



Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Προσομοίωση κυκλωμάτων με μεθόδους
υποβιβασμού τάξης μεγέθους και χρήση
εκτεταμένων υποχώρων Krylov

Circuit Simulation with Model Order Reduction
Methods using Extended Krylov Subspaces

Διπλωματική Εργασία
του
Χρήστου Κωνσταντά

Επιβλέποντες: Γεώργιος Σταμούλης
Καθηγητής Π.Θ.
Φώτιος Πλέσσας
Αναπληρωτής Καθηγητής Π.Θ.
Αντώνιος Δαδαλιάρης
Διδακτικό Ερευνητικό Προσωπικό Π.Θ.

15 Νοεμβρίου 2020

Περίληψη

Στα πλαίσια της παρούσας διπλωματικής εργασίας θέσαμε σαν στόχο να μελετήσουμε και να πειραματιστούμε μέσω υλοποίησης πάνω σε μεθόδους για τον υποθιβάσιμο τάξης μοντέλου. Μέθοδοι υποθιβάσιμου της τάξης των μοντέλων χρησιμοποιούνται εκτεταμένα σε εφαρμογές προσομοίωσης κυκλωμάτων όπου τα γραμμικά συστήματα που απορρέουν από πραγματικά κυκλώματα καθιστούν τη συμβατική επίλυση αδύνατη.

Με την εκτόξευση του μεγέθους των κυκλωμάτων, η συμβατική σχεδίαση και μελέτη της συμπεριφοράς των κυκλωμάτων έγινε μη βιώσιμη. Η λύση για τη σχεδίαση κυκλωμάτων εκατομμυρίων στοιχείων προήλθε μέσω της επιστήμης της μοντελοποίησης κυκλωμάτων τα οποία αποτέλεσαν τη βάση για την εμφάνιση εργαλείων προσομοίωσης κυκλωμάτων.

Η βασική μέθοδος επίλυσης γραμμικών συστημάτων μεγάλης κλίμακας είναι η προσέγγιση της λύσης με όσο το δυνατόν μικρότερο σφάλμα. Ανάλογα με την επιλογή του εκάστοτε αλγορίθμου, εμφανίζεται ένας συμβιβασμός μεταξύ της ταχύτητας και της ακρίβειας επίλυσης.

Οι μέθοδοι που παρουσιάζονται στην παρούσα διπλωματική εργασία βασίζονται πάνω στον υποθιβάσιμο τάξης μοντέλου (Model Order Reduction (MOR)). Μια εδραιωμένη μεθοδολογία για τον υποθιβάσιμο τάξης μοντέλου αποτελεί το ταίριασμα στιγμών (moment-matching) καθώς είναι αρκετά απλή στην υλοποίηση και παρουσιάζει πολύ καλά αποτελέσματα στην απόδοση.

Οι παραπάνω μέθοδοι ταιριάσματος στιγμών οι οποίες βασίζονται στους τυπικούς υποχώρους Krylov δεν καταφέρνουν να προσεγγίσουν με καλή ακρίβεια την συμπεριφορά του αρχικού κυκλώματος.

Στα πλαίσια της διπλωματικής εργασίας θα μελετήσουμε τις βασικές μεθόδους ταιριάσματος στιγμών αλλά θα έχουμε και την ευκαιρία να μελετήσουμε μια μέθοδο η οποία εκμεταλλεύεται την ιδιότητα της υπέρθεσης (superposition property) έτσι ώστε να μπορέσει να διαχειριστεί τις πολλαπλές θύρες του κυκλώματος.

Τα αποτελέσματα που παρουσιάζονται μέσω της πρακτικής υλοποίησης μας δίνουν μια πολύ καλή μείωση του λάθους σε σχέση με τις συμβατικές λύσεις που προσεγγίζει το 84%.

Abstract

In this master thesis we set as a goal to study and experiment through implementation into Model Order Reduction (MOR) techniques. The MOR techniques are intensively being used in circuit simulation where the linear systems that derive from real-world circuit systems are impossible to be solved with conventional methods.

With the advent of circuit systems that contain millions of elements, the conventional methods of finding the solution were deprecated since they could not provide a solution fast enough. The way to approach such linear systems came from the science of circuit modeling that drove the appearance of circuit simulation tools.

The standard approach to solve big order linear systems is to not try and find the exact solution but to approximate it. Depending on the selection of the approximation algorithm we face a tradeoff between accuracy and performance.

The methods that we study in this thesis are based on MOR. A well established methodology to achieve a MOR is the moment-matching technique which offers a simple approach with good performance.

Conventional moment matching techniques that are based on Krylov subspaces come short on achieving adequate approximation on the initial circuit behaviour.

In the thesis we will study the well established moment matching techniques but we will also implement and experiment on a new approach that takes advantage of the superposition property in order to handle the great number of terminals.

The results that we received from the experiments that we run suggest an error reduction that reaches up to 84%.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές της παρούσας διπλωματικής εργασίας κ. Σταμούλη Γεώργιο, κ. Φώτιο Πλέσσα και κ.Αντώνιο Δαδαλιάρη για την ανεκτίμητη υποστήριξη τους στην εκπόνηση της εργασίας.

Εξαιρετικά σημαντική ήταν και η συνεισφορά του καλού μου φίλου και υποψηφίου διδάκτορα κ.Χρυσόστομου Χατζηγεωργίου ο οποίος με καθοδήγησε σε όλα τα στάδια της διπλωματικής εργασίας. Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω και στον Διδάκτορα κ.Γεώργιο Φλώρο ο οποίος έδρασε καταλυτικά για την έκβαση της εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω όλη μου την οικογένεια και ιδιαίτερα τη σύντροφο μου Παναγιώτα η οποία έδειξε εξαιρετική υπομονή και στήριξη κατά την πραγματοποίηση της εργασίας.

Μεταπτυχιακή Διατριβή για την απόκτηση του Μεταπτυχιακού διπλώματος Ειδίκευσης «Επιστήμη και Τεχνολογία Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών» του Πανεπιστημίου Θεσσαλίας, στο πλαίσιο του Προγράμματος Μεταπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Κωνσταντίνος Χρήστος, Διπλωματούχος Μηχανικός ΗΥ, Τηλεπικοινωνιών και Δικτύων Πανεπιστημίου Θεσσαλίας

Copyright © Konstantas Christos, 2020

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Περιεχόμενα

1	Εισαγωγή	5
1.1	Προσομοίωση κυκλωμάτων στην επιστήμη της πληροφορικής	5
1.2	Η συμβολή της παρούσας διπλωματικής εργασίας	6
1.3	Διάρθρωση της εργασίας	6
2	Το πρόβλημα της Προσομοίωσης Κυκλωμάτων	8
2.1	Βασικές έννοιες ηλεκτρικών κυκλωμάτων	8
2.1.1	Γραμμικά κυκλώματα	8
2.1.2	Κυκλωματικά στοιχεία	9
2.1.3	Τροποποιημένη ανάλυση κόμβων	9
2.2	Μεταβατική ανάλυση και ανάλυση συνεχούς	11
2.2.1	Ανάλυση συνεχούς	11
2.2.2	Μεταβατική ανάλυση	11
3	Μαθηματικό Υπόβαθρο	14
3.1	Αραιοί Πίνακες στα Γραμμικά Συστήματα	14
3.1.1	Αποθήκευση και απεικόνιση Αραιών Πινάκων	14
3.1.2	Αλγοριθμική επίλυση αραιών πινάκων	15
3.2	Μέθοδοι Παραγοντοποίησης Πινάκων	16
3.3	Χρήση προορυθμιστών με χρήση επαναληπτικών μεθόδων επίλυσης	18
3.4	Υποχώροι Krylov	18
4	Μέθοδοι Υποβιβασμού Τάξης Μοντέλου	19
4.1	Η μέθοδος υποβιβασμού τάξης μοντέλου PRIMA	19
4.2	Η μέθοδος υποβιβασμού τάξης μοντέλου SPRIM	21
5	Μέθοδοι προβολής σε υποχώρους Krylov	23
5.1	Block Arnoldi (BA)	23
5.2	Extended Block Arnoldi (EBA)	24
6	Πειραματική Διαδικασία	25
6.1	Περιγραφή της πειραματικής διαδικασίας	25
6.1.1	Μορφή αρχείων εισόδου	25
6.1.2	Επέκταση υπάρχουσας βιβλιοθήκης CSparse	25
6.1.3	Ανάκτηση πίνακα \mathcal{Q} από τα διανύσματα Householder	26
6.2	Αποτελέσματα Πειραματικής Διαδικασίας	27
6.2.1	Απαραίτητες τροποποιήσεις πλεγμάτων ενέργειας	27
6.2.2	Αξιολόγηση πλεγμάτων ενέργειας αναφοράς	27
6.2.3	Εφαρμογή και σύγκριση με υπάρχουσες μεθόδους	28

6.2.4 Πίνακες συνολικών αποτελεσμάτων και περιγραφή μεταβλητών σύγκρισης	28
A' Ακρωνύμια και συντομογραφίες	33
B' Βιβλιοθήκη CSparse	34
Γ' Μορφή αρχείων εισόδου Matrix Market	39
Γ.1 Περιγραφή μορφής αρχείου	39

Κεφάλαιο 1

Εισαγωγή

1.1 Προσομοίωση κυκλωμάτων στην επιστήμη της πληροφορικής

Καθώς το μέγεθος των ηλεκτρικών κυκλωμάτων έχει φτάσει σε κλίμακα εκατομμυρίων στοιχείων, η μελέτη συμπεριφοράς τους είναι αδύνατη χωρίς τη χρήση ενός λογισμικού προσομοίωσης του κυκλώματος. Τέτοιου είδους λογισμικά, πραγματοποιούν την αντιστοίχιση του κυκλώματος σε μαθηματικό μοντέλο το οποίο αναπαράγει τη συμπεριφορά του.

Ο στόχος των προσομοιωτών είναι μέσω της τροποποιημένης ανάλυσης κόμβων (modified nodal analysis), να δημιουργήσει εξισώσεις με βάση το ηλεκτρικό κύκλωμα, οι οποίες όταν λυθούν να υπολογίσουν τις τάσεις των κόμβων του κυκλώματος καθώς και τα ρεύματα ορισμένων κλάδων. Οι εξισώσεις ανάγονται σε ένα γραμμικό σύστημα το οποίο χρίζει επίλυσης.

Για μικρά κυκλώματα η επίλυση τέτοιων συστημάτων είναι εύκολη και η επιλογή μεθόδων επίλυσης γραμμικών συστημάτων ασήμαντη. Όσο τα γραμμικά συστήματα μεγαλώνουν (για ολοκληρωμένα κυκλώματα αλλά και δίκτυα παροχής ηλεκτρικής ενέργειας) η πολυπλοκότητα τους απαιτεί τη χρήση υψηλών υπολογιστικών πόρων. Απαραίτητη γίνεται και η ορθή επιλογή αλγορίθμων επίλυσης καθώς μη αποδοτικοί αλγόριθμοι ή αλγόριθμοι που δεν εκμεταλλεύονται τα ιδιαίτερα στοιχεία του γραμμικού συστήματος, μπορούν να οδηγήσουν είτε σε χρόνους εκτέλεσης απαγορευτικούς, είτε σε μη σταθερά συστήματα επίλυσης.

Για τον λόγο αυτό, η έρευνα στρέφεται προς την αναζήτηση τεχνικών για την βελτίωση των προσομοιωτών, ώστε να επιτυγχάνεται ταχύτερη επίλυση των γραμμικών συστημάτων.

Σημειώνουμε ότι μία από τις συνέπειες που προκύπτουν από διάφορες μεθόδους ταχύτερης επίλυσης γραμμικών συστημάτων είναι η μείωση της ακρίβειας της λύσης. Συνεπώς, προσπαθούμε να βρούμε τη χρυσή τομή μεταξύ ταχύτητας επίλυσης αλλά χωρίς να χαθεί η ακρίβεια.

1.2 Η συμβολή της παρούσας διπλωματικής εργασίας

Στην παρούσα εργασία, χρησιμοποιούμε τεχνικές οι οποίες αντικαθιστούν τα μεγάλης κλίμακας ηλεκτρικά μοντέλα με κάποια πολύς μικρότερης κλίμακας των οποίων η συμπεριφορά είναι παρόμοια με αυτή του αρχικού μοντέλου.

Η παραπάνω διαδικασία ονομάζεται Υποβιβασμός Τάξης Μοντέλου (Model Order Reduction (MOR)) και θα δοκιμάσουμε την τεχνική Extended Block Arnoldi (EBA) σε σύγκριση με κλασσική μέθοδο Ταιριάσματος Στιγμών (Moment Matching(MM)) με στόχο να επιτύχουμε μικρότερο λάθος λύση μας.

Μέρος της εργασίας αποτελεί και η υλοποίηση της κλασσικής μεθόδου Ταιριάσματος Στιγμών PRIMA καθώς και της μεθόδου Εκτεταμένου Υποχώρου Krylon με τη χρήση του Extended Block Arnoldi αλγορίθμου και η σύγκριση των λαθών που παράγουν κατά την εύρεση λύσης.

1.3 Διάρθρωση της εργασίας

Αρχικά θα μελετήσουμε το πρόβλημα της Προσομοίωσης Κυκλωμάτων, θα περιγράψουμε την μετατροπή των κυκλωματικών στοιχείων σε γραμμικά συστήματα μέσω της Τροποποιημένης Ανάλυσης Κόμβων (Modified Nodal Analysis (MNA)) ενώ θα αναφερθούμε και στην Ανάλυση Συνεχούς.

