίκων και έχει επίσης και μερικούς κώδικες για την επαναληπτική μέθοδο. Περισσότερες πληροφορίες υπάρχουν στον Ιστό, όπως και επίσης στον οδηγό "*Matrix Market: A Web Resource for Test Matrix Collections*" των Boisvert, Pozo, Remington, Barett και Dongarra.

Κατά την παραγοντοποίηση αραιών πινάκων, είναι κρίσιμης σημασίας η αντιμετάθεση των πινάκων στην (**Σφάλμα! Δεν έχει οριστεί στυλ.**.1). Η επιλογή έτσι ώστε να διατηρηθεί η αραιότητα στους παράγοντες όπως

επίσης να μην κινδυνέψει η αριθμητική σταθερότητα και πολλοί αλγόριθμοι έχουν αναπτυχθεί για αυτόν τον σκοπό. Στην γενική μη συμμετρική περίπτωση, αυτό οδηγεί στην ανάγκη ενός συμβιβασμού της στρατηγικής αριθμητικής οδήγησης έτσι ώστε να επιλεγούν οι οδηγοί που θα περιορίσουν το fill-in. Μία κοινή στρατηγική περιορισμού του fill-in, που επινόησε ο Markowitz, επιλέγει στοιχεία τέτοια ώστε το γινόμενο του αριθμού των καταχωρήσεων στην γραμμή και την στήλη του υποψήφιου οδηγού να ελαχιστοποιηθεί. Ένα στοιχείο επιλέγεται μόνο αν βρίσκεται εντός ενός κατωφλίου του μεγαλύτερου στην στήλη του. Η στρατηγική Markowitz-κατωφλίου αναλύεται λεπτομερώς παρακάτω. Οι δομές δεδομένων σχεδιάζονται έτσι ώστε μόνο τα μη-μηδενικά στοιχεία του πίνακα να καταχωρούνται.

Στην συμμετρική περίπτωση, το ανάλογο του Markowitz είναι ο ελάχιστος βαθμός όπου κάποιος επιλέγει τον οδηγό ώστε να έχει το μικρότερο αριθμό καταχωρήσεων στην γραμμή του. Αυτό το κριτήριο που προτάθηκε το 1967 αντέχει καλά στον χρόνο. Μία

διαφορετική κλάση διατάξεων είναι η ένθετη ανατομή (*nested dissection*) του George (1973). Σύμφωνα με αυτή, μία ομάδα κόμβων επιλέγεται για να χωρίσει τον γράφο, και αυτή η ομάδα τοποθετείται στο τέλος της αλληλουχίας οδήγησης. Οι υπογράφοι που αντιστοιχούν στα τμήματα χωρίζονται και αυτοί και η διαδικασία επαναλαμβάνεται ώστε να αναγνωριστούν οι οδηγοί σε ανάποδη σειρά. Ελάχιστος βαθμός, ένθετη ανατομή και αρκετές ακόμα συμμετρικές διατάξεις περιέχονται στο πακέτο SPARSPAK. Η εμπειρία σε πολλά πειράματα έδειξε πως ο ελάχιστος βαθμός ήταν η καλύτερη μέθοδος αναδιάταξης για γενικά συμμετρικά προβλήματα.

Δεν είναι άμεσα εμφανές πως οι πυρήνες που αναλύθηκαν στην προηγούμενη ενότητα μπορούν να χρησιμοποιηθούν στην παραγοντοποίηση αραιών πινάκων και πράγματι, μεγάλο μέρος της προσπάθειας που έγινε στα τέλη του 70 ήταν στην ακριβώς αντίθετη κατεύθυνση, δηλαδή να εκτελεστούν οι βασικές πράξεις απαλοιφής σε αραιά διανύσματα.

Ο πιο προφανής τρόπος εκμετάλλευσης των πυκνών πυρήνων στην αραιή παραγοντοποίηση είναι η αναδιάταξη του αραιού πίνακα έτσι ώστε τα μη μηδενικά στοιχεία του να βρίσκονται μαζεμένα κοντά στην διαγώνιο (ονομάζεται ελαχιστοποίηση εύρους ζώνης) και μετά η θεώρηση αυτού ως πίνακα ζώνης (*banded matrix*). Ωστόσο αυτό είναι φυσιολογικά άχρηστο διότι ακόμα και ο υψηλός ρυθμός υπολογισμών των BLAS επιπέδου 3 δεν αντισταθμίζουν το επιπλέον έργο. Ένα σχετικό αλλά πιο ελαστικό σχήμα είναι η μετωπική μέθοδος που οφείλει την προέλευσή της στα πεπερασμένα στοιχεία. Όλες αυτές οι τεχνικές απαιτούν ο πίνακας να αναδιαταχθεί ώστε η ζώνη του να είναι στενή.

Μία βασική έννοια της παραγοντοποίησης αραιού πίνακα είναι το δέντρο απαλοιφής (*elimination tree*). Αυτό ορίζεται για κάθε αραιό πίνακα του οποίου η μορφή αραιότητας είναι συμμετρική. Για έναν αραιό πίνακα τάξης n, το δέντρο απαλοιφής είναι ένα δέντρο n κόμβων έτσι ώστε ο κόμβος j να είναι γονέας του κόμβου i αν το στοιχείο $(i, j), j > i$ είναι το πρώτο στοιχείο κάτω από την διαγώνιο στην στήλη i του κάτω τριγωνικού παράγοντα.

Μία προσέγγιση για την χρήση BLAS υψηλότερου επιπέδου σε αραιούς άμεσους επιλυτές είναι μία γενίκευση της παραγοντοποίησης αραιής στήλης BLAS υψηλότερου επιπέδου μπορεί να χρησιμοποιηθεί αν στήλες με μία κοινή μορφή αραιότητας θεωρηθούν μαζί ως ένα ενιαίο τμήμα ή υπερκόμβος και οι αλγόριθμοι ονομάζονται υπερκόμβων στηλών, στηλών-υπερκόμβων ή υπερκόμβων-υπερκόμβων ανάλογα με το αν η πηγή, ο στόχος ή και τα δύο είναι υπερκόμβοι. Οι υπερκόμβοι αναλύονται περισσότερο στο κεφαλαίο 8.

Μία εναλλακτική στην χρήση των υπερκόμβων για την χρήση της BLAS επιπέδου 3 είναι η τεχνική πολλαπλών μετώπων σε συμμετρικούς αραιούς πίνακες. Σε αυτήν την προσέγγιση, οι μη μηδενικές καταχωρίσεις της γραμμής και στήλης οδήγησης κρατώνται στην πρώτη γραμμή και στήλη ενός πυκνού πίνακα και ο υπολογισμός του εξωτερικού γινόμενου σε αυτό το βήμα οδήγησης υπολογίζεται σε αυτόν τον πυκνό υποπίνακα. Ο πυκνός υποπίνακας ονομάζεται μετωπικός. Αν ένας δεύτερος οδηγός μπορεί να επιλεγεί μέσα από τον μετωπικό υποπίνακα, δηλαδή δεν υπάρχουν μη μηδενικές καταχωρήσεις στην γραμμή και στήλη του στον αραιό πίνακα που βρίσκονται εκτός του μετωπικού, τότε οι πράξεις για αυτόν τον οδηγό μπορούν να εκτελεστούν μέσα στον μετωπικό πίνακα. Πολλές φορές αυξάνουμε τεχνητά το μέγεθος του μετωπικού πίνακα ώστε

περισσότερες πράξεις οδήγησης να εκτελούνται σε έναν κόμβο. Έτσι ο πυρήνας ενός σχήματος πολλών μετώπων μπορεί να αναπαρασταθεί από του υπολογισμούς

$$F_{11} = L_1 U_1 \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.5)$$

και

$$F_{22}^{'} \leftarrow F_{22} - F_{21} U_1^{-1} L_1^{-1} F_{12} \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.6)$$

που εκτελούνται στον πυκνό μετωπικό πίνακα

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}. \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.7)$$

Το συμπλήρωμα κατά Schur $F_{22}^{'}$ της (**Σφάλμα! Δεν έχει οριστεί στυλ.**.7) στέλνεται στον γονικό κόμβο στο δέντρο όπου προστίθεται με τις συνεισφορές από τον αρχικό πίνακα και τα άλλα παιδιά για να σχηματίσει ακόμα έναν πυκνό υποπίνακα όπου εκτελούνται παρόμοιες πράξεις. Περισσότερες λεπτομέρειες για την μέθοδο των πολλαπλών μετώπων υπάρχουν στο κεφάλαιο 7.

Είναι πολύ συνηθισμένο να επιλύονται προβλήματα ελαχίστων τετραγώνων με το να σχηματίζονται οι κανονικές εξισώσεις

$$A^T Ax = A^T b$$ (**Σφάλμα! Δεν έχει οριστεί στυλ.**.8)

και να χρησιμοποιείται ένα σχήμα αραιής λύσης για συμμετρικά θετικής διακρίνουσας συστήματα σε αυτές της επαγόμενες εξισώσεις. Υπάρχουν, ωστόσο, άλλες μέθοδοι για την επίλυση του προβλήματος των ελαχίστων τετραγώνων. Η πιο αξιόλογη χρησιμοποιεί την QR παραγοντοποίηση του πίνακα των συντελεστών. Αυτή η παραγοντοποίηση μπορεί επίσης να υλοποιηθεί με τις ίδιες αρχές που εκμεταλλεύεται η μέθοδος των πολλαπλών μετώπων.

4 Εισαγωγή στις μεθόδους άμεσης επίλυσης

Οι μέθοδοι που θα αναλυθούν για την επίλυση αραιών γραμμικών συστημάτων μπορούν να ομαδοποιηθούν σε τέσσερις κύριες κατηγορίες: γενικές μέθοδοι, μετωπικές μέθοδοι (*frontal*), προσεγγίσεις πολλαπλών μετώπων (*multifrontal*), και αλγόριθμοι υπερκόμβων (*supernodal*). Θα γίνει μία εισαγωγή στους αλγόριθμους και τις προσεγγίσεις και θα εξετασθούν κάποιες βασικές πράξεις στους αραιούς πίνακες.

4.1 Οι τέσσερις κυριότερες προσεγγίσεις

Η πρώτη προσέγγιση στην επίλυση αραιών συστημάτων είναι η γενική μέθοδος όπως τυποποιείται από τον κώδικα MA48 της Harwell Subroutine Library (HSL) (Duff και Reid 1996a) ή Y12M (Zlatev, Waśniewski και Schaumburg 1981). Οι κύριες γραμμές αυτής της προσέγγισης είναι πως η αριθμητική οδήγηση και η οδήγηση για την διατήρηση της αραιότητας εκτελούνται ταυτόχρονα και πως οι αραιές δομές δεδομένων χρησιμοποιούνται καθ' όλη τη διάρκεια της εκτέλεσης – ακόμα και στους εσωτερικούς βρόγχους. Αυτά τα χαρακτηριστικά θα θεωρηθούν εμπόδια αναφορικά με την παραλληλότητα της εκτέλεσης. Το θετικό σημείο της γενικής προσέγγισης είναι πως θα δώσει ικανοποιητικά αποτελέσματα σε ένα μεγάλο εύρος δομών και είναι συνήθως η επιλεγόμενη μέθοδος επίλυσης για πολύ αραιά μη δομημένα προβλήματα.

Οι μετωπικές μέθοδοι μπορούν να θεωρηθούν μία επέκταση των μεθόδων ζώνης και μεταβλητής ζώνης και έχουν καλή απόδοση στα συστήματα εκείνα που έχουν μικρό εύρος ζώνης. Η απόδοση τέτοιων μεθόδων για την επίλυση προβλημάτων βασισμένων σε πλέγματα (για παράδειγμα η διακριτοποίηση μερικών διαφορικών εξισώσεων) εξαρτάται ζωτικά στην υποκείμενη γεωμετρία του προβλήματος. Μπορεί βέβαια να γραφούν

μετωπικές μέθοδοι για να επιλυθεί κάθε δυνατό σύστημα. Η διατήρηση της αραιότητας επιτυγχάνεται από μία αρχική διάταξη και η αριθμητική οδήγηση μπορεί να γίνει με αυτήν την διάταξη. Ένα χαρακτηριστικό των μετωπικών μεθόδων είναι πως δεν απαιτείται έμμεση διευθυνσιοδότηση στους εσωτερικούς βρόγχους και έτσι μπορούν να χρησιμοποιηθούν πυρήνες πυκνών πινάκων.

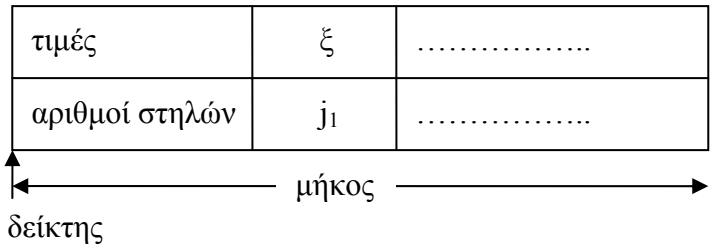
Μία επέκταση των μετωπικών μεθόδων είναι οι μέθοδοι πολλαπλών μετώπων. Η επέκταση αυτή επιτρέπει επιδόσεις για κάθε πίνακα του οποίου το μοτίβο των μη μηδενικών στοιχείων είναι συμμετρικό ή σχεδόν συμμετρικό και επιτρέπει οποιαδήποτε τεχνική διατήρησης της αραιότητας να χρησιμοποιηθεί. Ο περιορισμός σε σχεδόν συμμετρικά συστήματα προέρχεται από το γεγονός ότι η αρχική διάταξη γίνεται σε ένα μοτίβο αραιότητας που αντιστοιχεί στο λογικό άθροισμα των μοτίβων του A και A^T. Αυτή η προσέγγιση, ωστόσο, μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σύστημα.

Ακόμα μία τεχνική για να αποφευχθεί ή να αποσβεστεί το κόστος της έμμεσης διευθυνσιοδότησης είναι να συνδυαστούν οι κόμβοι σε υπερκόμβους.

