



Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και

Μηχανικών Υπολογιστών

**“Επίλυση εξισώσεων ενεργειακού δικτύου με
επαναληπτικές μεθόδους”**

Διπλωματική Εργασία

Γεωργιάδης Κωνσταντίνος

Επιβλέποντες Καθηγητές:

Βάβαλης Εμμανουήλ

Καθηγητής

Πανεπιστήμιο Θεσσαλίας

Χούστη Αικατερίνη

Καθηγήτρια

Πανεπιστήμιο Θεσσαλίας

Βόλος, Ελλάδα

Ιούνιος 2015

[Κενή Σελίδα]



University of Thessaly

School of Engineering
Department of Electrical and
Computer Engineering

**“Iterative solution of power-flow equations:
experimental study”**

Diploma Thesis Project

Georgiadis Konstantinos

Supervisors:

Vavalis Emmanouil
Professor
University of Thessaly

Housti Aikaterini
Professor
University of Thessaly

(Signature)

.....

(Signature)

.....

Volos, Greece

June 2015

(Υπογραφή)

.....

Γεωργιάδης Κωνσταντίνος

Copyright © Georgiadis Konstantinos, 2015

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 το συγγραφέα.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη που μου παρείχε κατά τη διάρκεια των φοιτητικών μου χρόνων. Εν συνεχεία να ευχαριστήσω τους επιβλέποντες καθηγητές μου κύριο Βάβαλη Εμμανουήλ και κυρία Χούστη Αικατερίνη που μου εμπιστεύτηκαν την εν λόγω διπλωματική εργασία και με έφεραν σε επαφή με το άκρως ενδιαφέρον αντικείμενο των Συστημάτων Ηλεκτρικής Ενέργειας. Επίσης ένα μεγάλο ευχαριστώ στους υποψήφιους διδάκτορες του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Φάιντι Νικηφόρο, Νασιάκου Αντωνία και στον κύριο Ζημέρη Δημήτρη των οποίων η εμπειρία και η βοήθεια υπήρξε άμεση και πολύτιμη. Τέλος να ευχαριστήσω τους φίλους μου για τη συμπαράσταση που μου έδειξαν όλα τα χρόνια των σπουδών μου και ιδιαίτερα τους προπτυχιακούς φοιτητές του τμήματος Σκετόπουλο Νίκο και Σαρδέλη Σπύρο με τους οποίους είχα και την τιμή να συνεργαστώ άψογα σε πολλές εργασίες.

*Κωνσταντίνος Θ. Γεωργιάδης
Βόλος
Ιούνιος 2015*

Περίληψη

Στόχο της παρούσας διπλωματικής εργασίας αποτέλεσε η μελέτη των επαναληπτικών αριθμητικών μεθόδων και πως αυτές μπορούν να χρησιμοποιηθούν ώστε να επιλύσουν το πρόβλημα ροής φορτίου (power - flow) σε συστήματα ηλεκτρικής ενέργειας με τη χρήση εργαλείων – λογισμικού προσομοίωσης.

Στο 1^ο Κεφάλαιο γίνεται διατύπωση - ανάλυση του προβλήματος ροής φορτίου καθώς και του μοντέλου που κατασκευάζεται για την επίλυσή του.

Στο 2^ο Κεφάλαιο γίνεται αναφορά στη γενική λειτουργία των επαναληπτικών μεθόδων και επικεντρωνόμαστε σε δύο εξ' αυτών· τη μέθοδο των *Συζυγών Κλίσεων (Conjugate Gradients)* και της μεθόδου *Γενικευμένου Ελάχιστου Υπολοίπου (GMRES)*.

Το 3^ο Κεφάλαιο παρουσιάζει το λογισμικό προσομοίωσης που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία, GridLAB-d, εκτελώντας ορισμένες προσομοιώσεις. Τροποποιώντας κάθε φορά τις παραμέτρους που χαρακτηρίζουν το δίκτυο εξάγονται και ερμηνεύονται αποτελέσματα βάσει των αλλαγών που έγιναν. Στη συνέχεια γίνεται αναφορά στην ανάπτυξη κώδικα στο GridLAB-D ώστε να απομονωθεί το γραμμικό σύστημα το οποίο θα επιλυθεί με χρήση επαναληπτικών μεθόδων.

Στο 4^ο Κεφάλαιο γίνεται αναφορά στα βασικά χαρακτηριστικά της βιβλιοθήκης PETSc που χρησιμοποιήθηκε για την κατασκευή του λογισμικού – επιλυτή (solver) για την επίλυση του συστήματος με επαναληπτικές μεθόδους. Παράλληλα παρέχεται και ο πηγαίος κώδικας του επιλυτή.

Το Κεφάλαιο 5 περιλαμβάνει τα πειράματα για την επίλυση του συστήματος με τις δυο προαναφερθείσες επαναληπτικές μεθόδους και παρουσίαση - αξιολόγηση της απόδοσής τους.

Abstract

The aim of this thesis was the study of iterative numerical methods and how these methods can be used in simulation software, in order to solve the power – flow problem in power systems.

The 1st Chapter analyzes the power – flow problem and defines a model for solving it.

The 2nd Chapter refers to the general operation of iterative methods and focuses on two of them; the method of Conjugate Gradients' and the Generalized Minimum Residual method.

The 3rd Chapter presents the simulation software used in this thesis, GridLAB-D, by performing simulations, modifying each time the parameters that characterize the network. We extract the results and interpret them with the changes made each time. Furthermore source code is developed in GridLAB-D to isolate the linear system which will be solved using iterative methods.

The 4th Chapter refers to the essential features of PETSc library used to develop the solver software which uses the above-mentioned iterative methods. Alongside the source code of this solver is provided.

The 5th Chapter includes experiments and results from solving the isolated system with the two aforementioned methods and evaluates their performance.

Περιεχόμενα

Κεφάλαιο 1	1
1.1 Ανάλυση Ροής Φορτίου.....	1
1.2 Εξισώσεις ροής φορτίου.....	2
1.3 Πρόβλημα ροής φορτίου	5
Κεφάλαιο 2	7
2.1 Εισαγωγή για τις επαναληπτικές μεθόδους	7
2.2 Βελτίωση επαναληπτικής λύσης	8
2.3 Ευαισθησία και Αριθμός Κατάστασης	8
2.4 Επαναληπτικές μέθοδοι μη σταθερού σημείου – Preconditioner πίνακες.....	10
2.5 Η μέθοδος Conjugate Gradients	10
2.5.1 Σύγκλιση	12
2.6 Η μέθοδος GMRES.....	13
2.6.1 Υλοποίηση	13
Κεφάλαιο 3	16
3.1 Εισαγωγή στο GridLAB-D.....	16
3.2 Πειράματα προσομοιώσεων	18
3.3 Ανάλυση – Επέκταση πηγαίου κώδικα του GridLAB-D	20
Κεφάλαιο 4	24
4.1 Εισαγωγή στη βιβλιοθήκη PETSc.....	24
4.2 Δομή Βιβλιοθήκης PETSc.....	25
4.3 Χαρακτηριστικά της βιβλιοθήκης.....	27
4.4 Διανύσματα – Vectors	28
4.5 Πίνακες - Mat	28
4.6 Γραμμικοί επιλυτές και πίνακες προρρυθμιστές	28
4.7 Υλοποίηση επιλυτή γραμμικών συστημάτων	30
Κεφάλαιο 5	31
5.1 Παράμετροι Εκτέλεσης.....	31
5.2 Πείραμα 1 ^ο	31
5.2.1 Μέθοδος επίλυσης Conjugate Gradients.....	32
5.2.2 Μέθοδος επίλυσης GMRES	32
5.3 Πείραμα 2 ^ο	32
5.3.1 Μέθοδος επίλυσης Conjugate Gradients.....	32
5.3.2 Μέθοδος επίλυσης GMRES	32

5.4 Πείραμα 3 ^ο	33
5.4.1 Μέθοδος επίλυσης Conjugate Gradients.....	33
5.4.2 Μέθοδος επίλυσης GMRES	33
5.5 Συμπεράσματα	33
Αναφορές	35

Πίνακας Εικόνων

Εικόνα 1. Μονογραμμικό Διάγραμμα Συστήματος	2
Εικόνα 2. Σύστημα εξισώσεων n κόμβων	3
Εικόνα 3. Preconditioned Μορφή της Conjugate Gradients Μεθόδου.....	11
Εικόνα 4. Preconditioned Μορφή της GMRES(m) Μεθόδου	14
Εικόνα 5. Οργάνωση της Βιβλιοθήκης PETSc	25
Εικόνα 6. Αριθμητικές Βιβλιοθήκες της PETSc	26
Εικόνα 7. Λίστα με Προρυθμιστές του PETSc	29

Κεφάλαιο 1

Ροή Φορτίου

1.1 Ανάλυση Ροής Φορτίου

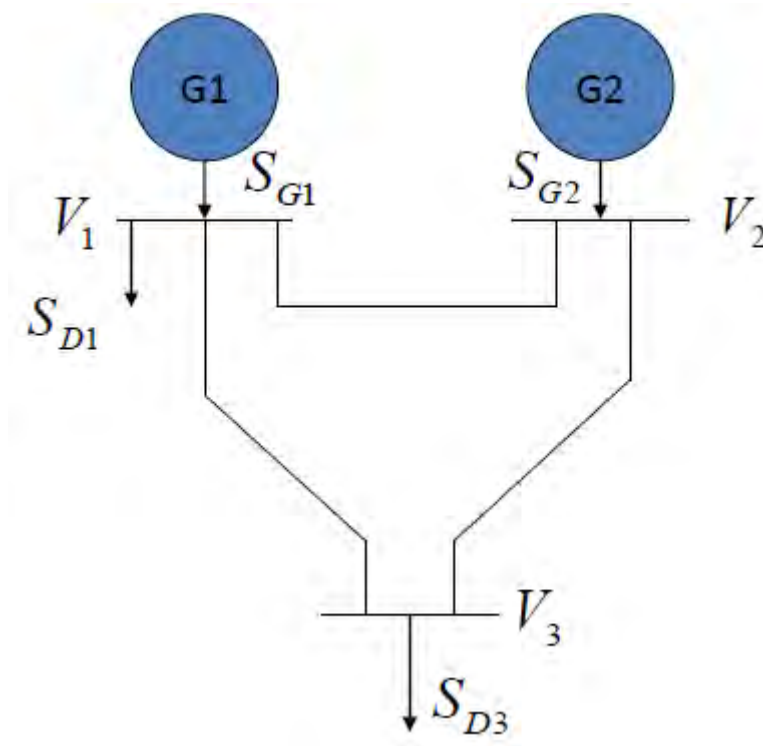
Σκοπό ενός συστήματος ηλεκτρικής ενέργειας αποτελεί η παροχή ηλεκτρικής ισχύος από τους παραγωγούς στους καταναλωτές σε πραγματικό χρόνο βάσει της τρέχουσας ζήτησης, τηρουμένων των προκαθορισμένων ορίων σε τάση και συχνότητα με τρόπο ταυτόχρονα αξιόπιστο και οικονομικό. Κατά τη θεώρηση του προβλήματος ροής ισχύος το σύστημα παροχής ηλεκτρικής ενέργειας αποτελείται από ένα σύνολο κόμβων οι οποίοι διασυνδέονται μεταξύ τους μέσω γραμμών μεταφοράς. Στους κόμβους αυτούς θεωρούμε ότι διασυνδέονται οι γεννήτριες και τα φορτία του ηλεκτρικού δικτύου, που εισάγουν και καταναλώνουν ηλεκτρική ισχύ αντίστοιχα.

Το μοντέλο αυτό χρησιμοποιείται ώστε να προβούμε σε υπολογισμούς των τάσεων και των ισχύων στους κόμβους, όταν το σύστημα ηλεκτρικής ενέργειας βρίσκεται στη σταθερή του κατάσταση, έχουν παρέλθει δηλαδή οποιαδήποτε μεταβατικά φαινόμενα. Το πρόβλημα αυτό ομοιάζει με το πρόβλημα εύρεσης τάσεων και ρευμάτων ενός κυκλώματος στη σταθερή του κατάσταση. Ο επιτακτικότερος λόγος που ωθεί στην επίλυση του είναι ότι αποτελεί αναπόσπαστο κομμάτι του στρατηγικού σχεδιασμού μιας εταιρείας παροχής ηλεκτρικής ενέργειας. Στους παραπάνω υπολογισμούς δίνεται επίσης έμφαση για το αν τηρούνται τα όρια τάσης, η αποφυγή υπερφόρτωσης τόσο των γραμμών μεταφοράς, όσο και των γεννητριών καθώς και ο παράγοντας ευστάθειας που έχει να κάνει με τη γωνία του ρεύματος το οποίο παράγουν οι γεννήτριες του δικτύου. Στόχος μας μετά την επίλυση αποτελεί η γνώση των σχέσεων που διέπουν τις 3 παρακάτω μεταβλητές:

- S_{Di} , της μιγαδικής ισχύος που απορροφάται από τα φορτία στον κόμβο i
- S_{Gi} , της μιγαδικής ισχύος που εγχέεται από τις γεννήτριες στον κόμβο i
- V_i , της τάσης στον κόμβο i

1.2 Εξισώσεις ροής φορτίου

Για τη μελέτη του παραπάνω προβλήματος καθίσταται απαραίτητη θεώρηση ενός μοντέλου εργασίας για τους παραπάνω υπολογισμούς. Αρχικά αναφερόμαστε σε τριφασικό σύστημα μεταφοράς ισχύος, στο οποίο κάθε φάση διαφέρει από τις υπόλοιπες κατά σταθερή γωνία. Επίσης θεωρούμε ότι η εμπέδηση που παρουσιάζεται από τους μετασχηματιστές τάσης του συστήματος, συναθροίζεται είτε στην εμπέδηση είτε του φορτίου είτε των γεννητριών του συστήματος καθώς χάριν ευκολίας οι μετασχηματιστές δεν εμφανίζονται στο παραπάνω μοντέλο. Μια αναπαράσταση του παραπάνω μοντέλου εμφανίζεται στην Εικόνα 1.2.1.



Εικόνα 1. Μονογραμμικό Διάγραμμα Συστήματος. Πηγή: [10]

Ορίζουμε τη μιγαδική ισχύ ανά φάση σε ένα κόμβο i ως τη διαφορά της ισχύος που εγχέεται, από αυτή που απορροφάται βάσει της παρακάτω σχέσης:

$$S_i = S_{Gi} - S_{Di}$$

Σχέση 1.2.2

Χρησιμοποιώντας τη διατήρηση της μιγαδικής ισχύος σε έναν κόμβο i μπορούμε να καταλήξουμε στην παρακάτω σχέση η οποία εκφράζει την ισχύ στον κόμβο i ως το αλγεβρικό άθροισμα των ισχύων που μεταφέρονται από και προς τον κόμβο i με κάθε άλλο κόμβο με τον οποίο διασυνδέεται.

$$S_i = \sum_{k=1}^n S_{ik} \quad i = 1, 2, \dots, n$$

Σχέση 1.2.3

Επίσης ορίζουμε το ρεύμα I_i που διαρρέει τον κόμβο i βάσει της παρακάτω σχέσης που αποτελεί ουσιαστικά διατύπωση του νόμου των ρευμάτων του Kirchhoff:

$$I_i = I_{Gi} - I_{Di} = \sum_{k=1}^n I_{ik}$$

Σχέση 1.2.4

Όπου το I_i αναπαριστά το ολικό ρεύμα μιας φάσης που εισέρχεται στο σύστημα μεταφοράς, με I_{Gi} το ρεύμα που εγχέεται στον κόμβο i και I_{Di} το ρεύμα που απορροφάται από τον κόμβο i .