Στη συνέχεια είναι απαραίτητο να ορίσουμε το Μαθηματικό Υπόβαθρο μέρους του οποίου είναι οι Αραιοί Πίνακες, οι ιδιότητες τους καθώς και η απεικόνιση τους στα υπολογιστικά συστήματα. Θα συνεχίσουμε με μια σύντομη αναφορά στην Αλγοριθμική επίλυση των αραιών πινάκων καθώς και διάφορες μεθόδους παραγοντοποίησης τους. Τέλος, θα γίνει αναφορά στους Υποχώρους Krylon οι οποίοι αποτελούν βάση του αλγορίθμου που θα εξετάσουμε στα πλαίσια της διπλωματικής εργασίας.

Στο επόμενο κεφάλαιο, θα εξετάσουμε τις βασικές μεθόδους για τον υποβιβασμό τάξης μοντέλου (Model Order Reduction (MOR)). Πιο συγκεκριμένα, θα αναφερθούμε στη μέθοδο PRIMA η οποία συνθέτει τον πλέον βασικό αλγόριθμο για τον υποβιβασμό τάξης ενός μοντέλου ενώ θα αναλύσουμε και την μέθοδο SPRIM εξίσου θεμελιώδους στην θεωρία υποβιβασμού τάξης μοντέλου.

Ένα από τα βασικά κεφάλαια στα πλαίσια της παρούσας εργασίας είναι το Κεφάλαιο 5 στο οποίο αναλύουμε τον αλγόριθμο Block Arnoldi (BA) ο οποίος χρησιμοποιείται για την κατασκευή του υποχώρου Krylon. Στη συνέχεια, θα αναλύσουμε την επέκταση του παραπάνω αλγορίθμου τον Extended Block Arnoldi που παράγει τον εκτεταμένο υποχώρο Krylon ο οποίος θα χρησιμοποιηθεί για να ενισχύσουμε την ακρίβεια στη μέθοδο Ταιριάσματος Στιγμών (Moment Matching (MM) και να πετύχουμε καλύτερα αποτελέσματα.

Με την ολοκλήρωση των παραπάνω κεφαλαίων έχουμε ολοκληρώσει τη θεωρητική βάση της παρούσας εργασίας. Στο Κεφάλαιο 6 περνάμε στην Πειραματική Διαδικασία στην οποία αναφερόμαστε στο πρακτικό κομμάτι της υλοποίησης που πραγματοποιήθηκε στα πλαίσια της εργασίας. Αρχικά αναφερόμαστε στα αρχεία εισόδου και την μορφή που έχουν, συνεχίζουμε με την αναφορά στην βιβλιοθήκη CSparse και αναφέρουμε κάποιες από τις επιπλέον συναρτήσεις που χρειάστηκε να κατασκευάσουμε. Σημαντικό κομμάτι της εργασίας στάθηκε και η ανάκτηση του παραγοντοποιημένου πίνακα Q απο

τα διανύσματα Householder που παράγει η μέθοδος QR της βιβλιοθήκης CParse. Τέλος, στο κεφάλαιο αυτό, γίνεται εκτενής αναφορά στα αποτελέσματα που λάβαμε κατά την πειραματική διαδικασία.

Κεφάλαιο 2

Το πρόβλημα της Προσομοίωσης Κυκλωμάτων

2.1 Βασικές έννοιες ηλεκτρικών κυκλωμάτων

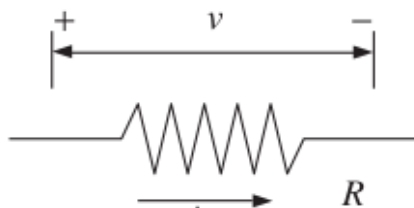
2.1.1 Γραμμικά κυκλώματα

Ο πρώτος ορισμός που πρέπει να δώσουμε είναι αυτός του ηλεκτρικού κυκλώματος. Το ηλεκτρικό κύκλωμα αποτελεί ένα κλειστό δίκτυο συνδεδεμένων μεταξύ τους ηλεκτρικών στοιχείων εντός του οποίου υπάρχουν διαδρομές ρεύματος.

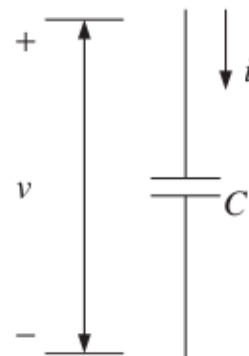
Το κάθε στοιχείο περιγράφεται από μια σχέση ρεύματος και τάσης. Για παράδειγμα, ένας πυκνωτής περιγράφεται από την συνάρτηση (με βάση τον νόμο του Ohm) $u = Ri$.

Οι συναρτήσεις των στοιχείων ενός ηλεκτρικού κυκλώματος καθορίζουν αν το κύκλωμα θα είναι γραμμικό ή μη γραμμικό. Γραμμικό στοιχείο είναι το στοιχείο του οποίου η συνάρτηση δεν περιλαμβάνει συντελεστές στη δύναμη του 2 ή μεγαλύτερη. Όταν ένα ηλεκτρικό κύκλωμα περιλαμβάνει μόνο στοιχεία γραμμικά τότε και το ίδιο το κύκλωμα είναι γραμμικό.

Στα πλαίσια της εργασίας, θα ασχοληθούμε αυστηρά μόνο με γραμμικά κυκλώματα τα οποία ακολουθούν την αρχή της επαλληλίας (superposition principle). Έτσι, η



Σχήμα 2.1: Στοιχείο αντίστασης



Σχήμα 2.2: Στοιχείο πυκνωτή

έξοδος ενός κυκλώματος $F(x)$, όταν εφαρμόζεται ένας γραμμικός συνδυασμός σημάτων $a_1x_1(t) + a_2x_2(t)$ ως είσοδος, θα ισούται με τον γραμμικό συνδυασμό των εξόδων των σημάτων $x_1(t)$ και $x_2(t)$ όταν αυτά εφαρμόζονται ξεχωριστά, δηλαδή:

$$F\{a_1x_1(t) + a_2x_2(t)\} = a_1F\{x_1(t)\} + a_2F\{x_2(t)\} \quad (2.1)$$

2.1.2 Κυκλωματικά στοιχεία

Με τον όρο κυκλωματικό στοιχείο στα πλαίσια της εργασίας μας, περιγράφουμε ηλεκτρικά στοιχεία με δύο τερματικά. Σαν γενικές κατηγορίες των στοιχείων αυτών έχουμε τα παθητικά και τα ενεργά στοιχεία. Τα ενεργά στοιχεία είναι αυτά που παρέχουν ηλεκτρική ενέργεια και δρουν ως πηγές ενός κυκλώματος. Οι πηγές που θα συναντήσουμε είναι ανεξάρτητες και έχουν την ιδιότητα ότι η τάση $v(t)$ ή το ρεύμα $i(t)$ που περνάει από μέσα τους εκφράζεται ως εξής:

- Ανεξάρτητη πηγή τάσης ορίζεται η πηγή της οποίας η τάση είτε είναι σταθερή ή μεταβάλλεται σε συνάρτηση με τον χρόνο $v = f(t)$.
- Ανεξάρτητη πηγή ρεύματος ορίζεται η πηγή της οποίας το ρεύμα είτε είναι σταθερό ή μεταβάλλεται σε συνάρτηση με τον χρόνο $i = f(t)$.

Αντίθετα με τα ενεργά στοιχεία, τα παθητικά στοιχεία καταναλώνουν και δεν παράγουν ενέργεια. Η ενέργεια που καταναλώνουν τα παθητικά στοιχεία είτε θα μετατραπεί σε κάποιου άλλου είδους ενέργειας (για παράδειγμα θερμότητα) είτε θα αποθηκευτεί (σε ενέργεια ηλεκτρικού ή μαγνητικού πεδίου). Τονίζουμε ότι κατά την μετατροπή της ενέργειας, η ισχύς που υπολογίζουμε στην έξοδο δεν θα ενισχυθεί. Τα πιο γνωστά παθητικά στοιχεία είναι το πηνίο, η αντίσταση και ο πυκνωτής.

Αν δούμε τη μορφή της εξίσωσης που χαρακτηρίζει το κάθε στοιχείο μπορούμε να διαφοροποιήσουμε την κατηγοριοποίηση σε δύο ομάδες. Για αριθμό στοιχείων m_1 έχουμε την πρώτη ομάδα που έχει την παρακάτω μορφή εξίσωσης:

$$i_k(t) = g_k u_k(t) + c_k \frac{du_k(t)}{dt} + s_k(t) \quad (2.2)$$

Στην ομάδα αυτή έχουμε αντιστάτες, πυκνωτές και πηγές ρεύματος.

Αντίστοιχα, η δεύτερη ομάδα με αριθμό στοιχείων m_2 εκφράζεται από την παρακάτω μορφή εξίσωσης:

$$u_k(t) = l \frac{di_k(t)}{dt} + s_k(t) \quad (2.3)$$

Στην δεύτερη ομάδα έχουμε πηνία και πηγές τάσης.

2.1.3 Τροποποιημένη ανάλυση κόμβων

Η συμπεριφορά ενός κυκλώματος περιγράφεται από ένα σύνολο εξισώσεων που σχηματίζεται συνδυάζοντας τις συναρτήσεις που προκύπτουν από τους νόμους του *Kirchoff* για την τάση (*Kirchoff's Voltage Law (KVL)*) και το ρεύμα (*Kirchoff's Current Law (KCL)*). Ορίζουμε τους παραπάνω νόμους ως εξής:

1. Ο νόμος ρευμάτων του Kirchoff (KCL) μας λέει ότι το αλγεβρικό άθροισμα των ρευμάτων σε κάθε κόμβο ενός κυκλώματος ισούται με το μηδέν.

$$\sum_{k=1}^n i_k(t) = 0 \Leftrightarrow A \vec{i}(t) = \vec{0} \quad (2.4)$$

2. Ο νόμος τάσεων του Kirchoff(KVL) μας λέει ότι σε κάθε κλειστό βρόγχο ενός κυκλώματος, το άθροισμα των τάσεων (διαφορών δυναμικού) των επιμέρους κλάδων που απαρτίζουν τον βρόγχο ισούται με το μηδέν.

$$\sum_{k=1}^m u_k(t) = 0 \Leftrightarrow \vec{u}(t) = A^T \vec{u}(t) \quad (2.5)$$

Χρησιμοποιώντας τις συναρτήσεις που προκύπτουν από τους παραπάνω νόμους και ακολουθώντας τις κλασσικές μεθοδολογίες επίλυσης γραμμικών συστημάτων μπορούμε να βρούμε τη λύση για πολύ απλά συστήματα. Καταλαβαίνουμε όμως, με την κλασσική επίλυση των συστημάτων δεν μπορούμε να έχουμε κλιμάκωση στο μέγεθος των ηλεκτρικών κυκλωμάτων.

Για τον παραπάνω λόγο, μπορούμε να χρησιμοποιήσουμε την Τροποποιημένη Ανάλυση Κόμβων (Modified Nodal Analysis (MNA)) για να λύσουμε με συστηματικό τρόπο συστήματα πολύ μεγάλης κλίμακας.

Τα βασικά βήματα για την Τροποποιημένη Ανάλυση Κόμβων είναι τα εξής:

1. Γράφουμε με βάση τον Νόμο Ρεύματος του Kirchoff την εξίσωση $Ai = 0$ όπου A είναι ο πίνακας μειωμένης πρόσπτωσης (reduced incidence matrix) και το i είναι διάνυσμα με όλα τα ρεύματα σε κάθε κλάδο
2. Χρησιμοποιώντας τις εξισώσεις που περιγράφουν το κάθε κυκλωματικό στοιχείο, προσπαθούμε να εξαλείψουμε όσο το δυνατόν περισσότερες μεταβλητές ρεύματος με τη βοήθεια του Νόμου Kirchoff για το ρεύμα. Με το βήμα αυτό καταλήγουμε να έχουμε εξισώσεις οι οποίες είναι σε συνάρτηση με την τάση $u = A^T v$.
3. Στο βήμα αυτό, χρησιμοποιούμε τον Νόμο Kirchoff για την τάση (Kirchoff's Voltage Law (KVL)) και αντικαθιστούμε όλα τις τάσεις κλάδων με τάσεις κόμβων στη γείωση.
4. Οι εξισώσεις στοιχείων των οποίων οι μεταβλητές τάσης δεν μπορούσαν να εξαλειφθούν, τις προσθέτουμε σαν επιπλέον εξισώσεις στο MNA σύστημα.

Με βάση τα παραπάνω βήματα καταλήγουμε στο εξής σύστημα [1]:

$$\begin{bmatrix} Y & B \\ C & Z \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} s_v \\ s_j \end{bmatrix} \quad (2.6)$$

Όπου ισχύει η εξίσωση $Z_i + Y_u = s$ για Z και Y πίνακες ενώ s είναι διάνυσμα.

$$\tilde{G} \vec{x}(t) + \tilde{C} \frac{d\vec{x}(t)}{dt} = \vec{e}(t) \Leftrightarrow \tilde{G} \vec{x}(t) + \tilde{C} \dot{\vec{x}}(t) = \vec{e}(t) \quad (2.7)$$

2.2 Μεταβατική ανάλυση και ανάλυση συνεχούς

Τα κυκλώματα στην πλειονότητα τους εκτός από ανιστάτες περιλαμβάνουν και δυναμικά στοιχεία (πυκνωτές και πηνεία) τα οποία ανάλογα με τον χρόνο παρουσιάζουν μεταβολή.

Με βάση την παραπάνω παρατήρηση, η ανάλυση μας μπορεί με βάση τον χρόνο να οριστεί σε δυο κατηγορίες:

- Ανάλυση συνεχούς: Σε αυτή την ανάλυση προσεγγίζουμε μια λύση του κυκλώματος για μια δεδομένη χρονική στιγμή.
- Μεταβατική ανάλυση: Αναφέρεται σε κυκλώματα με δυναμικά στοιχεία στα οποία μας ενδιαφέρει η απόκριση με βάση τον χρόνο.