4.2 Περιγραφή διατάξεων αραιότητας

Στην εισαγωγική αυτή ενότητα περιγράφεται η πιο δημοφιλής διάταξη αραιών δεδομένων, που είναι και αυτή που χρησιμοποιείται από τους περισσότερους γενικής χρήσης κώδικες. Η διάταξη για μία γραμμή του αραιού πίνακα απεικονίζεται στην παρακάτω εικόνα. Όλες οι γραμμές είναι αποθηκευμένες με τον ίδιο τρόπο, με τις πραγματικές τιμές και τον αριθμό στήλης σε δύο πίνακες που έχουν μία σχέση ένα προς ένα, έτσι ώστε η πραγματική τιμή στην θέση k βρίσκεται στην στήλη που δείχνει η καταχώρηση στην θέση k του πίνακα με τους αριθμούς στηλών. Ένας αραιός πίνακας μπορεί έτσι να αποθηκευτεί σαν μια συλλογή από τέτοιες αραιές γραμμές σε δύο

πίνακες: έναν με ακέραιους αριθμούς (αριθμοί στηλών) και έναν με πραγματικούς (πραγματικές τιμές). Ένας τρίτος πίνακας ακεραίων χρησιμοποιείται για να δείξει την θέση κάθε διάταξης αραιής γραμμής σε αυτούς τους δύο πίνακες. Η πρόσβαση στις τιμές μίας γραμμής είναι απλή, αν και απαιτείται έμμεση διευθυνσιοδότηση για να εντοπισθεί ο δείκτης στήλης μίας τιμής.



Εικόνα 4-1: Σχήμα αποθήκευσης για μία γραμμή

Αυτό το σχήμα απεικονίζεται με λεπτομέρεια παρακάτω.

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

Εικόνα 4-2: Ένας αραιός πίνακας 4x4

Αν θεωρηθεί ο πίνακας της εικόνας 4.2, μπορούμε να κρατήσουμε κάθε γραμμή σαν ένα συμπυκνωμένο αραιό διάνυσμα και τον πίνακα σαν μία συλλογή από τέτοια διανύσματα. Για κάθε μέλος της συλλογής, αποθηκεύουμε τον ακέραιο αριθμό στοιχείου της αρχής του και το μήκος του, όπως φαίνεται στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1. Το LEN(i) περιέχει τον αριθμό των στοιχείων στην γραμμή i, ενώ το IPTR(i) περιέχει την θέση στους πίνακες ICN και VALUE της πρώτης καταχώρησης της γραμμής i. Για παράδειγμα, η γραμμή 2 αρχίζει στην θέση 3 και το στοιχείο (2,3) έχει την τιμή 3. Επειδή το μήκος LEN(2) = 2, το τέταρτο στοιχείο ανήκει επίσης στην γραμμή 2, συγκεκριμένα το στοιχείο (2,1) έχει την τιμή -1.

Δείκτες	1	2	3	4	5	6	7
LEN	2	2	1	2			
IPTR	1	3	5	6			
ICN	4	1	3	1	2	2	4
VALUE	4	1	3	-1	2	-2	-4

Πίνακας 4.1: Πίνακας αποθηκευμένος ως συλλογή από αραιές γραμμές

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1

Αξίζει να σημειωθεί πως με αυτό το σχήμα, οι στήλες δεν χρειάζεται να είναι στην σειρά. Αυτό είναι σημαντικό γιατί περισσότερες καταχωρήσεις θα προστεθούν στον πίνακα κατά την διαδικασία απαλοιφής (κάτι που ονομάζεται *fill-in*) και αυτό θα διευκολυνθεί από αυτό το σχήμα.

Παρατηρείται επίσης πως υπάρχει επανάληψη της πληροφορίας με το να αποθηκεύονται ταυτόχρονα τα μήκη των γραμμών (LEN) και οι θέσεις τους (IPTR) όταν οι γραμμές βρίσκονται διατεταγμένες στην σειρά, όπως στον ανωτέρω πίνακα. Στην γενική περίπτωση όμως, οι πράξεις επάνω στους πίνακες καταλήγουν στο να χάνουν την σειρά τους οι γραμμές, οπότε και απαιτούνται και οι δύο πίνακες.

4.3 Χειρισμός διατάξεων αραιότητας

Για να δοθεί μια εικόνα των θεμάτων που εμπλέκονται στον σχεδιασμό αλγόριθμων αραιών πινάκων, εξετάζεται ο χειρισμός των διατάξεων αραιότητας που συμβαίνει συχνότερα καθώς και η LU παραγοντοποίηση. Ο συγκεκριμένος χειρισμός που ενδιαφέρει είναι η πρόσθεση του πολλαπλάσιου μίας γραμμής (η γραμμή-οδηγός, pivot row) του πίνακα στις άλλες γραμμές του πίνακα (non-pivot rows), όταν ο πίνακας είναι αποθηκευμένος στην μορφή που περιγράφθηκε παραπάνω.

Θεωρούμε πως υπάρχει ένας πίνακας ακεραίων, μεγέθους n, έστω IQ, που περιέχει μόνο θετικές τιμές και είναι διαθέσιμος, και πως υπάρχει αρκετός χώρος για να κρατηθεί προσωρινά ένα δεύτερο αντίγραφο της γραμμής-οδηγού σε αραιή μορφή. Θεωρώντας επίσης πως το δεύτερο αντίγραφο της γραμμής-οδηγού έχει γίνει. Τότε μία δυνατότητα υλοποίησης του αλγόριθμου είναι η εξής

1. Διερεύνησε την οδηγό γραμμή, και για κάθε καταχώρηση καθόρισε τον αριθμό στήλης από την τιμή του πίνακα ICN. Θέσε την αντίστοιχη καταχώρηση του πίνακα IQ στο αντίθετο της θέσης αυτής της καταχώρησης μέσα στην συμπυκνωμένη μορφή. Η αρχική τιμή του στοιχείου του πίνακα IQ κρατείται στο δεύτερο αντίγραφο της γραμμής-οδηγού.

Για κάθε γραμμή που δεν είναι η γραμμή-οδηγός, εκτέλεσε τα βήματα 2 και 3

2. Διερεύνησε την γραμμή. Για κάθε αριθμό στήλης, έλεγξε το αντίστοιχο στοιχείο του IQ. Αν είναι θετικό, συνέχισε με το επόμενο στοιχείο στην γραμμή. Αν είναι αρνητικό, άλλαξε το πρόσημό του, και ενημέρωσε την τιμή της αντίστοιχης καταχώρησης στην γραμμή (χρησιμοποιώντας την καταχώρηση από την γραμμή-οδηγό όπως υποδεικνύεται από το αντίθετο την καταχώρησης στον IQ)
3. Διερεύνησε το ανέπαφο αντίγραφο της γραμμής-οδηγού. Αν η αντίστοιχη τιμή του πίνακα IQ είναι θετική, άλλαξε το πρόσημό της. Αν είναι αρνητική, τότε υπάρχει υπερπλήρωση (fill-in) στην γραμμή που δεν είναι η γραμμή-οδηγός. Η νέα καταχώρηση (ένα πολλαπλάσιο της καταχώρησης της γραμμής-οδηγού που υποδεικνύεται από την θετική τιμή του πίνακα IQ)

προστίθεται στο τέλος της γραμμής, και συνεχίζουμε στην επόμενη καταχώρηση της γραμμής-οδηγού.

4. Τέλος, επανάφερε τον πίνακα IQ στις αρχικές τιμές του χρησιμοποιώντας πληροφορίες και από τα δύο αντίγραφα της γραμμής-οδηγού.

Ο αλγόριθμος απεικονίζεται καλύτερα αν αναλύσουμε την κατάσταση πριν από κάθε βήμα.

Πριν το βήμα 1

IQ $i_1 \quad i_2 \quad i_3 \quad i_4 \quad i_5 \quad \ldots\ldots \quad i_{n-1} \quad i_n$

Γραμμή-οδηγός $\begin{matrix} A & \alpha_1 & \alpha_2 & \alpha_2 \\ ICN & j_1 & j_2 & j_3 \end{matrix}$ + δεύτερο αντίγραφο

Κανονική γραμμή $\begin{matrix} A & \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ ICN & j_2 & j_3 & j_4 & j_5 \end{matrix}$

Πριν το βήμα 2

IQ $-2 \quad -1 \quad i_3 \quad -3 \quad i_5 \quad \ldots\ldots \quad i_{n-1} \quad i_n$

Γραμμή-οδηγός Καμία αλλαγή

Αντίγραφο Γ.Ο. $\begin{matrix} A & \alpha_1 & \alpha_2 & \alpha_2 \\ ICN & i_2 & i_1 & i_4 \end{matrix}$

Κανονική γραμμή Καμία αλλαγή

Πριν το βήμα 3

IQ $2 \quad -1 \quad i_3 \quad -3 \quad i_5 \quad \ldots\ldots \quad i_{n-1} \quad i_n$

Γραμμή-οδηγός Καμία αλλαγή

Κανονική γραμμή $\begin{matrix} A & \beta_1 + \zeta\alpha_2 & \beta_2 + \zeta\alpha_3 & \beta_3 & \beta_4 \\ ICN & j_2 & j_3 & j_4 & j_5 \end{matrix}$

Πριν το βήμα 4

IQ	$-2 \quad -1 \quad i_3 \quad -3 \quad i_5 \quad ..... \quad i_{n-1} \quad i_n$
Γραμμή-οδηγός	Καμία αλλαγή
Κανονική γραμμή	$A \quad \beta_1 + \zeta\alpha_2 \quad \beta_2 + \zeta\alpha_3 \quad \beta_3 \quad \beta_4 \quad \zeta\alpha_1$ $ICN \quad j_2 \quad j_3 \quad j_4 \quad j_5 \quad j_1$

Σημειωτέο είναι πως δεν χρειάζεται να τηρούνται οι στήλες των γραμμών στην κανονική τους σειρά όταν χρησιμοποιείται αυτό το σχήμα. Πιο εύκολες στην σύλληψη προσθέσεις διανυσμάτων μπορούν να προκύψουν όταν αυτές βρίσκονται στη σειρά, αλλά το επιπλέον έργο για να διατηρηθούν στη σειρά μπορεί να οδηγήσει σε μείωση της αποδοτικότητας.

Το σημαντικό σημείο σχετικά με αυτόν τον αρκετά περίπλοκο αλγόριθμο που μόλις περιγράφθηκε είναι ότι δεν υπάρχουν διερευνήσεις διανυσμάτων μεγέθους n. Έτσι, άμα αυτοί οι υπολογισμοί εκτελούνται σε κάθε κύριο βήμα της απαλοιφής κατά Gauss σε έναν πίνακα τάξης n, δεν υπάρχουν (εξαιτίας αυτής της πηγής) συνεισφορές της τάξης $O(n^2)$ στο συνολικό έργο. Εφόσον ο στόχος μας στους υπολογισμούς αραιών πινάκων είναι να αναπτύξουμε αλγόριθμους που είναι γραμμικοί με την τάξη του πίνακα και τον αριθμό των μη-μηδενικών στοιχείων, οι υπολογισμοί $O(n^2)$ είναι καταστροφικοί και θα κυριαρχήσουν την επίλυση αν ο n είναι αρκετά μεγάλος.

Πρόσθετα με την αποφυγή τέτοιων διερευνήσεων και επιτρέποντας τους δείκτες να τηρούνται με οποιαδήποτε σειρά, κανένας πίνακας πραγματικών αριθμών μεγέθους n δεν απαιτείται. Πραγματικά, ο πίνακας IQ μπορεί να χρησιμοποιηθεί για οποιονδήποτε άλλο σκοπό, με μόνη απαίτηση τα στοιχεία του να είναι όλα θετικά. Ένα επιπλέον πλεονέκτημα είναι πως δεν ελέγχουμε ποτέ κανένα στοιχείο για την τιμή μηδέν (που θα

ήταν αναμενόμενο αν κρατούσαμε ένα πλήρες διάνυσμα μεγέθους n), οπότε δεν γεννάται σύγχυση μεταξύ μηδενικών που τηρούνται ρητά και μηδενικών που δεν υπάρχουν στην αραιή διάταξη. Αυτό το πλεονέκτημα μπορεί να είναι πολύ σημαντικό κατά την επίλυση πολλών συστημάτων με κοινή δομή αλλά με διαφορετικές αριθμητικές τιμές.

5 Γενικές μέθοδοι επίλυσης

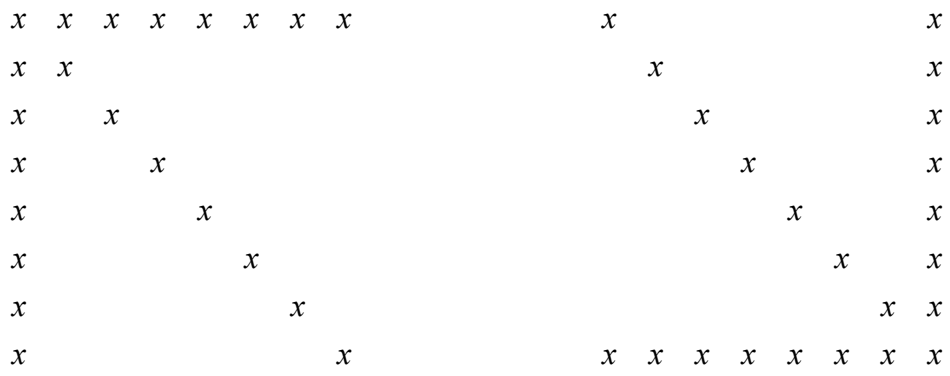
5.1 Fill-in και ταξινόμηση αραιότητας

Ένας σημαντικός προβληματισμός όταν ο πίνακας **A** είναι αραιός είναι ότι οι παράγοντες **L** και **U** θα είναι γενικά πιο πυκνοί από τον αρχικό **A**.

Μετά από *k* βήματα απαλοιφής σε έναν πίνακα τάξης *n*, ο μειωμένος πίνακας είναι ο κατώτερος *n-k* επί *n-k* πίνακας αλλαγμένος σε σχέση με τον αρχικό πίνακα σύμφωνα με τα πρώτα *k* βήματα οδήγησης. Αν συμβολίσουμε τα στοιχεία του αρχικού πίνακα με $\alpha_{\mathrm{ij}}^{(1)}$ και αυτούς του μειωμένου πίνακα μετά από *k* βήματα απαλοιφής κατά Gauss με $\alpha_{\mathrm{ij}}^{(\mathrm{k}+1)}$, τότε fill-in προκαλείται αν, στην βασική πράξη

$$\alpha_{\mathrm{ij}}^{(\mathrm{k}+1)} \leftarrow a_{ij}^{(k)} - a_{ik}^{(k)}\left[a_{kk}^{(k)}\right]^{-1} a_{kj}^{(k)}, \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.1)$$

το στοιχείο στην θέση *(i,j)* του αρχικού **A** είναι μηδέν. Η σειρά των γραμμών και των στηλών του **A** μπορεί να είναι σημαντική για την διατήρηση της αραιότητας των παραγόντων. Το παρακάτω σχήμα δίνει το παράδειγμα μιας περίπτωσης όπου η αναδιάταξη των γραμμών και των στηλών για την διατήρηση της αραιότητας κατά την απαλοιφή κατά Gauss είναι ιδιαίτερα αποτελεσματική. Αν τα στοιχεία-οδηγοί επιλέγονται από την διαγώνιο στην φυσική τους σειρά, ο αναδιατεταγμένος πίνακας διατηρεί όλα τα μηδενικά κατά την παραγοντοποίηση, ενώ στην αρχική σειρά δεν διατηρεί κανένα.