Για κάθε ηλεκτρικό δίκτυο ορίζουμε τον πίνακα αγωγιμοτήτων Y_{bus} . Βάσει του τρόπου με τον οποίο έχουμε ορίσει τον πίνακα αγωγιμοτήτων, τα ρεύματα που εγχέονται στους κόμβους σχετίζονται με τις τάσεις στους κόμβους. Ο εν λόγω πίνακας για n κόμβους φτάνει σε μέγεθος τα $n \times n$ στοιχεία εκ των οποίων μάλιστα τα περισσότερα είναι μηδενικά. Το παραπάνω αποτελεί απόρροια του γεγονότος ότι κάθε κόμβος συνδέεται φυσικά μέσω γραμμών με λίγους σχετικά κόμβους και η έλλειψη σύνδεσης αναπαρίσταται με χρήση μηδενικών για τα στοιχεία του πίνακα. Για τη δημιουργία μιας αναπαράστασης του δικτύου ανά φάση χρησιμοποιούμε σαν φάση αναφοράς την ουδέτερη και βάσει αυτής υπολογίζουμε τα ρεύματα και τις τάσεις στους κόμβους. Η σχέση μεταξύ των τάσεων και των ρευμάτων δίνεται από την παρακάτω σχέση:

$$I = Y_{bus} \cdot V$$

Σχέση 1.2.5

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_i \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1i} & \cdots & Y_{1n} \\ Y_{21} & Y_{22} & \cdots & Y_{2i} & \cdots & Y_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ Y_{i1} & Y_{i2} & \cdots & Y_{ii} & \cdots & Y_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{ni} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_i \\ \vdots \\ V_n \end{bmatrix}$$

Εικόνα 2. Σύστημα εξισώσεων n κόμβων. Πηγή [10]

Όπου:

- I : διάνυσμα ρευμάτων στους κόμβους
- Y_{bus} : πίνακας αγωγιμοτήτων δικτύου
- V : διάνυσμα τάσεων στους κόμβους

Κατά τη δημιουργία του πίνακα αγωγιμοτήτων πρέπει να λάβουμε υπόψιν τα εξής χαρακτηριστικά:

- Ο πίνακας αγωγιμοτήτων είναι συμμετρικός
- Τα διαγώνια στοιχεία του πίνακα Y_{ii} αποτελούν το άθροισμα όλων των επιμέρους αγωγιμοτήτων των συνδέσεων του κόμβου i με κάθε άλλο στοιχείο του δικτύου που αυτός συνδέεται
- Τα μη διαγώνια στοιχεία Y_{ij} της θέσης ij του πίνακα αγωγιμοτήτων είναι ίσα με το αρνητικό του αθροίσματος όλων των αγωγιμοτήτων των στοιχείων που συνδέονται μεταξύ των κόμβων i και j .

Βάσει της παραπάνω θεώρησης γίνεται προφανές ότι μπορούμε να αναπαραστήσουμε τη σχέση 1.2.4 ως εξής:

$$I_i = \sum_{k=1}^n Y_{ik} V_k$$

Σχέση 1.2.7

Στη συνέχεια παρατηρούμε ότι η μιγαδική ισχύς στον κόμβο i μπορεί να εκφραστεί από τη σχέση:

$$S_i = V_i I_i^*$$

Σχέση 1.2.8

όπου V_i η τάση σε εκθετική αναπαράσταση:

$$V_i = |V_i| e^{j\angle V_i} = |V_i| e^{j\theta_i}$$

Σχέση 1.2.9

και η συζυγής μορφή του ρεύματος στον κόμβο i :

$$I_i^*$$

Σχέση 1.2.10

οπότε η 1.2.8 γίνεται:

$$S_i = V_i \sum_{k=1}^n Y_{ik}^* V_k^*$$

Σχέση 1.2.11

Εαν αντικαταστήσουμε στην τελευταία σχέση την σύνθετη αγωγιμότητα ως:

$$Y_{ik} = G_{ik} + jB_{ik}$$

Σχέση 1.2.12

και έπειτα χωρίσουμε την 1.2.11 σε δύο επιμέρους εξισώσεις που περιλαμβάνουν το πραγματικό και το φανταστικό μέρος της ισχύος στον κόμβο i αντίστοιχα:

$$\sum_{k=1}^n |V_i| |V_k| (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) - P_i = 0$$

Σχέση 1.2.13

$$\sum_{k=1}^n |V_i| |V_k| (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) - Q_i = 0$$

Σχέση 1.2.14

Οι δυο τελευταίες εξισώσεις αποτελούν και τις εξισώσεις ροής φορτίου τις οποίες καλούμαστε να λύσουμε για κάθε κόμβο του ηλεκτρικού δικτύου.

1.3 Πρόβλημα ροής φορτίου

Συνεχίζοντας την ανάλυση του μοντέλου το οποίο υιοθετήσαμε για τη μελέτη του ηλεκτρικού δικτύου, προχωρούμε σε ένα διαχωρισμό των κόμβων ανάλογα με το είδος τους. Αρχικά έχουμε τους κόμβους των γεννητριών που διασυνδέονται με γεννήτριες που τροφοδοτούν το δίκτυο μας με ισχύ. Οι υπόλοιποι κόμβοι στους οποίους δε διασυνδέονται γεννήτριες ονομάζονται κόμβοι φορτίων. Στην εικόνα 1.2.1 παρατηρούμε τρεις κόμβους εκ των οποίων οι 1 και 2 αποτελούν κόμβους γεννητριών, ενώ ο κόμβος 3 αποτελεί κόμβο φορτίου.

Λειτουργικοί λόγοι υποδεικνύουν ότι σε ένα κόμβο γεννητριών, η ενεργός ισχύς P_{Gi} καθώς και το μέτρο της τάσης λειτουργίας μπορούν να προσδιοριστούν εύκολα. Σε όλους τους κόμβους του συστήματος θεωρούμε την μιγαδική ισχύ που S_{Di} που καταναλώνεται γνωστή. Στη γενική περίπτωση θεωρούμε ότι σε μερικούς κόμβους μπορούν να προσδιοριστούν τα P_i και Q_i ενώ σε άλλους μπορούν να προσδιοριστούν τα P_i και $|V_i|$. Πρέπει ωστόσο να τονιστεί ότι είναι αδύνατο να προβούμε σε υπολογισμό καθενός P_i ανεξάρτητα, καθώς τίθεται περιορισμός λόγω της ανάγκης εξισορρόπησης της ενεργού ισχύος στο σύστημα. Για ένα σύστημα μεταφοράς χωρίς απώλειες, το άθροισμα όλων των ενεργών ισχύων P_i , όλων των κόμβων του συστήματος παραμένει μηδέν. Συνεπώς κάθε μία από τις ενεργές ισχύς μπορεί να καθοριστεί με τον προσδιορισμό όλων των υπολοίπων. Από την άλλη, για ένα σύστημα με απώλειες το άθροισμα των ενεργών ισχύων, όλων των κόμβων του συστήματος πρέπει να ισούται με τις θερμικές απώλειες I^2R που συμβαίνουν κατά τη μεταφορά. Ένα πρόβλημα που εγείρεται από την ύπαρξη των παραπάνω απωλειών είναι η μη ακριβής γνώση των, πριν την εφαρμογή υπολογισμών για την επίλυση του προβλήματος ροής φορτίου. Η λύση για το παραπάνω ζήτημα είναι απλή και αποτελεσματική και για τις δύο περιπτώσεις. Για υπολογισμούς τιμών στη σταθερή κατάσταση η επιλογή είναι αυθαίρετη· ορίζουμε τις

ενεργές ισχύς σε όλους τους κόμβους πλην ενός. Η εισαγόμενη ισχύς στον κόμβο αυτό είναι ουσιαστικά αυτή η οποία εξισορροπεί τις ενεργές ισχύς. Συνηθίζεται δε, εντελώς αυθαίρετα, να αριθμούμε τους κόμβους ώστε η γεννήτρια που συνδέεται στον παραπάνω κόμβο για να επιτελέσει την παραπάνω λειτουργία, να είναι συνδεδεμένη στον κόμβο 1. Για την εν λόγω γεννήτρια δεν ορίζουμε την ενεργό ισχύ P_1 ή ισοδύναμα την P_{G1} , αλλά ορίζουμε την τάση $|V_1|/\angle V_1$. Με την επιλογή του ορίσματος της τάσης $\angle V_1$ δεν επιλέγουμε τίποτα περισσότερο από χρονική αναφορά. Για το λόγο αυτό και χάριν ευκολίας, επιλέγουμε τη γωνία αυτή ίση με $\theta_1 = \angle V_1 = 0$.

Εν κατακλείδι έχουμε τους εξής τρεις τύπους πηγών στους διαφορετικούς κόμβους:

- Μια πηγή τάσης. Τη θεωρούμε στον κόμβο 1
- Πηγές $P, |V|$. Στους υπόλοιπους κόμβους γεννητριών
- Πηγές P, Q . Στους κόμβους φορτίων

Ο κόμβος 1 ονομάζεται κόμβος αναφοράς τάσης και αναφέρεται στην αγγλική βιβλιογραφία επίσης ως “*slack bus*” είτε ως “*swing bus*”. Κόμβοι με πηγές $P, |V|$ ονομάζονται $P, |V|$ κόμβοι είτε κόμβοι ελέγχου τάσης. Κόμβοι με πηγές ενεργού και άεργου ισχύος ονομάζονται κόμβοι P, Q είτε κόμβοι φορτίου.

Τέλος πρέπει να αναφερθεί ότι ένας κόμβος μπορεί να προσδιοριστεί με σαφήνεια, είτε ως κόμβος γεννήτριας, είτε ως κόμβος φορτίου· στην περίπτωση ενός κόμβου φορτίου με πυκνωτές ο κόμβος μπορεί να θεωρηθεί P, Q αν οι πυκνωτές παρέχουν σταθερή άεργη ισχύ είτε ως $P, |V|$ αν οι πυκνωτές χρησιμοποιούνται ώστε να διατηρηθεί σταθερή η καθορισμένη ενεργός ισχύς P και μέτρο της τάσης $|V|$.

Εν κατακλείδι πρέπει να τονιστεί ότι έχουμε 4 συνολικά μεταβλητές $P, Q, |V|, \theta$ εκ των οποίων για κάθε κόμβο του συστήματος οι 2 θα είναι γνωστές και καλούμαστε να επιλύσουμε το πρόβλημα ως προς τις 2 άγνωστες.

Οι ενότητες 1.1, 1.2, και 1.3 περιέχουν υλικό που αντλήθηκε από την αναφορά [\[1\]](#).

Κεφάλαιο 2

Επαναληπτικές Μέθοδοι

2.1 Εισαγωγή για τις επαναληπτικές μεθόδους

Δεν είναι υπερβολή να πούμε ότι η επίλυση της πολύ μεγάλης πλειοψηφίας των προβλημάτων στην επιστήμη και στη μηχανική, ανάγεται με τον ένα ή με τον άλλο τρόπο στον υπολογισμό της λύσης συστημάτων γραμμικών αλγεβρικών εξισώσεων με χρήση αριθμητικών μεθόδων. Σε τελική φάση αναζητούμε ένα διάνυσμα \mathbf{x} το οποίο αν πολλαπλασιαστεί από αριστερά με τον πίνακα \mathbf{A} , δίνει το γνωστό διάνυσμα \mathbf{b} . Η παραπάνω προσέγγιση ακολουθείται και στην επίλυση του προβλήματος ροής φορτίου σε ένα δίκτυο ηλεκτρικής ενέργειας.

Τις προαναφερθείσες αριθμητικές μεθόδους μπορούμε να τις χωρίσουμε σε δύο μεγάλες κατηγορίες α) τις ευθείες ή άμεσες (direct) και β) τις επαναληπτικές (iterative) μεθόδους. Στις ευθείες μεθόδους οι υπολογισμοί τερματίζονται μετά από έναν πεπερασμένο αριθμό πράξεων, των οποίων το πλήθος είναι πολυωνυμική συνάρτηση του μεγέθους του συστήματος N . Στο τέλος των πράξεων και δεδομένου ότι ο υπολογιστής είναι ικανός να εκτελεί πράξεις με άπειρη ακρίβεια, η λύση θα ήταν ακριβής. Ωστόσο πραγματικές ή ουσιαστικές δυσκολίες εμφανίζονται όταν το μέγεθος του συστήματος είναι μεγάλο, περιλαμβάνοντας μερικές εκατοντάδες εξισώσεις. Οι επαναληπτικές μέθοδοι γεννούν διαδοχικές προσεγγίσεις στη λύση βασισμένες σε αλγορίθμους που χρησιμοποιούν πολλαπλασιασμούς πίνακα με διάνυσμα.

Μια τυπική επαναληπτική μέθοδος προχωράει επιλέγοντας κάποιο αρχικό διάνυσμα $\mathbf{x}^{(0)}$ και υπολογίζοντας στη συνέχεια την ακολουθία των διανυσμάτων:

$$\mathbf{x}^{(k+1)} = \mathbf{P} \cdot \mathbf{x}^{(k)} + \mathbf{c}$$

Σχέση 2.1.1

όπου \mathbf{P} είναι ο κατάλληλος πίνακας προβολής ή επαναληπτικός πίνακας και το \mathbf{c} είναι ένα σταθερό διάνυσμα. Τα \mathbf{P} και \mathbf{c} σχεδιάζονται έξυπνα συναρτήσει του πίνακα των συντελεστών \mathbf{A} και του σταθερού διανύσματος \mathbf{b} , έτσι ώστε όταν η ακολουθία $\mathbf{x}^{(k)}$ συγκλίνει, το όριο να συμπίπτει με τη λύση του γραμμικού συστήματος $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$.

Οι επαναληπτικές μέθοδοι είναι ιδιαίτερα αποτελεσματικές για αραιά συστήματα, όπου ο πίνακας των συντελεστών \mathbf{A} και ο επαναληπτικός πίνακας \mathbf{P} περιέχουν πολλά μηδενικά στοιχεία, καθώς οι άσκοποι πολλαπλασιασμοί με το μηδέν μπορούν να παρακαμφθούν εύκολα. Σε αντίθεση οι τριγωνικοί παράγοντες ενός αραιού πίνακα δεν είναι αναγκαστικά αραιοί με αποτέλεσμα οι άμεσες μέθοδοι να μην επωφελούνται γενικά από την απόφαση για αποσύζευξη των αγνώστων. Ένα σημαντικό γνώρισμα των επαναληπτικών μεθόδων είναι ότι υπόκεινται σε υπολογισμούς μέσω διανυσμάτων και σε παράλληλους υπολογισμούς. Για παράδειγμα ο πολλαπλασιασμός πίνακα με διάνυσμα $\mathbf{P} \cdot \mathbf{x}^{(k)}$ αποσυντίθεται σε διάφορα μπλοκ και ο πολλαπλασιασμός τμημάτων του $\mathbf{x}^{(k)}$ με αντίστοιχους υποπίνακες του \mathbf{P} ανατίθεται σε διαφορετικούς επεξεργαστές.

Από την άλλη πλευρά, οι επαναληπτικές μέθοδοι απαιτούν άπειρο αριθμό επαναληπτικών υπολογισμών πριν παράγουν την ακριβή λύση, ακόμα και σε έναν ιδανικό υπολογιστή με άπειρη ακρίβεια και μηδενικό σφάλμα στρογγυλοποίησης. Στην πράξη ωστόσο, αν η μέθοδος είναι καλά σχεδιασμένη, το αριθμητικό σφάλμα γίνεται υποφερτά μικρό ύστερα από μικρό ή μέτριο σε πλήθος αριθμό επαναλήψεων, πολύ λιγότερων από τη διάσταση του διανύσματος των αγνώστων. Αν δεν είμαστε σε αυτή την περίπτωση αποτελεί συνετή λύση να εγκαταλείψουμε την επαναληπτική μέθοδο υπέρ μιας άμεσης.

2.2 Βελτίωση επαναληπτικής λύσης

Μόλις βρεθεί η λύση ενός γραμμικού συστήματος, ας την καλέσουμε $\mathbf{x}^{(0)}$, θα πρέπει να επαληθευτεί επιβεβαιώνοντας ότι το μέγεθος του διανύσματος των υπολοίπων:

$$\mathbf{r}^{(0)} = \mathbf{A} \cdot \mathbf{x}^{(0)} - \mathbf{b}$$

Σχέση 2.2.1

είναι ικανοποιητικά μικρό, με την ελπίδα να είναι συγκρίσιμο με το σφάλμα στρογγυλοποίησης του υπολογιστή. Ατυχώς, λόγω της στρογγυλοποίησης, το μέγεθος του $\mathbf{r}^{(0)}$ μπορεί να είναι μεγαλύτερο από αυτό που μπορεί να γίνει ανεκτό.

