

Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
“Επιστήμης και Τεχνολογίας των Υπολογιστών”

Διπλωματική Εργασία

‘‘Επιτάχυνση της οικογένειας αλγορίθμων Spike μέσω τεχνικών επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη’’

ΒΑΣΙΛΕΙΟΣ ΚΑΛΑΝΤΖΗΣ

Τριμελής Επιτροπή

Ευστράτιος Γαλλόπουλος, Επιβλέπων Καθηγητής
Χρήστος Ζαρολιάργκης, Καθηγητής
Ανδρέας Μπουντουβής, Καθηγητής

Ιούλιος 2014, Πάτρα

Περίληψη

Στη παρούσα διπλωματική εργασία ασχολούμαστε με την αποδοτική επίλυση ταινιακών και γενικών, αραιών γραφικών συστημάτων σε παράλληλες αρχιτεκτονικές μέσω της οικογένειας αλγορίθμων Spike. Ζητούμενο είναι η βελτίωση (μείωση) του χρόνου επίλυσης μέσω τεχνικών επίλυσης γραφικών συστημάτων με πολλά δεξιά μέλη.

Πιο συγκεκριμένα, επικεντρωνόμαστε στην επίλυση της εξίσωσης μητρώου

$$AX = F \quad (1)$$

όπου $A \in \mathbb{R}^{n \times n}$ είναι το μητρώο συντελεστών και το οποίο είναι αραιό /και ταινιακό, $F \in \mathbb{R}^{n \times s}$ είναι ένα μητρώο με s στήλες το οποίο ονομάζεται μητρώο δεξιών μελών και $X \in \mathbb{R}^{n \times s}$ είναι η λύση του συστήματος. Μια σημαντική μέθοδος για την παράλληλη επίλυση της παραπάνω εξίσωσης, είναι η μέθοδος Spike και οι παραλλαγές της. Η μέθοδος Spike βασίζεται στη τεχνική διαίρει και βασίλευε και αποτελείται από δύο φάσεις: α) επίλυση ανεξάρτητων υπο-προβλημάτων τοπικά σε κάθε επεξεργαστή, και β) επίλυση ενός πολύ μικρότερου προβλήματος το οποίο απαιτεί επικοινωνία μεταξύ των επεξεργαστών. Οι δύο φάσεις συνδυάζονται ώστε να παραχθεί η τελική λύση X .

Η συνεισφορά της διπλωματικής εργασίας έγκειται στην επιτάχυνση της οικογένειας αλγορίθμων Spike για την επίλυση της εξίσωσης (1) μέσω της μελέτης, το σχεδιασμό και την υλοποίηση νέων, περισσότερο αποδοτικών αλγορίθμων σχημάτων τα οποία βασίζονται σε τεχνικές επίλυσης γραφικών συστημάτων με πολλά δεξιά μέλη. Αυτά τα νέα αλγορίθμικά σχήματα έχουν ως στόχο τη βελτίωση του χρόνου επίλυσης των γραφικών συστημάτων καθώς και άλλα οφέλη όπως η αποδοτικότερη χρήση μνήμης.

Περιεχόμενα

1 Εισαγωγή στην επίλυση αραιών γραφμικών συστημάτων	19
1.1 Επίλυση αραιών γραφμικών συστημάτων: Εφαρμογές και μέθοδοι	19
1.1.1 Οι άμεσες μέθοδοι	20
1.1.2 Οι επαναληπτικές μέθοδοι	22
1.1.3 Η έννοια της προκατάστασης	25
1.1.4 Διάφορες χλάσεις προκαταστατών	27
1.1.5 Αλγόριθμοι αναδιάταξης αραιών μητρώων για τη κατασκευή ταινιακών προκαταστατών	29
1.2 Η παράλληλη επίλυση αραιών γραφμικών συστημάτων και η οικογένεια αλγορίθμων Spike	34
2 Η οικογένεια αλγορίθμων Spike	37
2.1 Επίλυση ταινιακών γραφμικών συστημάτων	37
2.1.1 Ο αλγόριθμος Spike	37
2.1.2 Ο αλγόριθμος TR-Spike	40
2.2 Επίλυση γενικών, αραιών γραφμικών συστημάτων	43
2.2.1 Ο αλγόριθμος PSpike	43
2.2.2 Ο αλγόριθμος WB-Spike	43
3 Επιτάχυνση της οικογένειας αλγορίθμων Spike μέσω τεχνικών επίλυσης γραφμικών συστημάτων με πολλά δεξιά μέλη	47
3.1 Μέθοδοι επίλυσης γραφμικών συστημάτων με πολλά δεξιά μέλη	47
3.1.1 Μέθοδοι σπόρου	48
3.1.2 Μέθοδοι πλοκάδας	54
3.2 Εφαρμογή των μεθόδων πολλών δεξιών μελών στην οικογένεια αλγορίθμων Spike	57
3.2.1 Περίπτωση I: Χαμηλή τάξη των μη-διαγώνιων πλοκάδων	57
3.2.2 Περίπτωση II: Γενικά ταινιακά μητρώα	59
4 Πειράματα και αριθμητικά αποτελέσματα	63
4.1 Επίλυση ταινιακών γραφμικών συστημάτων	63
4.1.1 Τεχνητά μητρώα	63
4.1.2 Μητρώα από τη συλλογή University of Florida sparse matrix collection	65
4.2 Επίλυση γενικών, αραιών γραφμικών συστημάτων	69
4.3 Εφαρμογές στην ανάλυση δεδομένων	74
5 Επίλογος	81

5.1 Προοπτικές για το μέλλον	82
Παράρτημα	83
A' Στοιχεία γραμμικής άλγεβρας	83
A'.1 Συμβολισμοί και ορολογία	83
A'.2 Μητρώα ειδικής δομής	85
A'.3 Νόρμα διανύσματος	86
A'.4 Νόρμα μητρώου	87
A'.5 Η παραγοντοποίηση QR	88
A'.5.1 Gram-Schmidt	89
A'.6 Ιδιοτιμές και ιδιοδιανύσματα	90
A'.7 Η διάσπαση ιδιαζουσών τιμών και ο ψευδοαντίστροφος	91

Κατάλογος Αλγορίθμων

1	Επίλυση του γραμμικού συστήματος $LY = F$ με L χάτω τριγωνικό μητρώο	21
2	Ο αλγόριθμος Conjugate Gradient (CG) [15]	24
3	Ο αλγόριθμος BiCGSTAB [14]	25
4	Ο αλγόριθμος Preconditioned Conjugate Gradient (PCG) [23, 24]	27
5	Ο αλγόριθμος Preconditioned BiCGSTAB [23]	28
6	Ο αλγόριθμος Spike.	38
7	Ο αλγόριθμος TR-SPIKE	42
8	Αλγορίθμικό σχήμα επίλυσης γενικών, αραιών συστημάτων μέσω του αλγορίθμου PSPIKE . .	43
9	Ο αλγόριθμος PSpike - Το βήμα προκατάστασης.	44
10	Ο αλγόριθμος WB-Spike	45
11	Ο γενικός αλγόριθμος των μεθόδων σπόρου (SEED)	49
12	Ο αλγόριθμος Single-Seed Conjugate Gradient (SSCG) [20]	49
13	Ο αλγόριθμος Sparse single-seed Conjugate Gradient (SPARS-SCG) για δεξιά μέλη με μη-μηδενικά στοιχεία στις τελευταίες ζ γραμμές	53
14	Ο αλγόριθμος Block Conjugate Gradient (BCG) [21]	55
15	Επίλυση του γραμμικού συστήματος $KY = L$ με χρήση του αλγορίθμου PROJ	57
16	Η μέθοδος Spike και η περίπτωση των χαμηλής τάξης μη διαγώνιων πλοκών (LR-SPIKE) .	58
17	Η μέθοδος SC-SPIKE	61
18	Η μέθοδος Gram-Schmidt	89
19	Η μέθοδος A -Gram-Schmidt	90

Κατάλογος Σχημάτων

1	Η διάσπαση $A = DS$, παράδειγμα για $p = 3$.	16
1.1	Οπτική αναπαράσταση ενός αραιού μητρώου και ενός αραιού και ταινιακού μητρώου.	20
1.2	Ένα τυχαίο ταινιακό μητρώο $A \in \mathbb{R}^{n \times n}$ ημι-εύρους $m = 100$ και η μη-μηδενική δομή των παραγόντων L, U .	21
1.3	Μη-μηδενική δομή των μητρώων A, L, U της παραγοντοποίησης $A = LU$ για το μητρώο nd3k.	22
1.4	Μη-μηδενική δομή των μητρώων A, L της μη-πλήρους παραγοντοποίησης $A = LU$ για το μητρώο nd3k.	29
1.5	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και Fiedler για το μητρώο USpowerGrid	31
1.6	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και Fiedler για το μητρώο 3elt_dual	31
1.7	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και Fiedler για το μητρώο c-40	32
1.8	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο cage11	33
1.9	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο torso3	33
1.10	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο nd3k	33
1.11	Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο opf10000	34
2.1	Η διάσπαση $A = DS$, παράδειγμα για $p = 3$.	37
3.1	Πλήθος πράξεων μιας επανάληψης του αλγορίθμου sscg συναρτήσει των $\text{nnz}(A)/n$ και s .	50
3.2	Ταινιακό μητρώο A και η απόσβεση στα στοιχεία του A^{-1} .	52
3.3	Δείκτης κατάστασης του μητρώου α κατά την εκτέλεση του αλγορίθμου 16.	55
4.1	Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ για ένα μητρώο Toeplitz με μεταβλητό ημι-εύρος και βαθμό διαγώνιας κυριαρχίας.	64
4.2	Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ για ένα μητρώο Toeplitz με μεταβλητό ημι-εύρος και βαθμό διαγώνιας κυριαρχίας.	65
4.3	Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ ($p = 2, 4$) για τα μητρώα BenElechil και mc2depi.	66
4.4	Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ ($p = 2, 4$) για τα μητρώα shipsec5 και kim2.	66
4.5	Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ ($p = 2, 4$) για τα μητρώα af5k101 και boneS01.	67
4.6	Οι ιδιάζουσες τιμές των Spike-πλοκάδων V_1	72
4.7	Οι ιδιάζουσες τιμές των spike-πλοκάδων V_1	73
4.8	Μέσο σχετικό σφάλμα της προσέγγισης του $d(A^{-1})$ μέσω στοχαστικής εκτίμησης. Τα γραφήματα αποτελούν μέρος της εργασίας [27].	75
4.9	Αριθμητική τιμή των ιδιάζουσων τιμών του υπομητρώου B_1^M .	76
4.10	FLOPS των μεθόδων για $n = 2 \times 10^5, 3 \times 10^5$ και $\kappa = 2, 3$.	78

Κατάλογος Πινάκων

3.1 Πλήθος βασικών πράξεων μίας επανάληψης του αλγορίθμου SSCG. Σε αγκύλες το συνολικό αριθμητικό κόστος. Παρόμοια στοιχεία μπορούν να βρεθούν και στο [39].	50
4.1 Χαρακτηριστικά των μητρώων από τη συλλογή [34]	65
4.2 Χαρακτηριστικά των μητρώων από τη συλλογή [34]	67
4.3 Πλήθος μη-μηδενικών τιμών του παράγοντα L στη πλήρη και μη-πλήρη παραγοντοποίηση.	68
4.4 Χρόνοι εκτέλεσης του αλγορίθμου SPIKE για διαφορετικό πλήθος διαχωρισμάτων ($s = 10$).	68
4.5 Χρόνοι εκτέλεσης της μεθόδου Spike για διαφορετικό πλήθος διαχωρισμάτων ($s = 20$).	69
4.6 Πλήθος MV γινομένων με ή χωρίς προκατάσταση.	69
4.7 Χαρακτηριστικά των μητρώων από τη συλλογή [34]	70
4.8 Ήμι-εύρος m και απαιτούμενα ημι-εύρη \hat{m}_1 , \hat{m}_2 ώστε $\ M\ _F \geq 0.95 * \ A\ _F$ και $\ M\ _F \geq 0.99 * \ A\ _F$ ύστερα από τις συμμετρικές αναδιατάξεις.	70
4.9 Χρόνοι εκτέλεσης για $n = 2 \times 10^5$ και $\kappa = 2, 3$	77
4.10 Χρόνοι εκτέλεσης για $n = 3 \times 10^5$ και $\kappa = 2, 3$	77
A'.1 Πολυπλοκότητα πολλ/συμού μητρώων.	84

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν κατά τη διάρκεια των μεταπτυχιακών μου σπουδών αλλά και στην υλοποίηση της παρούσας διπλωματικής εργασίας.

Αρχικά θα ήθελα να ευχαριστήσω το καθηγητή μου Ευστράτιο Γαλλόπουλο για όλη τη βοήθεια που απλόχερα μου προσέφερε καθώς και για τις συνθήκες εργασίας τις οποίες διασφάλισε στο εργαστήριο. Ο Κ. Γαλλόπουλος πάντα ενδιαφερόταν για την εξέλιξή μου και κατέβαλε τις μέγιστες δυνατές προσπάθειες ώστε να γνωρίσω την ακαδημαϊκή ζωή εκ των έσω. Προσπαθούσε πάντοτε να με συστήνει σε νέους ανθρώπους και με την εν γένει στάση του με βοήθησε να αναπτύξω χαρακτηριστικά τα οποία θεωρώ εφόδια για το μέλλον.

Θα ήθελα επίσης να ευχαριστήσω τους Κ. Καθηγητές Χ. Ζαφολιάγκη και Α. Μπουντουβή οι οποίοι υπηρέτησαν ως μέλη της τριμελούς επιτροπής μου. Τους ευχαριστώ για τα σχόλια τα οποία παρέθεσαν καθώς και για την υπομονή τους κατά τη διάρκεια της τελικής εξέτασης.

Κατά τη διάρκεια της παρουσίας μου στο Πανεπιστήμιο Πατρών υπήρξαν πολλοί συνάδελφοι χάρη στους οποίους η καθημερινότητα μου έγινε λίγο πιο εύκολη. Ιδιαιτέρως θα ήθελα να ευχαριστήσω την Ευγενία της οποίας η βοήθεια και συμπαράσταση σε διάφορα στάδια της μέχρι τώρα ακαδημαϊκής μου πορείας υπήρξε (και συνεχίζει να είναι) ανεκτίμητη.

Τέλος, θέλω να ευχαριστήσω το Καθηγητή A. Sameh καθώς και τον ερευνητή F. Saied, οι οποίοι με δέχθηκαν στο πανεπιστήμιο Purdue για δύο εβδομάδες προκειμένου να εξοικειωθώ με τον αλγόριθμο Spike, και συνεισέφεραν σημαντικά στη κατεύθυνση της παρούσας εργασίας. Είναι αυτονόητο ότι οι εμπειρίες τις οποίες απέκτησα κατά την επίσκεψή μου εκεί ήταν (και είναι) κάτι παραπάνω από υπερπολύτιμες.

Βασίλης Καλαντζής
Πάτρα, Ιούλιος 2014

Συνεισφορά της εργασίας

Η συνεισφορά της παρούσας εργασίας έγκειται στη διερεύνηση και υλοποίηση νέων αλγορίθμικών σχημάτων για την επιτάχυνση της οικογένειας αλγορίθμων Spike για την επίλυση της γραμμικής εξίσωσης μητρώου

$$AX = F, \quad (2)$$

όπου $A \in \mathbb{R}^{n \times n}$ είναι το μητρώο συντελεστών και το οποίο είναι αραιό ή/και ταινιακό, $F \in \mathbb{R}^{n \times s}$ είναι ένα μητρώο με s στήλες το οποίο ονομάζεται μητρώο δεξιών μελών και $X \in \mathbb{R}^{n \times s}$ η λύση του συστήματος (1). Συστήματα αυτής της μορφής προκύπτουν σε πλήθος επιστημονικών εφαρμογών όπως στα πεδία μοντελοποίησης διάχυσης θερμότητας, πρόβλεψης καιρικών φαινομένων, ρευστοδυναμικής, προσομοίωσης ηλεκτρικών κυκλωμάτων, ανάλυσης δομικών στοιχείων, κβαντομηχανικής, ηλεκτρομαγνητικής καθώς και ως εργαλείο για την επίλυση προβλημάτων ιδιοτιμών.

Ο πιο γνωστός αλγόριθμος επίλυσης του (1) είναι η απαλοιφή Gauss. Η επίλυση μέσω απαλοιφής Gauss αποτελεί στην ουσία προσίμιο της επίλυσης της εξίσωσης (1) μέσω διάσπασης του A ως γινόμενο δύο ή περισσότερων μητρώων. Η περισσότερο διαδεδομένη διάσπαση (παραγοντοποίηση) του A εκφράζεται ως

$$PA = LU \quad (3)$$

με $P, L, U \in \mathbb{R}^{n \times n}$ όπου L είναι κάτω τριγωνικό μητρώο με μονάδες στη κύρια διαγώνιο, U άνω τριγωνικό μητρώο και P μητρώο μετάθεσης αντίστοιχα. Αφού η διάσπαση του A στη μορφή (3) λάβει χώρα, στη συνέχεια η επίλυση της εξίσωσης (1) πραγματοποιείται μέσω της επίλυσης των απλούστερων υποπροβλημάτων

$$F = PF, \quad LY = F, \quad UX = Y. \quad (4)$$

Λόγω των μητρώων συντελεστών μεγάλου μεγέθους τα οποία προκύπτουν από τις σύγχρονες εφαρμογές (μητρώα μεγέθους μερικών δεκάδων εκατομμυρίων δε θεωρούνται πλέον “μεγάλου” μεγέθους) αλλά και της ανάγκης για την τάχιστη επίλυση των γραμμικών συστημάτων, οι αριθμητικές μέθοδοι επίλυσης υλοποιούνται σε περιβάλλοντα τα οποία χρησιμοποιούν πλήθος επεξεργαστών/πυρήνων οπότε και η επίλυση λαμβάνει χώρα παράλληλα. Ο σχεδιασμός και η υλοποίηση μεθόδων που λαμβάνουν υπόψη τους τη παραλληλία του υπολογιστικού συστήματος είναι ένα σύνθετο πρόβλημα το οποίο εισάγει νέες έννοιες όπως αυτές της κλιμάκωσης, της ισορροπίας υπολογιστικού φόρτου καθώς και την ανάγκη επικοινωνίας μεταξύ των διαφόρων υπολογιστικών συστημάτων, και αποτελεί μια πρόκληση για την ερευνητική κοινότητα η οποία δραστηριοποιείται στην αντιμετώπιση των παραπάνω επιπρόσθετων παραμέτρων.

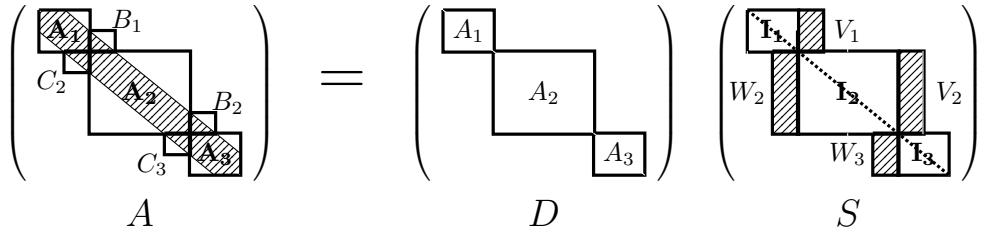
Στη παρούσα εργασία θα ασχοληθούμε με μια μέθοδο επίλυσης η οποία στοχεύει ακριβώς στην αποδοτική επίλυση του (1) σε παράλληλα περιβάλλοντα όταν το μητρώο A είναι μεγάλων διαστάσεων και ταινιακό.

αλγόριθμος αυτός ονομάζεται αλγόριθμος Spike. Η βασική μας συνεισφορά έγκειται στη περαιτέρω βελτίωση της μεθόδου Spike μέσω τεχνικών επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη.

Η βασική ιδέα της μεθόδου Spike έγκειται στη παραγονοτποίηση του μητρώου A ως

$$A = DS \quad (5)$$

με $D \in \mathbb{R}^{n \times n}$ μητρώο διαγώνιο κατά πλοκάδες και $S \in \mathbb{R}^{n \times n}$ μητρώο τριδιαγώνιο κατά πλοκάδες. Παρακάτω παραθέτουμε σχηματικά μια τέτοια διάσπαση. Στο σχήμα, το οποίο θα συναντήσουμε και αργότερα, στην ανάλυση πλέον του αλγορίθμου Spike, για απλότητα έχουμε ότι κάθε διαγώνια πλοκάδα ανατίθεται σε ένα επεξεργαστή. Αυτό φυσικά δεν αποτελεί πάντα τη βέλτιστη επιλογή αφού κάποιες διαγώνιες πλοκάδες δύναται να είναι μεγαλύτερες από κάποιες άλλες, όπως η πλοκάδα A_2 στο σχήμα που παραθέτουμε. Στη πραγματικότητα παραπάνω από ένας επεξεργαστές μπορούν να ανατεθούν για το χειρισμό κάποιας διαγώνιας πλοκάδας, το σχήμα παραθέτει τη γενική ιδέα και μόνον.



Σχήμα 1: Η διάσπαση $A = DS$, παράδειγμα για $p = 3$.

Η μέθοδος Spike χωρίζεται σε δύο στάδια, το στάδιο προεπεξεργασίας και το στάδιο μεταεπεξεργασίας. Αν υποθέσουμε ότι έχουμε στη διάθεσή μας p επεξεργαστές, τότε κατά το στάδιο προεπεξεργασίας κάθε επεξεργαστής σχηματίζει και επιλύει τοπικά ένα γραμμικό σύστημα με πολλά δεξιά μέλη (και πάλι χάριν απλότητας θεωρούμε ότι κάθε διαγώνια πλοκάδα συνδέεται με έναν και μόνο επεξεργαστή). Η επίλυση αυτών των συστημάτων πραγματοποιείται μέσω άμεσων μεθόδων, παραμένει όμως το ερώτημα αν αυτό είναι πάντοτε η βέλτιστη λύση. Όπως θα δούμε στην εργασία αυτή, η απάντηση είναι ότι -υπό προϋποθέσεις μπορούμε να κάνουμε κάτι καλύτερο με χρήση επαναληπτικών μεθόδων. Το στάδιο μεταεπεξεργασίας είναι το ίδιο ανεξαρτήτως της μεθόδου επίλυσης στο στάδιο προεπεξεργασίας και επομένως δεν επικεντρωνόμαστε ιδιαίτερα σε αυτό.

Επίσης, η μέθοδος Spike μπορεί να χρησιμοποιηθεί ως προκαστάτης για την επίλυση γενικών αραιών γραμμικών συστημάτων όπως στο (1). Σε αυτό το πλαίσιο, το μητρώο A αναδιατάσσεται και εξάγεται ένας ταυτιακός προκαταστάτης, η εφαρμογή του οποίου πραγματοποιείται μέσω του αλγορίθμου Spike. Η ευελιξία της μεθόδου Spike έχει οδηγήσει σε διάφορα αλγορίθμικά σχήματα ειδικού σκοπού τα οποία απαρτίζουν την οικογένεια αλγορίθμων Spike.

Ο στόχος της παρούσας εργασίας είναι η διερεύνηση και υλοποίηση νέων αλγορίθμων σχημάτων για την επιτάχυνση της οικογένειας αλγορίθμων Spike για την επίλυση της γραμμικής εξίσωσης (1) μέσω τεχνικών επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη. Πιο συγκεκριμένα, η επίλυση των γραμμικών συστημάτων τοπικά σε κάθε επεξεργαστή κατά τη διάρκεια της φάσης προεπεξεργασίας του αλγορίθμου Spike, υλοποιείται τώρα (από τη μελέτη της βιβλιογραφίας, αυτό φαίνεται ότι γίνεται για πρώτη φορά) μέσω επαναληπτικών σχημάτων τα οποία προσπαθούν να “διαμοιράσουν” τη πληροφορία που δημιουργείται κατά τη διάρκεια επίλυσης έτσι ώστε να επιτευχθεί ταχύτερη συνολική επίλυση. Παραθέτουμε δύο βασικές κατηγορίες

μεθόδων επίλυσης γραφικών συστημάτων με πολλά δεξιά μέλη, τις μεθόδους σπόρου και τις μεθόδους πλοκάδας και στη συνέχεια σχολιάζουμε τα πλεονεκτήματα και μειονεκτήματα της κάθε κατηγορίας. Στη συνέχεια, εξετάζουμε το κατά πόσο αυτές οι μέθοδοι μπορούν να συνδυαστούν με την οικογένεια αλγορίθμων Spike. Παραθέτουμε επίσης καινοτόμα σχήματα βάσει αυτών των μεθόδων. Τέλος, εξετάζουμε το πως αυτές οι μέθοδοι μπορούν να υλοποιηθούν αποδοτικά στο περιεχόμενο της οικογένειας αλγορίθμων Spike. Παραθέτουμε αριθμητικά πειράματα τα οποία στόχο έχουν να αναδείξουν τα οφέλη (αλλά και τις αδυναμίες) που προκύπτουν από αυτό το συνδυασμό.

Η δομή της εργασίας είναι οργανωμένη ως εξής: Το Κεφάλαιο 1 παρουσιάζει αναλυτικά το πρόβλημα το οποίο θέλουμε να επιλύσουμε και παραθέτει τις βασικές κατηγορίες αλγορίθμων επίλυσης. Το Κεφάλαιο 2 συνοψίζει τα κύρια σημεία της οικογένειας αλγορίθμων Spike και παραθέτει τα σημαντικότερα αλγορίθμικά σχήματα. Το Κεφάλαιο 3 παρουσιάζει συνοπτικά τις βασικές μεθόδους επίλυσης γραφικών συστημάτων με πολλά δεξιά μέλη και εισάγει νέα αλγορίθμικά σχήματα τα οποία βασίζονται σε αυτές τις μεθόδους. Σε αυτό το κεφάλαιο αναλύουμε πως αυτές οι τεχνικές μπορούν να βελτιώσουν την επίλυση γραφικών εξισώσεων μέσω της μεθόδου Spike. Τέλος, το Κεφάλαιο 4 παρουσιάζει αριθμητικά πειράματα μεταξύ επιλεγμένων αλγορίθμων όπου και σχολιάζουμε τα αποτελέσματα.

Επίσης, στο τέλος της εργασίας, παραθέτουμε παράρτημα σχετικό με το βασικό υπόβαθρο γραφικής άλγεβρας που χρησιμοποιείται.

Κεφάλαιο 1

Εισαγωγή στην επίλυση αραιών γραμμικών συστημάτων

1.1 Επίλυση αραιών γραμμικών συστημάτων: Εφαρμογές και μέθοδοι

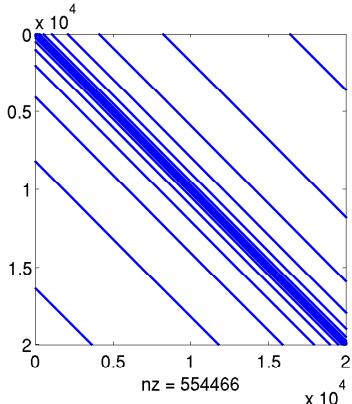
Στη παρούσα εργασία ασχολούμαστε με την αποδοτική επίλυση της εξίσωσης μητρώων (στο εξής απλώς εξίσωση)

$$AX = F, \quad (1.1)$$

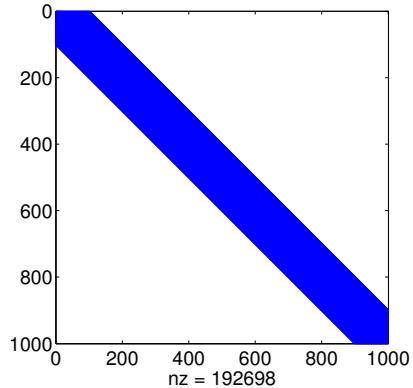
όπου $A \in \mathbb{R}^{n \times n}$ το αραιό μητρώο συντελεστών, $F \in \mathbb{R}^{n \times s}$ το μητρώο δεξιών μελών και $X \in \mathbb{R}^{n \times s}$ το μητρώο λύση της εξίσωσης. Στη συνέχεια της εργασίας, ο όρος αποδοτική επίλυση κυρίως θα αναφέρεται στη γρήγορη (σε χρονικούς όρους) επίλυση της εξίσωσης (1).

Συστήματα αυτής της μορφής προκύπτουν σε πλήθος επιστημονικών εφαρμογών όπως στα πεδία μοντελοποίησης διάχυσης θερμότητας, πρόβλεψης καιρικών φαινομένων, ρευστοδυναμικής, προσομοίωσης ηλεκτρικών κυκλωμάτων, ανάλυσης δομικών στοιχείων, κβαντομηχανικής, ηλεκτρομαγνητικής καθώς και ως εργαλείο για την επίλυση προβλημάτων ιδιοτυπών. Τα μητρώα συντελεστών τα οποία προκύπτουν σε αυτές τις εφαρμογές είναι αραιά. Επιπροσθέτως, σε εφαρμογές όπως η ρευστοδυναμική τα μητρώα συντελεστών πολλές φορές είναι και ταινιακά, δηλαδή όλα τα μη-μηδενικά στοιχεία του A βρίσκονται μεταξύ της m_u -στης και m_l -στης υπερδιαγωνίου και υποδιαγωνίου αντίστοιχα, όπου $m_u, m_l \leq n$. Στη παρούσα εργασία, όταν αναφερόμαστε σε ταινιακά μητρώα συντελεστών, οι ποσότητες m_u και m_l θα είναι ίσες. Θα λέμε τότε ότι το μητρώο συντελεστών A έχει ημι-εύρος m και εύρος $2m + 1$. Η περιοχή του ταινιακού μητρώου η οποία περιέχει τα μη-μηδενικά στοιχεία θα αναφέρεται και ως μπάντα.

Λόγω του πλήθους των εφαρμογών όπου συναντάται η εξίσωση (1) καθίσταται προφανές ότι αλγόριθμοι οι οποίοι επιλύουν αποδοτικά την εξίσωση (1) είναι ιδιαίτερα σημαντικοί. Διαχρίνουμε δύο κατηγορίες επίλυσης, τις άμεσες μεθόδους και τις επαναληπτικές μεθόδους.



(α) Ένα αραιό μητρώο, το μητρώο Trefethen 20000.



(β) Ένα ταινιακό μητρώο.

Σχήμα 1.1: Οπτική αναπαράσταση ενός αραιού μητρώου και ενός αραιού και ταινιακού μητρώου.

1.1.1 Οι άμεσες μέθοδοι

Η κλασική μέθοδος επίλυσης ενός αραιού γραμμικού συστήματος $AX = F$ βασίζεται στη παραγοντοποίηση LU, η οποία παραγοντοποιεί το μητρώο $A \in \mathbb{R}^{n \times n}$ ως

$$PA = LU \quad (1.2)$$

όπου $L \in \mathbb{R}^{n \times n}$ είναι κάτω τριγωνικό μητρώο με μονάδες στη κύρια διαγώνιο, $U \in \mathbb{R}^{n \times n}$ άνω τριγωνικό μητρώο και $P \in \mathbb{R}^{n \times n}$ μητρώο μετάθεσης αντίστοιχα. Αφού η διάσπαση του A στη μορφή (1.2) λάβει χώρα, στη συνέχεια η επίλυση της εξίσωσης (1) πραγματοποιείται μέσω της επίλυσης των απλούστερων υποπροβλημάτων

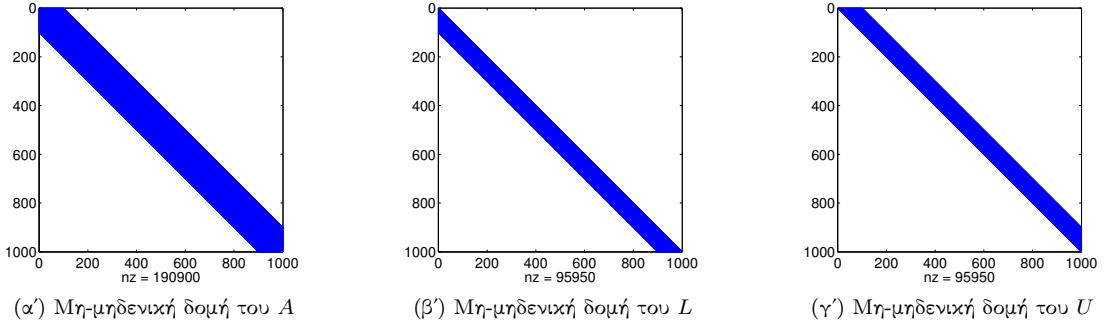
$$F = PF, \quad LY = F, \quad UX = Y. \quad (1.3)$$

Στη συνέχεια, εκτός αν αναφέρεται διαφορετικά, θα υποθέτουμε ότι το μητρώο P ισούται με το ταυτοικό μητρώο μεγέθους n , I_n . Επίσης, όταν το μητρώο είναι συμμετρικό και θετικά ορισμένο, η παραγοντοποίηση LU μπορεί να αντικατασταθεί από τη παραγοντοποίηση Cholesky [18] η οποία έχει τη μορφή

$$A = \bar{L}\bar{L}^\top \quad (1.4)$$

με $\bar{L} \in \mathbb{R}^{n \times n}$ κάτω τριγωνικό μητρώο (χωρίς απαραίτητα μονάδες στη κύρια διαγώνιο). Η παραγοντοποίηση Cholesky είναι οικονομικότερη από τη παραγοντοποίηση LU τόσο από άποψη υπολογιστικής πολυπλοκότητας όσο και από άποψη πολυπλοκότητας μνήμης. Στη συνέχεια, όταν το μητρώο συντελεστών είναι συμμετρικό και θετικά ορισμένο, η άμεση μέθοδος επίλυσης θα είναι πάντοτε η μέθοδος Cholesky.

Το κόστος αριθμητικών πράξεων (πολυπλοκότητα πράξεων) και πολυπλοκότητας μνήμης της LU παραγοντοποίησης ενός αραιού μητρώου $A \in \mathbb{R}^{n \times n}$ δε μπορεί να υπολογιστεί εκ των προτέρων εκτός και αν η δομή των μητρώων L και U μπορεί να καθοριστεί πριν η παραγοντοποίηση λάβει χώρα, όπως συμβαίνει για παράδειγμα για συμμετρικά θετικά ορισμένα μητρώα συντελεστών. Ιδανικά, τό κόστος αυτό θα πρέπει να είναι της τάξης των μή-μηδενικών στοιχείων του A , $\text{nnz}(A)$. Στη περίπτωση ταινιακών μητρώων συντελεστών ημι-εύρους m η υπολογιστική πολυπλοκότητα της παραγοντοποίησης φράσσεται από τη ποσότητα $2nm^2$ ενώ αυτό της μνήμης (πολυπλοκότητα μνήμης) φράσσεται από τη ποσότητα $2nm$ [35].



Σχήμα 1.2: Ένα τυχαίο ταινιακό μητρώο $A \in \mathbb{R}^{n \times n}$ ημι-εύρους $m = 100$ και η μη-μηδενική δομή των παραγόντων L, U .

Πράγματι, είναι εύκολο να διαπιστωθεί ότι οι L, U παράγοντες ενός ταινιακού μητρώου είναι και αυτοί ταινιακοί ημι-εύρους m (υποθέτοντας ότι δε χρειάστηκε οδήγηση κατά τη διάρκεια της παραγοντοποίησης) προς τη μια κατεύθυνση μόνον. Το Σχήμα 1.2 αναπαριστά τη μη-μηδενική δομή ενός ταινιακού, αυστηρά διαγώνια κυρίαρχου μητρώου ημι-εύρους $m = 100$ καθώς και αυτή των τριγωνικών παραγόντων L, U . Ως σύγκριση, η υπολογιστική πολυπλοκότητα παραγοντοποίησης πυκνών μητρώων είναι της τάξης του $O(n^3)$ και η αντίστοιχη πολυπλοκότητα μνήμης της τάξης του $O(n^2)$.

Αφού η LU παραγοντοποίηση του μητρώου συντελεστών πραγματοποιηθεί, ύστα πρέπει να επιλύσουμε τα απλούστερα υποπροβλήματα στο (1.3). Ο Αλγόριθμος 1 περιέχει τον αλγόριθμο επίλυσης του υποπροβλήματος $LY = F$ (αλγόριθμος εμπρός αντικατάστασης) όπου χάριν πληρότητος θεωρούμε ότι L είναι ένα γενικό, κάτω τριγωνικό μητρώο. Φυσικά στη περίπτωση αραιών μητρώων αγνοούμε τις πράξεις όπου υπάρχει μόνο ένας μή-μηδενικός τελεστής (πχ. πολλά/σημερινά με 0). Ο αλγόριθμος επίλυσης του συστήματος $UX = Y$ είναι παρόμοιος, μόνο που στη περίπτωση αυτή η αντικατάσταση ξεκινάει από κάτω προς τα πάνω (όπισθεν αντικατάσταση).

Αλγόριθμος 1 Επίλυση του γραμμικού συστήματος $LY = F$ με L κάτω τριγωνικό μητρώο

Είσοδοι: $L \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$

Έξοδοι: $Y \in \mathbb{R}^{n \times s}$

```

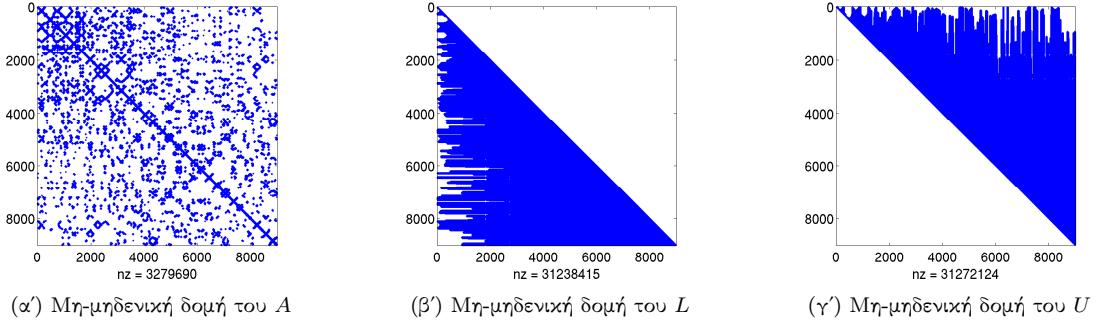
for  $j = 1, \dots, s$  do
     $y_1^{(j)} = f_1^{(j)} / \ell_{11}$ 
    for  $i = 2, \dots, n$  do
         $y_i^{(j)} = (f_i^{(j)} - \sum_{c=1}^{i-1} \ell_{ic} y_c^{(j)}) / \ell_{ii}$ 
    end for
end for

```

Το κόστος της εμπρός αντικατάστασης είναι $O(n^2)$ στη γενική περίπτωση των πυκνών μητρώων, και $O(nm)$ στη περίπτωση κάτω τριγωνικών μητρώων ημι-εύρους m . Αντιστοίχως, τα ίδια ισχύουν και για το κόστος της όπισθεν αντικατάστασης στη γενική και ταινιακή περίπτωση αντίστοιχα.

Οι άμεσες μέθοδοι έχουν τη δυνατότητα να βρουν την ακριβή λύση του γραμμικού συστήματος $AX = F$ σε συγκεκριμένο αριθμό βημάτων, ωστόσο η πρακτική τους εφαρμογή εξαρτάται από το κόστος και τις απαιτήσεις μνήμης της παραγοντοποίησης του A . Αν η μη-μηδενική δομή του A δεν είναι η επιθυμητή, το πλήρος των

μη-μηδενικών στοιχείων των επιμέρους μητρώων της παραγοντοποίησης ενδέχεται να ξεπεράσουν κατά πολύ αυτό του A , οδηγώντας έτσι σε μεγάλες απαιτήσεις μνήμης. Αυτό το φαινόμενο ονομάζεται “γέμισμα” (fill-in).



Σχήμα 1.3: Μη-μηδενική δομή των μητρώων A, L, U της παραγοντοποίησης $A = LU$ για το μητρώο nd3k.

Το Σχήμα 1.3 απεικονίζει τη μη-μηδενική δομή του μητρώου nd3k (ως μητρώο A) και των παραγόντων L, U που προκύπτουν από τη παραγοντοποίηση $A = LU$. Όπως μπορούμε εύκολα να διαπιστώσουμε, το πλήθος των μη-μηδενικών στοιχείων του A ($\text{nnz}(A)=3279690$) είναι πολύ μικρότερο από αυτό των L και U ($\text{nnz}(L)=31238415$, $\text{nnz}(U)=31272124$).

Κλείνοντας το υποκεφάλαιο αξίζει να σημειώσουμε ότι, όταν η χρήση τους είναι πρακτική και το γέμισμα στους επιμέρους παράγοντες δεν είναι εκτεταμένο, οι άμεσες μέθοδοι είναι η συνήθης επίλογή για την επίλυση γραμμικών συστημάτων. Σημειώνουμε μια υλοποίηση διαφόρων αλγορίθμων για τις άμεσες μεθόδους -θα αναφερθούμε και στη συνέχεια- τον άμεσο επίλυτη Pardiso [17] ο οποίος είναι μια πολυνηματική υλοποίηση.

Τέλος, πρέπει να σημειωθεί ότι από τη στιγμή που η παραγοντοποίηση του A έχει ολοκληρωθεί, αυτή μπορεί να χρησιμοποιηθεί για την επίλυση οποιουδήποτε δεξιού μέλους, ασχέτως αν εκείνο δεν είναι διαθέσιμο την ίδια χρονική στιγμή. Αυτό το πλεονέκτημα, της επαναχρησιμοποίησης του υπολογιστικού κόστους της παραγοντοποίησης, κάνει τις άμεσες μεθόδους ακόμα πιο ελκυστικές (όταν, επαναλαμβάνουμε, η παραγοντοποίηση είναι πρακτική).