Εικόνα 5-1: Αρχικός και αναδιατεταγμένος πίνακας

Ο πίνακας Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1 δείχνει τα οφέλη που μπορεί να καρπωθεί κάποιος στην αραιή περίπτωση αγνοώντας όλες ή σχεδόν όλες τις μηδενικές καταχωρήσεις στον αρχικό πίνακα και τους μετέπειτα υπολογισμούς. Η δεύτερη γραμμή του πίνακα δίνει στοιχεία για αραιή απαλοιφή χωρίς καμία αναδιάταξη εκμετάλλευσης αραιότητας, ενώ η τρίτη γραμμή δείχνει πως επιπλέον σημαντικά κέρδη μπορούν να γίνουν όταν πραγματοποιούνται αναδιατάξεις που εκμεταλλεύονται την αραιότητα. Σε μεγαλύτερα προβλήματα, εξαιτίας της $O(n^2)$ και $O(n^3)$ πολυπλοκότητας των πυκνών κωδίκων για χρόνο και χώρο αντίστοιχα, μπορούμε να περιμένουμε ακόμα μεγαλύτερα οφέλη.

Διεργασία	Συνολικός χώρος (Kwords)	Πράξεις (Mflops)	Χρόνος (secs)
Χειρισμός συστήματος ως πυκνό	4084	5503	34.5
Αποθήκευση και εργασία μόνο στα μη-μηδενικά στοιχεία	71	1073	3.4
Οδήγηση με εκμετάλλευση αραιότητας	14	42	0.9

Πίνακας 5.1: Οφέλη από την αραιότητα σε πίνακα τάξης 2021 με 7353 μη-μηδενικά στοιχεία Π**Σφάλμα! Δεν έχει**

οριστεί

στυλ..1

Ακόμη μία διαφορά μεταξύ πυκνών και αραιών συστημάτων εντοπίζεται άμα αναλογισθεί κανείς την κοινή περίπτωση όπου μία επόμενη λύση απαιτείται σε έναν πίνακα που έχει την ίδια διάταξη αραιότητας με τον αρχικό. Ενώ αυτό δεν έχει καμία επίδραση στην πυκνή περίπτωση, η επιρροή στην αραιή είναι σημαντική μιας και οι πληροφορίες από την πρώτη παραγοντοποίηση μπορούν να χρησιμοποιηθούν για να απλουστεύσουν την δεύτερη. Πραγματικά, συχνά η αναδιάταξη των δεδομένων μπορεί να γίνει πριν εκτελεστεί οποιαδήποτε αριθμητική παραγοντοποίηση.

Τα πλεονεκτήματα της αναδιάταξης των δεδομένων για την διατήρηση της αραιότητας έγινε εμφανές στην εικόνα 5.1. Μία απλή αλλά αποτελεσματική στρατηγική για να διατηρηθεί η αραιότητα οφείλεται στον Markowitz (1957). Σε κάθε βήμα της απαλοιφής κατά Gauss, επιλέγεται ως στοιχείο οδηγός το μη-μηδενικό στοιχείο του εναπομένοντος μειωμένου πίνακα με το μικρότερο γινόμενο του αριθμού των καταχωρημένων στοιχείων στην γραμμή του με τον αριθμό των καταχωρημένων στοιχείων στην στήλη του.

Πιο συγκεκριμένα, πριν από το *k*-στο κύριο βήμα της απαλοιφής κατά Gauss, έστω πως $r_i^{(k)}$ δείχνει τον αριθμό των καταχωρήσεων στην γραμμή i του μειωμένου $(n-k+1)\times(n-k+1)$ υποπίνακα και $c_j^{(k)}$ ο αριθμός των καταχωρήσεων στην στήλη j. Το κριτήριο κατά Markowitz επιλέγει την καταχώρηση $\alpha_{ij}^{(k)}$ του μειωμένου υποπίνακα που ελαχιστοποιεί το γινόμενο

$$\left(r_i^{(k)}-1\right)\left(c_j^{(k)}-1\right), \qquad \textbf{(Σφάλμα!}$$

Δεν έχει

οριστεί

στυλ..2)

όπου το $\alpha_{ij}^{(k)}$ ικανοποιεί και κάποιο αριθμητικό κριτήριο.

Αυτή η στρατηγική μπορεί να ερμηνευτεί με πολλούς τρόπους, για παράδειγμα, διαλέγοντας τον οδηγό που τροποποιεί τον μικρότερο πλήθος συντελεστών στον απομένοντα υποπίνακα. Μπορεί επίσης να θεωρηθεί ως επιλογή του στοιχείου οδηγού που απαιτεί τον μικρότερο αριθμό πολλαπλασιασμών και διαιρέσεων. Τέλος η (**Σφάλμα! Δεν έχει οριστεί στυλ.**.2) μπορεί να θεωρηθεί ένας τρόπος περιορισμού του fill-in σε αυτό το στάδιο, γιατί θα ήταν ίση με το fill-in αν όλα τα $\left(r_i^{(k)}-1\right)\left(c_j^{(k)}-1\right)$ τροποποιημένα στοιχεία ήταν προηγουμένως μηδέν.

Γενικά, για την στρατηγική αναδιάταξης κατά Markowitz, είναι καλό να λαμβάνονται κάποια μέτρα ώστε να διατηρηθεί η αριθμητική σταθερότητα. Συγκεκριμένα, περιορίζεται η επιλογή σύμφωνα με την (**Σφάλμα! Δεν έχει οριστεί στυλ.**.2) σε αυτά τα στοιχεία που ικανοποιούν την ανισότητα

$$\left|a_{kk}^{(k)}\right| \geq u\left|a_{ik}^{(k)}\right|, \quad i \geq k\,, \qquad \text{(Σφάλμα! Δεν έχει οριστεί στυλ.}.3)$$

όπου το u είναι ένα προκαθορισμένο κατώφλι στο εύρος $0 < u \leq 1$.

Βλέποντας πίσω στην (**Σφάλμα! Δεν έχει οριστεί στυλ.**.1) είναι εμφανές πως ο ρόλος του u είναι να περιορίσει την μέγιστη δυνατή αύξηση της αριθμητικής τιμής μίας

καταχώρησης του πίνακα σε ένα βήμα της απαλοιφής κατά Gauss στο $(1+1/u)$, και επομένως επηρεάζει την σταθερότητα της παραγοντοποίησης.

Αυτό το αποτέλεσμα απεικονίζεται στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.2, και από αυτόν μπορεί να εξαχθεί το συμπέρασμα πως ένας καλός συμβιβασμός μεταξύ σταθερότητας και ελευθερίας επιλογής στοιχείου οδηγού κατά την αναδιάταξη υπάρχει για την τιμή 0.1 της παραμέτρου u.

u	Καταχωρήσεις στους παράγοντες	Σφάλμα στην λύση
1.00	16767	3.00E-09
0.25	14249	6.00E-10
0.10	13660	4.00E-09
0.01	15045	1.00E-05
1.00E-04	16198	1.00E+02
1.00E-10	16553	3.00E+23

Πίνακας 5.2: Αποτέλεσμα της διακύμανσης της παραμέτρου u

(πίνακας τάξης 541 με 4285 καταχωρήσεις)

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.2

Αξίζει να παρατηρηθεί πως με ολοένα μικρότερες τιμές της παραμέτρου u ο αριθμός των καταχωρήσεων στους παράγοντες αυξάνονται. Αυτό μπορεί να αντιβαίνει την κοινή λογική, αλλά εξηγείται εύκολα αν αναλογισθεί κανείς πως προκαλείται από την δυσκολία του να βρεθούν καλά στοιχεία οδηγοί στα μετέπειτα βήματα λόγω κακών επιλογών που έγιναν στα πρώτα βήματα.

Σε περίπτωση που δεν ενδιαφέρει η ακριβής παραγοντοποίηση, θυσιάζεται η ακρίβεια προκειμένου να αυξηθεί η αραιότητα όχι μέσω της παραμέτρου u, αλλά μέσω της ανοχής απόρριψης tol. Καταχωρήσεις που συναντιούνται κατά την παραγοντοποίηση με τιμή χαμηλότερη της tol, η χαμηλότερη της tol πολλαπλασιασμένης με την

μεγαλύτερη τιμή στην γραμμή ή στήλη, απορρίπτονται από την διάταξη, και αποκτάται μία μη ακριβής ή μερική παραγοντοποίηση του πίνακα.

5.2 Σύγκριση με πυκνούς κώδικες

Κάθε φορά που υπάρχουν βελτιώσεις στους αλγόριθμους ή στις δυνατότητες των υπολογιστικών συστημάτων που ανεβάζουν αισθητά την απόδοση των κωδίκων επίλυσης πυκνών πινάκων, εμφανίζονται κάποιοι που υποστηρίζουν πως το μέλλον των αραιών και άμεσων αλγορίθμων είναι μελανό, και πως οι πυκνοί επιλυτές πρέπει να χρησιμοποιούνται πλέον χωρίς διάκριση και για τα αραιά συστήματα. Αυτή τους η θέση είναι εσφαλμένη για δύο λόγους.

Πρώτα από όλα, ενώ η απόδοση των πυκνών κωδίκων είναι εντυπωσιακή, ακόμα και για συντηρητικές τιμές της τάξης n του πίνακα, η $O(n^3)$ πολυπλοκότητα ενός πυκνού επιλυτή κάνει τους υπολογισμούς απαγορευτικά χρονοβόρους, εάν βέβαια οι απαιτήσεις σε αποθηκευτικό χώρο της τάξης $O(n^2)$ δεν καθιστούν την επίλυση αδύνατη. Εκτός από την θεωρητική αυτή παρατήρηση, μια πολύ πιο τρανταχτή απόκρουση της θέσης έρχεται από τους τρέχοντες άμεσους αραιούς επιλυτές όπως ο κώδικας MA48 της βιβλιοθήκης HSL. Από τα αποτελέσματα στον πίνακα της σελίδας 36 βλέπουμε πως ο MA48 αποδίδει πολύ καλύτερα από ότι ο κώδικας SGESV της βιβλιοθήκης LAPACK σε ένα CRAY Y-MP ακόμα και για αραιούς πίνακες αρκετά συντηρητικής τάξης. Είναι δεκτό πως η απόδοση του MA48 εξαρτάται σημαντικά από την δομή του αραιού πίνακα, όπως φαίνεται από δύο εκτελέσεις σε δύο διαφορετικούς πίνακες τάξης 1224, ενώ ο κώδικας LAPACK εξαρτάται μόνο από την τάξη και εμφανίζει την αναμενόμενη $O(n^3)$ συμπεριφορά. Παρόλα αυτά, ακόμα και στην χειρότερη περίπτωση, ο MA48 υπερτερεί

του SGESV με άνεση. Πρέπει να τονιστεί επίσης πως οι μετέπειτα παραγοντοποιήσεις πινάκων που έχουν την ίδια δομή είναι πολύ πιο γρήγορες στον αραιό κώδικα, για παράδειγμα ο πίνακας BCSSTK27 μπορεί να παραγοντοποιηθεί ξανά από τον MA48 σε μόλις 0.33 δευτερόλεπτα, χρησιμοποιώντας την ίδια αλληλουχία στοιχείων οδηγών όπως και πριν.

Κατά δεύτερον, η απόρριψη των πυκνών κωδίκων για την επίλυση αραιών συστημάτων προέρχεται από το γεγονός ότι ο αλγόριθμος MA48 παρακολουθεί την πυκνότητα του μειωμένου πίνακα όσο προχωράει η απαλοιφή και γυρνάει σε πυκνό τρόπο επίλυσης. Έτσι, πυκνοί επιλυτές υψηλής απόδοσης μπορούν να ενσωματωθούν μέσα στον αραιό κώδικα έτσι ώστε οι πυκνοί επιλυτές να μην μπορούν ποτέ να υπερνικήσουν την απόδοση των αραιών. Αυτό βέβαια είναι σε ένα βαθμό μόνο σωστό, μιας και υπάρχει επιπλέον κόστος κατά την μετάβαση από την αραιή στην πυκνή επίλυση, αλλά τα πειράματα έδειξαν πως το κατώφλι μετάβασης ως προς την πυκνότητα για την ολική ελαχιστοποίηση του χρόνου μπορεί συχνά να είναι χαμηλό (τυπικά 20% πυκνότητα) και πως κέρδη της τάξης του τετραπλάσιου μπορούν να αποκομισθούν ακόμα και με απλούς πυκνούς κώδικες (Duff 1984).

Πίνακας	Τάξη	Καταχωρήσεις	MA48	SGESV
FS 980 3	680	2646	0.06	0.96
PORES 2	1224	9613	0.54	4.54
BCSSTK27	1224	56126	2.07	4.55
NNC1374	1374	8606	0.70	6.19
WEST2021	2021	7353	0.21	18.88
ORSREG 1	2205	14133	2.65	24.25
ORAN1678	2529	90158	1.17	36.37

Πίνακας 5.3: Σύγκριση μεταξύ του MA48 και του LAPACK (SGESV) σε ένα εύρος πινάκων του Harwell-Boeing Sparse Matrix Collection.

Οι χρόνοι είναι επίλυση σε δευτερόλεπτα σε έναν επεξεργαστή ενός CRAY Y-MP

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.3

5.3 Άλλες προσεγγίσεις

Μέχρι τώρα δόθηκε έμφαση στην προσέγγιση άμεσης επίλυσης αραιών γραμμικών συστημάτων όπως υποδεικνύονται από τους κώδικες MA48 (Duff, Reid 1996a) και Y12M (Zlatev 1981). Ενώ αυτοί οι κώδικες είναι πιθανώς οι πιο διαδεδομένοι για την επίλυση γενικών αραιών συστημάτων όπως αυτά προκύπτουν σε ένα ευρύ φάσμα πεδίου εφαρμογής, υπάρχουν και άλλες αλγοριθμικές προσεγγίσεις και κώδικοι για την επίλυση μη συμμετρικών συστημάτων.