Για να διορθώσουμε αυτή την ατέλεια, εκτελούμε *εκ των υστέρων (a posteriori)* βελτιώσεις. Η ακριβής λύση \mathbf{X} ικανοποιεί την εξίσωση $\mathbf{0} = \mathbf{A} \cdot \mathbf{X} - \mathbf{b}$. Αφαιρώντας τα αντίστοιχα μέρη από τη σχέση 2.1.2 προκύπτει:

$$\mathbf{A} \cdot \mathbf{e} = \mathbf{r}^{(0)}$$

Σχέση 2.2.2

Όπου $\mathbf{e} = \mathbf{x}^{(0)} - \mathbf{X}$ είναι το σφάλμα. Τα τρία βασικά βήματα που ακολουθούμε είναι:

- Επιλύουμε τη σχέση 2.1.3 ως προς \mathbf{e} .
- Υπολογίζουμε τη βελτιωμένη λύση $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \mathbf{e}$.
- Επαναλαμβάνουμε εάν είναι αναγκαίο.

2.3 Ευαισθησία και Αριθμός Κατάστασης

Ας δεχθούμε ότι το \mathbf{X} είναι η μοναδική λύση κάποιου γραμμικού συστήματος $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$. Ας τροποποιήσουμε τώρα το σταθερό διάνυσμα \mathbf{b} κατά μικρή ποσότητα $\Delta \mathbf{b}$, διατηρώντας τον \mathbf{A} σταθερό. Η λύση του γραμμικού συστήματος θα μεταβληθεί από \mathbf{X} σε $\Delta \mathbf{X}$. Θέλουμε να γνωρίσουμε ποια είναι η σχετική μεταβολή του \mathbf{X} σε σχέση με τη σχετική μεταβολή του \mathbf{b} . Δηλαδή θέλουμε να εκτιμήσουμε την ευαισθησία της λύσης σε μια μικρή διαταραχή του δεξιού μέλους. Προφανώς το ζήτημα αυτό σχετίζεται με τη σημασία του σφάλματος στρογγυλοποίησης που παριστάνεται με το $\Delta \mathbf{b}$.

Προκαταρκτικά εισάγουμε μία νόρμα για τα διανύσματα x και b και μια αντίστοιχη νόρμα για τον πίνακα A , που δηλώνονται με τους συμβολισμούς $\|x\|$, $\|b\|$ και $\|A\|$. Η μέγιστη από τις νόρμες των στοιχείων ενός διανύσματος είναι μια κατάλληλη νόρμα για τα x και b και δηλώνεται ως $\| \cdot \|_{\infty}$. Η αντίστοιχη νόρμα για τον A είναι το μέγιστο από τα αθροίσματα των νορμών των στοιχείων κάθε γραμμής. Μια άλλη επιλογή είναι το Ευκλείδειο μήκος ενός διανύσματος για τη νόρμα των x και b και δηλώνεται ως $\| \cdot \|_2$.

Το σύνολο των ιδιοτιμών ενός πίνακα καλείται αλλιώς και φάσμα των ιδιοτιμών και η μέγιστη από τις νόρμες όλων των πραγματικών και μιγαδικών ιδιοτιμών καλείται φασματική ακτίνα. Η τετραγωνική ρίζα της φασματικής ακτίνας του Ερμιτιανού και θετικά ορισμένου πίνακα $A^A \cdot A$ είναι η αντίστοιχη νόρμα για τον A (ως A^A θεωρούμε τον αναστροφοσυζυγή πίνακα του A).

Χρησιμοποιώντας τις ιδιότητες των νορμών διανυσμάτων και πινάκων, μπορεί να αποδειχθεί ότι η κλασματική αύξηση $\|\Delta X\|/\|X\|$ μπορεί να κυμανθεί οπουδήποτε μέσα στην περιοχή:

$$\frac{1}{\text{Cond}(A)} \cdot \frac{\|\Delta b\|}{\|b\|} \leq \frac{\|\Delta X\|}{\|X\|} \leq \text{Cond}(A) \cdot \frac{\|\Delta b\|}{\|b\|}$$

Σχέση 2.3.1

Όπου $\text{Cond}(A)$ είναι μια θετική βαθμωτή σταθερά που καλείται αριθμός κατάστασης και ορίζεται ως:

$$\text{Cond}(A) = \|A\| \|A^{-1}\|$$

Σχέση 2.3.2

Για παράδειγμα $\text{Cond}_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = (\lambda_1/\lambda_N)^{1/2}$, όπου λ_1 και λ_N , είναι αντίστοιχα οι ιδιοτιμές του $A \cdot A^T$ με τη μέγιστη και την ελάχιστη νόρμα.

Η τιμή του αριθμού κατάστασης εξαρτάται προφανώς από την επιλογή της νόρμας πίνακα, αλλά είναι πάντα μεγαλύτερος από τη μονάδα, στην πραγματικότητα μπορεί να δειχθεί ότι $\text{Cond}(A) \geq |\lambda_1/\lambda_N|$, όπου λ_1 και λ_N είναι αντίστοιχα, οι ιδιοτιμές του A με τη μέγιστη και ελάχιστη νόρμα. Το δεξιό μέρος της τελευταίας ανισότητας ορίζει ένα κάτω φράγμα που χρησιμοποιείται αρκετές φορές στη θέση του αριθμού κατάστασης.

Η σχέση 2.3.1 φανερώνει ότι η περιοχή της σχετικής διακύμανσης $\|\Delta X\|/\|X\|$ γίνεται πιο πλατιά, καθώς ο αριθμός κατάστασης μεγαλώνει. Έτσι ένας μεγάλος αριθμός κατάστασης τυπικά φανερώνει ότι η λύση είναι ευαίσθητη σε μικρές μεταβολές στο δεξιό μέρος (διάνυσμα b).

Όμοια αποτελέσματα προκύπτουν όταν θεωρούμε τη σχετική διακύμανση της λύσης σε σχέση με τη διατάραξη και του A και του b . Ένα σύστημα ονομάζεται ασθενούς κατάστασης (ill conditioned) όταν η λύση του είναι ευαίσθητη σε μικρές αλλαγές στους όρους των A και b .

Οι ενότητες 2.1, 2.2, 2.3 περιέχουν υλικό που αντλήθηκε από την αναφορά [2].

2.4 Επαναληπτικές μέθοδοι μη σταθερού σημείου – Preconditioner πίνακες

Οι επαναληπτικές μέθοδοι μη σταθερού σημείου διαφέρουν από τις σταθερού σημείου στο γεγονός ότι υπολογισμοί περιέχουν πληροφορία η οποία αλλάζει με κάθε επανάληψη της μεθόδου. Δύο από αυτές τις μεθόδους που ανήκουν στην παραπάνω κατηγορία είναι η μέθοδος Conjugate Gradients' (Μέθοδος Συζυγών Κλίσεων) και η GMRES τις οποίες αναλύουμε παρακάτω.

Ο ρυθμός σύγκλισης των επαναληπτικών μεθόδων εξαρτάται από τις φασματικές ιδιότητες του πίνακα συντελεστών. Ως εκ τούτου μπορεί κανείς να μετατρέψει το γραμμικό σύστημα σε ένα ισοδύναμο, από την άποψη ότι θα έχει την ίδια λύση, αλλά με περισσότερα επιθυμητά χαρακτηριστικά όσον αφορά τις φασματικές του ιδιότητες. Ένας πίνακας προρυθμιστής (preconditioner), είναι ένας πίνακας που επιφέρει τέτοιου είδους μετασχηματισμό.

Για παράδειγμα, αν ένας πίνακας M προσεγγίζει κατά κάποιο τρόπο τον πίνακα συντελεστών το μετασχηματισμένο σύστημα

$$M^{-1} \cdot A \cdot x = M^{-1} \cdot b$$

Σχέση 2.4.1

Έχει την ίδια λύση με το σύστημα :

$$A \cdot x = b$$

Σχέση 2.4.2

ωστόσο οι φασματικές ιδιότητες του πίνακα συντελεστών $M^{-1} \cdot A$ ίσως να είναι περισσότερο επιθυμητές. Κατά τη δημιουργία ενός τέτοιου πίνακα βρισκόμαστε αντιμέτωποι με μια επιλογή εύρεσης ενός πίνακα M ο οποίος προσεγγίζει τον πίνακα A και η επίλυσή του συστήματος με τη χρήση του πίνακα M είναι ευκολότερη απ' ό,τι με τον A , είτε εύρεσης ενός πίνακα M που προσεγγίζει τον A^{-1} , ώστε το μόνο που θα χρειάζεται να είναι ένας πολλαπλασιασμός με τον M . Η πλειονότητα των preconditioners επιλέγονται βάσει της πρώτης περίπτωσης.

2.5 Η μέθοδος Conjugate Gradients

Η μέθοδος των συζυγών κλίσεων αποτελεί μια αποτελεσματική επαναληπτική μέθοδο όταν ο πίνακας του συστήματος το οποίο καλούμαστε να επιλύσουμε είναι συμμετρικός και θετικά ορισμένος. Αποτελεί μάλιστα την παλαιότερη εκ των δύο μεθόδων τις οποίες θα αναλύσουμε. Η μέθοδος αυτή προχωράει παράγοντας ακολουθίες διανυσμάτων μέσω επαναλήψεων (όπως για παράδειγμα τις επιτυχημένες προσεγγίσεις της λύσης), υπόλοιπα (residuals) τα οποία αντιστοιχούν στις επαναλήψεις του αλγορίθμου καθώς και αναζητώντας κατευθύνσεις τις οποίες χρησιμοποιεί στην ανανέωση των επαναλήψεων και των υπολοίπων. Αν και το μήκος των παραπάνω ακολουθιών μπορεί να γίνει πολύ μεγάλο, μόνο ένας μικρός αριθμός διανυσμάτων από αυτά πρέπει να κρατιέται στη μνήμη. Σε κάθε επανάληψη της μεθόδου εκτελούνται δυο πράξεις εσωτερικών γινομένων έτσι ώστε να υπολογιστούν τα ανανεωμένα μονόμετρα μεγέθη τα οποία έχουμε ορίσει ώστε να

εξασφαλίσουμε ότι οι ακολουθίες μας ικανοποιούν συγκεκριμένες καταστάσεις ορθογωνιότητας. Σε ένα συμμετρικό, θετικά ορισμένο γραμμικό σύστημα οι εν λόγω καταστάσεις συνεπάγονται ότι η απόσταση της εκτίμησης από την πραγματική λύση έχει ελαχιστοποιηθεί σε κάποια συγκεκριμένη νόρμα. Ο αλγόριθμος δίνεται παρακάτω:

Αλγόριθμος 3

```

Compute  $r^{(0)} = b - Ax^{(0)}$  for some initial guess  $x^{(0)}$ 
for  $i = 1, 2, \dots$ 
  solve  $Mz^{(i-1)} = r^{(i-1)}$ 
   $\rho_{i-1} = r^{(i-1)T} z^{(i-1)}$ 
  if  $i = 1$ 
     $p^{(1)} = z^{(0)}$ 
  else
     $\beta_{i-1} = \rho_{i-1} / \rho_{i-2}$ 
     $p^{(i)} = z^{(i-1)} + \beta_{i-1} p^{(i-1)}$ 
  endif
   $q^{(i)} = Ap^{(i)}$ 
   $\alpha_i = \rho_{i-1} / p^{(i)T} q^{(i)}$ 
   $x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$ 
   $r^{(i)} = r^{(i-1)} - \alpha_i q^{(i)}$ 
  check convergence; continue if necessary
end

```

Εικόνα 3. Preconditioned Μορφή της Conjugate Gradients Μεθόδου. Πηγή [8]

Το διάνυσμα προσέγγισης λύσης $x^{(i)}$ ανανεώνεται σε κάθε επανάληψη κατά ένα πολλαπλάσιο (α_i) του διανύσματος της διεύθυνσης αναζήτησης $p^{(i)}$:

$$x^{(i)} = x^{(i-1)} + \alpha_i \cdot p^{(i)}$$

Σχέση 2.5.1

Αντίστοιχα τα υπόλοιπα $r^{(i)} = b - A \cdot x^{(i)}$ ανανεώνονται ως εξής:

$$r^{(i)} = r^{(i-1)} - \alpha_i \cdot q^{(i)}, \text{ όπου } q^{(i)} = A \cdot p^{(i)}$$

Σχέση 2.5.2

Η επιλογή $\alpha = \alpha_i = r^{(i-1)T} \cdot r^{(i-1)} / p^{(i)T} \cdot A \cdot p^{(i)}$ ελαχιστοποιεί την ποσότητα $r^{(i-1)T} \cdot A^{-1} \cdot r^{(i)}$, για όλες τις πιθανές επιλογές του α στη σχέση 2.5.2.

Οι διευθύνσεις αναζήτησεως ανανεώνονται με χρήση των υπολοίπων ως εξής:

$$p^{(i)} = r^{(i)} + \beta_{i-1} \cdot p^{(i-1)}$$

Σχέση 2.5.3

όπου η επιλογή $\beta_i = r^{(i)T} \cdot r^{(i)} / r^{(i-1)T} \cdot r^{(i-1)}$, εξασφαλίζει ότι τα $p^{(i)}$ και $A \cdot p^{(i)}$ – ή αντίστοιχα τα $r^{(i)}$ και $r^{(i-1)}$ – είναι ορθογώνια. Πρακτικά, μπορεί κάποιος να δείξει ότι η επιλογή του β_i κάνει τα $p^{(i)}$ και $r^{(i)}$ ορθογώνια σε όλα τα προηγούμενα $A \cdot p^{(j)}$ και $r^{(j)}$ αντίστοιχα.

Ο Αλγόριθμος 3 χρησιμοποιεί έναν πίνακα preconditioner M . Θέτοντας $M = I$ μπορούμε να πάρουμε τον αλγόριθμο της μεθόδου συζυγών κλίσεων χωρίς πίνακα preconditioner. Στην περίπτωση αυτή παρακάμπτουμε την γραμμή 3 του Αλγορίθμου 3 και αντικαθιστούμε τον όρο $z^{(i-1)}$ με τον $r^{(i-1)}$ (και τον όρο $z^{(0)}$ με τον $r^{(0)}$).

Η μορφή της μεθόδου χωρίς τον πίνακα preconditioner κατασκευάζει στην i -οστή επανάληψη την εκτίμηση λύσης $x^{(i)}$ σαν στοιχείο του $x^{(0)} + \text{span}\{r^{(0)}, \dots, A^{i-1} \cdot r^{(0)}\}$, έτσι ώστε η ποσότητα $(x^{(i)} - \hat{x})^T \cdot A \cdot (x^{(i)} - \hat{x})$ ελαχιστοποιείται, όπου \hat{x} είναι η ακριβής λύση του συστήματος $A \cdot x = b$. Εγγύηση της ύπαρξης του ανωτέρω ελαχίστου αποτελεί στη γενική περίπτωση, το γεγονός ότι ο A είναι συμμετρικός και θετικά ορισμένος. Αντίθετα η μορφή της μεθόδου με τον πίνακα preconditioner χρησιμοποιεί διαφορετικό υποχώρο για την κατασκευή των προσεγγίσεων λύσης, αλλά ικανοποιεί την ίδια ιδιότητα ελαχιστοποίησης παρόλη τη χρήση διαφορετικού υποχώρου. Απαιτεί επιπρόσθετα να είναι και ο πίνακας M συμμετρικός και θετικά ορισμένος.

Η παραπάνω ελαχιστοποίηση του σφάλματος είναι ταυτόσημη με το να είναι η ποσότητα $r^{(i)} = b - A \cdot x^{(i)}$ ορθογώνια με τον M^{-1} (γεγονός που εκφράζεται ύπαρξη του μηδενικού εσωτερικού τους γινομένου $r^{(i)T} \cdot M^{-1} \cdot r^{(i)} = 0$, εάν $i \neq j$). Από τη στιγμή που για συμμετρικό πίνακα A μια ορθογώνια βάση για τον υποχώρο Krylov $\text{span}\{r^{(0)}, \dots, A^{i-1} \cdot r^{(0)}\}$ δύναται να κατασκευαστεί με μόνο τριών όρων επαναλήψεις, από τη στιγμή που κάθε επανάληψη ικανοποιεί ταυτόχρονα και την παραγωγή των υπολοίπων.