2.2.1 Ανάλυση συνεχούς

Η παρούσα ανάλυση όπως τονίσαμε αναφέρεται σε συγκεκριμένες χρονικές στιγμές. Συνεπώς, υπολογίζουμε για διάφορα DC σήματα την απόκριση που παρουσιάζει το κύκλωμα. Το χαρακτηριστικό της ανάλυσης είναι ότι λόγω της μη μεταβολής του χρόνου, έχουμε μηδενική χρονική παράγωγο για τις διεγέρσεις και τις αποκρίσεις.

Το σύστημα της εξίσωσης 2.7 γίνεται:

$$\begin{bmatrix} A_1 & GA_1^T \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} \vec{u}_1 \\ \vec{t}_2 \end{bmatrix} = \begin{bmatrix} -A_1 \vec{s}_1 \\ \vec{s}_2 \end{bmatrix} \quad (2.8)$$

Για την παραπάνω ανάλυση είναι σημαντικό να αναφερθούμε στους συμμετρικούς και θετικά ορισμένους πίνακες Symmetric and Positive Definite (SPD) οι οποίοι παρουσιάζουν ιδιαίτερες ιδιότητες που μας βοηθούν για ταχύτερη επίλυση των συστημάτων. Για να προκύψουν τέτοιοι πίνακες είναι απαραίτητο το κύκλωμα να αποτελείται μόνο από αντιστάσεις, πυκνωτές και πηγές ρεύματος. Όλα τα υπόλοιπα στοιχεία πρέπει να απουσιάζουν.

Η συμμετρία symmetric ενός πίνακα A ορίζεται όταν ο πίνακας ισούται με τον αντίστροφο του, δηλαδή ισχύει $A(i, j) = A(j, i)$ για $i \neq j$.

$$A = A^T \Leftrightarrow A \text{ is symmetric, } \forall A \in \mathbb{R}^{n \times n} \quad (2.9)$$

Για την ιδιότητα του θετικά ορισμένου (positive definite) πίνακα A , είναι όταν η τετραγωνική μορφή είναι θετική για κάθε διάνυσμα εκτός του μηδενικού:

$$\vec{x}^T A \vec{x} > 0, \forall \vec{x} \in \mathbb{R}^n, \vec{x} \neq \vec{0} \Leftrightarrow A \text{ is positive definite} \quad (2.10)$$

2.2.2 Μεταβατική ανάλυση

Αντίθετα με την ανάλυση συνεχούς (DC analysis) όπου πραγματοποιείται η ανάλυση για συγκεκριμένες χρονικές στιγμές, στην μεταβατική ανάλυση Transient analysis βασιζόμαστε στον υπολογισμό της χρονικής απόκρισης του κυκλώματος για δεδομένες διεγέρσεις.

Αρχικά αναζητούμε επίλυση για το πρόβλημα των αρχικών τιμών στο οποίο για δεδομένο διάνυσμα διεγέρσεων $e(t)$ και θεωρώντας πως το διάνυσμα των αποκρίσεων $x(t)$ είναι καθορισμένο για μία δεδομένη χρονική στιγμή t_0 ως $x(t_0) = x_0$ (αρχική συνθήκη), έχουμε:

$$\begin{cases} \bar{G}x(t) + \bar{C}\dot{x}(t) = e(t) \\ x(t_0) = x_0 \end{cases}$$

έχει μοναδική λύση $x(t)$ (υπό ορισμένες συνθήκες) σε ένα διάστημα $[t_0, t_f]$ σε διακριτούς χρόνους $t_0 < t_1 < t_2 < \dots < t_m = t_f$.

Οι παρακάτω προσεγγίσεις μας βοηθάνε στο πρόβλημα της εύρεσης των αρχικών τιμών (Initial Value Problem (IVP)):

1. Προσέγγιση Backward Euler (BE) ή Implicit Euler (IE)

Σύμφωνα με αυτήν και για μονοδιάστατη περίπτωση, γράφουμε μια Taylor εκτεταμένη σειρά στο t_{n+1} και έχουμε:

$$x(t) = x(t_{n+1}) + (t - t_{n+1})x'(t_{n+1}) + \frac{(t - t_{n+1})^2}{2}x''(\xi) \quad (2.11)$$

για κάποιο $\xi \text{ μετα} x_{t_{n+1}}$ και t . Τότε, για $t = t_n = t_{n+1} - h$ έχουμε:

$$x(t_n) = x(t_{n+1})hx'(t_{n+1}) + \frac{h^2}{2}x''(\xi) \quad (2.12)$$

Για πολυδιάστατο m χώρο έχουμε:

$$x(t_n) = x_{n+1} + hf(x_{n+1}, t_{n+1}) \Rightarrow x_{n+1} = x_n + hf_{n+1} \quad (2.13)$$

2. Προσέγγιση Trapezoidal Rule(TR)

Για ακόμη καλύτερη προσέγγιση στη λύση μας, χρειάζεται να καταφύγουμε σε μεθόδους υψηλότερης τάξης. Μια τέτοια μέθοδος είναι η Trapezoidal Rule(TR). Για μονοδιάστατη περίπτωση χρησιμοποιούμε μια Taylor εκτεταμένη σειρά στο t_{n+1} και έχουμε:

$$x(t) = x(t_n) + (t - t_n)x'(t_n) + \frac{(t - t_n)^2}{2}x''(t_n) + \frac{t - t_n}{6}x'''(\xi) \quad (2.14)$$

για κάποιο $\xi \text{ μετα} x_{t_n}$ και t_{n+1} το οποίο μπορούμε να παραγοντοποιήσουμε και να λάβουμε:

$$x'(t) = x'(t_n) + (t - t_n)x''(t_n) + \frac{(t - t_n)^2}{2}x'''(\xi) \quad (2.15)$$

γράφοντας τα δύο αποτελέσματα από τις 2.14 και 2.15 και για $t = t_{n+1} = t_n + h$, πολλαπλασιάζουμε την πρώτη ισότητα με 2 και την δεύτερη με $-h$ για να λάβουμε:

$$2x'(t_{n+1}) = 2x'(t_n) + 2hx'(t_n) + h^2x''(t_n) + \frac{h^3}{3}x'''(\xi) \quad (2.16)$$

$$-hx'(t_{n+1}) = -hx'(t_n) - h^2x''(t_n) - \frac{h^3}{2}x'''(\xi) \quad (2.17)$$

Προσθέτοντας τις δύο ισότητες και διαιρώντας με το 2 παίρνουμε :

$$x(t_{n+1}) - x(t_n) = \frac{h}{2}[x'(t_{n+1}) + x'(t_n)] - \frac{h^3}{12}x'''(\xi) \quad (2.18)$$

Για m διαστάσεις, μπορούμε να γράψουμε την παραπάνω ισότητα ως :

$$x(t_{n+1}) - x(t_n) = \frac{h}{2}[x'(t_{n+1}) + x'(t_n)] + O(h^3) \quad (2.19)$$

Καταλήγουμε στον Trapezoidal Rule (TR):

$$x_{n+1} = x_n + \frac{h}{2}[f(x_{n+1}, t_{n+1}) + f(x_n, t_n)] \quad (2.20)$$

ή πιο απλά :

$$x_{n+1} = x_n + \frac{h}{2}(f_{n+1} + f_n) \quad (2.21)$$

Σημειώνουμε ότι μεταξύ των δυο παραπάνω προσεγγίσεων, η μέθοδος Trapezoidal Rule (TR) είναι η πιο συχνά χρησιμοποιούμενη μέθοδος για την προσομοίωση κυκλωμάτων. Επίσης, προσεγγίζει τη λύση καλύτερα και με μεγαλύτερη ακρίβεια την πραγματική λύση x_n για τις διακριτές στιγμές t_n . Ένα ακόμη πλεονέκτημα της *TR* έναντι της *BE* είναι μπορούμε να έχουμε μεγαλύτερα βήματα στο χρόνο και σε δειγματοληψία η οποία είναι μεταβλητή.

Κεφάλαιο 3

Μαθηματικό Υπόβαθρο

3.1 Αραιοί Πίνακες στα Γραμμικά Συστήματα

Με το όρο Αραιοί Πίνακες (Sparse Matrices) αναφερόμαστε σε πίνακες όπου τα περισσότερα στοιχεία τους είναι 0. Για την επιστήμη της πληροφορικής, οι πίνακες αυτοί έχουν ιδιαίτερο ενδιαφέρον καθώς μας δίνουν την δυνατότητα να συμπίεσουμε την πληροφορία που περιέχουν σε μικρότερο χώρο από ότι θα χρειάζονταν ένας πυκνός (dense) πίνακας. Επίσης, μπορούμε να εκμεταλλευτούμε την ιδιότητα που έχουν οι πίνακες αυτοί και να προσεγγίσουμε μια λύση του γραμμικού συστήματος πιο γρήγορα από ότι θα κάναμε σε έναν πυκνό (dense) πίνακα.

3.1.1 Αποθήκευση και απεικόνιση Αραιών Πινάκων

Όπως τονίσαμε, η ιδιότητα των αραιών πινάκων να έχουν τα περισσότερα στοιχεία τους 0, μας δίνει την δυνατότητα να εφαρμόσουμε πιο αποτελεσματικούς τρόπους αποθήκευσης από το να κρατούσαμε όλα τα στοιχεία (μηδενικά και μη μηδενικά) στη μνήμη.

Υπάρχουν διάφοροι τρόποι αποθήκευσης. Στα πλαίσια της διπλωματικής και λόγω της εξωτερικής βιβλιοθήκης που χρησιμοποιούμε, θα σταθούμε σε 2 βασικές μεθόδους απεικόνισης των αραιών πινάκων.

Τρίδυμη Δομή Απεικόνισης

Η δομή αυτή [2] αποτελεί τον πιο απλό τρόπο απεικόνισης των αραιών πινάκων σε ένα υπολογιστικό σύστημα. Για κάθε μη μηδενικό στοιχείο του πίνακα, κρατάμε την γραμμή, τη στήλη και τη τιμή του στοιχείου.

Εκ πρώτης όψεως θα μπορούσε να παρατηρήσει κάποιος ότι χρειαζόμαστε 3 στοιχεία στη μνήμη μας για να απεικονίσουμε ένα μη μηδενικό στοιχείο. Δεν είναι όσο αποδοτικό όσο η απλή απεικόνιση που έχουμε για έναν πυκνό πίνακα όπου κρατάμε όλα τα στοιχεία στη μνήμη.

Το σημαντικό στοιχείο στην απεικόνιση ενός αραιού πίνακα είναι ότι τα στοιχεία που θα χρειαστεί να αποθηκεύσουμε θα είναι πολύ λιγότερα από το μέγιστο αριθμό στοιχείων που μπορεί να έχει ένας πίνακας μεγέθους $n \times m$.

Παράδειγμα απεικόνισης αραιού πίνακα σε Τρίδυμη Δομή

0	0	11.0
0	3	14.0
0	5	16.0
1	1	22.0
1	3	25.0
1	5	26.0
2	2	33.0
2	3	34.0
2	5	36.0
3	0	41.0
3	2	43.0
3	3	44.0
3	5	46.0

Κάθε γραμμή του παραπάνω πίνακα έχει την παρακάτω μορφή:

$I J A(I, J)$

όπου τα I και J είναι οι δείκτες για τον x και y άξονα αντίστοιχα και είναι με βάση το 0.

Δομή της Συμπιεσμένης Στήλης Compressed Column Storage (CCS)

Η δομή αυτή αποτελεί τον πιο συνηθισμένο τρόπο απεικόνισης ενός αραιού πίνακα σε ένα υπολογιστικό σύστημα. Η δομή αυτή αναφέρεται και ως Harwell-Boeing απεικόνιση [3]. Σε σχέση με την Τρίδυμη Δομή Απεικόνισης, προσφέρει τη δυνατότητα να αποθηκεύσουμε μόνο τα μη μηδενικά στοιχεία ενός αραιού πίνακα, αλλά χωρίς να χρειαζόμαστε 3 στοιχεία της μνήμης για την απεικόνιση ενός στοιχείου του πίνακα. Συνολικά για την απεικόνιση ενός αραιού πίνακα A στη μορφή Συμπιεσμένης Στήλης (CCS) χρειαζόμαστε 3 διαφορετικούς μονοδιάστατους πίνακες:

- Πίνακας Τιμών: περιέχει τις τιμές του πίνακα
- Πίνακας Δεικτών των γραμμών: Περιέχει τους δείκτες για όλες τις γραμμές που υπάρχουν μη μηδενικά στοιχεία
- Πίνακας Δεικτών Στιλών: Στον πίνακα αυτό βρίσκονται οι δείκτες για τα στοιχεία στον Πίνακα τιμών που ξεκινάει μια στήλη του πίνακα A .

3.1.2 Αλγοριθμική επίλυση αραιών πινάκων

Η επίλυση αραιών πινάκων μπορεί να πραγματοποιηθεί είτε με την χρήση άμεσων επιλύσεων (direct solvers) είτε με τη χρήση επαναληπτικών μεθόδων.

Οι μέθοδοι που χρησιμοποιούνται για την άμεση επίλυση γραμμικών συστημάτων που χρησιμοποιούν αραιούς πίνακες βασίζονται στην απαλοιφή κατά Gauss.