Μία τεχνική είναι να προηγείται μία αναδιάταξη των γραμμών και έπειτα να επιλέγονται τα στοιχεία οδηγοί από κάθε γραμμή στην σειρά, πρώτα ενημερώνοντας την γραμμή οδηγό σύμφωνα με τα προηγούμενα βήματα οδήγησης και μετά επιλέγοντας το στοιχείο οδηγό χρησιμοποιώντας ένα κριτήριο κατωφλίου στο κατάλληλο μέρος της ενημερωμένης γραμμής. Αν οι στήλες είναι και αυτές προδιατεταγμένες κατάλληλα για την διατήρηση της αραιότητας, τότε γίνεται πρώτα μία απόπειρα για να διαπιστωθεί αν στοιχεία της διαγωνίου της αναδιατεταγμένης μορφής είναι κατάλληλα. Αυτή η προσέγγιση χρησιμοποιείται από τους κώδικες NSPFAC και NSPIV του Sherman

(1978). Η προσέγγιση αυτή μοιάζει με τις διαδοχικές παραγοντοποιήσεις που αναλύθηκαν παραπάνω και είναι εύκολη στον προγραμματισμό της. Μπορεί όμως να υποστεί πλήγμα από το fill-in αν δεν βρεθεί μία καλή αρχική αναδιάταξη ή αν η αριθμητική οδήγηση απαγορεύει την παραμονή κοντά σε αυτήν την αναδιάταξη.

Μία άλλη προσέγγιση είναι να παραχθεί μία διάταξη δεδομένων που, με επιλεγμένη μία διάταξη στηλών, εμπεριέχει όλες τις επιλογές των στοιχείων οδηγών (George και Ng 1985). Είναι αξιοπρόσεκτο ότι αυτό μερικές φορές δεν είναι υπερβολικά απαιτητικό σε αποθηκευτικό χώρο και έχει το πλεονέκτημα μία καλής σταθερότητας αλλά μέσα σε μία επακολούθως στατική διάταξη δεδομένων. Υπάρχουν βέβαια περιπτώσεις όπου οι υπερβάλλουσες απαιτήσεις σε αποθηκευτικό χώρο είναι υψηλές.

Μέθοδοι που βασίζονται στην αραιή QR παραγοντοποίηση μπορούν να χρησιμοποιηθούν επίσης για τα γενικά μη συμμετρικά συστήματα. Αυτές βασίζονται στην δουλειά του George και Heath (1980) και πρώτα δημιουργούν την δομή του R μέσω συμβολικής παραγοντοποίησης την δομής του πίνακα των κανονικών εξισώσεων $A^t A$. Είναι κοινό να μην κρατιέται ο Q αλλά να επιλύεται το σύστημα χρησιμοποιώντας τις ημι-κανονικές εξισώσεις

$$R^T Rx = A^T b,$$

με διαδοχικές βελτιώσεις να γίνονται προκειμένου να αποφευχθούν αριθμητικά προβλήματα. Μία QR παραγοντοποίηση μπορεί φυσικά να χρησιμοποιηθεί για το πρόβλημα των ελαχίστων τετραγώνων και υλοποιήσεις έχουν αναπτυχθεί για τον σκοπό αυτό.

Μία άλλη προσέγγιση ειδικά σχεδιασμένη για τους μη συμμετρικούς πίνακες έχει αναπτυχθεί από τον Davis και Yew (1990). Εδώ μία ομάδα από στοιχεία οδηγούς

επιλέγεται ταυτόχρονα με το κριτήριο του Markowitz και το κριτήριο αριθμητικής σταθερότητας έτσι ώστε αν γίνει μία αντιμετάθεση της ομάδας στο άνω τμήμα του πίνακα, το αντίστοιχο τμήμα του πίνακα θα είναι διαγώνιο και όλες οι πράξεις που αντιστοιχούν σε αυτά τα στοιχεία οδηγούς μπορούν να εκτελεστούν ταυτόχρονα. Είναι δυνατό να σχεδιαστεί ένας αλγόριθμος που θα διενεργεί την αναζήτηση οδηγών σε παράλληλη επεξεργασία. Επακόλουθες ομάδες από ανεξάρτητους οδηγούς επιλέγονται με όμοιο τρόπο μέχρι ο μειωμένος πίνακας να γίνει αρκετά πυκνός ώστε να γίνει η αλλαγή σε πυκνό κώδικα. Αντίστοιχο σχήμα προτείνει ο Alaghband (1980).

6 Μετωπικές μέθοδοι

6.1 Εισαγωγή

Οι μετωπικές μέθοδοι έχουν τις ρίζες τους στην επίλυση προβλημάτων πεπερασμένων στοιχείων από την δομική ανάλυση. Ένα από τα πρώτα προγράμματα που υλοποίησαν την μετωπική μέθοδο είναι αυτό του Irons (1970). Η υλοποίησή του έχει να κάνει μόνο με την περίπτωση των συμμετρικών θετικής διακρίνουσας πινάκων. Αυτή η μέθοδος μπορεί να επεκταθεί και στους μη συμμετρικούς πίνακες (Hood 1976) και το πεδίο εφαρμογής της δεν περιορίζεται μόνο στα πεπερασμένα στοιχεία.

Η κλασσική προσέγγιση για την περιγραφή των μετωπικών μεθόδων είναι να θεωρηθεί η εφαρμογή της στα προβλήματα πεπερασμένων στοιχείων όπου ο πίνακας A εκφράζεται ως το σύνολο των συνεισφορών των στοιχείων μίας δομής πεπερασμένων στοιχείων. Συγκεκριμένα

$$A = \sum_{l=1}^{m} A^{[l]}\,, \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.1)$$

όπου ο $A^{[l]}$ είναι μη μηδενικός μόνο για τις γραμμές και στήλες εκείνες που αντιστοιχούν στις μεταβλητές στο πεπερασμένο στοιχείο *l*. Αν a_{ij} και $a_{ij}^{[l]}$ αναπαριστούν την (i,j) καταχώρηση του A και $A^{[l]}$ αντίστοιχα, η βασική πράξη σύστασης του πίνακα A είναι στην μορφή

$$a_{ij} \Leftarrow a_{ij} + a_{ij}^{[l]}. \qquad \textbf{(Σφάλμα!}$$

Δεν έχει

οριστεί

στυλ..2)

Είναι προφανές πως η βασική πράξη της απαλοιφής κατά Gauss

$$a_{ij} \Leftarrow a_{ij} - a_{ip}\left[a_{pp}\right]^{-1} a_{pj} \qquad \textbf{(Σφάλμα!}$$

Δεν έχει

οριστεί

στυλ..3)

μπορεί να εκτελεστεί μόλις όλοι οι όροι του τριπλού γινομένου της (**Σφάλμα! Δεν έχει οριστεί στυλ.**.3) είναι πλήρως αθροισμένοι (δηλαδή δεν εμπλέκονται πλέον στις αθροίσεις της μορφής (**Σφάλμα! Δεν έχει οριστεί στυλ.**.2)). Η σύσταση του πίνακα και η απαλοιφή κατά Gauss μπορούν λοιπόν να συνδυαστούν και η σύσταση του Α δεν ολοκληρώνεται ποτέ ρητά. Αυτό επιτρέπει στους ενδιάμεσους μηχανισμούς να εκτελεστούν σε έναν πυκνό πίνακα, επονομαζόμενος μετωπικός (*frontal*) πίνακας, του οποίου οι γραμμές και οι στήλες αντιστοιχούν στις μεταβλητές που δεν έχουν ακόμα εξαλειφθεί αλλά εμφανίζονται σε τουλάχιστον ένα από τα πεπερασμένα στοιχεία που έχουν συσταθεί στον Α.

Για προβλήματα που δεν αφορούν πεπερασμένα στοιχεία, οι γραμμές του Α (εξισώσεις) προστίθενται στον μετωπικό πίνακα μία-μία. Μία μεταβλητή αντιμετωπίζεται ως πλήρως αθροισμένη όταν η εξίσωση στην οποία εμφανίζεται για τελευταία φορά έχει συσταθεί στον Α. Ο μετωπικός πίνακας θα είναι σε αυτήν την περίπτωση ορθογώνιος. Μία πλήρης περιγραφή της εισόδου των εξισώσεων μπορεί να βρεθεί στον Duff (1984a).

Προκειμένου για μέθοδο όπως εφαρμόζεται αυτή στα πεπερασμένα στοιχεία, μετά την σύσταση ενός στοιχείου, αν οι k πλήρως αθροισμένες μεταβλητές αντιμετατίθενται στις πρώτες γραμμές και στήλες του μετωπικού πίνακα, μπορούμε να διασπάσουμε τον μετωπικό πίνακα F στην μορφή

$$F = \begin{pmatrix} B & C \\ D & E \end{pmatrix},$$

(Σφάλμα! Δεν έχει οριστεί στυλ..4)

όπου B είναι ένας τετράγωνος πίνακας τάξης k και E είναι τάξης $r \times r$. Σημειωτέον πως το $k + r$ ισούται με το μέγεθος του μετωπικού πίνακα, και πως γενικά $k << r$. Τυπικά, ο B είναι τάξης 10 έως 20, ενώ ο E είναι τάξης 200 με 500. Οι γραμμές και οι στήλες του B, οι γραμμές του C, και οι στήλες του D είναι πλήρως αθροισμένες. Οι μεταβλητές του E δεν είναι ακόμα πλήρως αθροισμένες. Στοιχεία οδηγοί μπορούν να επιλεγούν από οπουδήποτε μέσα στον B. Για συμμετρικά θετικής διακρίνουσας συστήματα, μπορούν να επιλεγούν από την διαγώνιο στην σειρά αλλά για την περίπτωση μη συμμετρικού πίνακα πρέπει να επιλεγούν έτσι ώστε να ικανοποιούν το κριτήριο κατωφλίου.

6.2 Μέθοδοι ζώνης και αριθμητική οδήγηση

Μία μορφή μετωπικής μεθόδου είναι η μέθοδος που εφαρμόζεται σε πίνακες ζώνης (band matrices), δηλαδή πίνακες των οποίων οι μη-μηδενικές τιμές βρίσκονται στην διαγώνιο και σε θέσεις κοντά από αυτήν.

Μία συνήθης τεχνική για την οργάνωση της παραγοντοποίησης ενός πίνακα ζώνης τάξης n με μισό εύρος ζώνης b είναι να δεσμευτεί μνήμη για έναν πίνακα $b \times 2b - 1$ που είναι και ο μετωπικός πίνακας, και να χρησιμοποιηθεί αυτός ως «παράθυρο» που

ολισθαίνει προς τα κάτω ακολουθώντας την ζώνη όσο προχωράει η απαλοιφή. Έτσι, στην αρχή, ο μετωπικός πίνακας περιέχει τις γραμμές 1 έως b του συστήματος. Αυτή η διάταξη επιτρέπει το πρώτο βήμα οδήγησης να εκτελεστεί (συμπεριλαμβανομένου και της οδήγησης εάν αυτό απαιτείται) και, αν η γραμμή οδηγός μετατοπιστεί εκτός του μετωπικού πίνακα, η γραμμή b+1 του πίνακα ζώνης μπορεί να συμπεριληφθεί στον μετωπικό πίνακα. Έτσι μπορεί κανείς να εκτελέσει το επόμενο βήμα οδήγησης εντός του μετωπικού πίνακα. Τυπικά, ένας πιο μεγάλος μετωπικός πίνακας χρησιμοποιείται διότι μπορεί να επιτευχθεί μεγαλύτερη απόδοση μετακινώντας ομάδες γραμμών ταυτόχρονα. Έτσι συνήθως μπορούν να εκτελεστούν και πολλά βήματα οδήγησης ταυτόχρονα εντός του μετωπικού πίνακα.

Αυτή η μέθοδος «παραθύρου» μπορεί να επεκταθεί σε πίνακες με ζώνη μεταβλητού εύρους (*variable-band matrices*). Σε αυτήν την περίπτωση, ο μετωπικός πίνακας πρέπει να έχει τάξη τουλάχιστον $\max_{a_{ij}\neq 0}\{|i-j|\}$. Περαιτέρω επέκταση σε γενικούς πίνακες είναι δυνατή αν παρατηρηθεί πως οποιοσδήποτε πίνακας μπορεί να αντιμετωπιστεί ως πίνακας ζώνης μεταβλητού εύρους. Εδώ έγκειται και το μεγαλύτερο πρόβλημα με αυτήν την θεώρηση: για οποιοδήποτε τυχαίο πίνακα με τυχαία διάταξη, το απαιτούμενο μέγεθος για τον μετωπικό πίνακα μπορεί να είναι πολύ μεγάλο. Παρόλα αυτά, για διακριτοποιήσεις μερικών διαφορικών εξισώσεων (είτε από πεπερασμένα στοιχεία είτε από πεπερασμένες διαφορές), καλές διατάξεις συναντιόνται συνήθως (για παράδειγμα Duff, Reid και Scott (1989b) και Sloan και Randolph (1983)). Στην πράξη, πρόσφατα πειράματα των Duff και Scott (1997) δείχνουν πως υπάρχει ένα σημαντικό εύρος προβλημάτων όπου η μετωπική μέθοδος είναι η προτιμότερη, ιδιαίτερα αν ο πίνακας εκφράζεται ως το άθροισμα στοιχειωδών πινάκων στην μη διασπασμένη μορφή.

Προκειμένου για τις μετωπικές μεθόδους, η απαλοιφή κατά Gauss του μετωπικού πίνακα όπως εκφράζεται στην (**Σφάλμα! Δεν έχει οριστεί στυλ.**.4) εκτελείται στον τετράγωνο πίνακα B μιας και όλα του τα στοιχεία είναι πλήρως αθροισμένα. Ο στόχος είναι να εκτελεστούν k βήματα απαλοιφής στον μετωπικό πίνακα επιλέγοντας στοιχεία οδηγούς από τον B για να αποθηκευτούν οι παράγοντες $L_U U_B$ του B, DB^{-1}, και C σε βοηθητική μνήμη, και να παραχθεί το συμπλήρωμα κατά Schur $E - DB^{-1}C$ για χρήση στο επόμενο βήμα του αλγόριθμου.

Όπως αναφέρθηκε, τα στοιχεία οδηγοί μπορούν να επιλεγούν οπουδήποτε μέσα στον B. Στην προσέγγιση που επιλέγει ο Duff (1984a), χρησιμοποιείται η τυπική τεχνική για τους αραιούς πίνακες οδήγησης με κατώφλι, όπου το $b_{ij} \in B$ είναι κατάλληλο ως οδηγός μόνο αν

$$\left|b_{ij}\right| \geq u \max\left(\max_s \left|b_{sj}\right|, \max_s \left|d_{sj}\right|\right), \qquad \textbf{(Σφάλμα! Δεν έχει οριστεί στυλ.}.5)$$

όπου u είναι μία προκαθορισμένη παράμετρος στο εύρος $0 \leq u \leq 1$.