2.5.1 Σύγκλιση

Είναι δύσκολο να γίνουν ακριβείς προβλέψεις σχετικά με τη σύγκλιση ή μη των επαναληπτικών μεθόδων, ωστόσο μπορούν να εξαχθούν χρήσιμα όρια. Για τη μέθοδο των συζυγών κλίσεων το σφάλμα μπορεί να οριοθετηθεί βάσει του φασματικού αριθμού κατάστασης κ_2 του πίνακα $M^{-1} \cdot A$. Αν λ_{max} και λ_{min} είναι η μέγιστη και η ελάχιστη ιδιοτιμή ενός πίνακα B τότε ο φασματικός αριθμός κατάστασης του B ορίζεται ως $\kappa_2(B) = \lambda_{max}(B) / \lambda_{min}(B)$. Εάν η ακριβής λύση του συστήματος $A \cdot x = b$ παριστάνεται από το διάνυσμα \hat{x} και ο πίνακας A είναι συμμετρικός και θετικά ορισμένος, τότε με τη χρήση ενός επίσης συμμετρικού και θετικά ορισμένου πίνακα preconditioner M , μπορούμε να δείξουμε ότι:

$$\|x^{(i)} - \hat{x}\|_A \leq 2 \cdot a^i \|x^{(0)} - \hat{x}\|_A$$

Σχέση 2.5.4

όπου:

$$a = \frac{(\sqrt{k_2} - 1)}{(\sqrt{k_2} + 1)} \quad \|y\|_A^2 \equiv (y, A \cdot y)$$

Σχέση 2.5.5

Από την παραπάνω σχέση (από Golub και Van Loan [3] και Kaniel [4]) παρατηρούμε ότι ο αριθμός των επαναλήψεων για να πετύχουμε μια σχετική μείωση του σφάλματος μηχανής ϵ στο σφάλμα είναι ανάλογη της τετραγωνικής ρίζας του φασματικού αριθμού κατάστασης.

2.6 Η μέθοδος GMRES

Η μέθοδος Γενικευμένου Ελάχιστου Υπολοίπου (Generalized Minimum Residual Method) δημιουργήθηκε για την επίλυση μη συμμετρικών γραμμικών συστημάτων (από Saad και Schultz [5]). Η πιο διαδεδομένη μορφή της εν λόγω μεθόδου βασίζεται στην παραλλαγμένη διαδικασία Gram-Schmidt, και χρησιμοποιεί επανεκκινήσεις για να ελέγξει τις απαιτήσεις αποθήκευσης.

Εάν δε χρησιμοποιηθούν επανεκκινήσεις η μέθοδος GMRES θα συγκλίνει σε – το πολύ – n βήματα (υποθέτοντας την ικανότητα τέλεσης πράξεων ακρίβειας, χωρίς σφάλμα). Ωστόσο το παραπάνω δεν έχει καμία πρακτική αξία όταν το n είναι μεγάλο. Επιπρόσθετα οι απαιτήσεις σε υπολογισμούς και αποθηκευτικό χώρο κατά την απουσία επανεκκινήσεων, καθιστούν τη μέθοδο απαγορευτική. Πράγματι, το κρίσιμο στοιχείο για την επιτυχημένη εφαρμογή της GMRES(m), περιστρέφεται γύρω από την απόφαση για το πότε η μέθοδος πρέπει να επανεκκινήσει· δηλαδή την ορθή επιλογή του m . Δυστυχώς υπάρχουν παραδείγματα για τα οποία η μέθοδος καθυστερεί και η σύγκλιση επιτυγχάνεται μόνο κατά το n -οστό βήμα. Για τέτοια συστήματα οποιαδήποτε επιλογή του m , μικρότερη από n οδηγεί σε αδυναμία σύγκλισης.

Οι Saad και Schultz [5] απέδειξαν ορισμένα χρήσιμα αποτελέσματα. Πιο συγκεκριμένα έδειξαν ότι εάν ο πίνακας συντελεστών A είναι πραγματικός και «σχεδόν» θετικά ορισμένος, τότε υπάρχει κάποια λογική τιμή του m η οποία πρέπει να επιλεγεί. Οι επιπτώσεις της επιλογής αυτής του m συζητιούνται παρακάτω

2.6.1 Υλοποίηση

Μια κοινή υλοποίηση της μεθόδου προτάθηκε από τους Saad και Schultz [5] και στηρίζεται στη χρήση μετασχηματισμένης ορθογωνιοποίησης Gram-Schmidt. Για την παραπάνω μέθοδο έχουν προταθεί και μετασχηματισμοί Householder οι οποίοι ωστόσο είναι σχετικά δαπανηροί. Η προσέγγιση των μετασχηματισμών Householder αυξάνει την απαιτούμενη εργασία, ωστόσο η σύγκλιση μπορεί να είναι καλύτερη, ειδικά για συστήματα ασθενούς κατάστασης (ill-conditioned) (από Walker [6]). Από τη σκοπιά του παραλληλισμού η ορθογωνιοποίηση Gram – Schmidt μπορεί να θεωρηθεί ελκυστικότερη από την άποψη ότι παρέχει και κάποια σταθερότητα σε ιδιότητες παραλληλοποίησης της μεθόδου(από Demel, Heath και Van der Vorst [7]). Η προσέγγιση τροποποιημένης Gram-Schmidt μεθόδου παρουσιάζεται παρακάτω:


```

 $x^{(0)}$  is an initial guess
for  $j = 1, 2, \dots$ 
  Solve  $r$  from  $Mr = b - Ax^{(0)}$ 
   $v^{(1)} = r/\|r\|_2$ 
   $s := \|r\|_2 e_1$ 
  for  $i = 1, 2, \dots, m$ 
    Solve  $w$  from  $Mw = Av^{(i)}$ 
    for  $k = 1, \dots, i$ 
       $h_{k,i} = (w, v^{(k)})$ 
       $w = w - h_{k,i}v^{(k)}$ 
    end
     $h_{i+1,i} = \|w\|_2$ 
     $v^{(i+1)} = w/h_{i+1,i}$ 
    apply  $J_1, \dots, J_{i-1}$  on  $(h_{1,i}, \dots, h_{i+1,i})$ 
    construct  $J_i$ , acting on  $i$ th and  $(i + 1)$ st component
    of  $h_{\cdot,i}$ , such that  $(i + 1)$ st component of  $J_i h_{\cdot,i}$  is 0
     $s := J_i s$ 
    if  $s(i + 1)$  is small enough then (UPDATE( $\tilde{x}, i$ ) and quit)
  end
  UPDATE( $\tilde{x}, m$ )
end

```

In this scheme UPDATE(\tilde{x}, i)
replaces the following computations:

```

Compute  $y$  as the solution of  $Hy = \tilde{s}$ , in which
the upper  $i \times i$  triangular part of  $H$  has  $h_{i,j}$  as
its elements (in least squares sense if  $H$  is singular),
 $\tilde{s}$  represents the first  $i$  components of  $s$ 
 $\tilde{x} = x^{(0)} + y_1 v^{(1)} + y_2 v^{(2)} + \dots + y_i v^{(i)}$ 
 $s^{(i+1)} = \|b - A\tilde{x}\|_2$ 
if  $\tilde{x}$  is an accurate enough approximation then quit
else  $x^{(0)} = \tilde{x}$ 

```

Εικόνα 4. Preconditioned Μορφή της GMRES(m) Μεθόδου. Πηγή [8]

Το μεγαλύτερο μειονέκτημα της μεθόδου GMRES είναι ότι ο μεγάλος φόρτος εργασίας και αποθηκευτικού χώρου που απαιτείται για κάθε επανάληψη, αυξάνεται γραμμικά με κάθε νέα επανάληψη. Μόνο αν κάποιος είναι αρκετά τυχερός ώστε να πετύχει εξαιρετικά γρήγορη σύγκλιση μπορεί να αποφύγει την ταχεία αύξηση του κόστους που κάνει τη μέθοδο απαγορευτική. Ο συνηθισμένος τρόπος για να ξεπερνάμε αυτόν τον περιορισμό είναι με τη χρήση των επανεκκινήσεων κάθε επανάληψης. Μετά την επιλογή του αριθμού m των επαναλήψεων, τα συσσωρευμένα δεδομένα απομακρύνονται και τα ενδιάμεσα αποτελέσματα χρησιμοποιούνται ως αρχικά δεδομένα για τις επόμενες επαναλήψεις.

Η διαδικασία αυτή επαναλαμβάνεται μέχρι την επίτευξη σύγκλισης. Η δυσκολία έγκειται στην κατάλληλη επιλογή της τιμής του m . Αν η τιμή αυτή είναι πολύ μικρή η $GMRES(m)$ είναι πιθανό να συγκλίνει αργά ή να αδυνατεί να συγκλίνει εξ' ολοκλήρου. Για τιμή του m που είναι μεγαλύτερη από την απαιτούμενη προσθέτουμε στους υπολογισμούς μας επιπλέον φόρτο (και χρήση μεγαλύτερου αποθηκευτικού χώρου). Δυστυχώς δεν υπάρχουν σαφείς κανόνες που διέπουν την επιλογή του m για επανεκκίνηση, αλλά στηρίζεται σε εμπειρικούς κανόνες.

Οι ενότητες 2.4, 2.5 και 2.6 περιέχουν υλικό που αντλήθηκε από την αναφορά [\[8\]](#).

Κεφάλαιο 3

GridLAB-D

3.1 Εισαγωγή στο GridLAB-D

Η πλατφόρμα GridLAB-D αποτελεί ένα εργαλείο προσομοίωσης και ανάλυσης του συστήματος διανομής ηλεκτρικής ενέργειας, που παρέχει πληροφορίες στους χρήστες που ασχολούνται με τη σχεδίαση και λειτουργία των συστημάτων διανομής, καθώς επίσης και σε επιχειρήσεις και οργανισμούς που επιθυμούν να επωφεληθούν από τις νέες ενεργειακές τεχνολογίες. Ενσωματώνει προηγμένες τεχνικές μοντελοποίησης με χρήση υψηλής απόδοσης αλγορίθμων, ώστε να παραδώσει εν τέλει το βέλτιστο στην τελική χρήση μοντέλων. Αναπτύχθηκε το εργαστήριο Pacific Northwest National Laboratory σε συνεργασία με τη βιομηχανία και τον ακαδημαϊκό κόσμο κάτω από τη χρηματοδότηση του Υπουργείου Ενέργειας των Η.Π.Α. Τα παραπάνω αναφέρονται στο επίσημο site του GridLAB-d [\[11\]](#). Το GridLAB-D έχει σχεδιαστεί ως εργαλείο ανοιχτού κώδικα σε γλώσσα προγραμματισμού C++ και είναι ελεύθερα διαθέσιμο σε κάθε χρήστη που θέλει να τρέξει προσομοιώσεις πάνω σε ένα σύστημα διανομής.

Η περιγραφή του δικτύου γίνεται μέσω κατάλληλης για ανάγνωση από το εργαλείο script γλώσσα. Τα αρχεία αυτά περνάνε ως ορίσματα στην πλατφόρμα μέσω της γραμμής εντολών του GridLAB-D, GridLAB-D Console και έχουν κατάληξη .glm αφού μετακινηθούμε στο directory που βρίσκεται το επιθυμητό .glm αρχείο και ξεκινάμε την προσομοίωση πληκτρολογώντας την εντολή gridlabd <file_name>.glm

Στη συνέχεια τρέχουμε ορισμένα πειράματα για την περιγραφή ενός απλού δικτύου εκτελώντας αλλαγές κάθε φορά στο τρέχων δίκτυο και εξετάζοντας-ερμηνεύοντας κάθε φορά τα παραγόμενα αποτελέσματα εξόδου που μας παρέχει το εργαλείο:

```
// $id$
// Copyright (C) 2008 Battelle Memorial Institute

clock {
    starttime '2010-01-21 00:00:00';
    stoptime '2010-01-22 00:00:00';
    timezone EST+5EDT;
}

module powerflow
{
    solver_method NR;
    NR_iteration_limit 150;
};
module tape;

object overhead_line_conductor:100 {
    geometric_mean_radius 0.0244;
    resistance 0.306;
}

object overhead_line_conductor:101 {
    geometric_mean_radius 0.00814;
    resistance 0.592;
}
```

```

object line_spacing:200 {
    distance_AB 2.5;
    distance_BC 4.5;
    distance_AC 7.0;
    distance_AN 5.656854;
    distance_BN 4.272002;
    distance_CN 5.0;
}

object line_configuration:300 {
    conductor_A overhead_line_conductor:100;
    conductor_B overhead_line_conductor:100;
    conductor_C overhead_line_conductor:100;
    conductor_N overhead_line_conductor:101;
    spacing line_spacing:200;
}

object transformer_configuration:400 {
    connect_type DELTA_DELTA;
    install_type PADMOUNT;
    power_rating 7000;
    primary_voltage 12470;
    secondary_voltage 4160;
    resistance 0.01;
    reactance 0.06;
}

object node:1 {
    phases "ABCD";
    bustype SWING;
    voltage_A +7199.558+0.000j;
    voltage_B -3599.779-6235.000j;
    voltage_C -3599.779+6235.000j;
    nominal_voltage 7200;
}

object overhead_line:12 {
    phases "ABCD";
    from node:1;
    to node:2;
    length 2000;
    configuration line_configuration:300;

    object recorder {
        file line12.csv;
        interval 900; //900s= 15min
        property current_out_C,
current_out_A, current_out_B;
    };
}

object node:2 {
    phases "ABCD";
    voltage_A +7199.558+0.000j;
    voltage_B -3599.779-6235.000j;
    voltage_C -3599.779+6235.000j;
    nominal_voltage 7200;

    object recorder {
        file node2.csv;
        interval 900; //900s= 15min
        property voltage_C, voltage_A,
voltage_B;
    };
}

```

```

object transformer:23 {
    phases "ABCD";
    from node:2;
    to node:3;
    configuration transformer_configuration:400;
}

object node:3 {
    phases "ABCD";
    voltage_A +2401.777+0.000j;
    voltage_B -1200.889-2080.000j;
    voltage_C -1200.889+2080.000j;
    nominal_voltage 2400;
    object recorder {
        file node3.csv;
        interval 900; //900s= 15min
        property voltage_C, voltage_A, voltage_B;
    };
}

object overhead_line:34 {
    phases "ABCD";
    from node:3;
    to load:4;
    length 2500;
    configuration line_configuration:300;
}

object load:4 {
    phases "ABCD";
    voltage_A +2401.777+0.000j;
    voltage_B -1200.889-2080.000j;
    voltage_C -1200.889+2080.000j;
    constant_power_A +1800000.000+871779.789j;
    constant_power_B +1800000.000+871779.789j;
    constant_power_C +1800000.000+871779.789j;
    maximum_voltage_error 0.001;
    nominal_voltage 2400;
}

```

3.2 Πειράματα προσομοιώσεων

1^ο Πείραμα

Το παραπάνω αρχείο αποτελεί παράδειγμα δικτύου με 4 κόμβους και επιλέχθηκε για πρώτο τρέξιμο χάριν απλότητας. Η προκαθορισμένη μέθοδος επίλυσης είναι η μέθοδος Newton - Raphson. Το πείραμα έχει υλοποιηθεί να επιλύσει το πρόβλημα για διάρκεια μίας ημέρας και ανά 15'.

Σαν έξοδο, έχουμε ορίσει τις τάσεις `voltage_A`, `voltage_B` και `voltage_C` οι οποίες αναπαριστούν τις τάσεις στις φάσεις A, B και C του συστήματος. Εκτυπώνουμε τα αποτελέσματα τόσο για τον κόμβο 2 όσο και για τον κόμβο 3 του συστήματος στα αρχεία `node2_a` και `node3_a` αντίστοιχα. Αντίστοιχα για τη γραμμή που συνδέει τους κόμβους 1, 2 έχουμε ορίσει να εκτυπώνεται το ρεύμα των τριών φάσεων A, B και C το οποίο εμφανίζεται στο αρχείο `line12_a`.