1.1.2 Οι επαναληπτικές μέθοδοι

Οι επαναληπτικές μέθοδοι επίλυσης μπορούν να κατηγοριοποιηθούν σε δύο υποκατηγορίες: α) τις μεθόδους σταθερού σημείου, πχ. Jacobi, Gauss-Seidel, SOR [18], και β) τις επαναληπτικές μεθόδους υποχώρου Krylov, πχ. BiCGSTAB [14], Conjugate Gradient [15]. Οι μέθοδοι αυτές διαφέρουν από τις άμεσες μεθόδους στο ότι προσεγγίζουν τη λύση μέσω ενδιάμεσων βημάτων σε αντίθεση με τις άμεσες μεθόδους που παράγουν τη τελική λύση στη μέγιστη δυνατή ακρίβεια και μόνο αφού ολοκληρωθεί η διαδικασία επίλυσης. Το τελευταίο συνήθως δεν είναι απαραίτητο και μόνο μερικά σημαντικά ψηφία της λύσης X χρειάζονται. Έτσι η οικονομία μπορεί να είναι ιδιαίτερα μεγάλη στη περίπτωση που η επαναληπτική μέθοδος μπορεί να προσεγγίσει γρήγορα την επιθυμητή ακρίβεια και/ή η άμεση μέθοδος είναι χρονοβόρα.

Οι επαναληπτικές μέθοδοι, εκτός του μητρώου A και του δεξιού μέλους F απαιτούν μια αρχική προσέγγιση

της λύσης (μπορεί να είναι το μηδενικό διάνυσμα) x_0 και μια μεταβλητή η οποία επιδεικνύει την επιθυμητή ακρίβεια στη προσέγγιση της λύσης, έστω tol .

1.1.2.1 Μέθοδοι σταθερού σημείου

Οι επαναληπτικές μέθοδοι σταθερού σημείου αποτελούν τις πιο παλιές επαναληπτικές μεθόδους επίλυσης γραμμικών συστημάτων. Ονομάζονται μέθοδοι σταθερού σημείου επειδή η επαναληπτική διαδικασία για την επίλυση του γραμμικού συστήματος $Ax = f$ έχει τη μορφή

$$x_{k+1} = H(x_k) \quad (1.5)$$

όπου x_i συμβολίζει τη προσέγγιση της λύσης x στην i -στή επανάληψη και H ένας τελεστής. Για να δούμε τι μορφή έχει αυτός ο τελεστής, γράφουμε το A ως

$$A = M - K \quad (1.6)$$

με αντιστρέψιμο M , το οποίο οδηγεί στην εξίσωση

$$Mx = Kx + b \quad (1.7)$$

ή ισοδύναμα

$$x = x + M^{-1}(f - Ax) \quad (1.8)$$

όπου αντικαταστήσαμε το K με το ισοδύναμο του και πολλ/με από τα αριστερά με το μητρώο M^{-1} . Καταλήγουμε τέλος ότι

$$x_{k+1} = M^{-1}Kx_k + M^{-1}b \quad (1.9)$$

όπου για το σφάλμα $x - x_{k+1}$ ισχύει ότι

$$x - x_{k+1} = M^{-1}K(x - x_k), k \geq 0 \quad (1.10)$$

και επομένως είναι προφανές ότι η σύγκλιση της επαναληπτικής μεθόδου, ισοδύναμη με τη συνθήκη $x - x_{k+1} \rightarrow 0$, είναι εγγυημένη αν $\|M^{-1}K\| < 1$ για κάποια νόρμα.

Οι μέθοδοι σταθερού σημείου έχουν πλέον ζεπεραστεί στο σύνολο τους από μια άλλη υποκατηγορία μεθόδων τις οποίες θα δούμε αμέσως τώρα.

1.1.2.2 Μέθοδοι υποχώρου Krylov

Οι επαναληπτικές μέθοδοι υποχώρου Krylov λαμβάνουν ως είσοδο μια αρχική προσέγγιση της λύσης του γραμμικού συστήματος $Ax = f$ και στη συνέχεια, μέσω διαδοχικών επαναλήψεων, υπολογίζουν μια προσέγγιση της λύσης x . Το πόσο αποδοτικές μπορούν να είναι αυτές οι μέθοδοι εξαρτάται από τις ιδιότητες του προβλήματος, με κυριότερη αυτή της κατανομής του φάσματος του μητρώου A .

Ορισμός 1.1. Έστω το γραμμικό σύστημα $Ax = f$ με $A \in \mathbb{R}^{n \times n}$, $f \in \mathbb{R}^n$ όπου με x_0 συμβολίζουμε την αρχική προσέγγιση της λύσης και με $r_0 = f - Ax_0$ συμβολίζουμε το αρχικό κατάλοιπο. Θα ονομάζουμε Krylov υπόχωρο, διάστασης m , τον υπόχωρο \mathcal{K}_m που σχηματίζεται βάσει του

$$\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}.$$

Οι επαναληπτικές μέθοδοι υποχώρου Krylov, στο βήμα i , παράγουν μια προσέγγιση της λύσης x , x_i , για την οποία ισχύει ότι

$$x_i \in x_0 + \mathcal{K}_i, \quad r_i \perp \mathcal{C}_i$$

όπου \mathcal{C}_i είναι ένας κατάλληλα ορισμένος υπόχωρος, διάστασης i .

Οι επαναληπτικές μέθοδοι, σε αντίθεση με τις άμεσες μεθόδους, έχουν ελάχιστες απαιτήσεις μνήμης ωστόσο η σύγκλιση τους ενδέχεται να είναι αργή. Για να βελτιωθεί ο ρυθμός σύγκλισης, στη πλειονότητα των περιπτώσεων, οι επαναληπτικές μέθοδοι εφαρμόζονται σε συνδυασμό με ένα προκαταστάτη (preconditioner).

Αλγόριθμος 2 Ο αλγόριθμος Conjugate Gradient (CG) [15]

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $tol \in \mathbb{R}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

$$R_0 = F - AX_0$$

$$P_0 = R_0$$

for $j = 1, \dots, s$ do

while $\|r_i^{(j)}\|/\|f^{(j)}\| \geq tol$ do

$$\alpha = ((r_i^{(j)})^\top r_i^{(j)}) / ((p_i^{(j)})^\top A p_i^{(j)})$$

$$x_{i+1}^{(j)} = x_i^{(j)} + p_i^{(j)} \alpha$$

$$r_{i+1}^{(j)} = r_i^{(j)} - A p_i^{(j)} \alpha$$

$$\beta = ((r_{i+1}^{(j)})^\top r_{i+1}^{(j)}) / ((r_i^{(j)})^\top r_i^{(j)})$$

$$p_{i+1}^{(j)} = r_{i+1}^{(j)} + p_i^{(j)} \beta$$

end while

end for

Ο Αλγόριθμος 2 απεικονίζει τον αλγόριθμο Conjugate Gradient για την επίλυση του γραμμικού συστήματος $AX = F$ με A Συμμετρικό και Θετικά Ορισμένο (ΣΘΟ). Όπως μπορούμε να διαχρίνουμε, η βασική πράξη για κάθε γραμμικό σύστημα είναι ένα μητρώο-διανυσμα (MV) γινόμενο μεταξύ του A και ενός διανύσματος, και ακολουθούν κάποιες πράξεις εσωτερικών γινομένων και ανανέωσης πρώτης τάξης. Εφόσον τα δεξιά μέλη είναι γνωστά εκ των προτέρων, θα μπορούσαμε να υπολογίσουμε ταυτόχρονα το γινόμενο $AP_i = A[p_i^{(1)}, \dots, p_i^{(s)}]$ για όλα τα δεξιά μέλη - το οποίο θα ονομάζουμε υλοποίηση κατά φευδοπλοκάδα. Επιλέξαμε ωστόσο να κρατήσουμε τη παρουσίαση του αλγορίθμου σε απλούστερο επίπεδο. Να σημειωθεί ότι παρότι όλα τα γινόμενα με το μητρώο A θα μπορούσαν να πραγματοποιηθούν μεσώ ενός γινομένου του A με το P_i , κάθε γραμμικό σύστημα λύνεται ανεξάρτητα το ένα από το άλλο.

Τα παραπάνω, ισχύουν και για τον αλγόριθμο BiCGSTAB (Αλγόριθμος 3) ο οποίος εφαρμόζεται για την επίλυση γραμμικών συστημάτων στα οποία το μητρώο συντελεστών A είναι μη-συμμετρικό.

Πριν προχωρήσουμε παρακάτω, αξίζει να σημειώσουμε ότι επαναληπτικά των επαναληπτικών μεθόδων σε σχέση με τις άμεσες μεθόδους και το οποίο σχετίζεται με τη μορφή στην οποία έχουμε διαθέσιμο το μητρώο συντελεστών. Οι επαναληπτικές μέθοδοι χρησιμοποιούν το μητρώο συντελεστών μόνο υπό τη μορφή MV γινομένων και επομένως μπορούν να επιλύσουν γραμμικά συστήματα όπου το μητρώο συντελεστών δεν είναι άμεσα διαθέσιμο, πχ. το A είναι γινόμενο δύο η περισσοτέρων μητρώων. Αντίθετα, οι άμεσες μέθοδοι θα πρέπει να σχηματίσουν το μητρώο συντελεστών ως ένα μοναδικό μητρώο πριν το διασπάσουν.

Αλγόριθμος 3 Ο αλγόριθμος BiCGSTAB [14]

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $tol \in \mathbb{R}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

$$R_0 = F - AX_0$$

Επιλογή ενός τυχαίου μητρώου \hat{R}_0 για το οποίο ισχύει ότι $(\hat{r}_0^{(j)})^\top r_0^{(j)} \neq 0$, $\forall j = 1, \dots, s$

$$\pi_0^{(j)} = \alpha^{(j)} = \omega_0^{(j)} = 1$$

$$v_0^{(j)} = p_0^{(j)} = 0$$

for $j = 1, \dots, s$ do

while $\|r_i^{(j)}\|/\|f^{(j)}\| \geq tol$ do

$$\pi_i^{(j)} = (\hat{r}_0^{(j)})^\top r_{i-1}^{(j)}$$

$$\beta = (\pi_i^{(j)}/\pi_{i-1}^{(j)})(\alpha^{(j)}/\omega_{i-1}^{(j)})$$

$$p_i^{(j)} = r_{i-1}^{(j)} + \beta(p_{i-1}^{(j)} - \omega_{i-1}^{(j)} v_{i-1}^{(j)})$$

$$v_i^{(j)} = Ap_i^{(j)}$$

$$\alpha^{(j)} = \pi_i^{(j)} / (r_0^{(j)})^\top v_i^{(j)}$$

$$s = r_{i-1}^{(j)} - \alpha^{(j)} v_i^{(j)}$$

$$t_i^{(j)} = As_i^{(j)}$$

$$\omega_i^{(j)} = (t_i^{(j)})^\top s_i^{(j)} / (t_i^{(j)})^\top t_i^{(j)}$$

$$x_i^{(j)} = x_i^{(j)} + \alpha^{(j)} p_i^{(j)} + \omega_i^{(j)} s_i^{(j)}$$

$$r_i^{(j)} = s_i^{(j)} - \omega_i^{(j)} t_i^{(j)}$$

end while

end for

1.1.3 Η έννοια της προκατάστασης

Για τη βελτίωση του ρυθμού σύγκλισης των επαναληπτικών μεθόδων, μια ιδιαίτερα δημοφιλής τεχνική είναι η τεχνική της προκατάστασης (preconditioning) του γραμμικού συστήματος $AX = F$. Η προκατάσταση υλοποιείται μέσω ενός αντιστρέψιμου μητρώου M το οποίο ονομάζεται προκαταστάτης και επιλέγεται έτσι ώστε $M^{-1} \approx A^{-1}$, και το γραμμικό σύστημα $My = r$ να μπορεί να επιλυθεί σχετικά γρήγορα. Αν το M^{-1} προσεγγίζει ικανοποιητικά το A^{-1} , τότε η σύγκλιση των μεθόδων υποχώρου Krylov μπορεί να είναι αρκετά γρήγορη (όσον αφορά το ρυθμό σύγκλισης).

Έτσι, αντί για το αρχικό πρόβλημα $AX = F$, επιλύουμε το ισοδύναμο πρόβλημα

$$M^{-1}AX = M^{-1}B. \quad (1.11)$$

Να σημειωθεί ότι ο υπόχωρος Krylov που σχηματίζεται από την επίλυση ενός συστήματος με χρήση προκαταστάτη, $M^{-1}AX = M^{-1}F$, είναι διαφορετικός από αυτό που σχηματίζεται για την επίλυση του συστήματος $AX = F$. Το γινόμενο $M^{-1}A$ δε σχηματίζεται ποτέ κανές στη πλειονότητα των περιπτώσεων το κόστος του πολλαπλασιασμού μεταξύ των δύο αυτών μητρώων θα ξεπεράσει κατά πολύ το κόστος επίλυσης του αρχικού συστήματος $AX = F$. Στη πραγματικότητα, ο υπολογισμός ενός γινομένου της μορφής $y = M^{-1}Av$, με $v \in \mathbb{R}^n$, υλοποιείται μέσω δύο φάσεων: α) του γινομένου $r = Av$, και β) της επίλυσης του γραμμικού συστήματος $My = r$.

Συνοψίζοντας, τα κύρια χαρακτηριστικά που πρέπει να έχει ένας προκαταστάτης M είναι:

- το M^{-1} είναι μια “καλή” προσέγγιση του A^{-1} .

- το κόστος εύρεσης/κατασκευής του M δεν είναι απαγορευτικό.
- το γραμμικό σύστημα $My = r$ λύνεται σχετικά γρήγορα.
- η επίλυση του $My = r$ να είναι κλιμακώσιμη σε αντιστοιχία με τον αριθμό επεξεργαστών p (για παράλληλες αρχιτεκτονικές)

Τα παραπάνω τέσσερα σημεία ίσως αποτελούν ένα “τραχύ” ορισμό των χαρακτηριστικών που πρέπει να διαλέγεται ένα αποδοτικός προκαταστάτης. Για παράδειγμα, πότε ακριβώς το κόστος εύρεσης/κατασκευής του M γίνεται απαγορευτικό; Ή πως μπορούμε να ορίσουμε την σχετική έννοια της φράσης “γρήγορη επίλυση” από τη στιγμή που δε ξέρουμε πόσο δύσκολο είναι το αρχικό πρόβλημα $AX = F$ και πόσο διαρκεί η επίλυση του δίχως προκατάσταση; Θεωρητικά δε μπορούμε, πρακτικά όμως ξέρουμε ή έχουμε μια ένδειξη του πόσο “δύσκολα” είναι προβλήματα από συγκεκριμένες εφαρμογές. Τι κάνει όμως ένα πρόβλημα “δύσκολο” και πιο συγκεκριμένα, πόσο “δύσκολο” είναι ένα πρόβλημα; Στη περίπτωση των επαναληπτικών μεθόδων η απάντηση είναι απλή - το φάσμα, $\rho(A)$, του A . Αν το φάσμα του A είναι στενό, δηλαδή $\lambda_{max} - \lambda_{min} = O(1)$ και $\lambda_{min} \gg 0$, τότε η επαναληπτική μέθοδος θα πρέπει να συγκλίνει αρκετά γρήγορα. Ιδιαίτερα, οι ιδιοτιμές του A θα είναι μαζεμένες γύρω από τη τιμή ένα. Από την άλλη, ιδιοτιμές κοντά στη τιμή μηδέν συνήθως δυσκολεύουν κατά πολύ την επίλυση. Δε θα επεκταθούμε περαιτέρω στη σύγκλιση των επαναληπτικών μεθόδων άλλα θα τονίσουμε ότι, όπως ίσως έγινε προφανές, ο προκαταστάτης M επιλέγεται έτσι ώστε οι ιδιοτιμές $\lambda'_1, \lambda'_2, \dots, \lambda'_n$ του $M^{-1}A$ να είναι όσο το δυνατόν πιο κοντά στη τιμή ένα.

Υπάρχουν διάφοροι τρόποι υλοποίησης επιλυτών ενός γραμμικού συστήματος μέσω προκατάστασης. Διακρίνουμε τρεις διαφορετικές υλοποιήσεις οι οποίες όμως επιδρούν στο φάσμα του A κατά τον ίδιο τρόπο.

Προκατάσταση εξ αριστερών Η επαναληπτική μέθοδος εφαρμόζεται στο σύστημα $M^{-1}AX = M^{-1}B$. Προσοχή όμως ότι τυχόν συμμετρία των M , A δε συνεπάγεται συμμετρία του $M^{-1}A$. Ωστόσο αν το μητρώο M είναι ΣΘΟ, τότε μπορεί να οριστεί ένα νέο εσωτερικό γινόμενο μέσω της σχέσης $[x, y]_M = (x, My)$. Δεδομένου ότι το μητρώο $M^{-1}A$ είναι συμμετρικό ως προς το διοσμένο αυτό εσωτερικό γινόμενο οπότε μπορούν να χρησιμοποιηθούν μέθοδοι επίλυσης οι οποίες έχουν ως προαπαιτούμενο τη συμμετρία του μητρώου συντελεστών A .

Προκατάσταση εκ δεξιών Η επαναληπτική μέθοδος εφαρμόζεται στο σύστημα $AM^{-1}Y = F$ όπου $X = M^{-1}Y$. Η δεξιά προκατάσταση έχει το πλεονέκτημα ότι επηρεάζει μόνο το μητρώο A και όχι τα δεξιά μέλη. Επίσης, η νόρμα του σφάλματος της i -στής επαναληψης, $\|Y - Y_i\|_2$, μπορεί να είναι πολύ μικρότερη από τη πραγματική, $\|X - X_i\|_2 = \|M^{-1}(Y - Y_i)\|_2$, οπότε θα πρέπει να είμαστε σίγουροι ότι ελέγχουμε το σωστό χριτήριο σύγκλισης.

Προκατάσταση δυο πλευρών Έστω ότι $M = M_1M_2$. Τότε η επαναληπτική μέθοδος μπορεί να εφαρμοστεί στο σύστημα $M_1^{-1}AM_2^{-1}Z = M_1^{-1}F$ με $X = M_2^{-1}Z$. Η προκατάσταση δυο πλευρών χρησιμοποιείται όταν ο προκαταστάτης είναι γνωστός στη παραγοντοποιημένη μορφή του και μπορεί να ιδωθεί ως ένας συνδυασμός των προηγούμενων τεχνικών προκατάστασης. Η προκατάσταση δυο πλευρών είναι ιδιαίτερα χρήσιμη όταν δε μπορεί να οριστεί ένα κατάλληλο εσωτερικό γινόμενο όπως στη περίπτωση της αριστερής προκατάστασης [13].

Ο Αλγόριθμος 4 απεικονίζει τη μέθοδο Conjugate Gradient με χρήση προκαταστάτη (PCG) [23, 24]. Ο αλγόριθμος είναι παρόμοιος με αυτόν χωρίς προκατάσταση μόνο που τώρα έχει προστεθεί ένα βήμα προκατάστασης, το οποίο έχει ως στόχο να “βρίσκει” μεγαλύτερο μέρος της λύσης σε κάθε επανάληψη. Ο

προκαταστάτης θα πρέπει να είναι ένα ΣΘΟ μητρώο καθώς σε αντίθετη περίπτωση η συμμετρία/σύγκλιση του αλγορίθμου δεν είναι εγγυημένη.

Αλγόριθμος 4 Ο αλγόριθμος Preconditioned Conjugate Gradient (PCG) [23, 24]

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $M \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $tol \in \mathbb{R}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

$$R_0 = F - AX_0$$

$$Z_0 = M^{-1}R_0$$

$$P_0 = Z_0$$

for $j = 1, \dots, s$ **do**

while $\|r_i^{(j)}\|/\|f^{(j)}\| \geq tol$ **do**

$$\alpha = ((r_i^{(j)})^\top z_i^{(j)}) / ((p_i^{(j)})^\top A p_i^{(j)})$$

$$x_{i+1}^{(j)} = x_i^{(j)} + p_i^{(j)} \alpha$$

$$r_{i+1}^{(j)} = r_i^{(j)} - Ap_i^{(j)} \alpha$$

$$z_{i+1}^{(j)} = M^{-1}r_{i+1}^{(j)}$$

$$\beta = ((z_{i+1}^{(j)})^\top r_{i+1}^{(j)}) / ((z_i^{(j)})^\top r_i^{(j)})$$

$$p_{i+1}^{(j)} = z_{i+1}^{(j)} + p_i^{(j)} \beta$$

end while

end for

Τέλος, ο Αλγόριθμος 5 απεικονίζει τη μέθοδο BiCGSTAB με χρήση προκατάστασης. Αξίζει να αναφερθεί ότι τώρα η προκατάσταση εφαρμόζεται δυο φορές για κάθε επανάληψη (το μητρώο συντελεστών δεν είναι συμμετρικό).

1.1.4 Διάφορες κλάσεις προκαταστατών

1.1.4.1 Προκατάσταση με χρήση υπομητρώων

Η προκατάσταση με χρήση υπομητρώων εξάγει ένα υπομητρώο του αρχικού μητρώου A και το χρησιμοποιεί ως προκαταστάτη M . Συνήθως, η χρήση μεγαλύτερου υπομητρώου βελτιώνει το ρυθμό σύγκλισης της επαναληπτικής διαδικασίας, αυξάνει ωστόσο το κόστος επίλυσης με τον προκαταστάτη, πχ. το ίδιο το μητρώο A είναι ο ιδιαίτερος προκαταστάτης και οποιαδήποτε επαναληπτική μέθοδος υποχώρου Krylov θα συγκλίνει μετά από μία μόλις επανάληψη, το αριθμητικό κόστος όμως δε θα έχει μειωθεί αφού θα πρέπει να επιλύσουμε το αρχικό γραμμικό σύστημα!

Προκαταστάτης Jacobi Η πιο απλή εκδοχή προκατάστασης υπομητρώου έγκειται στην εξαγωγή και χρήση της κύριας διαγώνιου του A ως προκαταστάτη, δηλαδή $M = \text{diag}(A)$ σε συμβολισμό MATLAB. Η προκατάσταση αυτού του τύπου ονομάζεται προκατάσταση Jacobi. Τα πλεονεκτήματα του προκαταστάτη Jacobi είναι ο εύκολος σχηματισμός του M καθώς και το πολύ χαμηλό κόστος εφαρμογής του. Από την άλλη, ο προκαταστάτης Jacobi συνήθως βελτιώνει ελάχιστα το ρυθμό σύγκλισης της επαναληπτικής μεθόδου εκτός και αν το αρχικό μητρώο συντελεστών είναι αυστηρά διαγώνια κυριαρχο (ή εμφανίζει κάποιας μορφής διαγώνιας κυριαρχίας).

Προκαταστάτης Jacobi κατά πλοκάδες Η προκατάσταση Jacobi κατά πλοκάδες είναι μια επέκταση της προκατάστασης Jacobi και συνήθως οδηγεί σε ταχύτερη σύγκλιση. Η προκατάσταση βασίζεται στην εξαγωγή των διαγώνιων πλοκάδων του μητρώου συντελεστών και τη χρήση αυτών ως προκαταστάτη

Αλγόριθμος 5 Ο αλγόριθμος Preconditioned BiCGSTAB [23]

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $M \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $tol \in \mathbb{R}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

$$R_0 = F - AX_0$$

Επιλογή ενός τυχαίου μητρώου \hat{R}_0 για το οποίο ισχύει ότι $(\hat{r}_0^{(j)})^\top r_0^{(j)} \neq 0$, $\forall j = 1, \dots, s$

$$\pi_0^{(j)} = \alpha^{(j)} = \omega_0^{(j)} = 1$$

$$v_0^{(j)} = p_0^{(j)} = 0$$

for $j = 1, \dots, s$ do

while $\|r_i^{(j)}\|/\|f^{(j)}\| \geq tol$ do

$$\pi_i^{(j)} = (\hat{r}_0^{(j)})^\top r_{i-1}^{(j)}$$

$$\beta = (\pi_i^{(j)}/\pi_{i-1}^{(j)})(\alpha^{(j)}/\omega_{i-1}^{(j)})$$

$$p_i^{(j)} = r_{i-1}^{(j)} + \beta(p_{i-1}^{(j)} - \omega_{i-1}^{(j)} v_{i-1}^{(j)})$$

$$y = M^{-1}p_i^{(j)}$$

$$v_i^{(j)} = Ay$$

$$\alpha^{(j)} = \pi_i^{(j)}/(r_0^{(j)})^\top v_i^{(j)}$$

$$s = r_{i-1}^{(j)} - \alpha^{(j)} v_i^{(j)}$$

$$z = M^{-1}s$$

$$t_i^{(j)} = Az$$

$$\omega_i^{(j)} = (M^{-1}t_i^{(j)})^\top M^{-1}s_i^{(j)} / (M^{-1}t_i^{(j)})^\top M^{-1}t_i^{(j)}$$

$$x_i^{(j)} = x_i^{(j)} + \alpha^{(j)} p_i^{(j)} + \omega_i^{(j)} s_i^{(j)}$$

$$r_i^{(j)} = s_i^{(j)} - \omega_i^{(j)} t_i^{(j)}$$

end while

end for

M . Η προκατάσταση Jacobi κατά πλοκάδες προσφέρεται για παράλληλα συστήματα καθώς η εφαρμογή της προκατάστασης (η επίλυση δηλαδή γραμμικών συστημάτων με το M) μπορεί να γίνει ανεξάρτητη για κάθε διαγώνια πλοκάδα. Από την άλλη, το αριθμητικό κόστος τώρα είναι αυξημένο σε σχέση με τον απλό προκαταστάτη Jacobi.

1.1.4.2 Προκατάσταση με χρήση ατελούς παραγοντοποίησης

Σε αντίθεση με τη προκατάσταση με χρήση υπομητρώων, η προκατάσταση με χρήση ατελούς παραγοντοποίησης βασίζεται σε μια ατελή παραγοντοποίηση του μητρώου συντελεστών [22].

Ορισμός 1.2. Εστω ένα μητρώο $A \in \mathbb{R}^{n \times n}$ και τα μητρώα K_1, \dots, K_j , $j \geq 2$ όπου $K_j \in \mathbb{R}^{n \times n} \forall j$. Θα λέμε ότι μια παραγοντοποίηση είναι μη-πλήρης όταν $A = K_1 \cdots K_j + E$ για κάποιο μη-μηδενικό μητρώο $E \in \mathbb{R}^{n \times n}$.

Η χρήση ατελούς παραγοντοποίησης προσπαθεί να αντιμετωπίσει το φαινόμενο του γεμίσματος που προκύπτει από μια πλήρη παραγοντοποίηση του μητρώου συντελεστών (όπως η παραγοντοποίηση $A = LU$). Αυτό επιτυγχάνεται μέσω της απόρριψης κάποιων στοιχείων της πραγματικής παραγοντοποίησης βάσει είτε της θέσης της οποίας βρίσκονται στους επιμέρους παράγοντες είτε του μεγέθους τους.

Μια συνηθισμένη τακτική απόρριψης στοιχείων σε μια μη-πλήρη παραγοντοποίηση είναι η απόρριψη των στοιχείων τα οποία καταλαμβάνουν θέσεις εκεί όπου το μητρώο συντελεστών περιέχει μηδενικές τιμές. Πιο συγκεκριμένα, οι παράγοντες της ατελούς παραγοντοποίησης αποδέχονται όλα τα στοιχεία τα οποία

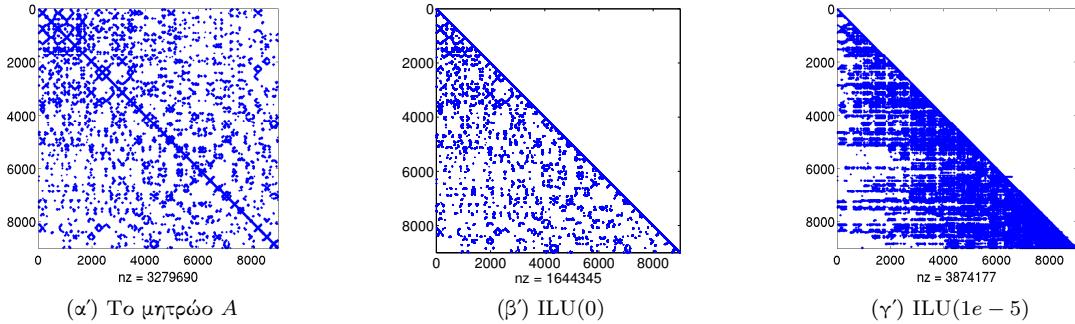
προκύπτουν από τη παραγοντοποίηση και τα οποία καταλαμβάνουν τη θέση (i, j) που ανήκει στο σύνολο \mathcal{S} , με

$$\mathcal{S} = \{(i, j) | a_{ij} \neq 0.\}$$

Η παραπάνω τεχνική ονομάζεται ατελής παραγοντοποίηση (0), πχ. ILU(0) ή IC(0). Μια δεύτερη τεχνική απόρριψης στοιχείων είναι η απόρριψη των στοιχείων τα οποία προκύπτουν και έχουν απόλυτο μέγεθος μικρότερο από κάποιο κατώφλι t , δηλαδή

$$\mathcal{S} = \{(i, j) | \ell_{ij}, u_{ij} > t\}.$$

Το Σχήμα 1.4 απεικονίζει τη μη-μηδενική δομή του μητρώου $nd3k$, τη μη-μηδενική δομή του L το οποίο προκύπτει από τη μη-πλήρη παραγοντοποίηση $ILU(0)$ και τη μη-μηδενική δομή του L το οποίο προκύπτει από τη μη-πλήρη παραγοντοποίηση $ILU(10^{-5})$. Η μη-πλήρης παραγοντοποίηση με χρήση κατωφλίου οδηγεί σε μια



Σχήμα 1.4: Μη-μηδενική δομή των μητρώων A, L της μη-πλήρους παραγοντοποίησης $A = LU$ για το μητρώο $nd3k$.

περισσότερο πιστή αναπαράσταση του πραγματικού μητρώου L της παραγοντοποίησης $A = LU$ καθώς τώρα απορρίπτονται λιγότερα στοιχεία. Το μειονέκτημα της χρήσης κατωφλίου είναι ότι δε μπορεί να προβλεφθεί εξαρχής το πλήρος των μη-μηδενικών στοιχείων στους τελικούς παράγοντες L και U .

1.1.5 Αλγόριθμοι αναδιάταξης αραιών μητρώων για τη κατασκευή ταινιακών προκαταστατών

Σε κάποιες περιπτώσεις μπορούμε να ορίσουμε επακριβώς τη δομή του προκαταστάτη σύμφωνα με τα δικά μας κριτήρια. Αν υποθέσουμε ότι έχουμε στη διάθεση μας έναν εξαιρετικά γρήγορο αλγόριθμο επίλυσης ταινιακών συστημάτων, είναι λογικό να προσπαθήσουμε να το εκμεταλλευτούμε και να εισάγουμε το περιορισμό ότι ο προκαταστάτης πρέπει να είναι ταινιακός. Εξάγοντας απλώς ένα ταινιακό υπομητρώο του A ίσως δε βοηθήσει αρκετά τη σύγκλιση της επαναληπτικής μεθόδου αν το ημι-εύρος δεν είναι υψηλό, αφού η τάξη του υπομητρώου του A έχει από τη μπάντα του προκαταστάτη ίσως παραμένει μεγάλη για μικρά ημι-εύρη (δηλαδή για $m \ll n$). Ιδανικά θα θέλαμε τα μεγαλύτερα κατά απόλυτη τιμή στοιχεία του A να βρίσκονται πολύ κοντά στη κύρια διαγώνιο ώστε να μπορούμε να εξάγουμε έναν αποδοτικό ταινιακό προκαταστάτη όσο το δυνατόν μικρότερου ημι-εύρους.

Συνήθως τα στοιχεία του A δεν ακολουθούν την ιδεατή συμπεριφορά - τα μεγαλύτερα κατά απόλυτη τιμή στοιχεία του A μπορεί να βρίσκονται μακριά από τη κύρια διαγώνιο. Σε αυτή τη περίπτωση, έχει προταθεί

η (μη)-συμμετρική αναδιάταξη των στοιχείων του A ώστε τα μεγαλύτερα κατά απόλυτη τιμή στοιχεία να βρεθούν πιο κοντά στη κύρια διαγώνιο.

Στη περίπτωση συμμετρικής αναδιάταξης, αν $P \in \mathbb{R}^{n \times n}$ είναι το μητρώο αναδιάταξης, τότε το σύστημα $AX = F$ γράφεται ως

$$(P^\top AP)\tilde{X} = P^\top F \quad (1.12)$$

όπου

$$X = P\tilde{X}. \quad (1.13)$$

Η συμμετρική αναδιάταξη διατηρεί τη συμμετρία του μητρώου συντελεστών και επομένως το σύστημα (1.12) μπορεί να επιλυθεί μέσω επιλυτών οι οποίοι εκμεταλλεύονται αυτή την ιδιότητα, από την άλλη όμως δε δύναται να μετακινήσει τα στοιχεία της κύριας διαγώνιου του A μακριά από τη κύρια διαγώνιο (αναδιατάσσει όμως τα διαγώνια στοιχεία κατά μήκος της κύριας διαγώνιου). Το τελευταίο μπορεί να είναι επιθυμητό λόγω ενδεχόμενης ύπαρξης στοιχείων μικρού απολύτου μεγέθους στη κύρια διαγώνιο του A τα οποία ενδέχεται να οδηγήσουν σε αριθμητική αστάθεια κατά τη διάρκεια επίλυσης μέσω άμεσων επιλυτών. Στη περίπτωση των επαναληπτικών επιλυτών, στοιχεία μικρού απολύτου μεγέθους στη κύρια διαγώνιο οδηγούν σε μη-βέλτιστο προκαστάτη M . Αν χρησιμοποιηθεί μη-συμμετρική αναδιάταξη, τότε το σύστημα $AX = F$ γράφεται ως

$$(QAP)\tilde{X} = QF \quad (1.14)$$

με $Q \in \mathbb{R}^{n \times n}$ το μητρώο αναδιάταξης το οποίο δρα στις γραμμές του A και P το μητρώο αναδιάταξης το οποίο δρα στις στήλες. Είναι προφανές ότι σε αυτή τη περίπτωση η συμμετρία του προβλήματος χάνεται.

Στη συνέχεια θα αναφερθούμε συνοπτικά σε μερικούς γνωστούς αλγορίθμους συμμετρικής αναδιάταξης αφαιών μητρώων. Στόχος είναι να βρεθεί ένα διάνυσμα $\pi \in \mathbb{R}^n$ (ισοδύναμο με το μητρώο αναδιάταξης P) τέτοιο ώστε το ημι-εύρος του αναδιαταγμένου μητρώου A , $A_{\pi,\pi}$, να είναι (ελπίζουμε αρκετά) μικρότερο από αυτό του αρχικού A . Στα παρακάτω, κάνουμε χρήση της θεωρίας γράφων σύμφωνα με την οποία κάθε μητρώο $A \in \mathbb{R}^{n \times n}$ μπορεί να ιδωθεί ως ένας γράφημα G με n κόμβους.

Ορισμός 1.3. Έστω ένα μητρώο $A \in \mathbb{R}^{n \times n}$. Ορίζουμε το γράφημα $G_A := (V, E)$ ως ένα γράφο με $|V| = n$ κόμβους και ακμές $E = \{(i, j) | a_{ij} \neq 0\}$. Κάθε ακμή (i, j) φέρει βάρος $w_{ij} = a_{ij}$. Αν το μητρώο A είναι μη-συμμετρικό, το γράφημα G_A είναι κατευθυνόμενο ενώ αν το A είναι συμμετρικό, το γράφημα G_A είναι μη-κατευθυνόμενο.

Ο παραπάνω ορισμός ισχύει και αντίστροφα, δηλαδή κάθε γράφημα $G := (V, E)$ με $|V| = n$ μπορεί να εκφραστεί μεσάνες μητρώου γειτνίασης $A \in \mathbb{R}^{n \times n}$.

Αντίστροφος Cuthill-McKee Ένας αλγόριθμος αναδιάταξης αφαιών, συμμετρικών μητρώων είναι ο αλγόριθμος Cuthill-McKee [7] ο οποίος αποτελεί παραλλαγή ενός βασικού αλγορίθμου διάσχισης γραφημάτων, του αναζητηση πρωτα κατα πλατος. Ο αλγόριθμος Cuthill-McKee επίσης διασχίζει τους κόμβους ενός γραφήματος κατά επίπεδα, ωστόσο η προτεραιότητα σε κάθε επίπεδο δίνεται στο κόμβο με το μικρότερο βαθμό γειτνίασης.

Μια βελτίωση του αλγορίθμου Cuthill-McKee είναι ο αντίστροφος (reverse) Cuthill-McKee [8] (RCM) ο οποίος διαφέρει από τον (ευθύν) Cuthill-McKee μόνο ως προς το τελευταίο στάδιο όπου αντιστρέφεται η σειρά με την οποία έχουν διασχιστεί οι κόμβοι του γραφήματος.

Αναδιάταξη Fiedler Ένας επίσης γνωστός αλγόριθμος αναδιάταξης αφαιών, συμμετρικών μητρώων βασίζεται στο δεύτερο μικρότερο ιδιοζεύγος του μητρώου Laplace του γραφου G_A [38]. Πιο

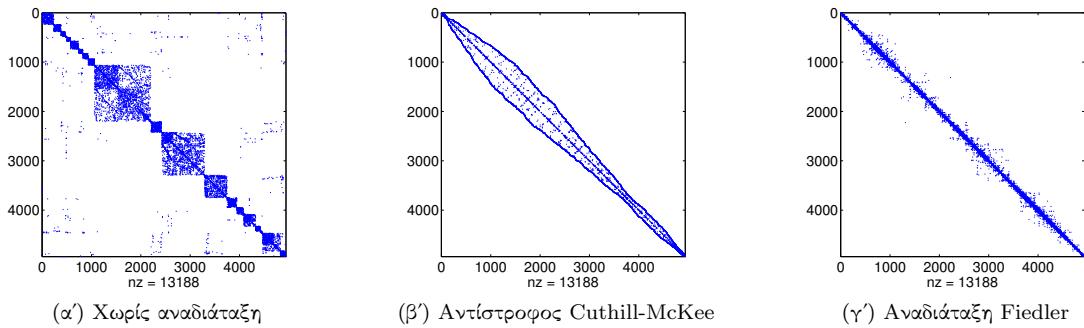
συγκεκριμένα, το μητρώο Laplace $L^A \in \mathbb{R}^{n \times n}$ ενός συμμετρικού μητρώου A , ορίζεται ως

$$\ell_{ij}^A = \begin{cases} \sum_{j=1}^n |d_i| & \text{αν } i = j \\ -1 & \text{αν } i \neq j \text{ και } a_{ij} \neq 0 \end{cases}$$

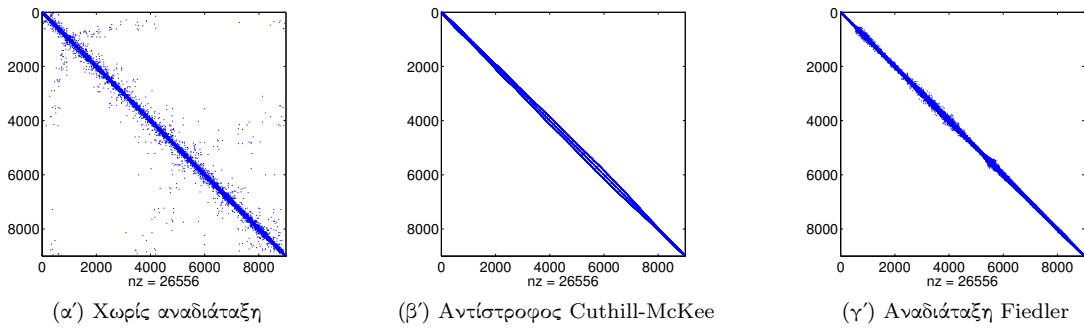
όπου

$$d_i = \{k | a_{ik} \neq 0, k = 1, \dots, n \ (k \neq i)\}. \quad (1.15)$$

Η δεύτερη μικρότερη ιδιοτιμή του L^A ονομάζεται αλγεβρική συνδεσμούτητα του γραφήματος και το αντίστοιχο ιδιοδιάνυσμα είναι γνωστό ως διάνυσμα Fiedler [9]. Το διάνυσμα Fiedler περιέχει την επιθυμητή αναδιάταξη του αραιού μητρώου η οποία προκύπτει ύστερα από την επίλυση ενός προβλήματος ελαχιστοποίησης. Για την εύρεση του διανύσματος Fiedler μπορούν να χρησιμοποιηθούν διάφορες αριθμητικές μέθοδοι όπως η μέθοδος Lanczos [37].

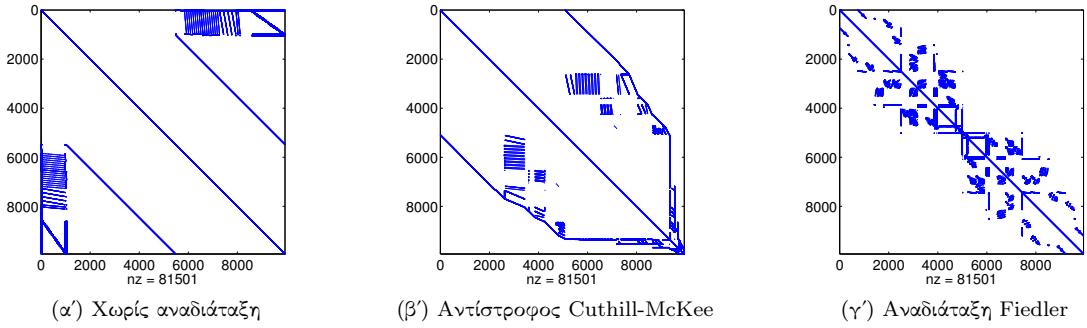


Σχήμα 1.5: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και Fiedler για το μητρώο USpowerGrid



Σχήμα 1.6: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και Fiedler για το μητρώο 3elt_dual

Τα Σχήματα 1.5 - 1.7 απεικονίζουν τη μη-μηδενική δομή τριών διαφορετικών μητρώων και των αναδιατάξεων που προκύπτουν από τους αλγορίθμους αναδιάταξης (i) RCM και (ii) Fiedler για ορισμένα μητρώα από τη συλλογή [34]. Μπορούμε εύκολα να παρατηρήσουμε τη μείωση του (ημι-)εύρους ύστερα από την αναδιάταξη. Μια ακόμα παρατήρηση είναι ότι ενώ ο RCM φαίνεται να μειώνει καλύτερα το (ημι-)εύρος, η αναδιάταξη η οποία είναι βασισμένη στο διάνυσμα Fiedler μπορεί να “συμπυκνώσει” καλύτερα τα στοιχεία γύρω από τη κύρια διαγώνιο.