Είναι σημαντικό να τονιστεί πως, αν και επιλέγονται οδηγοί από τον υποπίνακα B, πρέπει πάντα να γίνεται ο έλεγχος των υποψηφίων οδηγών στοιχείων για αριθμητική σταθερότητα συγκρίνοντας το μέγεθός τους με τις καταχωρήσεις στον B και στον D. Αυτό σημαίνει πως μεγάλες τιμές στον D μπορεί να αποτρέψουν την επιλογή κάποιων οδηγών στον B. Όταν συμβαίνει αυτό εκτελούνται $k_1 \leq k$ βήματα απαλοιφής κατά Gauss και το επακόλουθο συμπλήρωμα κατά Schur $E - DB_1^{-1}C$, όπου B_1 είναι ένα τετράγωνο υποσύνολο του B τάξης k_1, θα έχει μία τάξη $r + k - k_1$. Ενώ αυτό αυξάνει το μέγεθος της

μνήμης και του έργου που πρέπει να γίνει, το επιπλέον κόστος είναι τυπικά πολύ μικρό, και όλα τα βήματα οδήγησης θα εκτελεστούν τελικά εφόσον ο τελικός μετωπικός πίνακας έχει τον υποπίνακά του Ε μηδενικής τάξης.

Μία ενδιαφέρουσα οπτική των μετωπικών μεθόδων είναι πως όλα τα βήματα απαλοιφής γίνονται σε πυκνούς πίνακες, όποτε οι γνωστές τεχνικές των πυκνών πινάκων (μαζί με αυτές που εκμεταλλεύονται τα κατανεμημένα συστήματα και την παράλληλη επεξεργασία) μπορούν να χρησιμοποιηθούν. Είναι επίσης σημαντικό πως ο k είναι συνήθως μεγαλύτερος του 1, στην οποία περίπτωση περισσότερες από μία απαλοιφές μπορούν να γίνουν στον μετωπικό πίνακα και η BLAS επιπέδου 2 και 3 αλγόριθμοι μπορούν να χρησιμοποιηθούν ως πυρήνες υπολογισμών. Πραγματικά, σε κάποιες αρχιτεκτονικές (για παράδειγμα στον SGI Power Challenge) μπορεί να είναι ευεργετικό να αυξάνεται το μέγεθος του Β κάνοντας μερικές επιπλέον συνθέσεις πριν το στάδιο απαλοιφής, ακόμα και αν αυτό απαιτεί περισσότερες πράξεις κινητής υποδιαστολής για την ολοκλήρωση της παραγοντοποίησης (Cliffe, Duff και Scott (1998)).

6.3 Παράλληλη υλοποίηση μετωπικών μεθόδων

Ο τρόπος που εκμεταλλεύεται ο παραλληλισμός όταν χρησιμοποιούνται μετωπικές μέθοδοι είναι όμοιος με την αποσύνθεση πεδίου (*domain decomposition*), όπου τμηματοποιείται το υποκείμενο "πεδίο" σε υποπεδία, εκτελείται η μετωπική αποσύνθεση σε κάθε υποπεδίο χωριστά (αυτό μπορεί να γίνει παράλληλα) και μετά παραγοντοποιείται ο πίνακας που αντιστοιχεί στις υπολειπόμενες «συνοριακές» (*boundary* ή *interface*) μεταβλητές (στο σύστημα του συμπληρώματος κατά Schur), ίσως επίσης χρησιμοποιώντας μη μετωπική τεχνική, όπως στον Duff και Scott (1994), ή με οποιονδήποτε άλλο κατάλληλο επιλυτή. Αυτή η στρατηγική αντιστοιχεί σε μία φραγμένη

τμηματοποιημένη διαγώνια αναδιάταξη του πίνακα και μπορεί να ενθυλακωθεί. Πιο πρόσφατα, μία κλάση αλγόριθμων έχει δημιουργηθεί (Ashcraft και Liu 1996) που συνδυάζει διάφορες στρατηγικές αναδιάταξης στις συνοριακές μεταβλητές.

Αν και το κύριο κίνητρο για την χρήση πολλαπλών μετώπων είναι συνήθως για να γίνει εκμετάλλευση της παράλληλης επεξεργασίας, είναι επίσης σημαντικό το ότι, ο όγκος του έργου μπορεί να μεταβληθεί σημαντικά τμηματοποιώντας το πεδίο, και σε μερικές περιπτώσεις μπορεί να ελαττωθεί σημαντικά. Για παράδειγμα σε μία διακριτοποίηση πεπερασμένων διαφορών 5 σημείων σε ένα $2N \times 2N$ πλέγμα καταλήγει κανείς σε ένα πίνακα τάξης $4N^2$ και εύρους ημι-ζώνης $2N$. Τότε η απευθείας λύση χρησιμοποιώντας ένα μετωπικό σχήμα απαιτεί $32N^4 + O(N^3)$ πράξεις κινητής υποδιαστολής, ενώ αν το πεδίο είναι τμηματοποιημένο σε τέσσερα υποπεδία τάξης $N \times N$, ο αριθμός των πράξεων μπορεί να ελαττωθεί σε $18{,}6N^4 + O(N^3)$. Δεν πρέπει να αμεληθεί πως η διάταξη των μη-μηδενικών στοιχείων εντός του πίνακα είναι σημαντική για να επιτευχθεί αυτή η απόδοση.

Αυτή η αλλαγή στον αριθμό των πράξεων κινητής υποδιαστολής γίνεται εμφανής όταν εκτελείται ένας κώδικας πολλαπλών μετώπων σε ένα πρόγραμμα-μοντέλο ενός πλέγματος 48x48 πεπερασμένων στοιχείων σε έναν επεξεργαστή ενός CRAY Y-MP (Duff και Scott 1984). Με ένα μόνο πεδίο, ο χρόνος του επεξεργαστή ήταν 68,4 sec και εκτελέστηκαν 16970 εκατομμύρια πράξεις κινητής υποδιαστολής κατά την παραγοντοποίηση, ενώ τα ίδια μεγέθη για τμηματοποίηση σε τέσσερα υποπεδία είναι 48,4 sec και 10350. Με περισσότερα υποπεδία τα μεγέθη βελτιώνονται ακόμα περισσότερο, για παράδειγμα με 8 υποπεδία είναι 40,3 sec και 8365.

Αυτή η προσέγγιση εξετάστηκε σε δύο παράλληλα περιβάλλοντα: σε ένα σύστημα 8 επεξεργαστών με κοινόχρηστη μνήμη CRAY Y-MP8I και σε ένα δίκτυο 5 DEC Alpha σταθμών εργασίας χρησιμοποιώντας το PVM. Σε κάθε περίπτωση υποδιαιρούμε το πρόβλημα σε τόσα υποπεδία όσοι και οι επεξεργαστές που χρησιμοποιήθηκαν. Οι χρόνοι που υπολογίστηκαν στον πίνακα που ακολουθεί είναι στην περίπτωση που οι παραπάνω υπολογιστικές διατάξεις ήταν κάτω από χαμηλό φόρτο εργασίας.

Τα αποτελέσματα στον CRAY είναι ενθαρρυντικά και δείχνουν καλή επιτάχυνση (*speedup*). Αυτά στους Alpha είναι αρκετά συγκρίσιμα και δείχνουν ότι η επιβάρυνση από το PVM και την επικοινωνία μεταξύ των επεξεργαστών δεν κυριαρχούν.

Παρατηρείται επίσης πως η επιτάχυνση όταν αυξάνεται ο αριθμός των υποπεδίων από 2 σε 4 είναι μεγαλύτερη του 2. Αυτή η μη αναμενόμενη συμπεριφορά δημιουργείται από την μείωση των πράξεων κινητής υποδιαστολής για τα τέσσερα υποπεδία όπως εξηγήθηκε παραπάνω.

Περισσότερη έρευνα στην παράλληλη υλοποίηση της μετωπικής απαλοιφής έγινε από τους Benner, Montry και Weigand (1987) χρησιμοποιώντας λεπτομερές παραλληλισμό σε CRAY Y-MP και ELXSI, και από τους Lucas, Blank και Tiemann (1987) σε ένα hypercube.

Υποπεδία	CRAY Y-MP		DEC Alpha (5x)	
	Χρόνος	Επιτάχυνση	Χρόνος	Επιτάχυνση
1	98,8		1460,3	
2	64,6	1,5	1043,0	1,4
4	30,7	3,2	457,5	3,2
8	15,3	6,5		

Πίνακας 6.1: Απόδοση του MA42 σε πολλαπλούς επεξεργαστές σε ένα πλέγμα 48x48 πεπερασμένων στοιχείων

ΠΣφάλμα! Δεν έχει οριστεί στυλ..1

7 Μέθοδοι πολλαπλών μετώπων

7.1 Εισαγωγή

Ακολουθώντας το λογικό συμπέρασμα που προκύπτει από την εκμετάλλευση του παραλληλισμού στις μετωπικές μεθόδους, γίνεται εμφανές πως πολλαπλά μέτωπα μπορούν να αναπτυχθούν ταυτόχρονα και μπορούν να επιλεγούν χρησιμοποιώντας μία διάταξη διατήρησης της αραιότητας όπως αυτή της ελάχιστης τάξης. Αυτές οι μέθοδοι ονομάζονται μέθοδοι πολλαπλών μετώπων. Η ιδέα είναι να συνδυαστεί η διατήρηση της αραιότητας με την αποδοτικότητα ενός πυρήνα μετωπικού πίνακα έτσι ώστε να είναι δυνατή η εκμετάλλευση των υπολογιστών υψηλών επιδόσεων (Duff et al. (1986), Duff (1986a, 1986b, 1989)).

Τα σημαντικά σημεία των μεθόδων αυτών, καθώς και η έννοια του δέντρου απαλοιφής μπορούν να γίνουν κατανοητά με ένα μικρό παράδειγμα. Το δέντρο απαλοιφής ορίζει μία σειρά προτεραιότητας κατά την παραγοντοποίηση (Duff (1986a), Liu (1990)). Η παραγοντοποίηση αρχίζει στα φύλλα του δέντρου και τα δεδομένα προχωράνε προς την ρίζα ακολουθώντας τις ακμές του δέντρου. Για να ολοκληρωθεί το έργο που ακολουθεί έναν κόμβο, όλα τα δεδομένα που προέρχονται από προηγούμενους κόμβους πρέπει να έχουν συλλεχθεί, αλλιώς το έργο σε διαφορετικούς κόμβους είναι ανεξάρτητο. Το παράδειγμα του παρακάτω σχήματος απεικονίζει τη μέθοδο πολλαπλών μετώπων και την ερμηνεία της σε συνάρτηση με το δέντρο απαλοιφής.

Ο πίνακας της εικόνας [**Σφάλμα! Δεν έχει οριστεί στυλ.**.1] έχει μία διάταξη μη-μηδενικών στοιχείων που είναι συμμετρική. Οποιοσδήποτε πίνακας μπορεί βέβαια να θεωρηθεί ότι είναι σε αυτήν την μορφή αρκεί να αποθηκευτούν ρητά μηδενικά στοιχεία.

Ο πίνακας έχει τέτοια διάταξη ώστε οι οδηγοί να βρίσκονται στην διαγώνιό του. Στο πρώτο βήμα μπορούμε να εκτελέσουμε ένα βήμα απαλοιφής με οδηγό το στοιχείο $(1,1)$, πρώτα «συνθέτοντας» την γραμμή 1 και την στήλη 1 για να προκύψει ο πίνακας του σχήματος [**Σφάλμα! Δεν έχει οριστεί στυλ.**.2]. Με τον όρο «σύνθεση» εννοείται η τοποθέτηση των καταχωρήσεων της γραμμής και στήλης 1 σε έναν υποπίνακα τάξης και τον αριθμό των στοιχείων στην γραμμή και στήλη 1. Έτσι τα μηδενικά στοιχεία a_{12} και a_{21} κάνουν την γραμμή και στήλη 2 να παραλειφθούν στο σχήμα [**Σφάλμα! Δεν έχει οριστεί στυλ.**.2], και έτσι ένα διάνυσμα δεικτών απαιτείται για να εντοπιστούν οι γραμμές και στήλες που βρίσκονται στον υποπίνακα. Προκειμένου για το σχήμα [**Σφάλμα! Δεν έχει οριστεί στυλ.**.2] το διάνυσμα δεικτών θα είχε την τιμή $(1,3,4)$ για τις γραμμές και στήλες. Η στήλη 1 απαλείφεται χρησιμοποιώντας τον οδηγό $(1,1)$ για να δώσει τον μειωμένο υποπίνακα τάξης 2 με διάνυσμα δεικτών $(3,4)$. Στην κανονική απαλοιφή κατά Gauss, η μορφή της ενημέρωσης είναι

$$a_{ij} \Leftarrow a_{ij} - a_{i1}\left[a_{11}\right]^{-1} a_{1j} \qquad \text{(Σφάλμα! Δεν έχει οριστεί στυλ.}.1)$$

και θα μπορούσε να εκτελεστεί αμέσως για όλα τα (i,j) έτσι ώστε $\alpha_{i1}a_{1j} \neq 0$. Σε αυτήν την περίπτωση όμως τα μεγέθη

$$a_{i1}\left[a_{11}\right]^{-1} a_{1j} \qquad \text{(Σφάλμα! Δεν έχει οριστεί}$$

στυλ..2)

κρατούνται στον μειωμένο υποπίνακα και οι αντίστοιχες πράξεις ενημέρωσης δεν εκτελούνται αμέσως. Αυτές οι ενημερώσεις δεν είναι απαραίτητες μέχρι η αντίστοιχη καταχώρηση απαιτηθεί από μία μεταγενέστερη γραμμή ή στήλη οδήγησης. Ο μειωμένος υποπίνακας μπορεί να αποθηκευτεί μέχρι τότε.

x		x	x
	x	x	x
x	x	x	
x	x		x

Εικόνα 7-1: Πίνακας που απεικονίζει την μέθοδο των πολλαπλών μετώπων

[**Σφάλμα! Δεν έχει οριστεί στυλ.**.1]

x	x	x
x		
x		

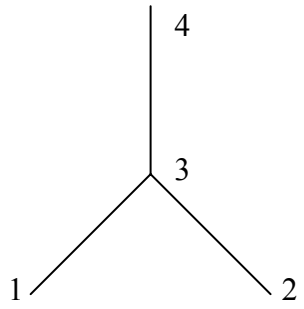
Εικόνα 7-2: Σύνθεση της πρώτης γραμμή και στήλης

[**Σφάλμα! Δεν έχει οριστεί στυλ.**.2]

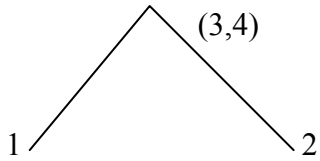
Η γραμμή (και στήλη) 2 συντίθεται, η καταχώρηση $(2,2)$ χρησιμοποιείται ως οδηγός για να εξαλειφθεί η στήλη 2, και ο μειωμένος υποπίνακας τάξης 2 – με τους αντίστοιχους δείκτες γραμμής και στήλης που έχουν την τιμή $(3,4)$ – αποθηκεύεται. Αυτοί οι υποπίνακες ονομάζονται μετωπικοί πίνακες. Περισσότεροι του ενός μετωπικοί πίνακες αποθηκεύονται ανά πάσα στιγμή (στην περίπτωσή που εξετάσθηκε 2). Για αυτόν τον λόγο η μέθοδος ονομάζεται πολλών μετώπων.