Το αποτέλεσμα των πειραμάτων έδειξε ότι η τάση στους κόμβους 2 και 3 όπως επίσης και το ρεύμα στη γραμμή διασύνδεσης των 1 και 2 παραμένουν χρονικά αμετάβλητα. Μια πιθανή εξήγηση για το γεγονός αποτελεί η μη μεταβολή του τρέχοντος φορτίου (το οποίο ορίζεται ως σταθερό στον κόμβο 4) κατά τη διάρκεια εκτέλεσης του πειράματος.

Για να παρατηρήσουμε αλλαγές στην έξοδο βάσει της εισόδου τρέχουμε το πείραμα μεταβάλλοντας το `.glm` αρχείο.

2° Πείραμα

Στη συνέχεια εκτελούμε την εξής μεταβολή στο `object transformer_configuration:400`. Αλλάζουμε την τιμή της μεταβλητής `power_rating 7000` σε `power_rating 8000` και ξανά-εκτελούμε προσομοίωση. Το μέγεθος αυτό καθορίζει την ισχύ την οποία μπορεί να προσδίδει ως έξοδο ο εν λόγω μετασχηματιστής.

Τα αποτελέσματα του συγκεκριμένου πειράματος φαίνονται στα αρχεία `node2_b`, `node3_b` και `line12_b` που παράγονται ως έξοδος. Πιο συγκεκριμένα φέρνοντας σε αντιπαράθεση τα αρχεία `node2_a` και `node2_b` παρατηρούμε μια ελάχιστη έως και μηδαμινή μεταβολή στις φάσεις των τριών τάσεων. Αντίστοιχα στα αρχεία `node3_a` και `node3_b` η τάση στη φάση C από $2277.5 \angle 116.84^\circ$ σε $2291.5 \angle 117.20^\circ$, στη φάση A από $2272.8 \angle -3.0382^\circ$ σε $2286.7 \angle -2.6803^\circ$ και στη φάση B από $2279.5 \angle 236.89^\circ$ σε $2293.4 \angle 237.25^\circ$.

Από τα παραπάνω αποτελέσματα συμπεραίνουμε ότι ενώ η σχετική διαφορά των ορισμάτων μεταξύ των φάσεων παραμένει σταθερή, τα μέτρα της τάσης και στις τρεις φάσεις αυξάνονται. Το παραπάνω γεγονός μπορεί να εξηγηθεί από το ότι αφού προσδίδουμε στο σύστημα περισσότερη ισχύ P , βάσει του τύπου $P=V^2/R$ και για σταθερή αντίσταση R του δικτύου θα έχουμε αναγκαστικά αύξηση και του μέτρου των τάσεων.

Συνεχίζοντας το πείραμα και θέλοντας να αυξήσουμε κατά πολύ την παρεχόμενη από το μετασχηματιστή ισχύ θέτουμε `power_rating 10000`.

Παρατηρούμε ότι τόσο ο μετασχηματιστής όσο και η γραμμή λειτουργούν σε κάθε τρέξιμο πάνω από το 100% της ονομαστικής τους ισχύος και για το λόγο αυτό τα μηνύματα που εμφανίζονται στην κονσόλα αναφέρονται ως “WARNINGS”, καθώς όπως είναι λογικό μπορεί να προκύψει δυσλειτουργία στο δίκτυο από τη λειτουργία των εξαρτημάτων του σε ισχύ μεγαλύτερη της ονομαστικής. Παράλληλα εκτυπώνεται και σφάλμα από την επίλυση του προβλήματος ροής φορτίου. Το σφάλμα αυτό μας ενημερώνει ότι η προσομοίωση σταματάει επειδή η μέθοδος Newton-Raphson αποτυγχάνει, λόγω του άνω ορίου των 150 επαναλήψεων που έχουμε θέσει στο αρχείο, καθώς αδυνατεί να συγκλίνει σε αυτό τον αριθμό επαναλήψεων.

3ο Πείραμα

Ως τρίτο πείραμα επιλέχτηκε η αύξηση του μήκους της γραμμής μεταξύ των κόμβων 1 και 2 ώστε να παρατηρήσουμε τις επιπτώσεις που θα έχει ένας δεκαπλασιασμός του μήκους της συγκεκριμένης γραμμής στην πτώση τάσης του κόμβου 2. Για το λόγο αυτό αλλάζουμε την `length 2000` σε `length 20000` για το `object overhead_line:12`.

Λόγω της ύπαρξης ισχύος πάνω από την ονομαστική τιμή γραμμής στις φάσεις A και B αυτή τη φορά, το πείραμα ουσιαστικά δε μπορεί να εκτελεστεί όπως και προηγουμένως.

Στη συνέχεια μειώνουμε το μήκος της γραμμής σε 2010 m, δηλαδή σε μήκος του αρχικού αυξημένο κατά 10 m. Το γεγονός ότι αυξήσαμε το μήκος της γραμμής ισοδυναμεί με την

αύξηση της αντίστασης που συναντάει το ρεύμα στη γραμμή μεταξύ κόμβου 1 και 2 γεγονός που έχει ως επακόλουθη πτώση τάσης και στις 3 φάσεις για τον κόμβο 2.

Πράγματι εξετάζοντας τα παραγόμενα αρχεία `node2_a` και `node2_c` παρατηρούμε ότι για τη φάση C από $7126.2 \angle 119.59^\circ$ η τάση έπεσε σε $7125.8 \angle 119.59^\circ$ στη φάση A η τάση από $7112.5 \angle 0.28872^\circ$ έπεσε σε $7112 \angle 0.28872^\circ$ και στη φάση B από $7132.2 \angle 239.64^\circ$ σε $7131.8 \angle 239.64^\circ$.

Ενδιαφέρον παρατηρεί το ότι παρόλο το γεγονός ότι το μέτρο των τάσεων των A, B και C τάσεων μεταβάλλεται, η γωνία τους παραμένει σταθερή τόσο πριν όσο και μετά την αύξηση του μήκους της γραμμής. Το παραπάνω συμβαίνει καθώς εισάγουμε στο σύστημα με την παραπάνω ενέργεια, πραγματική-ωμική αντίσταση και όχι χωρητική ή επαγωγική οι οποίες θα είχαν ως αποτέλεσμα την αύξηση και μείωση της γωνίας αντίστοιχα.

3.3 Ανάλυση – Επέκταση πηγαίου κώδικα του GridLAB-D

Ως επόμενο στάδιο της παρούσας εργασίας έχουμε την ανάλυση - επέκταση της πλατφόρμας προσομοίωσης, στο τμήμα που εκτελεί τους υπολογισμούς για τη ροή φορτίου. Το εν λόγω τμήμα του λογισμικού ονομάζεται `powerflow`. Ο πηγαίος κώδικας αυτού του τμήματος βρίσκεται στο `PROJECT_DIR\gridlab-d-code-5028-trunk\powerflow`, όπου θεωρούμε ως `PROJECT_DIR` το directory αποθήκευσης του πηγαίου κώδικα του GridLAB-D. Επικεντρωνόμαστε στα αρχεία `solver_nr.cpp` και `solver_nr.h`. Στα αρχεία αυτά ορίζεται η μέθοδος Newton - Raphson για τον υπολογισμό της ροής φορτίου στο δίκτυο μέσω της επίλυσης ενός γραμμικού και αραιού συστήματος $A \cdot x = b$.

Στόχος μας αποτελεί ο εντοπισμός της συνάρτησης `superLU` η οποία καλείται μέσα στο αρχείο ώστε να προβεί στην επίλυση του συστήματος που έχει δημιουργήσει η μέθοδος Newton – Raphson. Τους πίνακες που περνιούνται στην `SuperLU` ως παράμετροι θα τους χρησιμοποιήσουμε ως εισόδους στον νέο επιλυτή (`solver`) συστημάτων που θα δημιουργήσουμε στο επόμενο κεφάλαιο ώστε να μελετήσουμε τη συμπεριφορά των επαναληπτικών μεθόδων σε συστήματα που προκύπτουν από τη μοντελοποίηση του ενεργειακού δικτύου. Η εν λόγω συνάρτηση καλείται στη γραμμή 4846 του αρχείου `solver_nr.cpp` ως εξής:

```
pdgssv(NR_superLU_procs, &A_LU, perm_c, perm_r, &L_LU, &U_LU, &B_LU, &info);
```

Η εν λόγω συνάρτηση αποτελεί τμήμα της βιβλιοθήκης **Opensees**, την οποία μπορεί κανείς να κατεβάσει από τον παρακάτω σύνδεσμο:

<http://opensees.berkeley.edu/>

Αναλυτικότερα στο `documentation` της βιβλιοθήκης γίνεται αναφορά στα ορίσματα με τα οποία καλείται η συνάρτηση και παρατίθενται παρακάτω:

1. Ως πρώτο όρισμα περνιέται `NR_superLU_procs`, το οποίο αποτελεί τον αριθμό των νημάτων (`threads`) που θέλουμε να χρησιμοποιήσει η `superLU` ώστε να μας υπολογίσει τη λύση. Στο GridLAB-D ο αριθμός αυτός αρχικοποιείται σε 1 στο αρχείο `powerflow.h` σειρά 59 με την παρακάτω δήλωση:

```
59 GLOBAL int NR_superLU_procs INIT(1);
```

2. Οι πίνακες περνιούνται στην κλήση της συνάρτησης ως structs SuperMatrix της οποίας ο ορισμός φαίνεται παρακάτω:

```
typedef struct {
45  Stype_t Stype; /* Storage type: interprets the storage structure
46  pointed to by *Store. */
47  Dtype_t Dtype; /* Data type. */
48  Mtype_t Mtype; /* Matrix type: describes the mathematical property of
49  the matrix. */
50  int_t nrow; /* number of rows */
51  int_t ncol; /* number of columns */
52  void *Store; /* pointer to the actual storage of the matrix */
53 } SuperMatrix;
```

Ο *A_LU* που είναι και δεύτερο όρισμα της συνάρτησης, αρχικοποιείται με τα εξής στοιχεία στο αρχείο *solver_nr.cpp*:

```
4653 A_LU.Store = (void *)gl_malloc(sizeof(NCformat));
4660 A_LU.Stype = SLU_NC;
4661 A_LU.Dtype = SLU_D;
4662 A_LU.Mtype = SLU_GE;
4663 A_LU.nrow = n;
4664 A_LU.ncol = m;
```

Το δεξί διάνυσμα αποτελεί το 7^ο όρισμα της συνάρτησης και αρχικοποιείται με τα εξής στοιχεία στο αρχείο *solver_nr.cpp*:

```
4653 B_LU.Store = (void *)gl_malloc(sizeof(DNformat));
4660 B_LU.Stype = SLU_DN;
4661 B_LU.Dtype = SLU_D;
4662 B_LU.Mtype = SLU_GE;
4663 B_LU.nrow = m;
4664 B_LU.ncol = 1;
```

Το *SLU_NC* αναφέρεται στο ότι ο πίνακας *A_LU* είναι αποθηκευμένος ως αραιός (sparse) ενώ το *SLU_DN* στο ότι ο πίνακας *B_LU* είναι αποθηκευμένος ως πυκνός (dense). Το όρισμα *B_LU* είναι επίσης η δομή αποθήκευσης της λύσης που επιστρέφει η *superLU* μετά την περάτωσή της.

Οι πίνακες *L_LU* και *U_LU* αποθηκεύουν τη διάσπαση του πίνακα *A_LU* σε κάτω και άνω τριγωνικό πίνακα αντίστοιχα με το πέρας της *superLU*, ως επιστρεφόμενοι τύποι από την όλη διαδικασία.

3. Σαν ορίσματα εισόδου περνιούνται και οι δύο πίνακες μεταθέσεων (permutation matrices) έναν για πολλαπλασιασμό του *A_LU* από αριστερά (*perm_c*) και έναν για πολλαπλασιασμό από δεξιά (*perm_r*) τα οποία δηλώνονται ως πίνακες *int* και το *perm_c* υπολογίζεται στη συνάρτηση *get_perm_c()*, ενώ για το *perm_r* δεσμεύεται χώρος *m* θέσεων στο αρχείο *solver_nr.cpp*

```
4724 perm_r = (int *) gl_malloc(m * sizeof(int));
4728 perm_c = (int *) gl_malloc(m * sizeof(int));
```

4. Τέλος υπάρχει και ως τελευταίο όρισμα η μεταβλητή *info* η οποία αποτελεί και τον κωδικό σφάλματος (*int*) για σφάλματα σε κάθε νήμα (thread) της *superLU*.

Στόχο μας, όπως προαναφέρθηκε, αποτελεί η απομόνωση του συστήματος που επιλύει η μέθοδος Newton – Raphson με τη βοήθεια της *superLU* και η εισαγωγή του σε έναν νέο

επιλυτή (solver) που θα υλοποιεί άλλες επαναληπτικές μεθόδους των οποίων τη συμπεριφορά και θα εκτιμήσουμε. Ο τρόπος με τον οποίο θα περάσουμε το σύστημά μας στο νέο επιλυτή θα είναι μέσω της χρήσης ASCII αρχείων. Για το σκοπό αυτό αναπτύχθηκαν δύο νέες συναρτήσεις εγγραφής δομών τύπου SuperMatrix σε αρχεία. Η αναπαράσταση των πινάκων στα αρχεία ASCII ακολουθεί τις παρακάτω συμβάσεις

Για τους αραιούς πίνακες στην πρώτη γραμμή του αρχείου εκτυπώνεται με την εξής σειρά τρεις αριθμοί των οποίων ο συμβολισμός ακολουθεί τον παρακάτω κανόνα:

<Μέγεθος γραμμών πίνακα> <Μέγεθος στηλών πίνακα> <Αριθμός μη μηδενικών στοιχείων πίνακα>

Για τις υπόλοιπες γραμμές του πίνακα ακολουθείται η παρακάτω σύμβαση:

<Γραμμή στοιχείου> <Στήλη στοιχείου> <Τιμή πίνακα στη [Γραμμή στοιχείου , Στήλη στοιχείου]>

Πρέπει να τονιστεί ότι κάθε αναφορά σε στοιχεία του πίνακα, αναφέρεται σε μη μηδενικά στοιχεία στην αντίστοιχη γραμμή και στήλη του πίνακα. Τα μηδενικά στοιχεία δεν εμφανίζονται σε αυτή την αναπαράσταση. Η συνάρτηση που θα μας δίνει την παραπάνω αναπαράσταση, θα παίρνει ως όρισμα τη μοναδική δομή που είναι αποθηκευμένη σε αραιή (sparse) μορφή, η οποία δεν είναι άλλη από την A_LU. Ως δεύτερο όρισμα έχουμε περάσει το όνομα του αρχείου που θα δημιουργηθεί για την παραπάνω αποθήκευση. Ο κώδικας της εν λόγω συνάρτησης εμφανίζεται πιο κάτω

```
void fprintf_sparse_SuperMatrix(char* what, SuperMatrix *A) {
    NCformat *Astore;
    register int i,n,j,cnt=0;
    double *dp;
    FILE *fd;

    fd=fopen(what,"w");
    n = A->ncol;
    Astore = (NCformat *) A->Store;
    dp = (double *) Astore->nzval;
    fprintf(fd,"%d %d %d\n", A->nrow,A->ncol,Astore->nnz);
    for (j=0; j<n-1; j++) {
        for (i=Astore->colptr[j]; i<Astore->colptr[j+1]; i++) {
            cnt++;
            fprintf(fd,"%d %d %e\n", Astore->rowind[i], j, dp[i],cnt);
        }
        //Printing the elements of the last matrix's column
        for(i=Astore->colptr[j]; i<Astore->nnz; i++) {
            cnt++;
            fprintf(fd,"%d %d %e\n", Astore->rowind[i], j, dp[i],cnt);
        }
        fclose(fd);
    }
}
```

Παράλληλα υλοποιήθηκε συνάρτηση εκτύπωσης πυκνών (dense) πινάκων σε αρχεία ASCII καθώς η δομή B_LU που περιέχει αρχικά το διάνυσμα του δεξιού μέλους (πριν την κλήση της superLU) και έπειτα το διάνυσμα της λύσης που μας επιστράφηκε.