Σχήμα 1.7: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και Fiedler για το μητρώο c-40

Σταθμισμένη φασματική αναδιάταξη (WSR) Οι προηγούμενες μέθοδοι αναδιάταξης έχουν τη δυνατότητα να μειώσουν το εύρος ενός μητρώου, ωστόσο δε κατηγοριοποιούν τα στοιχεία ανάλογα με το μέγεθός τους. Ένα στοιχείο εκτός της κυρίας διαγώνιου μπορεί να αναδιαταχθεί στη κύρια διαγώνιο ανεξαρτήτως του πόσο μικρή ή μεγάλη είναι η απόλυτη τιμή του. Το πρώτο μειονέκτημα αυτής της τακτικής είναι η μεγαλύτερη πιθανότητα εύρεσης ενός “κακού” οδηγού στοιχείου (pivot element) κατά τη διάρκεια της παραγοντοποίησης LU. Το δεύτερο μειονέκτημα έγκειται στην υπόθεση ότι ακόμα και αν το εύρος του μητρώου μειωθεί αρκετά μέσω των παραπάνω αλγορίθμων αναδιάταξης, μπορεί ακόμα να παραμείνει “μεγάλο” και η επίλυση του αναδιαταγμένου συστήματος με κάποιον ταινιακό επιλυτή να μην είναι αποδοτική. Για αυτό το λόγο, η WSO περισσότερο στοχεύει στην αναδιάταξη των στοιχείων μεγάλου μεγέθους (κατά απόλυτη τιμή) σε μικρή απόσταση από τη κύρια διαγώνιο του αναδιαταγμένου μητρώου, παρά στην ελαχιστοποίηση του εύρους του.

Δοθέντος ενός μητρώου $A \in \mathbb{R}^{n \times n}$ και μιας τιμής γ , ορίζουμε το πρόβλημα της σταθμισμένης μείωσης του εύρους του A [4], ως το πρόβλημα της εύρεσης ενός μητρώου $M \in \mathbb{R}^{n \times n}$ ελάχιστου εύρους, τέτοιο ώστε

$$\frac{\sum_{i,j=1}^{i,j=n} |a_{ij} - m_{ij}|}{\sum_{i,j=1}^{i,j=n} |a_{ij}|} \leq \gamma \quad (1.16)$$

με $m_{ij} = a_{ij}$ αν $|i - j| \leq m$ και $m_{ij} = 0$ στην αντίθετη περίπτωση.

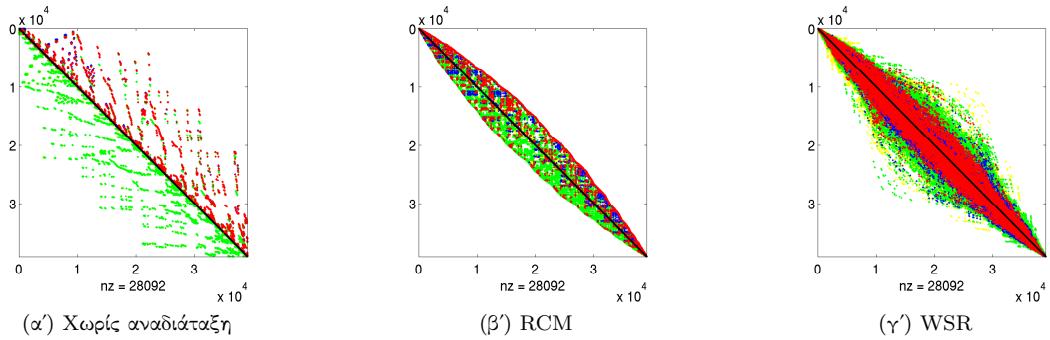
Η σταθμισμένη αναδιάταξη βασίζεται σε μεγάλο βαθμό στη (μη σταθμισμένη) φασματική αναδιάταξη. Η φασματική αναδιάταξη είναι μια αλγεβρική τεχνική, η οποία εφαρμόζεται ευρέως για την εύρεση μιας προσεγγιστικής λύσης διαφόρων προβλημάτων βελτιστοποίησης γραφημάτων [11]. Η κύρια ιδέα της φασματικής αναδιάταξης βρίσκεται στον υπολογισμό ενός διανύσματος $x \in \mathbb{R}^n$ το οποίο ελαχιστοποιεί τη ποσότητα

$$\sigma_A(x) = \sum_{i,j: a_{ij} \neq 0} (x_i - x_j)^2 \quad (1.17)$$

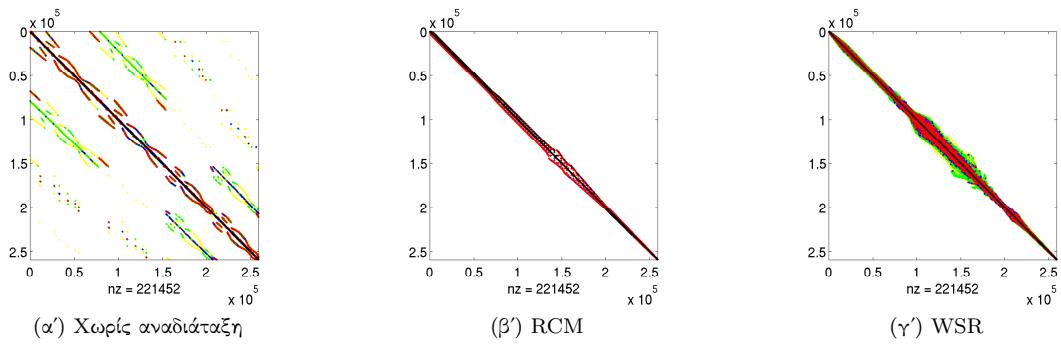
υπό τη προϋπόθεση ότι $\|x\| = 1$, $x^T e = 0$ και το μητρώο $A \in \mathbb{R}$ είναι συμμετρικό.

Η λύση του παραπάνω προβλήματος ελαχιστοποίησης είναι το διάνυσμα Fiedler του μητρώου Laplace που προκύπτει από το A . Ωστόσο, ενώ όπως είδαμε και προηγουμένως, η αναδιάταξη η οποία βασίζεται στο διάνυσμα Fiedler, δύναται να μειώσει αρκετά το εύρος, δεν υπάρχει κάποια μέριμνα για τον ιδιαίτερο χειρισμό στοιχείων τα οποία είναι μεγάλου μεγέθους. Μια πρώτη προσπάθεια γενίκευσης της αναδιάταξης Fiedler ώστε να λαμβάνεται υπόψιν το μέγεθος των στοιχείων του A μπορεί να βρεθεί

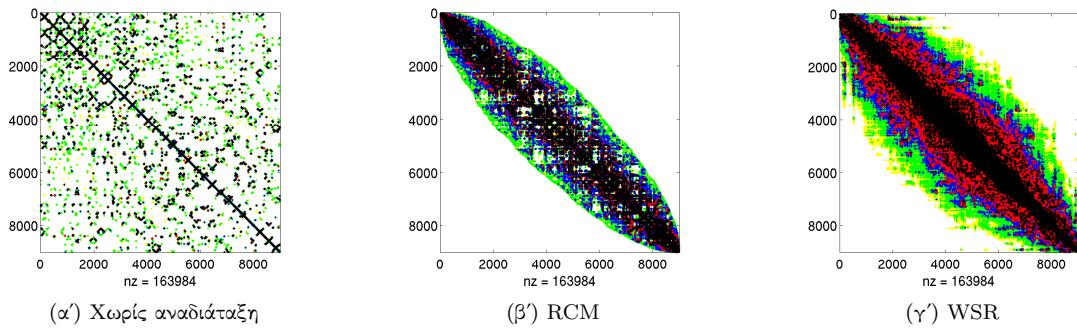
στο [16]. Μια καλύτερη προσέγγιση, που οδηγεί και στον στη μέθοδο αναδιάταξης WSR, περιγράφεται στο [4].



Σχήμα 1.8: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο cage11

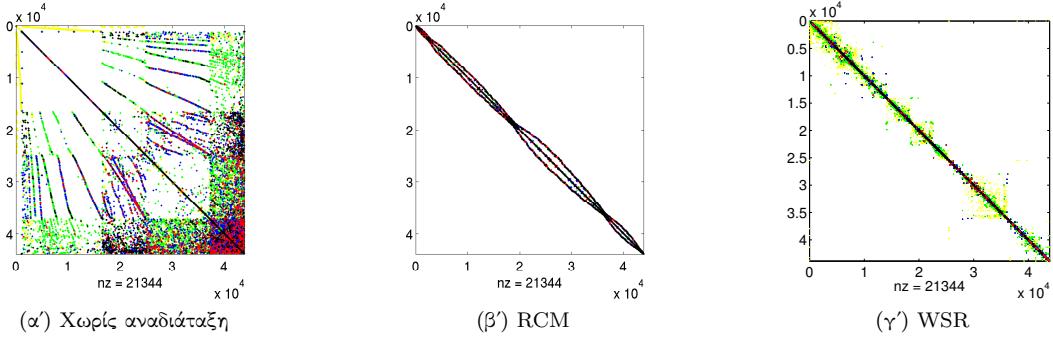


Σχήμα 1.9: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο torso3



Σχήμα 1.10: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο nd3k

Τα Σχήματα 1.8 - 1.11 απεικονίζουν τη μη-μηδενική δομή τεσσάρων διαφορετικών μητρώων και των αναδιατάξεων που προκύπτουν από τους αλγορίθμους αναδιάταξης (i) RCM και (ii) WSR. Σε αντίθεση με τα προηγούμενα Σχήματα 1.5 - 1.7, τώρα πέραν της μη-μηδενικής δομής που προκύπτει ύστερα από την αναδιάταξη, ενδιαφερόμαστε για το κατά πόσο τα στοιχεία μεγάλου απολύτου μεγέθους αναδιατάσσονται κοντά στη κύρια διαγώνιο. Στα Σχήματα 1.8 - 1.11, τα στοιχεία μεγαλύτερου



Σχήμα 1.11: Η μη-μηδενική δομή πριν και μετά τις αναδιατάξεις RCM και WSR για το μητρώο opf10000

μεγέθους σημειώνονται με μαύρο χρώμα και όσο πιο ανοικτός είναι ο χρωματισμός, τόσο μικρότερο είναι το μέγεθος του στοιχείου. Μπορούμε να διαπιστώσουμε ότι για όλα τα μητρώα τα οποία δοκιμάσαμε:

- Το (ημι-)εύρος συνήθως μειώνεται περισσότερο αν χρησιμοποιηθεί η αναδιάταξη RCM, και
- Η αναδιάταξη WSR έχει πολύ καλύτερα αποτελέσματα στο να φέρει τα στοιχεία μεγάλου απολύτου μεγέθους χοντά στη κύρια διαγώνιο, δίνοντας έτσι ένα αποδοτικότερο ταινιακό προκαταστάτη.

1.2 Η παράλληλη επίλυση αραιών γραμμικών συστημάτων και η οικογένεια αλγορίθμων Spike

Λόγω των μητρώων συντελεστών μεγάλου μεγέθους τα οποία προκύπτουν από τις σύγχρονες εφαρμογές (μητρώα μεγέθους μερικών δεκαδών εκατομμυρίων δε ύφεωρούνται πλέον υπερβολικά μεγάλου μεγέθους) αλλά και της ανάγκης για την τάχιστη επίλυση των γραμμικών συστημάτων, οι αριθμητικές μέθοδοι επίλυσης (άμεσες και επαναληπτικές) υλοποιούνται σε περιβάλλοντα τα οποία χρησιμοποιούν πλήθος επεξεργαστών/πυρήνων οπότε και η επίλυση λαμβάνει χώρα παράλληλα. Ο σχεδιασμός και η υλοποίηση μεθόδων που λαμβάνουν υπόψη τους τη παραλληλία του υπολογιστικού συστήματος είναι ένα σύνθετο πρόβλημα το οποίο εισάγει νέες έννοιες όπως της κλιμάκωσης, της ισορροπίας υπολογιστικού φόρτου καθώς και την ανάγκη επικοινωνίας μεταξύ των διαφόρων υπολογιστικών συστημάτων κάτι το οποίο αυξάνει τη πολυπλοκότητα υλοποίησης και απαιτεί να ληφθούν υπόψη διάφορες άλλες παράμετροι.

Στη παρούσα εργασία επιλέγουμε να χτίσουμε νέες αλγορίθμικές μεθόδους βασιζόμενοι σε μια οικογένεια αλγορίθμων η οποία έχει επιδείξει μια ιδιαίτερα υψηλή επίδοση σε παράλληλες αρχιτεκτονικές. Η οικογένεια αυτή είναι η οικογένεια αλγορίθμων Spike.

Η αρετηρία της οικογένειας αυτής υπήρξε στα τέλη της δεκαετίας του 1970 όπου οι Sameh και Kuck [10] σχεδίασαν έναν άμεσο επιλυτή ο οποίος απομακρυνόταν από τη διάσπαση $A = LU$. Η μέθοδος Spike διασπά το μητρώο A ως

$$A = DS \quad (1.18)$$

με $D \in \mathbb{R}^{n \times n}$ μητρώο διαγώνιο κατά πλοκάδες και $S \in \mathbb{R}^{n \times n}$ μητρώο τριδιαγώνιο κατά πλοκάδες. Σε αντίθεση με την υλοποίηση της παραγοντοποίησης LU σε παράλληλες αρχιτεκτονικές, η μέθοδος Spike ελαχιστοποιεί τις μεταφορές στοιχείων από και προς τη μνήμη [46] και εισάγει πράξεις οι οποίες μπορούν

να εκτελεστούν σε τετριμένη παραλληλία (περισσότερα στο Κεφάλαιο 2). Η βασική μέθοδος Spike έχει επεκταθεί με τη πάροδο του χρόνου και έχει οδηγήσει σε πλήθος διαφορετικών υπο-αλγορίθμων για το χειρισμό γραμμικών συστημάτων ειδικού τύπου όπως για τη περίπτωση διαγώνια κυρίαρχων ταινιακών μητρώων. Παραλλαγή της βασικής μεθόδου Spike μπορεί επίσης να χρησιμοποιηθεί ως προκαταστάτης για την επίλυση αραιών γραμμικών συστημάτων όπου τα γραμμικά συστήματα εφαρμογής του προκαταστάτη επιλύονται μέσω της μεθόδου Spike. Αυτές οι διαφοροποιήσεις αλλά και μια ακόμα που θα δούμε στο Κεφάλαιο 2 έχουν οδηγήσει στο πολυαλγόριθμο ή αλλιώς οικογένεια αλγορίθμων Spike.

Κεφάλαιο 2

Η οικογένεια αλγορίθμων Spike

2.1 Επίλυση ταινιακών γραμμικών συστημάτων

2.1.1 Ο αλγόριθμος Spike

Έστω το γραμμικό σύστημα

$$AX = F \quad (2.1)$$

με αντιστρέψιμο και ταινιακό $A \in \mathbb{R}^{n \times n}$, ημι-εύρους m , και $F \in \mathbb{R}^{n \times s}$ ($s \geq 1$). Για την επίλυση του γραμμικού συστήματος, υποθέτουμε ότι έχουμε στη διάθεση μας p επεξεργαστές (processors). Η μέθοδος Spike [1, 10] βασίζεται στη παραγοντοποίηση $A = DS$ όπου $D \in \mathbb{R}^{n \times n}$ είναι ένα μητρώο διαγώνιο κατά πλοκάδες και $S \in \mathbb{R}^{n \times n}$ είναι ένα μητρώο ειδικής δομής όπως φαίνεται και στο σχήμα (2.1) για ένα στιγμιότυπο τριών διαχωρισμάτων. Το μητρώο S είναι γνωστό και ως spike-μητρώο. Στο σημείο αυτό πρέπει να σημειώσουμε ότι παραπάνω του ενός επεξεργαστές μπορούν να ανατεθούν σε κάθε διαφορετικό διαχώρισμα, ανάλογα με το μέγεθος αυτού. Στη συνέχεια της εργασίας θα θεωρήσουμε ότι όλα τα διαχωρίσματα είναι ισομεγέθη και κάθε ένα από αυτά αντιστοιχεί σε έναν και μόνο επεξεργαστή (ακριβώς ένας επεξεργαστής ανατίθεται σε κάθε διαχώρισμα). Έτσι, το πλήθος των επεξεργαστών αυτόματα υποδεικνύει και το πλήθος των διαχωρισμάτων. Φυσικά θα υποθέτουμε πάντοτε ότι ο αριθμός των διαθέσιμων επεξεργαστών είναι μεγαλύτερος η ίσος του πλήθους διαχωρισμάτων.

$$\left(\begin{array}{c|cc} A_1 & B_1 \\ \hline C_1 & A_2 \\ \hline C_2 & B_2 \\ \hline C_3 & A_3 \end{array} \right) = \left(\begin{array}{c|c|c} A_1 & & \\ \hline & A_2 & \\ \hline & & A_3 \end{array} \right) \left(\begin{array}{c|cc} I_1 & V_1 & \\ \hline W_1 & & V_2 \\ \hline W_2 & I_2 & \\ \hline W_3 & I_3 & \\ \hline & & I_{3..} \end{array} \right)$$

Σχήμα 2.1: Η διάσπαση $A = DS$, παράδειγμα για $p = 3$.

Αρχικά, στον επεξεργαστή j , $1 \leq j \leq p$, ανατίθεται ένα σύνολο διαδοχικών γραμμών των A_j , F_j το οποίο θα θεωρούμε ότι είναι ίσο με n/p , εκτός αν αναφέρεται διαφορετικά. Στη συνέχεια, ο επεξεργαστής j ορίζει τα μητρώα $A_j \in \mathbb{R}^{n/p \times n/p}$, $F_j \in \mathbb{R}^{n/p \times s}$, B_j , $C_j \in \mathbb{R}^{m \times m}$ όπου A_j είναι η j -στή διαγώνια πλοκάδα του

$A, B_j (C_j)$ η κάτω (άνω) τριγωνική, μη-μηδενική πλοκάδα δεξιά (αριστερά) του A_j και F_j είναι η πλοκάδα των n/p γραμμών οι οποίες αντιστοιχούν στο j -στό επεξεργαστή¹. Όπως είναι προφανές, ο επεξεργαστής $j = 1$ ($j = p$), πέραν του A_j ορίζει μόνο το μητρώο $B_j (C_j)$.

Αλγόριθμος 6 Ο αλγόριθμος Spike.

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $p \in \mathbb{Z}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

{Κάθε επεξεργαστής j :

```

if  $2 \leq j \leq p - 1$  then
     $A_j[V_j, W_j, G_j] = [\hat{B}_j, \hat{C}_j, F_j]$ 
else if  $j = 1$  then
     $A_1[V_1, G_1] = [\hat{B}_1, F_1]$ 
else
     $A_p[W_p, G_p] = [\hat{C}_p, F_p]$ 
end if

```

{Τώρα όλοι οι επεξεργαστές}

Επίλυση του spike-συστήματος $S_r X_r = G_r$

Τπολογισμός των τελικών λύσεων

$$\begin{cases} X'_1 = G'_1 - V'_1 X_2^{(q)} & j = 1 \\ X'_j = G'_j - V'_j X_{j+1}^{(q)} - W'_j X_{j-1}^{(1)} & 2 \leq j \leq p - 1 \\ X'_p = G'_p - W'_p X_{p-1}^{(1)} & j = p \end{cases}$$

Ο αλγόριθμος Spike (Αλγόριθμος 6) διαιρείται σε δύο βασικά στάδια:

(i) Στάδιο πρόεπεξεργασίας

(ii) Στάδιο μέταεπεξεργασίας

Στάδιο πρόεπεξεργασίας Σε αυτό το στάδιο, κάθε διαγώνια πλοκάδα A_j παραγοντοποιείται σε μια απλούστερη μορφή, πχ. $A_j = L_j U_j$, και στη συνέχεια επιλύονται τα γραμμικά συστήματα

$$A_j V_j = \hat{B}_j \quad (2.2)$$

και

$$A_j W_j = \hat{C}_j \quad (2.3)$$

όπου το μητρώο $\hat{B}_j (\hat{C}_j)$ αποτελεί την επέκταση της πλοκάδας $B_j (C_j)$ προς τα πάνω (κάτω) με ένα μηδενικό μητρώο μεγέθους $(n_j - m) \times m$. Αν το μητρώο A έχει κάποια ιδιότητα, πχ. είναι συμμετρικό και θετικά ορισμένο², τότε μπορεί να χρησιμοποιηθεί μια περισσότερο αποδοτική παραγοντοποίηση όπως η παραγοντοποίηση Cholesky, $A_j = \bar{L}_j \bar{L}_j^\top$ όπου $\bar{L}_j \in \mathbb{R}^{n \times n}$ είναι κάτω τριγωνικό μητρώο με θετικά διαγώνια στοιχεία (όχι απαραίτητα ίσα με τη μονάδα). Θα θεωρούμε πάντα, εκτός και αν αναφέρεται αντιθέτως, ότι η παραγοντοποίηση δε χρειάστηκε οδήγηση.

¹Σε συμβολισμό MATLAB: $A_j = A(1 + (j - 1)n/p : jn/p, 1 + (j - 1)n/p : jn/p)$
 $B_j = A(1 + (j - 1)n/p : jn/p, jn/p + 1 : jn/p + m)$, για $j \leq p$

$C_j = A(1 + (j - 1)n/p : jn/p, (j - 1)n/p - m : (j - 1)n/p - 1)$, για $j \geq 1$

²Ένα μητρώο $Q \in \mathbb{R}^{n \times n}$ είναι θετικά ορισμένο όταν όλες του οι ιδιοτιμές είναι θετικές.

Το στάδιο προεπεξεργασίας εκτελείται τοπικά σε κάθε επεξεργαστή και κατά την ίδια χρονική στιγμή, εισάγοντας έτσι μια μορφή τετριμένης παραλληλίας κατά μήκος των p επεξεργαστών. Αυτή η μορφή παραλληλίας διαφέρει από τη παράλληλη υλοποίηση της παραγοντοποίησης $A = LU$ υπό την έννοια ότι δεν αποτελεί μια παράλληλη εκδοχή ενός κατά βάση σειριακού αλγορίθμου (όπως είναι $A = LU$) αλλά “δημιουργείται” χάρη στη διαφορετική προσέγγιση της παραγοντοποίησης του A , ως $A = DS$.

Πριν προχωρήσουμε στο στάδιο μετεπεξεργασίας αξίζει να επαναλάβουμε ότι η παραγοντοποίηση $A = DS$ εισάγει πρόσθιτη πολυπλοκότητα υλοποίησης υπό την έννοια ότι τώρα θα πρέπει να ληφθεί υπόψη η ισορροπία του υπολογιστικού φόρτου κατά μήκος των επεξεργαστών αφού ο συνολικός χρόνος ολοκλήρωσης του σταδίου προεπεξεργασίας είναι ίσος με το μεγαλύτερο χρόνο επίλυσης του $A_j[V_j, W_j] = [\hat{B}_j, \hat{C}_j]$ για κάποιο $j \neq 1, p^3$. Με βάση τα παραπάνω, ο διαχωρισμός του A κατά γραμμές ίσως να μην είναι ο ενδεδειγμένος ή τουλάχιστον όχι ο βέλτιστος καθώς υπάρχει περίπτωση ο αριθμός των μη-μηδενικών στοιχείων του A_i να είναι πολύ μεγαλύτερος αυτού του A_j με συνέπεια ο χρόνος παραγοντοποίησης και επίλυσης με το A_i να είναι υψηλότερος. Τόπο αυτές τις συνθήκες, ο διαχωρισμός του A με βάση τον ίσο ή περίπου ίσο αριθμό μη-μηδενικών στοιχείων του A σε κάθε διαχώρισμα είναι διαισθητικά περισσότερο φυσικός. Αυτή η υποσημείωση βρίσκει περισσότερο εφαρμογή για μητρώα A τα οποία είναι αραιά εσωτερικά της μπάντας. Στη παρούσα εργασία δε θα συναντήσουμε τέτοια φαινόμενα (εννοούμε φαινόμενα ετεροβαρής κατανομής φόρτου).

Στάδιο μέτεπεξεργασίας Σε αυτό το στάδιο, το μητρώο δεξιών μελών F εισάγεται στην εξίσωση και λαμβάνει χώρα η δεύτερη φάση επίλυσης του γραμμικού συστήματος $AX = F$. Πιο συγκεκριμένα, αρχικά επιλύονται τοπικά τα γραμμικά συστήματα

$$A_j G_j = F_j, \quad j = 1, \dots, p \quad (2.4)$$

και στη συνέχεια οι λύσεις της παραπάνω εξίσωσης, G_j , σχηματίζουν το μητρώο

$$G = \begin{bmatrix} G_1 \\ \vdots \\ G_p \end{bmatrix}$$

το οποίο χρησιμοποιείται ως μητρώο δεξιών μελών στο γραμμικό σύστημα

$$SX = G \quad (2.5)$$

από όπου προκύπτει και η λύση της αρχικής εξίσωσης μητρώου $AX = F$. Το γραμμικό σύστημα της εξίσωσης (2.5) ονομάζεται και spike-σύστημα. Μέχρι τη φάση επίλυσης του spike-συστήματος όλοι οι προηγούμενοι υπολογισμοί έχουν περατωθεί χρησιμοποιώντας προφανή παραλληλία. Η επίλυση του συστήματος (2.5) απαιτεί επικοινωνία μεταξύ των επεξεργαστών και μπορεί να πραγματοποιηθεί με χρήση είτε άμεσων είτε επαναληπτικών μεθόδων. Οι επαναληπτικές μέθοδοι έχουν συνήθως καλύτερη κλιμάκωση (scalability) της απόδοσης τους συναρτήσει της αύξησης του πλήθους των διαθέσιμων επεξεργαστών p ωστόσο η σύγκλιση τους ενδέχεται να είναι αργή αν δε χρησιμοποιηθεί κάποια κατάλληλη τεχνική προκατάστασης. Η βασική εκδοχή του αλγορίθμου Spike επιλύει το spike-σύστημα με χρήση άμεσων μεθόδων.

³κατά το γνωστό ρητό “μια αλυσίδα είναι τόσο δυνατή όσο ο πιο αδύνατος κρίκος της”

Αυτό που κάνει τη μέθοδο Spike ιδιαίτερα ελκυστική είναι η παφατήρηση ότι η επίλυση του spike-
συστήματος, $SX = G$, μπορεί να απλοποιηθεί περαιτέρω καθώς είναι εύκολο να δειχθεί ότι το σύστημα
αυτό μπορεί να μειωθεί σε ένα πολύ μικρότερο, μεγέθους $2m(p - 1)$.

Πιο συγκεκριμένα, οι spike-πλοκάδες V_j , W_j μπορούν να διαχωριστούν ως

$$V_j = \begin{bmatrix} V_j^{(q)} \\ V_j^{(')} \\ V_j^{(1)} \end{bmatrix}, \quad W_j = \begin{bmatrix} W_j^{(q)} \\ W_j^{(')} \\ W_j^{(1)} \end{bmatrix}$$

όπου $V_j^{(1)}$, $V_j^{(')}$, $V_j^{(q)}$ και $W^{(1)}$, $W_j^{(')}$, $W_j^{(q)}$ αναπαριστούν τις πρώτες m , μεσαίες $n_j - 2m$ και
τελευταίες m γραμμές των V_j , W_j αντίστοιχα. Θεωρώντας τον ίδιο διαχωρισμό και για τα X και G ,
προκύπτει ότι από το spike-σύστημα (2.5) μπορεί να εξαχθεί ένα νέο σύστημα, έστω $S_r X_r = G_r$, το
οποίο είναι τριδιαγώνιο κατά πλοκάδες και αποτελείται από τις m πρώτες γραμμές, άνω και κάτω του
κάθε διαχωρισμού όπως φαίνεται και στο σχήμα 2.6 για ένα στιγμιότυπο με $p = 3$ επεξεργαστές.

$$\begin{bmatrix} I & V_1^{(1)} & 0 & 0 \\ W_2^{(q)} & I & 0 & V_2^{(q)} \\ W_2^{(1)} & 0 & I & V_2^{(1)} \\ 0 & 0 & W_3^{(q)} & I \end{bmatrix} \begin{bmatrix} X_1^{(1)} \\ X_2^{(q)} \\ X_2^{(1)} \\ X_3^{(q)} \end{bmatrix} = \begin{bmatrix} G_1^{(1)} \\ G_2^{(q)} \\ G_2^{(1)} \\ G_3^{(q)} \end{bmatrix}. \quad (2.6)$$

Τελικώς, η λύση X υπολογίζεται μέσω του τύπου:

$$\begin{cases} X'_1 = G'_1 - V'_1 X_2^{(q)} & j = 1 \\ X'_j = G'_j - V'_j X_{j+1}^{(q)} - W'_j X_{j-1}^{(1)} & 2 \leq j \leq p-1 \\ X'_p = G'_p - W'_p X_{p-1}^{(1)} & j = p \end{cases}$$

2.1.2 Ο αλγόριθμος TR-Spike

Η μέθοδος Spike, η οποία περιγράφηκε στη προηγούμενη ενότητα, χειρίζεται ταυνιακά μητρώα γενικού τύπου,
τα οποία δεν χαρακτηρίζονται από κάποια ιδιότητα (εκτός φυσικά από το ότι είναι αντιστρέψιμα). Στη
περίπτωση που το μητρώο συντελεστών A είναι αυστηρά διαγώνια κυρίαρχο, μπορεί να εφαρμοστεί μια
αλγορίθμική εξέλιξη της βασικής μεθόδου Spike. Αυτή η εξέλιξη είναι ο αλγόριθμος TR-Spike [1].

Ορισμός 2.1. Εάν μητρώο $A \in \mathbb{R}^{n \times n}$ είναι αυστηρώς διαγώνια κυρίαρχο αν για $i = 1, \dots, n$ ισχύει ότι

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|. \quad (2.7)$$

Ορισμός 2.2. Εστω $A \in \mathbb{R}^{n \times n}$ ένα αυστηρώς διαγώνια κυρίαρχο μητρώο. Τότε ο αντίστροφος βαθμός
κυριαρχίας ορίζεται ως

$$\epsilon = \max_i \frac{\sum_{j=1, j \neq i}^n a_{ij}}{|a_{ii}|}. \quad (2.8)$$

Η διαφορά του αλγορίθμου TR-Spike από τη βασική μέθοδο Spike έγκειται στη περικοπή (truncation) του spike-μητρώου, το οποίο πλέον είναι διαγώνιο κατά πλοκάδες σε αντιδιαστολή με το τριδιαγώνιο κατά πλοκάδες spike-μητρώο του αλγορίθμου SPIKE. Αυτή η διαφοροποίηση εισάγει δύο σημαντικά οφέλη (βλέπε [1]) τα οποία είναι α) καλύτερη απόδοση στις παράλληλες αρχιτεκτονικές καθώς και β) αποφυγή του κόστους εύρεσης τμημάτων των μη-διαγώνιων πλοκάδων του spike-μητρώου (δηλαδή των πλοκάδων V_j , W_j).

Πιο συγκεκριμένα, μπορεί να δειχθεί ότι το μέγεθος (magnitude) των στοιχείων των πλοκάδων V_j μειώνεται καθώς κινούμαστε από κάτω προς τα πάνω ενώ το μέγεθος των στοιχείων των πλοκάδων W_j αποσβένει καθώς κινούμαστε από πάνω προς τα κάτω [2]. Όσο περισσότερο διαγώνια κυρίαρχο είναι το μητρώο A , τόσο πιο έντονη είναι η απόσβεση αυτή.

Λήμμα 2.3. Εστω $A \in \mathbb{R}^{n \times n}$ ένα αυστηρώς διαγώνια κυρίαρχο μητρώο με αντίστροφο βαθμό κυριαρχίας ϵ . Τότε, η τελευταία $m \times m$ πλοκάδα του V_j , $V_j^{(1)}$, ικανοποιεί την ανισότητα

$$\|V_j^{(1)}\|_\infty \leq \epsilon. \quad (2.9)$$

Απόδειξη. Βλέπε [2]. □

Το Λήμμα 2.3 παραδέτει ένα άνω όριο για το μέγεθος κάθε στοιχείου το οποίο περιέχεται στη πλοκάδα $V_j^{(1)}$, όπου είναι προφανές ότι όσο περισσότερο διαγώνια κυρίαρχο είναι το μητρώο (επομένως τόσο μικρότερος είναι ο αντίστροφος βαθμός κυριαρχίας), τόσο μικρότερο είναι το άνω όριο του μεγέθους των στοιχείων στη πλοκάδα $V_j^{(1)}$. Το Λήμμα 2.3 αποτελεί τη βάση του Θεωρήματος 2.4.

Θεώρημα 2.4. Εστω $A \in \mathbb{R}^{n \times n}$ ένα αυστηρώς διαγώνια κυρίαρχο μητρώο με αντίστροφο βαθμό κυριαρχίας ϵ και έστω $q = \lceil n_j/m \rceil$ για κάποιο j . Τότε, για τα $m \times m$ υπομητρώα $V_j^{(1)}, \dots, V_j^{(q)}$ ισχύει ότι

$$\|V_j^{(j)}\| \leq \epsilon^j, \quad \forall j, \quad j = 1, \dots, q. \quad (2.10)$$

Απόδειξη. Βλέπε [2]. □

Εφόσον για τη διάσταση του μητρώου A , n , ισχύει ότι $n \gg m$ τότε είναι λογικό να υποθέσουμε ότι τα στοιχεία τα οποία βρίσκονται στο άνω μέρος των πλοκάδων V_j όπως και αυτά τα οποία βρίσκονται στο κάτω μέρος των πλοκάδων W_j είναι αρκετά μικρά σε μέγεθος και επομένως μπορούν, κατά προσέγγιση, να αγνοηθούν.

Το spike-σύστημα $SX = G$ μπορεί και πάλι να απλοποιηθεί σε ένα μικρότερο, ίδιων διαστάσεων με το σύστημα $S_r X_r = G_r$ τις μεθόδου Spike. Τώρα όμως, σε αντίθεση με τη προηγούμενη περίπτωση, το προκύπτον σύστημα $S_t X_t = G_t$ είναι διαγώνιο κατά πλοκάδες και θα έχει τη μορφή (στιγμιότυπο για $p = 3$ επεξεργαστές),

$$\begin{bmatrix} I & V_1^{(1)} & 0 & 0 \\ W_2^{(q)} & I & 0 & 0 \\ 0 & 0 & I & V_2^{(1)} \\ 0 & 0 & W_3^{(q)} & I \end{bmatrix} \begin{bmatrix} X_1^{(1)} \\ X_2^{(q)} \\ X_2^{(1)} \\ X_3^{(q)} \end{bmatrix} = \begin{bmatrix} G_1^{(1)} \\ G_2^{(q)} \\ G_2^{(1)} \\ G_3^{(q)} \end{bmatrix}$$

Βάση της αρχικής υπόθεσης, το μητρώο A είναι αυστηρώς διαγώνια κυρίαρχο και επομένως η LU παραγοντοποίηση κάθε διαγώνιας πλοκάδας A_j , $j = 1, \dots, p$ μπορεί να πραγματοποιηθεί δίχως οδήγηση [18]. Στη συνέχεια, η πλοκάδα $V_j^{(1)}$ υπολογίζεται εμπλέκοντας μόνο τις $m \times m$ πλοκάδες που ορίζονται

από τη διασταύρωση των τελευταίων m γραμμών και τελευταίων m στηλών των παραγόντων L , U . Ο υπολογισμός της πλοκάδας $W_j^{(q)}$ απαιτεί τη πλήρη εύρεση της πλοκάδας W_j , ωστόσο μπορεί να υπολογιστεί η UL παραγοντοποίηση του A_j και τότε η πλοκάδα $W_j^{(q)}$ υπολογίζεται εμπλέκοντας μόνο τις $m \times m$ πλοκάδες που οφίζονται από τη διασταύρωση των πρώτων m γραμμών και πρώτων m στηλών των παραγόντων U , L αντίστοιχα.

Από τη στιγμή που οι πλοκάδες V_j , W_j δεν έχουν υπολογιστεί ολόκληρες, παρά μόνον τμήματα αυτών, η λύση X δεν μπορεί να υπολογιστεί με τρόπο ανάλογο της μεθόδου Spike. Αντιθέτως, τα δεξιά μέλη “απαλλάσσονται” από τη συνεισφορά των μη-διαγώνιων πλοκάδων B_j , C_j και στη συνέχεια η λύση του γραμμικού συστήματος $AX = F$ υπολογίζεται μέσω των τύπων:

$$\begin{cases} A_1 X_1 = F_1 - B_1 X_2^{(q)} & j = 1 \\ A_j X_j = F_j - B_j X_{j+1}^{(q)} - C_j X_{j-1}^{(1)} & 2 \leq j \leq p-1 \\ A_p X_p = F_p - C_p X_{p-1}^{(1)} & j = p \end{cases}$$

Αλγόριθμος 7 Ο αλγόριθμος TR-SPIKE

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $p \in \mathbb{Z}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

{Κάθε επεξεργαστής j :

if $2 \leq j \leq p-1$ then

Τυπολογισμός των $A_j = L_j U_j$, $A_j = U_j L_j$ (χωρίς οδήγηση) και των $V_j^{(1)}$, $W_{j+1}^{(q)}$, $G_j^{(1)}$, $G_j^{(q)}$

else if $j = 1$ then

Τυπολογισμός του $A_j = L_j U_j$ (χωρίς οδήγηση) και των $V_1^{(1)}$, $G_1^{(1)}$

else

Τυπολογισμός του $A_j = U_j L_j$ (χωρίς οδήγηση) και των $W_p^{(q)}$, $G_p^{(q)}$

end if

{Τώρα όλοι οι επεξεργαστές}

Επίλυση του συστήματος $S_t X_t = G_t$

Επίλυση των παρακάτω γραμμικών συστημάτων (μέσω της παραγοντοποίησης LU)

$$\begin{cases} A_1 X_1 = F_1 - B_1 X_2^{(q)} & j = 1 \\ A_j X_j = F_j - B_j X_{j+1}^{(q)} - C_j X_{j-1}^{(1)} & 2 \leq j \leq p-1 \\ A_p X_p = F_p - C_p X_{p-1}^{(1)} & j = p \end{cases}$$

Ο αλγόριθμος TR-Spike επιδεικνύει καλύτερη απόδοση στις παράλληλες αρχιτεκτονικές καθώς το κομμάτι που αφορά την επικοινωνία μεταξύ επεξεργαστών, η επίλυση του $S_t X_t = G_t$, είναι τώρα πιο εύκολο να λυθεί παράλληλα αφού η επικοινωνία περιορίζεται μόνο μεταξύ γειτονικών επεξεργαστών. Κέρδος επίσης προκύπτει και από το μερικό μόνο υπολογισμό των spike-πλοκάδων, ειδικά σε αρχιτεκτονικές όπου η επικοινωνία μεταξύ επεξεργαστών είναι αρκετά πιο χρονοβόρα από ότι η εκτέλεση αριθμητικών πράξεων σε κάθε επεξεργαστή. Τέλος, να σημειώσουμε ότι αν το μητρώο δεν είναι αρκετά διαγώνια κυρίαρχο, τότε ενδέχεται το τμήμα των spike-πλοκάδων το οποίο θα περικοπεί να μην είναι αμελητέο. Τότε ο αλγόριθμος TR-Spike παύει να είναι ένας άμεσος επιλυτής και ίσως χρειαστεί ένα εξωτερικό επαναληπτικό σχήμα (η σύγκλιση όμως συνήθως θα είναι τάχιστη - στην ουσία ο αλγόριθμος TR-Spike δρα ως προκαταστάτης).

2.2 Επίλυση γενικών, αραιών γραμμικών συστημάτων

2.2.1 Ο αλγόριθμος PSpike

Η μέθοδος Spike μπορεί να λειτουργήσει και ως προκαταστάτης για την επίλυση γενικών, αραιών γραμμικών συστημάτων της μορφής $AX = F$, βλέπε [5]. Το αλγόριθμικό σχήμα βασίζεται σε δύο στάδια, α) αναδιάταξη (reordering) των στοιχείων του μητρώου A ώστε να μειωθεί το εύρος του και εξαγωγή επιλεγμένου αριθμού διαγωνίων γύρω από τη κύρια διαγώνιο, και β) δημιουργία ενός μητρώου $M \in \mathbb{R}^{n \times n}$ το οποίο περιέχει τις διαγωνίους που επελέγησαν από το βήμα (α) και χρησιμοποιείται ως προκαταστάτης για την επίλυση του $AX = F$ μέσω κάποιας επαναληπτικής μεθόδου. Ο προκαταστάτης M είναι ταινιακός και επομένως κάθε εσωτερικό γραμμικό σύστημα μπορεί να λυθεί μέσω της μεθόδου Spike.

Για να είναι αποδοτικός ο προκαταστάτης (δηλαδή να επιτευχθεί σύγκλιση μετά από μικρό αριθμό επαναλήψεων) θα πρέπει τα στοιχεία τα οποία απαρτίζουν το μητρώο M να είναι τα μεγαλύτερα κατά απόλυτη τιμή στοιχεία του A (ή εκ των μεγαλύτερων) ώστε να ελαχιστοποιηθεί η τάξη του μητρώου $\hat{A} - M$ όπου \hat{A} είναι το αναδιαταγμένο μητρώο A . Ιδανικά, ο προκαταστάτης M έχει πολύ μικρό ημι-εύρος.

