*ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ*

*ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ*

*ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ*

*ΤΟΜΕΑΣ ΓΕΩΦΥΣΙΚΗΣ*

## *ΤΣΟΚΑΣ ΑΡΣΕΝΙΟΣ- ΝΙΚΟΛΑΟΣ*

# Διερεύνηση της ποιότητας λύσης μεγάλων γραμμικών συστημάτων με τη χρήση της MATLAB και εφαρμογή σε γεωφυσικά προβλήματα

$$\begin{pmatrix} 0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 8 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 4 & 0 \\ 0 & 1 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \end{pmatrix}$$

**ΕΠΙΒΛΕΠΟΝΤΕΣ ΚΑΘΗΓΗΤΕΣ:** *ΚΩΝΣΤΑΝΤΙΝΟΣ Β. ΠΑΠΑΖΑΧΟΣ*

*ΠΑΝΑΓΙΩΤΗΣ ΤΣΟΥΡΛΟΣ*

ΘΕΣΣΑΛΟΝΙΚΗ 2014

# ΠΕΡΙΕΧΟΜΕΝΑ

# 1 Εισαγωγή

Η παρούσα διπλωματική εργασία εκπονήθηκε στον Τομέα Γεωφυσικής του τμήματος Γεωλογίας της Σχολής Θετικών Επιστημών του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, υπό την επίβλεψη του Καθηγητή Κωνσταντίνου Β. Παπαζάχου και του Αναπληρωτή Καθηγητή Παναγιώτη Τσούρλου, κατά το ακαδημαϊκό έτος 2013-2014. Μεγάλο τμήμα πραγματοποιήθηκε σε χώρους του Τομέα Γεωφυσικής και χρησιμοποιήθηκε εξοπλισμός που παραχωρήθηκε από τον Τομέα.

Θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές κκ. Κωνσταντίνο Β. Παπαζάχο και Παναγιώτη Τσούρλο για την εμπιστοσύνη και την κατανόησή τους καθ' όλη τη διάρκεια εκπόνησης της εργασίας, καθώς η συνδρομή τους τόσο στην κατανόηση του θεωρητικού υποβάθρου του θέματος, όσο και στην ανάπτυξη του απαιτούμενου λογισμικού ήταν συνεχής και καθοριστική. Επίσης, θα ήθελα να ευχαριστήσω τον διδάκτορα Νικόλαο Παπαδόπουλο, ερευνητή του Εργαστηρίου Γεωφυσικής, Δορυφορικής Τηλεπισκόπισης και Αρχαιοπεριβάλλοντος του Ινστιτούτου Μεσογειακών Σπουδών, για την παραχώρηση κώδικα της γλώσσας C για την εφαρμογή της μεθόδου LSQR. Ακόμη, θα ήθελα να ευχαριστήσω την υποψήφια διδάκτορα Χρύσα Βεντούζη για την παραχώρηση ενός πίνακα από ένα πραγματικό Γεωφυσικό πρόβλημα, το οποίο μελετήθηκε σύμφωνα με όσα αναφέρονται στην εργασία αυτή και εφαρμόστηκαν τα λογισμικά που δημιουργήθηκαν για τους σκοπούς της εργασίας.

Η ανάπτυξη των λογισμικών που χρησιμοποιήθηκαν για την επίλυση όλων των προβλημάτων που μελετήθηκαν έγινε στο περιβάλλον προγραμματισμού της MATLAB. Επίσης στο περιβάλλον της MATLAB, έγιναν όλοι οι απαραίτητοι υπολογισμοί κι ακόμη δημιουργήθηκαν τα διαγράμματα και τα οπτικοποιημένα δεδομένα που συνοδεύουν τα αποτελέσματα.

Στην εργασία αυτή μελετήθηκαν μεγάλα γραμμικά προβλήματα υπό το πρίσμα της διερεύνησης αφενός των ποιοτικών χαρακτηριστικών της λύσης και πολύ περισσότερο της ακρίβειας των προσεγγιστικών μεθόδων υπολογισμού της λύσης. Έμφαση δόθηκε στη μελέτη των μέτρων ποιότητας της λύσης και ειδικότερα των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας. Έτσι, μελετήθηκαν κάποιες προσεγγιστικές μορφές των πινάκων και συγκρίθηκαν με τις ακριβείς μορφές τους.

Στο δεύτερο κεφάλαιο δίνονται γενικές πληροφορίες που αφορούν τα γραμμικά προβλήματα και τη λύση ελαχίστων τετραγώνων, αναλύονται όλες οι παράμετροι που υπεισέρχονται σε τέτοια προβλήματα και παρουσιάζονται οι τρόποι εκτίμησης της ποιότητας της λύσης. Αναλύονται ακόμη κάποιες προσεγγιστικές μέθοδοι και τέλος, γίνεται μια παρουσίαση του περιβάλλοντος προγραμματισμού MATLAB, των ιδιαίτερων χαρακτηριστικών του καθώς και της ευχέρειας που παρέχει σε γραμμικά προβλήματα.

Στο τρίτο κεφάλαιο γίνεται αναφορά στην ανάπτυξη του κώδικα που χρησιμοποιήθηκε κατά τις δοκιμές που πραγματοποιήθηκαν, καθώς και στους λόγους που μας οδήγησαν σε συγκεκριμένες επιλογές. Παρουσιάζονται επίσης οι εφαρμογές που έγιναν σε τυχαίους πίνακες, τα αποτελέσματα και κάποιες πρώτες εκτιμήσεις. Τέλος, παρουσιάζεται η εφαρμογή του κώδικα σε πραγματικό πρόβλημα, τα αποτελέσματα και τα συμπεράσματα.

Τέλος, στο τέταρτο κεφάλαιο παρουσιάζονται συνοπτικά τα τελικά συμπεράσματα από τις δοκιμές που πραγματοποιήθηκαν.

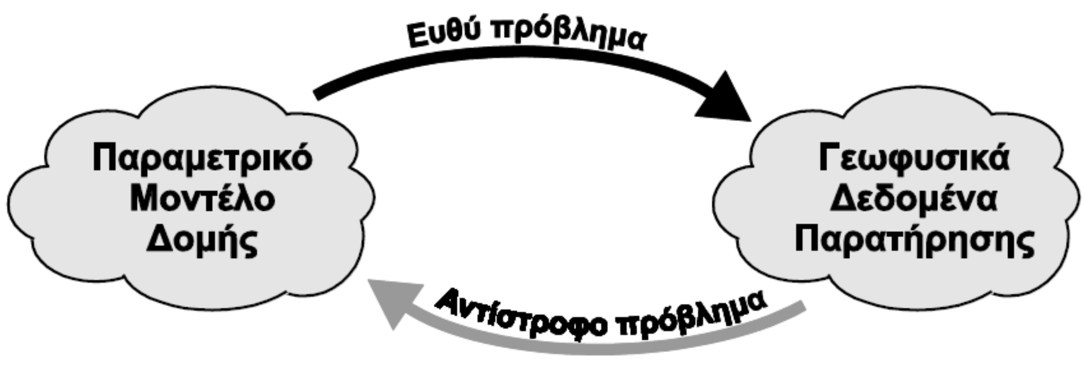
# 2 Γραμμικά προβλήματα, τρόποι επίλυσης και η γλώσσα προγραμματισμού MATLAB

## 2.1 Θεωρία αντιστροφής και γραμμικά προβλήματα

Στην παράγραφο αυτή παρουσιάζονται σε γενικές γραμμές οι αρχές που διέπουν τα γραμμικά προβλήματα, τα οποία είναι το αντικείμενο της συγκεκριμένης μελέτης. Παρουσιάζονται επίσης οι αιτίες που οδήγησαν στη μελέτη των γραμμικών προβλημάτων, καθώς και κάποια στοιχεία για τη θεωρία αντιστροφής, η οποία αποτελεί το γενικότερο πλαίσιο στο οποίο εντάσσονται τα γραμμικά προβλήματα τα οποία μελετήθηκαν στην παρούσα μελέτη.

### 2.1.1 Γενική θεωρία αντιστροφής

Η θεωρία αντιστροφής στις Θετικές Επιστήμες μας επιτρέπει να εξαγάγουμε συμπεράσματα για ένα φυσικό φαινόμενο ή σύστημα από ένα σύνολο μετρήσεων. Με αυτόν τον τρόπο, γίνεται εφικτό να πάρουμε πληροφορίες για φυσικά μεγέθη τα οποία δεν μπορούμε να παρατηρήσουμε απευθείας. Τα φυσικά μεγέθη τα οποία θέλουμε να μελετήσουμε συνήθως παριστάνονται από ένα σύνολο παραμέτρων ή συναρτήσεων, τις οποίες προσπαθούμε να προσδιορίσουμε. Το αντίστροφο πρόβλημα έγκειται στον προσδιορισμό των παραμέτρων του μοντέλου που έχει κατασκευαστεί με σκοπό να περιγράψει μια φυσική δομή.
Για να επιλυθεί το αντίστροφο πρόβλημα, είναι απαραίτητο να έχει προηγουμένως λυθεί το ευθύ πρόβλημα, δηλαδή ο υπολογισμός των συνθετικών δεδομένων, μα άλλα λόγια των αναμενόμενων παρατηρήσεων για το συγκεκριμένο, γνωστό μοντέλο φυσικής δομής.




*Σχήμα 2.1: Ευθύ και αντίστροφο πρόβλημα ([3] §8.10)*

Πιο συγκεκριμένα, η μετατροπή από τις παραμέτρους του μοντέλου **m** στα δεδομένα **d**, καθώς και η αντίστροφη μετατροπή, είναι το αποτέλεσμα της φυσικής αλληλεπίδρασης μεταξύ των υπό μελέτη παραμέτρων και των δεδομένων και περιγράφεται από έναν γενικό τελεστή **G**, ο οποίος δρα στις παραμέτρους του μοντέλου, παράγοντας έτσι τα δεδομένα:

$$\boldsymbol{d} = \boldsymbol{G}(\boldsymbol{m}) \qquad (2.1)$$

Ο τελεστής **G** αντιπροσωπεύει τις εξισώσεις και συναρτήσεις που συσχετίζουν τις παραμέτρους του μοντέλου με τα παρατηρούμενα δεδομένα. Το αντίστροφο πρόβλημα έγκειται στην εύρεση του μοντέλου που ικανοποιεί την παραπάνω σχέση. Ωστόσο, μια ακριβής και μοναδική λύση δεν είναι σχεδόν ποτέ εφικτή στην πραγματικότητα για διάφορους λόγους. Αφενός τα δεδομένα είναι αποτελέσματα μετρήσεων, κι ως τέτοια δεν είναι ποτέ ακριβή καθώς η ακρίβεια μετρήσεων είναι πεπερασμένη, αλλά και υπάρχει σημαντικό ποσοστό θορύβου. Αφετέρου, η λύση ενδέχεται να είναι ασταθής, δηλαδή μικρή μεταβολή των δεδομένων προκαλεί μεγάλη μεταβολή της λύσης, γεγονός που πρέπει να ληφθεί υπόψη αφού τα δεδομένα αποτελούν (συνήθως) πειραματικά αποτελέσματα.

Δεν πρέπει να αγνοείται ακόμη το ότι είναι πιθανό να έχουν γίνει σημαντικές παραλείψεις και λάθη κατά την παραμετροποίηση του προβλήματος. Συνήθως υπάρχουν πολλοί τρόποι για να παραμετροποιήσει κανείς ένα πρόβλημα, και αυτό είναι το πρώτο βήμα για την επίλυση κάθε φυσικού προβλήματος, αλλά παράλληλα και ένα από τα σημαντικότερα.

### 2.1.2 Γραμμικά και γραμμικοποιημένα αντίστροφα προβλήματα

Πολύ συχνά στη Γεωφυσική απαντώνται γραμμικά αντίστροφα προβλήματα, ή αντίστροφα προβλήματα που ανάγονται σε γραμμικά μέσω απλών μετασχηματισμών. Ο λόγος για τον οποίο συνηθίζεται διάφορα προβλήματα να ανάγονται σε γραμμικά είναι η εκτενής μελέτη που έχει πραγματοποιηθεί πάνω στα προβλήματα αυτού του τύπου, τους τρόπους επίλυσης και τις ιδιαιτερότητες που παρουσιάζουν. Αντιθέτως, είναι περιορισμένη η έρευνα για τα μη γραμμικά προβλήματα.

Στη γραμμική περίπτωση, η σχέση που συνδέει το διάνυσμα **m** των παραμέτρων του μοντέλου με το διάνυσμα **d** των δεδομένων μπορεί να εκφραστεί με ένα γραμμικό σύστημα του τύπου:

$$\boldsymbol{d} = \boldsymbol{G}\,\boldsymbol{m} + \boldsymbol{e} \qquad (2.2)$$

όπου, αν έχουμε N δεδομένα και M παραμέτρους, τότε ο **G** είναι ένας N×M πίνακας και **e** είναι το διάνυσμα των σφαλμάτων. Στην ιδανική περίπτωση, θα θέλαμε να ισχύει **e**=**0**, οπότε ο υπολογισμός των παραμέτρων θα ήταν μια πολύ πιο απλοποιημένη υπόθεση. Αυτό βέβαια δεν συμβαίνει ποτέ στην πράξη, και έτσι τα δεδομένα δεν καθορίζονται μόνο από τον τελεστή **G**, αλλά και από τα σφάλματα.