Πριν εκτελεστούν οι πράξεις για την οδήγηση με οδηγό το στοιχείο $(3,3)$, η ενημέρωση των πρώτων δύο απαλοιφών (οι αποθηκευμένοι πίνακες τάξης 2) πρέπει να εκτελεστεί στην αρχική γραμμή και στήλη 3, χρησιμοποιώντας το διάνυσμα δεικτών για τον έλεγχο της άθροισης. Το αποτέλεσμα είναι ένας συντεθειμένος υποπίνακας τάξης 2

και με δείκτες $(3,4)$ για τις γραμμές και στήλες. Η πράξη οδήγησης που απαλείφει την στήλη 3 με τον οδηγό $(3,3)$ αφήνει έναν μειωμένο υποπίνακα τάξης ένα με διάνυσμα γραμμών και στηλών το (4). Το τελευταίο βήμα αθροίζει αυτόν τον πίνακα με το στοιχείο $(4,4)$ του αρχικού πίνακα. Αυτή η αλληλουχία κύριων βημάτων απεικονίζεται στο δέντρο του σχήματος [**Σφάλμα! Δεν έχει οριστεί στυλ.**.3].



Εικόνα 7-3: Δέντρο απαλοιφής για τον πίνακα της εικόνας [Σφάλμα! Δεν έχει οριστεί στυλ.**.1]**



Εικόνα 7-4: Δέντρο σύνθεσης για τον πίνακα της εικόνας [Σφάλμα! Δεν έχει οριστεί στυλ.**.1] μετά από την συνένωση των κόμβων**

Ο ίδιος χώρος και αριθμητική απαιτούνται αν το στοιχείο $(4,4)$ συντεθεί ταυτόχρονα με το στοιχείο $(3,3)$, και σε αυτήν την περίπτωση δύο βήματα οδήγησης μπορούν να γίνουν στον ίδιο υποπίνακα. Αυτή η διαδικασία αντιστοιχεί στην συνένωση των κόμβων 3 και 4 του δέντρου της εικόνας [**Σφάλμα! Δεν έχει οριστεί στυλ.**.3] για να δώσει το δέντρο της εικόνας [**Σφάλμα! Δεν έχει οριστεί στυλ.**.4]. Σε τυπικά προβλήματα η συνένωση δέντρου δίνει ένα δέντρο με αριθμό κόμβων ίσο με περίπου το μισό της τάξης του πίνακα. Αυτό το δέντρο ονομάζεται δέντρο σύνθεσης (*assembly tree*).

Πρόσθετο πλεονέκτημα μπορεί να αποκομισθεί από συνενώσεις που δεν διατηρούν τον αριθμό των αριθμητικών πράξεων, δηλαδή όταν υπάρχουν μεταβλητές σε κόμβους παιδιά που δεν υπάρχουν στους κόμβους γονείς. Αυτό ονομάζεται χαλαρή συνένωση (*relaxed amalgamation*) (Duff και Reid (1984), Ashcraft (1987), Liu (1990)).

Οι υπολογισμοί σε έναν κόμβο του δέντρου είναι απλά η σύνθεση της πληροφορίας που αφορά τον κόμβο, μαζί με την σύνθεση των μειωμένων υποπινάκων των παιδιών του, ακολουθούμενοι από μερικά βήματα απαλοιφής κατά Gauss. Κάθε κόμβος αντιστοιχεί στον σχηματισμό ενός μετωπικού πίνακα της μορφής (**Σφάλμα! Δεν έχει οριστεί στυλ.**.4) μαζί με μερικά βήματα απαλοιφής, μετά από τα οποία το συμπλήρωμα κατά Schur μεταφέρεται για σύνθεση στους κόμβους γονείς.

Το να θεωρήσει κανείς την παραγοντοποίηση χρησιμοποιώντας το δέντρο σύνθεσης έχει πολλά πλεονεκτήματα. Επειδή μόνο μία μερική διάταξη ορίζεται από το δέντρο, η μόνη απαίτηση για μία αριθμητική παραγοντοποίηση με την ίδια ποσότητα αριθμητικών πράξεων είναι πως οι υπολογισμοί πρέπει να εκτελεστούν για όλους τους κόμβους παιδιά πριν ολοκληρωθούν αυτοί στους κόμβους γονείς. Έτσι πολλές διαφορετικές διατάξεις με τον ίδιο αριθμό πράξεων κινητής υποδιαστολής είναι δυνατές και μπορούν να παραχθούν από το δέντρο απαλοιφής. Συγκεκριμένα, διατάξεις μπορούν να επιλεγούν για οικονομία μνήμης, για αποδοτικότητα σε εργασίες εκτός του πυρήνα, ή για παράλληλη υλοποίηση. Επιπρόσθετα, μικρές διαταραχές στο δέντρο και τον αριθμό των πράξεων κινητής υποδιαστολής μπορούν να γίνουν για να αντιμετωπίσουν την ασυμμετρία ή να βελτιώσουν την αριθμητική οδήγηση.

Ο Liu (1990) παρουσιάζει μία μελέτη του ρόλου των δέντρων απαλοιφής στην απαλοιφή κατά Gauss, προτείνει την αποδοτική δημιουργία δέντρων σε χρόνο πρακτικά

γραμμικό με την τάξη του πίνακα και συζητά το αποτέλεσμα των διαφορετικών διατάξεων των δέντρων και πράξεις στα δέντρα που διατηρούν τις ιδιότητες της απαλοιφής. Για παράδειγμα, η διάταξη ενός δέντρου μπορεί να επηρεάσει το χώρο που απαιτούν οι ενδιάμεσοι μετωπικοί πίνακες που παράγονται κατά την απαλοιφή. Δυνατές πράξεις σε δέντρα εμπεριέχουν την περιστροφή δέντρου, που επιτρέπει σε ένα δέντρο, για παράδειγμα, να είναι πιο κατάλληλο για να οδηγήσει μια παραγοντοποίηση που εκμεταλλεύεται καλύτερα την παραλληλότητα (Simon, Vu και Yang 1989).

7.2 Απόδοση σε παράλληλες αρχιτεκτονικές

Στις μετωπικές μεθόδους υπάρχουν τρία είδη εργασιών, η σύνθεση της πληροφορίας των παιδιών, η επιλογή των οδηγών και οι απαλοιφές που συνεπάγονται. Οι Duff και Reid (1983), στην υλοποίηση που προτείνουν για παράλληλες αρχιτεκτονικές, επιλέγουν να αποθηκεύσουν όλες τις εργασίες που είναι διαθέσιμες για εκτέλεση σε μία μοναδική ουρά με μία ετικέτα αναγνώρισης της δουλειάς που αντιστοιχεί στην εργασία. Μόλις ελευθερωθεί ένας επεξεργαστής, μεταβαίνει στην αρχή της ουράς, επιλέγει την εκεί εργασία, διαβάζει την ετικέτα και εκτελεί τις απαραίτητες ενέργειες. Αυτή η διεργασία μπορεί να γεννήσει καινούργιες εργασίες που θα προστεθούν στο τέλος της ουράς. Αυτό το μοντέλο χρησιμοποιήθηκε από τον Duff, ο οποίος σχεδίασε ένα πρωτότυπο παράλληλο κώδικα από τον κώδικα MA37 της HSL, και οι επιταχύνσεις που πέτυχε στον Alliant FX/8 φαίνονται στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1 (Duff 1989b). Αυτός ο πρωτότυπος κώδικας εξελίχθηκε και έγινε ο MA41.

Αρ. επεξεργαστών	Χρόνος	Επιτάχυνση
1	2,59	
2	1,36	1,9
4	0,74	3,5
6	0,57	4,5
8	0,46	5,6

Πίνακας 7.1: Επιτάχυνση σε έναν Alliant FX/8 μίας Laplace διακριτοποίησης 5 σημείων ενός πλέγματος 30x30

Το κρίσιμο σημείο των μεθόδων πολλών μετώπων που επιτρέπει την εκμετάλλευση της παραλληλότητας είναι ότι το έργο στους διάφορους κόμβους του δέντρου σύνθεσης είναι ανεξάρτητο και πως ο μόνος συγχρονισμός που χρειάζεται είναι ότι τα δεδομένα από τα παιδιά να είναι διαθέσιμα πριν οι υπολογισμοί σε έναν κόμβο μπορέσουν να ολοκληρωθούν. Ο υπολογισμός σε οποιοδήποτε φύλλο μπορεί να εκτελεστεί άμεσα και ταυτόχρονα. Αν και αυτό παρέχει πολύ παραλληλισμό κοντά στα φύλλα του δέντρου, υπάρχει λιγότερος κοντά στην ρίζα και, φυσικά, για ένα μη απλοποιήσιμο πρόβλημα υπάρχει μόνο ένας κόμβος ρίζα. Αν οι κόμβοι του δέντρου θεωρηθούν ατομικοί (δηλαδή οι υπολογισμοί που εκτελούν δεν μπορούν να χωριστούν σε τμήματα), τότε το επίπεδο παραλληλότητας είναι ένα στην ρίζα και συνήθως αυξάνεται αργά όσο προχωράμε μακριά από την ρίζα. Αν, όμως, αναγνωρίσουμε πως ο παραλληλισμός μπορεί να εκμεταλλευτεί και εντός των υπολογισμών ενός κόμβου (αντιστοιχίζοντας σε ένα ή περισσότερα βήματα απαλοιφής κατά Gauss σε πυκνό πίνακα), πολύ μεγαλύτερος παραλληλισμός μπορεί να επιτευχθεί. Η απώλεια παραλληλισμού θεωρώντας τους κόμβους ως ατομικούς αυξάνεται από το γεγονός ότι το μεγαλύτερο μέρος αριθμητικής κινητής υποδιαστολής εκτελείται κοντά στην ρίζα, οπότε είναι ζωτικής σημασίας να εκμεταλλευτεί ο παραλληλισμός και μέσα στους υπολογισμούς των κόμβων. Το ποσό παραλληλισμού που είναι διαθέσιμο φαίνεται στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.2, όπου απεικονίζονται το μέγεθος και ο αριθμός των κόμβων στα φύλλα και κοντά

στην ρίζα. Έτσι βλέπουμε πως ενώ υπάρχει πολύ παραλληλισμός στην αρχή της παραγοντοποίησης, αυτός ελαττώνεται σημαντικά κοντά στην ρίζα, εάν οι κόμβοι θεωρηθούν ατομικοί. Επιπρόσθετα, 75% του έργου της παραγοντοποίησης εκτελείται στα 3 κορυφαία επίπεδα. Επειδή το μέγεθος των μετωπικών πινάκων κοντά στην ρίζα του δέντρου είναι πολύ μεγαλύτερο, μπορούμε να εκμεταλλευτούμε τον παραλληλισμό σε αυτό το στάδιο της παραγοντοποίησης χρησιμοποιώντας, για παράδειγμα, παραλλαγές της BLAS επιπέδου 3 για τις πράξεις απαλοιφής εντός του κόμβου. Τα αποτελέσματα αυτής της προσέγγισης μετρήθηκαν από τους Amestoy, Daydé, Duff και Merère (1995) και φαίνονται καθαρά στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.3, όπου οι βελτιωμένες επιταχύνσεις της στήλης 2 οφείλονται στον παραλληλισμό εντός των κόμβων.

Πίνακας	Τάξη	Κόμβοι	Φύλλα		Κορυφαία 3 επίπεδα	
			Αριθμός	Μέσο μεγ.	Αριθμός	Μέσο μεγ.
BCSSTK15	3948	576	317	13	10	376
BCSSTK33	8738	545	198	5	10	711
BBMAT	38744	5716	3621	23	10	1463
GRE1107	1107	344	250	7	12	129
SAYLR4	3564	1341	1010	5	12	123
GEMAT11	4929	1300	973	10	112	148

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.2

Πίνακας 7.2: Στατιστικές στα μετωπικά μεγέθη στο δέντρο

Πίνακας	Τάξη	Μη μηδενικά	(1)		(2)	
			Mflop/s	επιτάχυνση	Mflop/s	επιτάχυν
WANG3	26064	177168	1062	1,42	3718	4,98
WANG4	26068	177196	1262	1,70	3994	5,39
BBMAT	38744	1771722	2182	3,15	3777	5,46

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.3

Πίνακας 7.3: Περίληψη απόδοσης παραγοντοποίησης πολλαπλών μετώπων με τον MA41 σε 7 επεξεργαστές ενός CRAY C98. Στις στήλες (1) εκμεταλλεύεται μόνο η παραλληλότητα στο δέντρο. Στις στήλες (2) συνδυάζονται οι δύο μορφές παραλληλότητας.

Ο κώδικας που χρησιμοποιήθηκε στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.3 ήταν ο MA41 (Amestoy και Duff 1993, Amestoy και Duff 1997) που αναπτύχθηκε για

παράλληλους υπολογιστές διαμοιραζόμενης μνήμης. Έμφαση πρέπει να δοθεί στο ότι, λόγω της φορητότητας που παρέχεται μέσω των κωδίκων BLAS, ο MA41 είναι πρακτικά ίδιος για όλους τους υπολογιστές κοινής μνήμης. Ο MA41 τρέχει με μικρές τροποποιήσεις και σε υπολογιστές εικονικά διαμοιραζόμενης μνήμης, αν και θα ήταν ευνοϊκό να γίνουν σημαντικότερες αλλαγές για καλύτερη απόδοση.