Αλγόριθμος 8 Αλγόριθμικό σχήμα επίλυσης γενικών, αραιών συστημάτων μέσω του αλγορίθμου PSPIKE

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $p \in \mathbb{Z}$, $tol \in \mathbb{R}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

1. Επιλογή αλγορίθμου αναδιάταξης του μητρώου $A \rightarrow \hat{A}$
 2. Ορισμός του μητρώου M
 3. Επίλυση του γραμμικού συστήματος $M^{-1}\hat{A}X = M^{-1}\hat{F}$ μέσω μιας επαναληπτικής μεθόδου υποχώρου Krylov (πχ BiCGSTAB), όπου η επίλυση των εσωτερικών γραμμικών συστημάτων της μορφής $My = r$ πραγματοποιείται μέσω του αλγορίθμου PSPIKE.
-

Ο αλγόριθμος PSpike πραγματοποιεί την LU παραγοντοποίηση της διαγώνιας πλοκάδας M_j , $M_j = L_j U_j$ βάσει του επιλυτή Pardiso[17], έναν επιλυτή γενικών, αραιών γραμμικών συστημάτων και υπό μια έννοια αποτελεί επέκταση του αλγορίθμου TR-Spike. Η διαφοροποίηση από τον αλγόριθμο TR-Spike έγκειται στο ότι ο επιλυτής Pardiso έχει τη δυνατότητα να υπολογίζει επιλεγμένα τιμήματα της λύσης ενός γραμμικού συστήματος με μία και μόνο παραγοντοποίηση. Επίσης, ο αλγόριθμος PSpike δεν απαιτεί το μητρώο A να είναι αυστηρώς διαγώνια κυρίαρχο αφού επιδρά μόνο στο προκαταστάτη M . Όπως αναφέρθηκε και νωρίτερα, η μόνη προύποθεση είναι το M να περιέχει, κατά μια έννοια, τα στοιχεία του A τα οποία είναι τα μεγαλύτερα κατά απόλυτη τιμή. Η απάτηση αυτή μπορεί να γίνει περισσότερο συγκεκριμένη βάσει ενός κριτηρίου σύμφωνα με το οποίο ο λόγος $\|M\|_F / \|A\|_F$ είναι μεγαλύτερος από ένα μια τιμή, πχ. 0.95. Καθώς η τιμή αυτή αυξάνεται, περιμένουμε ότι ο M θα είναι περισσότερο αποδοτικός ως προκαταστάτης καιών και ότι το (ημι)-εύρος του θα αυξάνεται.

2.2.2 Ο αλγόριθμος WB-Spike

Ο αλγόριθμος WB-Spike αποτελεί μια επέκταση του αλγορίθμου WB-Spike (κατά συνέπεια και της μεθόδου Spike) για την επίλυση γραμμικών συστημάτων των οποίων η σταθμισμένη φασματική αναδιάταξη δε καταφέρνει να συγκεντρώσει μεγάλο ποσοστό της νόρμας του μητρώου A αρκετά κοντά στη κύρια διαγώνιο του (ισοδύναμα, το ημι-εύρος του προκαταστάτη δεν είναι αρκετά μικρό).

Αλγόριθμος 9 Ο αλγόριθμος PSpike - Το βήμα προκατάστασης.

Είσοδος: $M \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $p \in \mathbb{Z}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

{Κάθε επεξεργαστής j :

Τπολογισμός της παραγοντοποίησης $M_j = L_j U_j$ μέσω του Pardiso
if $2 \leq j \leq p-1$ then

Τπολογισμός των $V_j^{(1)}$, $W_{j+1}^{(q)}$, $G_j^{(1)}$, $G_j^{(q)}$ (Pardiso)

else if $j=1$ then

Τπολογισμός των $V_1^{(1)}$, $G_1^{(1)}$ (Pardiso)

else

Τπολογισμός των $W_p^{(q)}$, $G_p^{(q)}$ (Pardiso)

end if

{Τώρα όλοι οι επεξεργαστές}

Επίλυση του συστήματος $S_t X_t = G_t$

Επίλυση των παρακάτω γραμμικών συστημάτων μέσω του Pardiso

$$\begin{cases} L_1 U_1 X_1 = F_1 - B_1 X_2^{(q)} & j=1 \\ L_j U_j X_j = F_j - B_j X_{j+1}^{(q)} - C_j X_{j-1}^{(1)} & 2 \leq j \leq p-1 \\ L_p U_p X_p = F_p - C_p X_{p-1}^{(1)} & j=p \end{cases}$$

Ο αλγόριθμος WB-Spike βασίζεται και αυτός στην αναδιάταξη του μητρώου συντελεστών. Ωστόσο, αφού η αναδιάταξη λάβει μέρος, στη συνέχεια εξάγεται ως προκαταστάτης ένα υπομητρώο M το οποίο δεν είναι απαραίτητα ταινιακό. Η άρση του περιορισμού της ταινιακής δομής του προκαταστάτη μπορεί να οδηγήσει στη βελτίωση της σύγκλισης καθώς δύναται να μειώσει ακόμα περισσότερο τη τάξη του μητρώου $\hat{A} - M$. Από την άλλη, η επιλογή ενός μη-ταινιακού προκαταστάτη μπορεί να αυξήσει το κόστος επίλυσης με αυτόν, πχ. το γέμισμα της παραγοντοποίησης των διαγώνιων πλοκάδων $M_j = L_j U_j$, $j = 1, \dots, p$ μπορεί τώρα να είναι αρκετά μεγαλύτερο.

Το πρακτικό πλεονέκτημα του αλγορίθμου WB-Spike είναι ότι χάρη στις λιγότερες επαναλήψεις της επαναληπτικής μεθόδου για την επίλυση του $AX = F$, η επικοινωνία μεταξύ των επεξεργαστών μπορεί να μειωθεί αρκετά ενώ το πρόσθετο υπολογιστικό κόστος που εισάγεται από την άρση του περιορισμού ενός προκαταστάτη μικρού εύρους είναι κατανεμημένο παράλληλα.

Ο Αλγόριθμος 10 απεικονίζει τον αλγόριθμο WB-Spike ο οποίος βασίζεται στη τεχνική επίλυσης του αλγορίθμου PSpike. Όπως αναφέρθηκε, οι διαγώνιες πλοκάδες M_j , $j = 1, \dots, p$ δεν είναι κατά ανάγκην ταινιακά μητρώα και επομένως τα B_j^M , C_j^M έχουν αυθαίρετη δομή. Επομένως, μπορούμε να επιλέξουμε τον αριθμό των στηλών από τις οποίες θα αποτελούνται οι μη διαγώνιες πλοκάδες B_j^M , C_j^M . Ο αριθμός αυτός ορίζεται από τη μεταβλητή bw στον Αλγόριθμο 10. Καθώς η τιμή του bw μεγαλώνει, περιμένουμε ταχύτερη σύγκλιση και μειωμένο ρυθμό επικοινωνίας μεταξύ των επεξεργαστών. Είναι προφανές ότι για τη τιμή $bw = 0$, η μέθοδος ταυτίζεται με το προκαταστάτη Jacobi κατά πλοκάδες. Περισσότερα για τον αλγόριθμο WB-Spike μπορούν να βρεθούν στο [31] όπου εισάγεται ο αλγόριθμος DDPS.

Αξίζει να επιμείνουμε λίγο ακόμα στον αλγόριθμο WB-Spike προκειμένου να εξετάσουμε αποδοτικούς

Αλγόριθμος 10 Ο αλγόριθμος WB-Spike

Είσοδος: $M \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $p \in \mathbb{Z} \geq 1$, $bw \in \mathbb{Z} \geq 0$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

```
{Κάθε επεξεργαστής j:}
ορισμός των  $B_j \in \mathbb{R}^{n_j \times bw}$ ,  $C_j \in \mathbb{R}^{n_j \times bw}$ 
if  $2 \leq j \leq p - 1$  then
     $M_j[V_j, W_j, G_j] = [B_j, C_j, F_j]$ 
else if  $j = 1$  then
     $M_1[V_1, G_1] = [B_1, F_1]$ 
else
     $M_p[W_p, G_p] = [C_p, F_p]$ 
end if
```

{Τώρα όλοι οι επεξεργαστές}

Επίλυση του συστήματος $S_r X_r = G_r$

Υπολογισμός των τελικών λύσεων

$$\begin{cases} X'_1 = G'_1 - V'_1 X_2^{(q)} & j = 1 \\ X'_j = G'_j - V'_j X_{j+1}^{(q)} - W'_j X_{j-1}^{(1)} & 2 \leq j \leq p - 1 \\ X'_p = G'_p - W'_p X_{p-1}^{(1)} & j = p \end{cases}$$

τρόπους υλοποίησής της. Όπως αναφέρθηκε, στη περίπτωση που η σταθμισμένη φασματική αναδιάταξη αποτύχει να συγκεντρώσει ένα πολύ μεγάλο μέρος της νόρμας του αναδιεταγμένου μητρώου A κοντά στη κύρια διαγώνιο του, τότε η εξαγωγή ενός ταινιακού M με μικρό εύρος δε θα αυξήσει κατά πολύ το ρυθμό σύγκλισης της επαναληπτικής μεθόδου που χρησιμοποιείται για την επίλυση του $AX = F$. Η επιλογή ενός προκαταστάτη πολύ μεγάλου εύρους ναι μεν θα αυξήσει το ρυθμό σύγκλισης, ωστόσο δεν είναι ιδεατή καθώς θα αυξήσει το υπολογιστικό κόστος επίλυσης με το προκαταστάτη καθώς οι διαγώνιες πλοκάδες M_1, \dots, M_p ενδέχεται να μη μπορούν να παραγοντοποιηθούν με μικρό κόστος. Από την άλλη, η παραγοντοποίηση των M_1, \dots, M_p είναι αναγκαία για την επίλυση των επιμέρους συστημάτων, μια επίλυση όμως η οποία εκτελείται τοπικά σε κάθε επεξεργαστή και μπορεί να επιτευχθεί με χρήση επαναληπτικών μεθόδων. Πράγματι, η χρήση επαναληπτικών μεθόδων λύνει το πρόβλημα πολυπλοκότητας μνήμης που θα εισαγόταν από τη χρήση άμεσων μεθόδων και επιτρέπει την επιλογή ενός προκαταστάτη ο οποίος μπορεί να είναι αρκετά μεγάλου εύρους ώστε να περιέχει το επιλυμητό ποσοστό της συνολικής νόρμας του A . Για πρακτικούς λόγους, θα πρέπει να θέσουμε τη μεταβλητή bw ως $bw \ll n$.

Συνολικά, οι διάφορες επιλογές της οικογένειας αλγορίθμων Spike συνοψίζονται ως εξής:

- Οι πλοκάδες V, W_j μπορούν να υπολογιστούν:
 1. Μερικώς ή πλήρως
 2. Προσεγγιστικά
- Τα γραμμικά συστήματα με το A_j ως μητρώο συντελεστών μπορούν να λυθούν
 1. Με χρήση άμεσων μεθόδων (LU, Cholesky)

2. Με χρήση επαναληπτικών μεθόδων (με προκατάσταση)
 - To Spike-σύστημα μπορεί να λυθεί
 1. Με χρήση άμεσων μεθόδων
 2. Με χρήση επαναληπτικών μεθόδων
 3. Με χρήση περικοπής

Κεφάλαιο 3

Επιτάχυνση της οικογένειας αλγορίθμων Spike μέσω τεχνικών επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στην οικογένεια αλγορίθμων Spike και σχολιάσαμε την εφαρμογή τους για την επίλυση ταινιών και γενικών, αραιών γραμμικών συστημάτων. Όπως καταστήσαμε σωφρές, πέραν των πολλών δεξιών μελών τα οποία ενδέχεται να περιέχονται στο F , γραμμικά συστήματα με πολλά δεξιά μέλη προκύπτουν και κατά τον υπολογισμό των μηδιαγώνιων πλοκάδων του spike-μητρώου, $A_j[V_j, W_j] = [\hat{B}_j, \hat{C}_j]$, $j = 2, \dots, p-1$ και $A_j V_j = \hat{B}_j$ ($A_j W_j = \hat{C}_j$) αν $j = 1$ ($j = p$).

Στο παρών κεφάλαιο που αποτελεί και το πιο σημαντικό της παρούσας εργασίας, θα αναφερθούμε στις βασικές μεθόδους επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη μέσω επαναληπτικών επιλυτών και θα δούμε ότι αυτές οι μέθοδοι μπορούν να οδηγήσουν σε αρκετά πιο γρήγορη επίλυση από ότι αν κάθε δεξιό μέλος επιλυόταν ανεξάρτητα. Στη συνέχεια, θα ενώσουμε τη παρούσα θεωρία με αυτή του προηγουμένου κεφαλαίου και θα παρουσιάσουμε αλγορίθμικά σχήματα τα οποία έχουν ως στόχο την επιτάχυνση της οικογένειας αλγορίθμων Spike μέσω της εφαρμογής των τεχνικών επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη. Απότατος στόχος μας είναι να αντικαταστήσουμε τον άμεσο επιλυτή τοπικά σε κάθε επεξεργαστή με ένα επαναληπτικό επιλυτή πολλών δεξιών μελών.

3.1 Μέθοδοι επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη

Θα ξεκινήσουμε τη παρούσα ενότητα με ένα παράδειγμα. Έστω το αρχικό γραμμικό σύστημα $AX = F$ και ότι επιλύουμε την ειδική περίπτωση $F = [f^{(1)}, f^{(2)}]$ και επομένως πρέπει να επιλύσουμε τα συστήματα $Ax^{(1)} = f^{(1)}$ και $Ax^{(2)} = f^{(2)}$. Έστω τώρα ότι συμβαίνει να ισχύει ότι $f^{(2)} = \delta f^{(1)}$ για κάποιο μη-μηδενικό $\delta \in \mathbb{R}$ - ένα σενάριο το οποίο ίσως είναι λίγο εξιδανικευμένο στη πράξη, εξυπηρετεί όμως τη προσπάθεια μας να αναδείξουμε ορισμένα ενδιαφέροντα θέματα.

Εφαρμόζοντας τη κλασσική προσέγγιση, θα επιλύμε τα δύο συστήματα ανεξάρτητα το ένα από το άλλο (πχ. με χρήση της μεθόδου BiCGSTAB). Ωστόσο, όπως είναι προφανές από τη σχέση μεταξύ των δεξιών μελών, το $f^{(2)}$ είναι παράλληλο προς το $f^{(1)}$ και επομένως το ίδιο ισχύει και για τις λύσεις $x^{(1)}$, $x^{(2)}$ (φυσικά το A είναι αντιστρέψιμο)! Πιο συγκεκριμένα, αν $x^{(1)} = A^{-1}f^{(1)}$ είναι η λύση του γραμμικού συστήματος το οποίο ορίζεται από το πρώτο δεξί μέλος, τότε η λύση του γραμμικού συστήματος που ορίζεται από το δεύτερο δεξί μέλος είναι $x^{(2)} = A^{-1}f^{(2)} = A^{-1}\delta f^{(1)} = \delta x^{(1)}$. Επομένως, λύνοντας μόνο την εξίσωση $Ax^{(1)} = f^{(1)}$ μπορούμε αμέσως να υπολογίσουμε και τη λύση της εξίσωσης $Ax^{(2)} = f^{(2)}$.

Το συμπέρασμα του παραπάνω παραδείγματος είναι ότι αν τα δεξιά μέλη μοιράζονται κάποια ιδιότητα, όπως το ότι μια τυχαία τους βάση έχει διάσταση ϕ και $\phi \ll s$ (επομένως η τάξη του μητρώου F είναι ϕ), τότε μπορούμε να επιλύσουμε το γραμμικό σύστημα $AX = F$ χωρίς να χρειαστεί να λύσουμε όλα τα επί μέρους γραμμικά συστήματα τα οποία σχηματίζονται από τα s δεξιά μέλη. Φυσικά αυτή η ιδέα από μόνη της είναι μη πλήρης αφού δε ξέρουμε ακόμα για ποια δεξιά μέλη αρκεί να επιλύσουμε τις αντίστοιχες εξισώσεις. Σε μια πρώτη εκδοχή μπορούμε να δοκιμάσουμε και να βρούμε μια βάση για το γραμμικό χώρο στηλών του F , $\text{range}(F)$, μέσω της παραγοντοποίησης QR ή της παραγοντοποίησης SVD. Αν $s \ll n$ όπως είναι και η αρχική μας υπόθεση, τότε το κυρίαρχο κόστος αυτής της προσέγγισης είναι $O(n)$ το οποίο σε πρώτη ανάγνωση δεν είναι απαγορευτικό. Ενδέχεται όμως το κόστος αποθήκευσης των ενδιάμεσων μητρώων να είναι ιδιαίτερα σημαντικό καθώς τα δεξιά μέλη ίσως είναι αραιά όπως ισχύει στη φάση προεπεξεργασίας της μεθόδου Spike. Όπως θα δούμε παρακάτω ωστόσο, για τη περίπτωση της μεθόδου Spike μπορούμε να χρησιμοποιήσουμε αυτού του είδους τη προσέγγιση βασιζόμενοι σε μια απλή παρατήρηση.

Στη περίπτωση που η παραπάνω προσέγγιση δεν είναι ικανοποιητική για λόγους πολυπλοκότητας πράξεων ή μηνίμης, μπορούμε να χρησιμοποιήσουμε αλγορίθμους οι οποίοι έχουν την ιδιότητα να ανακαλύπτουν αυτόμata τυχόν ύπαρξη μη πλήρους τάξης του μητρώου F . Η παραπάνω ιδιότητα ισχύει θεωρητικά (άπειρη αριθμητικής ακρίβεια), ωστόσο έχει αποδειχθεί ότι δουλεύει αρκετά καλά και στη πράξη. Οι μέθοδοι αυτές ονομάζονται μέθοδοι σπόρου (seed methods).

Μέθοδοι οι οποίες αντικατωπίζουν συνολικά την επίλυση γραμμικών συστημάτων με πολλά δεξιά μέλη μπορούν να χρησιμοποιηθούν και στη περίπτωση πλήρους τάξης του μητρώου F . Σε αυτή τη κατηγορία ανήκουν οι μέθοδοι πλοκάδας (block methods) οι οποίες επιλύουν ταυτόχρονα όλα τα δεξιά μέλη. Θα αναλύσουμε και αυτή την οικογένεια μεθόδων παρακάτω, προς το παρόν θα πούμε μόνον ότι αυτού του είδους οι μέθοδοι μπορούν να είναι χρήσιμες στην περίπτωση που οι ιδιοτιμές που βρίσκονται στα άκρα του φάσματος του A δημιουργούν συστάδα, δηλαδή είναι σε μικρή απόσταση μεταξύ τους.

Στη συνέχεια παραθέτουμε εν συντομίᾳ τις δύο βασικές κλάσεις μεθόδων επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη. Τις μεθόδους σπόρου και τις μεθόδους πλοκάδας.

3.1.1 Μέθοδοι σπόρου

Οι μέθοδοι σπόρου αρχικά επιλέγουν ένα δεξί μέλος (έστω το πρώτο) το οποίο ονομάζεται και σύστημα σπόρου, και επιλύουν το αντίστοιχο γραμμικό σύστημα. Καθώς το επιλεγμένο σύστημα επιλύεται, οι προτεινόμενες λύσεις των γραμμικών συστημάτων τα οποία δεν έχουν ακόμα επιλυθεί ανανεώνονται σε κάθε επανάληψη. Η ανανέωση συνήθως πραγματοποιείται μέσω κάποιων κριτηρίων (ορθογώνιας ή πλάγιας) προβολής ως προς το συμπλήρωμα του Krylov υπόχωρου ο οποίος σχηματίζεται κατά τη διάρκεια επίλυσης. Στη συνέχεια, αφού το επιλεγμένο σύστημα σπόρους επιλυθεί, ένα νέο σύστημα επιλέγεται και η διαδικασία

ζεκινάει από την αρχή [20, 28, 29, 43, 44, 45]. Ο Αλγόριθμος 11 συνοψίζει τη φιλοσοφία των μεθόδων σπόρου για την επίλυση γραμμικών συστημάτων με πολλά δεξιά μέλη. Ο υποδείκτης i δεικτοδοτεί την επανάληψη την οποία εκτελεί το σύστημα σπόρος, ο πρώτος υπερδείκτης δεικτοδοτεί το σύστημα το οποίο λύνεται ενώ ο δεύτερος υπερδείκτης δεικτοδοτεί το σύστημα του οποίου η προσέγγιση ανανεώνεται.

Αλγόριθμος 11 Ο γενικός αλγόριθμος των μεθόδων σπόρου (SEED)

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, tol
Έξοδος: $X \in \mathbb{R}^{n \times s}$

```

for  $j = 1, \dots, s$  do
    for  $i = 1, \dots$  do
        το  $j$ -στό γραμμικό σύστημα εκτελεί την  $i$ -στή επανάληψη
        for  $g = j + 1, \dots, s$  do
            ανανέωση της προσέγγισης του  $x_i^{(j,g)}$ 
        end for
    end for
end for

```

Ο αλγόριθμος SSCG (Αλγόριθμος 12) απεικονίζει έναν αλγόριθμο σπόρου για ΣΘΟ μητρώα, ο οποίος βασίζεται στον αλγόριθμο CG [20]. Ο αλγόριθμος SSCG επιλέγει το j -στό δεξιά μέλος και λύνει το αντίστοιχο γραμμικό σύστημα ($Ax^{(j)} = f^{(j)}$) με χρήση του αλγορίθμου CG. Κατά τη διάρκεια της επίλυσης, όλα τα δεξιά μέλη τα οποία δεν έχουν επιλυθεί, ανανεώνουν τη προσέγγιση της λύσης τους μέσω κάποιων κριτηρίων ορθογωνιότητας.

Αλγόριθμος 12 Ο αλγόριθμος Single-Seed Conjugate Gradient (sscg) [20]

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $tol \in \mathbb{R}$
Έξοδος: $X \in \mathbb{R}^{n \times s}$

```

 $R_0 = F - AX_0$ 
 $P_0 = R_0$ 
for  $j = 1, \dots, s$  do
    while  $\|r_i^{(j)}\| / \|f^{(j)}\| \geq tol$  do
         $\alpha = ((r_i^{(j)})^\top r_i^{(j)}) / ((p_i^{(j)})^\top A p_i^{(j)})$ 
         $x_{i+1}^{(j)} = x_i^{(j)} + p_i^{(j)} \alpha$ 
         $r_{i+1}^{(j)} = r_i^{(j)} - A p_i^{(j)} \alpha$ 
         $\beta = ((r_{i+1}^{(j)})^\top r_{i+1}^{(j)}) / ((r_i^{(j)})^\top r_i^{(j)})$ 
         $p_{i+1}^{(j)} = r_{i+1}^{(j)} + p_i^{(j)} \beta$ 
        for  $k = j + 1, \dots, s$  do
             $\eta_i^{(j,k)} = (p_i^{(j)})^\top r_i^{(j,k)} / (p_i^{(j)})^\top A p_i^{(j)}$ 
             $x_{i+1}^{(j,k)} = x_i^{(j,k)} + \eta_i^{(j,k)} p_i^{(j)}$ 
             $r_{i+1}^{(j,k)} = r_i^{(j,k)} - \eta_i^{(j,k)} A p_i^{(j)}$ 
        end for
    end while
end for

```

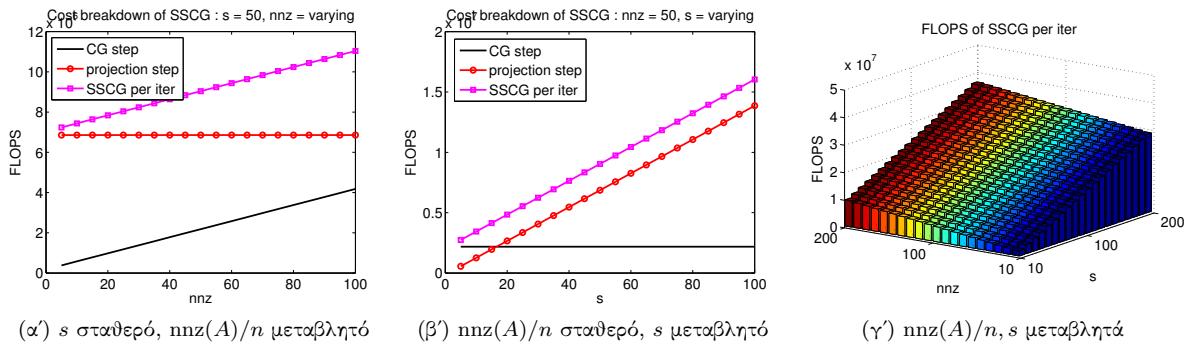
Ο Πίνακας 3.1 περιέχει το κόστος μίας επανάληψης του αλγορίθμου SSCG καθώς και αυτό της ανανέωσης της προσέγγισης των λύσεων, συναρτήσει των γραμμικών συστημάτων των δεξιών μελών τα οποία δεν έχουν ακόμα επιλυθεί. Από το πίνακα, είναι εύκολο να διαπιστωθεί ότι η ανανέωση της προσέγγισης εισάγει αριθμό

πράξεων, γραμμικό ως προς το πλήθος των δεξιών μελών. Όσον αφορά τη πολυπλοκότητα μνήμης, από τον Αλγόριθμο 12 μπορεί να υπολογιστεί ότι ο αλγόριθμος σποράς εισάγει την ανάγκη αποθήκευσης $2(s - k)$ πρόσθετων διανυσμάτων μεγέθους n όπου k είναι ο αριθμός των γραμμικών συστημάτων τα οποία έχουν ήδη επιλεγεί ως σύστημα σπόρος.

Πίνακας 3.1: Πλήθος βασικών πράξεων μίας επανάληψης του αλγορίθμου SSCG. Σε αγκύλες το συνολικό αριθμητικό κόστος. Παρόμοια στοιχεία μπορούν να βρεθούν και στο [39].

	MV	DOT	AXPY
Κόστος επανάληψης CG	$1 ([2\text{nnz} - 1])$	$2 ([4n - 2])$	$3 ([6n])$
Κόστος προβολής	-	$(s - j) ([((s - j)(2n - 1)])$	$2(s - j) ([((s - j)(4n)])$

Μένοντας λίγο ακόμα στο κόστος του αλγορίθμου SSCG και των μεθόδων σπόρου γενικότερα, μπορούμε να διακρίνουμε δύο περιπτώσεις. Στη πρώτη περίπτωση, το μητρώο A είναι αρκετά αραιό, δηλαδή $(\text{nnz}(A)/n) \approx 1$. Τότε είναι εύκολο να δούμε από το Πίνακα 3.1 ότι το κόστος της φάσης των προβολών είναι αρκετά υπολογίσιμο και μάλιστα για $s >> 1$, το κύριο κόστος του αλγορίθμου SSCG προέρχεται από εκεί. Επομένως για να έχουμε κέρδος από άποψη υπολογιστικής πολυπλοκότητας, θα πρέπει η σύγκλιση των δεξιών μελών τα οποία προβάλλονται στον υποχώρῳ να είναι αρκετά βελτιωμένη. Στη δεύτερη περίπτωση, το πλήθος των μη-μηδενικών στοιχείων του A είναι μεγάλο (το A όμως παραμένει αραιό). Τότε το κόστος των προβολών δεν είναι ιδιαίτερα μεγάλο, παρ' όλα αυτά για μεγάλες τιμές του s , για μεγάλο αριθμό δεξιών μελών δηλαδή, και πάλι το κόστος αυτής της φάσης θα ξεπεράσει το κόστος εφαρμογής του αλγορίθμου CG κατά την επίλυση των πρώτων δεξιών μελών.



Σχήμα 3.1: Πλήθος πράξεων μίας επανάληψης του αλγορίθμου sscg συναρτήσει των $\text{nnz}(A)/n$ και s .

Το Σχήμα 3.1 απεικονίζει το υπολογιστικό κόστος μιας επανάληψης του αλγορίθμου SSCG συναρτήσει είτε του πλήθους των μη-μηδενικών στοιχείων ανά γραμμή του A για $n = 2 \times 10^4$ (3.1α'), είτε του πλήθους των δεξιών μελών s (3.1β'), είτε και των δύο μαζί (3.1γ'). Από το τελευταίο σχήμα μπορούμε να διαπιστώσουμε ότι για συγκεκριμένο μέγεθος n , διπλασιασμός του s οδηγεί σε μεγαλύτερη αύξηση του κόστους ανά επανάληψη από ότι διπλασιασμός του αριθμού των μη-μηδενικών στοιχείων ανά γραμμή του A με τη διαφορά να είναι περισσότερο έκδηλη για μικρότερες τιμές μη-μηδενικών στοιχείων ανά γραμμή.

Το σημαντικό ερώτημα τώρα είναι το γιατί να “πληρώσουμε” το πρόσθετο κόστος των μεθόδων σποράς αντί να λύσουμε κάθε δεξιό μέλος ανεξάρτητα. Πιο συγκεκριμένα, γιατί αυτή η τεχνική μπορεί να οδηγήσει σε ένα πιο γρήγορο υπολογισμό του γραμμικού συστήματος $AX = F$; Στην αρχή του κεφαλαίου δώσαμε ένα διαισθητικό παράδειγμα μίας περίπτωσης όπου οι μέθοδοι σπόρου μπορούν να μας βοηθήσουν και στη

συνέχεια όταν δούμε πως αυτό εξηγείται θεωρητικά. Όπως όταν δούμε στο αμέσως επόμενο θεώρημα, οι μέθοδοι σπόρου μπορούν να είναι χρήσιμες ακόμα και στη περίπτωση που τα δεξιά μέλη δεν είναι συγγραμμικά σε μεγάλο βαθμό. Σημειώνουμε ότι η A -νόρμα ενός διανύσματος x ορίζεται ως

$$\|x\|_A = \sqrt{x^\top Ax}.$$

Θεώρημα 3.1. Εστω το γραμμικό σύστημα $AX = F$ όπου $s = 2$, $Ax^{(1)} = f^{(1)}$ και $Ax^{(2)} = f^{(2)}$. Υποθέτουμε ότι το πρώτο σύστημα έχει ήδη λυθεί με χρήση του αλγορίθμου CG σε m βήματα. Εστω ότι $x_0^{1,2}$ είναι η προσέγγιση της λύσης $x^{(2)}$ που παράγεται από τον αλγόριθμο SSCG ύστερα από την επίλυση του $Ax^{(1)} = f^{(1)}$ και $\bar{x}_0^{1,2}$ είναι τέτοιο ώστε $x^{(2)} - \bar{x}_0^{1,2}$ είναι η προβολή του $x^{(2)} - x_0^{1,2}$ πάνω στο $\{q^{(\phi)} : \phi \in I\}^\perp$ όπου $I = 1, \dots, l, n-l+1, \dots, n$. Εστω τώρα ότι $\bar{x}_i^{1,2}$ είναι η i -στή επανάληψη του συστήματος $Ax^{(2)} = b^{(2)}$ με $\bar{x}_0^{1,2}$ ως αρχική λύση. Τότε, το άνω φράγμα για το A -σφάλμα ικανοποιεί

$$\|x^{(2)} - x_i^{1,2}\|_A^2 \leq 4\|x^{(2)} - \bar{x}_i^{1,2}\|_A^2 \left(\frac{\sqrt{\kappa'} - 1}{\sqrt{\kappa'} + 1} \right)^{2i} + \delta_2$$

όπου

$$\delta_2 = \|P_m^\perp x^{(2)}\| \sum_{\phi \in I} \lambda_\phi \omega_\phi^2 \tan^2 \theta_\phi,$$

όπου $P_m^\perp = I - V_m T_m^{-1} V_m A$ είναι η A -ορθογώνια προβολή πάνω στο K_m^\perp , και \bar{p}_i είναι ένα πολυώνυμο βαθμού $to πολύ i$ με $p(0) = 1$ και $\kappa' = \frac{\lambda_{n-l+1}}{\lambda_l}$.

Απόδειξη. Βλέπε [20]. □

Το Θεώρημα 3.1 παρουσιάζει ένα γνωστό αποτέλεσμα των επαναληπτικών μεθόδων υποχώρου Krylov, το οποίο δηλώνει ότι αν ο υπόχωρος Krylov προσεγγίζει αρκετά καλά τα ιδιοδιανύσματα του A τα οποία αντιστοιχούν στις ακραίες ιδιοτιμές και τα αρχικά κατάλοιπα των γραμμικών συστημάτων τα οποία δεν έχουν ακόμα επιλυθεί προβληθούν στον υπόχωρο του ορθογώνιου συμπληρώματος αυτών των ιδιοδιανυσμάτων, τότε ο ρυθμός σύγκλισης της επίλυσης τους βελτιώνεται. Το θεώρημα επεκτείνεται και στη μη-συμμετρική περίπτωση [29]. Αξίζει να παρατηρήσουμε ότι το Θεώρημα 3.1 αναφέρεται στη περίπτωση άπειρης αριθμητικής ακρίβειας. Στα πρακτικά υπολογιστικά μοντέλα έχει δειχθεί ότι η σύγκλιση των συστημάτων τα οποία δεν έχουν επιλυθεί μπορεί να είναι όμοια με αυτή πριν της πλάγιας προβολής ως προς το συμπλήρωμα του υπόχωρου Krylov. Παραπέμπουμε στο [27] για μια προσέγγιση αποφυγής αυτού του φαινομένου.

Θεώρημα 3.2. Εστω $F = [f^{(1)}, \dots, f^{(s)}]$ όπου ισχύει $\text{rank}(F) = k$, $k < s$. Τότε, υπάρχει α , ανεξάρτητο του αριθμού επαναλήψεων m_p , τέτοιο ώστε για το κατάλοιπο των συστημάτων τα οποία δεν έχουν ακόμα επιλυθεί, να ισχύει

$$\|r_0^{(j,k)}\| \leq \alpha \sum_{h=1}^j |\beta_{m_p+1}|, \quad k = j+1, \dots, s,$$

όπου η τιμή β_{m_p+1} προέρχεται από την ακολουθία Lanczos αν θεωρήσουμε ότι το h -στό σύστημα λύνεται με τον αλγόριθμο Lanczos.

Απόδειξη. Βλέπε [20]. □

Το Θεώρημα 3.2 παρουσιάζει μια σημαντική ιδιότητα των μεθόδων σπόρου, σύμφωνα με την οποία αν το μητρώο των δεξιών μελών έχει τάξη μικρότερη από s , έστω ϕ , τότε ο αλγόριθμος 12 θα επιλύσει όλα τα

γραμμικά συστήματα μετά από φ επιλύσεις συνολικά. Επομένως, επιτρέποντας το διαμοιρασμό πληροφορίας κατά τη διάρκεια επίλυσης των διαφόρων δεξιών μελών μπορούμε να εκμεταλλευτούμε την ύπαρξη χαμηλής τάξης και να επιλύσουμε άμεσα μόνο μερικά εξ αυτών. Τα υπόλοιπα θα επιλυθούν μέσω των προβολών στον Αλγόριθμο 11. Παρόμοιο θεώρημα έχει αποδειχθεί και για μη-συμμετρικά μητρώα [29].

Στη συνέχεια θα παραθέσουμε μια νέα υλοποίηση των μεθόδων σπόρου για την επίλυση γραμμικών συστημάτων με ταινιακά μητρώα συντελεστών, τα οποία είναι είτε ΣΤΘΟ είτε αυστηρά διαγώνια κυρίαρχα, και τα δεξιά μέλη έχουν όλα τα μη-μηδενικά στοιχεία συγκεντρωμένα σε μερικές διαδοχικές γραμμές. Αφετηρία για αυτή τη νέα υλοποίηση αποτελεί το Θεώρημα 3.3.

Θεώρημα 3.3. Εστω ένα ΣΤΘΟ μητρώο $A \in \mathbb{R}^{n \times n}$ το οποίο έχει (ημι)-εύρος m . Τότε,

$$|A_{ij}^{-1}| \leq C\lambda^{|i-j|}, \quad i, j = 1, \dots, n$$

όπου

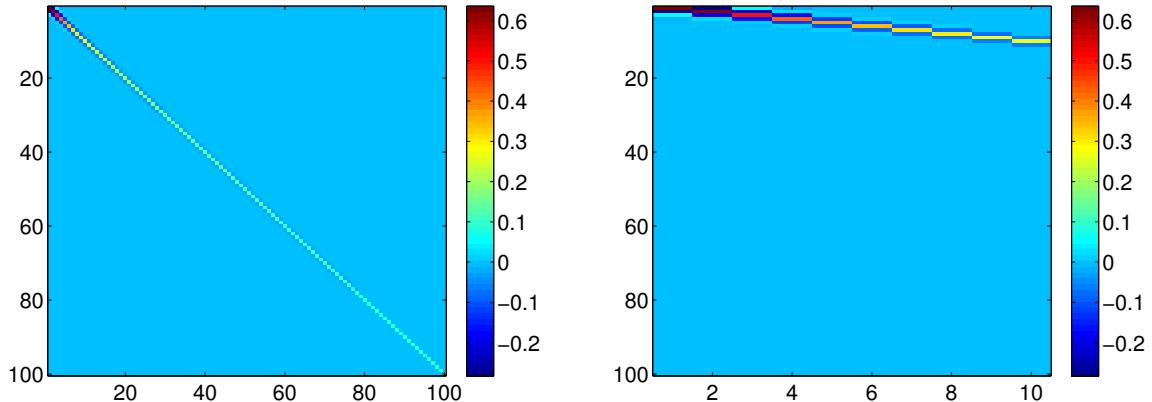
$$\lambda = \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^{2/m}$$

και

$$C = \|A^{-1}\| \max \left\{ 1, \left(1 + \sqrt{\kappa(A)} \right)^2 / (2\kappa(A)) \right\}$$

Απόδειξη. Βλέπε [33]. □

Όσο μικρότερα είναι το ημι-εύρος και ο δείκτης κατάστασης ενός ΣΤΘΟ μητρώου A , τόσο μεγαλύτερη απόσβεση, καθώς κινούμαστε μακριά από τη κύρια διαγώνιο, περιμένουμε στα στοιχεία του A^{-1} . Από την άλλη, όπως αναφέρθηκε, είναι εύκολο να διαπιστώσουμε ότι εάν τα δεξιά μέλη έχουν όλα τα μη-μηδενικά τους στοιχεία σε ένα περιορισμένο εύρος διαδοχικών γραμμών, πχ. στις τελευταίες ζ γραμμές, τότε η λύση του γραμμικού συστήματος μπορεί να υπολογιστεί μέσω του πολλαπλασιασμού του μη-μηδενικού μέρους των δεξιών μελών με τις στήλες του A^{-1} οι οποίες αντιστοιχούν σε αυτές τις μη-μηδενικές γραμμές, πχ. τις τελευταίες ζ στήλες του A^{-1} .



(α') Απόλυτες τιμές των στοιχείων του A^{-1}

(β') Απόλυτες τιμές των στοιχείων των πέντε πρώτων στηλών του A^{-1}

Σχήμα 3.2: Ταινιακό μητρώο A και η απόσβεση στα στοιχεία του A^{-1} .

Ως ένα σύντομο αλλά ενδεικτικό παράδειγμα, παραθέτουμε το Σχήμα 3.2. Σε αυτό το παράδειγμα χρησιμοποιήσαμε ένα ΣΘΟ A τετρικού μεγέθους $n = 100$ με $m = 10$. Τα στοιχεία του A επιλέχθηκαν ως $A_{ii} = 1 + i^{1/2}$, $i = 1, \dots, 100$ και $A_{ij} = 1/|i - j|$, $i, j = 1, \dots, 100$ για $i \neq j$. Το Σχήμα 3.2α' απεικονίζει το μέγεθος των στοιχείων του αντιστρόφου A^{-1} ενώ το Σχήμα 3.2β' επικεντρώνεται στις πέντε πρώτες στήλες του ίδιου μητρώου. Μπορούμε να παρατηρήσουμε ότι το μέγεθος των στοιχείων μακριά από τη κύρια διαγώνιο είναι αμελητέο σε σχέση με αυτό των στοιχείων επάνω ή πολύ κοντά στη κύρια διαγώνιο.

Αυτές οι παρατηρήσεις οδηγούν στην ιδέα της μη ανανέωσης όλων των στοιχείων των προτεινόμενων λύσεων των γραμμικών συστημάτων τα οποία δεν έχουν επιλυθεί, αφού - ανάλογα και τα χαρακτηριστικά του προβλήματος - πολλά από αυτά θα έχουν μικρό μέγεθος. Έτσι, το κόστος της ανανέωσης των προτεινόμενων λύσεων για τα μη επιλυμένα γραμμικά συστήματα μειώνεται σημαντικά αφού πλέον οι πράξεις ανανέωσης εμπλέκουν διανύσματα μεγέθους $O(\zeta)$ έναντι διανυσμάτων μεγέθους n .

Αλγόριθμος 13 Ο αλγόριθμος Sparse single-seed Conjugate Gradient (SPARS-SCG) για δεξιά μέλη με μη-μηδενικά στοιχεία στις τελευταίες ζ γραμμές

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, $tol \in \mathbb{R}$, $\delta \geq \zeta \in \mathbb{Z}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

$$(X_0)_{1:(n-\delta+1)}^{1:s} = 0$$

$$R_0 = F - AX_0$$

$$P_0 = R_0$$

for $j = 1, \dots, s$ **do**

while $\|r_i^{(j)}\|/\|f^{(j)}\| \geq tol$ **do**

$$\alpha = ((r_i^{(j)})^\top r_i^{(j)}) / ((p_i^{(j)})^\top Ap_i^{(j)})$$

$$x_{i+1}^{(j)} = x_i^{(j)} + p_i^{(j)}\alpha$$

$$r_{i+1}^{(j)} = r_i^{(j)} - Ap_i^{(j)}\alpha$$

$$\beta = ((r_{i+1}^{(j)})^\top r_{i+1}^{(j)}) / ((r_i^{(j)})^\top r_i^{(j)})$$

$$p_{i+1}^{(j)} = r_{i+1}^{(j)} + p_i^{(j)}\beta$$

for $k = j + 1, \dots, s$ **do**

$$\eta_i^{(j,k)} = (p_i^{(j)})^\top r_i^{(j,k)} / (p_i^{(j)})^\top Ap_i^{(j)}$$

$$(x_{i+1}^{(j,k)})_{(n-\delta+1):n} = (x_i^{(j,k)})_{(n-\delta+1):n} + \eta_i^{(j,k)}(p_i^{(j)})_{(n-\delta+1):n}$$

$$(r_{i+1}^{(j,k)})_{(n-2\delta+2):n} = (r_i^{(j,k)})_{(n-2\delta+2):n} - \eta_i^{(j,k)}(Ap_i^{(j)})_{(n-2\delta+2):n}$$

end for

end while

end for

Ο Αλγόριθμος 13 περιγράφει τον αλγόριθμο της αραιής μεθόδου σποράς (sparse single-seed) για την επίλυση του γραμμικού συστήματος $AX = F$ όπου A είναι ένα ταινιακό, ΣΘΟ μητρώο συντελεστών και $F \in \mathbb{R}^{n \times s}$ το μητρώο δεξιών μελών με μη-μηδενικά στοιχεία μόνο στις τελευταίες ζ γραμμές. Ο Αλγόριθμος 13 διαφέρει από τον Αλγόριθμο 12 μόνο στο σημείο ανανέωσης των προτεινόμενων λύσεων των μη-επιλυμένων συστημάτων όπου μόνο ένα υποσύνολο των γραμμών (στοιχείων) ανανεώνεται. Το ποιες γραμμές θα ανανεωθούν εξαρτάται από τη θέση των μη-μηδενικών στοιχείων του F .