Στην περίπτωση μη γραμμικού προβλήματος, η γραμμικοποίηση γίνεται ως εξής: υποθέτουμε ότι το μοντέλο συνδέεται με τα δεδομένα μέσω της σχέσης **d=f(m).** Έχοντας τα δεδομένα από τις παρατηρήσεις $\mathbf{d}^{\mathbf{obs}}$, υποθέτουμε μια αρχική προσέγγιση του μοντέλου $\mathbf{m}^{\mathbf{0}}$, και θεωρούμε το ανάπτυγμα Taylor της συνάρτησης **f** στο σημείο $\mathbf{m}^{\mathbf{0}}$:

$$\boldsymbol{f}(\boldsymbol{m}^{\boldsymbol{0}} + \boldsymbol{\Delta m}) = \boldsymbol{f}(\boldsymbol{m}^{\boldsymbol{0}}) + \boldsymbol{\nabla f}(\boldsymbol{m}^{\boldsymbol{0}})\boldsymbol{\Delta m} + \ldots \qquad (2.3)$$

Με την παραδοχή ότι το αρχικό μοντέλο βρίσκεται αρκετά κοντά στο πραγματικό, οι όροι ανώτερης τάξης μπορούν να θεωρηθούν αρκετά μικροί και κατά συνέπεια να παραληφθούν χωρίς να έχουμε σημαντικό σφάλμα, κι έτσι:

$$\boldsymbol{f(m^1) - f(m^0) = f(m^0 + \Delta m) - f(m^0) \cong \nabla f(m^0)\Delta m \Leftrightarrow}$$
$$\boldsymbol{d^{obs} - d^0 = \Delta d \cong J\,\Delta m} \quad (2.4)$$

όπου **J** είναι ο Ιακωβιανός πίνακας των μερικών παραγώγων των δεδομένων ως προς τις παραμέτρους του μοντέλου ενώ το διάνυσμα $\mathbf{d^0}$ των υποθετικών δεδομένων υπολογίζεται από την (2.1) ως εξής: $\mathbf{d^0}=\mathbf{G}(\mathbf{m^0})$. Η επίλυση του γραμμικού αυτού συστήματος δίνει τη διαφορά **Δm** και κατά συνέπεια μια νέα προσέγγιση του μοντέλου $\mathbf{m^1}$. Αυτή είναι όντως μια νέα προσέγγιση και δεν ταυτίζεται με το πραγματικό μοντέλο διότι στην παραπάνω σχέση (2.4), η ισότητα δεν ισχύει ακριβώς – οι όροι ανώτερης τάξης έχουν παραληφθεί. Η νέα προσέγγιση $\mathbf{m^1}$ χρησιμοποιείται αντί της αρχικής $\mathbf{m^0}$ ώστε το σύστημα να επιλυθεί εκ νέου και να δώσει και πάλι μια νέα προσέγγιση. Η διαδικασία επαναλαμβάνεται πολλές φορές μέχρι να βρεθεί μια ικανοποιητική προσέγγιση του πραγματικού μοντέλου, όταν δηλαδή δύο διαδοχικές προσεγγίσεις είναι αρκετά κοντά, ανάλογα με την απαιτούμενη ακρίβεια του εκάστοτε προβλήματος.

Για την επίλυση του γραμμικού συστήματος (Εξίσωση 2.2), μια μεγάλη κατηγορία σφαλμάτων είναι τα σφάλματα παρατήρησης, αυτά που οφείλονται στην πεπερασμένη ακρίβεια μέτρησης και στο ότι στα δεδομένα υπεισέρχεται θόρυβος. Αυτά υπάρχουν πάντα, και σε αρκετές περιπτώσεις μπορούν και να προσδιοριστούν, ωστόσο είναι συνήθως τα μικρότερα και πολύ συχνά αγνοούνται, καθώς η επιρροή τους στην αντιστροφή είναι συχνά αμελητέα ή μικρή.

Αντίθετα, τα σφάλματα που συνήθως παίζουν το μεγαλύτερο ρόλο είναι αφενός τα σφάλματα που οφείλονται στην παραμετροποίηση του προβλήματος, κι αφετέρου τα σφάλματα γραμμικοποίησης. Για τα πρώτα, αν το μοντέλο που έχει επιλεγεί αδυνατεί να περιγράψει με ακρίβεια τη φυσική πραγματικότητα (π.χ. λόγω απλοποιήσεων), τότε η επίλυση του αντίστροφου γραμμικού προβλήματος ενδέχεται να οδηγήσει σε εντελώς λάθος συμπεράσματα. Όσον αφορά τη γραμμικοποίηση, που πραγματοποιείται στα μη γραμμικά προβλήματα, όταν το αρχικό μοντέλο $\mathbf{m^0}$ απέχει πολύ από το πραγματικό, οι όροι ανώτερης τάξης του αναπτύγματος Taylor δεν είναι αρκετά μικροί, η επαναληπτική επίλυση του συστήματος μπορεί να μην οδηγεί προς το πραγματικό μοντέλο, δηλαδή οι διαδοχικές προσεγγίσεις δεν συγκλίνουν προς τη ζητούμενη λύση.

Στην πλειονότητα των προβλημάτων τα δεδομένα είναι περισσότερα από τις παραμέτρους του μοντέλου (ισοδύναμα οι εξισώσεις είναι περισσότερες από τους αγνώστους). Πράγματι, στην πράξη πραγματοποιούνται πολλές παρατηρήσεις, στοχεύοντας στο να έχουμε όσο το δυνατό περισσότερες πληροφορίες για το πρόβλημα που μελετάται. Κατά συνέπεια ο πίνακας **G** δεν είναι τετραγωνικός, επομένως ο αντίστροφός του δεν υπάρχει. Αυτό σημαίνει, από μαθηματικής πλευράς, ότι το πρόβλημα δεν έχει μία και μοναδική λύση. Μάλιστα, στα περισσότερα προβλήματα δεν υπάρχει μοντέλο που να ικανοποιεί ακριβώς την (2.2). Συνήθως, επιλέγεται ως λύση το «καλύτερο» μοντέλο, αυτό που πληροί κάποιες συγκεκριμένες προϋποθέσεις. Ωστόσο, μια μαθηματικά «καλή» λύση δεν ανταποκρίνεται πάντα στη φυσική πραγματικότητα. Ενδέχεται να υπάρχουν άπειρες λύσεις με τα ίδια ή παρόμοια χαρακτηριστικά, οι οποίες ικανοποιούν στον ίδιο ή παρόμοιο βαθμό την (2.2).

Ο τελεστής **G** δρα στο μοντέλο **m** και παράγει τα δεδομένα **d**. Αντίστοιχα, με όποιον τρόπο κι αν επιλέξουμε να λύσουμε το πρόβλημα, μπορούμε να θεωρήσουμε ότι υπάρχει ένας τελεστής που καλείται ένας γενικευμένος αντίστροφος του **G,** ο $\mathbf{G}^{-g}$, που θα δράσει πάνω στα δεδομένα **d** και θα παραγάγει μια εκτίμηση $\hat{\mathbf{m}}$ των πραγματικών δεδομένων **m** που προσπαθούμε να προσδιορίσουμε:

$$\hat{\boldsymbol{m}} = \boldsymbol{G}^{-g}\,\boldsymbol{d} \qquad (2.5)$$

## 2.2 Επίλυση γραμμικών συστημάτων

Η πιο γνωστή μέθοδος επίλυσης των γραμμικών συστημάτων είναι η μέθοδος ελαχίστων τετραγώνων, η οποία θα μελετηθεί σε αυτήν την παράγραφο. Πρέπει ωστόσο να μην ξεχνάμε ότι η λύση αυτή δεν είναι παρά μια στατιστική λύση, η οποία προσπαθεί να ικανοποιήσει όσο το δυνατό περισσότερο τις εξισώσεις, όπως θα φανεί, αλλά δεν τις ικανοποιεί πλήρως. Επιπλέον, εκτός από τη λύση, ενδιαφέρον παρουσιάζουν και τα μέτρα ποιότητας αυτής, καθώς δίνουν πολλές πληροφορίες σχετικά με το πόσο αξιόπιστη είναι η λύση.

### 2.2.1 Η λύση ελαχίστων τετραγώνων

Δεδομένου ότι τα σφάλματα των δεδομένων είναι άγνωστα και δεν μπορούν να προσδιοριστούν και κατά συνέπεια δεν μπορούν απλά να αφαιρεθούν από την (2.2), η προσοχή μας στρέφεται στην ελαχιστοποίηση της επίδρασης των σφαλμάτων στη λύση. Συνήθως υποθέτουμε ότι τα σφάλματα των δεδομένων είναι τυχαία, κάτι που σημαίνει ότι τα παρατηρησιακά δεδομένα δεν παράγονται αιτιοκρατικά, αλλά με βάση κάποια πιθανοθεωρητική κατανομή, η οποία μάλιστα είναι η N-διάστατη κανονική με μέση τιμή τα πραγματικά δεδομένα και πίνακα συνδιασπορών $\mathbf{C_d}$. Σε αυτήν την περίπτωση, η ελαχιστοποίηση των σφαλμάτων ισοδυναμεί με τη μεγιστοποίηση της πιθανότητας να πάρουμε από τις μετρήσεις τα δεδομένα που έχουν πράγματι μετρηθεί πειραματικά. Αυτό επιτυγχάνεται όταν ελαχιστοποιείται η ποσότητα:

$$(\boldsymbol{d}-\boldsymbol{Gm})^{\boldsymbol{T}}\,\boldsymbol{C}_{\boldsymbol{d}}^{-\mathbf{1}}\,(\boldsymbol{d}-\boldsymbol{Gm}) \quad (2.6)$$

Στην περίπτωση που τα δεδομένα έχουν ταυτοτικά ίδια και ασυσχέτιστα σφάλματα, δηλαδή ισχύει $\mathbf{C_d}=\boldsymbol{\sigma}_\mathbf{d}^{\mathbf{2}}\mathbf{I}$, τότε από την (2.6) προκύπτει ότι αρκεί να ελαχιστοποιηθεί η ποσότητα:

$$|\boldsymbol{d}-\boldsymbol{Gm}|^{\mathbf{2}} \quad (2.7)$$

στην οποία οφείλει το όνομά της η μέθοδος, αφού ισοδυναμεί με την ελαχιστοποίηση των τετραγώνων των υπολοίπων **d**-**Gm**. Στην πράξη, πολύ συχνά θεωρούμε ότι τα σφάλματα είναι πράγματι ίδια κι ασυσχέτιστα, καθώς ο υπολογισμός των από κοινού σφαλμάτων (συνδιασπορών) των δεδομένων είναι μια χρονοβόρα ή και πρακτικά αδύνατη διαδικασία, και κατά κανόνα παραλείπεται. Ακόμη, όπως ήδη αναφέρθηκε, τα σημαντικότερα σφάλματα οφείλονται στην παραμετροποίηση και γραμμικοποίηση του προβλήματος και κατά συνέπεια δεν μπορούν να υπολογιστούν.