Τα τελευταία χρόνια έχει εδραιωθεί η επίλυση γραμμικών συστημάτων αραιών πινάκων με τη χρήση επαναληπτικών μεθόδων. Οι μέθοδοι αυτοί επιτυγχάνουν την γρηγορότερη και πιο αποδοτική επίλυση των γραμμικών συστημάτων με τη χρήση των παρακάτω

μεθοδολογιών: Μετάθεση των εξισώσεων και των αγνώστων έτσι ώστε να εγγυηθούμε μια σταθερή LU αποσύνθεση του πίνακα Οργάνωση των υπολογισμών με βάση τη διαθέσιμη υπολογιστική ισχύ του υλικού (hardware) στο οποίο πραγματοποιείται η επίλυση. Τα τελευταία χρόνια έχει εδραιωθεί η χρήση πολυνηματικών υπολογιστικών μονάδων από τη μεριά του επεξεργαστή αλλά και της κάρτας γραφικών (με τη χρήση τεχνολογιών τύπου CUDA, OpenCL) που επιτρέπουν τη μαζική και παράλληλη επίλυση γραμμικών συστημάτων.

3.2 Μέθοδοι Παραγοντοποίησης Πινάκων

Η βασική μέθοδος επίλυσης ενός συστήματος γραμμικών εξισώσεων $Ax = b$ είναι ο πολλαπλασιασμός του δεξιού μέλους με τον αντίστροφο $x = A^{-1}b$ (όταν ο A έχει ορθογώνιες στήλες). Ο περιορισμός της ορθογωνικότητας των στηλών καθώς και η μη αποδοτική επίλυση του συστήματος, καθιστά την απλή αυτή λύση μη βιώσιμη για γραμμικά συστήματα που συναντάμε σε πραγματικά προβλήματα.

Ένα βασικό εργαλείο για την επίλυση γραμμικών συστημάτων (ή για την προσέγγιση μιας λύσης) είναι η παραγοντοποίηση ενός πίνακα. Η βασική αρχή της παραγοντοποίησης είναι η αποσύνθεση ενός πίνακα σε γινόμενο από πίνακες. Ανάλογα με την εκάστοτε μέθοδο παραγοντοποίησης, οι επιμέρους πίνακες έχουν και διαφορετικές ιδιότητες τις οποίες μπορούμε να αξιοποιήσουμε κατά την επίλυση του γραμμικού συστήματος.

Η πιο βασική μέθοδος παραγοντοποίησης είναι η απαλοιφή κατά Gauss. Η απαλοιφή αυτή μπορεί να εφαρμοστεί σε αντιστρέψιμους πίνακες όπου ο A μετατρέπεται σε έναν άνω τριγωνικό U πίνακα ή παραγοντοποιείται σε ένα γινόμενο πινάκων LU όπου L είναι ο κάτω τριγωνικός και U ο άνω τριγωνικός. Η μέθοδος αυτή αν και πολύ διαδεδομένη και χρήσιμη, έχει σαν βασικό μειονέκτημα ότι μπορεί να είναι "ασταθής" για ορισμένα γραμμικά συστήματα. Επίσης, είναι δύσκολο να προσδιορίσουμε το σφάλμα που έχει προκύψει κατά την παραγοντοποίηση.

Τα παραπάνω προβλήματα που εμφανίζει η απαλοιφή κατά Gauss έρχεται να αντιμετωπίσει η Μέθοδος Παραγοντοποίησης QR .

Στα πλαίσια των αλγορίθμων που θα αναλύσουμε, γίνεται χρήση της QR παραγοντοποίησης η οποία μπορεί να επιτευχθεί με τη χρήση διάφορων μεθόδων.

Η QR παραγοντοποίηση χρησιμοποιείται κυρίως για την επίλυση γραμμικών συστημάτων, προβλήματα με ιδιοτιμές και προσέγγιση λύσης με βάση την μέθοδο ελαχίστων τετραγώνων.

Μέθοδος Παραγοντοποίησης QR

Ορίζουμε την Παραγοντοποίηση QR ως εξής:

Definition 3.2.1. QR Για πίνακα A διαστάσεων $m \times n$ όπου $m \geq n$ και έχει γραμμικά ανεξάρτητες στήλες. Τότε υπάρχει ένας μοναδικός $m \times n$ πίνακας Q με $Q^T Q = D$, $D = \text{diag}(d_1, \dots, d_n)$, $d_k > 0$, $k = 1, \dots, n$ και ένας μοναδικός άνω τριγωνικός πίνακας R με $r_k k = 1$, $k = 1, 2, \dots, n$ έτσι ώστε $A = QR$

Η Παραγοντοποίηση QR θα δούμε στην συνέχεια ότι αποτελεί βασικό στοιχείο στον EBA αλγόριθμο καθώς χρησιμοποιείται σε κάθε επαναληπτικό βήμα.

Επίλυση Παραγοντοποίησης XP με χρήση Householder μεθόδου

Η χρήση Householder μεθόδου μας δίνει τη δυνατότητα να επιλύσουμε το ζητούμενο γραμμικό σύστημα πετυχαίνοντας μικρότερες αποκλίσεις στα αποτελέσματα σε κάθε βήμα με αποτέλεσμα να αποφύγουμε την εύρεση Q πίνακα ο οποίος δεν είναι ορθογώνιος [4].

Η ιδέα πίσω απο την μέθοδο Householder είναι να σχεδιάσουμε επαναληπτικά πίνακες Q_1, \dots, Q_n οι οποίοι σταδιακά μετατρέπουν τον A πίνακα σε άνω τριγωνικό πίνακα :

$$A = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \rightarrow Q_1 A = A = \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \rightarrow Q_2 Q_1 A = A = \begin{bmatrix} x & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix} \rightarrow$$

$$Q_3 Q_2 Q_1 A = A = \begin{bmatrix} x & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & 0 \end{bmatrix}$$

όπου x είναι μια μηδενική τιμή. Σημειώνουμε οτι ο Q_k χρησιμοποιεί τις γραμμές $k : m$ και δεν πειράζει καθόλου τις πρώτες $k - 1$ γραμμές και στήλες. Συνεπώς η μορφή του πίνακα Q κατα την επανάληψη k θα είναι η εξής:

$$Q_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & F \end{bmatrix}$$

όπου I_{k-1} είναι ένας $(k - 1) * (k - 1)$ μοναδιαίος πίνακας και ο F έχει την εξής ιδιότητα :

$$Fx = \|x\|e_1$$

έτσι ώστε να εισαγάγει μηδενικά στο κάτω μέρος της στήλης k .

Ο πίνακας F καλείται και Housholder ανακλαστήρας (Housholder reflector)

Για $u = \|x\|e_1 - x$ και $P = \frac{uu^*}{u^*u}$ τότε έχουμε:

$$F = I - 2P = I - 2\frac{uu^*}{u^*u}$$

Για το u επιλέγουμε το εξής:

$$u = x + \text{sign}(x(1))\|x\|e_1$$

Κατά την ολοκλήρωση της μεθόδου, ο πίνακας A περιέχει τον πίνακα R της Παραγοντοποίησης QR και τα διανύσματα u_1, \dots, u_n είναι τα διανύσματα αντανάκλασης τα οποία θα χρησιμοποιηθούν επαναληπτικά για να σχηματίσουμε το $Q * b$ σε ένα γραμμικό σύστημα.

3.3 Χρήση προρυθμιστών με χρήση επαναληπτικών μεθόδων επίλυσης

Κατά την επίλυση μεγάλης κλίμακας γραμμικών συστημάτων με τη χρήση επαναληπτικών μεθόδων προτείνεται η χρήση προρυθμιστών έτσι ώστε να επιτευχθεί γρηγορότερη και πιο αποδοτική λύση.

Υπάρχουν επιλύσεις οι οποίες δεν χρησιμοποιούν προρυθμιστές αλλά δεν είναι ανταγωνιστικοί και βιώσιμοι σε προβλήματα μεγάλης κλίμακας που βρίσκουμε σε πραγματικά προβλήματα. Μια τέτοια επίλυση είναι η Επαναληπτική Υπερχαλάρωση (Successive Overrelaxation (SOR)) [5] μέθοδος η οποία αποτελεί προσέγγιση της Gauss-Seidel (Liebmann). Στη μέθοδο αυτή για γραμμικό σύστημα

$$Ax = b$$

Χωρίζουμε τον πίνακα A σε τρεις(3) υποπίνακες

$$A = D + L + U$$

όπου

- A : περιέχει τις τιμές του πίνακα
- D : Πίνακας που περιλαμβάνει μόνο τη διαγώνιο του A
- L : Κάτω τριγωνικός πίνακας που περιλαμβάνει όλα τα στοιχεία απο τη διαγώνιο και κάτω του πίνακα A
- U : Άνω τριγωνικός πίνακας που περιλαμβάνει όλα τα στοιχεία απο τη διαγώνιο και πάνω

Για την επίλυση του συστήματος χρησιμοποιούμε επαναληπτική μέθοδο που αξιοποιεί τιμές του x από προηγούμενο βήμα για την εύρεση του x στο τρέχον βήμα.

Οι πιο σύγχρονες μέθοδοι που χρησιμοποιούνται για την επίλυση μεγάλης κλίμακας γραμμικά συστήματα χρησιμοποιούν προρυθμιστές υποχώρου Krylov.

3.4 Υποχώροι Krylov

Οι υποχώροι Krylov εμφανίζονται σε πολλές επαναληπτικές μεθόδους για την επίλυση γραμμικών συστημάτων αραιών πινάκων.

Μπορούμε να ορίσουμε τους υποχώρους Krylov ως εξής:

Definition 3.4.1. Krylov Για μη-αντιστρέψιμο πίνακα $A \in C^{N \times N}$ και $y \neq 0 \in C^N$ ο N -οστος Krylov (υπο)χώρος $K_n(A, y)$ που παράγεται απο τον A στον y είναι

$$K_n := K_n(A, y) := \text{span}(y, Ay, \dots, A^{n-1}y)$$

Χαρακτηριστικά που μπορούμε να παρατηρήσουμε στον παραπάνω ορισμό είναι τα παρακάτω:

$$K_1 \subseteq K_2 \subseteq K_3 \subseteq \dots$$

Οι διαστάσεις αυξάνονται κατά μια σε κάθε βήμα.

Κεφάλαιο 4

Μέθοδοι Υποβιβασμού Τάξης Μοντέλου

Όπως τονίσαμε και στην εισαγωγή της διπλωματικής εργασίας (1) τα κυκλώματα που εμφανίζονται σε πραγματικές εφαρμογές έχουν φτάσει τα εκατομμύρια στοιχεία. Τα γραμμικά συστήματα που προκύπτουν από τέτοιας κλίμακας κυκλώματα καθιστούν την συμβατική επίλυση αδύνατη.

Μια από τις πλέον εδραιωμένες τεχνικές για τη μείωση της πολυπλοκότητας των γραμμικών κυκλωμάτων οι οποίες αποσκοπούν στη μείωση των απαιτούμενων υπολογισμών ή στην απλοποίηση των μαθηματικών μοντέλων είναι οι μέθοδοι υποβιβασμού τάξης μοντέλου. (Model Order Reduction (MOR)).

Οι μέθοδοι υποβιβασμού τάξης μοντέλου που θα εξετάσουμε στα πλαίσια της παρούσας διπλωματικής εργασίας βασίζονται στους υποχώρους Krylov 3.4. Τέτοιες μέθοδοι αποτελούν οι Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA) [6] και Structure-Preserving Reduced-order Interconnect Macromodeling (SPRIM) [7]

4.1 Η μέθοδος υποβιβασμού τάξης μοντέλου PRIMA

Ο πλέον βασικός αλγόριθμος για υποβιβασμό τάξης μοντέλου MOR είναι ο Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA). Μέρος του αλγορίθμου είναι και η μέθοδος Block Arnoldi που θα εξετάσουμε και υλοποιήσαμε στα πλαίσια της διπλωματικής εργασίας. Ο Block Arnoldi αλγόριθμος χρησιμοποιείται για την παραγωγή ορθοκανονικής βάσης για έναν πίνακα μετασχηματισμού.

Πρίν περιγράψουμε τον αλγόριθμο PRIMA ορίζουμε τους πίνακες οι οποίοι προκύπτουν από τη μέθοδο MNA:

$$\begin{aligned} Cx_n &= -Gx_n + Bv_p \\ i_p &= L^T x_n \end{aligned} \tag{4.1}$$

όπου τα διανύσματα i_p και v_p δηλώνουν τις πηγές ρεύματος και τάσης αντίστοιχα

$$G = \begin{bmatrix} N & E \\ -E^T & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} \mathcal{Q} & 0 \\ 0 & H \end{bmatrix}$$

$$x_n = \begin{bmatrix} v \\ i \end{bmatrix}$$

Οι $n * n$ πίνακες G και C αντιπροσωπεύουν τους πίνακες αγωγιμότητας και επιδεκτικότητας. Οι N , \mathcal{Q} και H πίνακες περιλαμβάνουν τις αντιστάσεις, πυκνωτές και πηνία αντίστοιχα. Ο πίνακας E περιλαμβάνει μονάδες, αρνητικές μονάδες και μηδενικά που αντιστοιχούν στις μεταβλητές ρεύματος των νόμων του Kirchhoff (KCL). Στην περίπτωση που οι N -θύρες του κυκλώματος περιλαμβάνουν μόνο θετικά γραμμικά στοιχεία, \mathcal{Q} , H και N είναι συμμετρικοί, θετικά δηλωμένοι πίνακες (Symmetric Positive Definite (SPD)). Με βάση το παραπάνω ο C είναι επίσης SPD.