Το πλεονέκτημα του Αλγορίθμου 13 είναι το μικρό αριθμητικό κόστος της φάσης ανανέωσης των προτεινόμενων λύσεων. Πιο συγκεκριμένα, παρότι οι πράξεις που εκτελούνται σε κάθε βήμα του αλγορίθμου είναι οι ίδιες με αυτές του Αλγορίθμου 12, τώρα εκτελούνται σε δεδομένα μικρότερου μεγέθους, $O(\delta)$. Επομένως, αν $\delta \ll n$ το κόστος του αλγορίθμου μπορεί να μειωθεί σημαντικά. Από την άλλη, επιλογή μικρής

τιμής ως είσοδο δ ενδέχεται να μειώσει το ρυθμό σύγκλισης της επίλυσης.

Πέραν των ταινιακών ΣΤΘΟ μητρώων, μπορεί να δειχθεί ότι παρόμοια συμπεριφορά απόσβεσης της απόλυτης τιμής των στοιχείων καθώς κινούμαστε μακριά από τη κύρια διαγώνιο βρίσκεται και στον αντίστροφο ενός ταινιακού, αυστηρά διαγώνια κυρίαρχου μητρώου A . Η απόσβεση εξαρτάται από τη νόρμα του A καθώς και το ημι-εύρος του [36].

3.1.2 Μέθοδοι πλοκάδας

Οι μέθοδοι πλοκάδας είναι η δεύτερη κατηγορία μεθόδων η οποία χρησιμοποιείται για τη ταχύτερη επίλυση ενός γραμμικού σύστηματος $AX = F$ με πολλά δεξιά μέλη. Σε αντίθεση με τις μεθόδους σποράς όπου η διαδικασία επίλυσης κάθε δεξιού μέλους είναι διακριτή από την επίλυση των υπόλοιπων, δηλαδή μόνο γραμμικά συστήματα με ένα δεξιά μέλος λύνονται σε κάθε χρονική στιγμή, οι μέθοδοι πλοκάδας δρουν συνολικά σε όλα τα δεξιά μέλη και κατά την ίδια χρονική στιγμή. Πριν σχολιάσουμε τα πλεονεκτήματα και μειονεκτήματα αυτής της προσέγγισης θα χρειαστεί να ορίσουμε - σε αντιστοιχία με τον ορισμό του υπόχωρου Krylov $K_m(A, r_0)$ - το πλοκαδικό υπόχωρο Krylov, $K_m^B(A, R_0)$.

Ορισμός 3.4. Εστω το γραμμικό σύστημα $AX = F$ με $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$ όπου με X_0 συμβολίζουμε την αρχική προσέγγιση της λύσης και με $R_0 = F - AX_0$ συμβολίζουμε το αρχικό κατάλοιπο. Θα ονομάζουμε πλοκαδικό Krylov υπόχωρο διάστασης m , τον υπόχωρο K_m^B που σχηματίζεται βάσει του

$$K_m^B(A, R_0) = \text{span}\{R_0, AR_0, \dots, A^{m-1}R_0\}.$$

Το πρώτο (και μεγαλύτερο) πλεονέκτημα των μεθόδων πλοκάδας είναι ότι η λύση $X = [x^{(1)}, \dots, x^{(s)}]$ προσεγγίζεται από έναν υπόχωρο μεγαλύτερης διάστασης. Οι επαναληπτικές μέθοδοι πλοκαδικών Krylov υπόχωρων, παράγουν στο βήμα i τη προσέγγιση της λύσης $x_i^{(j)}$, $x_i^{(j)}$, για την οποία ισχύει ότι

$$x_i^{(j)} \in x_0^{(j)} + K_i^B(A, R_0).$$

Σε αριθμητική άπειρης ακρίβειας, η διάσταση του K_m^B είναι s φορές μεγαλύτερη από αυτήν που θα είχαμε αν λύναμε ένα γραμμικό σύστημα με ένα μόνο δεξιό μέλος (υποθέτοντας ότι αρχικά όλα τα δεξιά μέλη είναι γραμμικά ανεξάρτητα). Ισχύει δηλαδή ότι

$$K_m^B(A, R_0) \subseteq K_m(A, r_0^{(1)}) + \dots + K_m(A, r_0^{(s)}).$$

Λόγω των γραμμικών εξαρτήσεων οι οποίες μπορούν να προκύψουν στη πλοκάδα των καταλοίπων (πχ. ταχύτερη σύγκλιση ορισμένων δεξιών μελών), πρακτικά ισχύει ότι ο πλοκαδικός υπόχωρος Krylov που σχηματίζεται είναι το πολύ s φορές μεγαλύτερος από αυτόν που θα σχηματίζόταν αν επιλύψαμε κάθε δεξιό μέλη ανεξάρτητα (υποθέτοντας τον ίδιο αριθμό επαναλήψεων για όλες τις περιπτώσεις). Επομένως, το πρώτο πλεονέκτημα των μεθόδων πλοκαδικών Krylov υπόχωρων είναι ο βελτιωμένος ρυθμός σύγκλισης των δεξιών μελών αφού πλέον η αντίστοιχη λύση τους αναζητείται σε μεγαλύτερους υπόχωρους. Το δεύτερο πλεονέκτημα των μεθόδων πλοκάδας, ιδιαίτερα χρήσιμο στις παραλληλες αρχιτεκτονικές, είναι ο μικρότερος αριθμός μεταφορών του μητρώου A από και προς τη κύρια μνήμη, αφού πλέον το μητρώο δρα πάνω σε μια πλοκάδα διανυσμάτων (ή πολυδιάνυσμα) για κάθε μεταφορά του προς τη κύρια μνήμη.

Στην αντίπερα όχθη, όπως θα δούμε και παραχάτω, τα παραπάνω πλεονεκτήματα θα πρέπει να “συνυπάρξουν” με την αριθμητική πεπερασμένης ακρίβειας του υπολογιστικού συστήματος, η επίδραση της οποίας είναι

περισσότερο εμφανής στις μεθόδους πλοκάδας και είναι το κύριο μειονέκτημα τους. Ένα ακόμα μειονέκτημα των μεθόδων πλοκάδας είναι οι μεγαλύτερες απαιτήσεις σε μνήμη αφού τώρα το πλήθος των διανυσμάτων που πρέπει να χρατούνται στη μνήμη πολλαπλασιάζεται επί το μέγεθος της πλοκάδας, s .

Αλγόριθμος 14 Ο αλγόριθμος Block Conjugate Gradient (BCG) [21]

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $X_0 \in \mathbb{R}^{n \times s}$, tol

Έξοδος: $X \in \mathbb{R}^{n \times s}$

$$R_0 = F - AX_0$$

$$P_0 = R_0$$

for $j = 1, \dots, s$ **do**

while $\|R_i\|/\|F\| \geq tol$ **do**

$$\alpha = (R_i^\top R_i)^{-1}(P_i^\top A P_i)$$

$$X_{i+1} = X_i + P_i \alpha$$

$$R_{i+1} = R_i - AP_i \alpha$$

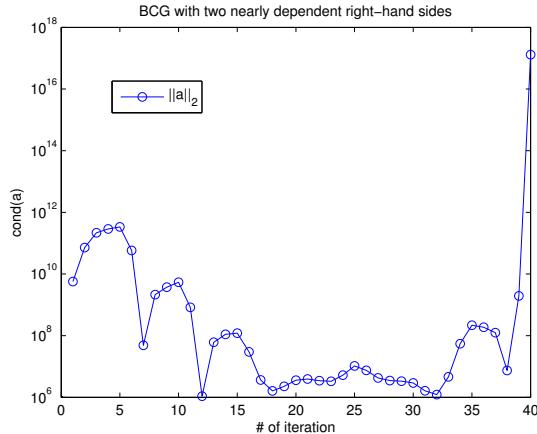
$$\beta = (R_{i+1}^\top R_{i+1})^{-1}(R_i^\top R_i)$$

$$P_{i+1} = R_{i+1} + P_i \beta$$

end while

end for

Ο Αλγόριθμος 14 απεικονίζει τον αλγόριθμο CG υλοποιημένο κατά πλοκάδες (στο εξής BCG). Ο αλγόριθμος BCG είναι παρόμοιος με τον αλγόριθμο CG με τη διαφορά ότι τώρα όλοι οι βαθμωτοί έχουν αντικατασταθεί από μητρώα μεγέθους $s \times s$. Είναι ακριβώς ο (πιθανός) μεγάλος δείκτης κατάστασης αυτών των μητρώων που εξηγεί τη μεγαλύτερη έκθεση των μεθόδων πλοκάδας ως προς το μοντέλο πεπερασμένης ακρίβειας.



Σχήμα 3.3: Δείκτης κατάστασης του μητρώου α κατα την εκτέλεση του αλγορίθμου 16.

Το Σχήμα 3.3 απεικονίζει ένα σύντομο αλλά ενδεικτικό παράδειγμα σχετικά με την αστάθεια των μεθόδων πλοκάδας στη περίπτωση που το μητρώο δεξιών μελών F περιέχει στήλες οι οποίες είναι σχεδόν γραμμικά εξαρτημένες. Σε αυτό το παράδειγμα επιλέξαμε $s = 2$ όπου το πρώτο δεξί μέλος ήταν ένα τυχαίο διάνυσμα ψευδοτυχαίας κατανομής ενώ το δεύτερο δεξί μέλος επιλέχθηκε ως το ίδιο διάνυσμα με τη διαφορά ότι τώρα προσθέσαμε ένα διάνυσμα "υιορύου" το οποίο είχε αρκετά μικρή νόρμα (της τάξης του 10^{-6}). Εκτελέσαμε 40 επαναλήψεις με τον Αλγόριθμο 14 και στο Σχήμα 3.3 παρουσιάζουμε το δείκτη κατάστασης του 2×2 μητρώου α . Όπως μπορούμε να δούμε, ο δείκτης κατάστασης κινείται σε αρκετά υψηλά επίπεδα μέχρι τη τελευταία επανάληψη όπου πλέον το μητρώο α γίνεται μη αντιστρέψιμο οπότε και ο Αλγόριθμος 14 αδυνατεί

να επιστρέψει τη λύση X . Το πρόβλημα για τη παραπάνω συμπεριφορά δεν εντοπίζεται στο μητρώο A , τα στοιχεία του οποίου είναι αδιάφορα στη προκειμένη περίπτωση - ο δείκτης κατάστασης του A ήταν πολύ μικρός. Το πρόβλημα αφορά τα δεξιά μέλη τα οποία ήταν σχεδόν συγγραμμικά. Αν ο θόρυβος που προστέθηκε στο δεύτερο διάνυσμα ήταν λίγο πιο έντονος (νόρμα περίπου ίση με 10^{-5}), ο Αλγόριθμος 14 θα επέστρεψε τις πραγματικές λύσεις - ως προς την ακρίβεια που είχε οριστεί.

Είναι ενδιαφέρον στο σημείο αυτό να παρατηρήσουμε ότι οι Αλγόριθμοι 12 και 14 συμπληρώνουν κατά μια έννοια ο ένας τον άλλον. Όταν τα δεξιά μέλη είναι περίπου παράλληλα, αυτό είναι το ιδανικό σενάριο για τον Αλγόριθμο 12 ενώ ταυτόχρονα είναι το χειρότερο δυνατό για τον Αλγόριθμο 14.

Θεώρημα 3.5. *Εστω ότι επιλύουμε το γραμμικό σύστημα $AX = F$ μέσω του αλγορίθμου BCG και έχουμε πραγματοποιήσει $m - 1$ βήματα (επαναλήψεις). Τότε η προσέγγιση της λύσης X , X_{m-1} , ικανοποιεί το ακόλουθο:*

$$\min_{X^* \in \mathbb{R}^{n \times s}} \text{tr}(\|X - X^*\|_A^2) = \text{tr}(\|X - X_{m-1}\|_A^2), \quad X_{m-1} \in X_0 + K_{m-1}^B \{A, R_0\}. \quad (3.1)$$

Απόδειξη. Βλέπε [21]. □

Το Θεώρημα 3.5 δηλώνει ότι η προσέγγιση της λύσης X μέσω του X_{m-1} που προκύπτει από τον Αλγόριθμο 14 ελαχιστοποιεί το άθροισμα των επιμέρους σφαλμάτων κάθε δεξιού μέλους ως προς την νόρμα $\|\cdot\|_A$ και για όλα τα $X^* \in \Delta_{s(m-1)}^n$, όπου $\Delta_{s(m-1)}^n$ συμβολίζει μια οποιαδήποτε συλλογή $s(m-1)$ διανυσμάτων διάστασης n με $n > s(m-1)$.

Διάφορες παραλλαγές των κλασικών μεθόδων πλοκάδας έχουν προταθεί προκειμένου να επιτευχθεί μια ισορροπία ανάμεσα στα πλεονεκτήματα και μειονεκτήματα τους. Μια από αυτές είναι η διαμέριση του μητρώου F σε w ανεξάρτητες ομάδες δεξιών μελών $F_B^{(1)}, \dots, F_B^{(w)}$ οι οποίες ιδιαίτερα περιέχουν τον ίδιο αριθμό από δεξιά μέλη (βλέπε και [39, 27, 41]). Η προσέγγιση αυτή μειώνει τις απαιτήσεις μνήμης των πλοκαδικών μεθόδων ταυτόχρονα όμως μειώνει και τον μέγιστο δυνατό ρυθμό σύγκλισης. Η μείωση αυτή του ρυθμού σύγκλισης ενδέχεται να είναι σημαντική ή και πολύ μικρή - με την απάντηση να εξαρτάται από τα χαρακτηριστικά του προβλήματος. Επίσης, με τη τεχνική της διαμέρισης των δεξιών μελών, συνήθως βελτιώνεται και η αριθμητική συμπεριφορά της μεθόδου με τα μητρώα α και β στον αλγόριθμο BCG να έχουν μικρότερη δείκτη κατάστασης κατά τη διάρκεια εκτέλεσης. Και πάλι ωστόσο η συμπεριφορά αυτή εξαρτάται από τα χαρακτηριστικά του προβλήματος αφού μπορούν να βρεθούν μητρώα A για τα οποία μέγεθος πλοκάδας ακόμα και ίσο με δύο, μπορεί να οδηγήσει σε αστάθεια.

Μια ακόμα τεχνική για τη βελτίωση της αριθμητικής συμπεριφοράς των πλοκαδικών μεθόδων βασίζεται στην αραίρεση δεξιών μελών για τα οποία η λύση έχει ήδη προσεγγιστεί με την επιμηκυνητή ακρίβεια. Πράγματι, υπάρχει η δυνατότητα για κάποια δεξιά μέλη να συγκλίνουν αρκετά πιο γρήγορα από τα άλλα και επομένως το κατάλοιπο τους να είναι κοντά στο μηδενικό διάνυσμα. Τότε, το μητρώο R_{i+1} τείνει να μην έχει πλήρη τάξη, κάτι το οποίο σημαίνει ότι το μητρώο $(R_{i+1}^\top R_{i+1})$ τείνει να είναι μη αντιστρέψιμο - ο αντίστροφος του όμως χρειάζεται για τον υπολογισμό του β και επομένως η μέθοδος θα τερματίσει χωρίς όλα τα δεξιά μέλη να έχουν συγκλίνει. Συνίσταται επομένως η αραίρεση των δεξιών μελών τα οποία έχουν κατάλοιπο με πολύ μικρή νόρμα ή είναι περίπου συγγραμμικά με κάποια άλλα δεξιά μέλη. Για περισσότερα πάνω σε αυτό το θέμα παραπέμπουμε στο [42].

Τέλος, αναφέρουμε ότι οι μέθοδοι σπόρου και πλοκάδας μπορούν να συνδυαστούν με διάφορους τρόπους. Ένας από αυτούς είναι η διαμέριση του μητρώου δεξιών μελών F σε ανεξάρτητες ομάδες όπου στη συνέχεια

οι μέθοδοι σπόρου εκτελούν τη διαδικασία που είδαμε και νωρίτερα όμως τώρα γενικεύουμε σε πλοκάδες (οι λύσεις κάθε διαμέρισης καθώς και οι προβολές των μη ακόμα επιλυμένων διαμερίσεων υλοποιούνται μέσω μεθόδων πλοκάδας), βλέπε και [40, 27].

3.2 Εφαρμογή των μεθόδων πολλών δεξιών μελών στην οικογένεια αλγορίθμων Spike

3.2.1 Περίπτωση I: Χαμηλή τάξη των μη-διαγώνιων πλοκάδων

Στη παρούσα ενότητα θα εξετάσουμε τη περίπτωση όπου οι μη-διαγώνιες πλοκάδες B_j, C_j (για τα j τα οποία ορίζονται) είναι τάξης ϕ και έχουμε ότι $\phi < m$ (ιδανικά $\phi \ll m$).

Αλγόριθμος 15 Επίλυση του γραμμικού συστήματος $KY = L$ με χρήση του αλγορίθμου PROJ

Είσοδος: $K \in \mathbb{R}^{n \times n}, L \in \mathbb{R}^{n \times s}, tol \in \mathbb{R}$

Έξοδος: $Y \in \mathbb{R}^{n \times s}$

1. $[Q, R] = QR(L)$ όπου $\text{rank}(L)=\phi$ με $1 \leq \phi \leq s$
 2. $\ell = \{i : l^{(i)} \text{ ανήκει στη βάση του } L\}$
 3. $Y_\ell = \text{SOLVE}(K, L_\ell, tol)$
 4. $D = L_\ell (L_\ell^\top L_\ell)^{-1} (L_\ell^\top L_{\notin \ell})$
 5. $Y_{\notin \ell} = Y_\ell D$
-

Ο Αλγόριθμος 15 εισάγει τον αλγόριθμο PROJ για την επίλυση του γραμμικού συστήματος $KY = L$ με $K \in \mathbb{R}^{n \times n}$ και $L \in \mathbb{R}^{n \times s}$. Ο αλγόριθμος PROJ είναι ιδιαίτερα αποδοτικός στη περίπτωση που η τάξη του L , ϕ , είναι αρκετά μικρότερη από τη μέγιστη (υποθέτουμε ότι $s \ll n$). Έτσι, ο αλγόριθμος επιλύει μόνο τα ϕ γραμμικά ανεξάρτητα δεξιά μέλη τα οποία σχηματίζουν μια τυχαία βάση (έστω τη πρώτη που βρίσκεται) του χώρου στηλών του L , και ύστερα συνθέτει τις λύσεις των γραμμικά εξαρτημένων δεξιών μελών συνδυάζοντας τις λύσεις των δεξιών μελών που απαρτίζουν τη βάση.

Η επίλυση των δεξιών μελών τα οποία απαρτίζουν την επιλεγμένη βάση μπορεί να γίνει με οποιονδήποτε αλγόριθμο επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη. Ωστόσο, στη πράξη επιλέγεται ένας αλγόριθμος που επιλύει το γραμμικό σύστημα είτε βάσει μεθόδων πλοκάδας είτε βάσει της κλασσικής προσέγγισης. Ο λόγος για αυτή την επιλογή είναι ότι τα διανύσματα τα οποία απαρτίζουν τη βάση δεν αναμένεται να έχουν κάποια εξάρτηση μεταξύ τους (συνήθως μάλιστα επιλέγονται ως ορθογώνια). Τέλος, για να βρούμε μια βάση του χώρου στηλών του L , πραγματοποιούμε κάποια παραγοντοποίηση η οποία μπορεί να αποκαλύψει μια βάση του L , πχ. τη παραγοντοποίηση QR ή τη παραγοντοποίηση SVD, τις οποίες είδαμε στο πρώτο κεφάλαιο.

Ο αλγόριθμος PROJ μπορεί άμεσα να εφαρμοστεί στους αλγορίθμους Spike, TR-Spike, PSpike για την εύρεση των spike-πλοκάδων V_j, W_j . Εφόσον οι αλγόριθμοι εφαρμόζονται σε ταινιακά μητρώα, όλα τα μη μηδενικά στοιχεία του μητρώου B_j (C_j) εντοπίζονται στις τελευταίες (πρώτες) m γραμμές του. Επομένως, η παραγοντοποίηση QR (ή SVD), αρκεί να πραγματοποιηθεί μόνο στο υπομητρώο μεγέθους $m \times m$ το οποίο περιέχει τα μη-μηδενικά στοιχεία και εισάγει σταθερό κόστος $O(m^3)$ ως προς τη διάσταση του προβλήματος.

Αλγόριθμος 16 Η μέθοδος Spike και η περίπτωση των χαμηλής τάξης μη διαγώνιων πλοκάδων (LR-SPIKE)

Είσοδος: $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times s}$, $p \in \mathbb{Z}$, $c \in \{0, 1\}$

Έξοδος: $X \in \mathbb{R}^{n \times s}$

```

{Κάθε επεξεργαστής  $j$ :}
if  $2 \leq j \leq p - 1$  then
    if  $c = 0$  then
         $[V_j, W_j] = \text{PROJ}(A_j, [\hat{B}_j, \hat{C}_j])$ 
    else
         $[V_j, W_j] = \text{SEED}(A_j, [B_j, C_j])$ 
    end if
    Υπολογισμός του  $G_j$ 
else if  $j = 1$  then
    if  $c = 0$  then
         $V_j = \text{PROJ}(A_j, \hat{B}_j)$ 
    else
         $V_j = \text{SEED}(A_j, B_j)$ 
    end if
    Υπολογισμός του  $G_j$ 
else
    if  $c = 0$  then
         $W_j = \text{PROJ}(A_j, \hat{C}_j)$ 
    else
         $W_j = \text{SEED}(A_j, C_j)$ 
    end if
    Υπολογισμός του  $G_j$ 
end if

```

Επίλυση του spike-συστήματος $S_r X_r = G_r$

Υπολογισμός των τελικών λύσεων

$$\begin{cases} X_1' = G_1' - V_1' X_2^{(q)} & j = 1 \\ X_j' = G_j' - V_j' X_{j+1}^{(q)} - W_j' X_{j-1}^{(1)} & 2 \leq j \leq p-1 \\ X_p' = G_p' - W_p' X_{p-1}^{(1)} & j = p \end{cases}$$

Στη περίπτωση που οι πλοκάδες B_j , C_j έχουν μη-μηδενικά στοιχεία τα οποία είναι διάσπαρτα (όπως στον αλγόριθμο WB-Spike), τυχόν παραγοντοποίηση τους ενδέχεται να εισάγει μεγάλη πολυπλοκότητα κόστους ή/και μνήμης. Τότε, η εναλλακτική που προτείνεται είναι να χρησιμοποιηθούν μέθοδοι σποράς οι οποίες εκμεταλλεύονται τη χαμηλή τάξη των μη-διαγώνιων πλοκάδων χωρίς να χρειάζεται να είναι εκ των προτέρων γνωστή κάποια βάση τους.

Ο Αλγόριθμος 16 απεικονίζει τη τροποποίηση της μεθόδου Spike για τη περίπτωση που οι πλοκάδες B_j , C_j έχουν τάξη $\phi \ll s$. Η τιμή της μεταβλητής $c \in \{0, 1\}$ ορίζει ποια εκ των μεθόδων SEED ή PROJ θα χρησιμοποιηθεί για τη δημιουργία των spike-πλοκάδων V_j , W_j . Αν το μητρώο A είναι ταινιακό τότε $c = 0$ και τα γραμμικά συστήματα λύνονται με τη μέθοδο PROJ. Άλλως, αν το μητρώο A θεωρείται ότι είναι γενικής, αραιής δομής (το ίδιο θεωρούμε και τα ταινιακά μητρώα μεγάλου εύρους) τα γραμμικά συστήματα λύνονται

με βάση τον αλγόριθμο 15.

3.2.2 Περίπτωση II: Γενικά ταινιακά μητρώα

Αν κοιτάζουμε περισσότερο προσεκτικά τις εξισώσεις $A_j V_j = \hat{B}_j$ και $A_j W_j = \hat{C}_j$, μπορούμε να παρατηρήσουμε ότι οι λύσεις δεν είναι κάτι παραπάνω από το πολλαπλασιασμό των $B_j \in \mathbb{R}^{m \times m}$, $C_j \in \mathbb{R}^{m \times m}$ με τις m τελευταίες και πρώτες στήλες του A_j^{-1} αντίστοιχα.

Αν εκφράσουμε το A_j ως ένα 2×2 πλοκαδικό μητρώο

$$A_j = \begin{bmatrix} A_j^{(1)} & A_j^{(2)} \\ A_j^{(3)} & A_j^{(4)} \end{bmatrix} \quad (3.2)$$

όπου $A_j^{(1)} \in \mathbb{R}^{(n_j-z) \times (n_j-z)}$, $A_j^{(2)} \in \mathbb{R}^{(n_j-z) \times z}$, $A_j^{(3)} \in \mathbb{R}^{z \times (n_j-z)}$, $A_j^{(4)} \in \mathbb{R}^{z \times z}$ και $z \geq m$, τότε το αντίστροφο μητρώο, $A_j^{(-1)}$, μπορεί να γραφτεί ως:

$$A_j^{-1} = \begin{bmatrix} * & -(A_j^{(1)} - A_j^{(2)}(A_j^{(4)})^{-1}A_j^{(3)})^{-1}A_j^{(2)}(A_j^{(4)})^{-1} \\ * & (A_j^{(4)})^{-1} + (A_j^{(4)})^{-1}A_j^{(3)}(A_j^{(1)} - A_j^{(2)}(A_j^{(4)})^{-1}A_j^{(3)})^{-1}A_j^{(2)}(A_j^{(4)})^{-1} \end{bmatrix}$$

όπου με $*$ συμβολίζουμε τις πρώτες $n_j - m$ στήλες του $A_j^{(-1)}$. Βάση αυτής της παρατήρησης, είναι εύκολο να διαπιστωθεί ότι για να υπολογιστούν οι πλοκάδες V_j , αρκεί να επιλύσουμε τις εξισώσεις:

$$A_j^{(4)} Y_j^{(1)} = \bar{B}_j \quad (3.3)$$

$$S_{A_j^{(4)}} Y_j^{(2)} = A_j^{(2)} Y_j^{(1)} \quad (3.4)$$

$$A_j^{(4)} Y_j^{(3)} = A_j^{(3)} Y_j^{(2)} \quad (3.5)$$

όπου $S_{A_j^{(4)}} = (A_j^{(1)} - A_j^{(2)}(A_j^{(4)})^{-1}A_j^{(3)})$ είναι το συμπλήρωμα Schur του υπομητρώου $A_j^{(4)}$ [30] και με \bar{B}_j συμβολίζεται η προς τα πάνω επέκταση του B_j με $z - m$ μηδενικές γραμμές.

Αντιστοίχως, το A_j μπορεί επίσης να εκφραστεί βάσει της εξίσωσης (3.2) με τα υπομητρώα αυτή τη φορά να ορίζονται ως $A_j^{(1)} \in \mathbb{R}^{z \times z}$, $A_j^{(2)} \in \mathbb{R}^{z \times (n_j-z)}$, $A_j^{(3)} \in \mathbb{R}^{(n_j-z) \times z}$, $A_j^{(4)} \in \mathbb{R}^{(n_j-z) \times (n_j-z)}$ και έτσι το μητρώο A_j^{-1} εκφράζεται ως:

$$A_j^{-1} = \begin{bmatrix} (A_j^{(1)})^{-1} + (A_j^{(1)})^{-1}A_j^{(2)}(A_j^{(4)} - A_j^{(3)}(A_j^{(1)})^{-1}A_j^{(2)})^{-1}A_j^{(3)}(A_j^{(1)})^{-1} & * \\ -(A_j^{(4)} - A_j^{(3)}(A_j^{(1)})^{-1}A_j^{(2)})^{-1}A_j^{(3)}(A_j^{(1)})^{-1} & * \end{bmatrix}$$

όπου με $*$ συμβολίζουμε τις τελευταίες $n_j - m$ στήλες. Έτσι, οι πλοκάδες W_j μπορούν να υπολογιστούν ως

$$A_j^{(1)} Y_j^{(1)} = \bar{C}_j \quad (3.6)$$

$$S_{A_j^{(1)}} Y_j^{(2)} = A_j^{(3)} Y_j^{(1)} \quad (3.7)$$

$$A_j^{(1)} Y_j^{(3)} = A_j^{(2)} Y_j^{(2)} \quad (3.8)$$

όπου $S_{A_j^{(1)}}(A_j^{(4)} - A_j^{(3)}(A_j^{(1)})^{-1}A_j^{(2)})$ είναι το συμπλήρωμα Schur του υπομητρώου $A_j^{(1)}$ και με \bar{C}_j συμβολίζεται η προς τα κάτω επέκταση του C_j με $z - m$ μηδενικές γραμμές. Στη συνέχεια θα επικεντρωθούμε στις

εξισώσεις (3.3) - (3.5) που απαιτούνται για τον υπολογισμό των πλοκάδων V_j και θα σχολιάζουμε όποτε η επέκταση για τον υπολογισμό των πλοκάδων W_j δεν είναι τετριμμένη.

Αν υποθέσουμε ότι $z \ll n_j$, τότε οι εξισώσεις (3.3), (3.5) λύνονται σχετικά γρήγορα αφού απαιτείται μόνον η παραγοντοποίηση του $A_j^{(4)}$. Επομένως, το κόστος υπολογισμού της πλοκάδας V_j εξαρτάται τώρα από αυτό του κόστους επίλυσης της εξίσωσης (3.4). Μια αρχική προσέγγιση θα μπορούσε να είναι ο υπολογισμός του συμπληρώματος Schur και η επίλυση του γραμμικού συστήματος μέσω κάποιας άμεσης μεθόδου, πχ. παραγοντοποίηση LU, όμως δεν υπάρχει λόγος αυτή η προσέγγιση να έχει χαμηλότερο υπολογιστικό κόστος αφού η διάσταση του $S_{A_j^{(4)}}$ είναι περίπου ίση με n_j . Από την άλλη, η επίλυση του (3.4) με χρήση επαναληπτικών μεθόδων δύναται να αυξήσει το συνολικό υπολογιστικό κόστος.

Η ιδέα είναι να παρατηρήσουμε ότι η λύση της εξίσωσης (3.4), η οποία -αν δε λάβουμε υπόψη το αρνητικό πρόσημο- αποτελεί τις πρώτες $n_j - z$ γραμμές της πλοκάδας V_j , ισούται με το πολλαπλασιασμό της πλοκάδας B_j με τις πρώτες $n_j - z$ γραμμές των τελευταίων m στηλών του A_j^{-1} . Αν θεωρήσουμε ότι η τάξη της πλοκάδας B_j είναι πλήρης, δηλαδή $\text{rank}(B_j) = m$, τότε όπως δείχνει και το Λήμμα ;;, η τάξη της λύσης του γραμμικού συστήματος της εξίσωσης (3.4) είναι ίση με τη τάξη του υπομητρώου του A_j^{-1} , $(A_j^{-1})_{1:n_j-m}^{n_j-m+1:n_j}$ όπου ο υποδείκτης δεικτοδοτεί τις γραμμές του A_j^{-1} και ο υπερδείκτης τις στήλες.

Η τάξη των μη-διαγώνιων πλοκάδων των αντιστρόφων των ταινιακών μητρώων είναι ένα θέμα το οποίο έχει εξεταστεί σε διάφορες εργασίες, [32, 33]. Επίσης, για τη περίπτωση ΣΘΟ, ταινιακών μητρώων, μπορεί να δειχτεί ότι το μέγεθος των στοιχείων του αντιστρόφου ακολουθεί μια αποσβένουσα συμπεριφορά.

Θεώρημα 3.6. Έστω ένα μητρώο $A \in \mathbb{R}^{n \times n}$ το οποίο έχει (ημι)-εύρος m . Τότε, για οποιαδήποτε μη-διαγώνια πλοκάδα του A^{-1} της οποίας τα στοιχεία βρίσκονται στις θέσεις όπου το A έχει μηδενικά στοιχεία, ισχύει ότι η τάξη της είναι το πολύ m .

Απόδειξη. Βλέπε [32] □

Ο Αλγόριθμος 17 απεικονίζει τον αλγόριθμο εύρεσης των πλοκάδων V_j , W_j μέσω της προσέγγισης που προτάθηκε στο παρόν υποκεφάλαιο και ονομάζεται SC-SPIKE. Με τον όρο LU, εννοούμε την επίλυση ενός γραμμικού συστήματος μέσω της παραγοντοποίησης LU. Καθώς η τιμή του z αυξάνεται, περιμένουμε να αυξάνεται και το κόστος επίλυσης με χρήση της παραγοντοποίησης LU. Από την άλλη, επίσης αναμένουμε ότι θα μειώνεται η τάξη της λύσης $Y_j^{(2)}$ της εξίσωσης (3.4).

Αλγόριθμος 17 Η μέθοδος SC-SPIKE

Input : $A_j \in \mathbb{R}^{n \times n}$, $B_j \in \mathbb{R}^{m \times m}$, $C_j \in \mathbb{R}^{m \times m}$, $z \in \mathbb{Z}$, $tol \in \mathbb{R}$
Output : $V_j \in \mathbb{R}^{n \times m}$, $W_j \in \mathbb{R}^{n \times m}$

{Κάθε επεξεργαστής j :

{Τηλογισμός V_j ($\alpha v j < p$):}

$$\begin{aligned} A_j^{(4)} Y_j^{(1)} &= \hat{B}_j \\ Y_j^{(2)} &= \text{PROJ}(A_j^{(1)} - A_j^{(2)}(A_j^{(4)})^{-1} A_j^{(3)}, A_j^{(2)} Y_j^{(1)}, tol) \\ A_j^{(4)} Y_j^{(3)} &= A_j^{(3)} Y_j^{(2)} \\ (V_j)_{n_j-z+1:n_j}^{1:m} &= Y_j^{(1)} + Y_j^{(3)} \\ (V_j)_{1:n_j-z}^{1:m} &= -Y_j^{(2)} \end{aligned}$$

{Τηλογισμός W_j ($\alpha v j > 1$):}

$$\begin{aligned} A_j^{(1)} Y_j^{(1)} &= \hat{C}_j \\ Y_j^{(2)} &= \text{PROJ}(A_j^{(4)} - A_j^{(3)}(A_j^{(1)})^{-1} A_j^{(2)}, A_j^{(3)} Y_j^{(1)}, tol) \\ A_j^{(1)} Y_j^{(3)} &= A_j^{(2)} Y_j^{(2)} \\ (W_j)_{1:z}^{1:m} &= Y_j^{(1)} + Y_j^{(3)} \\ (W_j)_{z+1:n_j}^{1:m} &= -Y_j^{(2)} \end{aligned}$$

Επίλυση του συστήματος (με επαναληπτική μέθοδο) $A_j G_j = F_j$

{Τώρα όλοι οι επεξεργαστές}

Επίλυση του συστήματος $S_r X_r = G_r$

Τηλογισμός των τελικών λύσεων

$$\left\{ \begin{array}{ll} X_1' = G_1' - V_1' X_2^{(q)} & j = 1 \\ X_j' = G_j' - V_j' X_{j+1}^{(q)} - W_j' X_{j-1}^{(1)} & 2 \leq j \leq p-1 \\ X_p' = G_p' - W_p' X_{p-1}^{(1)} & j = p \end{array} \right.$$

Κεφάλαιο 4

Πειράματα και αριθμητικά αποτελέσματα

Στο παρόν κεφάλαιο θα παρουσιάσουμε μερικά πειράματα και αριθμητικά αποτελέσματα από αλγορίθμους που παρουσιάσαμε στα προηγούμενα κεφάλαια.

Τα υπολογιστικά περιβάλλοντα στα οποία εκτελέσαμε τα αριθμητικά πειράματα είναι α) Το υπολογιστικό σύστημα PTOLEMY, το οποίο αποτελείται από 1 επεξεργαστή AMD Opteron(TM) 6274 με 16 πυρήνες στα 1.4 GHz¹. Το μέγεθος της μνήμης RAM είναι 1GB ανά πυρήνα (σύνολο 16GB), και β) Το υπολογιστικό σύστημα SLOTNICK, το οποίο αποτελείται από 2 επεξεργαστές Inter(R) Xeon(R) με συνολικά 12 πυρήνες στα 2.8 GHz². Το μέγεθος της μνήμης RAM είναι 2GB ανά πυρήνα (σύνολο 32GB). Στο σημείο αυτό οφείλουμε να ευχαριστήσουμε το τους συνεργάτες μας, το Καθηγητή του Πανεπιστημίου Purdue, Dr. Ahmed Sameh όπως και τον ερευνητή Faisal Saied για την άδεια τους και την υποστήριξή τους στην εκτέλεση αριθμητικών πειραμάτων στο υπολογιστικό σύστημα SLOTNICK. Ο Δρ. Γεώργιος Κόλλιας επίσης βοήθησε τα μέγιστα στην εγκατάσταση και υποστήριξη του απαραίτητου λογισμικού.

Όλοι οι χρόνοι που αναγράφονται αναφέρονται πάντοτε σε δευτερόλεπτα.

4.1 Επίλυση ταινιακών γραμμικών συστημάτων

Στη παρούσα ενότητα θα ασχοληθούμε με ταινιακά μητρώα και θα επικεντρωθούμε στους αλγόριθμους SC-SPIKE και Spike.

4.1.1 Τεχνητά μητρώα

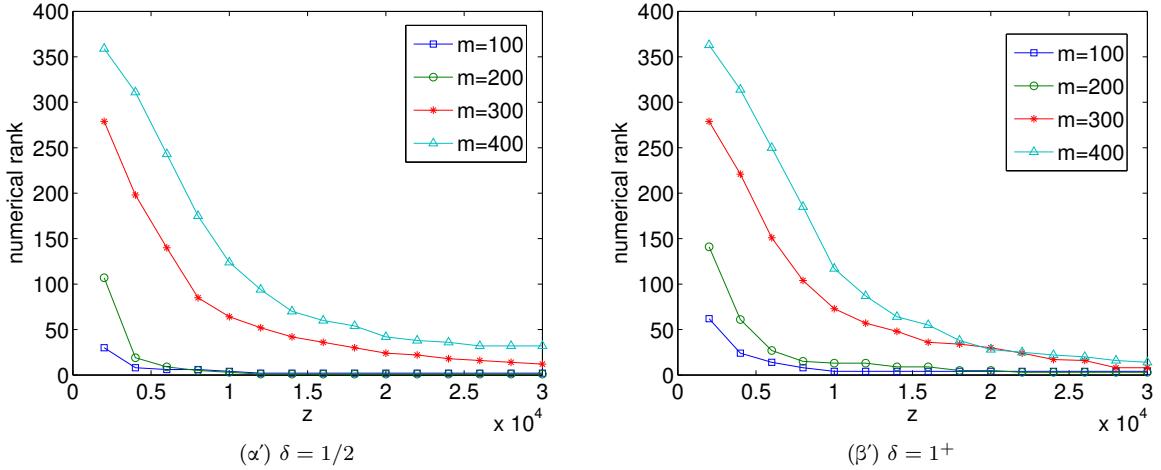
Σε αυτή την υποενότητα θα ασχοληθούμε με τεχνητά μητρώα τα οποία κατασκευάστηκαν βάσει προκαθορισμένων τιμών. Αρχικά, θα παραθέσουμε κάποια παραδείγματα όπου τα μητρώα συντελεστών έχουν συγκεκριμένη δομή, πιο συγκεκριμένα είναι συμμετρικά και Toeplitz.

¹<http://products.amd.com/en-us/OpteronCPUResult.aspx?f1=AMD+Opteron>

²<http://ark.intel.com/products/47921/>

Ορισμός 4.1. Ενα μητρώο $A \in \mathbb{R}^{n \times n}$ θα ονομάζεται μητρώο Toeplitz όταν τα στοιχεία του ακολουθούν την ιδιότητα $a_{i,j} = a_{i+1,j+1}, i, j = 1, \dots, n - 1$.

Για τις ανάγκες των πειραμάτων, δημιουργήσαμε μητρώα με μεταβλητό ημι-εύρος z και μεταβλητό βαθμό διαγώνιας κυριαρχίας. Η διάσταση του μητρώου συντελεστών επιλέχθηκε ως $n = 4 \times 10^5$ και ο αριθμός των επεξεργαστών ίσος με $p = 2$. Σκοπός μας είναι να εξετάσουμε την αριθμητική τάξη των πλοκάδων $(V_j)_{1:n_j-z}^{1:m}$ (όταν ορίζονται) έτσι ώστε να αναδειχτεί η χρησιμότητα της μεθόδου SC-SPIKE. Για λόγους ευκολίας στη παρουσίαση και κατανόηση των αποτελεσμάτων, θα παραθέσουμε τα αποτελέσματα τα οποία είναι σχετικά μόνο με το υπομητρώο $(V_1)_{1:n_j-z}^{1:m}$.



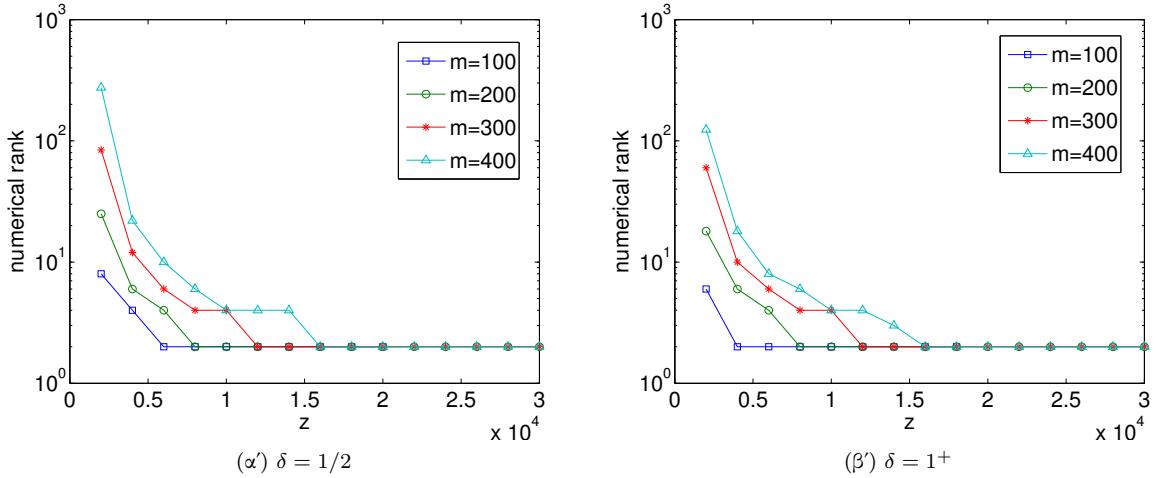
Σχήμα 4.1: Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ για ένα μητρώο Toeplitz με μεταβλητό ημι-εύρος και βαθμό διαγώνιας κυριαρχίας.