Η λύση ελαχίστων τετραγώνων (least squares) που ελαχιστοποεί την (2.7) είναι η λύση που προκύπτει από την επίλυση των κανονικών εξισώσεων (normal equations). Αυτές προκύπτουν από το σύστημα **d** = **Gx** πολλαπλασιάζοντας από αριστερά με $\mathbf{G^T}$:

$$\boldsymbol{G^T G x = G^T d} \quad (2.8)$$

Έτσι η λύση ελαχίστων τετραγώνων είναι:

$$\boldsymbol{m_{LSQ} = (G^T G)^{-1} G^T d} \quad (2.9)$$

Στην περίπτωση που τα δεδομένα δεν έχουν ίδια κι ασυσχέτιστα σφάλματα οι κανονικές εξισώσεις είναι οι εξής:

$$\boldsymbol{G^T C_d^{-1} G x = G^T C_d^{-1} d} \quad (2.10)$$

και η λύση που προκύπτει είναι:

$$\boldsymbol{m_{LSQ} = (G^T C_d^{-1} G)^{-1} G^T C_d^{-1} d} \quad (2.11)$$

### 2.2.2 Επίλυση γραμμικών συστημάτων με πρόσθετους περιορισμούς

Όπως αναφέρθηκε, η λύση ελαχίστων τετραγώνων δεν είναι η μοναδική λύση του συστήματος, αλλά αποτελεί μια στατιστική προσέγγιση. Αυτή η προσέγγιση ενδέχεται να είναι υπερβολικά ασταθής. Πράγματι, αν ο πίνακας $\mathbf{G^T G}$ (ή $\mathbf{GTC_d^{-1}G}$) είναι ιδιάζων, ή ακόμα κι αν η ορίζουσά του είναι πολύ κοντά στο μηδέν, τότε ο αντίστροφός του είτε δεν θα υπάρχει είτε θα είναι πολύ ασταθής, γεγονός που καθιστά τη λύση ελαχίστων τετραγώνων αναξιόπιστη, αφού τα σφάλματά της θα είναι πολύ σημαντικά. Επιπλέον, πολλές φορές στα γεωφυσικά προβλήματα, απαιτείται η λύση να κυμαίνεται μέσα σε κάποια συγκεκριμένα πλαίσια (π.χ. αναμενόμενα όρια φυσικών παραμέτρων). Μια διαδικασία που μπορεί να πετύχει τον έλεγχο της αστάθειας της λύσης είναι μέσω της εισαγωγής πρόσθετων περιορισμών που έχουν τη μορφή:

$$\boldsymbol{h = L\,m} \quad (2.12)$$

και εισάγονται στο σύστημα **d**=**Gm** ως επιπλέον εξισώσεις. Οι περιορισμοί της σχέσης (2.12) έχουν ως αποτέλεσμα να οδηγούν σε διαφορετική λύση, η οποία περιγράφει σε πιο ρεαλιστικό βαθμό τη φυσική πραγματικότητα. Το νέο επαυξημένο σύστημα είναι:

$$\boldsymbol{\binom{G}{\lambda L} m = \binom{d}{\lambda h}} \quad (2.13)$$

όπου ο συντελεστής λ καθορίζει τη βαρύτητα των περιορισμών στην τελική λύση. Η μέθοδος ελαχίστων τετραγώνων όταν εφαρμοστεί στο (2.13), ελαχιστοποιεί την ποσότητα:

$$\boldsymbol{|d - Gm|^2 + \lambda^2 |Lm - h|^2} \quad (2.14)$$

και οδηγεί στη λύση:

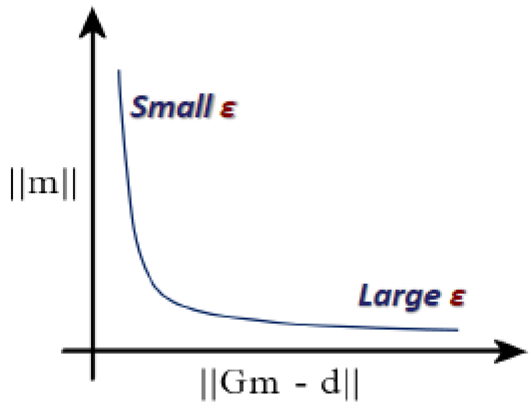
$$m_{gLSQ} = (G^T G + \lambda^2 L^T L)^{-1}(G^T d + \lambda^2 L^T h) \quad (2.15)$$

Ειδικότερα, σε πολλά γεωφυσικά προβλήματα, μας ενδιαφέρει η λύση να έχει όσο το δυνατό μικρότερο μέτρο. Για γραμμικοποιημένα προβλήματα, η ελαχιστοποίηση του μέτρου της λύσης είναι ιδιαίτερης σημασίας, καθώς τα «ελάχιστα τετράγωνα» δεν υπολογίζουν απευθείας το μοντέλο, αλλά τη διαφορά του νέου μοντέλου από το αρχικό (συνθετικό) μοντέλο (βλ. εξίσωση 2.4). Έτσι, επιδιώκεται το νέο, βελτιωμένο μοντέλο να είναι όσο πιο κοντά γίνεται στο αρχικό, δηλαδή το μέτρο της λύσης να είναι κοντά στο μηδέν. Αυτό επιτυγχάνεται θέτοντας **L**=**I** και **h**=**0** δηλαδή ελαχιστοποιώντας την ποσότητα

$$|d - Gm|^2 + \varepsilon^2 |m|^2 \quad (2.16)$$

Έτσι, το σύστημα (2.11) γίνεται:

$$\binom{G}{\varepsilon I} m = \binom{d}{0} \quad (2.17)$$




*Σχήμα 2.2:Η επίδραση του βάρους των περιορισμών στην τελική λύση ([7] σελ.35)*

κι οδηγεί στη λύση:

$$m_{DLSQ} = (G^T G + \varepsilon^2 I)^{-1} G^T d \quad (2.18)$$

Η μέθοδος αυτή είναι γνωστή ως μέθοδος ελαχίστων τετραγώνων ελαχίστου μέτρου (damped least squares).

## 2.2.3 Μέτρα ποιότητας της λύσης ελαχίστων τετραγώνων: Οι πίνακες συμμεταβλητότητας και διακριτικής ικανότητας

Όπως προαναφέρθηκε, μπορούμε να υποθέσουμε ότι γενικά υπάρχει ένας τελεστής $\mathbf{G}^{-g}$ που δρα στα δεδομένα και παράγει τη ζητούμενη λύση (Εξισωση 2.5).

Για τη λύση αυτή, ο πίνακας συνδιασποράς ή συμμεταβλητότητας του μοντέλου (model covariance matrix) είναι:

$$\boldsymbol{C_m} = \langle \widehat{\boldsymbol{m}}, \widehat{\boldsymbol{m}}^{\boldsymbol{T}} \rangle = \langle \boldsymbol{G^{-g} d}, (\boldsymbol{G^{-g} d})^{\boldsymbol{T}} \rangle \Longrightarrow \boldsymbol{C_m} = \boldsymbol{G^{-g} C_d} (\boldsymbol{G^{-g}})^{\boldsymbol{T}} \quad (2.19)$$

Όπως φαίνεται από την παραπάνω σχέση, τα σφάλματα του μοντέλου εξαρτώνται από τα σφάλματα των δεδομένων, αλλά και από τον γενικευμένο αντίστροφο.

Για τη μέθοδο των ελαχίστων τετραγώνων και θεωρώντας ότι τα δεδομένα έχουν σφάλματα ίσα με $\sigma_d$ και ασυσχέτιστα μεταξύ τους ($\mathbf{C_d}=\sigma_d^2\mathbf{I}$), τότε ο αντίστροφος τελεστής είναι:

$$\boldsymbol{G^{-g}} = (\boldsymbol{G^T G})^{-\boldsymbol{1}} \boldsymbol{G^T} \quad (2.20)$$

Επομένως ο πίνακας συμμεταβλητότητας της λύσης μπορεί να υπολογιστεί από τη σχέση:

$$\boldsymbol{C_m} = (\boldsymbol{G^T G})^{-\boldsymbol{1}} \boldsymbol{G^T} \boldsymbol{\sigma_d^2} \boldsymbol{I} ((\boldsymbol{G^T G})^{-\boldsymbol{1}} \boldsymbol{G^T})^{\boldsymbol{T}} = \boldsymbol{\sigma_d^2} (\boldsymbol{G^T G})^{-\boldsymbol{1}} \quad (2.21)$$

Αν τώρα συνδυάσουμε τις σχέσεις (2.2) και (2.5), έχουμε:

$$\widehat{\boldsymbol{m}} = \boldsymbol{G^{-g} G}\, \boldsymbol{m} + \boldsymbol{G^{-g} e} \quad (2.22)$$

Ο πίνακας $\mathbf{G^{-g}G}$ συμβολίζεται με **R** και λέγεται πίνακας διακριτικής ικανότητας (Resolution matrix). Ο πίνακας **R** δίνει ένα μέτρο της ποιότητας της λύσης, δηλαδή της εγγύτητας της εκτίμησης $\widehat{\mathbf{m}}$ στο πραγματικό μοντέλο **m**. Θα θέλαμε να ισχύει **R=I** (η ιδανική περίπτωση), ή έστω ο πίνακας **R** να είναι όσο πιο κοντά γίνεται στο μοναδιαίο, αλλά και πάλι δεν θα είχαμε ακριβή εκτίμηση του πραγματικού μοντέλου. Η παραπάνω σχέση γράφεται και ως

$$\widehat{\boldsymbol{m}} = \boldsymbol{m} + (\boldsymbol{R} - \boldsymbol{I})\, \boldsymbol{m} + \boldsymbol{G^{-g} e} \quad (2.23)$$

από όπου φαίνεται ότι η εκτίμηση διαφέρει από το πραγματικό μοντέλο εξαιτίας τόσο της περιορισμένης διακριτικής ικανότητας (**R≠I**), όσο και λόγω των σφαλμάτων. Για τη λύση ελαχίστων τετραγώνων χωρίς πρόσθετους περιορισμούς έχουμε:

$$\boldsymbol{R_{LSQ}} = (\boldsymbol{G^T G})^{-\boldsymbol{1}} \boldsymbol{G^T G} = \boldsymbol{I} \quad (2.24)$$

## 2.3 Ανάλυση Ιδιαζόντων τιμών

Η ανάλυση ιδιαζόντων τιμών (Singular Value Decomposition, για συντομία SVD) είναι μια τεχνική η οποία εκτός από την επίλυση προβλημάτων ελαχίστων τετραγώνων, μας δίνει πρόσθετες πληροφορίες για τον πίνακα **G** του συστήματος. Υποθέσαμε ότι έχουμε N δεδομένα και M παραμέτρους του μοντέλου. Σύμφωνα με την SVD, οι πίνακες $\mathbf{G^TG}$ και $\mathbf{GG^T}$ έχουν τις ίδιες μη μηδενικές ιδιοτιμές $\lambda_i^2$ πλήθους p και οι υπόλοιπες M-p και N-p αντίστοιχα ιδιοτιμές είναι ίσες με μηδέν. Επειδή οι πίνακες αυτοί είναι συμμετρικοί, τα ιδιοδιανύσματά τους σχηματίζουν ορθοκανονικές βάσεις στους χώρους του μοντέλου και των δεδομένων, αντίστοιχα. Έτσι, ο πίνακας **G** μπορεί να γραφεί ως γινόμενο:

$$\boldsymbol{G} = \boldsymbol{U}\,\boldsymbol{\Lambda}\,\boldsymbol{V^T} \qquad (2.25)$$

όπου:

- Ο **U** είναι ένας ορθογώνιος N×N πίνακας του οποίου ο στήλες είναι τα ιδιοδιανύσματα $\mathbf{u_i}$ του $\mathbf{GG^T}$.
- Ο **V** είναι ένας ορθογώνιος M×M πίνακας του οποίου ο στήλες είναι τα ιδιοδιανύσματα $\mathbf{v_i}$ του $\mathbf{G^TG}$.
- Ο **Λ** είναι ένας N×M διαγώνιος πίνακας, του οποίου τα διαγώνια στοιχεία είναι οι τιμές $\lambda_i$, οι θετικές τετραγωνικές ρίζες των ιδιοτιμών των πινάκων $\mathbf{G^TG}$ και $\mathbf{GG^T}$, διατεταγμένες σε φθίνουσα σειρά.

Προφανώς στις i–οστές στήλες των **U** και **V** βρίσκεται το ιδιοδιάνυσμα που αντιστοιχεί στην i–οστή ιδιοτιμή. Πολλές φορές, ιδιοτιμές οι οποίες είναι πολύ μικρές θεωρούνται ίσες με μηδέν. Ο Λ έχει τη μορφή:

$$\mathbf{\Lambda} = \underbrace{\left(\begin{array}{c|c} \mathbf{\Lambda_p} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right)}_{(M)} (N) \qquad (2.26)$$

Αν γράψουμε τώρα τους **U** και **V** ως $[\mathbf{U_p}|\mathbf{U_0}]$ και $[\mathbf{V_p}|\mathbf{V_0}]$, όπου οι $\mathbf{U_p}$ και $\mathbf{V_p}$ περιέχουν τα ιδιοδιανύσματα που αντιστοιχούν στις μη μηδενικές ιδιοτιμές κι αντίστοιχα οι $\mathbf{U_0}$ και $\mathbf{V_0}$ περιέχουν τα ιδιοδιανύσματα που αντιστοιχούν στις μηδενικές ιδιοτιμές τότε είναι, με βάση την (2.26):

$$\boldsymbol{G} = \boldsymbol{U}\,\boldsymbol{\Lambda}\,\boldsymbol{V^T} = \boldsymbol{U_p\Lambda_pV_p^T} \qquad (2.27)$$