Συνεπώς, επιστρέφοντας στην ισότητα 4.1, δηλώνουμε:

$$A = -G^{-1}C \quad (4.2)$$

$$R = G^{-1}B \quad (4.3)$$

Μετά από $\left\lfloor \frac{q}{N} \right\rfloor + 1$ επαναλήψεις (η επιπλέον επανάληψη είναι απαραίτητη μόνο όταν $\frac{q}{N}$ δεν είναι ακέραιος) του PRIMA, ο $n * q$ πίνακας X είναι ο εξής:

$$\text{colsp}(X) = K_r(A, R, \left\lfloor \frac{q}{N} \right\rfloor)$$

$$X^T X = I_q$$

Στον βασικό αλγόριθμο PRIMA χρησιμοποιείται μια παραλλαγή της μεθόδου Block Arnoldi η οποία μας δίνει το εξής:

$$(X^T C X) \tilde{x}_q = -(X^T G X) \tilde{x}_q + (X^T B) u_p$$

$$i_p = (L^T X) \tilde{x}_q$$

Το υποβιβασμένης τάξης μοντέλο των MNA πινάκων είναι:

$$\tilde{C} = X^T C X$$

$$\tilde{G} = X^T G X$$

$$\tilde{B} = X^T B$$

$$\tilde{L} = X^T L$$

Μετά τον υποβιβασμό του μοντέλου του πίνακα $Y(s)$ στο οποίο αναφερόμαστε ως $\hat{Y}(s)$ είναι πλέον:

$$\hat{Y}(s) = \tilde{L}^T (\tilde{G} + s\tilde{C})^{-1} \tilde{B}$$

Τυπικά, το μέγεθος των πινάκων \tilde{G} και \tilde{C} είναι μικρό με αποτέλεσμα αν είναι εύκολο να βρούμε τους πόλους και τα μηδενικά του $\hat{Y}(s)$.

4.2 Η μέθοδος υποβιβασμού τάξης μοντέλου SPRIM

Για την περιγραφή του αλγορίθμου SPRIM θα χρειαστεί να ορίσουμε τους εξής πίνακες:

$$G = \begin{bmatrix} G_1 & G_2^T \\ -G_2 & 0 \end{bmatrix}$$
$$C = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}$$
$$B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$$

Στο πρώτο βήμα του αλγορίθμου, δημιουργούμε τους εξής πίνακες A και R ως εξής:

$$A = (G + sC)^{-1}C$$

$$R = (G + sC)^{-1}B$$

Στη συνέχεια και ένα από τα βασικά σημεία του αλγορίθμου είναι η επιλογή της μεθόδου για την παραγωγή του υπο-χώρου Krylov. Το σημείο αυτό είναι που εισέρχεται και η μελέτη που πραγματοποιήσαμε στα πλαίσια της διπλωματικής εργασίας για την χρήση του Εκτεταμένου Block Arnoldi αλγορίθμου 5.2.

Με τη χρήση του επιθυμητού αλγορίθμου λαμβάνουμε το εξής αποτέλεσμα:

$$\text{span}(V_n) = K_n(A, R)$$

Συνεχίζουμε με το επόμενο βήμα να είναι ο κατακερματισμός του πίνακα V_n σε κομμάτια (blocks):

$$V_n = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

ο οποίος μετατρέπεται στην εξής μορφή:

$$\tilde{V}_n = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}$$

Υπολογίζουμε τους νέους πίνακες μέσω της ελαχιστοποίησης:

$$\tilde{G}_1 = V_q^T G_1 V_1$$

$$\tilde{G}_2 = V_T^2 G_2 V_1$$

$$\tilde{C}_1 = V_T^1 C_1 V_1$$

$$\tilde{C}_2 = V_T^2 C_2 V_2$$

$$\tilde{B}_1 = V_T^1 B_1$$

και με βάση του πίνακες αυτούς έχουμε:

$$\widetilde{G}_n = \begin{bmatrix} \widetilde{G}_1 & \widetilde{G}_2^T \\ -\widetilde{G}_2 & 0 \end{bmatrix}$$

$$\widetilde{C}_n = \begin{bmatrix} \widetilde{C}_1 & 0 \\ 0 & \widetilde{C}_2 \end{bmatrix}$$

$$\widetilde{B}_n = \begin{bmatrix} \widetilde{B}_1 \\ 0 \end{bmatrix}$$

Τέλος, από τους νέους πίνακες υπολογίζεται η συνάρτηση μεταφοράς του συστήματος με τον εξής τρόπο :

$$\tilde{Z}_n(s) = \tilde{B}_n^T (\tilde{G}_n + s\tilde{C}_n)^{-1} \tilde{B}_n \quad (4.4)$$

Κεφάλαιο 5

Μέθοδοι προβολής σε υποχώρους Krylov

5.1 Block Arnoldi (BA)

Ο αλγόριθμος Block Arnoldi (BA) θα εφαρμοστεί στους πίνακες F, G όπου $F \in \mathbb{R}^{n \times n}$ και $G \in \mathbb{R}^{n \times s}$. Ο υποχώρος Krylov δίνεται από:

$$K_m(F, G) = \text{Range}([G, FG, F^2G, \dots, F^{m-1}G]) \quad (5.1)$$

Σημειώνουμε ότι ο υποχώρος $K_m(F, G)$ είναι το άθροισμα από s υποχώρους Krylov

$$K_m(F, G) = \sum_{i=1}^s K_m(F, G^{(i)}) \quad (5.2)$$

όπου $K_m(F, G^{(i)})$ είναι ο υποχώρος Krylov που έχει παραχθεί από τον F και την i -στη στήλη $G^{(i)}$ από τον $n \times s$ πίνακα G . Στη συνέχεια ορίζουμε τον αλγόριθμο BA [8–10]

Algorithm 1: Αλγόριθμος Block-Arnoldi

Είσοδοι: F πίνακας $n \times n$ G πίνακας $n \times s$ και m ακέραιος

Παραγοντοποίηση QR του $G = V_1 R_1$

for $j \leftarrow 1$ **to** $m - 1$ **do**

$W_j = FV_j$ **for** $i \leftarrow 1$ **to** $j - 1$ **do**

$H_{i,j} = V_i^T W_j$

$W_j = W_j - V_j H_{i,j}$

end

 Πραγματοποιούμε παραγοντοποίηση QR του $Q_j R_j = Q_j$

$H_{j+1,j} = R_j$

end

5.2 Extended Block Arnoldi (EBA)

Η επαναληπτική μέθοδος της Extended Block Arnoldi (EBA) χρησιμοποιείται έτσι ώστε να κατασκευαστεί ένας ορθοκανονικός πίνακας

$$V_m = [V_1, V_2, \dots, V_m]$$

ο οποίος αποτελεί βάση του υποχώρου Krylov $K_m(A, V)$. Ο περιορισμός του πίνακα A στον υποχώρο Krylov $K_m(A, V)$ δίνεται από το

$$T_m = V_m^T A V_m$$

Ο αλγόριθμος EBA ορίζεται ως εξής [8, 11–14]:

Algorithm 2: Αλγόριθμος Extended Block-Arnoldi (EBA)

Είσοδοι: A πίνακας $n \times n$, V πίνακας $n \times r$ και m ακέραιος

Παραγοντοποίηση QR του $[V, A^{-1}V]$

Θέτουμε $V_0 = []$

for $j \leftarrow 1$ **to** $m - 1$ **do**

 Θέτουμε $V_j^{(1)}$ τις πρώτες r στήλες του V_j

 Θέτουμε $V_j^{(2)}$ τις επόμενες r στήλες του V_j

$V_j = [V_{j-1}, V_j]$

$\hat{V}_{j+1} = [AV_j^{(1)}, A^{-1}V_j^{(2)}]$

 Ορθογωνοποιούμε \hat{V}_{j+1} με βάση τον V_j για να πάρουμε τον V_{j+1}

for $i \leftarrow 1$ **to** $j - 1$ **do**

$H_{ij} = V_i^T \hat{V}_{j+1}$

$\hat{V}_{j+1} = \hat{V}_{j+1} - V_j H_{ij}$

end

 Πραγματοποιούμε παραγοντοποίηση QR του \hat{V}_{j+1} : $\hat{V}_{j+1} = V_{j+1} H_{j+1,j}$

end

Κεφάλαιο 6

Πειραματική Διαδικασία

6.1 Περιγραφή της πειραματικής διαδικασίας

Στόχος της πειραματικής διαδικασίας είναι η εφαρμογή των μεθόδων υποβιβασμού τάξης μεγέθους και η χρήση εκτεταμένων υποχώρων Krylov σε πραγματικά κυκλώματα μεσαίας και μεγάλης κλίμακας.

Μέσω της τροποποιημένης ανάλυσης κόμβων μετατρέπουμε τα δίκτυα τροφοδοσίας ολοκληρωμένων συστημάτων σε γραμμικά συστήματα στα οποία θα εφαρμόσουμε τον Extended Block Arnoldi (EBA) αλγόριθμο για να πετύχουμε υποβιβασμό της τάξης μεγέθους και τελικά επίλυση των συστημάτων.

Τα πραγματικά κυκλώματα που θα μελετήσουμε είναι τα δίκτυα τροφοδοσίας των ολοκληρωμένων κυκλωμάτων της IBM.

Σαν οδηγό στην υλοποίηση μας χρησιμοποιούμε το εργαλείο MATLAB το οποίο είναι σχεδιασμένο για γρήγορη και άμεση υλοποίηση (χρησιμοποιώντας συναρτήσεις συστήματος για τις περισσότερες πράξεις μεταξύ πινάκων) αλγορίθμων.

Με τη βοήθεια του παραπάνω εργαλείου μπορούμε να αξιοποιήσουμε ότι η υλοποίηση μας είναι ορθή σε όλα τα στάδια της.

6.1.1 Μορφή αρχείων εισόδου

Τα αρχεία εισόδου είναι σε μορφή Matrix Market. Περιγραφή της παραπάνω μορφής μπορείτε να βρείτε στο Παράρτημα Γ'. Όπως μπορούμε να παρατηρήσουμε στη μορφή του αρχείου, ο πίνακας ξεκινάει από τη θέση 1,1. Στις δομές δεδομένων που χρησιμοποιούμε (όπως είναι η Τρίδυμη Δομή Απεικόνισης 3.1.1 και Δομή της Συμπιεσμένης Στήλης 3.1.1) στην υλοποίηση μας, οι πίνακες έχουν σαν αρχικές συντεταγμένες το (0,0). Συνεπώς, μια βασική αλλαγή που πρέπει να πραγματοποιήσουμε είναι να αφαιρούμε -1 από κάθε συντεταγμένη γραμμής και στήλης έτσι ώστε να έχουμε τον ίδιο πίνακα.

6.1.2 Επέκταση υπάρχουσας βιβλιοθήκης CSparse

Η βιβλιοθήκη CSparse που χρησιμοποιήσαμε κατά την υλοποίηση μας περιλαμβάνει ένα πλήθος συναρτήσεων για πράξεις για διαχείριση και πραγματοποίηση πράξεων

μεταξύ αραιών πινάκων (Παράρτημα Β'). Για την υλοποίηση του αλγορίθμου EBA (Κεφάλαιο 5.2) χρειάστηκε να υλοποιήσουμε επιπλέον συναρτήσεις τις πιο σημαντικές από τις οποίες θα αναφέρουμε στη συνέχεια.

6.1.3 Ανάκτηση πίνακα Q από τα διανύσματα Householder

Η βιβλιοθήκη CSparse κατά την παραγοντοποίηση QR χρησιμοποιεί τη μέθοδο Householder (Κεφάλαιο 3.2.1). Η μέθοδος αυτή αν και εξαιρετικά αποτελεσματική για επίλυση προβλημάτων όπως των Ελαχίστων Τετραγώνων, στη δική μας περίπτωση όπου χρειαζόμαστε τον άνω τριγωνικό πίνακα Q , τα διανύσματα απεικόνισης (reflectors) δεν μπορούν να χρησιμοποιηθούν στην επίλυση μας. Συνεπώς, θα χρειαστεί να πραγματοποιήσουμε ένα επιπλέον βήμα όπου από τα διανύσματα απεικόνισης (reflectors) θα χρειαστεί να ανακτήσουμε τον πίνακα Q .

Μέθοδοι Ανάκτησης Πίνακα Παραγόντων Q

Στις περιπτώσεις όπου χρειάζεται η ανάκτηση του πίνακα παραγόντων Q μπορούμε να ακολουθήσουμε τις παρακάτω μεθόδους [15]:

- **Εμπρόσθια Συσσώρευση** (Forward Accumulation) Για την ανάκτηση του πίνακα Q με την παραπάνω μέθοδο ακολουθούμε τον εξής αλγόριθμο:

Algorithm 3: Εμπρόσθια Συσσώρευση(Forward Accumulation)

```

 $Q = I_n$ 
for  $j \leftarrow 1$  to  $r - 1$  do
  |  $Q = QQ_j$ 
end

```

- **Οπίσθια Συσσώρευση** (Backward Accumulation)

Algorithm 4: Οπίσθια Συσσώρευση(Backwards Accumulation)

```

 $Q = I_n$ 
for  $j \leftarrow r$  to  $1 - 1$  do
  |  $Q = Q_j Q$ 
end

```

Μπορεί εύκολα να αποδειχθεί ότι το $(j - 1) \times (j - 1)$ κομμάτι του Q_j είναι ο ταυτοτικός πίνακας οπότε κατά το ξεκίνημα της Οπίσθιας Συσσώρευσης, ο Q κατά κύριο λόγο είναι ο ταυτοτικός πίνακας και σε κάθε βήμα γίνεται όλο και πιο πλήρης όσο ο αλγόριθμος εξελίσσεται. Το συγκεκριμένο χαρακτηριστικό του αλγορίθμου μπορεί να αξιοποιηθεί έτσι ώστε να μειώσουμε τον αριθμό των πράξεων που χρειάζονται. Σε αντίθεση, στην Εμπρόσθια Συσσώρευση, μετά το πρώτο βήμα, ο πίνακας είναι ήδη πλήρης. Με βάση τα παραπάνω, ο αλγόριθμος μπορεί να μετατραπεί:

Algorithm 5: Βελτιωμένη Οπίσθια Συσσώρευση(Backwards Accumulation)

$$Q = I_n$$

for $j \leftarrow r$ **to** $1 - 1$ **do**

$$\left| \begin{array}{l} v(j : n) = \begin{bmatrix} 1 \\ A(j + 1 : n, j) \end{bmatrix} \\ Q(j : n, j : n) = (I - \beta_j v(j : n) v(j : n)^T) Q(j : n, j : n) \end{array} \right.$$

end

Στα πλαίσια της διπλωματικής εργασίας, εφαρμόσαμε τον αλγόριθμο Βελτιωμένης Οπίσθιας Συσσώρευσης.