Το Σχήμα 4.1 απεικονίζει την αριθμητική τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$, συναρτήσει του z και για διαφορετικό ημι-εύρος και βαθμό διαγώνιας κυριαρχίας. Τα μη-διαγώνια στοιχεία του A επελέγησαν ως ψευδοτυχαίες τιμές από τη κανονική κατανομή³ ενώ τα διαγώνια στοιχεία προέκυψαν από το άνθροισμα των απολύτων τιμών των μη-διαγώνιων στοιχείων κάθε γραμμής το οποίο στη συνέχεια πολλαπλασιάστηκε με τον επιμψητό βαθμό διαγώνιας κυριαρχίας, δ . Προκειμένου να μειώσουμε τις πιθανότητες τυχόν θετικής ορισμότητας, κάθισε 100στό διαγώνιο στοιχείο επιλέχθηκε να έχει αρνητικό πρόσημο.

Όπως ήταν αναμενόμενο, μεγαλύτερο ημι-εύρος του μητρώου συντελεστών οδηγεί σε μεγαλύτερη τάξη για τα υπομητρώα $(V_1)_{1:n_j-z}^{1:m}$ (θεωρούμε ότι η διάσταση n είναι σταθερή). Επίσης, όπως μπορεί να φανεί από το Σχήμα 4.1, για μεγαλύτερες τιμές του z , η τάξη μειώνεται και έτσι περιμένουμε καλύτερη επίδοση από τους αλγορίθμους PROJ, SEED.

Το Σχήμα 4.2 απεικονίζει την ίδια πληροφορία με το Σχήμα 4.1 όμως τώρα τα στοιχεία του μητρώου A επελέγησαν διαφορετικά. Πιο συγκεκριμένα, το a_{ij} στοιχείο επιλέχθηκε ως $a_{ij} = 1/|i - j|$ αν $i = 2g$ όπου g είναι θετικός ακέραιος, αλλιώς $a_{ij} = -1/|i - j|$. Για τα διαγώνια στοιχεία ισχύει ότι και παραπάνω. Πάλι μπορεί να διαπιστωθεί ότι καθώς το z αυξάνεται, η τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ μειώνεται. Λόγω της αποσβένουσας συμπεριφοράς των στοιχείων του A^{-1} , για $z \geq 15000$ αρκεί να λύσουμε μόνο ένα γραμμικό σύστημα (με ένα δεξί μέλος).

³Σε συμβολισμό MATLAB: $a(i, j) = \text{randn}(1)$, $i \neq j$



Σχήμα 4.2: Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ για ένα μητρώο Toeplitz με μεταβλητό ημι-εύρος και βαθμό διαχώνιας κυριαρχίας.

4.1.2 Μητρώα από τη συλλογή University of Florida sparse matrix collection

Σε αυτή την υποενότητα παραθέτουμε πειράματα όπου τα μητρώα συντελεστών προέρχονται από διάφορες εφαρμογές. Τα μητρώα βρίσκονται στη συλλογή αραιών μητρώων του Πανεπιστημίου της Florida⁴ [34]. Όπως και νωρίτερα, σκοπός μας είναι να εξετάσουμε την αριθμητική τάξη των πλοκάδων $(V_j)_{1:n_j-z}^{1:m}$ (όταν ορίζονται) έτσι ώστε να αναδειχτεί η χρησιμότητα της μεθόδου SC-SPIKE. Για λόγους ευκολίας στη παρουσίαση και κατανόηση των αποτελεσμάτων, θα παραθέσουμε τα αποτελέσματα τα οποία είναι σχετικά μόνο με το υπομητρώο $(V_1)_{1:n_j-z}^{1:m}$.

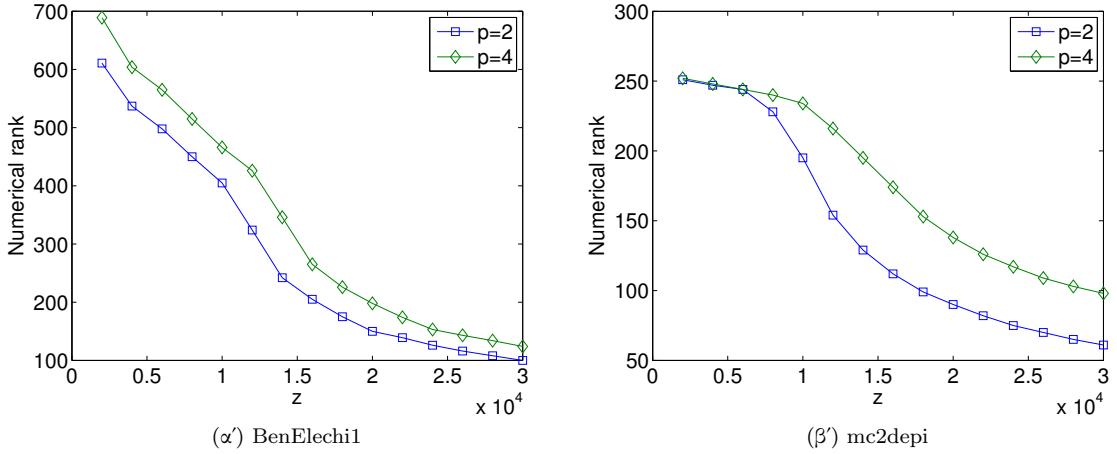
Πίνακας 4.1: Χαρακτηριστικά των μητρώων από τη συλλογή [34]

A	n	$\text{nnz}(A)$	ημι-εύρος	Εφαρμογή
1. BenElechi1	245,874	13,150,496	821	2D/3D
2. shipsec5	259,156	4,429,042	3923	2D/3D
3. kim2	456,976	11,330,020	1354	2D/3D
4. mc2depi	525,825	2,100,225	513	2D/3D
5. boneS01	127,224	5,516,602	3722	Μείωση μοντέλου
6. af_5_k101	503,625	17,550,675	859	Δομικό πρόβλημα

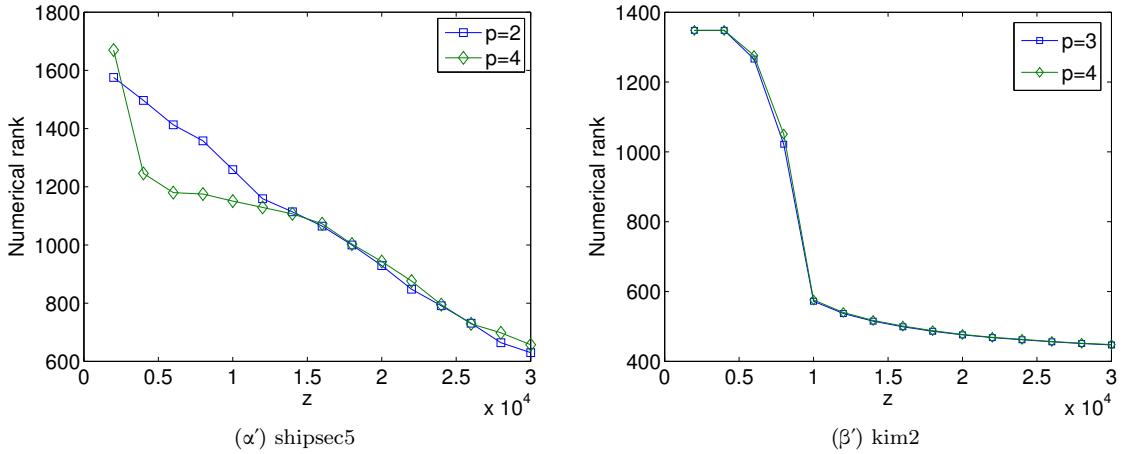
Τα Σχήματα 4.3 - 4.5 απεικονίζουν την αριθμητική τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$, συναρτήσει του z και για διαφορετικό αριθμό διαχωρισμάτων (επεξεργαστών). Το γενικό συμπέρασμα που προκύπτει από τα παραπάνω σχήματα είναι ότι καθώς αυξάνεται ο αριθμός των διαχωρισμάτων (δηλαδή αυξάνεται ο αριθμός των επεξεργαστών και επομένως κάθε επεξεργαστής χειρίζεται μητρώα μικρότερου μεγέθους) η αριθμητική τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ μειώνεται με χαμηλότερο ρυθμό ως πρός το z .

Όσον αφορά μητρώα τα οποία είναι μεγαλύτερης διάστασης και το εύρος τους είναι σχετικά χαμηλό, η τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ μειώνεται με ένα αρκετά πιο γρήγορο ρυθμό. Για παράδειγμα, θεωρώντας ένα

⁴<http://www.cise.ufl.edu/research/sparse/matrices/>



Σχήμα 4.3: Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ ($p = 2, 4$) για τα μητρώα BenElechi1 και mc2depi.

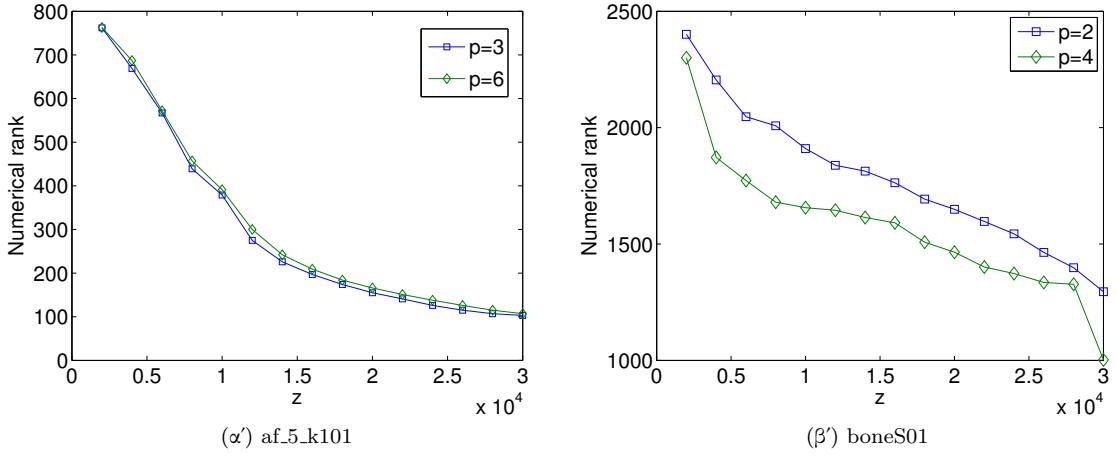


Σχήμα 4.4: Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ ($p = 2, 4$) για τα μητρώα shipsec5 και kim2.

συγκεκριμένο z , μητρώα τα οποία έχουν μικρό σχετικό εύρος ως προς της διάσταση τους, επιστρέφουν μικρότερη τάξη για το υπομητρώο $(V_1)_{1:n_j-z}^{1:m}$.

Σε αυτή την υποενότητα επίσης παραθέτουμε πειράματα για την επίλυση γραμμικών συστημάτων $AX = F$ με ταινιακά μητρώα συντελεστών μέσω του αλγορίθμου SPIKE. Υλοποιήσαμε τον αλγόριθμο Spike ώστε οι spike-πλοκάδες να υπολογίζονται μέσω ενός επαναληπτικού σχήματος. Σκοπός μας είναι να δείξουμε ότι -υπό συνθήκες- ο άμεσος επιλυτής μπορεί να αντικατασταθεί από ένα επαναληπτικό σχήμα το οποίο συνδυάζει τεχνικές προκατάστασης και μεθόδους επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη. Θα επικεντρωθούμε χυρίως στη πρώτη φάση επίλυσης του αλγορίθμου Spike καθώς η δεύτερη φάση είναι ανεξάρτητη του επιλυτή που χρησιμοποιούμε για να λύσουμε τα γραμμικά συστήματα με το μητρώο A_j , $j = 1, \dots, p$.

Τα πειράματα εκτελέστηκαν σε $p = 2, 4$ και 8 πυρήνες (διαχωρίσματα) και τα μητρώα επιλέχθηκαν και πάλι



Σχήμα 4.5: Τάξη του υπομητρώου $(V_1)_{1:n_j-z}^{1:m}$ ($p = 2, 4$) για τα μητρώα af5k101 και boneS01.

από τη συλλογή [34]. Τα χαρακτηριστικά των μητρώων απεικονίζονται στο Πίνακα 4.2 όπου παραθέτουμε τη διάσταση n , το ημι-εύρος m , το μέσο πλήθος μη-μηδενικών στοιχείων ανά γραμμή $\text{nnz}(A)/n$ καθώς και το δείκτη κατάστασης για κάθε μητρώο. Όλα τα μητρώα σε αυτή την υποενότητα είναι ΣΘΟ και επομένως το επαναληπτικό σχήμα βασίζεται στον αλγόριθμο CG και τις παραλλαγές του. Η ακρίβεια επίλυσης για τον επαναληπτικό επιλυτή ήταν πάντα $\text{tol} = 10^{-8}$ (για το σχετικό κατάλοιπο).

Πίνακας 4.2: Χαρακτηριστικά των μητρώων από τη συλλογή [34]

	n	m	$\text{nnz}(A)/n$	$\kappa(A)$	Εφαρμογή
1. bcsstk16	4884	140	59.4	$O(10^9)$	Δομικό πρόβλημα
2. fv1	9604	99	8.8	$O(10)$	2D/3D
3. t2dah_e	11445	231	15.3	$O(10^9)$	Μείωση μοντέλου
4. crystm02	13965	459	11.6	$O(10^2)$	Υλικά
5. wathen100	30401	304	7.8	$O(10^4)$	2D/3D
6. wathen120	36441	304	7.8	$O(10^3)$	2D/3D
7. minsurfo	40806	202	2.5	$O(10^2)$	Πρόβλημα βελτιστοποίησης
8. jnlbrng1	40000	200	2.5	$O(10^2)$	Πρόβλημα βελτιστοποίησης
9. apache1	80800	808	3.3	$O(10^6)$	Δομικό πρόβλημα

Ως άμεσο επιλυτή επιλέξαμε τη μέθοδο παραγοντοποίησης Cholesky. Από την άλλη, το επαναληπτικό σχήμα βασίζεται σε δύο διαφορετικές μεθόδους - τα γραμμικά συστήματα για την εύρεση των spike-πλοκάδων, $A_j[V_j, W_j] = [\hat{B}_j, \hat{C}_j]$ επιλύονται μέσω της μεθόδου CG (με η χωρίς προκατάσταση) ενώ το γραμμικό σύστημα $A_j G_j = F_j$ επιλύεται μέσω της μεθόδου BCG (με η χωρίς προκατάσταση). Όταν δε χρησιμοποιούμε προκατάσταση θα ονομάζουμε το σχήμα ως Αλγόριθμος1 ενώ όταν χρησιμοποιούμε προκατάσταση θα ονομάζουμε το σχήμα ως Αλγόριθμος2. Ως προκαταστάτη χρησιμοποιήσαμε την ατελή παραγοντοποίηση Cholesky διατηρώντας τη μη-μηδενική δομή του A , $\text{IC}(0)$.

Ο Πίνακας 4.3 παραθέτει το πλήθος των μη-μηδενικών στοιχείων που περιέχονται στο κάτω τριγωνικό μητρώο L που επιστρέφεται από τη πλήρη και μη-πλήρη παραγοντοποίηση Cholesky και για τις διάφορες τιμές διαχωρισμάτων. Όπως μπορεί εύκολα να διαπιστωθεί οι απαιτήσεις μνήμης της πλήρους παραγοντοποίησης είναι αισθητά μεγαλύτερες από αυτές της ατελούς παραγοντοποίησης.

Πίνακας 4.3: Πλήθος μη-μηδενικών τιμών του παράγοντα L στη πλήρη και μη-πλήρη παραγοντοποίηση.

	$p = 2$		$p = 4$		$p = 8$	
	chol	ichol	chol	ichol	chol	ichol
bcsstk16	300392	72676	142124	34495	62631	15376
fv1	470547	23571	230471	11639	110383	5670
t2dah_e	709932	46665	353615	231111	162464	11316
crystm02	$O(10^6)$	82559	474608	39787	210149	18160
wathen100	$O(10^6)$	124843	$O(10^6)$	61768	$O(10^6)$	30337
wathen120	$O(10^6)$	149893	$O(10^6)$	74293	$O(10^6)$	36495
minsurfo	$O(10^6)$	61007	$O(10^6)$	30351	994697	15072
jnlbrng1	$O(10^6)$	59700	$O(10^6)$	29750	965199	14775
apache1	$O(10^6)$	155342	$O(10^6)$	77267	$O(10^6)$	38179

Οι Πίνακες 4.4-4.5 παραθέτουν τους χρόνους (σε δευτερόλεπτα) που απαιτήθηκαν για την επίλυση της πρώτης φάσης της μεθόδου Spike καθώς και αυτό της επίλυσης $A_j G_j = F_j$. Όπως μπορούμε να διαπιστώσουμε το επαναληπτικό σχήμα (Αλγόριθμος1 και Αλγόριθμος2) είναι ελκυστικό στη περίπτωση που ο δείκτης κατάστασης του A είναι χαμηλός (ο δείκτης κατάστασης κάθε υπομητρώου A_j φράσσεται από αυτόν του A). Ο Πίνακας 4.4 παραθέτει τους χρόνους περάτωσης για τη περίπτωση $s = 10$ (το μητρώο F περιέχει δέκα στήλες) ενώ ο Πίνακας 4.5 τους αντίστοιχους για $s = 20$ (το μητρώο F τώρα περιέχει είκοσι στήλες). Σε κάθε περίπτωση οι στήλες του F επιλέχθηκαν ως τυχαία διανύσματα - κάθε στοιχείο του F επιλέχθηκε από την ομοιόμορφη κατανομή και στο διάστημα $(0, 1)$.

Πίνακας 4.4: Χρόνοι εκτέλεσης του αλγορίθμου SPIKE για διαφορετικό πλήθος διαχωρισμάτων ($s = 10$).

	$p = 2$			$p = 4$			$p = 8$		
	Cholesky	Αλγ1	Αλγ2	Cholesky	Αλγ1	Αλγ2	Cholesky	Αλγ1	Αλγ2
bcsstk16	0.9	5.1	2.4	0.4	2.5	0.9	0.2	0.7	0.3
fv1	2.8	0.5	0.4	1.3	0.3	0.2	0.5	0.2	0.1
t2dah_e	2.7	7.5	2.1	1.0	3.1	1.2	0.5	1.9	0.6
crystm02	54.7	7.7	0.9	23.8	5.8	0.5	9.6	3.1	0.2
wathen100	12.7	16.7	5.3	6.5	7.3	2.4	2.9	3.7	1.3
wathen120	15.7	21.7	8.2	6.5	6.9	2.9	3.4	5.3	1.7
minsurfo	16.0	10.2	7.7	6.4	2.0	1.0	4.3	1.2	0.7
jnlbrng1	12.8	11.0	6.2	5.8	7.9	4.4	4.2	4.6	2.2
apache1	204.2	F	644.9	117.1	F	278.5	53.6	F	105.2

Τέλος, ο Πίνακας 4.6 απεικονίζει το συνολικό πλήθος (αραιών) MV γινομένων τα οποία χρειάστηκαν για να επιτευχθεί η επιψυμητή ακρίβεια κατά την επίλυση των γραμμικών συστημάτων $A_j[V_j, W_j, G_j] = [\bar{B}_j, \bar{C}_j, F_j]$ για τη περίπτωση $s = 10$. Λόγω της ταχύτερης σύγκλισης μέσω προκατάστασης, το πλήθος των MV γινομένων είναι πολύ μικρότερο από ότι χωρίς χρήση προκαταστάσης. Βέβαια, για να είμαστε απολύτως δίκαιοι, θα έπρεπε να δείξουμε και το κόστος εφαρμογής του προκαταστάτη για κάθε επανάληψη, ωστόσο τα αποτελέσματα δεν είχαν κάποια ενδιαφέρουσα παράμετρο - η εφαρμογή του προκαταστάτη δεν ήταν ακριβή σε όρους χρόνου όπως μπορεί να φανεί και από τους Πίνακες 4.4 - 4.5.

Πίνακας 4.5: Χρόνοι εκτέλεσης της μεθόδου Spike για διαφορετικό πλήρως διαχωρισμάτων ($s = 20$).

	$p = 2$			$p = 4$			$p = 8$		
	Cholesky	Aλγ1	Aλγ2	Cholesky	Aλγ1	Aλγ2	Cholesky	Aλγ1	Aλγ2
bcsstk16	0.9	7.0	2.4	0.4	4.0	0.9	0.2	0.7	0.4
fv1	2.4	0.8	0.4	1.0	0.5	0.2	0.5	0.2	0.1
t2dah_e	2.8	10.1	2.4	1.1	4.8	1.2	0.6	3.4	0.6
crystm03	54.3	8.4	1.0	23.4	6.5	0.6	9.6	3.6	0.3
wathen100	12.6	18.8	7.1	9.4	8.9	2.5	3.9	3.9	1.5
wathen120	19.5	24.6	8.5	8.2	6.9	2.9	3.5	6.2	1.8
minsurfo	16.5	16.5	7.7	9.7	4.3	1.2	4.3	1.5	0.8
jnlbrng1	19.3	18.4	6.3	10.7	7.6	3.6	4.5	4.2	2.2
apache1	280.6	F	689.0	148.7	F	275.9	46.6	F	103.6

Πίνακας 4.6: Πλήρως MV γινομένων με ή χωρίς προκατάσταση.

	$p = 2$		$p = 4$		$p = 8$	
	$M = \text{speye}()$	$M = \text{ichol}()$	$M = \text{speye}()$	$M = \text{ichol}()$	$M = \text{speye}()$	$M = \text{ichol}()$
bcsstk16	22600	3900	20500	2900	14480	2330
fv1	2626	872	2804	872	14480	862
t2dah_e	26100	3173	15243	2932	20325	3163
crystm02	23950	938	33878	938	29567	938
wathen100	1736	2836	21832	2836	21386	2836
wathen120	19092	2836	18608	2836	24832	3140
minsurfo	15566	4280	6086	1120	6996	1696
jnlbrng1	19790	3840	21910	4630	21000	4420
apache1	F	114160	F	108802	F	81170

4.2 Επίλυση γενικών, αραιών γραμμικών συστημάτων

Στη παρούσα ενότητα θα εξετάσουμε το κατά πόσο η μεθόδος Spike μπορεί να συνδυαστεί με τεχνικές επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη για την επίλυση του συστήματος $AX = F$ με A ένα αραιό μητρώο - όχι απαραίτητα ταινιακό. Πιο συγκεκριμένα θα εξετάσουμε το αν οι μεθόδοι σποράς μπορούν να έχουν κάποια χρησιμότητα. Η προσέγγιση που ακολουθούμε βασίζεται στον Αλγόριθμο 8.

Αρχικά το μητρώο A αναδιατάσσεται συμμετρικά μέσω των μεθόδων αναδιάταξης που είδαμε στο προηγούμενο κεφάλαιο. Τστερα εξάγεται ένας ταινιακός προκαταστάτης M ημι-εύρους \hat{m}_1 (\hat{m}_2) ώστε $\|M\|_F \geq 0.95 * \|A\|_F$ ($\|M\|_F \geq 0.99 * \|A\|_F$), δηλαδή ο προκαταστάτης M να περιέχει τουλάχιστον το 95% (99%) της Frobenius νόρμας του A . Παρακάτω, όταν θα αναφέρουμε τη λέξη νόρμα θα εννοούμε πάντοτε τη νόρμα Frobenius.

Ο Πίνακας 4.7 παραθέτει τα μητρώα με τα οποία θα ασχοληθούμε στη παρούσα ενότητα. Τα μητρώα επιλέχθηκαν να είναι συμμετρικά (όχι όμως απαραίτητα ΣΘΟ). Στο Πίνακα 4.8 παραθέτουμε το ημι-εύρος m ύστερα από τη συμμετρική αναδιάταξη κάθε μητρώου και για τις τρεις διαφορετικές αναδιατάξεις (RCM, Fiedler, Weighted Spectral Reordering) που είδαμε στο προηγούμενο κεφάλαιο. Παραθέτουμε επίσης το απαιτούμενο ημι-εύρος του προκαταστάτη M ώστε αυτός να καταλαμβάνει το 95% και 99% της νόρμας του αναδιαταγμένου μητρώου A . Τα ημι-εύρη αυτά συμβολίζονται ως \hat{m}_1 και \hat{m}_2 αντίστοιχα.

Αρχικά, όπως είναι προφανές, ισχύει ότι $\hat{m}_2 \geq \hat{m}_1$ για κάθε μητρώο και για κάθε αναδιάταξη αφού για να

Πίνακας 4.7: Χαρακτηριστικά των μητρώων από τη συλλογή [34]

	<i>n</i>	Εφαρμογή
1. bcsstk28	4410	Δομικό πρόβλημα
2. eurqsa	7245	Δομικό πρόβλημα
3. OPF_3754	15735	Πρόβλημα δικτύου ηλεκτρικής ενέργειας 2D/3D
4. dubcova1	16129	
5. raefsky4	19779	Δομικό πρόβλημα
6. rail_20209	20209	Πρόβλημα μείωσης μοντέλου
7. g7jac180sc	53370	Οικονομικό πρόβλημα

Πίνακας 4.8: Ημι-εύρος m και απαιτούμενα ημι-εύρη \hat{m}_1 , \hat{m}_2 ώστε $\|M\|_F \geq 0.95 * \|A\|_F$ και $\|M\|_F \geq 0.99 * \|A\|_F$ ύστερα από τις συμμετρικές αναδιατάξεις.

	RCM				Fiedler				WSO			
	<i>m</i>	\hat{m}_1	\hat{m}_2	(sec)	<i>m</i>	\hat{m}_1	\hat{m}_2	(sec)	<i>m</i>	\hat{m}_1	\hat{m}_2	(sec)
bcsstk28	406	271	274	-	815	224	344	0.4	993	10	17	0.7
eurqsa	438	357	383	-	1568	335	978	2.3	1733	77	321	3.5
OPF_3754	800	550	729	-	3240	74	393	44.9	5289	7	28	31.5
dubcova1	500	108	446	-	653	97	245	4.9	691	95	236	3.1
raefsky4	1052	539	823	-	5918	911	1251	5.4	5105	12	24	15.5
rail_20209	276	30	221	-	610	40	132	34.7	579	39	130	9.7
g7jac180sc	10976	2812	3065	-	20859	1917	1950	32.1	26554	9	10	39.3

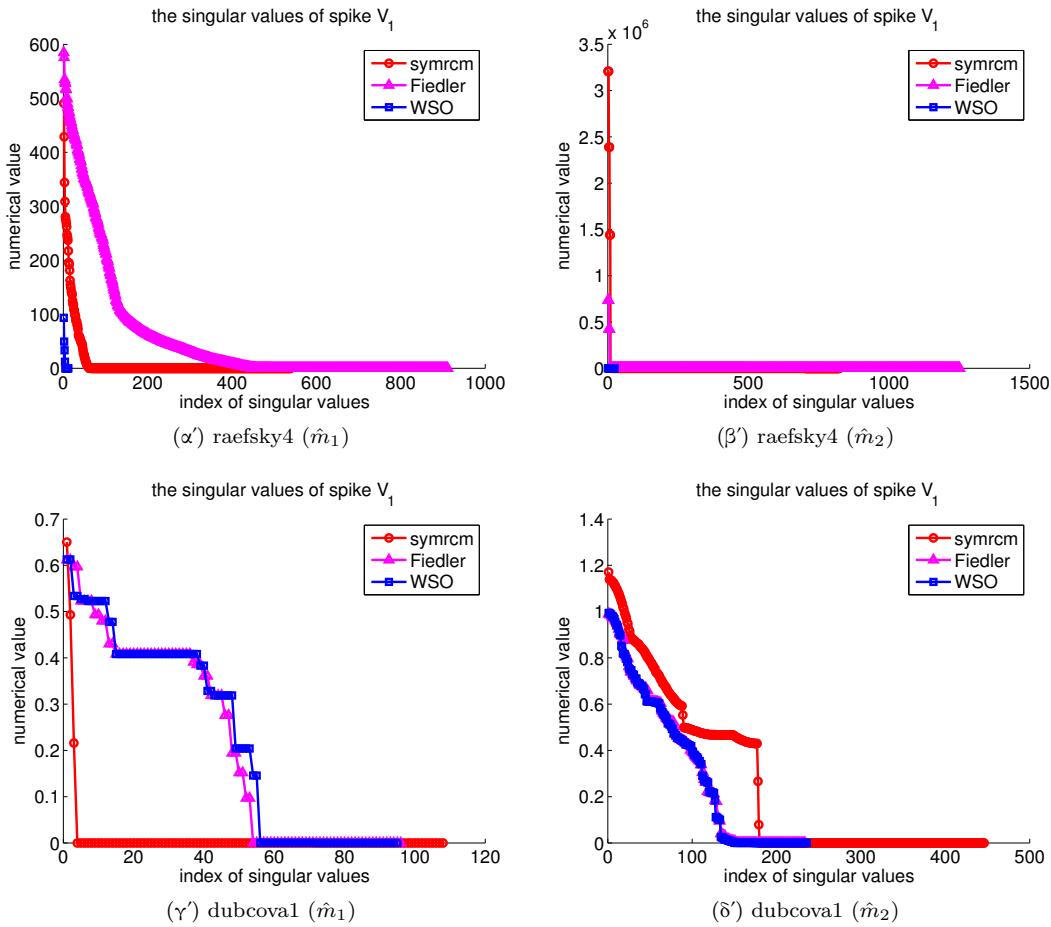
καταλαμβάνει ο προκαταστάτης M το 99% της νόρμας του αναδιαταγμένου μητρώου θα πρέπει οπωσδήποτε να καταλαμβάνει και το 95%. Μπορούμε να δούμε στο Πίνακα 4.8 ότι η σταθμισμένη φασματική αναδιάταξη (WSO) επιτυγχάνει τη μεγαλύτερη μείωση του αναγκαίου ημι-εύρους για να περιέχει ο προκαταστάτης M το ζητούμενο ποσοστό νόρμας του A . Το αποτέλεσμα αυτό ήταν αναμενόμενο αφού η σταθμισμένη φασματική αναδιάταξη διαφέρει σε ακριβώς αυτό το σημείο από τις άλλες δύο μεθόδους αναδιάταξης. Επίσης, από τα πειράματα διαπιστώνεται ότι η σταθμισμένη φασματική αναδιάταξη είναι η μέθοδος που μειώνει λιγότερο αποδοτικά το γενικότερο ημι-εύρος m . Σχετικά με τις μεθόδους RCM και φασματικής αναδιάταξης (Fiedler) δεν έχουμε κάποια καθαρό νικητή αφού η απόδοσή τους εξαρτάται από το μητρώο A . Μπορούμε ωστόσο να δούμε ότι η μέθοδος RCM μειώνει καλύτερα το γενικότερο εύρος m στο αναδιαταγμένο μητρώο συντελεστών.

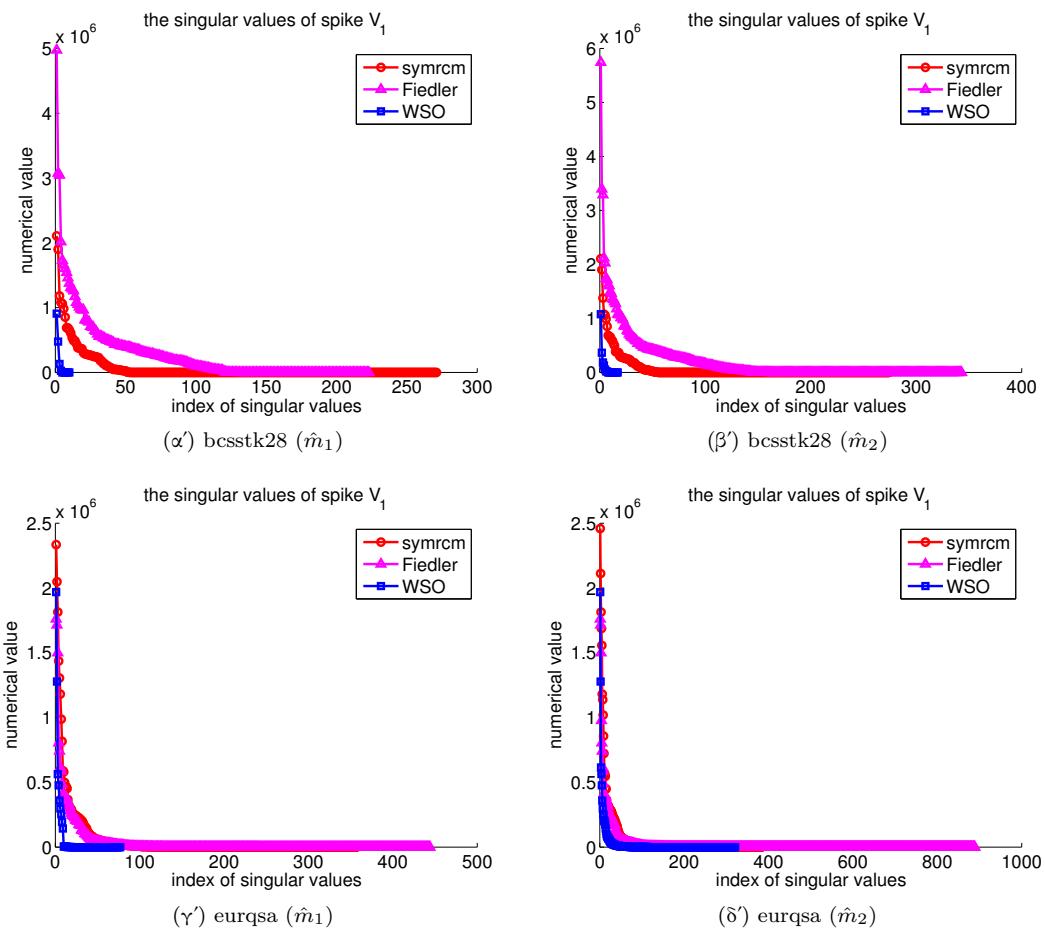
Τώρα που διαπιστώσαμε ότι η αναδιάταξη του μητρώου A μπορεί να οδηγήσει σε ένα αποδοτικό προκαταστάτη με μικρό σχετικά ημι-εύρος (χυρίως η σταθμισμένη φασματική αναδιάταξη), είναι επόμενο να εξετάσουμε αν μπορεί να υπάρξει εύφορο έδαφος για την αποδοτική επίλυση γραμμικών συστημάτων με το προκαταστάτη M ως μητρώο συντελεστών (όπως αυτά που λύνονται εσωτερικά σε κάθε επανάληψη όταν χρησιμοποιούμε προκατάσταση στην επαναληπτική μέθοδο).

Εφόσον ο προκαταστάτης M είναι ένα ταινιακό μητρώο, μπορεί να επιλυθεί μέσω της μεθόδου Spike. Η ιδέα που εξετάζουμε εδώ είναι το κατά πόσον μπορούμε να χρησιμοποιήσουμε μεθόδους σποράς για τη ταχύτερη επίλυση των γραμμικών συστημάτων που προκύπτουν στη φάση προεπεξεργασίας της μεθόδου Spike. Στη συνέχεια θα επικεντρωθούμε στο προκαταστάτη M που προκύπτει ύστερα από την αναδιάταξη του μητρώου A μέσω της σταθμισμένης φασματικής αναδιάταξης (αυτή η αναδιάταξη βρέθηκε ως η βέλτιστη από τα πειράματα μας).

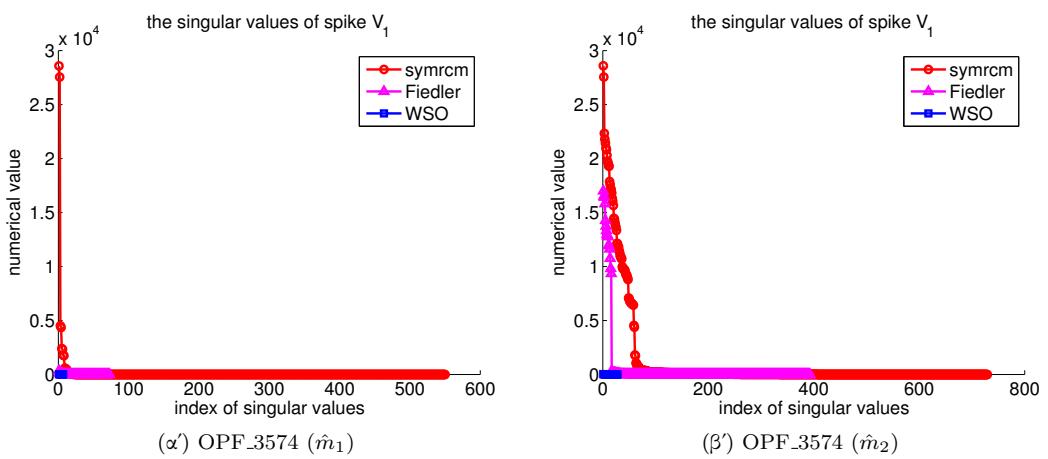
Η σταθμισμένη φασματική αναδιάταξη αναδιατάσσει τα στοιχεία μεγαλύτερου απολύτου μεγέθους εγγύτερα στη κύρια διαγώνιο, επομένως άμεσα προκύπτει η ιδέα του αν μπορούμε να προσεγγίσουμε τα στοιχεία που είναι μακριά από τη κύρια διαγώνιο βάσει ενός μητρώου χαμηλότερης τάξης - το οποίο είναι ισοδύναμο με το ότι τα μητρώα B_j^M , C_j^M είναι μητρώα χαμηλής τάξης (τάξη αρκετά μικρότερη από τις τιμές \hat{m}_1 , \hat{m}_2). Αν όντως ισχύει αυτό, τότε βάσει των μεθόδων σποράς και της συζήτησης που προηγήθηκε στο προηγούμενο κεφάλαιο, μπορούμε να λύσουμε μόνο για τα συστήματα τα οποία σχηματίζουν μια τυχαία βάση.

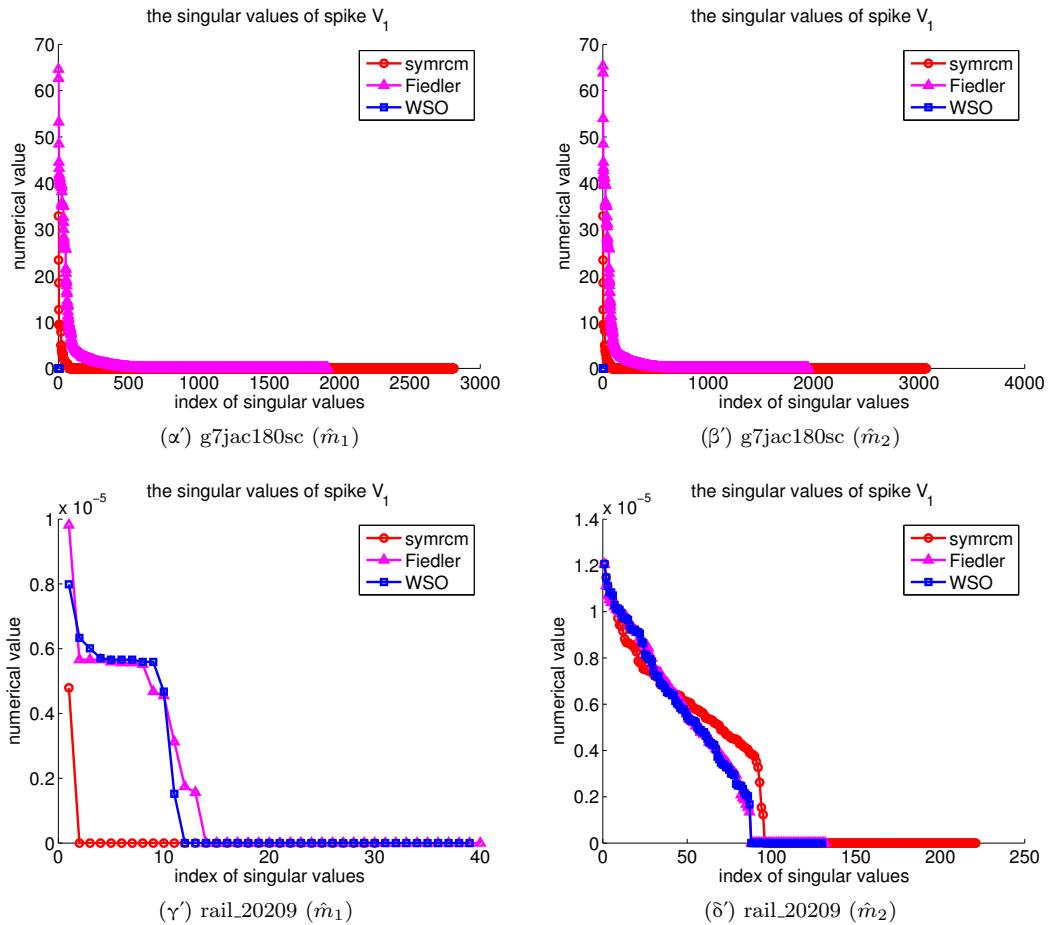
Στην αναζήτηση μας αυτή, στα Σχήματα 4.6 και 4.7 παραθέτουμε τις ιδιάζουσες τιμές του μητρώου B_1^M για τους διαφορούς προκαταστάτες M που προκύπτουν από κάθε διαφορετικό αλγόριθμο αναδιάταξης και για $p = 2$ διαχωρίσματα. Μπορούμε να δούμε ότι το μητρώο B_1^M έχει αρκετές ιδιάζουσες τιμές ίσες με μηδέν, που πρακτικά σημαίνει ότι η τάξη του είναι αρκετά μικρότερη από τη μέγιστη δυνατή (η οποία είναι \hat{m}_1 ή \hat{m}_2 ανάλογα με το μέγεθος της νόρμας του A που καταλαμβάνει το μητρώο M). Από τις τρεις διαφορετικές τεχνικές αναδιάταξης, η σταθμισμένη φασματική αναδιάταξη είναι αυτή για την οποία οι ιδιάζουσες τιμές του μητρώου B_1^M είναι πιο κοντά στο μηδέν. Μια τελευταία παρατήρηση είναι ότι ενώ βλέπουμε τη πλειονότητα των ιδιαζουσών τιμών να είναι μηδενικές -για όλα τα μητρώα και για όλες τις τεχνικές αναδιάταξης- συχνά αυτό προκύπτει λόγω μηδενικών στηλών και όχι γιατί υπάρχει κάποια μη-τετραμένη γραμμική εξάρτηση. Η παρατήρηση αυτή χρήζει επιπρόσθετης ανάλυσης την οποία σκοπεύουμε να διεξάγουμε στο μέλλον.