από όπου φαίνεται ότι ο πίνακας **G** καθορίζεται πλήρως από τους $\mathbf{U_p}$ και $\mathbf{V_p}$, ενώ οι $\mathbf{U_0}$ και $\mathbf{V_0}$ περιττεύουν. Οι τιμές $\lambda_i$ ονομάζονται ιδιάζουσες τιμές (singular values) του πίνακα **G** και τα $\mathbf{u_i}$ και $\mathbf{v_i}$ ιδιάζοντα διανύσματα του **G**. Οι νέοι πίνακες είναι αριστερά-ορθογώνιοι:

$$\boldsymbol{U_p^TU_p = I_N, V_p^TV_p = I_M} \qquad (2.28)$$

Ο υποχώρος $\mathbf{U_0}$ (αν υπάρχει) αποτελεί ένα κομμάτι του χώρου **U** των δεδομένων, το οποίο δεν μπορούμε να εξερευνήσουμε. Για οποιοδήποτε μοντέλο, τα συνθετικά δεδομένα $\hat{\mathbf{d}}$ που παράγονται ανήκουν στο υποχώρο $\mathbf{U_p}$. Έτσι, αν τα πραγματικά δεδομένα έχουν συνιστώσα στον $\mathbf{U_0}$, δεν θα είναι ποτέ δυνατό να βρεθεί λύση που να μηδενίζει τα υπόλοιπα **d**-**Gm**. Αυτό φαίνεται καλύτερα από την παρακάτω σχέση:

$$\boldsymbol{Gm} = \widehat{\boldsymbol{d}} \Rightarrow \boldsymbol{U_0^T G m} = \boldsymbol{U_0^T \widehat{d}} \Rightarrow \boldsymbol{U_0^T U_p \Lambda_p V_p^T} = \boldsymbol{U_0^T \widehat{d}} \Rightarrow \boldsymbol{U_0^T \widehat{d}} = \boldsymbol{0} \quad (2.29)$$

Αντίστοιχα, δεν μπορούμε να εξερευνήσουμε τον υποχώρο $\mathbf{V_0}$ (αν υπάρχει) του χώρου **V** των παραμέτρων του μοντέλου. Ως αποτέλεσμα, υπάρχουν άπειρες λύσεις που δίνουν ίδια υπόλοιπα **d**-**Gm.** Μπορούμε να προσθέσουμε οποιοδήποτε διάνυσμα του $\mathbf{V_0}$ στη λύση χωρίς να αλλάξουν να συνθετικά δεδομένα. Πράγματι, έστω ότι $\mathbf{m}=\mathbf{m_p}+\mathbf{m_0}$ , όπου $\mathbf{m_p}$ και $\mathbf{m_0}$ είναι οι συνιστώσες του **m** από τους χώρους $\mathbf{V_p}$ και $\mathbf{V_0}$ αντίστοιχα. Τότε είναι:

$$\boldsymbol{Gm} = \boldsymbol{G}\left(\boldsymbol{m_p} + \boldsymbol{m_0}\right) = \boldsymbol{U_p \Lambda_p V_p^T}\left(\kappa \boldsymbol{V_p} + \lambda \boldsymbol{V_0}\right) = \kappa \boldsymbol{U_p \Lambda_p V_p^T V_p} + \lambda \boldsymbol{U_p \Lambda_p V_p^T V_0} = \kappa \boldsymbol{U_p \Lambda_p} \quad (2.30)$$

Με βάση τη σχέση (2.25) μπορούμε να δημιουργήσουμε έναν γενικευμένο αντίστροφο του **G**:

$$\boldsymbol{G_g^{-1}} \;=\; \boldsymbol{V_p \Lambda_p^{-1} U_p^T} \quad (2.31)$$

ο οποίος δίνει τη λύση $\mathbf{m_g}= \mathbf{G_g}^{-1}\,\mathbf{d}$ που ισοδυναμεί με τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου (damped least squares). Πιο συγκεκριμένα, αν υπάρχει ο χώρος $\mathbf{U_0}$, η (2.31) ελαχιστοποιεί την ποσότητα $|\mathbf{d}\text{-}\mathbf{Gm}|^2$ κι αν υπάρχει ο χώρος $\mathbf{V_0}$, η (2.31) επιλέγει τη λύση με το ελάχιστο μέτρο ($|\mathbf{m}|^2 \rightarrow$min). Προφανώς, αν υπάρχουν και οι δύο αυτοί χώροι, η λύση που μας δίνει η SVD, επιτυγχάνει και τα δύο. Ο πίνακας διακριτικής ικανότητας είναι:

$$\boldsymbol{R} = \boldsymbol{G_g^{-1} G} = \; \boldsymbol{V_p \Lambda_p^{-1} U_p^T U_p \Lambda_p V_p^T} = \boldsymbol{V_p V_p^T} \quad (2.32)$$

από όπου φαίνεται ότι αν $\mathbf{V_0}=\mathbf{0}$ (ο χώρος μηδενικής διάστασης)**,** τότε **R**=**I** κι έχουμε τέλεια διακριτική ικανότητα, κάτι που δεν ισχύει αν $\mathbf{V_0}\neq\mathbf{0}$**.** Ο πίνακας συμμεταβλητότητας δίνεται από τη σχέση:

$$\boldsymbol{C_{m_g}} = \boldsymbol{G_g^{-1} C_d (G_g^{-1})^T} = \boldsymbol{V_p \Lambda_p^{-1} U_p^T \sigma_d^2 (V_p \Lambda_p^{-1} U_p^T)^T} = \; \boldsymbol{\sigma_d^2 V_p \Lambda_p^{-2} V_p^T} \quad (2.33)$$

## 2.4 Επίλυση μεγάλων γραμμικών συστημάτων

Τα γραμμικά ή γραμμικοποιημένα προβλήματα που απαντώνται στην πραγματικότητα στη Γεωφυσική (και όχι μόνο) είναι συχνά πολύ μεγάλα, γεγονός που καθιστά τον απευθείας υπολογισμό της λύσης με τη σχέση (2.9) ιδιαίτερα απαιτητικό ως προς τον υπολογιστικό χρόνο και τη μνήμη που πρέπει να δεσμευτεί στον υπολογιστή, ή ίσως και πρακτικά αδύνατο. Ακόμη κι αν υποθέσουμε ότι το τετραγωνικό σύστημα $\mathbf{(G^TG)x=G^Td}$ δεν θα λυθεί με την εύρεση του αντιστρόφου του πίνακα $\mathbf{G^TG}$, αλλά με κάποια άλλη πιο οικονομική μέθοδο (π.χ. LU), πάλι σε γραμμικά συστήματα με πολλές παραμέτρους ο υπολογιστικός χρόνος είναι πολύ μεγάλος, με αποτέλεσμα να συναντώνται δυσκολίες. Για παράδειγμα, όταν απαιτείται πολλαπλή εφαρμογή της λύσης ελαχίστων τετραγώνων, όπως στα γραμμικοποιημένα προβλήματα, τότε και ο υπολογιστικός χρόνος πολλαπλασιάζεται ανάλογα. Αν όμως θέλουμε να έχουμε και τα μέτρα ποιότητας της λύσης, -τους πίνακες συμμεταβλητότητας και διακριτικής ικανότητας- εκτός από τη λύση, τότε απαιτείται μεταξύ άλλων ο υπολογισμός του αντιστρόφου του $\mathbf{G^TG}$.

Τα προβλήματα αυτά έχουν οδηγήσει στην εύρεση άλλων τρόπων υπολογισμού τόσο της λύσης, όσο και κάποιων μέτρων ποιότητας αυτής. Πρόκειται για προσεγγιστικές μεθόδους, οι οποίες μπορούν να δώσουν ικανοποιητικά αποτελέσματα, με κάποιες προϋποθέσεις. Στη συνέχεια παρουσιάζονται μια επαναληπτική μέθοδος εύρεσης λύσης, η LSQR (Paige and Saunders, 1982), η οποία είναι από τις πιο ευρέως χρησιμοποιούμενες, καθώς και δύο προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας.

### 2.4.1 Η μέθοδος LSQR

Η μέθοδος LSQR είναι μια επαναληπτική μέθοδος συζυγών διευθύνσεων η οποία χρησιμοποιείται για την επίλυση των εξής προβλημάτων:

1. Επίλυση του συστήματος $\mathbf{Gx} = \mathbf{d}$
2. Επίλυση του συστήματος $\mathbf{Gx} = \mathbf{b}$ με τη μέθοδο των ελαχίστων τετραγώνων, δηλαδή ελαχιστοποίηση του $|\mathbf{Gx}-\mathbf{d}|^2$
3. Επίλυση του συστήματος $\mathbf{Gx} = \mathbf{d}$ με τη μέθοδο damped least squares, δηλαδή ελαχιστοποίηση του $|\mathbf{Gx}-\mathbf{d}|^2 + \lambda^2|\mathbf{x}|^{\mathbf{2}}$

Συνήθως, το όριο ανεκτικότητας σχετικά με την εγγύτητα της λύσης που δίνει η LSQR με τη λύση ελαχίστων τετραγώνων στην οποία θεωρητικά συγκλίνει καθορίζεται από το χρήστη. Το μεγάλο πλεονέκτημα της μεθόδου είναι η ελαχιστοποίηση του υπολογιστικού χρόνου. Η διαφορά γίνεται εμφανής σε μεγάλα γραμμικά συστήματα, καθώς ο χρόνος που απαιτεί περιορίζεται σημαντικά. Αυτό οφείλεται εν μέρει στο ότι η LSQR δεν υπολογίζει τον αντίστροφο $\mathbf{(G^TG)^{-1}}$ που συχνά απαιτεί ο απευθείας υπολογισμός της λύσης.

Συνοπτικά, στο πρώτο βήμα γίνονται οι εξής πράξεις:

$$\beta_1 = |\boldsymbol{d}|, \ \boldsymbol{u}_1 = \boldsymbol{d}/\beta_1 \qquad (2.34)$$

$$\boldsymbol{v_1} = \boldsymbol{G^T u}_1, \ \alpha_1 = |\boldsymbol{v_1}|, \boldsymbol{v}_1 = \boldsymbol{v_1}/\alpha_1 \qquad (2.35)$$

ενώ στο i-στο βήμα γίνονται οι εξής πράξεις:

$$\boldsymbol{u_i} = \boldsymbol{G}\boldsymbol{v_{i-1}} - \alpha_{i-1}\boldsymbol{u_{i-1}},\ \beta_i = |\boldsymbol{u_i}|,\ \boldsymbol{u_i} = \boldsymbol{u_i}/\beta_i \quad (2.36)$$

$$\boldsymbol{v_i} = \boldsymbol{G^T}\boldsymbol{u_i} - \beta_i\boldsymbol{v_{i-1}},\alpha_i = |\boldsymbol{v_i}|,\ \boldsymbol{v_i} = \boldsymbol{v_i}/\alpha_i \quad (2.37)$$

Οι παραπάνω σχέσεις μπορούν να γραφούν σε μορφή πινάκων ως εξής:

$$\boldsymbol{U_{k+1}}(\beta_1\boldsymbol{e_1}) = \boldsymbol{b} \quad (2.38)$$

$$\boldsymbol{G}\boldsymbol{V_k} = \boldsymbol{U_{k+1}}\boldsymbol{B_k} \quad (2.39)$$

$$\boldsymbol{G^T}\boldsymbol{U_{k+1}} = \boldsymbol{V_k}\boldsymbol{B_k^T} + \alpha_{k+1}\boldsymbol{v_{k+1}}\boldsymbol{e_{k+1}^T} \quad (2.40)$$

όπου **B**$_k$ = $\begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}$ , **U**$_k$ = [**u**$_1$, **u**$_2$, … , **u**$_k$], **V**$_k$ = [**v**$_1$, **v**$_2$, … , **v**$_k$] και **e**$_i$ είναι η i-οστή στήλη ενός μοναδιαίου πίνακα ανάλογης διάστασης. Αν οι πράξεις γίνονταν με απόλυτη ακρίβεια, θα ίσχυε $\mathbf{U}^T{}_{k+1}\mathbf{U}_{k+1}$ = **I** και $\mathbf{V}^T{}_k\mathbf{V}_k$ = **I**.