Κώδικες πολλαπλών μετώπων έχουν επίσης αναπτυχθεί για υπολογιστές κατανεμημένης μνήμης. Μία αποδοτική έκδοση παραλληλισμού δεδομένων του κώδικα πολλαπλών μετώπων υλοποιήθηκε από τους Conroy, Kratzer και Lucas (1994). Οι πιο εντυπωσιακοί χρόνοι και επιταχύνσεις που έχουν αναφερθεί είναι των Gupta και Kumar (1994) για μια υλοποίηση του κώδικα πολλαπλών μετώπων σε έναν υπολογιστή 1024 επεξεργαστών nCUBE-2. Τα αποτελέσματα των μετρήσεων φαίνονται στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.4. Για μεγαλύτερους πίνακες δείχνουν ακόμα καλύτερες επιδόσεις με επιταχύνσεις της τάξης του 350 σε 1024 επεξεργαστές για τον πίνακα BCSSTK31 τάξης 35588 με περίπου 6.4 εκατομμύρια μη-μηδενικά στοιχεία. Σε επόμενα πειράματα, οι Gupta, Karypis και Kumar (1996) υλοποίησαν εκδόσεις των αλγορίθμων σε έναν CRAY T3D, χρησιμοποιώντας το SHMEM για την ανταλλαγή μηνυμάτων, και κατάφεραν επίσης υψηλές επιδόσεις και καλή επιτάχυνση.

Αρ. επεξεργαστών	1	4	16	64	256
Χρόνος	103,7	26,7	8,3	3,2	1,5
Επιτάχυνση	1	3,9	12,5	32,4	67,8
Απόδοση (%)	100	97	78	51	27
Φόρτος εργασίας (%)	100	98	91	87	84

Πίνακας 7.4: Απόδοση του κώδικα των Gupta και Kumar σε έναν nCUBE-2 για τον πίνακα BCSSTK33. Χρόνοι σε δευτερόλεπτα.

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.4

Εμφανώς ο παραλληλισμός που είναι διαθέσιμος μέσω του δέντρου απαλοιφής εξαρτάται από την διάταξη του πίνακα. Γενικά, κοντά και «θαμνώδη» δέντρα

προτιμώνται από τα ψηλά και λεπτά, εφόσον ο αριθμός των επιπέδων υποδηλώνει και την εν γένει διαδοχικότητα των υπολογισμών. Δύο κοινές στρατηγικές αναδιάταξης για γενικά αραιά συμμετρικά συστήματα είναι ο ελάχιστος βαθμός (*minimum degree*) και η ένθετη ανατομή (*nested dissection*) (για παράδειγμα, Duff et al. (1986) και George και Liu (1981)). Ενώ αυτές οι αναδιατάξεις είναι παρόμοιες σε συμπεριφορά όσο αφορά την αριθμητική και την αποθήκευση, δίνουν αρκετά διαφορετικά επίπεδα παραλληλισμού όταν χρησιμοποιούνται για να κατασκευάσουν το δέντρο απαλοιφής. Αυτό φαίνεται καθαρά στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.5, όπου η μέγιστη επιτάχυνση υπολογίζεται από μία εξομοίωση της απαλοιφής ως τον λόγο του αριθμού των πράξεων στο γραμμικό πρόβλημα προς τον αριθμό των διαδοχικών πράξεων στην παράλληλη έκδοση, λαμβάνοντας υπόψη και την μετακίνηση των δεδομένων όπως των πράξεων κινητής υποδιαστολής (Duff και Johnson 1989).

Διάταξη	Ελάχιστος βαθμός	Ένθετη ανατομή
Αριθμός επιπέδων στο δέντρο	52	15
Αριθμός οδηγών στην μεγαλύτερη διαδρομή	232	61
Μέγιστη επιτάχυνση	9	47

Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.5

Πίνακας 7.5: Σύγκριση των δύο αλγόριθμων αναδιάταξης για την δημιουργία του δέντρου απαλοιφής μίας λύσης πολλαπλών μετώπων. (το πρόβλημα είναι το αποτέλεσμα μίας διακριτοποίησης 5 σημείων ενός πλέγματος 10x100)

7.3 Εκμετάλλευση της δομής

Οι μέθοδοι πολλαπλών επιπέδων που εξετάστηκαν κατασκευάζουν το δέντρο απαλοιφής και τις αντίστοιχες πληροφορίες καθοδήγησης για την αριθμητική παραγοντοποίηση χωρίς πρόσβαση στις αριθμητικές τιμές και θεωρώντας πως η επιλεχθείσα σειρά οδήγησης θα είναι αριθμητικά κατάλληλη. Για τους πίνακες που δεν έχουν θετική

διακρίνουσα αυτό δεν είναι πάντα ο κανόνας, οπότε και η αριθμητική ολοκλήρωση πρέπει να είναι ικανή να αντέχει επιβεβλημένες αλλαγές στην προβλεφθείσα αλληλουχία κάνοντας επιπλέον βήματα οδήγησης. Για γενικά συμμετρικά συστήματα αυτό δεν είναι συνήθως μεγάλο πρόβλημα και οι επιβαρύνσεις από την επιπρόσθετη οδήγηση είναι σχετικά χαμηλές. Παρόλα αυτά συμμετρικά συστήματα της μορφής

$$\begin{pmatrix} H & A \\ A^T & 0 \end{pmatrix},$$

(**Σφάλμα! Δεν έχει οριστεί στυλ.**.3)

που ονομάζονται επαυξημένα συστήματα προκύπτουν συχνά σε πολλά πεδία εφαρμογών (Duff 1994). Είναι απαραίτητο να γίνει εκμετάλλευση της δομής κατά την διάρκεια της συμβολικής ανάλυσης έτσι ώστε οι οδηγοί να μην επιλέγονται από τον μηδενικό υποπίνακα και αυτός να διατηρηθεί κατά την διάρκεια της παραγοντοποίησης. Το αποτέλεσμα του να λαμβάνει κανείς υπόψη αυτήν την δομή μπορεί να είναι εντυπωσιακό. Για παράδειγμα σε έναν πίνακα της μορφής (**Σφάλμα! Δεν έχει οριστεί στυλ.**.3) με το H να είναι διαγώνιος πίνακας τάξης 1028 με 504 άσσους και 524 μηδενικά και ο A ο 1028 x 524 πίνακας FFFFF800 από την συλλογή του Gay (1985), ένας προηγούμενος κώδικας πολλαπλών μετώπων που δεν εκμεταλλεύεται την δομή του πίνακα και τον μηδενικό υποπίνακα (ο MA27 της HSL) προβλέπει 1,5 εκατομμύρια πράξεις κινητής υποδιαστολής αλλά τελικά απαιτεί 16,5 εκατομμύρια. Ο MA47 της HSL (Duff και Reid 1995, Duff και Reid 1996b), αντιθέτως, προβλέπει και απαιτεί μόνο 7954 πράξεις κινητής υποδιαστολής. Δυστυχώς ο νέος κώδικας είναι πολύ περισσότερο σύνθετος και απαιτεί περισσότερη μετακίνηση δεδομένων αφού οι μετωπικοί πίνακες δεν

απορροφώνται υποχρεωτικά από τον κόμβο γονέα και μπορούν να ανέβουν το δέντρο. Επιπλέον, η ποινή για την απομάκρυνση από την προβλεφθείσα αλληλουχία οδήγησης μπορεί να είναι ιδιαίτερα αυστηρή.

7.4 Μέθοδοι πολλαπλών μετώπων για μη συμμετρικά συστήματα

Αν και τα σχήματα πολλαπλών μετώπων που αναλύθηκαν είναι σχεδιασμένα για δομικά συμμετρικούς πίνακες, δομικά μη συμμετρικοί πίνακες μπορούν να αντιμετωπιστούν αποθηκεύοντας ρητά μηδενικές καταχωρίσεις. Είναι πιο σύνθετο το να σχεδιάσει κανείς ένα αποτελεσματικό σχήμα πολλαπλών μετώπων για πίνακες που είναι δομικά μη συμμετρικοί. Η κύρια διαφορά είναι πως η απαλοιφή δεν μπορεί να εκφραστεί ως ένα δέντρο αλλά απαιτείται ένας κατευθυντικός μη κυκλικός γράφος (Eisenstat και Liu 1992). Οι μετωπικοί πίνακες δεν είναι πλέον τετράγωνοι και δεν απορροφώνται υποχρεωτικά από τον γονικό κόμβο και μπορούν να διατηρηθούν στον γράφο. Τέλος η περιπλοκότητα μίας εκ των υστέρων αριθμητικής οδήγησης είναι ακόμα περισσότερο προβληματική με αυτό το σχήμα, με αποτέλεσμα η προσέγγιση που υιοθετείται είναι να λαμβάνονται υπόψη οι πραγματικές τιμές όταν υπολογίζεται ο γράφος και η σειρά οδήγησης.

Ένας μη συμμετρικός κώδικας πολλαπλών μετώπων των Davis και Duff, βασισμένος σε αυτήν την προσέγγιση, εμπεριέχεται στην υπορουτίνα MA38 της Harwell Subroutine Library. Η σύγκρισή του με έναν συμμετρικό κώδικα (τον MA41 της HSL) φαίνεται στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.6.

Τάξη	13535	62424	26064	22560	120750	360
Μη-μηδενικές τιμές	390607	1717792	177168	1014951	1224224	227
Δείκτης ασυμμετρίας	0,00	0,00	0,00	0,36	0,76	0,8
Πράξεις κινητής υποδ. (δισ.)						
MA41	0,3	2,3	10,4	2,9	38,2	0,
MA38	3,8	5,3	62,2	9,0	7,0	0,
Χρόνος παραγοντοποίησης						
MA41	8	46	174	81	809	1
MA38	85	127	1255	226	220	1

Πίνακας 7.6: Σύγκριση μεταξύ του «συμμετρικού» (MA41) και «μη συμμετρικού» (MA38) κώδικα.

Τα προβλήματα στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.6 είναι σε σειρά αυξανόμενης ασυμμετρίας, όπου ο δείκτης ασυμμετρίας ορίζεται ως

$$\frac{\text{Αριθμός ζευγαριών ώστε } a_{ij} = 0, a_{ji} \neq 0}{\text{Συνολικός αριθμός μη διαγώνιων στοιχείων}},$$

(**Σφάλμα! Δεν έχει οριστεί στυλ.**.4)

έτσι ώστε ένας συμμετρικός πίνακας να έχει δείκτη ασυμμετρίας 0. Αυτά τα αποτελέσματα δείχνουν καθαρά τα οφέλη από τον απευθείας χειρισμό της ασυμμετρίας.

Στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.7 υπάρχει μία σύγκριση μεταξύ του HSL MA38 κώδικα με τους HSL κώδικες MA41 και MA48, από τους Davis και Duff (1997a). Αυτά τα αποτελέσματα δείχνουν ότι ο νέος κώδικας μπορεί να είναι πολύ ανταγωνιστικός, μερικές φορές δίνοντας καλύτερες επιδόσεις από τους άλλους.

Τάξη	13535	62424	26064	22560	120750	360
Μη-μηδενικές τιμές	390607	1717792	177168	1014951	1224224	227
Δείκτης ασυμμετρίας	0,00	0,00	0,00	0,36	0,76	0,8
Πράξεις κινητής υποδ. (δισ.)						
MA41	0,3	2,3	10,4	2,9	38,2	0,
MA38	3,8	5,3	62,2	9,0	7,0	0,
Χρόνος παραγοντοποίησης						
MA41	8	46	174	81	809	1
MA38	85	127	1255	226	220	1

Πίνακας 7.7: Σύγκριση μεταξύ του MA38 και τους MA48 και MA41 σε μερικούς δοκιμαστικούς πίνακες. Οι χρόνοι σε δευτερόλεπτα σε έναν Sun ULTRA-1 σταθμό εργασίας

8 Εκμετάλλευση του παραλληλισμού - υπερκόμβοι

Αν και η προσέγγιση των πολλαπλών μετώπων είναι ιδιαίτερα κατάλληλη για την εκμετάλλευση του παραλληλισμού, δεν είναι η μόνη προσέγγιση στην οποία γίνεται έρευνα. Πραγματικά, ο αλγόριθμος του Cholesky μπορεί να υλοποιηθεί για αραιά συστήματα και μπορεί επίσης να τμηματοποιηθεί χρησιμοποιώντας μία έκφραση υπερκόμβων παρόμοια με την τεχνική συνένωσης κόμβων του κεφαλαίου 7. Ένας κώδικας βασισμένος σε αυτήν την προσέγγιση πέτυχε πολύ υψηλή επίδοση σε προβλήματα δομικής ανάλυσης και τεχνητά δημιουργημένα προβλήματα σε έναν CRAY Y-MP (Simon et al. 1989). Μία παραλλαγή του τυπικού «στηλοσταφούς» αραιού Cholesky αλγόριθμου έχει επίσης υλοποιηθεί σε hypercubes (George, Heath, Liu και Ng 1988, George, Heath, Liu και Ng 1989). Κώδικες υψηλών επιδόσεων βασισμένους σε παραγοντοποίηση υπερκόμβων για αρχιτεκτονικές πολλών εντολών πολλών δεδομένων (MIMD), συγκεκριμένα για έναν INTEL Paragon, έχουν αναπτυχθεί από τον Rothberg (1994).

Η έννοια των υπερκόμβων επεκτάθηκε πρόσφατα σε μη συμμετρικά συστήματα από τον Demmel, Eisenstat, Gilbert, Li και Liu (1995). Είναι πλέον δυνατόν να χρησιμοποιηθούν αποδοτικά οι πυρήνες BLAS επιπέδου 3. Ωστόσο, οι Demmel et al. (1995) έχουν αναπτύξει μία υλοποίηση που εκτελεί πολλαπλασιασμούς πυκνών πινάκων σε τμήματα διανυσμάτων και, αν και αυτοί δεν μπορούν να γραφτούν ως άλλοι πυκνοί πίνακες, δείχνουν πως αυτή η BLAS επιπέδου 2,5 έχει τα περισσότερα χαρακτηριστικά επίδοσης όπως και η επιπέδου 3 μιας και η επαναλαμβανόμενη χρήση του ίδιου πυκνού πίνακα επιτρέπει την καλή χρήση της λανθάνουσας μνήμης και γενικά της ιεραρχίας της μνήμης. Στον πίνακα Π**Σφάλμα! Δεν έχει οριστεί στυλ.**.1 συγκρίνεται ο κώδικάς τους,

SuperLU, με την προσέγγιση πολλαπλών μετώπων σε ένα εύρος παραδειγμάτων. Ο κώδικας πολλαπλών μετώπων MA41 χρησιμοποιήθηκε στις περιπτώσεις που ο δείκτης ασυμμετρίας ήταν κάτω από 0,5, ο κώδικας MA38 στην αντίθετη περίπτωση.