$\Sigma\chi\rho\mu\alpha$ 4.6: Οι ιδιάζουσες τιμές των Spike-πλοκών V_1





Σχήμα 4.7: Οι ιδιάζουσες τιμές των spike-πλοκών V_1

4.3 Εφαρμογές στην ανάλυση δεδομένων

Στη παρούσα ενότητα θα επικεντρωθούμε στην εφαρμογή των παραπάνω αλγορίθμων σε ένα πρόβλημα ανάλυσης δεδομένων. Πιο συγκεκριμένα, θα ασχοληθούμε με το πρόβλημα της ποσοτικοποίησης της αβεβαιότητας (uncertainty quantification) όπου τα μητρώα συνδιακύμανσης διαδραματίζουν κεντρικό ρόλο. Η ποσοτικοποίηση της αβεβαιότητας ασχολείται με την ανάλυση του βαθμού εμπιστοσύνης των οποίοι μπορούμε να έχουμε στα δεδομένα μας και τα οποία μπορούν να προέρχονται από διάφορα πεδία επιστημών (χρηματοοικονομική, γεωλογία κάποια από αυτά). Επειδή αυτά τα δεδομένα στη συνέχεια χρησιμοποιούνται για την εξαγωγή διαφόρων συμπερασμάτων (παραδείγματος χάριν για την επιλογή του αν μια πιθανή επένδυση αποτελεί μεγάλο ή μικρό ρίσκο) είναι προφανές ότι αυτό το βήμα προ-επεξεργασίας είναι ιδιαίτερα σημαντικό. Μια διαδεδομένη μετρική της εμπιστοσύνης που μπορούμε να έχουμε στα δεδομένα, προέρχεται από τη διαγώνιο του αντιστρόφου του μητρώου συνδιακύμανσης [25]. Λόγω των μεγάλων διαστάσεων καθώς και του μεγάλου αριθμού μη-μηδενικών στοιχείων των μητρώων συνδιακύμανσης σε διάφορες εφαρμογές, ο υπολογισμός της πραγματικής διαγωνίου εισάγει απαγορευτικό κόστος για ένα σύγχρονο υπολογιστικό σύστημα. Μια εναλλακτική είναι να χαλαρώσουμε την απαίτηση για τον υπολογισμό της ακριβής διαγωνίου του αντιστρόφου με μια ικανοποιητική προσέγγιση (συνήθως μερικά σημαντικά ψηφία). Η προσέγγιση αυτή επιτυγχάνεται μέσω πιθανοτικών τεχνικών Monte Carlo, και πιο συγκεκριμένα μέσω στοχαστικής εκτίμησης.

Έστω $A \in \mathbb{R}^{n \times n}$ το μητρώο συνδιακύμανσης και έστω $F \in \mathbb{R}^{n \times s}$ ένα μητρώο με s στήλες όπου ισχύει ότι $f_i^{(j)} = \pm 1, \forall i, j, i = 1, \dots, n$ και $j = 1, \dots, s$. Δηλαδή, τα στοιχεία του F παίρνουν τις τιμές $+1, -1$. Στη συνέχεια, θα θεωρήσουμε ότι η πιθανότητα ενός συγκεκριμένου στοιχείου i , μιας στήλης j , να ισούται με $+1, -1$ είναι ίση με $1/2$. Τότε, η διαγώνιος του A^{-1} , $d(A^{-1})$, μπορεί να εκτιμηθεί (προσεγγιστεί) ως

$$\hat{d}(A^{-1}) = \left[\sum_{j=1}^s f^{(j)} \odot A^{-1} f^{(j)} \right] \oslash \left[\sum_{j=1}^s f^{(j)} \odot f^{(j)} \right] \quad (4.1)$$

όπου \odot, \oslash συμβολίζουν το στοιχείο προς στοιχείο πολλαπλασιασμό και διαίρεση αντίστοιχα [26].

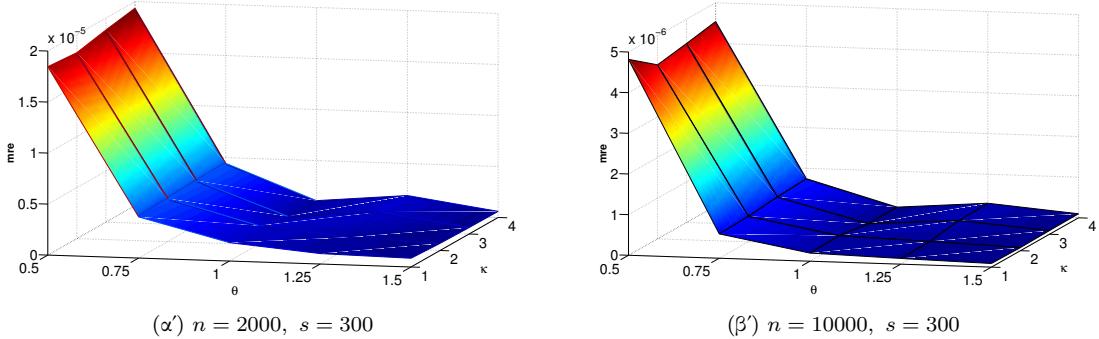
Όπως μπορούμε να παρατηρήσουμε, η στοχαστική εκτίμηση του $d(A^{-1})$ απαιτεί την επίλυση s γραμμικών συστημάτων της μορφής $Ax^{(j)} = f^{(j)}, \forall j, j = 1, \dots, s$. Εφόσον όλα τα δεξιά μέλη $f^{(j)}$ είναι άμεσα διαθέσιμα, το πρόβλημα μπορεί να γραφεί ως $A[x^{(1)}, \dots, x^{(s)}] = [f^{(1)}, \dots, f^{(s)}]$ όπου είναι φανερό ότι πρόκειται για ένα γραμμικό σύστημα με πολλά δεξιά μέλη. Αν το μητρώο συντελεστών είναι διαθέσιμο μόνον μέσω πράξεων MV, MM, τότε οι τεχνικές επίλυσης που βασίζονται στην οικογένεια αλγορίθμων Spike δεν μπορούν να χρησιμοποιηθούν, όπως το ίδιο ισχύει και για όποια μέθοδο χρησιμοποιεί κάποιας μορφής παραγοντοποίηση του A . Σε αυτή τη περίπτωση μπορούν να εφαρμοστούν τεχνικές οι οποίες δεν απαιτούν το μητρώο A να είναι άμεσα διαθέσιμο, βλέπε [27].

Αν θεωρήσουμε ότι το μητρώο συνδιακύμανσης είναι άμεσα διαθέσιμο (ίσως μετά από κάποιο στάδιο προ-επεξεργασίας και ανάλυσης), τότε μπορούμε να επιλύσουμε το γραμμικό σύστημα $AX = F$ βάσει των μεθόδων που παραθέσαμε στα προηγούμενα κεφάλαια. Η ιδέα μας είναι να εξάγουμε ένα ταινιακό υπομητρώο M το οποίο θα χρησιμοποιηθεί ως προκαταστάτης για την επίλυση του γραμμικού συστήματος $AX = F$. Στη συνέχεια, η εφαρμογή του προκαταστάτη υλοποιείται με βάση αλγορίθμους επίλυσης γραμμικών συστημάτων με πολλά δεξιά μέλη. Στη περίπτωση που το μητρώο συνδιακύμανσης δεν είναι άμεσα διαθέσιμο, π.χ. ορίζεται ως το γινόμενο $A = KK^\top$ με $K \in \mathbb{R}^{n \times n}$, θα πρέπει να κατασκευάσουμε το υπομητρώο M μέσω εσωτερικών γινομένων μεταξύ των στηλών-γραμμών του K .

Για τους σκοπούς της εργασίας θα θεωρήσουμε πρότυπα μητρώα συνδιαχύμανσης για τα οποία έχει υπάρξει κάποια τυπική προεπεξεργασία. Τα πρότυπα αυτά μητρώα ορίζονται ως

$$a_{ii} = 1 + i^\theta, \quad a_{ij} = \frac{1}{|i - j|^\kappa} \quad (i \neq j), \quad i, j = 1, \dots, n \quad (4.2)$$

όπου θ και κ είναι θετικοί βαθμωτοί και θα αναφερόμαστε σε αυτούς ως βαθμό κυριαρχίας και βαθμό απόσβεσης αντιστοίχως. Αξίζει να σημειωθεί ότι παρότι τα στοιχεία των πρότυπων μητρώων συνδιαχύμανσης τα οποία χρησιμοποιούμε ακολουθούν μια ομοιόμορφη απόσβεση η οποία είναι ίσως εξιδανικευμένη, εν τούτοις δεν εισάγουμε κάποια σημαντική ιδιομορφία στο υπό εξέταση πρόβλημα καθώς οι αλγόριθμοι μας κάνουν χρήση της γενικής αποσβένουσας συμπεριφοράς του A και όχι της αυστηρής της μορφής. Το τελευταίο σχετίζεται περισσότερο με την ακρίβεια την οποία μπορεί να επιτύχει κάποιος μέσω αλγορίθμων στοχαστικής εκτίμησης.

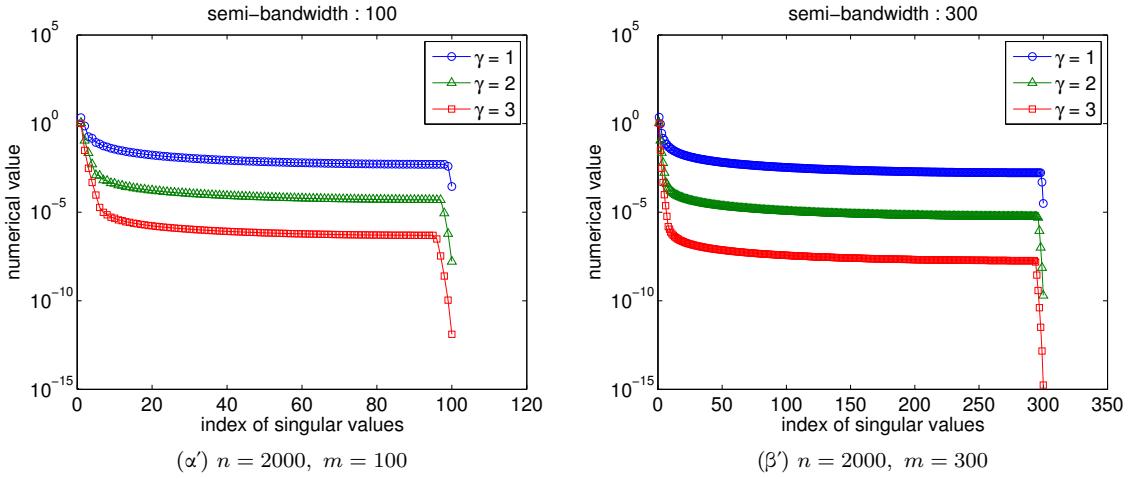


Σχήμα 4.8: Μέσο σχετικό σφάλμα της προσέγγισης του $d(A^{-1})$ μέσω στοχαστικής εκτίμησης. Τα γραφήματα αποτελούν μέρος της εργασίας [27].

Το Σχήμα 4.8 απεικονίζει το μέσο σχετικό σφάλμα της προσέγγισης του $d(A^{-1})$ μέσω στοχαστικής εκτίμησης (εξίσωση 4.1) για δύο πρότυπα μητρώα διαφορετικών διαστάσεων, τα οποία έχουν κατασκευαστεί βάσει της εξίσωσης (4.2). Όπως μπορεί να φανεί, για τα πρότυπα μητρώα στα οποία αναφερόμαστε, η στοχαστική εκτίμηση μπορεί να προσεγγίσει αρκετά καλά τη ποσότητα $d(A^{-1})$, ιδιαίτερα καθώς αυξάνεται ο βαθμός διαγώνιας κυριαρχίας. Στη παρούσα εργασία δε θα εξετάσουμε περαιτέρω την ακρίβεια της προσέγγισης της $d(A^{-1})$ μέσω στοχαστικής εκτίμησης, παραπέμπουμε στο [26] για πληροφορίες σχετικά με τη μέθοδο και στα [25, 26, 27] για εφαρμογές της μεθόδου σε διάφορα μητρώα.

Έστω τώρα ότι θέλουμε να επιλύσουμε το γραμμικό σύστημα που προέκυψε από την εξίσωση (4.1) μέσω μιας επαναληπτικής μεθόδου. Λόγω του ότι το “βαρύ” μέρος του A βρίσκεται κοντά στη κύρια διαγώνιο, είναι λογικό να εξάγουμε ως προκαταστάτη M ένα ταινιακό υπομητρώο του A το οποίο αποτελείται από τη κύρια διαγώνιο του A και μερικές από τις πρώτες υπερδιαγωνίους/υποδιαγωνίους. Επειδή το A είναι συμμετρικό επιλέγουμε τον ίδιο αριθμό από υπερδιαγωνίους/υποδιαγωνίους, έστω $m - 1$ από αυτές έτσι ώστε το M να είναι ταινιακό με ημι-εύρος m . Επιλέγουμε p διαχωρίσματα και ορίζουμε τα υπομητρώα του M , $M_1, \dots, M_p, B_1^M, \dots, B_{p-1}^M$ και C_2^M, \dots, C_p^M .

Το Σχήμα 4.9 απεικονίζει τις ιδιάζουσες τιμές του υπομητρώου B_1^M για $m = 100$ και $m = 300$ ενός πρότυπου μητρώου συνδιαχύμανσης A μικρού μεγέθους και για διαφορετικούς βαθμούς απόσβεσης κ . Ο αριθμός των διαχωρισμάτων ήταν ίσος με δύο ($p = 2$). Μπορούμε να δούμε ότι οι ιδιάζουσες τιμές του B_1^M ακολουθούν



Σχήμα 4.9: Αριθμητική τιμή των ιδιάζουσαν τιμών του υπομητρώου B_1^M .

μια αποσβένουσα συμπεριφορά και ανάλογα του βαθμού απόσβεσης κ , υπάρχει μόνο ένας μικρός αριθμός από αυτές που η τιμή τους δεν είναι κοντά στο μηδέν. Τα αποτελέσματα είναι αντιπροσωπευτικά για τη συγκεκριμένη κλάση μητρώων ανεξαρτήτως του μεγέθους n .

Τέλος, θεωρήσαμε την επίλυση του γραμμικού συστήματος $AX = F$ με $F \in \mathbb{R}^n$ ένα τυχαίο διάνυσμα και A ορισμένο όπως στην εξίσωση (4.2). Θεωρήσαμε δύο διαφορετικά μεγέθη $n = 2 \times 10^5$ και $n = 3 \times 10^5$ και δύο διαφορετικούς βαθμούς απόσβεσης $\kappa = 2$ και $\kappa = 3$. Όπως προκύπτει και από την εξίσωση (4.2), όσο μεγαλύτερη η τιμή του κ , τόσο πιο έντονος ο βαθμός απόσβεσης στα μη-διαγώνια στοιχεία του A καθώς απομακρυνόμαστε από τη κύρια διαγώνιο. Προσέξτε ότι το A είναι ένα πυκνό μητρώο και η πλειονότητα των στοιχείων του εκτός μιας στενής μπάντας γύρω από τη κύρια διαγώνιο αποτελείται από στοιχεία τα οποία πρακτικά είναι πολύ κοντά στη τιμή μηδέν. Επομένως μπορούμε να θεωρήσουμε ένα προκαταστάτη M με ημι-εύρος m ο οποίος συγκεντρώνει τα μεγαλύτερα στοιχεία του μητρώου A (το μητρώο A επιδεικνύει μια φυσική αποσβένουσα συμπεριφορά επομένως δε χρειάζεται κάποια αναδιάταξη). Όποτε χρησιμοποιήθηκε κάποιος επαναληπτικό επιλυτής, η ακρίβεια tol τέθηκε πάντα ίση με $tol = 10^{-4}$.

Επιλύσαμε το γραμμικό σύστημα $AX = F$ μέσω προκατάστασης με προκαταστάτη M ένα τανιακό υπομητρώο του A , με κάθε εσωτερικό βήμα προκατάστασης να εκτελείται βάσει της μεθόδου Spike όπου η φάση προεπεξεργασίας υλοποιήθηκε βάσει τριών διαφορετικών μεθόδων. Ως άμεσο επιλυτή επιλέξαμε τη παραγοντοποίηση Cholesky καθώς τα πρότυπα μητρώα ήταν ΣΘΟ. Από τη κλάση των επαναληπτικών μεθόδων, επιλέξαμε δύο διαφορετικά επαναληπτικά σχήματα. Το πρώτο σχήμα βασίζεται εξολοκλήρου στον αλγόριθμο CG βάσει του οποίου επιλύονται όλα τα τοπικά γραμμικά συστήματα. Το δεύτερο αλγορίθμικό σχήμα βασίζεται στο συνδυασμό των αλγορίθμων SSCG και CG. Πιο συγκεκριμένα, επιλέξαμε τα πρώτα τρία δεξιά μέλη να επιλύονται μέσω της μεθόδου SSCG ενώ τα υπόλοιπα $m - 3$ μέσω του αλγορίθμου CG. Η αφορμή για το δεύτερο αλγορίθμικό σχήμα δόθηκε από πειράματα τα οποία διεξάγαμε με τον αλγόριθμο SSCG και στα οποία είδαμε ότι ο χρόνος περάτωσης της φάσης προβολών ήταν υπερβολικά υψηλός αν όλα τα δεξιά μέλη επιλύονταν μέσω αυτού (ακόμα και αν η σύγκλιση σε όρους αριθμού επαναλήψεων ήταν ταχύτερη). Εκτός αυτού, η υπολογιστική απόδοση του σχήματος ήταν ιδιαίτερα χαμηλή λόγω των πολλών εσωτερικών γινομένων τα οποία έπρεπε να εκτελεστούν.

Πίνακας 4.9: Χρόνοι εκτέλεσης για $n = 2 \times 10^5$ και $\kappa = 2, 3$.

	SSCGa	SSCGb	(SS)CGc	CG	Cholesky
$\kappa = 2$					
$m = 100$	0.5	0.7	0.1	2.0	3.3
$m = 200$	1.0	1.5	0.8	5.3	5.1
$m = 300$	2.3	3.0	4.3	18.2	10.8
$m = 400$	2.7	3.9	10.4	34.7	13.0
$m = 500$	2.4	4.1	18.9	62.6	17.6
$\kappa = 3$					
$m = 100$	1.8	1.1	-	2.0	3.3
$m = 200$	1.9	2.1	-	6.4	6.0
$m = 300$	2.4	3.8	-	19.2	9.8
$m = 400$	3.4	5.3	-	36.8	11.6
$m = 500$	4.3	7.1	-	64.0	14.4

Πίνακας 4.10: Χρόνοι εκτέλεσης για $n = 3 \times 10^5$ και $\kappa = 2, 3$.

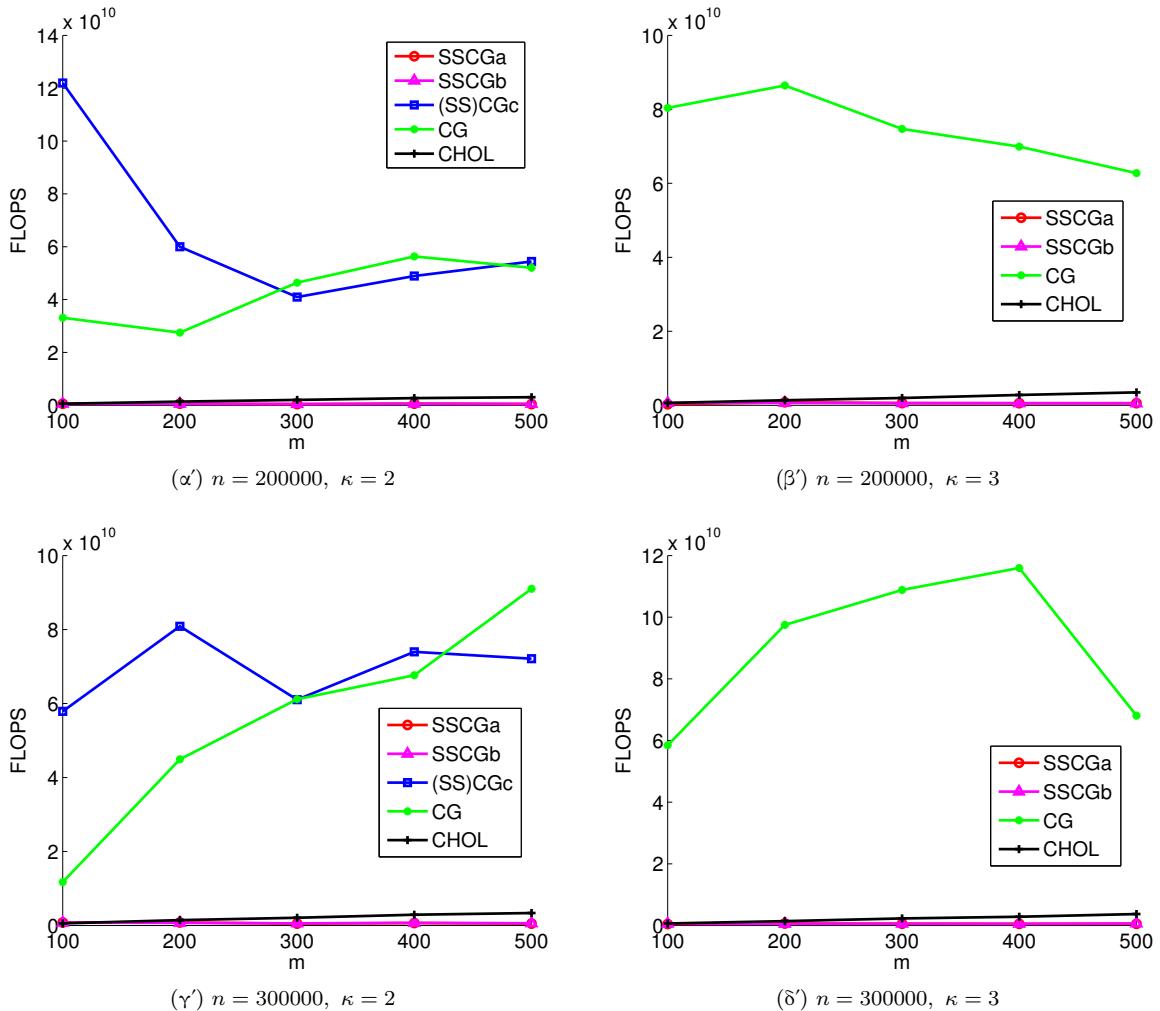
	SSCGa	SSCGb	(SS)CGc	CG	Cholesky
$\kappa = 2$					
$m = 100$	1.4	1.6	0.2	4.2	4.9
$m = 200$	2.0	3.2	1.2	9.8	8.6
$m = 300$	3.3	5.0	5.2	21.0	13.3
$m = 400$	5.2	7.1	10.5	32.9	16.4
$m = 500$	7.1	8.7	18.0	72.6	22.4
$\kappa = 3$					
$m = 100$	1.5	1.8	-	4.1	4.8
$m = 200$	2.2	4.0	-	9.9	8.7
$m = 300$	3.9	5.6	-	19.9	12.0
$m = 400$	5.3	7.8	-	33.2	17.2
$m = 500$	6.6	8.5	-	72.0	20.3

Οι Πίνακες 4.9 και 4.10 παραθέτουν τους χρόνους εκτέλεσης για κάθε δυνατό συνδυασμό των παραμέτρων n , m και κ . Ο Πίνακας 4.9 παραθέτει τα αποτελέσματα για $n = 2 \times 10^5$ ενώ ο Πίνακας 4.10 αυτά για $n = 3 \times 10^5$. Οι συντομεύσεις SSCGa και SSCGb συμβολίζουν το χρόνο που αναλώθηκε στις φάσεις επίλυσης και προβολών κατά τη διάρκεια του αλγορίθμου SSCG αντιστοιχα (μόνο τα τρία πρώτα δεξιά μέλη επιλύθηκαν). Η συντόμευση (SS)CGc αντιστοιχεί στην επίλυση των υπολοίπων $m - 3$ δεξιών μελών όπου οι αρχικές προτεινόμενες λύσεις έχουν υπολογιστεί από τον αλγόριθμο SSCG για την επίλυση των πρώτων τριών δεξιών μελών μόνο. Να σημειωθεί ότι οι φάσεις (SS)CGc και CG υλοποιήθηκαν κατά φευδοπλοκάδες.

Τα αποτελέσματα είναι παρόμοια και στις δύο περιπτώσεις και κάποια γενικά συμπεράσματα μπορούν να εξαχθούν. Αρχικά μπορούμε να δούμε ότι η μέθοδος CG είναι ιδιαίτερα χρονοβόρα, ειδικά όσο αυξάνεται το ημι-εύρος m . Σε όλες τις περιπτώσεις, η επίλυση με χρήση της μειόδου CG χρειάστηκε 4-5 επαναλήψεις. Από την άλλη, αν περάσουμε ως προτεινόμενες λύσεις στον αλγόριθμο CG αυτές που προκύπτουν από τον αλγόριθμο SSCG τότε το σχήμα μας μπορεί να είναι αρκετά πιο γρήγορο όπως αποδεικνύουν και οι χρόνοι των SSCGa, SSCGb και (SS)CGc. Αυτό μπορεί να εξηγηθεί από τον αριθμό επαναλήψεων της φάσης (SS)CGc που είναι μόλις μια για $\kappa = 2$ και καμία για $\kappa = 3$, ανεξαρτήτως του n . Τέλος, μπορούμε να δούμε ότι για

$\kappa = 2$ η άμεση μέθοδος είναι ταχύτερη από τα επαναληπτικά σχήματα, όμως για $\kappa = 3$ το επαναληπτικό μας σχήμα είναι ταχύτερο. Αυτό οφείλεται στο ότι τα δεξιά μέλη κατά τον υπολογισμό των spike-πλοκάδων μοιάζουν αρκετά και για συγκεκριμένη ακρίβεια tol , μπορούν να επιλυθούν κατευθείαν χωρίς την ανάγκη να εκτελεστεί η φάση (SS)CGc.

Θα πρέπει να σημειώσουμε ότι το πείραμα μας αφορούσε ένα μόνο δεξί μέλος στην εξίσωση $AX = F$. Στη πράξη θα χρειαστούμε αρκετά παραπάνω για να προσεγγίσουμε τη διαγώνιο του αντιστρόφου με καλή ακρίβεια. Το πόσα παραπάνω χυρίως εξαρτάται από τις μεταβλητές θ και κ . Στα πειράματα μας είδαμε ότι για τις συγκεκριμένες τιμές μεταβλητών τις οποίες εξετάσαμε, το επαναληπτικό σχήμα παραμένει ταχύτερο στη περίπτωση $\kappa = 3$ ακόμα και αν έχουμε $s = 20$ δεξιά μέλη. Πέρα από αυτό το σημείο, η άμεση μέθοδος είναι περισσότερο αποδοτική.



Σχήμα 4.10: FLOPS των μεθόδων για $n = 2 \times 10^5, 3 \times 10^5$ και $\kappa = 2, 3$.

Τέλος, το Σχήμα 4.10 δίνει μια εικόνα της απόδοσης με βάση τη ποσότητα FLOPS (δηλαδή το πλήθος πράξεων α.κ.υ. ανά δευτερόλεπτο) για τους διαφορετικούς αλγορίθμους επίλυσης της φάσης προεπεξεργασίας. Όπως μπορούμε να διαπιστώσουμε, το βήμα που βασίζεται στη μέθοδο CG με υλοποίηση

κατά ψευδοπλοκάδα έχει τη καλύτερη απόδοση, ακολουθούμενο από την επίλυση μέσω παραγοντοποίησης Cholesky. Το βήμα προβολών στη μέθοδο SSCG έχει, όπως αναμενόταν λόγω της υλοποίησης μέσω εσωτερικών γινομένων, τη χειρότερη απόδοση.

Κεφάλαιο 5

Επίλογος

Στη παρούσα εργασία ασχοληθήκαμε με την αποδοτική επίλυση της εξίσωσης μητρώων

$$AX = F, \quad (5.1)$$

όπου $A \in \mathbb{R}^{n \times n}$ το αραιό μητρώο συντελεστών, $F \in \mathbb{R}^{n \times s}$ το μητρώο δεξιών μελών και $X \in \mathbb{R}^{n \times s}$ το μητρώο λύση της εξίσωσης. Ο στόχος μας ήταν η επίλυση προβλημάτων μεγάλης κλίμακας, όπου οι διαστάσεις του μητρώου συντελεστών μπορεί να είναι της τάξης των εκατομμυρίων. Για αυτό το λόγο επικεντρωθήκαμε σε μεθόδους οι οποίες χρησιμοποιούν άνω του ενός επεξεργαστή (υπολογιστική μονάδα) για την επίλυση του προβλήματος. Στην εργασία αυτή περιοριστήκαμε στη χρήση μέχρι και οκτώ επεξεργαστών, σκοπός μας όμως είναι στο μέλλον να διεξάγουμε πειράματα με χρήση $O(100)$ επεξεργαστών.

Αρχικά αναφέραμε κάποια από τα πεδία εφαρμογών του υπό επίλυση προβλήματος. Στη συνέχεια περιγράψαμε τις βασικές κατηγορίες αλγορίθμων επίλυσης, τις άμεσες μεθόδους και τις επαναληπτικές μεθόδους. Παραθέσαμε τις βασικές ιδέες που διέπουν την κάθε κατηγορία και επισημάναμε τις διαφορές τους. Για την περίπτωση των επαναληπτικών μεθόδων, εισάγαμε την έννοια της προκατάστασης και παραθέσαμε σχήματα αναδιάταξης του μητρώου συντελεστών τα οποία μας βοηθούν να δημιουργήσουμε αποδοτικούς ταινιακούς προκαταστάτες.

Στο επόμενο κεφάλαιο, επικεντρωθήκαμε στην οικογένεια αλγορίθμων Spike η οποία περιέχει παράλληλους αλγόριθμους επίλυσης για διάφορα στιγμιότυπα του προβλήματος (1) όπως όταν το μητρώο συντελεστών A είναι ένα γενικό αραιό μητρώο, ή όταν το μητρώο συντελεστών είναι αραιό και ταινιακό (αραιό ή πυκνό εσωτερικά της μπάντας). Η αρχική μέθοδος Spike, η οποία εξελίχθηκε σημαντικά από τη πρώτη της μορφή στα τέλη της δεκαετίας του 1970, χειρίζεται την επίλυση της εξίσωσης (1) όταν το μητρώο συντελεστών είναι ταινιακό. Η εξαιρετική της επίδοση σε παράλληλες αρχιτεκτονικές έχει επαληθευτεί σε πλήθος επιστημονικών εργασιών. Στη περίπτωση που το μητρώο συντελεστών είναι και διαγώνια κυρίαρχο, η μέθοδος Spike μπορεί να εξειδικευτεί στον αλγόριθμο TR-Spike για να λάβει υπόψιν της αυτή την ιδιότητα και να κάνει ακόμα πιο γρήγορη την επίλυση του (1). Στη περίπτωση γενικών, αραιών μητρώων συντελεστών η μέθοδος Spike μπορεί να λειτουργήσει ως προκαταστάτης για την επίλυση της εξίσωσης (1) μέσω επαναληπτικών μεθόδων. Δεδομένου ότι η αρχική μέθοδος Spike χειρίζεται αποδοτικά τη παράλληλη επίλυση ταινιακών συστημάτων, είναι επόμενο να επιχειρήσουμε να χρησιμοποιήσουμε ταινιακούς προκαταστάτες ώστε να εκμεταλλευτούμε αυτό το στοιχείο. Ταυτόχρονα, ο προκαταστάτης θα πρέπει να είναι αποδοτικός υπό την έννοια ότι η επαναληπτική μέθοδος θα πρέπει να συγκλίνει μετά από ένα (πολύ) μικρό αριθμό επαναλήψεων. Σχολιάσαμε

τη χρησιμότητα των τεχνικών αναδιάταξης μητρώων από το προηγούμενο κεφάλαιο και παραθέσαμε τον αλγόριθμο PSpike για την επίλυση αραιών γραμμικών συστημάτων.

Στο τρίτο κεφάλαιο, παραθέσαμε τη βασική συνεισφορά της εργασίας η οποία βασίζεται στο συνδυασμό της οικογένειας αλγορίθμων Spike με τεχνικές επίλυσης γραμμικών συστημάτων για πολλά δεξιά μέλη. Πιο συγκεκριμένα, παρουσιάζουμε και αναλύουμε τις δύο βασικές κατηγορίες επίλυσης γραμμικών συστημάτων για πολλά δεξιά μέλη, τις μεθόδους σπόρου και τις μεθόδους πλοκάδας. Οι μέθοδοι αυτές βασίζονται σε επαναληπτικές μεθόδους επίλυσης γραμμικών συστημάτων. Η ιδέα πίσω από τις μεθόδους επίλυσης πολλών δεξιών μελών μελών βρίσκεται στην αξιοποίηση μαθηματικής πληροφορίας που έχει προέλθει από την επίλυση ως προς ένα ή περισσότερα δεξιά μέλη για τη ταχύτερη σύγκλιση (λιγότερες επαναλήψεις) των δεξιών μελών ως προς τα οποία δεν έχει ακόμα υπολογιστεί η λύση. Ανάλογα με τις συνθήκες του προβλήματος, οι μέθοδοι αυτές μπορούν να είναι ιδιαίτερα αποδοτικές. Αναλύσαμε αυτές τις συνθήκες και στη συνέχεια αναλύσαμε που υπάρχει προοπτική για καλύτερη επίδοση των μεθόδων Spike με χρήση μεθόδων επίλυσης πολλών δεξιών μελών και προτείναμε σχήματα τα οποία αποτελούν συνδυασμό της οικογένειας αλγορίθμων Spike και των μεθόδων επίλυσης πολλών δεξιών μελών.

Τέλος, στο τέταρτο κεφάλαιο παραθέσαμε μια ποικιλία αριθμητικών πειραμάτων τα οποία ως στόχο έχουν να αναδείξουν τα πλεονεκτήματα των αλγορίθμων τους οποίους προτείναμε. Είδαμε ότι σε διάφορες περιπτώσεις οι αλγόριθμοι αυτοί βελτίωσαν το χρόνο επίλυσης της οικογένειας μεθόδων Spike σε σχέση με τη χρήση άμεσων μεθόδων για την επίλυση των τοπικών προβλημάτων, και κατά συνέπεια του αρχικού προβλήματος (1).

5.1 Προοπτικές για το μέλλον

Γράφοντας τις τελευταίες γραμμιές της εργασίας αυτής, θα παραθέσουμε κάποιους προβληματισμούς για το μέλλον.

Όσον αφορά τις στήλες του μητρώου δεξιών μελών F , αυτές πάντα υποθέταμε ότι ήταν άμεσα διαθέσιμες. Αυτό όμως δεν ισχύει πάντα οπότε είναι εύλογο το ερώτημα για το αν μπορούμε να κάνουμε κάτι και για αυτή τη περίπτωση. Φυσικά μπορούμε να εφαρμόσουμε το επιλεγμένο αλγορίθμικό σχήμα ανεξάρτητα κάθε φορά. Είναι όμως αυτό το καλύτερο που μπορούμε να κάνουμε; Πιστεύουμε ότι ίσως υπάρχουν καλύτερα σχήματα τα οποία βασίζονται σε επαναληπτικές μεθόδους εύρεσης ιδιαίτερων τιμών του μητρώου $M^{-1}A$ με M το προκαταστάτη του μητρώου συντελεστών A .

Θα θέλαμε επίσης να συγκρίνουμε την επίδοση της οικογένειας αλγορίθμων Spike με χρήση μεθόδων πολλών δεξιών μελών ενάντια στην επίδοση ορισμένων παράλληλων άμεσων επιλυτών όπως οι MUMPS και Pardiso.

Τέλος, θα θέλαμε να εξετάσουμε περαιτέρω την αριθμητική συμπεριφορά των μεθόδων σπόρου και τη συμπεριφορά τους στο μοντέλο πεπερασμένης αριθμητικής ακρίβειας. Το πρόβλημα αυτό ίσως να είναι ανεξάρτητο από τη παρούσα εργασία, πάραμετα θα μπορούσε να έχει ιδιαίτερη χρησιμότητα και εδώ.

Στο μέλλον επίσης σκοπεύουμε να εντάξουμε τα παραπάνω αλγορίθμικά σχήματα στον αλγόριθμο WB-Spike καθώς λόγω της γενικής αραιής δομής των διαγώνιων πλοκάδων A_j οι επαναληπτικές μέθοδοι έχουν τη ποιότητα για ακόμα καλύτερα αποτελέσματα σε σχέση με τους άμεσους επιλυτές.

Παράρτημα Α'

Στοιχεία γραμμικής άλγεβρας

A'.1 Συμβολισμοί και ορολογία

Ονομάζουμε μητρώο $A \in \mathbb{R}^{m \times n}$

$$A = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1n} \\ \vdots & \ddots & \vdots \\ \alpha_{m1} & \dots & \alpha_{mn} \end{bmatrix}$$

ένα ορθογώνιο σχηματισμό $m n$ πραγματικών τιμών και διάνυσμα-στήλη $b \in \mathbb{R}^m$

$$b = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$$

μια στήλη m πραγματικών τιμών. Αν το b είναι διάνυσμα-γραμμή

$$b = [\beta_1 \ \dots \ \beta_m]$$

τότε $b \in \mathbb{R}^{1 \times m}$. Αν οι τιμές του $A(b)$ είναι μιγαδικοί αριθμοί, τότε $A \in \mathbb{C}^{m \times n}$ ($b \in \mathbb{C}^m$). Στη συνέχεια θα αναφερόμαστε πάντα στο χώρο \mathbb{R} , εκτός αν δηλωθεί αντιθέτως. Σε αντιστοιχία με την αριθμητική άλγεβρα, ορίζουμε πρόξεις μεταξύ μητρώων.

Ορισμός A'.1. Εστω δύο μητρώα A, C τα οποία είναι ιδίων διαστάσεων και ανήκουν στον $\mathbb{R}^{m \times n}$. Τότε ορίζουμε το άθροισμα τους $A + C \in \mathbb{R}^{m \times n}$ ως το άθροισμα των αντίστοιχων στοιχείων

$$(A + C)_{ij} = A_{ij} + C_{ij}. \quad (\text{A}'1)$$

Ομοίως, ορίζουμε την αφαίρεση $A - C$ των μητρώων A, C ως

$$(A - C)_{ij} = A_{ij} - C_{ij}. \quad (\text{A}'2)$$

Μια ακόμα πράξη που ορίζεται μεταξύ δύο μητρώων A, C είναι αυτή του πολλαπλασιασμού. Για να οριστεί ο πολλαπλασιασμός μεταξύ των μητρώων A, C θα πρέπει το πλήθος στηλών του A να ισούται με το πλήθος γραμμών του C .

Ορισμός Α'.2. Εστω μητρώο $A \in \mathbb{R}^{m \times p}$ και μητρώο $C \in \mathbb{R}^{p \times n}$. Τότε ορίζουμε το γινόμενο τους, $AC \in \mathbb{R}^{m \times n}$, όπου το στοιχείο (i, j) μπορεί να οριστεί ως το εσωτερικό γινόμενο της γραμμής i του A με τη στήλη j του C

$$(AC)_{ij} = \sum_{k=1}^p A_{ik} C_{kj}. \quad (\text{A'}.3)$$

Ανάλογα με τις διαστάσεις των μητρώων τα οποία πολλαπλασιάζονται μεταξύ τους, μπορούμε να διαχρίνουμε τρεις κύριες κατηγορίες: α) τα μητρώα είναι διανύσματα (εσωτερικό γινόμενο (DOT) ή εξωτερικό γινόμενο), β) το ένα από τα δύο μητρώα είναι διάνυσμα (MV), και γ) και τα δύο μητρώα έχουν παφαπάνω από μία γραμμή/στήλη (MM). Ο Πίνακας Α'.1 περιέχει ενδεικτικές πολυπλοκότητες για κάθε περίπτωση.

Πίνακας Α'.1: Πολυπλοκότητα πολλ/σμού μητρώων.

	$n = m = 1, p > 1$	$n = 1, p, m > 1$ ή $m = 1, p, n > 1$	$m, p, n > 1$
Όνομα πράξης	Εσωτερικό γινόμενο (DOT)	Μητρώο - διάνυσμα (MV)	Μητρώο - μητρώο (MM)
Πολυπλοκότητα	$2p - 1$	$m(2p - 1)$ ή $n(2p - 1)$	$mn(2p - 1)$

Αν $m = n$, το μητρώο A ονομάζεται τετραγωνικό (square matrix) και ενδέχεται να ορίζεται το μητρώο $A^{-1} \in \mathbb{R}^{n \times n}$ το οποίο είναι επίσης τετραγωνικό και έχει την ιδιότητα

$$AA^{-1} = A^{-1}A = I \quad (\text{A'}.4)$$

όπου $I \in \mathbb{R}^{n \times n}$ είναι το ταυτοτικό μητρώο (identity matrix) για το οποίο ισχύει ότι $I_{ij} = 1$ αν $i = j$ (αλλιώς $I_{ij} = 0$). Αν για κάποιο μητρώο A ορίζεται το αντίστοιχο A^{-1} , θα λέμε ότι το μητρώο A αντιστρέφεται και το μητρώο A^{-1} θα ονομάζεται ο αντιστροφός (inverse matrix) του A .

Εκτός του αντιστρόφου, ορίζουμε τον ανάστροφο (transpose matrix) ενός μητρώου $A \in \mathbb{R}^{m \times n}$ ως το μητρώο $A^\top \in \mathbb{R}^{n \times m}$ τα στοιχεία του οποίου ορίζονται ως $(A^\top)_{ij} = A_{ji}$. Αν ισχύει ότι $A = A^\top$, το A ονομάζεται συμμετρικό μητρώο (symmetric matrix). Ο ανάστροφος ενός μητρώου A υπάρχει πάντα.