Έστω τώρα ένα διάνυσμα **y**$_k$, με βάση το οποίο ορίζουμε:

$$\boldsymbol{x_k} = \boldsymbol{V_k}\boldsymbol{y_k} \quad (2.41)$$

$$\boldsymbol{t_{k+1}} = \beta_1\boldsymbol{e_1} - \boldsymbol{B_k}\boldsymbol{y_k} \quad (2.42)$$

Προκύπτει τότε από τις (2.38), (2.39), (2.41) και (2.42) ότι:

$$\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x_k} = \boldsymbol{U_{k+1}}\boldsymbol{t_{k+1}} \quad (2.43)$$

Ο πίνακας **U**$_{k+1}$ είναι (θεωρητικά) ορθοκανονικός και σίγουρα φραγμένος. Επομένως η ελαχιστοποίηση της ποσότητας |**b**-**Ax**$_k$|, που είναι και το ζητούμενο για την επίλυση με τη μέθοδο ελαχίστων τετραγώνων, ισοδυναμεί με την ελαχιστοποίηση της ποσότητας |**t**$_{k+1}$| = |β$_1$**e**$_1$-**B**$_k$**y**$_k$|. Προφανώς η επίλυση αυτού του προβλήματος είναι πιο εύκολη από την επίλυση του αρχικού. Αρκεί δηλαδή να υπολογιστεί το **t**$_{k+1}$ αντί του **x**$_k$. Χρησιμοποιώντας μάλιστα τη συνήθη ανάλυση **QR** του πίνακα **B** (όπου ισχύει **Q**$_k$**B**$_k$ = $\begin{pmatrix}\boldsymbol{R_k}\\ 0\end{pmatrix}$ με ανάλογες διαστάσεις), είναι:

$$\boldsymbol{Q_k}[\boldsymbol{B_k}\ \beta_1\boldsymbol{e_1}] = \begin{bmatrix}\boldsymbol{R_k} & \boldsymbol{f_k} \\ 0 & \bar{\varphi}_{k+1}\end{bmatrix} \quad (2.44)$$

όπου μάλιστα **R**$_k$= $\begin{pmatrix} \rho_1 & \theta_2 & & & \\ & \rho_2 & \theta_3 & & \\ & & \ddots & \ddots & \\ & & & \rho_{k-1} & \theta_k \\ & & & & \rho_k \end{pmatrix}$ , **f**$_k$= [φ$_1$ φ$_2$ … φ$_k$]$^T$, και για τον **Q**$_k$

ισχύει $\mathbf{Q_k}= \mathbf{Q_{k,k+1}} \dots \mathbf{Q_{2,3}Q_{1,2}}$. Το τελευταίο είναι ένα γινόμενο πινάκων-περιστροφών το οποίο απαλείφει τα στοιχεία $\beta_2$, $\beta_3$, ... , $\beta_k$ του πίνακα $\mathbf{B_k}$ που βρίσκονται κάτω από τη διαγώνιο. Τότε τα διανύσματα $\mathbf{y_k}$ και $\mathbf{t_{k+1}}$ μπορούν να υπολογιστούν από τις:

$$\boldsymbol{R_k y_k = f_k} \quad (2.45)$$

$$\boldsymbol{t_{k+1} = Q_k^T} \left[ \begin{matrix} \mathbf{0} \\ \boldsymbol{\bar{\varphi}_{k+1}} \end{matrix} \right] \quad (2.46)$$

Η μέθοδος LSQR είναι, όπως αναφέρθηκε, μέθοδος συζυγών διευθύνσεων και ως τέτοια θα πρέπει, θεωρητικά, να συγκλίνει στην ακριβή λύση σε m το πολύ βήματα (όπου m το πλήθος των αγνώστων). Κάτι τέτοιο θα ίσχυε αν οι πράξεις γίνονταν με απόλυτη ακρίβεια. Ωστόσο, ακόμα κι αν χρειαστούν περισσότερες επαναλήψεις, το υπολογιστικό έργο είναι μικρότερο από την επίλυση του προβλήματος με κάποια αναλυτική μέθοδο.

Οι απαιτήσεις σε μνήμη είναι κι αυτές περιορισμένες, αλλά μπορούν να μειωθούν περαιτέρω. Ο πίνακας [$\mathbf{R_k}$ $\mathbf{f_k}$] είναι ίδιος με τον [$\mathbf{R_{k-1}}$ $\mathbf{f_{k-1}}$] με μοναδική διαφορά ότι προστίθενται μια επιπλέον σειρά και μια στήλη. Συνδυάζοντας τις (2.41) και (2.45), έχουμε:

$$\boldsymbol{x_k = V_k R_k^{-1} f_k = D_k f_k} \quad (2.47)$$

όπου οι στήλες του πίνακα $\mathbf{D_k}$= [$\mathbf{d_1}$ $\mathbf{d_2}$ … $\mathbf{d_k}$] μπορούν να υπολογιστούν από το σύστημα $\mathbf{R_k^T D_k^T}$= $\mathbf{V_k^T}$ με προς τα εμπρός αντικατάσταση. Θέτοντας $\mathbf{d_0}$= $\mathbf{x_0}$= $\mathbf{0}$, προκύπτει:

$$\boldsymbol{D_k} = \frac{\mathbf{1}}{\boldsymbol{\rho_k}}(\boldsymbol{v_k} - \theta_k \boldsymbol{d_{k-1}}) \quad (2.48)$$

$$\boldsymbol{x_k = x_{k-1} + \varphi_k d_k} \quad (2.49)$$

και κατά συνέπεια δεν χρειάζεται να αποθηκευτούν παρά μόνο οι τελευταίες επαναλήψεις της μεθόδου.

## 2.4.2 Προσεγγιστικές μορφές πίνακα συμμεταβλητότητας και διακριτικής ικανότητας

Η αδυναμία της LSQR, όπως και όλων των προσεγγιστικών μεθόδων υπολογισμού της λύσης γραμμικών συστημάτων, να υπολογίσει τα μέτρα ποιότητας της λύσης, έχει οδηγήσει στο να προταθούν προσεγγιστικές μορφές των πινάκων αυτών (π.χ. Nolet *et al.* 1999). Η ύπαρξη του τελεστή $\mathbf{L=G^{-g}}$ , δηλαδή ενός γενικευμένου αντιστρόφου προϋποθέτει ότι ισχύει:

$$\boldsymbol{GLG = G} \quad (2.50)$$

ή, κατά προσέγγιση:

$$\boldsymbol{GL_2} \cong \boldsymbol{I_n} \text{ και } \boldsymbol{L_3G} \cong \boldsymbol{I_m} \quad (2.51)$$

Επιλύοντας καθεμιά από τις παραπάνω εξισώσεις, καταλήγουμε σε δύο προσεγγιστικές μορφές του αντίστροφου τελεστή. Τα συστήματα (2.51) δεν επιλύονται αναλυτικά, διότι κάτι τέτοιο θα απαιτούσε πολύ υπολογιστικό χρόνο και δέσμευση μεγάλου τμήματος της μνήμης του υπολογιστή, κάτι το οποίο θέλουμε να αποφύγουμε. Ο στόχος της διαδικασίας που περιγράφεται είναι να δώσει κάποιες προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας σε χρόνο σημαντικά μικρότερο από αυτόν που απαιτείται για τον αναλυτικό υπολογισμό των πινάκων αυτών.

Οι εξισώσεις (2.51) επιλύονται για κάθε διάνυσμα-στήλη των πινάκων $\mathbf{L_2}$ και $\mathbf{L_3}$ ξεχωριστά, χρησιμοποιώντας τη μέθοδο της προς τα πίσω προβολής, επιβάλλοντας δηλαδή ότι η επόμενη προσέγγιση θα πρέπει να βρίσκεται στη διεύθυνση της αρνητικής κλίσης $-\nabla$ της προηγούμενης προσέγγισης. Η μέθοδος είναι επαναληπτική αλλά πραγματοποιείται μόνο μια επανάληψη. Ορίζοντας αρχικά την k-στήλη του πίνακα $\mathbf{L_2}$ ως $\mathbf{c^k}$ και αντίστοιχα την k-στήλη του μοναδιαίου πίνακα $\mathbf{I_n}$ ως $\mathbf{e^k}$ έχουμε να λύσουμε για κάθε k= 1,2,…,n το σύστημα:

$$\boldsymbol{Gc^k} = \boldsymbol{e^k} \quad (2.52)$$

Ως αρχική προσέγγιση θεωρούμε το μηδενικό διάνυσμα $\mathbf{c_0^k}=\mathbf{0}$. Η διεύθυνση της προς τα πίσω προβολής είναι το m-διάστατο διάνυσμα:

$$\boldsymbol{v^k} \equiv -\boldsymbol{G^T}(\boldsymbol{Gc_0^k} - \boldsymbol{e^k}) = \boldsymbol{G^T e^k} \quad (2.53)$$

άρα το i-οστό στοιχείο του $\mathbf{v^k}$ είναι ίσο με $\mathbf{G_{ik}^T}=\mathbf{G_{ki}}$. Επομένως το $\mathbf{c^k}$ θα πρέπει να βρίσκεται στη διεύθυνση του $\mathbf{v^k}$. Είναι:

$$\boldsymbol{c^k} = a_k \boldsymbol{v^k} \quad (2.54)$$

Για να υπολογίσουμε το $\alpha_k$, επιβάλλουμε τη συνθήκη το υπόλοιπο να ελαχιστοποιείται, οπότε το διάνυσμα των υπολοίπων θα πρέπει να είναι κάθετο με το διάνυσμα $\mathbf{Gc^k}$, δηλαδή $< \alpha_k\mathbf{Gc^k} - \mathbf{e^k}, \alpha_k\mathbf{Gc^k} >$, από όπου υπολογίζουμε την τιμή του συντελεστή $\alpha_k$:

$$a_k = \frac{\sum_{j=1}^{m} A_{kj}^2}{\sum_{i=1}^{n}\left(\sum_{j=1}^{m} A_{ij}A_{kj}\right)\left(\sum_{q=1}^{m} A_{iq}A_{kq}\right)} \quad (2.55)$$

Εφόσον $\mathbf{(L_2)_{ik}}= \alpha_k \mathbf{G_{ki}}$, η λύση μπορεί να υπολογιστεί εύκολα από τη σχέση:

$$\boldsymbol{L_2} = \boldsymbol{G^T D} \quad (2.56)$$

όπου $\mathbf{D}$ είναι ένας διαγώνιος n×n πίνακας με:

$$\boldsymbol{D_{kk}} = \frac{(\boldsymbol{GG^T})_{kk}}{\sum_{i=1}^{n}(\boldsymbol{GG^T})_{ik}^2} \qquad (2.57)$$

Αντίστοιχα, η προσεγγιστική επίλυση της δεύτερης εξίσωσης με την ίδια μέθοδο δίνει:

$$\boldsymbol{L_3} = \boldsymbol{D'G^T} \qquad (2.58)$$

όπου **D’** είναι ένας διαγώνιος m×m πίνακας με:

$$\boldsymbol{D'_{kk}} = \frac{(\boldsymbol{G^TG})_{kk}}{\sum_{i=1}^{m}(\boldsymbol{G^TG})_{ik}^2} \qquad (2.59)$$

Η πρώτη προσέγγιση ικανοποιεί στο μεγαλύτερο δυνατό βαθμό τη συνθήκη $\mathbf{GL=I_n}$, δηλαδή την $\mathbf{GL\hat{d}=\hat{d}}$. Το σφάλμα μεταξύ των παρατηρησιακών και των πραγματικών δεδομένων ελαχιστοποιείται, ωστόσο αυτή η λύση συνεπάγεται μικρή διακριτική ικανότητα του μοντέλου. Αντίθετα, η δεύτερη προσεγγιστική μορφή μεγιστοποιεί τη διακριτική ικανότητα των παραμέτρων του μοντέλου.

Μια πιο ποιοτική εκτίμηση του αντίστροφου τελεστή θα απαιτούσε πολλαπλές επαναλήψεις της μεθόδου της προς τα πίσω προβολής, κάτι τέτοιο όμως θα σήμαινε πολλαπλάσιο υπολογιστικό έργο. Η ακρίβεια των προσεγγίσεων εναπόκειται στην αποτελεσματικότητα του πρώτου βήματος της προς τα πίσω προβολής. Για σποραδικούς πίνακες, το πρώτο βήμα δίνει το μεγαλύτερο κομμάτι της συνολικής σύγκλισης που μπορεί να δώσει η μέθοδος. Έτσι, αναμένεται οι προσεγγιστικές μορφές να οδηγήσουν σε κάποια αρχικά συμπεράσματα, δεδομένου ότι οι πίνακες $\mathbf{GG^T}$ και $\mathbf{G^TG}$ θα είναι έντονα διαγώνια κυρίαρχοι, αν ο πίνακας **G** είναι σποραδικός, όπως συμβαίνει στα πραγματικά γεωφυσικά προβλήματα.

## 2.5 Η γλώσσα προγραμματισμού MATLAB

Η MATLAB είναι ένα μαθηματικό λογισμικό πακέτο που αποτελεί ένα από τα πιο ευρέως χρησιμοποιούμενα περιβάλλοντα προγραμματισμού και ταυτόχρονα ένα πολύ ισχυρό υπολογιστικό εργαλείο. Χρησιμοποιείται σε ερευνητικές και άλλες εφαρμογές οποιουδήποτε επιστημονικού κλάδου απαιτεί υπολογισμούς που σχετίζονται με υπολογιστικά και εφαρμοσμένα μαθηματικά. Το όνομά της προέρχεται από τις λέξεις MATrix LABoratory, καθώς η γλώσσα στηρίζεται στην επεξεργασία και τις πράξεις πινάκων, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών, η svd και η αντιστροφή πινάκων κλπ.

Η MATLAB έχει κατασκευαστεί ώστε να επιλύει τα προβλήματα χρησιμοποιώντας αριθμητική πεπερασμένης ακρίβειας, δηλαδή δεν βρίσκει ακριβή αλλά προσεγγιστική λύση. Η MATLAB επιτρέπει επιπλέον την οπτικοποίηση-σχεδίαση δεδομένων με ποικίλους τρόπους απεικόνισης όπως γραφήματα, ιστογράμματα, ραβδογράμματα κ.α., γεγονός που το καθιστά ιδιαίτερα φιλικό προς το χρήστη. Τα γραφήματα στο MATLAB δεν είναι παρά σύνολα σημείων τα οποία παρουσιάζονται με τη μορφή πινάκων ή διανυσμάτων.