Πίνακας	Τάξη	Αρ. καταχωρήσεων	Χρόνος ανάλυσης και παραγοντοποίησης		Καταχωρήσεις στους παράγοντες LU (εκ)	
			SuperLU	Multif	SuperLU	Multif
ONETONE2	36057	227628	9	11	1,3	1,3
TWOTONE	120750	1224224	758	221	24,7	9,8
WANG3	26024	177168	1512	174	27,0	11,4
VENKAT50	62424	1717792	172	46	18,0	11,9
RIM	22560	1014951	78	80	9,7	7,4
GARON2	13535	390607	60	8	5,1	2,4

Πίνακας 8.1: Σύγκριση μεταξύ αλγόριθμου πολλαπλών μετώπων και αλγόριθμου υπερκόμβων.

Π**Σφάλμα ! Δεν έχει οριστεί στυλ.**.1

9 Λογισμικό

Αν και υπάρχει πολύ λογισμικό που υλοποιεί τις άμεσες μεθόδους για την επίλυση αραιών γραμμικών συστημάτων, λίγο είναι ελεύθερο ή ανοιχτού κώδικα. Υπάρχουν πολλοί λόγοι για αυτήν την κατάσταση, ο κυριότερος όντας ότι το λογισμικό αραιών πινάκων είναι ενσωματωμένο σε πολύ μεγαλύτερα πακέτα (για παράδειγμα, δομική ανάλυση) και πως πολύ δουλειά για την ανάπτυξη αραιών κωδίκων χρηματοδοτείται εμπορικά έτσι ώστε τα αποτελέσματα αυτής της δουλειάς να απαιτούν άδεια χρήσης. Υπάρχουν αρκετοί ερευνητικοί κώδικες που μπορούν να αποκτηθούν από τους συγγραφείς τους, αλλά συνήθως έχουν λίγη έως ελάχιστη τεκμηρίωση και συχνά χρειάζονται υποστήριξη από τους συγγραφείς.

Μεταξύ του ελεύθερου λογισμικού υπάρχουν κάποιες ρουτίνες από το Collected Algorithms του ACM (διαθέσιμες μέσω του netlib), κυρίως για τον πολλαπλασιασμό πινάκων (για παράδειγμα, μείωση του εύρους ζώνης, διάταξη σε τριγωνική κατά τμήματα μορφή) παρά για την επίλυση εξισώσεων, αν και ο κώδικας NSPIV του Sherman (1978) είναι διαθέσιμος ως Αλγόριθμος 533.

Ο Y12M και ο MA28 της HSL είναι διαθέσιμοι στην netlib, αν και όσοι παίρνουν τον MA28 με αυτόν τον τρόπο είναι υποχρεωμένοι να υπογράψουν μία άδεια χρήσης, και η χρήση του νεότερου MA48 είναι προτιμητέα. Μία ερευνητική έκδοση του MA38, που ονομάζεται UMFPACK και εμπεριέχει μία έκδοση για μιγαδικούς πίνακες, διανέμεται δωρεάν στην netlib όπως επίσης και ο C κώδικας SuperLU που υλοποιεί την παραγοντοποίηση με υπερκόμβους του Demmel et al. (1995). Υπάρχει επίσης και ένας σκελετός κώδικα αραιής LU παραγοντοποίησης των Banks και Smith στην συλλογή

"διάφορα" της netlib, και ο Joseph Liu διανέμει τον κώδικά των πολλαπλών ελαχίστων βαθμών σε κάθε ενδιαφερόμενο.

Μεταξύ των κωδίκων που διατίθενται υπό όρους με άδεια χρήσης είναι αυτοί της Harwell Subroutine Library, ένα υποσύνολο των οποίων προωθούνται από την NAG ως Harwell Sparse Matrix Library. Η βιβλιοθήκη IMSL έχει επίσης κώδικες για την άμεση επίλυση αραιών συστημάτων, και ένας αραιός LU κώδικας είναι διαθέσιμος στην τρέχουσα έκδοση του ESSL για τον IBM RS/6000 και τους υπολογιστές SP2. Κώδικες επίλυση αραιών γραμμικών εξισώσεων είναι επίσης διαθέσιμοι στους χρήστες των υπολογιστών Cray κατόπιν αιτήσεως στην Cray Research Inc.

Άλλα πακέτα περιλαμβάνουν το SPARSPAK πακέτο, που αναπτύσσεται κυρίως στο πανεπιστήμιο του Waterloo (George, Liu και Ng 1980), και που επιλύει γραμμικά προβλήματα και προβλήματα ελαχίστων τετραγώνων, και ρουτίνες στην PORT βιβλιοθήκη των Bell Labs (Kaufmann 1982), λεπτομέρειες των οποίων μπορούν να ληφθούν από την netlib. Εκδόσεις του πακέτου YSMP, προϊόν ανάπτυξης στο Yale University, μπορεί να ληφθεί από την Scientific Computing Associates στο Yale και περιέχει πολλές ρουτίνες που υλοποιούν επαναληπτικές μεθόδους για αραιές εξισώσεις.

Μία ελεύθερη έκδοση του SuperLU για υπολογιστές διαμοιραζόμενης μνήμης, SuperLU_MT, είναι διαθέσιμο από το Berkeley και ο MA41 κώδικας της HSL έχει επίσης μία έκδοση για υπολογιστές διαμοιραζόμενης μνήμης. Για υπολογιστές κατανεμημένης μνήμης υπάρχει ο κώδικας CAPSS των Heath και Raghavan, και βρίσκεται στο πακέτο SCALAPACK. Ο Gupta μοιράζει ελεύθερα την φορητή έκδοση του πηγαίου κώδικα WSSMP που διατίθεται με άδεια χρήσης για τον IBM SP2. Οι Koster και Bisseling σχεδιάζουν μία δημοσίευση του παράλληλου Markowitz/κατωφλίου

επιλυτή SPLU, σχεδιασμένο για αρχιτεκτονικές μεταβίβασης μηνύματος, ενώ το ευρωπαϊκό πρόγραμμα LTR PARASOL εργάζεται στην ανάπτυξη μίας ομάδας άμεσων και επαναληπτικών αραιών επιλυτών για αρχιτεκτονικές μεταβίβασης μηνύματος που απευθύνονται κυρίως σε εφαρμογές πεπερασμένων στοιχείων.

Αυτή η παρουσίαση των υλοποιήσεων των αλγορίθμων για αραιά συστήματα αφορά μόνο κώδικες που είναι ολοκληρωμένοι και υποστηρίζονται πλήρως. Πολλοί άλλοι κώδικες είναι διαθέσιμοι, είτε στο στάδιο της ανάπτυξης, είτε σαν ερευνητικά εργαλεία (για παράδειγμα το πακέτο SMMS του Fernando Alvarado του Wisconsin και το πακέτο SPARSKIT του Yousef Saad στην Minneapolis.

10 Μελλοντικές τάσεις

Από ότι δείχνει η μέχρι τώρα πορεία των πραγμάτων, η ανάγκη για επίλυση ολοένα και μεγαλύτερων προβλημάτων δεν σταματά. Για παράδειγμα κάποια προβλήματα του προγράμματος ASCI των Η.Π.Α. έχουν διαστάσεις πολλών εκατομμυρίων και διασταυρωτές ζώων επιλύουν συστήματα 20 και 30 εκατομμυρίων βαθμών ελευθερίας.

Εμφανώς το μέγεθος του προβλήματος που μπορεί να επιλυθεί με μία άμεση μέθοδο εξαρτάται πολύ από την δομή του πίνακα. Για παράδειγμα, ένα διαγώνια ή τρι-διαγώνιο σύστημα δεν είναι πρόβλημα όταν η διάσταση του πίνακα αυξάνεται και πράγματι, αν το fill-in μπορεί να κρατηθεί χαμηλό, είναι συνήθως δυνατό να λυθούν πολύ μεγάλα προβλήματα με αραιή άμεση παραγοντοποίηση. Ωστόσο, για την διακριτοποίηση τρισδιάστατων μερικών διαφορικών εξισώσεων, οι περιορισμοί των άμεσων μεθόδων γίνονται υπερβολικά εμφανείς. Αν και προβλήματα των διακριτοποιήσεων πεπερασμένων στοιχείων τάξης σχεδόν ένα εκατομμύριο επιλύθηκαν από το MUMPS, σύμφωνα με την γνώμη πολλών η πιο πολλά υποσχόμενη ομάδα τεχνικών για την επίλυση πραγματικά μεγάλων προβλημάτων είναι αυτή που συνδυάζει ταυτόχρονα άμεσες και επαναληπτικές μεθόδους. Αυτό μπορεί να θεωρηθεί και μία προχωρημένη προπαρασκευή για μία επαναληπτική μέθοδο.

Πολλά υποσχόμενες τεχνικές είναι και αυτές που χρησιμοποιούν κατάτμηση γράφων για την υποδιαίρεση του προβλήματος, την επίλυση των τοπικών υποπροβλημάτων με άμεσες μεθόδους και την χρήση μιας επαναληπτικής μεθόδου για το πάντρεμα των τμημάτων στην κατάτμηση. Αυτή η προσέγγιση μοιάζει πολύ με τις μεθόδους που χρησιμοποιούνται στην επίλυση προβλημάτων διακριτοποίησης μερικών

διαφορικών εξισώσεων με χρήση της κατάτμησης πεδίων, που μπορεί να θεωρηθεί και ως αντιμετάθεση του πίνακα στην φραγμένη διαγώνια κατά τμήματα μορφή.

11 Συμπεράσματα

Στην παρούσα εργασία έγινε μία διερεύνηση των αλγορίθμων για την άμεση επίλυση αραιών γραμμικών συστημάτων της μορφής $\mathbf{Ax} = \mathbf{b}$. Ο στόχος των αραιών αλγορίθμων είναι η επίλυση του συστήματος σε χρόνο και χώρο ευθέως ανάλογο της τάξης του πίνακα και του αριθμού των μη-μηδενικών στοιχείων του σε αντίθεση με την τετραγωνική σε χώρος και κυβική σε χρόνο εξάρτηση που έχουν με την τάξη οι πυκνοί κώδικες.

Οι προκλήσεις στις οποίες καλούνται να ανταπεξέλθουν οι αραιοί κώδικες έχουν να κάνουν με τις αυξημένες απαιτήσεις διαχείρισης δεδομένων, με την διατήρηση της αριθμητικής σταθερότητας, με την αποφυγή του fill-in, και με την μεγάλη περιπλοκότητα των εσωτερικών και εξωτερικών βρόγχων τους.

Οι αλγόριθμοι χωρίζονται σε τέσσερις μεγάλες οικογένειες, η κάθε μία όντας μία εξέλιξη και συνθετότερη μορφή της προηγούμενης. Οι γενικοί αλγόριθμοι εκμεταλλεύονται τις διατάξεις αραιότητας. Οι μετωπικοί αλγόριθμοι επιτυγχάνουν καλύτερες επιδόσεις τμηματοποιώντας τον πίνακα έτσι ώστε να χρησιμοποιούν πυκνούς κώδικες στους εσωτερικούς βρόγχους. Επέκταση των μετωπικών είναι οι αλγόριθμοι πολλαπλών μετώπων, με την παράλληλη εργασία στον πίνακα σε πολλά τμήματά του. Τέλος οι αλγόριθμοι υπερκόμβων εργάζονται σε πολλά τμήματα του πίνακα ακολουθώντας έναν γράφο που ελαχιστοποιεί τις αλληλοεπιδράσεις και τον αριθμό των πράξεων.

Κατά κανόνα οι επιδόσεις των αραιών αλγορίθμων είναι τάξεις μεγέθους μεγαλύτερες σε χρόνο και αποθηκευτικό χώρο όταν ο βαθμός πλήρωσης του πίνακα A είναι κάτω του 10%. Λόγω του ότι όλοι οι κώδικες εκτός των γενικών εργάζονται σε

πυκνούς πυρήνες εντός των εσωτερικών τους βρόγχων, υπάρχει η άποψη πως κανένας πυκνός κώδικας δεν θα μπορέσει ποτέ να ξεπεράσει τους αραιούς σε επιδόσεις, μιας και αυτοί επιλέγουν την καλύτερη στρατηγική ανάλογα με τον βαθμό πλήρωσης του πίνακα όσο προχωράει η απαλοιφή κατά Gauss.

12 Βιβλιογραφία

P. Amestoy, I. Duff – "*Memory management issues in sparse multifrontal methods on multiprocessors*" – International Journal of Supercomputer Applications, 1993

R. Boisvert, R. Pozo, K. Remington, R. Barett, J. Dongarra – "*Matrix Market: A Web Resource for Test Matrix Collections*" – National Institute of Standards and Technology, 1996

I. Duff, J. Reid – "*The design of MA48, a code for the direct solution of sparse unsymmetric linear systems of equations*" – ACM Transactions on Mathematical Software, 1996

I. Duff, J. Reid – "*Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems*" – ACM Transactions on Mathematical Software, 1996

I. Duff, J. Reid, A. Erisman – "*Direct Methods for Sparse Matrices*" – Oxford University Press, 1987

I. Duff – "*Sparse Numerical Algebra: Direct Methods and Preconditioning*" – CERFACS, Toulouse, France, 1996

I. Duff – "*Direct Methods*" – CERFACS, Toulouse, France, 1998

I. Duff – "*Matrix Methods*" – Department for Computation and Information, Oxon, 1998

I. Duff – "*The impact of high performance Computing in the solution of linear systems: trends and problems*" – CERFACS, Toulouse, France, 1999

A. George, J. Liu – "*Computer Solution of Large Sparse Positive Definite Systems*" – Englewood Cliffs, New Jersey: Prentice-Hall, 1981

A. Gupta, G. Karypis, V Kumar – "*Highly scalable parallel algorithms for sparse matrix factorization*" – IEEE Transactions on Parallel and Distributed Systems, 1997

X. Li, J. Demmel – "*SuperLU DIST: A Scalable Distributed-Memory Sparse Direct for Unsymmetric Linear Systems*" – ACM Transactions on Mathematical Software, 2003

Y Saad, M Sosonkina – "*Distributed Schur complement techniques for general sparse linear systems*" – SIAM J. Scientific Computing, 1999

K Wu, B Milne – "Survey of packages for large linear systems" – Lawrence Berkeley National Laboratory, Berkeley, 2000

Z. Zlatev, J. Waśniewski, K. Schaumburg – "*Y12M – Solution of large and sparse systems of linear algebraic equations*" – Vol. 121 of *Lecture Notes in Computer Science,* Springer-Verlag, New York, 1981

13 Ευρετήριο πινάκων