Για τους πίνακες αυτούς ακολουθείται ο εξής συμβολισμός για κάθε γραμμή του αρχείου που εκτυπώνεται:

<Γραμμή πίνακα> < Τιμή πίνακα στη [Γραμμή στοιχείου , 1]>

Ο εν λόγω συμβολισμός υιοθετήθηκε τόσο για το δεξί μέλος όσο και για το διάνυσμα λύσης καθώς και τα δύο αποτελούν στην ουσία διανύσματα στήλες με ίδιο αριθμό γραμμών. Η συνάρτηση που θα μας δίνει την παραπάνω αναπαράσταση, θα παίρνει ως όρισμα δομή που είναι αποθηκευμένη σε πυκνή (dense) μορφή, η οποία δεν είναι άλλη από την B_LU. Ως δεύτερο όρισμα έχουμε περάσει το όνομα του αρχείου που θα δημιουργηθεί για την παραπάνω αποθήκευση. Ο κώδικας της εν λόγω συνάρτησης εμφανίζεται παρακάτω:

```
void fprintf_dense_SuperMatrix(char * what, SuperMatrix *B) {
    FILE *fd;
    DNformat *Bstore;
    register int i;
    double *dp;

    fd=fopen(what, "w");
    Bstore = (DNformat *) B->Store;
    dp = (double *) Bstore->nzval;
    for (i = 0; i < B->nrow; ++i) fprintf(fd, "%d %e\n",i, dp[i]);
    fclose(fd);
}
```

Η μορφή αναπαράστασης που ακολουθείται στα αρχεία ASCII αναφέρεται και ως μορφή MATLAB. Τις δύο αυτές συναρτήσεις τις ορίζουμε στο αρχείο solver_nr.cpp. Όπως προαναφέρθηκε για την εξαγωγή των πινάκων A_LU, B_LU σε ASCII αρχεία πρέπει να καλέσουμε ακριβώς πριν από την κλήση της superLU μία φορά `fprintf_sparse_SuperMatrix(char* what, SuperMatrix *A)` με ορίσματα την A_LU και το επιθυμητό όνομα του ASCII αρχείου και μία φορά την `fprintf_dense_SuperMatrix(char * what, SuperMatrix *B)` με ορίσματα τη δομή B_LU και το επιθυμητό όνομα του ASCII αρχείου που θα περιέχει το διάνυσμα του δεξιού μέλους. Ακριβώς μετά την κλήση της superLU καλούμε εκ νέου την συνάρτηση `fprintf_dense_SuperMatrix(char * what, SuperMatrix *B)` πάλι με όρισμα τη δομή B_LU (η οποία πλέον περιλαμβάνει το διάνυσμα λύσης) και το όνομα του αρχείου που θα περιλαμβάνει το διάνυσμα λύσης. Αφού εκτελέσουμε `compile` παίρνουμε το ανανεωμένο `powerflow.dll` που παράγεται κατά τη διαδικασία της μεταγλώττισης και το τοποθετούμε στο `path` με το `.glm` αρχείο για το οποίο θέλουμε να τρέξουμε την προσομοίωση.

Κεφάλαιο 4

Βιβλιοθήκη PETSc – Ανάπτυξη λογισμικού επαναληπτικών μεθόδων

4.1 Εισαγωγή στη βιβλιοθήκη PETSc

Στη συνέχεια και αφού έχουμε εξάγει τα δεδομένα βάσει των οποίων θα εξετάσουμε τις επαναληπτικές μας μεθόδους προχωρούμε στη δημιουργία του λογισμικού που θα υλοποιεί τις μεθόδους αυτές. Η υλοποίηση αυτή πραγματοποιήθηκε με τη χρήση της βιβλιοθήκης PETSc της οποίας τα βασικά χαρακτηριστικά και οι λόγοι που οδήγησαν στην επιλογή της αναλύονται παρακάτω.

Η δημιουργία λογισμικού προσομοίωσης για υπάρχοντα και αναδυόμενα συστήματα ισχύος, αποτελεί πρόκληση καθώς αποτελεί ένα δύσκολο και αρκετά χρονοβόρο έργο. Τη δυσκολία αυτή έρχεται να μετριάσει η χρήση υψηλής απόδοσης βιβλιοθηκών για επιστημονικούς υπολογισμούς, όπως η βιβλιοθήκη PETSc, η οποία έχει δοκιμαστεί σε ένα ευρύ φάσμα επιστημονικών εφαρμογών, από μονοπύρρηνα μηχανήματα έως και υπερυπολογιστές, έχοντας ιδιαίτερα βελτιστοποιημένη υλοποίηση και μια ευρύ σύνολο δοκιμασμένων αριθμητικών επιλυτών (solvers). Η χρήση τέτοιου είδους βιβλιοθηκών για την ανάπτυξη συστημάτων ηλεκτρικής ενέργειας, δεν είναι ακόμα ευρέως διαδεδομένη στην κοινότητα των συστημάτων ισχύος, ωστόσο έχουν χρησιμοποιηθεί επαρκώς σε προσομοιώσεις εφαρμογών που απαιτούν την επίλυση μερικών διαφορικών εξισώσεων.

Η ηλεκτρική βιομηχανία αναπτύσσεται μέσω της εισαγωγής νέων τεχνολογιών και ιδεών, οι οποίες καθιστούν τα ήδη υπάρχοντα δίκτυα περισσότερο ασφαλή και αξιόπιστα και βοηθούν στην ένωση επιμέρους δικτύων ενέργειας σε ένα ενιαίο. Πιο συγκεκριμένα η εισαγωγή των ανανεώσιμων πηγών ενέργειας στα δίκτυα καθώς και των Ευφυών Δικτύων (Smart Grids), δίνουν μια νέα οπτική στον έλεγχο και την επικοινωνία των τμημάτων του ηλεκτρικού δικτύου. Η επέκταση των συστημάτων ηλεκτρικής ενέργειας απαιτεί επίσης έλεγχο σε επίπεδο είτε τοπικό, είτε κρατικό είτε ακόμα και διακρατικό.

Αυτές οι εξελίξεις, καθιστούν τα υπολογιστικά προβλήματα των συστημάτων ισχύος πιο περίπλοκα και με απαιτήσεις για προσομοιώσεις υψηλότερων ταχυτήτων. Ο τελικός στόχος και παράλληλα το βέλτιστο σενάριο για τις εφαρμογές που αφορούν συστήματα ηλεκτρικής ενέργειας θα ήταν να έχουμε μοντέλα υψηλής πιστότητας, τα οποία θα επεξεργάζονται σε πραγματικό χρόνο, ώστε να μας δώσουν κάποια πρόβλεψη για τη συμπεριφορά του δικτύου και να μας οδηγήσουν σε κάποια έγκαιρη απόφαση. Προφανώς για να επιτευχθεί το παραπάνω σενάριο, καθίσταται απαραίτητη η χρήση παράλληλων μηχανημάτων, η οποία θα επιταχύνει σημαντικά τις εν λόγω εφαρμογές.

Για το λόγο αυτό σημαντικό κομμάτι της έρευνας για τα συστήματα ισχύος, έχει γίνει πάνω στην ανάπτυξη παράλληλων αλγορίθμων. Αποτελέσματα της παραπάνω έρευνας αποτέλεσε η ανάπτυξη παράλληλων αλγορίθμων σε προσομοιώσεις ροής φορτίου, μεταβατικής ευστάθειας, εκτίμησης κατάστασης και σε πολλά άλλα πεδία των συστημάτων ισχύος. Η κυριότερη έρευνα ωστόσο έχει γίνει στην περιοχή των εφαρμογών πάνω στη μεταβατική ευστάθεια για την επίλυση του μοντέλου διαφορικών-αλγεβρικών εξισώσεων ενός συστήματος.

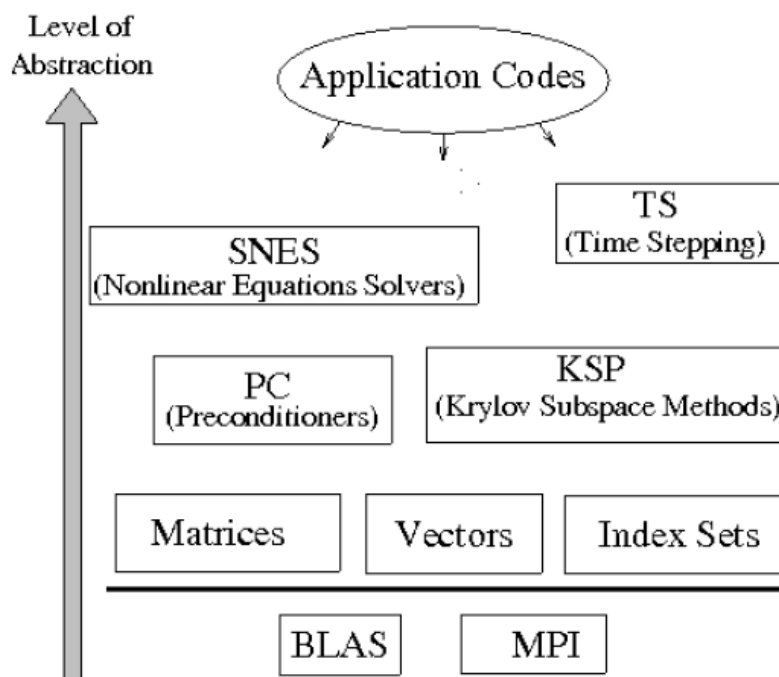
Ωστόσο η ανάπτυξη παράλληλων αλγορίθμων απαιτεί μεγάλο κόστος σε χρόνο και προσπάθεια μέσα από διάφορες φάσεις ανάπτυξης των εφαρμογών, όπως ο διαχωρισμός του έργου προς επεξεργασία, η διαχείριση της επικοινωνίας μεταξύ των επεξεργαστών, την αποσφαλμάτωση (debugging) και την επαλήθευση της ορθότητας του, με τον περισσότερο χρόνο να ξοδεύεται στην ανάπτυξη του γραμμικού επιλυτή (linear solver). Για το λόγο αυτό οι παράλληλες εφαρμογές αναπτύσσονται συνήθως χρησιμοποιώντας ένα συγκεκριμένο

σχεδιάγραμμα γραμμικής λύσης, δοκιμασμένο πάνω σε μια συγκεκριμένη αρχιτεκτονική για δεδομένη τοπολογία συστήματος.

Δεδομένου ότι οι εφαρμογές για δίκτυα ισχύος επόμενης γενιάς εξελίσσονται, είναι απαραίτητη η ύπαρξη συγκριτικής αξιολόγησης μεταξύ των διάφορων αλγορίθμων σε διαφορετικές τοπολογίες του συστήματος ώστε να επιλέξουμε τον βέλτιστο ή ένα σύνολο βέλτιστων αλγορίθμων. Το παραπάνω απαιτεί τον πειραματισμό με διαφορετικούς επιλυτές (solvers) και αποτελεί ένα έργο που συνεπάγεται σημαντικό χρόνο και προσπάθεια. Στην προσπάθεια αυτή έρχεται να βοηθήσει η βιβλιοθήκη PETSc μέσω της μεγάλης γκάμας επιλυτών και της διεπαφής αντικειμένων γραμμικής άλγεβρας για ανάπτυξη λογισμικού εξασφαλισμένης φορητότητας από τους χρήστες ως μια πλατφόρμα για ανάπτυξη εφαρμογών για το δίκτυο ηλεκτρικής ενέργειας.

4.2 Δομή Βιβλιοθήκης PETSc

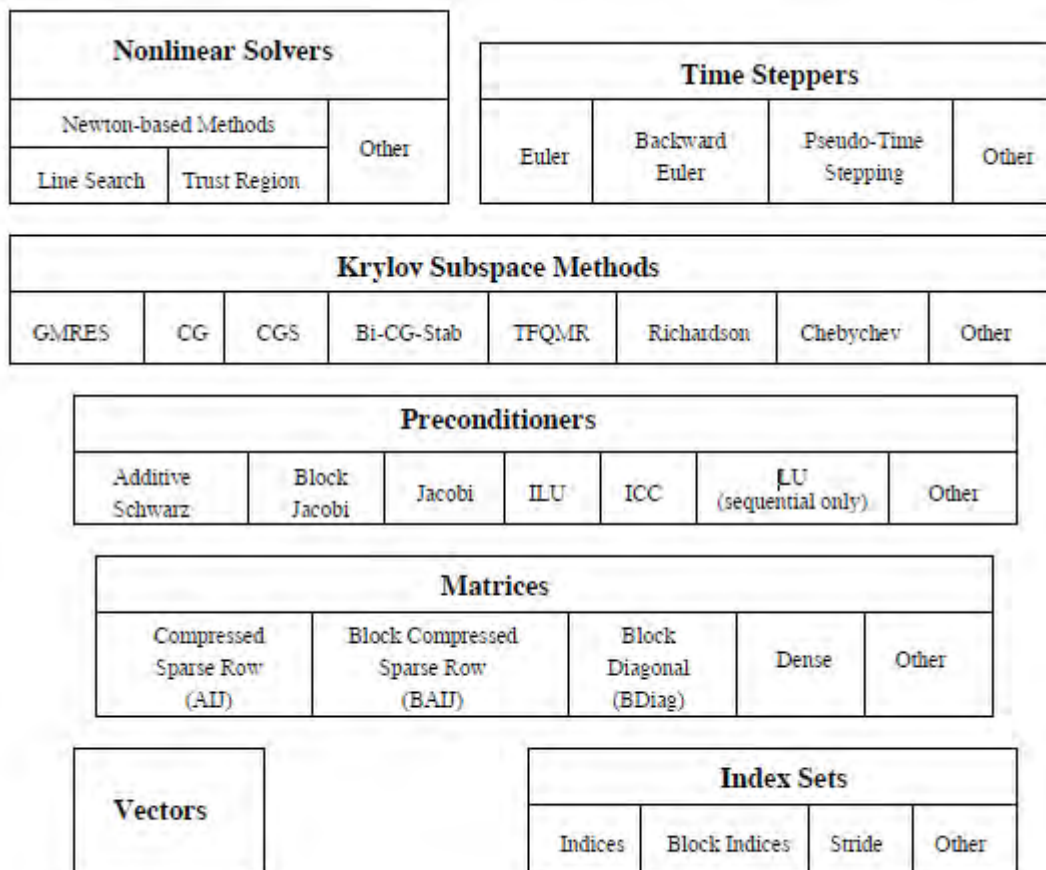
Η βιβλιοθήκη PETSc (Portable Extensible Toolkit for Scientific Computations), αποτελεί μια πλατφόρμα δομών δεδομένων και ρουτινών που παρέχει τα βασικά εργαλεία για τη δημιουργία μεγάλης κλίμακας εφαρμογών είτε σε παράλληλους είτε σε σειριακούς υπολογιστές ως προς τον τρόπο επεξεργασίας. Η βιβλιοθήκη αυτή χρησιμοποιεί το πρότυπο επικοινωνίας MPI για όλες τις επικοινωνίες τύπου ανταλλαγής μηνυμάτων. Παράλληλα περιλαμβάνει μια σουίτα διευρυνόμενων παράλληλων γραμμικών και μη γραμμικών επιλυτών που μπορούν να χρησιμοποιηθούν σε πηγαίο κώδικα εφαρμογών υλοποιημένων σε διάφορες γλώσσες προγραμματισμού, όπως *Fortran*, *C*, *C++*, *Python* και *Matlab*. Επίσης παρέχει πολλούς από τους μηχανισμούς που απαιτούνται μέσα στον πηγαίο κώδικα παράλληλων εφαρμογών, όπως για παράδειγμα ρουτίνες παράλληλης κατασκευής ενός πίνακα ή ενός διανύσματος. Η βιβλιοθήκη οργανώνεται βάσει μιας ιεραρχίας η οποία επιτρέπει στους χρήστες της να χρησιμοποιούν το επίπεδο αφαίρεσης το οποίο είναι περισσότερο κατάλληλο κάθε φορά για κάποιο συγκεκριμένο πρόβλημα. Παράλληλα η χρήση τεχνικών αντικειμενοστραφούς προγραμματισμού παρέχει στους χρήστες τεράστια ευελιξία. Η εν λόγω ιεραρχία απεικονίζεται παραστατικά βάσει του επιπέδου αφαιρετικότητας των στοιχείων στην παρακάτω εικόνα:



Εικόνα 5. Οργάνωση της Βιβλιοθήκης PETSc. Πηγή [9]

Η PETSc αποτελείται από μια ποικιλία επιμέρους βιβλιοθηκών (παρόμοια με το μοντέλο των κλάσεων που χρησιμοποιεί η C++). Κάθε μία από τις επιμέρους αυτές βιβλιοθήκες χειρίζεται μια συγκεκριμένη ομάδα αντικειμένων (όπως για παράδειγμα τα διανύσματα), μαζί με τις συναρτήσεις – λειτουργίες που θα μπορούσε να εφαρμόσει κανείς πάνω στα αντικείμενα αυτά. Η υλοποίηση των αντικειμένων αυτών και οι λειτουργίες που τα διέπουν προέρχονται μέσω της εμπειρίας στους επιστημονικούς υπολογισμούς.