Επιπλέον, η εισαγωγή και η εξαγωγή δεδομένων μπορούν να γίνουν με εύκολο, σχεδόν αυτόματο τρόπο, και μάλιστα σε μορφή συμβατή με άλλα προγράμματα, όπως το Excel του Microsoft Office.

### 2.5.1 Προγραμματισμός στη MATLAB

Το πακέτο MATLAB περιλαμβάνει πάρα πολλούς έτοιμους αλγορίθμους, υπορουτίνες και συναρτήσεις, οι οποίες μπορούν να κληθούν άμεσα από το χρήστη. Πρόσθετα, ο χρήστης έχει τη δυνατότητα να τροποποιήσει αυτούς τους αλγόριθμους, ανάλογα με τις ανάγκες του. Η κατασκευή νέων προγραμμάτων καθίσταται έτσι ιδιαίτερα εύκολη κι απλή, καθώς οι ρουτίνες που αντιστοιχούν σε ευρέως χρησιμοποιούμενες μαθηματικές μεθόδους δεν χρειάζεται να συνταχθούν. Τα νέα προγράμματα που δημιουργούνται αποθηκεύονται ως αρχεία (m-files) και προστίθενται στη βιβλιοθήκη του MATLAB, μαζί με τα ήδη υπάρχοντα που αντιστοιχούν στις έτοιμες και μπορούν να χρησιμοποιηθούν όπως και οι άλλες υπορουτίνες. Μάλιστα, νέοι κώδικες μπορούν να ενταχθούν σε επόμενες εκδόσεις της MATLAB ως συναρτήσεις βιβλιοθήκης, αφού αποσταλούν στην εταιρία που τη διανέμει.

Τα m-files ονομάζονται έτσι γιατί φέρουν το επίθεμα «.m». Μπορούν να δημιουργηθούν με οποιονδήποτε συντάκτη (editor), συνήθως όμως χρησιμοποιείται για το σκοπό αυτόν ο συντάκτης της MATLAB. Τα αρχεία αυτά διακρίνονται σε δύο μεγάλες κατηγορίες: τα αρχεία εντολών (script files) και τα αρχεία συναρτήσεων (function files), που διέπονται από διαφορετικούς κανόνες λειτουργίας.

#### I. Αρχεία εντολών

Τα αρχεία εντολών περιέχουν μια ακολουθία εντολών που εκτελείται όταν καλείται το αρχείο. Το αρχείο μπορεί να κληθεί είτε γράφοντας το όνομά του στο παράθυρο εντολών της MATLAB (ή και μέσα σε κάποιο άλλο m-file), είτε πατώντας το σχετικό πλήκτρο στο συντάκτη της MATLAB, έχοντας ήδη ανοίξει το αρχείο.

Τα αρχεία εντολών δρουν σε μεταβλητές του χώρου εργασίας και μπορούν να τις χρησιμοποιήσουν ή και να τις αλλάξουν. Επίσης, τα αρχεία εντολών πολύ συχνά δημιουργούν νέες μεταβλητές στο χώρο εργασίας, οι οποίες μπορούν στη συνέχεια να αποθηκευτούν.

**II. Αρχεία συναρτήσεων**

Τα αρχεία συναρτήσεων περιέχουν κι αυτά μια ακολουθία εντολών, αλλά δέχονται ορίσματα (μεταβλητές) εισόδου και υπολογίζουν κι επιστρέφουν τις μεταβλητές εξόδου. Οι μεταβλητές εισόδου, καθώς κι οι ενδιάμεσες μεταβλητές που βοηθούν στους υπολογισμούς, δεν μεταφέρονται στο χώρο εργασίας, είναι δηλαδή εσωτερικές μεταβλητές. Κάθε τέτοιο αρχείο ξεκινάει υποχρεωτικά με μια γραμμή της μορφής:

function [outputs]=name(inputs)

όπου outputs είναι οι μεταβλητές εξόδου και inputs είναι οι μεταβλητές εισόδου και χωρίζονται μεταξύ τους με κόμμα. Name είναι το όνομα της συνάρτησης και θα πρέπει να ταυτίζεται με το όνομα του αρχείου, όπως επίσης και να μην ταυτίζεται με το όνομα καποιας ήδη υπάρχουσας συνάρτησης.

Για να κληθεί μια συνάρτηση, είτε στο χώρο εργασίας είτε σε κάποιο άλλο m-file, αρκεί να γράψουμε το όνομα της συνάρτησης και σε παρένθεση τα ορίσματα εισόδου για τα οποία επιθυμούμε να υπολογίσουμε την τιμή ή τις τιμές της συνάρτησης.

Τα αρχεία συναρτήσεων μπορούν επίσης να περιέχουν υποσυναρτήσεις, που ορίζονται μέσα στο αρχείο, όπως και οι κανονικές συναρτήσεις, αλλά μπορούν να χρησιμοποιηθούν μόνο μέσα στο m-file, δηλαδή δεν μπορούν να κληθούν από το χώρο εργασίας ή από άλλο αρχείο.

### 2.5.2 Γραμμικά συστήματα στη MATLAB

Δεδομένου ότι οι μεταβλητές στη MATLAB έχουν τη μορφή πινάκων, η επίλυση γραμμικών συστημάτων είναι ιδιαίτερα απλή. Αρχικά, πρέπει να επισημανθεί ότι οι πράξεις πρόσθεση (+), αφαίρεση (-) και πολλαπλασιασμός (*) μπορούν να πραγματοποιηθούν τόσο σε αριθμούς, όσο και σε πίνακες. Όσον αφορά τη διαίρεση, η MATLAB διακρίνει ανάμεσα σε αριστερή διαίρεση (\) και δεξιά διαίρεση (/). Για αριθμούς, οι παραστάσεις β\α και α/β είναι ισοδύναμες κι υπολογίζουν το πηλίκο του α διά το β. Αν τώρα **A** είναι ένας πίνακας και **b** ένα διάνυσμα, η παράσταση **A****b** δίνει τη λύση του συστήματος **Ax**=**b**, ενώ η **A**/**b** δίνει τη λύση του συστήματος **xA**=**b**.

Στην περίπτωση που ο **A** είναι τετραγωνικός πίνακας, αν δεν είναι ιδιάζων, η λύση του συστήματος είναι η $\mathbf{A}^{-1}\mathbf{b}$ και η $\mathbf{b}\mathbf{A}^{-1}$ αντίστοιχα. Αν ο **A** δεν είναι τετραγωνικός πίνακας, τότε η MATLAB με την πράξη της διαίρεσης, είτε αριστερής είτε δεξιάς, υπολογίζει την αντίστοιχη λύση ελαχίστων τετραγώνων, με την προϋπόθεση ότι αυτή υπάρχει. Φυσικά, αν οι διαστάσεις των **A** και **b** δεν συμφωνούν, η MATLAB εμφανίζει το ανάλογο μήνυμα σφάλματος. Επίσης, στις περιπτώσεις που α) ο **A** τετραγωνικός αλλά ιδιάζων (ή σχεδόν ιδιάζων), ή β) ο **A** δεν είναι τετραγωνικός και ο $\mathbf{A}^{\mathbf{T}}\mathbf{A}$ είναι ιδιάζων (ή πρακτικά σχεδόν ιδιάζων), τότε η MATLAB δίνει τη (μια) λύση αλλά επισημαίνει ότι η λύση είναι ασταθής.

### 2.5.3 Πίνακες στη MATLAB

Η MATLAB έχει κάποιες εντολές ώστε ο χρήστης να μπορεί γρήγορα να ορίσει κάποιους βασικούς πίνακες. Αυτές οι εντολές είναι οι εξής: **eye**, **zeros**, **ones**, **rand**, **randn** οι οποίες συντάσσονται ακολουθούμενες από την παρένθεση (m,n) και παράγουν έναν m×n πίνακα με τα ανάλογα χαρακτηριστικά. Οι **zeros** και **ones** παράγουν πίνακες των οποίων όλα τα στοιχεία είναι ίσα με 0 και 1 αντίστοιχα. Η **eye** παράγει έναν μοναδιαίο πίνακα, δηλαδή με διαγώνια στοιχεία ίσα με 1 κι όλα τα άλλα ίσα με 0. Οι **rand** και **randn** παράγουν πίνακες των οποίων τα στοιχεία είναι αριθμοί παρμένοι από τη συνεχή ομοιόμορφη κατανομή U(0,1) και την κανονική κατανομή N(0,1) αντίστοιχα. Προφανώς, κάθε φορά που καλείται μια από αυτές τις εντολές δημιουργείται διαφορετικός πίνακας.

Επιπλέον, η MATLAB προσφέρει έτοιμες συναρτήσεις βιβλιοθήκης για διανύσματα και πίνακες, που είναι πολύ χρήσιμες γιατί πραγματοποιούν υπολογι-σμούς που χρειάζονται συχνά στην πράξη. Κάποιες από αυτές τις συναρτήσεις είναι οι: **max min sum prod mean std median**, οι οποίες δέχονται ως όρισμα ένα διάνυσμα κι επιστρέφουν μια τιμή, η οποία είναι αντίστοιχα το μέγιστο, το ελάχιστο, το άθροισμα, το γινόμενο, η μέση τιμή, η τυπική απόκλιση και η διάμεσος των στοιχείων του διανύσματος. Οι ίδιες συναρτήσεις μπορούν να δεχτούν ως όρισμα έναν πίνακα και τότε εκτελούν την ίδια διαδικασία στα διανύσματα-στήλες του πίνακα, επιστρέφοντας ένα διάνυσμα-γραμμή.

Υπάρχουν επίσης συναρτήσεις που αφορούν αποκλειστικά πίνακες, όπως οι: **diag size det trace rank inv eig svd**. Η πρώτη επιστρέφει ένα διάνυσμα με τα διαγώνια στοιχεία του πίνακα ενώ η δεύτερη επιστρέφει ένα διάνυσμα με στοιχεία τις διαστάσεις του πίνακα. Οι **det**, **trace** και **rank** επιστρέφουν ως τιμές την ορίζουσα, το ίχνος και τη βαθμίδα του πίνακα αντίστοιχα. Η **inv** υπολογίζει τον αντίστροφο ενός τετραγωνικού πίνακα. Οι **eig** και **svd** επιστρέφουν ένα διάνυσμα με τις ιδιοτιμές και τις ιδιάζουσες τιμές του πίνακα αντίστοιχα, ενώ αν ζητηθεί από το χρήστη, μπορούν να υπολογίσουν και τους αντίστοιχους πίνακες ιδιοδιανυσμάτων και ιδιαζόντων διανυσμάτων.

Τέλος η συνάρτηση **norm** υπολογίζει το μέτρο ενός διανύσματος ή το μέτρο ενός πίνακα, ανάλογα με το όρισμα που δέχεται.

# 3 Υπολογισμοί προσεγγιστικών μορφών πίνακα συμμεταβλητότητας και διακριτικής ικανότητας με τη χρήση της MATLAB

## 3.1 Δημιουργία κώδικα για την επίλυση γραμμικών προβλημάτων με τη χρήση της MATLAB

Στόχος μας ήταν η δημιουργία κώδικα ο οποίος να επιλύει το σύστημα (2.2) με όλους τους δυνατούς τρόπους και παράλληλα να δίνει τους πίνακες συμμεταβλητότητας και διακριτικής ικανότητας, τόσο στην ακριβή όσο και στην προσεγγιστική μορφή τους. Σκοπός ήταν η εφαρμογή όλων των μεθόδων σε μικρά γραμμικά συστήματα, με αριθμό εξισώσεων από 600 μέχρι 1400 και πλήθος αγνώστων από 50 έως 800, καθώς και με κλιμακούμενο πλήθος στοιχείων ίσα με μηδέν. Στα πραγματικά γραμμικοποιημένα αντίστροφα προβλήματα (π.χ. προβλήματα τομογραφίας) εμφανίζονται πολύ συχνά σποραδικοί (sparse) πίνακες. Για να πάρουμε σποραδικούς πίνακες από πίνακες τυχαίων αριθμών, επιλέξαμε να μηδενίσουμε όλα τα στοιχεία του πίνακα τα οποία είναι μικρότερα κατά απόλυτη τιμή από ένα σταθερό αριθμό. Ο αριθμός αυτός θα αναφέρεται στο εξής ως σημείο αποκοπής ή **cut**.

Στη συνέχεια, έπρεπε να αξιολογήσουμε την αποτελεσματικότητα τόσο της μεθόδου LSQR για την εύρεση της λύσης, όσο και των δύο προσεγγίσεων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας, καθώς και του υπολογιστικού χρόνου που απαιτείται για καθεμιά λύση. Τέλος, θέλαμε να συγκρίνουμε πώς συμπεριφέρονται όλες αυτές οι παράμετροι τροποποιώντας τις διαστάσεις του προβλήματος. Τα ίδια στοιχεία έπρεπε να εξετάσουμε και στην περίπτωση που θα εφαρμόζαμε τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου. Γι' αυτούς τους υπολογισμούς χρησιμοποιήθηκε ο ίδιος κώδικας με κάποιες προσθήκες.