6.2 Αποτελέσματα Πειραματικής Διαδικασίας

Για την αξιολόγηση της πειραματικής διαδικασίας διαλέξαμε τα διαθέσιμα πλέγματα ενέργειας power grids της IBM [16]. Τα χαρακτηριστικά των παραπάνω πλεγμάτων ενέργειας μπορούν να συνοψιστούν στον παρακάτω πίνακα 6.1:

Αρχείο Εισόδου	Διαστάσεις	#είσοδων
ibmpg1	44946	600
ibmpg2	127568	500
ibmpg3	852539	800
ibmpg4	954545	600
ibmpg5	1618397	600
ibmpg6	2506733	1000
ibmpg1t	54265	400
ibmpg2t	164897	800

Πίνακας 6.1: Πίνακας περιγραφής διαστάσεων και #είσοδων για τα πλέγματα ενέργειας IBM

6.2.1 Απαραίτητες τροποποιήσεις πλεγμάτων ενέργειας

Στον παραπάνω πίνακα 6.1 παρατηρούμε στις 2 τελευταίες γραμμές τα πλέγματα ενέργειας *ibmpg1t*, *ibmpg2t*. Τα 2 αυτά πλέγματα ενέργειας έχουν το γράμμα *t* στο τέλος το οποίο υποδηλώνει ότι αναφέρονται στην μεταβατική transient ανάλυση. Τα πλέγματα ενέργειας που αναφέρονται στην μεταβατική ανάλυση, χρειάζεται να παρέχουμε έναν πίνακα με στοιχεία αποθήκευσης ενέργειας (πυκνωτές και πηνία). Από την άλλη μεριά, για να πραγματοποιήσουμε μεταβατική ανάλυση στα πλέγματα ενέργειας συνεχούς DC ροής, στην προκειμένη περίπτωση *ibmpg1* - *ibmpg6* έπρεπε να προσθέσουμε έναν διαγώνιο πίνακα με πυκνωτικά στοιχεία τα οποία είχαν τυχαίες τιμές της τάξης του *picofarad*. Η παραπάνω προσθήκη είναι τυπική για τέτοιου είδους πλέγματα ενέργειας.

6.2.2 Αξιολόγηση πλεγμάτων ενέργειας αναφοράς

Για να καταφέρουμε να αξιολογήσουμε την μεθοδολογία μας σε μοναδικά πλέγματα ενέργειας αναφοράς, επιβάλαμε στον πίνακα πυκνωτικών στοιχείων των πλεγμάτων

ibmpg2 και ibmpg4 να έχουν τουλάχιστον έναν κόμβο ο οποίος δεν είχε ένωση με πυκνωτικό στοιχείο. Τα παραπάνω πλέγματα αναφοράς μαζί με τα ibmpg1t και ibmpg2t αντιπροσωπεύονται ως μοναδικά μοντέλα περιγραφής του [16].

6.2.3 Εφαρμογή και σύγκριση με υπάρχουσες μεθόδους

Όπως έχουμε τονίσει και σε προηγούμενα κεφάλαια, στα πλαίσια της διπλωματικής εργασίας, εφαρμόσαμε τον αλγόριθμο Extended Block Arnoldi (EBA) τον οποίο περιγράψαμε στο 5.2 και τον συγκρίναμε με μια βασική υλοποίηση Αντιστοίχισης Στιγμών (Moment-Matching (MM) η οποία επίσης χρησιμοποιούσε την αρχή της επαλληλίας (superposition property). Τα μοντέλα μειωμένης τάξης (Reduced-Order Models (ROMs)) αξιολογήθηκαν στο εύρος συχνοτήτων $[10^0, 10^{12}]$ συναρτήσει της ακρίβειας για τη δεδομένη τάξη του μοντέλου.

Για την αξιολόγηση μας επιλέξαμε τον κατάλληλο αριθμό στιγμών για το κάθε πλέγμα ενέργειας έτσι ώστε και η μέθοδος EBA που εφαρμόσαμε αλλά και η κλασική MM μέθοδος να καταλήγουν στην ίδια τάξη μοντέλου.

6.2.4 Πίνακες συνολικών αποτελεσμάτων και περιγραφή μεταβλητών σύγκρισης

Στη συνέχεια παραθέτουμε πίνακες με τα αποτελέσματα της ανάλυσης μας. Οι πίνακες περιλαμβάνουν τα δεδομένα για την εκτέλεση της βασικής μεθόδου Αντιστοίχισης Στιγμών (Moment-Matching (MM), της εκτέλεσης της μεθόδου Εκτεταμένου Υποχώρου Krylov (EBA) καθώς και έναν πίνακα με τη σύγκριση των λαθών για κάθε πλέγμα ενέργειας.

Ξεκινάμε παραθέτοντας τα αποτελέσματα κατά την εφαρμογή της βασικής μεθόδου Αντιστοίχισης Στιγμών (Moment-Matching (MM) 6.2.

Αρχείο Εισόδου	#στιγμών	Μέγιστο Λάθος
ibmpg1	2	0.037
ibmpg2	4	0.233
ibmpg3	2	0.253
ibmpg4	4	0.233
ibmpg5	2	0.242
ibmpg6	6	0.161
ibmpg1t	2	4.767
ibmpg2t	4	0.785

Πίνακας 6.2: Πίνακας αποτελεσμάτων του μέγιστου λάθους με την εφαρμογή βασικής μεθόδου Αντιστοίχισης Στιγμών για τα πλέγματα ενέργειας IBM

Συνεχίζουμε παραθέτοντας τα αποτελέσματα κατά την εφαρμογή της μεθόδου Εκτεταμένου Υποχώρου Krylov (EBA) 6.3.

Τέλος παραθέτουμε τα αποτελέσματα της σύγκρισης μεταξύ των 2 παραπάνω μεθόδων και το ποσοστό μείωσης λάθους της μεθόδου Εκτεταμένου Υποχώρου Krylov (EBA) σε σχέση με τη βασική μέθοδο 6.4:

Αρχείο Εισόδου	#στιγμών	Μέγιστο Λάθος
ibmpg1	1	0.014
ibmpg2	2	0.131
ibmpg3	1	0.146
ibmpg4	2	0.038
ibmpg5	1	0.063
ibmpg6	3	0.130
ibmpg1t	1	1.814
ibmpg2t	2	0.411

Πίνακας 6.3: Πίνακας αποτελεσμάτων του μέγιστου λάθους με την εφαρμογή μεθόδου Εκτεταμένου Υποχώρου Krylov (EBA) για τα πλέγματα ενέργειας IBM

Αρχείο Εισόδου	Μέγιστο Λάθος (MM)	Μέγιστο Λάθος (EKS-MM)	Ποσοστό Μείωσης Λάθους
ibmpg1	0.037	0.014	62.16%
ibmpg2	0.233	0.131	43.78%
ibmpg3	0.253	0.146	42.29%
ibmpg4	0.233	0.038	83.69%
ibmpg5	0.242	0.063	73.97%
ibmpg6	0.161	0.130	19.25%
ibmpg1t	4.767	1.814	61.95%
ibmpg2t	0.785	0.411	47.64%

Πίνακας 6.4: Πίνακας αποτελεσμάτων σύγκρισης του μέγιστου λάθους μεταξύ των μεθόδων για τα πλέγματα ενέργειας IBM

Για τους πίνακες 6.3, 6.4 το Μέγιστο Λάθος αναφέρεται στο λάθος μεταξύ των άπειρων νορμών των συναρτήσεων μεταφοράς. Στην προκειμένη περίπτωση, για τα αρχεία εισόδου έχουμε $\|H(s) - \tilde{H}(s)\|_{\infty}$

Βλέπουμε επίσης στον πίνακα 6.4 ότι η ποσοστιαία μείωση του λάθους κυμαίνεται από 19.25% έως 83.69%.

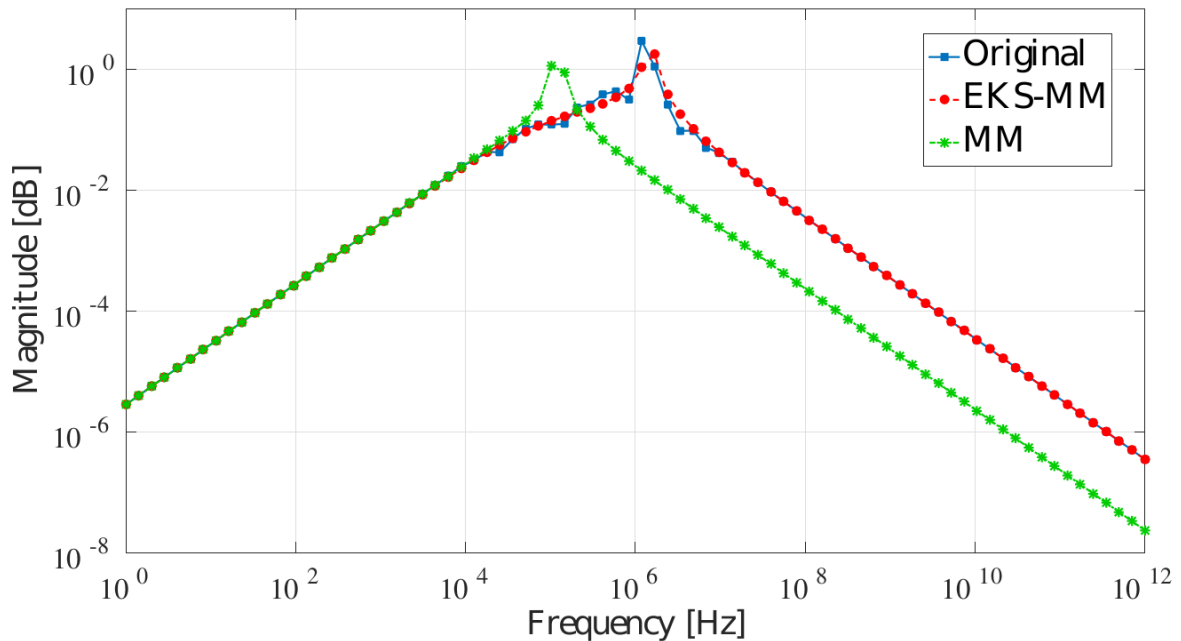
Για την καλύτερη απεικόνιση των αποτελεσμάτων, επιλέξαμε να δείξουμε ενδεικτικά τις συναρτήσεις μεταφοράς για τις δύο μεθόδους που υλοποιήσαμε (EKS-MM, MM) με αρχεία εισόδου ibmpg1, ibmpg2t, ibmpg6 καθώς και το μέγεθος λάθους με αρχείο εισόδου ibmpg6. Σαν σημείο αναφοράς έχουμε συμπεριλάβει και την αρχική συνάρτηση μεταφοράς.

Αρχικά θα απεικονίσουμε τα αποτελέσματα των πειραμάτων με το αρχείο εισόδου ibmpg1. Το παραπάνω αρχείο εισόδου είναι το μικρότερο από το σύνολο των αρχείων που τρέξαμε πειράματα με διάσταση 44946 και # θυρών 600. Στο διάγραμμα 6.1 παρατηρούμε ότι η συνάρτηση μεταφοράς για την βελτιωμένη προσέγγιση EKS-MM προσεγγίζει πολύ καλύτερα τη συνάρτηση μεταφοράς του αρχικού αλγορίθμου σε σχέση με την προσέγγιση MM.

Παρατηρούμε επίσης, ότι και για τις πιο υψηλές συχνότητες, η προσέγγιση EKS-MM παραμένει πολύ κοντά σε αντίθεση με αυτή της MM προσέγγισης η οποία αποκλίνει.

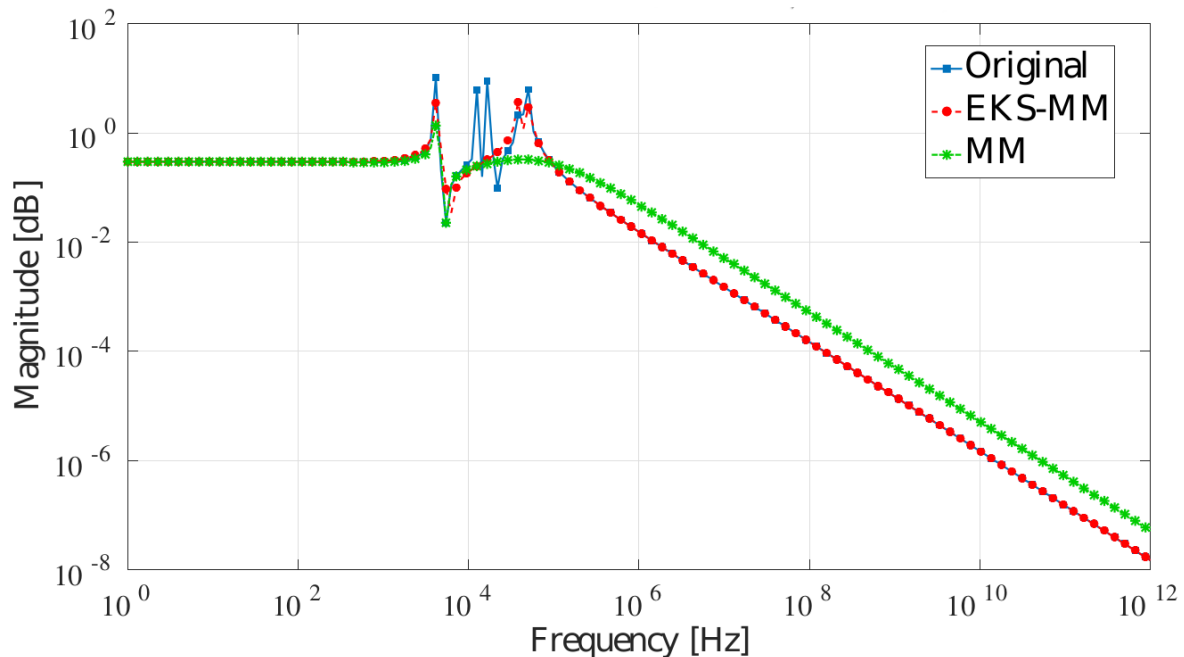
Στη συνέχεια, έχουμε την απεικόνιση της συνάρτησης μεταφοράς για το αρχείο εισόδου ibmpg2t. Υπενθυμίζουμε ότι το t στο τέλος του ονόματος του αρχείου σημαίνει ότι

Σχήμα 6.1: Σύγκριση της συνάρτησης μεταφοράς στα υποβιβασμένης τάξης μοντέλα (Reduced Order Models (ROMs) με τη χρήση της μεθόδου EKS-MM και της κλασσικής μεθόδου MM στο εύρος συχνοτήτων $[10^0, 10^{12}]$ για το αρχείο εισόδου `ibmpg1` για θύρες (9,9)



πραγματοποιήσαμε μεταβατική transient ανάλυση.