Για τετραγωνικά μητρώα, ορίζεται μια ακόμα ποσότητα η οποία ονομάζεται ορίζουσα (determinant) και θα τη συμβολίζουμε με $\det()$. Η ορίζουσα ενός μητρώου $A \in \mathbb{R}^{n \times n}$ μπορεί να μας δώσει πληροφορίες για την αντιστρεψιμότητα του, καθώς αν $\det(A) = 0$ το μητρώο A δε μπορεί να αντιστραφεί και ονομάζεται ασύνηθες (singular). Αν $\det(A) \neq 0$, το A είναι αντιστρέψιμο και ονομάζεται σύνηθες (non-singular). Επιγραμματικά, αναφέρουμε μερικές ιδιότητες της ορίζουσας:

- $\det(AB) = \det(BA)$
- $\det(A^\top) = \det(A)$
- $\det(\alpha A) = \alpha^n \det(A)$
- $\det(I) = 1$

Πριν προχωρήσουμε στο επόμενο υποκεφάλαιο, θα αναφέρουμε συνοπτικά κάποιες ιδιαίτερα σημαντικές έννοιες της γραμμικής άλγεβρας οι οποίες σχετίζονται με την έννοια των γραμμικών (υπό)χώρων.

Ορισμός Α'.3. Εστω ένα σύνολο s διανυσμάτων $S = \{v_1, \dots, v_s\}$ τότε το σύνολο όλων των γραμμικών συνδυασμών των διανυσμάτων αποτελούν το διάνοιγμα (span) του χώρου που ορίζουν τα διανύσματα v και

ορίζεται ως

$$\text{span}(v_1, v_2, \dots, v_n) = \left\{ y \mid y = \sum_{j=1}^n a_j v_j \right\}$$

Ορισμός A'.4. Ενα σύνολο s διανυσμάτων $S = \{v_1, \dots, v_s\}$ ονομάζεται γραμμικά ανεξάρτητο όταν ισχύει

$$\alpha_1 v_1 + \dots + \alpha_s v_s = 0 \quad (\text{A'.5})$$

με μόνη λύση $\alpha_1 = \dots = \alpha_s = 0$. Αν υπάρχει έστω και ένας βαθμωτός $\alpha_i \neq 0$ τότε το σύνολο ονομάζεται γραμμικά εξαρτημένο.

Ορισμός A'.5. Έστω μητρώο $A \in \mathbb{C}^{m \times n}$, ονομάζουμε τάξη του μητρώου A , $\text{rank}(A)$, το μέγιστο πλήθος των γραμμικά ανεξάρτητων στηλών (ή γραμμών) του. Ισχύει $\text{rank}(A) \leq \min(m, n)$.

A'.2 Μητρώα ειδικής δομής

Το κόστος πράξεων μεταξύ μητρώων (ή διανυσμάτων) εξαρτάται από το πλήθος των μη-μηδενικών στοιχείων τους. Για παράδειγμα η πράξη του εσωτερικού γινόμενου μεταξύ δύο διανυσμάτων μεγέθους n κοστίζει n πολλαπλασιασμούς και $n - 1$ προσθέσεις μεταξύ βαθμωτών. Αν όμως κάποια στοιχεία είναι 0, τότε δε χρειάζεται να ληφθούν υπόψη. Θα ονομάζουμε ένα μητρώο $A \in \mathbb{R}^{n \times n}$ ως πυκνό (dense) όταν $\text{nnz}(A) = O(n^2)$, όπου $\text{nnz}(.)$ είναι η συνάρτηση που δέχεται ως άρισμα ένα μητρώο και επιστρέφει το πλήθος των μη-μηδενικών του στοιχείων. Αν ισχύει $\text{nnz}(A) = O(n)$, το A θα ονομάζεται αραιό (sparse). Η αραιότητα ενός μητρώου είναι εξαιρετικής σημασίας στον επιστημονικό υπολογισμό και την αριθμητική γραμμική άλγεβρα καθώς επηρεάζει άμεσα τη πολυπλοκότητα κόστους και μνήμης των αριθμητικών αλγορίθμων. Η πλειονότητα των μητρώων τα οποία προκύπτουν από τις διάφορες εφαρμογές είναι αραιά, πχ. μητρώα από τη διαχριτοποίηση χωρίων μέσω των μεθόδων διαχριτοποίησης πεπερασμένων στοιχείων ή πεπερασμένων διαφορών.

Εκτός της αραιότητας ή μη που μπορεί να εμφανίζει, ένα μητρώο A μπορεί επίσης να κατηγοριοποιηθεί και ως προς τη δομή του. Η δομή του A δύναται να “υποδείξει” την επιλογή της μεθόδου για την επίλυση γραμμικών συστημάτων και να βοηθήσει στη κατασκευή αποδοτικών αλγορίθμων. Για παράδειγμα, τυχόν θετική ορισμότητα ενός συμμετρικού μητρώου A υποδεικνύει ότι το γραμμικό σύστημα θα μπορούσε να λυθεί με τη μέθοδο των συζυγών κλίσεων (περισσότερα για αυτό σε επόμενο κεφάλαιο). Για τους παραπάνω λόγους, η (μη) πυκνότητα και η δομή του μητρώου A λειτουργούν ως οδηγός για την επιλογή της μεθόδου επίλυσης.

Παρακάτω αναφέρουμε συνοπτικά διαφόρους τύπους μητρώων και μια σύντομη περιγραφή τους:

- Διαγώνιο μητρώο (diagonal matrix): $A_{ij} \neq 0$, για $i == j$.
- Μοναδιαίο μητρώο (unitary matrix): $A^T A = I$ όπου I είναι το ταυτοτικό μητρώο.
- Άνω τριγωνικό μητρώο (upper triangular matrix): $A_{ij} = 0$ όταν $i < j$.
- Κάτω τριγωνικό μητρώο (lower triangular matrix): $A_{ij} = 0$ όταν $i > j$.
- Ταινιακό μητρώο (banded matrix) με άνω ημι-εύρος m_u και κάτω ημι-εύρος m_l : $A_{ij} = 0$ όταν $j > i + m_u$ ή όταν $j < i - m_l$.

Ορισμός A'.6. Ενα $n \times n$ μητρώο A θα λέγεται συμμετρικά θετικά ορισμένο (symmetric positive definite (SPD)) (εν συντομίᾳ ΣΘΟ) όταν είναι συμμετρικό και για κάθε $x \in \mathbb{R}^n$ διάφορο του μηδενικού διανύσματος ισχύει $x^T Ax > 0$. Ενα ΣΘΟ μητρώο A είναι πάντοτε αντιστρέψιμο.

A'.3 Νόρμα διανύσματος

Ορισμός A'.7. Ονομάζουμε νόρμα διανύσματος τη συνάρτηση $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ η οποία ικανοποιεί τις παρακάτω ιδιότητες

1. $\|x\| \geq 0$, $x \in \mathbb{R}^n$.
2. $\|x + y\| \leq \|x\| + \|y\|$, $x, y \in \mathbb{R}^n$.
3. $\|\alpha x\| = |\alpha| \|x\|$, $\alpha \in \mathbb{R}$, $x \in \mathbb{R}^n$.

Ο παραπάνω ορισμός αναφέρει τις γενικές ιδιότητες της συνάρτησης νόρμα, στη πραγματικότητα μπορούμε να ορίσουμε πάνω από μια νόρμα διανύσματος βάση μιας μεταβλητής $p \in \mathbb{R}$. Η p νόρμα ορίζεται ως

$$\|x\|_p = (|x_1|^p + \dots + |x_n|^p)^{1/p}, \quad p \geq 1. \quad (\text{A'.6})$$

Οι πιο γνωστές p νόρμες είναι οι $p = 1, 2$ και ∞ ,

$$\|x\|_1 = |x_1| + \dots + |x_n| \quad (\text{A'.7})$$

$$\|x\|_2 = (|x_1|^2 + \dots + |x_n|^2)^{1/2} \quad (\text{A'.8})$$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \quad (\text{A'.9})$$

Η νόρμα $p = 2$ είναι επίσης γνωστή ως ευκλείδεια νόρμα.

Μια σημαντική ιδιότητα της νόρμας διανυσμάτων είναι η ανισότητα Holder όπου αν $x, y \in \mathbb{R}^n$, τότε ισχύει ότι

$$|x^T y| \leq \|x\|_p \|y\|_q, \quad \frac{1}{p} + \frac{1}{q} = 1. \quad (\text{A'.10})$$

Ιδιαίτερο επίσης ενδιαφέρον παρουσιάζει μια ειδική περίπτωση της ανισότητας Holder, η ανισότητα Cauchy-Bunyakovsky-Schwarz. Έστω $x, y \in \mathbb{R}^n$, τότε

$$|x^T y| \leq \|x\|_2 \|y\|_2, \quad (\text{A'.11})$$

με την ισότητα να ισχύει αν και μόνον αν τα x, y είναι συγγραμμικά. Όσον αφορά τη σχέση μεταξύ διανυσματικών νορμών, ισχύει το παρακάτω θεώρημα.

Θεώρημα A'.8. Όλες οι νόρμες (διανύσματος) επί του \mathbb{R}^n είναι ισοδύναμες υπό την έννοια ότι υπάρχουν σταθερές c_1, c_2 τέτοιες ώστε

$$c_1 \|x\|_\alpha \leq \|x\|_\beta \leq c_2 \|x\|_\alpha \quad \forall x \in \mathbb{C}^n \quad (\text{A'.12})$$

για οποιεσδήποτε νόρμες μ $p = \alpha, \beta$.

A'.4 Νόρμα μητρώου

Ορισμός A'.9. Ονομάζουμε νόρμα μητρώου τη συνάρτηση $\|\cdot\| : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ η οποία ικανοποιεί τις παρακάτω ιδιότητες

1. $\|A\| \geq 0, \quad A \in \mathbb{R}^{m \times n}$.
2. $\|A + B\| \leq \|A\| + \|B\|, \quad A, B \in \mathbb{R}^{m \times n}$.
3. $\|\alpha A\| = |\alpha| \|A\|, \quad \alpha \in \mathbb{R}, A \in \mathbb{R}^{m \times n}$.

Όπως και με τις νόρμες διανύσματων, ορίζονται πάνω από μια νόρμες μητρώου. Οι περισσότερο γνωστές νόρμες μητρώων είναι η νόρμα Frobenius (ή ευκλείδεια νόρμα μητρώου)

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2} \quad (\text{A'.13})$$

και η p νόρμα

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \quad (\text{A'.14})$$

Όπως και για τις νόρμες διανύσματος, υπάρχουν κάποιες νόρμες μητρώου οι οποίες είναι υπολογίσιμες.

1. Η νόρμα 1,

$$\|A\|_1 = \max_{j \in n} \left| \sum_{i=1}^n a_{ij} \right|. \quad (\text{A'.15})$$

2. Η νόρμα ∞ ,

$$\|A\|_{\infty} = \max_{i \in m} \left| \sum_{j=1}^n a_{ij} \right|. \quad (\text{A'.16})$$

3. Η νόρμα 2 (φασματική νόρμα),

$$\|A\|_2 = \sqrt{\lambda_n(A^\top A)} = \sigma_{max}. \quad (\text{A'.17})$$

Ένας περισσότερο φορμαλιστικός ορισμός της νόρμας ενός μητρώου συνδέεται με ένα τυχαίο, μη-μηδενικό διάνυσμα $x \in \mathbb{R}^n$ και κάποια νόρμα αυτού, $\|x\|$. Πιο συγκεκριμένα, δοθέντος κάποιας νόρμας διανύσματος $\|\cdot\|$, η νόρμα ενός μητρώου A ορίζεται ως

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}, \quad (\text{A'.18})$$

οπότε προκύπτει ότι η νόρμα ενός μητρώου A μετράει τη μέγιστη μεταβολή που μπορεί να προκαλέσει το μητρώο A στο μη-μηδενικό διάνυσμα x . Στη συνέχεια ορίζουμε μια εξαρετικά σημαντική έννοια η οποία συνοδεύει ένα τετραγωνικό μητρώο A , αυτή του δείκτη κατάστασης.

Ορισμός A'.10. Έστω ένα $n \times n$ μητρώο A . Ονομάζουμε δείκτη κατάστασης του A το γινόμενο $\kappa(A) = \|A\|_p \|A^{-1}\|_p$ με $\|\cdot\|_p$ μια οποιαδήποτε νόρμα. Αν το μητρώο A είναι ΣΤΟ, τότε $\kappa(A) = \frac{\lambda_n}{\lambda_1}$ με λ_n και λ_1 τη μεγαλύτερη και μικρότερη ιδιοτιμή του A αντίστοιχα. Εξ' ορισμού, $\kappa(A) \geq 1$, $\kappa(\gamma A) = \kappa(A)$ και $\kappa(A) = \infty$ αν το A δεν είναι αντιστρέψιμο.

Αναλύοντας τον ορισμό έχουμε

$$\|A\|\|A^{-1}\| = \left(\max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \right) \left(\min_{x \neq 0} \frac{\|Ax\|}{\|x\|} \right)^{-1} \quad (\text{A'.19})$$

που σημαίνει ότι ο δείκτης κατάστασης ενός μητρώου είναι ο λόγος της μέγιστης μεταβολής προς τη μέγιστη συρρίκνωση που το A προκαλεί σε ένα μη μηδενικό διάνυσμα x . Η πρωτική σημασία του δείκτη κατάστασης είναι ότι όσο μεγαλύτερη είναι η τιμή του, τόσο περισσότερο το A τείνει να είναι μη-αντιστρέψιμο.

Τέλος, κάτι το οποίο ισχύει και για τις νόρμες διανύσματος αλλά αναφέρεται μόνο εδώ για οικονομία χώρου, οι νόρμες $\|A\|_{2,F}$ ενός μητρώου $A \in \mathbb{R}^{m \times n}$ είναι αμετάβλητες υπό το πολλαπλασιασμό με ένα μοναδιαίο (ορθογώνιο) μητρώο $Q \in \mathbb{R}^{m \times m}$,

$$\|QA\|_2 = \|A\|_2, \quad \|QA\|_F = \|A\|_F. \quad (\text{A'.20})$$

A'.5 Η παραγοντοποίηση QR

Η παραγοντοποίηση QR εκφράζει το μητρώο $A \in \mathbb{R}^{m \times n}$ ($m \geq n$) ως γινόμενο δυο μητρώων $Q \in \mathbb{R}^{m \times m}$, $R \in \mathbb{R}^{m \times n}$ με Q ορθογώνιο και R άνω τριγωνικό, έτσι ώστε

$$A = QR = [Q_1, Q_2] \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = Q_1 R_1 + Q_2 R_2 \quad (\text{A'.21})$$

με $Q_1 \in \mathbb{R}^{m \times n}$, $R_1 \in \mathbb{R}^{n \times n}$. Εφόσον οι $m - n$ τελευταίες γραμμές του R είναι μηδενικές, οι στήλες του μητρώου Q_2 δε συνεισφέρουν στο γινόμενο και επομένως μπορούμε να εκφράσουμε τη παραγοντοποίηση QR στη συμπυκνωμένη μορφή $A = Q_1 R_1$ [18].

Η παραγοντοποίηση QR σχηματίζει μια ορθοκανονική βάση των στηλών του A δίχως να “περιστρέψει” το χώρο στηλών ($\text{range}(A)$). Ισχύει δηλαδή ότι

$$\text{span}\{a^{(1)}, \dots, a^{(k)}\} = \text{span}\{q^{(1)}, \dots, q^{(k)}\} \quad \forall k, \quad k = 1, \dots, n \quad (\text{A'.22})$$

όπου $a^{(k)}$, $q^{(k)}$ είναι η k -στη στήλη των A και Q αντίστοιχα. Επομένως, οι γραμμικοί συνδυασμοί που προκύπτουν από τις στήλες του A είναι αριθμώς οι ίδιοι που προκύπτουν από τις στήλες του Q_1 . Άν Q_1 , Q_2 οριστούν όπως πιο πάνω, τότε ισχύει ότι

$$\text{range}(A) = \text{range}(Q_1)$$

και

$$\text{range}(A)^\perp = \text{range}(Q_2).$$

Ο υπολογισμός της παραγοντοποίησης QR υλοποιείται βάσει διαφόρων μεθόδων όπως η διαδικασία Gram-Schmidt, οι μετασχηματισμοί Householder και οι περιστροφές Givens. Θα αναλύσουμε τη πρώτη και ίσως πιο εύκολη ως προς τη κατανόηση μέθοδο, τη μέθοδο Gram-Schmidt.

A'.5.1 Gram-Schmidt

Την θέση με χάριν απλότητας ότι το μητρώο $B \in \mathbb{R}^{n \times s}$ περιέχει s γραμμικά ανεξάρτητες στήλες. Δοθέντων δυο διανυσμάτων $z, y \in \mathbb{R}^n$, ορίζουμε το τελεστή ορθογώνιας προβολής του y πάνω στο z ,

$$\text{proj}_z(y) = \frac{z^\top y}{z^\top z} z, \quad (\text{A'}.23)$$

ο οποίος προβάλλει ορθογώνια το διάνυσμα y στον υπόχωρο (span) που ορίζεται από το διάνυσμα z . Ένα διάνυσμα y είναι κάθετο ως προς τον υπόχωρο $\text{span}\{z\}$ αν και μόνο αν $\text{proj}_z(y) = \mathbf{0}$. Αντιστρέφοντας το πρόσημο της εξίσωσης και προσθέτοντας το y και στα δύο μέλη της εξίσωσης προκύπτει ότι

$$y = y - \frac{z^\top y}{z^\top z} z \rightarrow y = \left(I_n - \frac{zz^\top}{z^\top z} \right) y \quad (\text{A'}.24)$$

το οποίο επιβεβαιώνει ότι το διάνυσμα y ανήκει στο ορθογώνιο συμπλήρωμα του $\text{span}\{z\}$. Η φιλοσοφία του αλγορίθμου Gram-Schmidt είναι η σειριακή ορθοκανονικοποίηση των στηλών του B όπως φαίνεται και από τον Αλγόριθμο 18. Κάθε στήλη του B ορθοκανονικοποιείται ως προς τις προηγούμενες στήλες μέσω ορθογώνιων προβολών στο ορθογώνιο συμπλήρωμα αυτών.

Αλγόριθμος 18 Η μέθοδος Gram-Schmidt

Είσοδος: $B \in \mathbb{R}^{m \times s}$
Έξοδος: $Q \in \mathbb{R}^{m \times s}$

```

for  $k = 1, \dots, s$  do
    for  $j = 1, \dots, k-1$  do
         $q^{(k)} = b^{(k)} - \text{proj}_{q_j}(b^{(k)})$ 
    end for
     $q^{(k)} = \frac{q^{(k)}}{\|q^{(k)}\|}$ 
end for

```

Πιο συγκεκριμένα, η πρώτη στήλη του B , $b^{(1)}$, διαιρείται με τη νόρμα της ώστε να έχει μέτρο ίσο με ένα και έτσι προκύπτει το πρώτο κανονικοποιημένο διάνυσμα $q^{(1)}$. Στη συνέχεια, η δεύτερη στήλη του B , $b^{(2)}$, ορθοκανονικοποιείται ως προς το διάνυσμα $q^{(1)}$ αφαιρώντας το κοινάτι που βρίσκεται στο χώρο που ορίζεται από το $q^{(1)}$ και διαιρώντας με τη νόρμα του υπολοίπου (το οποίο είναι ορθογώνιο ως προς το $q^{(1)}$). Συνεχίζοντας, η τρίτη στήλη του B , $b^{(3)}$, ορθοκανονικοποιείται ως προς τα διανύσματα $q^{(1)}, q^{(2)}$ ακολουθώντας την ίδια διαδικασία. Ο αλγόριθμος τερματίζει όταν και η s -στή στήλη του B ορθοκανονικοποιηθεί απέναντι στα διανύσματα $q^{(1)}, \dots, q^{(s-1)}$. Μετά το τερματισμό του αλγορίθμου ισχύει $q^{(i)\top} q^{(j)} = 0$, $\forall i, j \leq s$ και $i \neq j$.

Μια εναλλαγή του αλγορίθμου είναι η ορθοκανονικοποίηση ως προς την A νόρμα ενός ΣΘΟ μητρώου A , $\|\cdot\|_A$, ονομάζουμε αυτή τη παραλλαγή ως A -Gram-Schmidt. Τότε, αν y, z είναι δυο τυχαία διανύσματα εισόδου, μετά το πέρας της διαδικασίας A -Gram-Schmidt ισχύει ότι $y^\top Az = 0$. Η τροποποίηση που γίνεται στο κλασικό αλγόριθμο είναι ότι τώρα κάθε ευκλείδειο εσωτερικό γινόμενο αντικαθίσταται από ένα γινόμενο μητρώου-διανύσματος. Σε αντιστοιχία με τα προηγούμενα, ορίζουμε το τελεστή A -ορθογώνιας προβολής ως

$$\text{proj}_{A,z}(y) = \frac{z^\top Ay}{z^\top Az} z \quad (\text{A'}.25)$$

Τότε, ο αλγόριθμος A -ορθοκανονικοποίησης δίνεται στο παρακάτω αλγόριθμο.

Αλγόριθμος 19 Η μέθοδος A-Gram-Schmidt

Είσοδος: A, B
Έξοδος: Q

```

for  $k = 1, \dots, s$  do
    for  $j = 1, \dots, k - 1$  do
         $q^{(k)} = b^{(k)} - proj_{q_j}(Ab^{(k)})$ 
    end for
     $q^{(k)} = \frac{q^{(k)}}{\|q^{(k)}\|}$ 
end for

```

A'.6 Ιδιοτιμές και ιδιοδιανύσματα

Στην ενότητα αυτή εισάγουμε δύο θεμελιώδεις έννοιες της γραμμικής άλγεβρας, τις ιδιοτιμές και τα ιδιοδιανύσματα ενός μητρώου $A \in \mathbb{R}^{n \times n}$. Οι έννοιες αυτές ορίζονται για τετραγωνικά και μόνον μητρώα.

Ορισμός A'.11. Εστω τετραγωνικό μητρώο $A \in \mathbb{R}^{n \times n}$. Τότε ένα μη-μηδενικό διάνυσμα $u \in \mathbb{R}^n$ ονομάζεται ιδιοδιάνυσμα του A αν το διάνυσμα Au είναι παράλληλο στο u για κάποιο βαθμωτό λ , δηλαδή,

$$Au = \lambda u. \quad (\text{A'}.26)$$

Ο βαθμωτός λ ονομάζεται ιδιοτιμή του A και το διάνυσμα u ονομάζεται ιδιοδιάνυσμα που αντιστοιχεί στο λ . Σε συμβολισμό ιδιοζεύγους θα γράφουμε (λ, u) .

Αν $F \in \mathbb{R}^{n \times n}$ ένα αντιστρέψιμο μητρώο, τότε η αντιστοίχηση $A \mapsto F^{-1}AF$ ονομάζεται μετασχηματισμός ομοιότητας (similarity transformation) και δύο τετραγωνικά μητρώα A, B ονομάζονται όμοια αν μεταξύ τους ορίζεται ένας μετασχηματισμός ομοιότητας τέτοιος ώστε για αντιστρέψιμο $F \in \mathbb{R}^{n \times n}$ να ισχύει $B = F^{-1}AF$. Τα όμοια μητρώα μοιράζονται το ίδιο σύνολο ιδιοτιμών και επομένως την ίδια τάξη, ίχνος και ορίζουσα μεταξύ άλλων.

Θεώρημα A'.12. Εστω $F \in \mathbb{R}^{n \times n}$ το οποίο είναι αντιστρέψιμο. Τότε αν τα $A, B \in \mathbb{R}^{n \times n}$ είναι όμοια, έχουν τις ίδιες ιδιοτιμές.

Το να γνωρίζουμε συγκεκριμένα χαρακτηριστικά των ιδιοτιμών ενός μητρώου είναι εξαιρετικά σημαντικό καθώς πέρα από τη φυσική σημασία σε διάφορες εφαρμογές, (πχ. στη μηχανική, κατασκευή ηλεκτρονικών κυκλωμάτων), η γνώση αυτή μπορεί να μας οδηγήσει στην επιλογή της κατάλληλης αφιθμητικής μεθόδου για την επίλυση του προβλήματος το οποίο εξετάζεται, την επίλυση γραμμικών συστημάτων στη προκειμένη περίπτωση (ένα άλλο παράδειγμα είναι το πεδίο κυρτής βελτιστοποίησης όπου μια λύση είναι βέλτιστη αν και μόνο αν το μητρώο Hessian είναι ΣΤΟ). Για ΣΤΟ μητρώα έχουμε το παρακάτω λήμμα.

Λήμμα A'.13. Αν ένα μητρώο A είναι ΣΤΟ, τότε αν με $\lambda_1, \dots, \lambda_n$ συμβολίσουμε τις ιδιοτιμές του σε αύξουσα σειρά, ισχύει ότι $\lambda_i > 0$, $1 \leq i \leq n$. Το αντίστροφο επίσης ισχύει.

Ένα ΣΤΟ μητρώο A έχει n γραμμικά ανεξάρτητα ιδιοδιανύσματα τα οποία ορίζουν μια ορθοκανονική βάση του $\mathbb{R}^{n \times n}$ (χάτι το οποίο αποτελεί ιδιότητα των συμμετρικών μητρώων γενικότερα). Τότε το A μπορεί να εκφραστεί ως

$$A = U \Lambda U^\top \quad (\text{A'}.27)$$

όπου $\Lambda \in \mathbb{R}^{n \times n}$ είναι διαγώνιο μητρώο που στη θέση (i, i) περιέχει την ιδιοτιμή λ_i και $U \in \mathbb{R}^{n \times n}$ είναι ορθοκανονικό μητρώο όπου η στήλη i περιέχει το ιδιοδιάνυσμα u_i που αντιστοιχεί στην ιδιοτιμή λ_i . Η

ανάλυση ενός πυκνού μητρώου ως προς τα ιδιοζεύγη του ($A = U\Lambda U^\top$) έχει πολυπλοκότητα $O(n^3)$.

Οι ιδιοτιμές ενός συμμετρικού μητρώου έχουν ένα συγκεκριμένο “ελάχιστο-μέγιστο” χαρακτηριστικό το οποίο βασίζεται στη τετραγωνική φόρμα του λόγου $x^\top Ax/x^\top x$, το λεγόμενο “ελάχιστο-μέγιστο” θεώρημα των Courant-Fischer όπου

$$\lambda_i(A) = \max_{\dim(S)=i} \min_{0 \neq y \in S} \frac{y^\top A y}{y^\top y} \quad (\text{A'.28})$$

για $i = 1, \dots, n$ και $S \in \mathbb{R}^{n \times s}$ ένα ορθοκανονικό μητρώο. Άλλες γνωστές ιδιότητες των ιδιοτιμών ενός τυχαίου τετραγωνικού μητρώου A είναι οι

$$\text{tr}(A) = \sum_{i=1}^n \lambda_i = \lambda_1 + \dots + \lambda_n, \quad (\text{A'.29})$$

$$\det(A) = \prod_{i=1}^n \lambda_i = \lambda_1 \lambda_2 \dots \lambda_n, \quad (\text{A'.30})$$

όπου $\text{tr}(A)$ αναπαριστά το ίχνος του A .

Τέλος, ένα άνω όριο για τη μέγιστη αλγεβρική τιμή την οποία μπορεί να λάβει οποιαδήποτε ιδιοτιμή λ του A δίνεται από το θεώρημα Gershgorin το οποίο λαμβάνει υπόψιν μόνον τα στοιχεία του A . Εκτός και αν οι απόλυτες τιμές των διαγώνιων στοιχείων του A είναι κατά πολύ μεγαλύτερες από το άθροισμα των απολύτων τιμών των στοιχείων εκτός της κύριας διαγωνίου, το θεώρημα Gershgorin δεν έχει ιδιαίτερη πρακτική σημασία.

Θεώρημα A'.14. Έστω ένα μητρώο $A \in \mathbb{R}^{n \times n}$ και έστω μια οποιαδήποτε ιδιοτιμή του λ . Τότε ισχύει ότι το λ περιέχεται σε ένα δίσκο με κέντρο το α_{ii} και ακτίνα

$$\rho_i = \sum_{j=1, j \neq i}^n |\alpha_{ij}|. \quad (\text{A'.31})$$

Iσοδύναμα,

$$|\lambda - \alpha_{ii}| \leq \sum_{j=1, j \neq i}^n |\alpha_{ij}|. \quad (\text{A'.32})$$

Ο υπολογισμός ιδιοτιμών και ιδιοδιανυσμάτων ενός μητρώου A είναι εκ των πιο προβεβλημένων και δύσκολων προβλημάτων της αριθμητικής γραμμικής αλγεβρας. Όπως θα δούμε και στη συνέχεια, η εύρεση ιδιοδιανυσμάτων μπορεί να είναι αποφασιστικής σημασίας για την (αποδοτική) επίλυση γραμμικών συστημάτων με χρήση επαναληπτικών μεθόδων.

A'.7 Η διάσπαση ιδιαζουσών τιμών και ο ψευδοαντίστροφος

Σε αντίθεση με την ιδιοδιάσπαση τετραγωνικών μητρώων (όταν αυτή υπάρχει), η ανάλυση σε ιδιαζουσες τιμές μπορεί να πραγματοποιηθεί για κάθε μητρώο $A \in \mathbb{R}^{m \times n}$, τετραγωνικό ή μη. Η ανάλυση βασίζεται στη παραγοντοποίηση ιδιαζουσών τιμών (Singular Value Decomposition) η οποία παραγοντοποιεί το A ως

$$A = U\Sigma V^\top$$

με ορθογώνια $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$, δηλαδή,

$$U^\top U = I_m, \quad V^\top V = I_n,$$

και το διαγώνιο μητρώο $\Sigma \in \mathbb{R}^{m \times n}$ περιέχει τις ιδιάζουσες τιμές του A οι οποίες είναι πάντα μη αρνητικές. Συνοπτικά, μέσω της ανάλυσης ιδιάζουσών τιμών το A Μπορεί να γραφεί ως

$$A = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^\top. \quad (\text{A'.33})$$

Η παραγοντοποίηση ιδιάζουσών τιμών προσφέρει αρκετές πληροφορίες για τη δομή του A με τη πιο σημαντική από αυτές να είναι η σχέση

$$r = \text{rank}(A) = \text{nnz}(\Sigma), \quad (\text{A'.34})$$

δηλαδή η τάξη του μητρώου A ισούται το πλήθος των μη-μηδενικών ιδιάζουσών τιμών. Επίσης, η ανάλυση σε ιδιάζουσες τιμές περιέχει ορθογώνιες βάσεις για το χώρο στηλών και το μηδενοχώρο του A . Πιο συγκεκριμένα, αν υποθέσουμε ότι οι ιδιάζουσες τιμές διατάσσονται ως

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = \sigma_{\min\{m,n\}} = 0,$$

τότε έχουμε ότι

$$\text{null}(A) = \text{span}\{v_{r+1}, \dots, v_n\}, \quad \text{range}(A) = \text{span}\{u_1, \dots, u_r\}.$$

Η εύρεση των U , V και Σ απαιτεί την εύρεση των ιδιοζευγών των μητρώων $A^\top A$ και AA^\top . Πιο συγκεκριμένα, μπορούμε να δούμε ότι

$$A^\top A = V\Sigma^\top U^\top U\Sigma V^\top = V\Sigma^\top \Sigma V^\top$$

όπου το διαγώνιο μητρώο $\Sigma^\top \Sigma$ περιέχει τις ιδιοτιμές του $A^\top A$, και οι πρώτες r θέσεις του $\Sigma^\top \Sigma$ περιέχουν τα τετράγωνα των ιδιάζουσών τιμών. Το μητρώο V το οποίο αποτελεί το μητρώο ιδιοδιανυσμάτων είναι ορθογώνιο αφού το μητρώο $A^\top A$ είναι συμμετρικό (μάλιστα είναι συμμετρικό και θετικά ήμι-ορισμένο). Με τον ίδιο τρόπο είναι εύκολο να παρατηρήσουμε ότι το μητρώο U είναι το μητρώο ιδιοδιανυσμάτων του AA^\top .

Δούλευτος της παραγοντοποίησης SVD, μπορούμε να αντιστρέψουμε το A -εφόσον είναι τετραγωνικό και αντιστρέψιμο- ως

$$A^{-1} = V\Sigma^{-1}U^\top.$$

Στη περίπτωση ωστόσο όπου το A δεν είναι τετραγωνικό ή είναι τετραγωνικό και όχι αντιστρέψιμο μπορεί να οριστεί ο ψευδοαντίστροφος (pseudoinverse). Ο ψευδόαντίστροφος ενός μητρώου $A \in \mathbb{R}^{m \times n}$ ορίζεται ως το μητρώο $A^\dagger \in \mathbb{R}^{n \times m}$ για το οποίο ισχύει ότι α) $AA^\dagger A = A$, β) $A^\dagger AA^\dagger = A^\dagger$, γ) $(AA^\dagger)^\top = AA^\dagger$, και δ) $(A^\dagger A)^\top = A^\dagger A$. Ο παραπάνω ψευδοαντίστροφος ονομάζεται και ψευδοαντίστροφος Moore-Penrose. Ο πιο διαδεδομένος τρόπος υπολογισμού του ψευδοαντίστροφου βασίζεται στη παραγοντοποίηση SVD. Πιο συγκεκριμένα, αν $A = U\Sigma V^\top$ είναι η SVD παραγοντοποίηση του A , τότε ο ψευδοαντίστροφος, A^\dagger , ισούται με

$$A^\dagger = V\Sigma^\dagger U^\top$$

όπου ο ψευδοαντίστροφος ενός διαγώνιου μητρώου ορίζεται ως ένα επίσης διαγώνιο μητρώο, με ανεστραμμένες διαστάσεις, και αντεστραμμένες όλες τις μη μηδενικές τιμές.

Η παραγοντοποίηση ιδιαζουσών τιμών βρίσκει πολλές εφαρμογές και εκτός της γραμμικής άλγεβρας, πχ στη μελέτη γραμμικών αντίστροφων προβλημάτων τα οποία προκύπτουν στην επεξεργασία εικόνας, στην επεξεργασία σημάτων, στα συστήματα προτάσεων, στις μετεωρολογικές προβλέψεις και αλλού. Το χόστος της ανάλυσης ιδιαζουσών τιμών είναι, ασυμπτωτικά, $O(mn^2)$.

Βιβλιογραφία

- [1] E. Polizzi and A. Sameh, *A parallel hybrid banded system solver: the SPIKE algorithm*. Parallel Comput., **32**(2), pp. 177-194, 2006, Elsevier Science Publishers B. V.
- [2] C. C. K. Mikkelsen and M. Manguoglu, *Analysis of the truncated SPIKE algorithm*. SIAM J. Matrix Anal. Appl., **30**(4), pp. 1500-1519, 2008.
- [3] I. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*. SIAM J. Matrix Anal. Appl., **20**(4), pp. 889-901, 1999.
- [4] M. Manguoglu, M. Koyutürk, A. Sameh and A. Grama, *Weighted Matrix Ordering and Parallel Banded Preconditioners for Iterative Linear System Solvers*. SIAM J. Scientific Computing, **32**(3), pp. 1201-1216, 2010.
- [5] M. Manguoglu, A. Sameh and O. Schenck, *PSPIKE: A Parallel Hybrid Sparse Linear System Solver*. Proceedings of the 15th International Euro-Par Conference on Parallel Processing, pp. 797-808, 2009.
- [6] D. Carlson and T. L. Markham, *Schur complements of diagonally dominant matrices*. Czechoslovak Mathematical Journal, **29**(2), pp. 246-251, 1979.
- [7] E. Cuthill and J. McKee, *Reducing the bandwidth of sparse symmetric matrices*. ACM Proceedings of the 1969 24th national conference, pp. 157-172, 1969.
- [8] A. George and J. W. Lee, *Computer solution of large sparse positive definite systems*. Prentice Hall Professional Technical Reference, 1981.
- [9] M. Fiedler, *Algebraic connectivity of graphs*. Czechoslovak Math. J., **23**(98), pp. 298-305, 1973.
- [10] A. Sameh and D. J. Kuck, *On Stable Parallel Linear System Solvers*. Journal of the ACM, **25**(1), pp. 81-91, 1978.
- [11] B. Hendrickson and R. Leland, *An improved spectral graph partitioning algorithm for mapping parallel computations*. SIAM J. Scientific Computing, **16**, pp.452-469, 1995.
- [12] E. Polizzi and N. Ben Abdallah, *Subband decomposition approach for the simulation of quantum electron transport in nanostructures*. J. Comput. Phys. **202**(1),pp. 150-180, 2004.
- [13] J. Dongarra, I. Duff, D. Sorensen and H. Van der Vorst, *Numerical linear algebra for high performance computers*. Society for Industrial and Applied Mathematics, 1998.
- [14] H. Van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*. SIAM J. Sci. and Stat. Comput., **13**(2), pp.631-644, 1992.
- [15] M. R. Hestenes, E. Stiefel, *Methods of conjugate gradients for solving linear systems*. Journal of Research of the National Bureau of Standards, **49**(6), pp. 409-436, 1952.
- [16] P. K. Chan, M. D. F. Schlag and J. Y. Zien, *Spectral k-way ratio-cut partitioning and clustering*. IEEE Trans. Computer-Aided Design Integrated Circuits Systems, **13**, pp. 1088-1096, 1994.
- [17] O. Schenk and K. Gartner, *Solving unsymmetric sparse systems of linear equations with pardiso*. Future Generation Computer Systems, **20**(3), pp. 475-487, 2004.
- [18] G. Golub and C. Van Loan, *Matrix Computations (3rd ed)*. Johns Hopkins University Press, 1996.
- [19] I. Duff, R. Grimes and J. Lewis, *Sparse matrix test problems*. ACM Trans. Math. Soft., **15**(1), pp. 1-14, 1989
- [20] T. Chan and W. Lan, *Analysis of projection methods for solving linear systems with multiple right-hand sides*. SIAM J. Sci. Comput., **18**(6), pp. 1698-1721, 1997.
- [21] D. P. O'Leary, *The block conjugate gradient algorithms and related methods*. Linear Algebra and its Applications, **29**, pp. 293-322, 1980.
- [22] J. Meijerink and H. Van der Vorst, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems*. J. Comput. Phys., **44**(1), pp. 134-155, 1981.
- [23] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.
- [24] O. Axelsson, *Iterative solution methods*. Cambridge University Press, 1994.

- [25] C. Bekas, A. Curioni and I. Fedulova, *Low-cost data uncertainty quantification*. Concurrency and Computation: Practice & Experience, **24**(8), pp. 908-920, 2012.
- [26] C. Bekas, E. Kokiopoulou and Y. Saad, *An estimator for the diagonal of a matrix*. Appl. Numer. Math., **57**(11-12), pp. 1214-1229, 2007.
- [27] V. Kalantzis, C. Bekas, A. Curioni and E. Gallopoulos, *Accelerating data uncertainty quantification by solving linear systems with multiple right-hand sides*. Numer. Algor., **62**(4), pp. 637-653, 2013.
- [28] A. El Guennouni and K. Jbilou, *Block and seed BiCGSTAB algorithms for non-symmetric multiple linear systems*. Technical Report, 2000.
- [29] M. Kilmer, E. Miller and C. Rappaport, *QMR-Based projection techniques for the solution of non-hermitian systems with multiple right-hand sides*. SIAM J. Sci. Comput., **23**(3), pp. 761-780, 2001.
- [30] F. Zhang, *Schur complement and its applications*. Springer, 2005.
- [31] M. Manguoglu, *A domain-decomposing parallel sparse linear system solver*. J. Comput. and Appl. Math., **236**(3), pp. 319-325, 2011.
- [32] E. Asplund, *Inverses of matrices $\{\alpha_{ij}\} = 0$ which satisfy α_{ij} for $j > i + p$* . Math. Scand., **7**, pp. 57-60, 1959.
- [33] S. Demko, W. F. Moss and P. W. Smith, *Decay rates for inverses of band matrices*. Math. Comp., **43**(168), pp. 491-499, 1984.
- [34] T. A. Davis and Y. Hu, *The university of Florida sparse matrix collection*. ACM Trans. Math. Soft., **38**(1), pp. 1-25, 2011.
- [35] G. Golub and J. M. Ortega, *Scientific computing: An introduction with parallel computing*. Academic Press Professional, 1993.
- [36] S. Demko, *Inverses of band matrices and local convergence of spline projections*. SIAM J. Numerical Anal., **14**(4), pp. 616-619, 1977.
- [37] Y. Saad, *Numerical methods for large eigenvalue problems*. SIAM, Philadelphia USA, 2011.
- [38] M. Fiedler, *Laplacian of graphs and algebraic connectivity*. Combinatorics and Graph Theory, (25), pp. 57-70, 1989.
- [39] E. Gallopoulos, S. Hatzimihail and V. Kalantzis, *Towards Lightweight Projection Methods for Linear Systems with Multiple Right-Hand Sides*. Midwest Numerical Analysis Days 2011, presentation.
- [40] V. Simoncini and E. Gallopoulos, *Convergence properties of block GMRES and matrix polynomials*. Linear Alg. and its Appl., **247**, pp. 97-119, 1996.
- [41] V. Simoncini and E. Gallopoulos, *A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides*. J. of Comput. and Applied Mathematics, **66**(1), pp. 457-469, 1996.
- [42] M. Gutknecht and T. Schmelzer, *The block grade of a Krylov subspace*. Linear Alg. and its Appl., **430**(11), pp. 174-185, 2009.
- [43] C. F. Smith, A. F. Peterson and R. Mittra, *A conjugate gradient algorithm for the treatment of multiple incident electromagnetic fields*. IEEE Trans. Antennas and Propagation, **37**, pp. 1490-1493, 1989.
- [44] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides* SIAM J. Sci. Comput., **16**, pp. 917-933, 1995.
- [45] Y. Saad, *On the Lanczos method for solving symmetric linear systems with multiple right-hand sides*. Math. Comp., **48**, pp. 651-662, 1987.
- [46] M. Berry, K. Gallivan, E. Gallopoulos, A. Grama, B. Philippe, and Y. Saad, Eds., *High-performance scientific computing, Algorithms and Applications*. Springer-Verlag, 2012.