### 3.1.1 Η λύση ελαχίστων τετραγώνων

Ως λύσεις ελαχίστων τετραγώνων του συστήματος (2.2) **d=Gm** υπολογίσαμε τις εξής πέντε δυνατές λύσεις:

1. Τη λύση που δίνει η μέθοδος LSQR : "**xlsqr**".
2. Τη λύση που δίνει η MATLAB με τη μέθοδο ελαχίστων τετραγώνων, με επίλυση LU του συστήματος των κανονικών εξισώσεων με την εντολή **G\d** : "**xlsq**".
3. Την αναλυτική λύση ελαχίστων τετραγώνων, στην οποία υπολογίζεται η παράσταση $(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{d}$ : "**xinv**".
4. Τη λύση που δίνεται χρησιμοποιώντας την πρώτη προσεγγιστική μορφή, έστω $\mathbf{L}_2$, του αντίστροφου τελεστή **L**, και την ονομάσαμε "$\mathbf{x}_2$" : $\mathbf{x}_2=\mathbf{L}_2\,\mathbf{d}$.
5. Τη λύση που δίνεται χρησιμοποιώντας τη δεύτερη προσεγγιστική μορφή, έστω $\mathbf{L}_3$, του **L**, και την ονομάσαμε "$\mathbf{x}_3$" : $\mathbf{x}_3=\mathbf{L}_3\,\mathbf{d}$.

Για την πρώτη χρησιμοποιήσαμε κώδικα που παραχωρήθηκε από τον ερευνητή Νικόλαο Παπαδόπουλο, γραμμένος σε γλώσσα C. Ο κώδικας επιτρέπει στον χρήστη να ορίσει ο ίδιος τα επιθυμητά όρια ακρίβειας, έτσι ώστε οι επαναλήψεις να σταματούν όταν ξεπεραστεί το όριο αυτό, καθώς επίσης και το μέγιστο αριθμό επαναλήψεων. Η μετάφρασή του από C σε MATLAB δεν παρουσίασε δυσκολίες. Η LSQR μετατράπηκε σε υπορουτίνα με μορφή συνάρτησης, με ορίσματα τις διαστάσεις n και m του προβλήματος, τον πίνακα **G**, το διάνυσμα **d** και το μέγιστο αριθμό επαναλήψεων. Η συνάρτηση LSQR επιστρέφει το διάνυσμα της λύσης του προβλήματος, καθώς και το πλήθος των επαναλήψεων που τελικά χρειάστηκαν.

Ο κώδικας περιείχε υπορουτίνες υπολογισμού του γινομένου πίνακα επί διάνυσμα, οι οποίες παραλείφθηκαν κι όπου εμφανίζονταν στο κυρίως μέρος του κώδικα αντικαταστάθηκαν από την πράξη του πολλαπλασιασμού της MATLAB. Ο κώδικας περιείχε και τις αντίστοιχες υπορουτίνες για την περίπτωση που ο πίνακας ήταν σποραδικός και δινόταν σε ανάλογη μορφή. Στις δοκιμές που πραγματοποιήσαμε, δοκιμάσαμε να χρησιμοποιήσουμε την LSQR προσαρμοσμένη σε σποραδικούς πίνακες. Ωστόσο, διαπιστώσαμε ότι η πράξη του πολλαπλασιασμού ήταν γρηγορότερη για τον υπολογισμό του γινομένου, ακόμα κι αν η MATLAB δεν αναγνώριζε τον πίνακα ως σποραδικό. Έτσι, η προσαρμογή της LSQR για σποραδικούς πίνακες εγκαταλείφθηκε.

Όσον αφορά τη δεύτερη και την τρίτη λύση, αυτές είναι πρακτικά ίδιες, δηλαδή δίνουν το ίδιο αποτέλεσμα, σύμφωνα με τη MATLAB. Η διαφορά τους έγκειται στο ότι η MATLAB με την εντολή της αριστερά διαίρεσης **** δεν υπολογίζει τον αντίστροφο του πίνακα $\mathbf{G}^T\mathbf{G}$. Έτσι, για μεγάλα συστήματα η τρίτη λύση μπορεί να είναι πολύ πιο χρονοβόρα από τη δεύτερη. Ακόμη, αν ο πίνακας $\mathbf{G}^T\mathbf{G}$ είναι ιδιάζων ή σχεδόν ιδιάζων, τότε η τρίτη λύση είναι πολύ ασταθής και κατά συνέπεια όχι αξιόπιστη, σε αντίθεση με τη δεύτερη.

Η σύγκριση των τριών πρώτων λύσεων, περιλαμβάνει αφενός τον έλεγχο αν, και κατά πόσο, οι λύσεις ταυτίζονται κι αφετέρου τη σύγκριση του υπολογιστικού χρόνου για καθεμιά από αυτές. Οι χρόνοι αυτοί χρησίμευσαν επίσης στη μελέτη της συμπεριφοράς του προβλήματος, κατά τη μεταβολή των διαστάσεών του.

Όσο αφορά τις δύο τελευταίες λύσεις, υπολογίστηκαν για να δώσουν ένα επιπλέον μέτρο της ποιότητας των προσεγγίσεων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας, αν και στην πραγματικότητα δεν χρησιμοποιούνται ποτέ.

Ως λύσεις ελαχίστων τετραγώνων ελαχίστου μέτρου του συστήματος (1) **d=Gm** υπολογίσαμε τις ίδιες με προηγουμένως, με τη διαφορά ότι αντί για τους **G** και **d**, οι υπολογισμοί έγιναν με τους επαυξημένους πίνακες (βλ. σύστημα 2.17). Για την βαρύτητα ε του περιορισμού ελαχίστου μέτρου στην τελική λύση χρησιμοποιήσαμε την τιμή ε=20 για όλες τις δοκιμές.

### 3.1.2 Οι πίνακες συμμεταβλητότητας και διακριτικής ικανότητας

Υπενθυμίζουμε ότι οι δύο προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας στηρίζονται σε προσεγγιστικές μορφές του αντίστροφου τελεστή **L** ή $\mathbf{G}^{-g}$ που δρα στα δεδομένα. Ο υπολογισμός αυτού του αντίστροφου τελεστή ήταν απλός, καθώς οι πολλαπλασιασμοί πινάκων είναι στοιχειώδεις πράξεις στη MATLAB. Έτσι, η προσθήκη του υπολογισμού των προσεγγιστικών μορφών των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας ήταν εύκολος. Έτσι, για την πρώτη προσεγγιστική μορφή $\mathbf{L_2}$ έχουμε τον πίνακα συμμεταβλητότητας:

$$\boldsymbol{C_{m2} = L_2 L_2^T} \qquad (3.1)$$

και τον πίνακα διακριτικής ικανότητας:

$$\boldsymbol{R_2 = L_2 G} \qquad (3.2)$$

Αντίστοιχα, για τη δεύτερη προσεγγιστική μορφή $\mathbf{L_3}$ ο πίνακας συμμεταβλητότητας είναι:

$$\boldsymbol{C_{m3} = L_3 L_3^T} \qquad (3.3)$$

και ο πίνακας διακριτικής ικανότητας είναι:

$$\boldsymbol{R_3 = L_3 G} \qquad (3.4)$$

### 3.1.3 Σύγκριση κι εκτίμηση των αποτελεσμάτων για μεμονωμένα πειράματα

Η σύγκριση των λύσεων όσο και των μέτρων ποιότητας αυτών δεν μπορεί να πραγματοποιηθεί με την απλή παρατήρηση των αποτελεσμάτων, δεδομένου ότι οι διαστάσεις των υπό μελέτη προβλημάτων είναι πολύ μεγάλες. Επιπλέον, τα μέτρα της διαφοράς των εκτιμήσεων από τις πραγματικές παραμέτρους του προβλήματος, δηλαδή τα σχετικά σφάλματα, αν κι αρκετά χρήσιμα ως δείκτες της ποιότητας της εκτίμησης, εντούτοις δεν προσφέρουν πάντοτε μια γενική εικόνα σε ένα μεμονωμένο έλεγχο, καθώς είναι βαθμωτά μεγέθη.

Η σύγκριση των λύσεων έγινε με τη χρήση δισδιάστατων διαγραμμάτων στα οποία συγκρίναμε άμεσα τις λύσεις. Για κάθε παράμετρο του μοντέλου, αντιστοιχούμε ένα σημείο στο καρτεσιανό επίπεδο, του οποίου η τετμημένη είναι η τιμή της μιας λύσης και η τεταγμένη η τιμή μιας άλλης λύσης. Η πρώτη δηλαδή χρησιμοποιείται ως σημείο αναφοράς κι ελέγχουμε κατά πόσο η άλλη λύση ταυτίζεται με την πρώτη. Σε περίπτωση πλήρους ταύτισης, τα σημεία του γραφήματος θα έπρεπε να βρίσκονται πάνω στην κύρια διαγώνιο του καρτεσιανού επιπέδου. Ως σημείο αναφοράς επιλέξαμε την "**xlsq**", καθώς είναι η μοναδική, μαζί με την "**xinv**", ακριβής λύση και αυτή προτείνεται από τη MATLAB.

Για την εκτίμηση της ποιότητας των προσεγγιστικών μορφών του πίνακα συμμεταβλητότητας, όπου μας ενδιαφέρουν κυρίως τα διαγώνια στοιχεία, χρησιμοποιήσαμε την ίδια τακτική, τοποθετώντας στο επίπεδο τα σημεία με τετμημένη τη πραγματική διασπορά μιας παραμέτρου του μοντέλου και τεταγμένη την εκτίμησή της. Πάλι, η ποιότητα της εκτίμησης εξαρτάται από το πόσο κοντά θα βρίσκονται τα σημεία του γραφήματος στην κύρια διαγώνιο.

Τέλος, για την εκτίμηση της ποιότητας των προσεγγιστικών μορφών του πίνακα διακριτικής ικανότητας, η ίδια μέθοδος σύγκρισης χρησιμοποιήθηκε μόνο για τις περιπτώσεις όπου το πρόβλημα λύθηκε με τη μέθοδο ελαχίστων τετραγώνων ελαχίστου μέτρου. Αντίθετα, όπου εφαρμόστηκε απλή μέθοδος ελαχίστων τετραγώνων, προτιμήθηκε η σχεδίαση ιστογράμματος των διαγωνίων τιμών των προσεγγίσεων του πίνακα διακριτικής ικανότητας. Υπενθυμίζουμε, ότι ο πίνακας διακριτικής ικανότητας σε αυτήν την περίπτωση ταυτίζεται με το μοναδιαίο, επομένως η ποιότητα της προσέγγισης κρίνεται από το πόσα από τα διαγώνια στοιχεία της προσέγγισης είναι ίσα ή σχεδόν ίσα με τη μονάδα.

Στον κώδικα προστέθηκε επίσης ο υπολογισμός κάποιων βαθμωτών μεγεθών, τα οποία στα μεμονωμένα πειράματα δεν μπορούσαν να μας οδηγήσουν στην εξαγωγή συμπερασμάτων. Τα μεγέθη αυτά είναι τα εξής:

$$\mathbf{dx} = \frac{||\mathbf{xlsqr} - \mathbf{xlsq}||}{||\mathbf{xlsq}||} \qquad (3.5)$$

$$\mathbf{dx_2} = \frac{||\mathbf{x_2} - \mathbf{xlsq}||}{||\mathbf{xlsq}||} \qquad (3.6)$$

$$\mathbf{dx_3} = \frac{||\mathbf{x_3} - \mathbf{xlsq}||}{||\mathbf{xlsq}||} \qquad (3.7)$$

δηλαδή οι σχετικές διαφορές των προσεγγιστικών λύσεων από την πραγματική. Για τους πίνακες συμμεταβλητότητας και διακριτικής ικανότητας υπολογίσαμε τις σχετικές διαφορές των διανυσμάτων που έχουν τα στοιχεία των διαγωνίων των πινάκων αυτών. Έτσι, για τις δύο προσεγγιστικές μορφές του πίνακα συμμεταβλητότητας έχουμε:

$$\mathbf{dC_{m2}} = \frac{||\mathbf{diag}(\mathbf{C_{m2}} - \mathbf{C_m})||}{||\mathbf{diag}(\mathbf{C_m})||} \qquad (3.8)$$

$$\mathbf{dC_{m3}} = \frac{||\mathbf{diag}(\mathbf{C_{m3}} - \mathbf{C_m})||}{||\mathbf{diag}(\mathbf{C_m})||} \qquad (3.9)$$

Για τις προσεγγιστικές μορφές του πίνακα διακριτικής ικανότητας είναι:

$$\mathbf{dR_2} = \frac{||\mathbf{diag}(\mathbf{R_2} - \mathbf{R})||}{||\mathbf{diag}(\mathbf{R})||} \qquad (3.10)$$

$$\mathbf{dR_3} = \frac{||\mathbf{diag}(\mathbf{R_3} - \mathbf{R})||}{||\mathbf{diag}(\mathbf{R})||} \qquad (3.11)$$

### 3.1.4 Δημιουργία κώδικα για τη σύγκριση των αποτελεσμάτων κατά τη μεταβολή των διαστάσεων του προβλήματος

Τα σχετικά σφάλματα είναι εξαιρετικά χρήσιμα ως μέτρα για τη σύγκριση των αποτελεσμάτων διαφορετικών δοκιμών που υλοποιήσαμε. Μπορούμε πάντα να εξάγουμε συμπεράσματα για την ποιότητα των εκτιμήσεων, καθώς επίσης και το αν και κατά πόσο αυτή μεταβάλλεται όταν αλλάζουν οι διαστάσεις του γραμμικού συστήματος. Οι παράμετροι που καθορίζουν ένα συγκεκριμένο πρόβλημα ήταν τρεις: το πλήθος των εξισώσεων, το πλήθος των αγνώστων και το σημείο αποκοπής. Έτσι, έπρεπε να δημιουργηθεί κώδικας ώστε να επαναλαμβάνει τις δοκιμές κρατώντας σταθερές τις δύο από τις τρεις παραμέτρους και να μεταβάλλει την τρίτη.