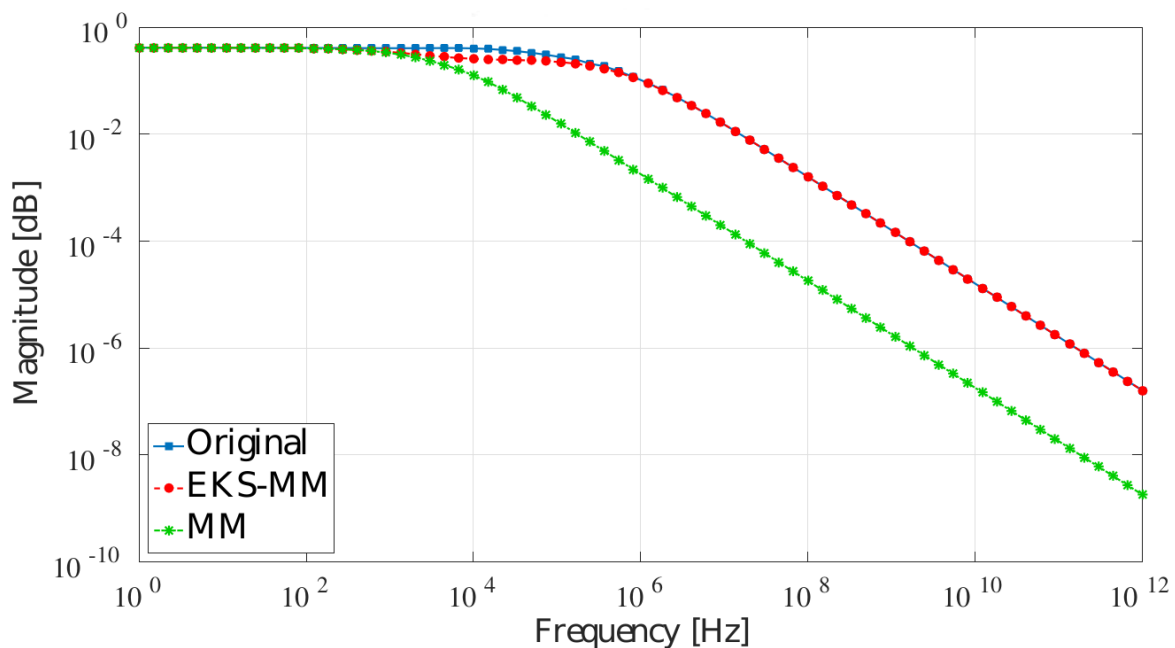
Σχήμα 6.2: Σύγκριση της συνάρτησης μεταφοράς στα υποβιβασμένης τάξης μοντέλα (Reduced Order Models (ROMs) με τη χρήση της μεθόδου EKS-MM και της κλασσικής μεθόδου MM στο εύρος συχνοτήτων $[10^0, 10^{12}]$ για το αρχείο εισόδου `ibmpg2t` για θύρες (4,4)



Στο σχήμα 6.2 βλέπουμε ότι και στην περίπτωση της μεταβατικής ανάλυσης, η συνάρτηση μεταφοράς της μεθόδου EKS-MM παρουσιάζει καλύτερη προσέγγιση στη συνάρτηση μεταφοράς του αρχικού αλγορίθμου από αυτόν της απλής MM μεθόδου.

Στη συνέχεια, θα παρουσιάσουμε τα αποτελέσματα από το αρχείο εισόδου `ibmpg6`. Υπενθυμίζουμε ότι αυτό το αρχείο εισόδου, περιλαμβάνει και τις μεγαλύτερες διαστάσεις (2506733) και # εισόδων(1000) από όλα τα υπόλοιπα αρχεία εισόδου(σχήμα 6.3).

Σχήμα 6.3: Σύγκριση της συνάρτησης μεταφοράς στα υποβιβασμένης τάξης μοντέλα (Reduced Order Models (ROMs) με τη χρήση της μεθόδου EKS-MM και της κλασσικής μεθόδου MM στο εύρος συχνοτήτων [10^0 , 10^{12}] για το αρχείο εισόδου `ibmpg6` για θύρες (5,5)

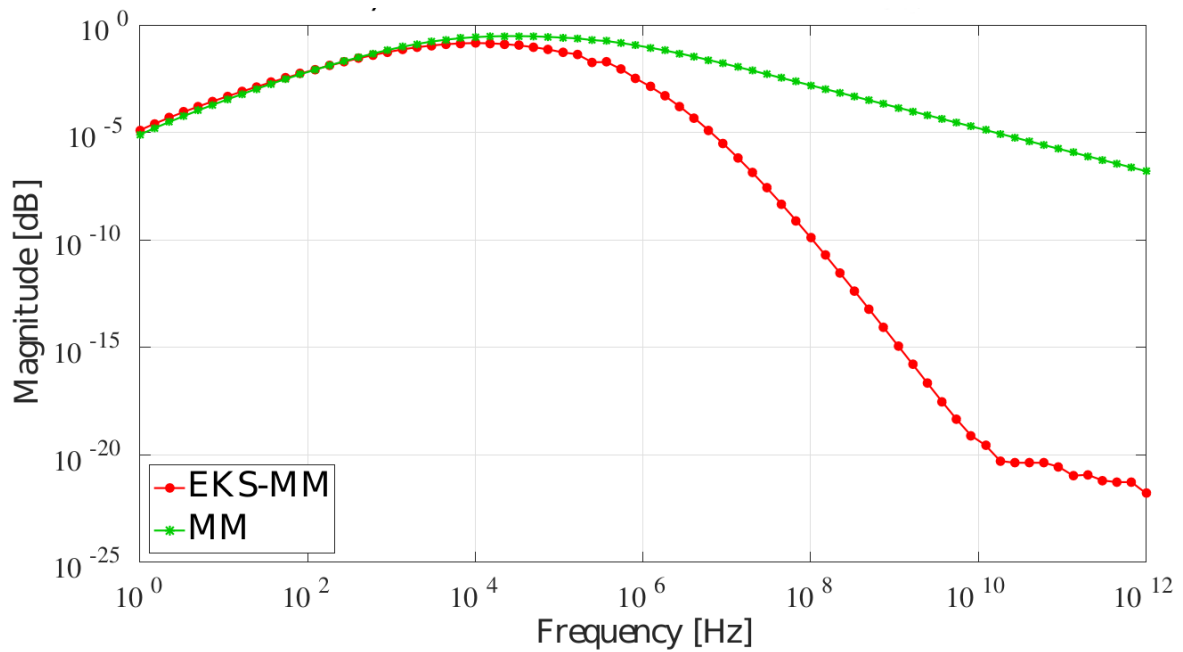


Το πείραμα μας πραγματοποιήθηκε για θύρες (5,5) και στο εύρος συχνοτήτων [10^0 , 10^{12}]. Τα αποτελέσματα όπως θα δούμε είναι πολύ ενθαρρυντικά καθώς η συνάρτηση μεταφοράς πλησιάζει αυτή του αρχικού αλγορίθμου ενώ έχουμε μεγάλη βελτίωση στο λάθος σε σχέση με τα αντίστοιχα αποτελέσματα της MM προσέγγισης ιδιαίτερα για συχνότητες που ξεπερνούν την 10^6 .

Στο διάγραμμα 6.3 μπορούμε να δούμε τη συνάρτηση μεταφοράς του αρχικού αλγορίθμου, της κλασσικής MM προσέγγισης και του βελτιωμένου αλγορίθμου EKS-MM.

Στο διάγραμμα 6.4 μπορούμε να δούμε το μέγεθος του λάθους μεταξύ των μεθόδων MM και EKS-MM.

Σχήμα 6.4: Μέγεθος του λάθους για τα υποβιβασμένης τάξης μοντέλα (Reduced Order Models (ROMs) με τη χρήση της μεθόδου EKS-MM και της κλασσικής μεθόδου MM στο εύρος συχνοτήτων $[10^0, 10^{12}]$ για το αρχείο εισόδου ibmpg6 για θύρες (5,5)



Παράρτημα Α΄

Ακρωνύμια και συντομογραφίες

MNA Modified Nodal Analysis

KCL Kirchhoff's Current Law

SPD Symmetric Positive Definite

MOR Model Order Reduction

PRIMA Passive Reduced-order Interconnect Macromodeling Algorithm

SPRIM Structure-Preserving Reduced-Order Interconnect Macromodeling

MM Moment Matching

ROMs Reduced-Order Models

BA Block Arnoldi

EBA Extended Block Arnoldi

Παράρτημα Β΄

Βιβλιοθήκη CSparse

Στο παρόν κεφάλαιο θα αναφέρουμε επιγραμματικά τη χρήση της βιβλιοθήκης CSparse σαν εξωτερική βιβλιοθήκη για την απεικόνιση αραιών πινάκων και πραγματοποίηση πράξεων και μετασχηματισμών πάνω σε αυτούς [2].

Οι συναρτήσεις της βιβλιοθήκης CSparse χωρίζονται σε 3 βασικές κατηγορίες: τις πρωτογενής, τις δευτερογενής και τις τριτογενής.

Οι πρωτογενής συναρτήσεις περιλαμβάνουν όλες τις συναρτήσεις που χρειάζεται μια εφαρμογή για να δημιουργήσει έναν πίνακα A , να λύσει ένα σύστημα $Ax = b$ και γενικά να εφαρμόσει βασικές μεθόδους πινάκων.

Οι δευτερεύουσες συναρτήσεις περιλαμβάνουν ταξινόμηση και παραγοντοποίηση πινάκων, μετασχηματισμοί για πίνακες μεταθέσεων κλπ.

Οι τριτογενής συναρτήσεις έχουν δημιουργηθεί κυρίως για εσωτερική χρήση εντός της βιβλιοθήκης.

Κάθε πρόγραμμα το οποίο χρησιμοποιεί τη CSparse βιβλιοθήκη πρέπει να περιλαμβάνει το header αρχείο:

```
#include "cs.h"
```

Στο αρχείο αυτό βρίσκονται οι δηλώσεις όλων των συναρτήσεων που παρέχει η βιβλιοθήκη καθώς και οι τύποι δεδομένων που χρησιμοποιούνται.

Πριν περιγράψουμε τις διάφορες συναρτήσεις που περιλαμβάνει η βιβλιοθήκη CSparse και χρησιμοποιούμε κατά την υλοποίηση στα πλαίσια της Διπλωματικής Εργασίας, θα χρειαστεί να ορίσουμε τα είδη των παραμέτρων και των δυνατών τιμών που μπορεί να επιστρέφονται από την κάθε συνάρτηση.

Κάθε παράμετρος ή τιμή που επιστρέφεται από τις συναρτήσεις της βιβλιοθήκης, περιγράφεται ως εξής:

- **in:** Η μνήμη της παραμέτρου δεσμεύεται και η τιμή της ορίζεται πριν τη κλήση της συνάρτησης και δεν μεταβάλλεται κατά την επιστροφή της
- **in/out:** Η μνήμη της παραμέτρου δεσμεύεται και η τιμή της ορίζεται πριν τη κλήση της συνάρτησης αλλά η τιμή της έχει μεταβληθεί μετά τη κλήση της συνάρτησης

- **out:** Η μνήμη πρέπει να έχει δεσμευτεί πριν τη κλήση της συνάρτησης. Η τιμή ορίζεται μετά το πέρας της συνάρτησης
- **work:** Η μνήμη πρέπει να έχει δεσμευτεί πριν τη κλήση της συνάρτησης. Η τιμή δεν ορίζεται ούτε πριν ούτε μετά το πέρας της συνάρτησης
- **returns:** Η επιστρεφόμενη τιμή κάθε συνάρτησης της βιβλιοθήκης. Σε περίπτωση που η συνάρτηση επιστρέφει δείκτη, εάν προκύψει λάθος, επιστρέφεται NULL

Επίσης να σημειώσουμε ότι σε όλη την υλοποίηση που κάναμε ορίζουμε ως m τον αριθμό των γραμμών και ως n τον αριθμό των στηλών.

Στη συνέχεια θα ορίσουμε τις συναρτήσεις που χρησιμοποιήσαμε από τη βιβλιοθήκη CSparse μαζί με μια συνοπτική περιγραφή της χρήσης της.