Κάθε αντικείμενο αποτελείται από μια αφηρημένη διεπαφή (ή αλλιώς ένα σύνολο καλούμενων ακολουθιών) και μίας ή περισσότερων υλοποιήσεων που χρησιμοποιούν συγκεκριμένες δομές δεδομένων. Με τον τρόπο αυτό η PETSc παρέχει τη δυνατότητα ανάπτυξης ορθού και αποτελεσματικού κώδικα για τις διάφορες φάσεις των εφαρμογών επίλυσης, με μια ενιαία προσέγγιση για κάθε ξεχωριστή κατηγορία προβλημάτων. Ο παραπάνω σχεδιασμός μάλιστα επιτρέπει την εύκολη σύγκριση και χρήση διαφορετικών αλγορίθμων (όπως για παράδειγμα τον πειραματισμό με διαφορετικές μεθόδους του υποχώρου Krylov και με διαφορετικούς πίνακες προρυθμιστές (preconditioners)). Επίσης οι επιμέρους βιβλιοθήκες επιτρέπουν την εύκολη παραμετροποίηση και επέκταση τόσο των αλγορίθμων όσο και των υλοποιήσεών τους σε κώδικα. Από την οπτική αυτή προωθείται η ευελιξία και η επαναχρησιμοποίηση κώδικα και διαχωρίζονται τα θέματα που αφορούν τον παραλληλισμό με τα θέματα επιλογής αλγορίθμου. Μια εικόνα για τα στοιχεία – επιμέρους βιβλιοθήκες που χρησιμοποιεί – περιέχει η PETSc παρατηρείται αναλυτικά στην παρακάτω εικόνα:



Εικόνα 6. Αριθμητικές Βιβλιοθήκες της PETSc. Πηγή [9]

4.3 Χαρακτηριστικά της βιβλιοθήκης

Η PETSc αποτελεί λογισμικό ανοιχτού κώδικα και μπορεί να χρησιμοποιηθεί οποιονδήποτε. Μπορεί να τρέξει σε διάφορα λειτουργικά συστήματα όπως *Linux*, *Windows*, *Macintosh* και *Unix*. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε σε περιβάλλον Windows και πιο συγκεκριμένα στην πλατφόρμα *Microsoft Visual Studio*. Μπορεί βάσει κατάλληλης ρύθμισης να δουλέψει με τύπους δεδομένων που αναπαριστούν είτε πραγματικούς, είτε μιγαδικούς αριθμούς, σε απλή ή διπλή ακρίβεια και για ακεραίους τόσο των 32-bit όσο και των 64-bit. Έχει δοκιμαστεί επίσης για πειράματα σε μια μεγάλη γκάμα στενά συνδεδεμένων - παράλληλα αρχιτεκτονικών όπως οι *Cray XT/5*, *Blue Gene/P*, *Earth Simulator*, καθώς επίσης και σε χαλαρά συνδεδεμένων αρχιτεκτονικών όπως σε δίκτυα απλών υπολογιστών.

Η PETSc χρησιμοποιεί τη φιλοσοφία *plug – in* ώστε να πετύχει διασύνδεση με εξωτερικό λογισμικό. Ορισμένα παραδείγματα εξωτερικού λογισμικού που μπορούν να εγκατασταθούν με αυτή αποτελούν οι *SuperLU*, *SuperLU_Dist*, *ParMetis*, *MUPS*, *PLAPACK*, *Chaco*, *Hypre* κ.α.. Για το εξωτερικό λογισμικό παρέχεται από τη βιβλιοθήκη μια διεπαφή ώστε το εξωτερικό αυτό λογισμικό να μπορεί να χρησιμοποιηθεί σε πηγαίο κώδικα εφαρμογών που χρησιμοποιούν την PETSc.

Επιπρόσθετα δίνεται η δυνατότητα στο χρήστη να τροποποιήσει παραμέτρους και επιλογές κατά το στάδιο εκτέλεσης (*runtime*). Για παράδειγμα μπορεί να αλλάξει τη μέθοδο επίλυσης ενός γραμμικού συστήματος από *GMRES* σε *άμεση παραγοντοποίηση LU*, ή να μεταβάλλει τον τύπο αποθήκευσης του πίνακα **A** του συστήματος, είτε τον πίνακα προρυθμιστή (*preconditioner*). Αν μια εφαρμογή χρησιμοποιεί μεγάλο αριθμό παραμέτρων, τότε αυτές μπορούν να παρέχονται μέσω ενός αρχείου κειμένου, το οποίο διαβάζεται προτού ξεκινήσει η εκτέλεση του κώδικα PETSc.

Η αποσφαλμάτωση (*debugging*) όπως είναι γνωστό είναι η πιο επίπονη διαδικασία κατά την ανάπτυξη οποιουδήποτε λογισμικού. Η βιβλιοθήκη PETSc προσφέρει διάφορα χαρακτηριστικά ώστε να την κάνει ευκολότερη. Αρχικά αρκετοί αποσφαλματωτές (*debuggers*) μπορούν να χρησιμοποιηθούν σε ένα πηγαίο κώδικα που χρησιμοποιεί την PETSc όπως οι *gdb*, *dbx* και *xxgdb*. Ο αποσφαλματωτής μπορεί να ενεργοποιηθεί είτε στην αρχή του προγράμματος είτε όταν προκύψει κάποιο σφάλμα. Επιπρόσθετα ένα υποσύνολο διαδικασιών μπορεί επίσης να επιλεγεί για την αποσφαλμάτωση σε κώδικες που χρησιμοποιούν παραλληλισμό. Επιπλέον το ευρέως χρησιμοποιούμενο λογισμικό *Valgrind* μπορεί να χρησιμοποιηθεί για την ανίχνευση σφαλμάτων μνήμης. Παράλληλα το πρόγραμμα κατά τη διαδικασία *runtime* εκτυπώνει μηνύματα σχετικά με την πρόοδο των εντολών του προγράμματος και τις αλλαγές στις δομές δεδομένων που έχει ορίσει ο χρήστης.

Για λεπτομέρειες σχετικά με τις συναρτήσεις που ορίζει η βιβλιοθήκη και του τρόπου με τον οποίο αυτές καλούνται, διατίθεται το σχετικό *manual* το οποίο επεξηγεί αναλυτικά την ορθή χρήση των εργαλείων που μας προσφέρει η βιβλιοθήκη και το οποίο μπορεί να βρεθεί στον παρακάτω σύνδεσμο:

<http://www.mcs.anl.gov/petsc/petsc-current/docs/manual.pdf>

Το παραπάνω εγχειρίδιο ανανεώνεται βάσει των τελευταίων ενημερώσεων της βιβλιοθήκης, ώστε να συμπεριλάβει την επεξήγηση των τελευταίων κάθε φορά αλλαγών οι οποίες έχουν γίνει σ' αυτήν. Στη συνέχεια γίνεται μια αναφορά στις κυριότερες βιβλιοθήκες που δημιουργεί η PETSc και που χρησιμοποιήθηκαν στην ανάπτυξη του επιλυτή επαναληπτικών μεθόδων.

4.4 Διανύσματα – Vectors

Το διάνυσμα, το οποίο συμβολίζεται στη βιβλιοθήκη από το αντικείμενο τύπου `Vec`, αποτελεί ένα από τα απλούστερα αντικείμενα στο PETSc. Ένα παράδειγμα χρήσης των διανυσμάτων είναι για την αποθήκευση της λύσης είτε του δεξιού μέλους ενός γραμμικού συστήματος. Η PETSc περιέχει αρκετούς βασικούς τύπους διανυσμάτων εκ των οποίων οι πιο συνηθισμένοι είναι ο ακολουθιακός και ο παράλληλος τύπος διανύσματος (βάσει του προτύπου MPI). Στη βιβλιοθήκη PETSc Vector είναι ενσωματωμένες βασικές λειτουργίες όπως η πράξη του εσωτερικού γινομένου διανυσμάτων, η πράξη του αθροίσματος διανυσμάτων κ.α.. Το σύνολο των λειτουργιών πάνω σε διανύσματα εξηγείται αναλυτικά στο προαναφερθέν εγχειρίδιο στην ιστοσελίδα του PETSc. Η ανάθεση τιμών σε επιμέρους στοιχεία ενός παράλληλου τύπου διανύσματος μπορεί να γίνει είτε με χρήση καθολικής, είτε με χρήση τοπικής αρίθμησης. Σε περίπτωση επιλογής της καθολικής αρίθμησης κάθε διαδικασία μπορεί να θέσει τα στοιχεία του διανύσματος αυτού και η PETSc διασφαλίζει ότι οι αλλαγές αποθηκεύονται αυτόματα στη σωστή τοποθεσία. Παρέχεται επίσης η δυνατότητα επιλογής ενός υποσυνόλου των στοιχείων ενός διανύσματος ώστε να τα προσθέσει ή να τα εισάγει σε ένα άλλο διάνυσμα.

4.5 Πίνακες - Mat

Η PETSc παρέχει ένα σύνολο ένα μεγάλο σύνολο υλοποιήσεων πινάκων καθώς δεν υπάρχει κάποια συγκεκριμένη μορφή πίνακα η οποία να είναι κατάλληλη για όλα τα προβλήματα τα οποία αντιμετωπίζονται. Ένας πίνακας αναπαρίσταται μέσω ενός αντικειμένου `Mat`. Στην παρούσα έκδοση της βιβλιοθήκης υποστηρίζονται ως κύριοι τύποι αποθήκευσης η μορφή πυκνών πινάκων (`dense`) και αραιοί πίνακες σε μορφή αποθήκευσης συμπιεσμένης γραμμής (`compressed row storage`) η οποία αναφέρεται και ως `aij`. Πιο εξειδικευμένοι τύποι πινάκων που υποστηρίζονται είναι η συμπιεσμένη μορφή αποθήκευσης με χρήση `blocks` που αναφέρεται και ως `baij`, καθώς και η μορφή συμμετρικού πίνακα είτε σε `aij` είτε σε `baij` μορφή. Οποιαδήποτε μορφή πίνακα είναι διαθέσιμη τόσο σε ακολουθιακή όσο και σε παράλληλη έκδοση. Παράλληλα παρέχεται και διεπαφή, ούτως ώστε να δίνεται η δυνατότητα στο χρήστη να ορίσει τη δική του μορφή πίνακα.

4.6 Γραμμικοί επιλυτές και πίνακες προρρυθμιστές

Το αντικείμενο KSP αποτελεί την καρδιά του PETSc, καθώς προσφέρει ομοιόμορφη και αποτελεσματική πρόσβαση στο σύνολο των επιλυτών γραμμικών συστημάτων της βιβλιοθήκης, συμπεριλαμβανομένων των παράλληλων, των ακολουθιακών, των άμεσων και των επαναληπτικών μεθόδων. Πρέπει να τονιστεί ότι το KSP είναι υπεύθυνο για την επίλυση μη ιδιόμορφων συστημάτων της μορφής:

$$A \cdot x = b$$

Σχέση 4.6.1

όμοια με αυτά που εξετάστηκαν στο Κεφάλαιο 2. Για να επιλυθεί το σύστημα το KSP καλεί την ίδια συνάρτηση είτε πρόκειται για άμεση είτε για επαναληπτική μέθοδο. Επιπρόσθετα οποιεσδήποτε ιδιαίτερες τεχνικές επίλυσης και οι αντίστοιχες επιλογές που τις προκαθορίζουν, μπορούν να επιλεγούν κατά το στάδιο του τρεξίματος (`runtime`). Ο συνδυασμός μια μεθόδου του υποχώρου Krylov μαζί με την κατάλληλη επιλογή του πίνακα προρρυθμιστή βρίσκεται σήμερα στο κέντρο της ανάπτυξης κώδικα για επαναληπτικούς επιλυτές γραμμικών συστημάτων. Από τη στιγμή που το ποσοστό σύγκλισης των μεθόδων

του υποχώρου Krylov για ένα συγκεκριμένο γραμμικό σύστημα, είναι ισχυρά εξαρτημένο από το φάσμα (spectrum) του, η διαδικασία της προρύθμισης (preconditioning) χρησιμοποιείται ώστε να αλλάξει αυτό το φάσμα με στόχο να επιταχύνει τη σύγκλιση της μεθόδου. Μια άλλη μορφή της σχέσης 2.4.1 που αναφέρθηκε στο κεφάλαιο 2 σχετικά με τους πίνακες προρύθμισης είναι η εξής:

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

Σχέση 4.6.2

Στην παραπάνω σχέση οι πίνακες M_L και M_R αφορούν υποδηλώνουν τον αριστερό και το δεξί πίνακα προρύθμισης αντίστοιχα. Πρέπει παράλληλα να τονίσουμε ότι εξ' ορισμού όλες οι υλοποιήσεις του KSP χρησιμοποιούν αριστερή προρύθμιση. Η δεξιά προρύθμιση μπορεί να ενεργοποιηθεί είτε κατά τη διαδικασία τρεξίματος είτε με την κλήση κατάλληλης συνάρτησης μέσα στον κώδικα. Κατά την τρέχουσα έκδοση η PETSc υποστηρίζει πάνω από 20 μεθόδους KSP και προρυθμιστές. Μια λίστα από μερικούς εκ των διαθέσιμων προρυθμιστών δίνεται στην παρακάτω εικόνα:

Preconditioners
Jacobi
Block Jacobi
SOR (and SSOR)
Incomplete Cholesky
Incomplete LU
Additive Schwartz
Combination of preconditioners
LU
Cholesky
Shell for user-defined preconditioner

Εικόνα 7. Λίστα με Προρυθμιστές του PETSc. Πηγή [9]

Ο τύπος προρυθμιστή PCComposite επιτρέπει στον προγραμματιστή να δημιουργήσει νέους πίνακες προρύθμισης βάσει των ήδη ορισμένων από τη βιβλιοθήκη. Παράλληλα επιτρέπεται το φώλιασμα επιλυτών και προρυθμιστών. Για παράδειγμα για έναν παράλληλο προρυθμιστή block Jacobi, ο προρυθμιστής που δημιουργείται χρησιμοποιώντας τα διαγώνια στοιχεία του πίνακα σε κάθε επεξεργαστή, μπορεί να είναι κάποιος από τους υπόλοιπους preconditioners όπως LU, ILU, SOR κλπ., και θα χρησιμοποιείται σε κάθε ξεχωριστό block.

Οι ενότητες 4.1, 4.2, 4.3, 4.4, 4.5 και 4.6 περιέχουν υλικό που αντλήθηκε από την αναφορά [9].

4.7 Υλοποίηση επιλυτή γραμμικών συστημάτων

Η εν λόγω ενότητα αναφέρεται στη δημιουργία του επιλυτή, με τη χρήση του οποίου θα τρέξουμε πειράματα ώστε να εξετάσουμε τη συμπεριφορά των επαναληπτικών μεθόδων που προαναφέρθηκαν· της μεθόδου των Συζυγών Κλίσεων (Conjugate Gradients) και της μεθόδου Γενικευμένου Ελάχιστου Υπολοίπου (GMRES) που αναλύθηκαν στο Κεφάλαιο 2. Για το λόγο αυτό ξεκινάμε με μια επεξήγηση της ακολουθίας των βημάτων που ακολουθούνται και τις λειτουργίες που επιτελεί το εν λόγω λογισμικό.

Αρχικά ως υπενθυμίσουμε ότι οι πίνακες που απομονώθηκαν από το GridLAB-D μέσω των συναρτήσεων `fprintf_sparse_SuperMatrix(char* what, SuperMatrix *A)` και `fprintf_dense_SuperMatrix(char* what, SuperMatrix *A)` απομονώσαμε από το GridLAB-D τον πίνακα A και τα διανύσματα x και b σε αρχεία στη μορφή MATLAB. Τα αρχεία αυτά θα τα περάσουμε ως ορίσματα κατά τη διάρκεια εκτέλεσης του γραμμικού επιλυτή. Το λογισμικό αρχικά διαβάζει το αρχείο για τον πίνακα A και τον αποθηκεύει στον τύπο αποθήκευσης `Mat` του PETSc. Στη συνέχεια διαβάζει τα αρχεία που περιέχουν το δεξί μέλος και το διάνυσμα λύσης και τα αποθηκεύει στον τύπο αποθήκευσης `Vec` του PETSc. Παράλληλα παραθέτουμε εντολές εκτύπωσης των `Mat` και `Vec` ώστε να εκτελέσουμε αποσφαλμάτωση. Το διάνυσμα της λύσης θα χρησιμοποιηθεί ως διάνυσμα αναφοράς από την άποψη ότι θα χρησιμοποιηθεί ως η πραγματική λύση για τον υπολογισμό κάθε φορά της νόρμας σφάλματος της προσέγγισης της λύσης που μας δίνει κάθε φορά η αντίστοιχη επαναληπτική μέθοδος. Στη συνέχεια ελέγχουμε την ακρίβεια των δεδομένων τα οποία διαβάστηκαν υπολογίζοντας τη δεύτερη νόρμα της διαφοράς του διανύσματος δεξιού μέλους από το διάνυσμα που προκύπτει με τον πολλαπλασιασμό του πίνακα A με την λύση που μας έδωσε η `superLU` και έχουμε αποθηκεύσει στο διάνυσμα u . Πιο παραστατικά η νόρμα ακρίβειας των δεδομένων που διαβάστηκαν προκύπτει από την παρακάτω σχέση:

$$\|b - A \cdot u\|_2$$

Σχέση 4.7.1

Στη συνέχεια δημιουργούμε το περιβάλλον μέσα στο οποίο θα εκτελεστεί η επίλυση του συστήματος με τη βοήθεια των επαναληπτικών μεθόδων. Επιλύουμε το σύστημα με τις επιθυμητές κάθε φορά παραμέτρους και ελέγχουμε τη σύγκλιση ή μη της μεθόδου, τον αριθμό επαναλήψεων της, το διάνυσμα σφάλματος της προσέγγισης της λύσης που μας δίνεται και έπειτα τη νόρμα του διανύσματος αυτού. Το διάνυσμα σφάλματος προσέγγισης προκύπτει ως η διαφορά της προσέγγισης λύσης x που μας επιστρέφει η επαναληπτική μέθοδος από το διάνυσμα αναφοράς u . Τέλος αποδεσμεύουμε τη μνήμη για τις δομές δεδομένων που χρησιμοποιήθηκαν. Ο πηγαίος κώδικας του λογισμικού που αναπτύχθηκε παρουσιάζεται στο παρακάτω link:

<https://github.com/konsgeorg/Thesis/commit/a36a71097114413540e85d66cbda4e09b851e6a7>

Κεφάλαιο 5

Πειραματικά αποτελέσματα επιλυτή - Συμπεράσματα

5.1 Παράμετροι Εκτέλεσης

Στο παρόν κεφάλαιο γίνεται δοκιμή του παραπάνω λογισμικού σε συγκεκριμένο γραμμικό σύστημα ώστε να διαπιστωθεί η συμπεριφορά τόσο της μεθόδου *Conjugate Gradients* όσο και της *GMRES*. Ως σύστημα επίλυσης έχουν εισαχθεί για όλα τα πειράματα με τη μορφή αρχείων πίνακας και διανύσματα δεξιού μέλους και λύσης με τα εξής χαρακτηριστικά: ο πίνακας A είναι μεγέθους 24^{uv} γραμμών και 24^{uv} στηλών, με 312 μη μηδενικά στοιχεία, ενώ τα διανύσματα αποτελούνται από 24 στοιχεία, τόσο μηδενικά όσο και μη μηδενικά. Να σημειώσουμε ότι το παραπάνω σύστημα που περνάμε στο λογισμικό του γραμμικού επιλυτή κατά τη διάρκεια του σταδίου εκτέλεσής του εκτελέσιμου αρχείου solver από τη γραμμή εντολών με την παρακάτω δήλωση στη γραμμή εντολών:

```
./solver -Ain <filename_1>.txt -rhs <filename_2>.txt -solu  
<filename_3>.txt
```

Αποτελεί σύστημα το οποίο έχει λυθεί και απομονωθεί από τον γραμμικό επιλυτή Newton – Raphson με τη βοήθεια της μεθόδου superLU. Για το σύστημα αυτό ακολουθούμε τη σύμβαση να περνάμε σαν πρώτο όρισμα και μετά το προσδιοριστικό `-Ain` το όνομα του αρχείου που περιέχει τον πίνακα του συστήματος, σαν δεύτερο όρισμα και μετά το προσδιοριστικό `-rhs` το αρχείο του διανύσματος δεξιού μέλους και τέλος σαν τρίτο όρισμα και μετά το προσδιοριστικό `-solu` το αρχείο του διανύσματος λύσης. Στα παρακάτω πειράματα εξετάστηκε η αποτελεσματικότητα των δύο επαναληπτικών μεθόδων ως προς τη νόρμα σφάλματος και τον αριθμό επαναλήψεων μέχρι την τελική σύγκλιση, παράμετροι που χαρακτηρίζουν πλήρως την αποτελεσματικότητα μιας επαναληπτικής μεθόδου στην προσέγγιση της λύσης του γραμμικού συστήματος. Για κάθε πείραμα εξετάζουμε και τις δύο προαναφερθείσες μεθόδους

5.2 Πείραμα 1^ο

Το πρώτο πείραμα εκτελείται βάσει της υλοποίηση του κώδικα που παρατίθεται εκτός σχολίων στο τέλος του Κεφαλαίου 4. Στο πείραμα αυτό χρησιμοποιούμε ως πίνακα για τη δημιουργία του προρυθμιστή (preconditioner) τον ίδιο τον πίνακα του συστήματος A . Το παραπάνω το επιτυγχάνουμε με τη γραμμή 217 του κώδικα μέσω της συνάρτησης `KSPSetOperators(ksp, A, A)` της οποίας το τρίτο όρισμα περιέχει τον πίνακα για τον προρυθμιστή του συστήματος, βάσει του εγχειριδίου (manual) της βιβλιοθήκης PETSc. Τη δεύτερη παράμετρος για το εν λόγω πείραμα αποτελεί η επιλογή της αρχικής εκτίμησης λύσης. Στο πείραμα αυτό αποφασίστηκε ως η αρχική εκτίμηση της λύσης που θα εισαχθεί στον solver να είναι η προκαθορισμένη από την PETSc αρχική εκτίμηση, που δεν είναι άλλη από το μηδενικό διάνυσμα. Ο αναγνώστης θα παρατηρήσει ότι στον παραπάνω κώδικα υπολογίζεται ως αρχικό διάνυσμα, διάνυσμα με στοιχεία μη μηδενικά και περνιέται στη συνάρτηση επίλυσης `KSPSolve(ksp, b, x)` της γραμμής 266· ωστόσο με την εισαγωγή της εντολής στη γραμμή 256 `KSPSetInitialGuessNonzero(ksp, PETSC_TRUE)` σε σχόλια, το διάνυσμα αυτό δε λαμβάνεται υπόψη και χρησιμοποιείται ως αρχική εκτίμηση λύσης το μηδενικό διάνυσμα.

5.2.1 Μέθοδος επίλυσης Conjugate Gradients

Η επιλεγμένη μέθοδος για την τρέχουσα μορφή κώδικα είναι η μέθοδος Συζυγών Κλίσεων και ορίζεται από τη συνάρτηση `KSPSetType(ksp, KSPCG)` της γραμμής 259. Κατά την εκτύπωση των αποτελεσμάτων παρατηρούμε αδυναμία σύγκλισης και ο αλγόριθμος σταματάει στις 3 επαναλήψεις εκτυπώνοντας την εξής νόρμα σφάλματος:

$$\|e\|_2 = 57.61107$$

Σχέση 5.1.1

5.2.2 Μέθοδος επίλυσης GMRES

Για το τρέχον πείραμα μεταβάλλουμε τη μέθοδο επίλυσης από Conjugate Gradients σε GMRES εισάγοντας σε σχόλια τη γραμμή 259 του κώδικα και εξάγοντας την γραμμή 263 εκτός σχολίων. Κατά την εκτύπωση των αποτελεσμάτων παρατηρούμε σύγκλιση της μεθόδου στις 7 επαναλήψεις εκτυπώνοντας την εξής νόρμα σφάλματος:

$$\|e\|_2 = 0.002252528$$

Σχέση 5.1.2

5.3 Πείραμα 2^ο

Στόχο μας αποτελεί η τροποποίηση των παραμέτρων που επηρεάζουν τη σύγκλιση ενός συστήματος ώστε να καταφέρουμε να οδηγήσουμε την μέθοδο Conjugate Gradients σε σύγκλιση. Για το λόγο αυτό επιλέξαμε να μεταβάλλουμε τον προρυθμιστή πίνακα (preconditioner) που χρησιμοποιούν οι δύο μέθοδοι. Αρχικά στις γραμμές 102 - 121 του πηγαίου κώδικα δημιουργείται ο πίνακας P, ο οποίος θα χρησιμοποιηθεί σαν τρίτο όρισμα στη συνάρτηση `KSPSetOperators(ksp, A, A)` της γραμμής 217 τροποποιώντας τη σε `KSPSetOperators(ksp, A, P)`. Ο πίνακας P που δημιουργούμε στις γραμμές 102 - 121 είναι πίνακας με διαστάσεις ίδιες με αυτές του πίνακα A και περιέχει μηδενικά σε όλες τις θέσεις εκτός από τη διαγώνιο στην οποία περιέχει άσσους· κατασκευάζουμε δηλαδή τον ταυτοτικό πίνακα.

5.3.1 Μέθοδος επίλυσης Conjugate Gradients

Όπως και στο πρώτο πείραμα για τη μέθοδο των Συζυγών Κλίσεων ενεργοποιούμε τη συνάρτηση της γραμμής 259 βγάζοντας την από τα σχόλια και παραθέτουμε σε σχόλια τη γραμμή 263. Κατά την εκτύπωση των αποτελεσμάτων παρατηρούμε εκ νέου αδυναμία σύγκλισης για τη μέθοδο των συζυγών κλίσεων και τερματισμό του αλγορίθμου στις 2 επαναλήψεις με την εξής νόρμα σφάλματος:

$$\|e\|_2 = 3268.317$$

Σχέση 5.2.1

5.3.2 Μέθοδος επίλυσης GMRES

Για να ενεργοποιήσουμε τη μέθοδο GMRES ακολουθούμε την ίδια διαδικασία με το 1^ο Πείραμα. Κατά την εκτύπωση των αποτελεσμάτων παρατηρούμε ότι η μέθοδος συγκλίνει στις 114 επαναλήψεις με την εξής νόρμα σφάλματος:

$$\|e\|_2 = 0.003279199$$

Σχέση 5.2.2

5.4 Πείραμα 3^ο

Λόγω της εκ νέου αδυναμίας σύγκλισης της μεθόδου των Συζυγών Κλίσεων αυτή τη φορά θα προσπαθήσουμε να οδηγήσουμε το σύστημα σε σύγκλιση μεταβάλλοντας την αρχική εκτίμηση λύσης. Πλέον το διάνυσμα αρχικής λύσης δε θα είναι το μηδενικό αλλά διάνυσμα που προκύπτει από την παρακάτω σχέση:

$$x^{(0)} = u + r * E[r] * 0.1$$

Σχέση 5.3.1

Όπου:

$x^{(0)}$: το διάνυσμα αρχικής λύσης

u : το διάνυσμα ακριβούς λύσης

r : τυχαίο διάνυσμα στο διάστημα [-1,1]

$E[r]$: μέση τιμή του διανύσματος r

Στόχο μας αποτελεί η δημιουργία ενός διανύσματος αρχικής εκτίμησης λύσης το οποίο θα αποκλίνει ελαφρώς από το διάνυσμα ακριβούς λύσης για να εξετάσουμε τη σύγκλιση ή μη των μεθόδων.

5.4.1 Μέθοδος επίλυσης Conjugate Gradients

Η μέθοδος Conjugate Gradients αποκλίνει εκ νέου, ωστόσο αυτή τη φορά ο αλγόριθμος τερματίζει στις 0 επαναλήψεις με την εξής νόρμα σφάλματος:

$$\|e\|_2 = 10.27065$$

Σχέση 5.3.2

5.4.2 Μέθοδος επίλυσης GMRES

Για το τρέχων παράδειγμα παρατηρήθηκε αδυναμία σύγκλισης και για τη μέθοδο GMRES. Ο αλγόριθμος τερματίζει στην μηδενική επανάληψη με την εξής νόρμα σφάλματος:

$$\|e\|_2 = 10.27065$$

Σχέση 5.3.2

5.5 Συμπεράσματα

Παρατηρούμε ότι η μέθοδος GMRES παρουσιάζει εμφανώς καλύτερη συμπεριφορά από τη CG για το δεδομένο γραμμικό σύστημα καθώς το σφάλμα στη GMRES είναι μικρότερο από αυτό της CG στις δύο εκ των τριών περιπτώσεων, γεγονός το οποίο δικαιολογείται από τη σύγκλιση της πρώτης και την αδυναμία σύγκλισης της δεύτερης. Δεδομένου ότι η CG συνέχισε να αποκλίνει και μετά την εισαγωγή διαφορετικών παραμέτρων, συμπεραίνουμε

ότι ο λόγος απόκλισης της μεθόδου δε συμβαίνει ούτε λόγω ακατάλληλου πίνακα προρυθμιστή, ούτε λόγω ακατάλληλης αρχικής εκτίμησης λύσης. Η αιτία έγκειται στα εγγενή χαρακτηριστικά του πίνακα A του συστήματος προς επίλυση. Θυμίζουμε από το 2^ο Κεφάλαιο ότι η μέθοδος των Συζυγών Κλίσεων αποτυγχάνει για πίνακες μη θετικά ορισμένους.

Αναφορές

- [1] Arthur R. Bergen, Vijay Vittal, *Power System Analysis (2nd Edition)*, Prentice Hall, Upper Saddle River, New Jersey, 2000
- [2] C. Pozrikidis, *Αριθμητικές μέθοδοι στην Επιστήμη και στη Μηχανική*, Εκδόσεις Τζιόλα, Θεσσαλονίκη
- [3] G. Golub, C. Van Loan, *Matrix Computations (2nd Edition)*, The Johns Hopkins University Press, Baltimore, 1989
- [4] S. Kaniel, Estimates for some computational techniques in linear algebra, *Mathematics of Computation*, 20 (1966), pp. 369–378
- [5] Saad, Shultz, GMRES: A generalized minimal residual algorithm for solving non - symmetric linear systems, *SIAM J. Sci. Statist. Comput.*, 7 (1986), pp. 856–869
- [6] H. F. Walker, Implementation of the GMRES method using Householder transformations, *SIAM J. Sci. Statist. Comput.*, 9 (1988), pp. 152–163
- [7] J. Demmel, M. Heath, H. Van Der Vorst, Parallel linear algebra, in *Acta Numerica*, Vol. 2, Cambridge Press, New York, 1993
- [8] Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June M. Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, Henk Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia
- [9] Shrirang Abhyankar, Barry Smith, Hong Zhang, Alexander Flueck, *Using PETSc to Develop Scalable Applications for Next-Generation Power Grid*
- [10] Property of National University of Singapore
- [11] GridLAB-D site <http://www.gridlabd.org/>
- [12] Opensees site <http://opensees.berkeley.edu/>
- [13] PETSc site <http://www.mcs.anl.gov/petsc/>