Αρχικά έχουμε την βασική δομή η οποία χρησιμοποιείται για την αναπαράσταση ενός αραιού πίνακα είτε σε μορφή τρίδυμης δομής απεικόνισης 3.1.1 είτε σε Δομή Συμπίεσμμένης Στήλης 3.1.1

cs:

```
typedef struct cs_sparse /* matrix in compressed-column or
                        triplet form */
{
    int nzmax ; /* maximum number of entries */
    int m ; /* number of rows */
    int n ; /* number of columns */
    int *p ; /* column pointers (size n+1) or col indices
             (size nzmax) */
    int *i ; /* row indices, size nzmax */
    double *x ; /* numerical values, size nzmax */
    int nz ; /* # of entries in triplet matrix, -1 for
             compressed-col */
} cs ;
```

Πρόσθεση πινάκων.

cs_add: $C = \alpha A + \beta B$

```
cs *cs_add (const cs *A, const cs *B, double alpha, double beta);
A          in          sparse matrix
B          in          sparse matrix
alpha     in          scalar
beta      in          scalar
          returns     C=alpha*A+beta*B; NULL on error
```

Μετατροπή πίνακα T Τρίδυμης Μορφής Απεικόνισης σε πίνακα C με Δομή Συμπίεσμμένης Στήλης. Οι στήλες του πίνακα C δεν είναι ταξινομημένες και μπορεί να υπάρχουν διπλές εγγραφές.

cs_compress:

```
cs *cs_compress (const cs *T) ;
T              in          sparse matrix in triplet form
          returns     C if successful; NULL on error
```

Αφαίρεση διπλών εγγραφών

cs_dupl:

```
int cs_dupl (cs *A) ;
A           in/out           sparse matrix;
           returns           1 if successful; 0 on error
```

Προσθήκη νέας τιμής σε έναν πίνακα Τρίδυμης Δομής Απεικόνισης. Σε περίπτωση που η μνήμη που έχει δεσμευτεί για τον πίνακα δεν επαρκεί, γίνεται επέκταση της μνήμης

cs_entry:

```
int cs_entry (cs *T, int i, int j, double x) ;
T           in/out           triplet matrix; new entry added on
           output
i           in               row index of new entry
j           in               column index of new entry
x           in               value of new entry
           returns           1 if successful; 0 on error
```

Πολλαπλασιασμός αραιών πινάκων.

cs_multiply: C = AB

```
cs *cs_multiply (const cs* A, const cs *B) ;
A           in               sparse matrix
B           in               sparse matrix
           returns           C = A*B; NULL on error
```

Εκτύπωση αραιού πίνακα.

cs_print:

```
int cs_print (const cs* A, int brief) ;
A           in               sparse matrix
brief      in               if 0 print all of A; a few entries
                           otherwise
           returns           1 if successful; 0 on error
```

Μετατόπιση αραιού πίνακα.

cs_transpose: C = A^T

```
cs *cs_transpose (const cs* A, int values) ;
A           in               sparse matrix
values     in               if 0 only pattern; both pattern and
                           values otherwise
           returns           C = A'; NULL otherwise
```

Δέσμευση μνήμης αραιού πίνακα.

cs_calloc:

```
void *cs_calloc (int n, size_t size) ;
n           in               number of items to allocate
size       in               size of each item in bytes
           returns           if successful pointer to allocated
                           block;
                           NULL on error
```

Απελευθέρωση μνήμης

cs_free:

```
void *cs_free (void *p) ;  
A          in/out      block to free  
          returns      NULL
```

Δομή για την ανάλυση μεθόδου Cholesky ή QR

css:

```
typedef struct cs_symbolic /* symbolic Cholesky, LU,  
                          or QR analysis */  
{  
    int *pinv ;           /* inverse row perm, for QR,  
                          fill perm for Chol */  
    int *q ;              /* fill-reducing column permutation  
                          for LU and QR */  
    int *parent ;        /* elimination tree for Cholesky  
                          and QR */  
    int *cp ;            /* column pointers for Cholesky,  
                          row counts for QR */  
    int *leftmost j      /* leftmost[i] = min(find(A(i,:))),  
                          for QR */  
    int m2 ;             /* # of rows for QR, after adding  
                          fictitious rows */  
    double lnz ;         /* # entries in L for LU or Cholesky;  
                          in V for QR */  
    double unz ;         /* # entries in U for LU;  
                          in R for QR */  
} css ;
```

Δομή απεικόνισης αποτελεσμάτων LU και QR παραγοντοποίησης

csn:

```
typedef struct cs_numeric /* numeric Cholesky, LU,  
                          or QR factorization */  
{  
    cs *L ;              /* L for LU and Cholesky,  
                          V for QR */  
    cs *U ;              /* U for LU, R for QR,  
                          not used for Cholesky */  
    int *pinv ;          /* partial pivoting for LU */  
    double *B ;          /* beta [0..n-1] for QR */  
} csn ;
```

Εξάλειψη μηδενικών από τους αραιούς πίνακες

cs_dropzeros:

```
int *cs_dropzeros (cs* A) ;  
A          in/out      array to remove the zeros from  
          returns      if successful 1; 0 otherwise
```


Παραγοντοποίηση QR αραιού πίνακα A , $n * m$ μεγέθους, όπου $m \geq n$

cs_qr:

```
csn *cs_qr (const cs* A, const css *S) ;  
A          in          matrix to factorize  
S          in          symbolic analysis from cs_sqr  
           returns    numeric factorization N;  
           NULL on error
```

Αλγεβρική επίλυση QR ή LU cs_sqr:

```
csn *cs_sqr (int order, const cs* A, const css *S) ;  
order      in          ordering method to use (0 to 3)  
A          in          matrix to factorize  
qr         in          analyze for QR if nonzero  
           or LU if zero  
           returns    S, symbolic analysis for cs_qr  
           or cs_lu; NULL on error
```

Παράρτημα Γ΄

Μορφή αρχείων εισόδου Matrix Market

Γ΄.1 Περιγραφή μορφής αρχείου

Η μορφή αρχείων εισόδου Matrix Market [17] αποτελεί μια βασική μορφή απεικόνισης πινάκων σε μορφή ASCII αρχείων. Το βασικό χαρακτηριστικό της μορφής αυτής είναι η απλότητα που τη διακρίνει κατά την περιγραφή ενός πίνακα. Ο στόχος της μορφής αυτής είναι να είναι εύκολα προσβάσιμο στη κατανόηση όσο από ανθρώπους όσο και από προγράμματα αλλά χωρίς να χάνεται η προσαρμοστικότητα σε εφαρμογές με πιο αυστηρές δομές ή η επεκτασιμότητα σε συναφείς δομές δεδομένων (όπως είναι η Τρίδυμη Δομή Απεικόνισης (3.1.1) και Δομή της Συμπιεσμένης Στήλης (3.1.1)).

Στα πλαίσια της παρούσας Διπλωματικής Εργασίας, θα εξετάσουμε δύο από τις βασικές μορφές δήλωσης πινάκων:

- **Μορφή Συντεταγμένων (Coordinate Format)**

Είναι η βασική μορφή που χρησιμοποιείται για την αναπαράσταση αραιών πινάκων. Η μορφή αυτή περιλαμβάνει μόνο τις μη-μηδενικές τιμές μαζί με τις συντεταγμένες τους στον πίνακα. Όπως συμπεραίνουμε, η μορφή αυτή είναι ιδανική μόνο όταν τα μη-μηδενικά στοιχεία είναι λιγότερα από τα μηδενικά.

- **Μορφή Πίνακα (Array Format)**

Είναι η βασική μορφή που χρησιμοποιείται για την αναπαράσταση πυκνών πινάκων. Στη μορφή αυτή, παρέχουμε όλες τις τιμές (μηδενικές και μη-μηδενικές) σε μια προ διατεταγμένη (ανά στήλη) σειρά.

Τα αρχεία εισόδου που διαχειριζόμαστε στην υλοποίηση μας είναι της πρώτης μορφής. Παραθέτουμε ένα απλό παράδειγμα πως ένας αραιός πίνακας αντικατοπτρίζεται σε αυτή τη μορφή:

10	15	0	0
0	2.34	0	221
1.32	0	0	23.324
0	0	0	98.58

Το παραπάνω παράδειγμα είναι από έναν γενικό αραιό πίνακα 4x4 με πραγματικές τιμές.

Στη μορφή Matrix Market το παραπάνω παράδειγμα θα παρουσιάζεται ως εξής :

```
%%MatrixMarket matrix coordinate real general
%=====
4 4 7
1 1 10
1 2 15
2 2 2.34
2 4 221
3 1 1.32
3 4 23.324
4 4 98.58
```

Η πρώτη γραμμή του αρχείου περιλαμβάνει τον τύπο της μορφής του αρχείου. Στην συγκεκριμένη περίπτωση, σημειώνει ότι το αντικείμενο εντός του αρχείου αντιπροσωπεύει έναν πίνακα σε Μορφή Συντεταγμένων (Coordinate Format) και οι τιμές που περιέχονται εντός του πίνακα είναι πραγματικές σε κανονική μορφή.

Υπάρχουν διάφορες παραλλαγές της παραπάνω μορφής που περιλαμβάνουν μιγαδικές αλλά και ακέραιες τιμές καθώς και μορφές που οι θέσεις από τις μη-μηδενικές τιμές περιγράφονται βάση προτύπου (pattern matrices).

Επίσης, επιπλέον παράμετροι μπορούν να χρησιμοποιηθούν για να δηλώσουμε συμμετρίες στον πίνακα οι οποίες μπορούν να μειώσουν ριζικά το μέγεθος των δεδομένων. Τέτοιες παράμετροι είναι symmetric, skew-symmetric, Hermitian. Στις περιπτώσεις αυτές μόνο οι τιμές για το κάτω τριγωνικό μέρος του πίνακα χρειάζεται να βρίσκονται εντός του αρχείου.

Στη συνέχεια, στο παραπάνω παράδειγμα βλέπουμε ότι η πρώτη γραμμή μετά τα σχόλια (τα οποία ξεκινάνε πάντα με τον χαρακτήρα % στην αρχή) έχουμε μια γραμμή η οποία έχει τη μορφή :

Αριθμός Γραμμών Αριθμός Στηλών Αριθμός Μη-Μηδενικών Στοιχείων

Η παραπάνω γραμμή μας βοηθάει στο να δεσμεύσουμε με ακρίβεια τον χώρο που θα χρειαστεί το πρόγραμμα μας για να αποθηκεύσει τον πίνακα.

Στη συνέχεια, όλες οι υπόλοιπες γραμμές έχουν την ίδια μορφή η οποία είναι η εξής :

Συντεταγμένη Γραμμής Συντεταγμένη Στήλης Τιμή

Κάθε μια τιμή που λαμβάνουμε από το αρχείο την τοποθετούμε εντός του πίνακα που έχουμε στην μνήμη.

Bibliography

- [1] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.
- [2] T. A. Davis, *Direct methods for sparse linear systems*. SIAM, 2006.
- [3] I. S. Duff, R. G. Grimes, and J. G. Lewis, "Sparse matrix test problems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 15, no. 1, pp. 1–14, 1989.
- [4] L. N. Trefethen and D. Bau III, *Numerical linear algebra*. Siam, 1997, vol. 50.
- [5] D. Young, "Iterative methods for solving partial difference equations of elliptic type," *Transactions of the American Mathematical Society*, vol. 76, no. 1, pp. 92–111, 1954.
- [6] A. Odabasioglu, M. Celik, and L. T. Pileggi, "Prima: Passive reduced-order interconnect macromodeling algorithm," in *The Best of ICCAD*. Springer, 2003, pp. 433–450.
- [7] R. W. Freund, "Sprim: structure-preserving reduced-order interconnect macromodeling," in *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004*. IEEE, 2004, pp. 80–87.
- [8] M. Heyouni, "Extended arnoldi methods for large low-rank sylvester matrix equations," *Applied numerical mathematics*, vol. 60, no. 11, pp. 1171–1182, 2010.
- [9] I. Jaimoukha and M. Ebrahim, "Krylov subspace methods for solving large lyapunov equations," *Siam Journal on Numerical Analysis - SIAM J NUMER ANAL*, vol. 31, 02 1994.
- [10] K. Jbilou, "Block krylov subspace methods for large algebraic riccati equations," *Numerical algorithms*, vol. 34, no. 2-4, pp. 339–353, 2003.
- [11] A. H. Bentbib, K. Jbilou, and E. Sadek, "On some extended block krylov based methods for large scale nonsymmetric stein matrix equations," *Mathematics*, vol. 5, no. 2, p. 21, 2017.
- [12] A. Bentbib, K. Jbilou, and E. M. Sadek, "On some krylov subspace based methods for large-scale nonsymmetric algebraic riccati problems," *Computers & Mathematics with Applications*, vol. 70, no. 10, pp. 2555–2565, 2015.

- [13] V. Druskin and L. Knizhnerman, "Extended krylov subspaces: approximation of the matrix square root and related functions," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 3, pp. 755–771, 1998.
- [14] V. Simoncini, "A new iterative method for solving large-scale lyapunov matrix equations," *SIAM Journal on Scientific Computing*, vol. 29, no. 3, pp. 1268–1288, 2007.
- [15] C. F. Van Loan and G. H. Golub, *Matrix computations*. Johns Hopkins University Press Baltimore, 1983.
- [16] S. R. Nassif, "Power grid analysis benchmarks," in *2008 Asia and South Pacific Design Automation Conference*. IEEE, 2008, pp. 376–381.
- [17] R. F. Boisvert, R. Pozo, and K. A. Remington, "The matrix market formats: Initial design," Technical report, Applied and Computational Mathematics Division, NIST, Tech. Rep., 1996.