Έχοντας ήδη τους κώδικες για την επίλυση ενός γραμμικού συστήματος και τον υπολογισμό των προσεγγιστικών μορφών των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας, τόσο με τη μέθοδο ελαχίστων τετραγώνων, όσο και με τη μέθοδο ελαχίστων τετραγώνων ελαχίστου μέτρου, το πρώτο βήμα ήταν να μετατρέψουμε τους δύο αυτούς κώδικες σε υπορουτίνες. Έτσι, οι κώδικες αυτοί έγιναν συναρτήσεις στη MATLAB με ορίσματα τις διαστάσεις του πίνακα, n και m, και το σημείο αποκοπής και με τιμές:

α) τους χρόνους υπολογισμού των **xlsqr**, **xlsq** και **xinv**.

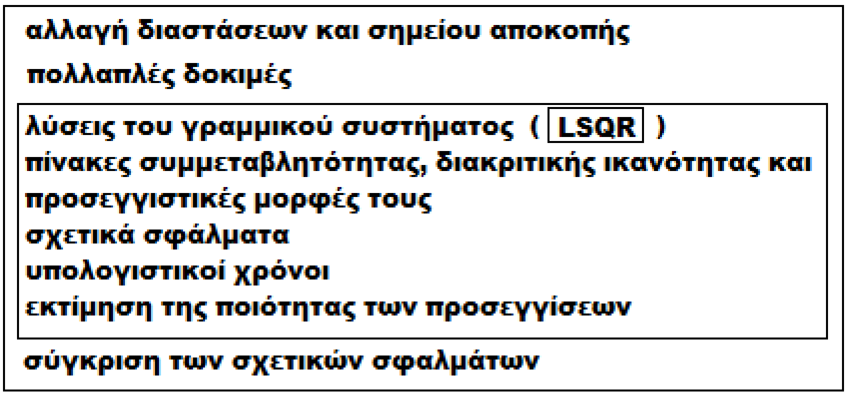
β) το σχετικό σφάλμα της **xlsqr**.

γ) τα σχετικά σφάλματα των διαγωνίων των προσεγγιστικών μορφών των πινάκων συμμεταβλητότητας.

δ) τα σχετικά σφάλματα των διαγωνίων των προσεγγιστικών μορφών των πινάκων διακριτικής ικανότητας.

Δημιουργήθηκαν νέοι κώδικες, οι οποίοι θα υλοποιούσαν την προηγούμενη διαδικασία, δηλαδή θα καλούσαν τη συνάρτηση που μόλις περιγράφηκε για όλες τις επιλεγμένες τιμές. Ωστόσο, πραγματοποιώντας μόνο μια φορά κάθε δοκιμή–επίλυση ενός τυχαίου συστήματος, υπήρχε ο κίνδυνος τα αποτελέσματα να εμφανίζουν μεταβλητότητα λόγω της δομή του τυχαίου και να μας οδηγήσουν σε λάθος συμπεράσματα. Κατά συνέπεια, δημιουργήσαμε τον κώδικα έτσι ώστε για συγκεκριμένες τιμές των παραμέτρων να καλείται πολλές φορές -συγκεκριμένα 20- η συνάρτηση και να λύνονται διαφορετικά συστήματα.

Τα αποτελέσματα από τις διάφορες δοκιμές αποθηκεύονταν σε πίνακες, έναν για κάθε μεταβλητή εξόδου της συνάρτησης, όπου τα στοιχεία μιας στήλης αντιστοιχούσαν στις δοκιμές με τις ίδιες παραμέτρους, ενώ οι διάφορες στήλες αντιστοιχούσαν στις τιμές που έπαιρνε η μεταβαλλόμενη παράμετρος. Το επόμενο βήμα ήταν ο υπολογισμός της μέσης τιμής, τυπικής απόκλισης και διαμέσου των αποτελεσμάτων, με τις αντίστοιχες εντολές της MATLAB **mean**, **std** και **median**. Οι εντολές αυτές υπολογίζουν τη μέση τιμή, τυπική απόκλιση και διάμεσο για τα στοιχεία κάθε στήλης ενός πίνακα.




*Σχήμα 4.1: Σχηματική αναπαράσταση του κώδικα που δημιουργήθηκε*
*Σε κάθε τετράγωνο αντιστοιχεί μια υπορουτίνα- συνάρτηση της MATLAB*

Για τη σύγκριση των αποτελεσμάτων χρησιμοποιήσαμε πάλι διαγράμματα. Με τη χρήση της εντολής **errorbar**, που επιτρέπει τη σχεδίαση της μέσης τιμής και της τυπικής απόκλισης με συνδεόμενες γραμμές δημιουργήσαμε τα διαγράμματα που θα μας επέτρεπαν να συγκρίνουμε τους χρόνους και τα σχετικά σφάλματα. Στη συνέχεια προστέθηκαν στα διαγράμματα και οι διάμεσοι των αποτελεσμάτων με τη χρήση της εντολής **plot**. Κάθε φορά, στον άξονα των τετμημένων παρουσιάζεται η μεταβαλλόμενη παράμετρος: το πλήθος των εξισώσεων, το πλήθος των αγνώστων ή το σημείο αποκοπής, ανάλογα με τη δοκιμή που εκτελούνταν κάθε φορά.

Δημιουργήθηκαν τρεις κώδικες σύμφωνα με τα προηγούμενα, ένας που να αντιστοιχεί σε κάθε παράμετρο: να παρέχει σε αυτή κάποιες προεπιλεγμένες τιμές, να διατηρεί τις άλλες δύο σταθερές και να επιλύει το σύστημα υπολογίζοντας και σχεδιάζοντας ό,τι προαναφέρθηκε. Καθένας από τους τρεις κώδικες έπρεπε να εκτελεστεί δύο φορές: μια υπολογίζοντας τη λύση ελαχίστων τετραγώνων και μια για τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου. Έτσι τελικά έγιναν έξι δοκιμές, καθεμιά από τις οποίες συσχετίστηκε με μια οικογένεια μεμονωμένων δοκιμών, καθώς για όλους τους επιλεγμένους συνδυασμούς των παραμέτρων πραγματοποιήθηκε και μια μεμονωμένη δοκιμή.

## 3.2 Εφαρμογή κώδικα σε τυχαίους πίνακες και αξιολόγηση των αποτελεσμάτων

Η εφαρμογή του κώδικα σε τυχαίους πίνακες έπρεπε να προσομοιάζει στο μεγαλύτερο δυνατό βαθμό τα πραγματικά τομογραφικά προβλήματα. Τα στοιχεία του πίνακα **G** ωστόσο έπρεπε να είναι τυχαίοι αριθμοί και γι' αυτό επιλέξαμε αριθμούς από την κανονική κατανομή N(0,1), το οποίο επιτυγχάνεται με την εντολή **randn**.

Δεδομένου ότι στα πραγματικά τομογραφικά προβλήματα εμφανίζονται σποραδικοί πίνακες, δεν χρησιμοποιήσαμε απευθείας τους τυχαίους πίνακες για τις δοκιμές αλλά τους μετατρέψαμε τεχνητά σε σποραδικούς. Αυτό επιτεύχθηκε μηδενίζοντας όλα τα στοιχεία του πίνακα **G** τα οποία ήταν μικρότερα κατά απόλυτη τιμή από κάποιον αριθμό, το σημείο αποκοπής ή **cut**. Στο τελευταίο δεν δώσαμε μια μοναδική τιμή σε όλες τις δοκιμές αλλά έπαιρνε διαφορετικές, ανάλογα με τις ανάγκες κάθε δοκιμής. Στον κώδικα προστέθηκε ένας έλεγχος για κάθε στοιχείο του πίνακα αν η τιμή του ήταν απόλυτα μικρότερη από το επιλεγμένο **cut**. Αν ίσχυε το προηγούμενο, η τιμή του στοιχείου αυτού αντικαθιστόταν από την τιμή 0.

Το διάνυσμα **d** των δεδομένων κατασκευάστηκε ως εξής: αρχικά κατασκευάστηκε το διάνυσμα της πραγματικής λύσης **x**, έτσι ώστε $\mathbf{x}_i$ = sin(2πi/m), όπου m είναι το πλήθος των αγνώστων – παραμέτων του μοντέλου. Για τη λύση αυτή το διάνυσμα των δεδομένων θα έπρεπε να είναι **d** = **Gx**, αν υπήρχε απόλυτη ακρίβεια στις μετρήσεις και τα δεδομένα δεν είχαν θόρυβο. Για να προσομοιάσουμε πραγματικά προβλήματα, "μολύναμε" τα δεδομένα με τυχαίο θόρυβο, ο οποίος πολλαπλασιάζεται με έναν τυχαίο αριθμό, οδηγούμενα στη σχέση:

$$\boldsymbol{d} = \boldsymbol{Gx} + \theta \acute{o} \rho \upsilon \beta o \varsigma \times \boldsymbol{randn}(n, 1) \qquad (3.12)$$

Για τη λύση ελαχίστων τετραγώνων θέσαμε την παράμετρο του θορύβου ίση με 0.5 ενώ για τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου την θέσαμε ίση με 0.02.

Πραγματοποιήθηκαν έξι οικογένειες δοκιμών, σε καθεμιά από τις οποίες μεταβαλλόταν μια από τις παραμέτρους του προβλήματος, δηλαδή ο αριθμός εξισώσεων, ο αριθμός αγνώστων και το σημείο αποκοπής ενώ οι άλλες δύο παρέμεναν σταθερές. Θα εξετάσουμε κάθε περίπτωση ξεχωριστά.

### I. Μεταβολή σημείου αποκοπής

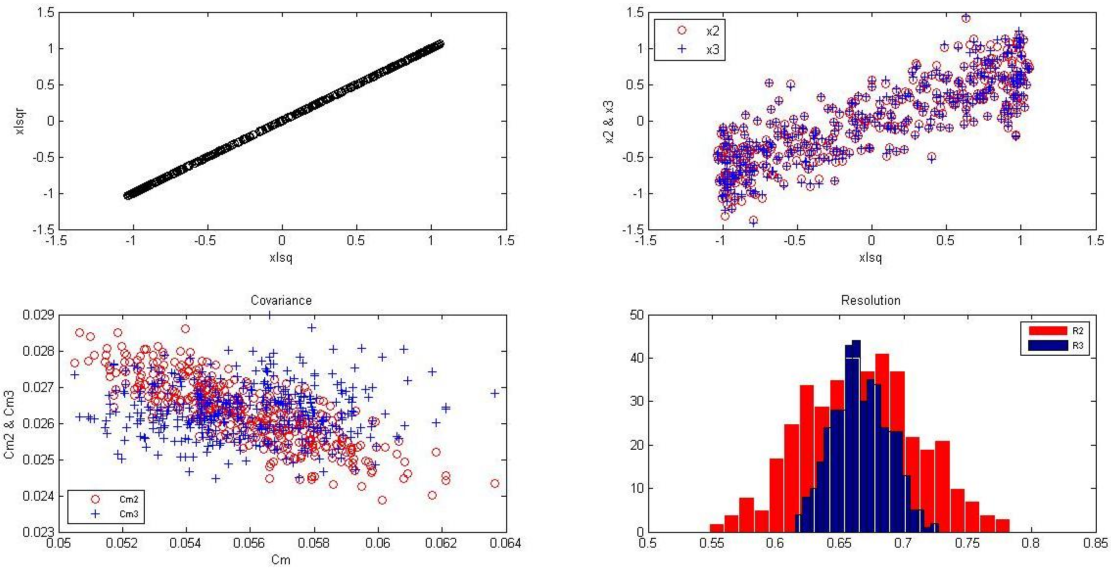
Στην πρώτη περίπτωση θεωρήσαμε τυχαίους πίνακες 800×400 κι αφήσαμε το σημείο αποκοπής να πάρει τις εξής τιμές:

**0 0.1 0.3 0.5 0.8 1 1.2**

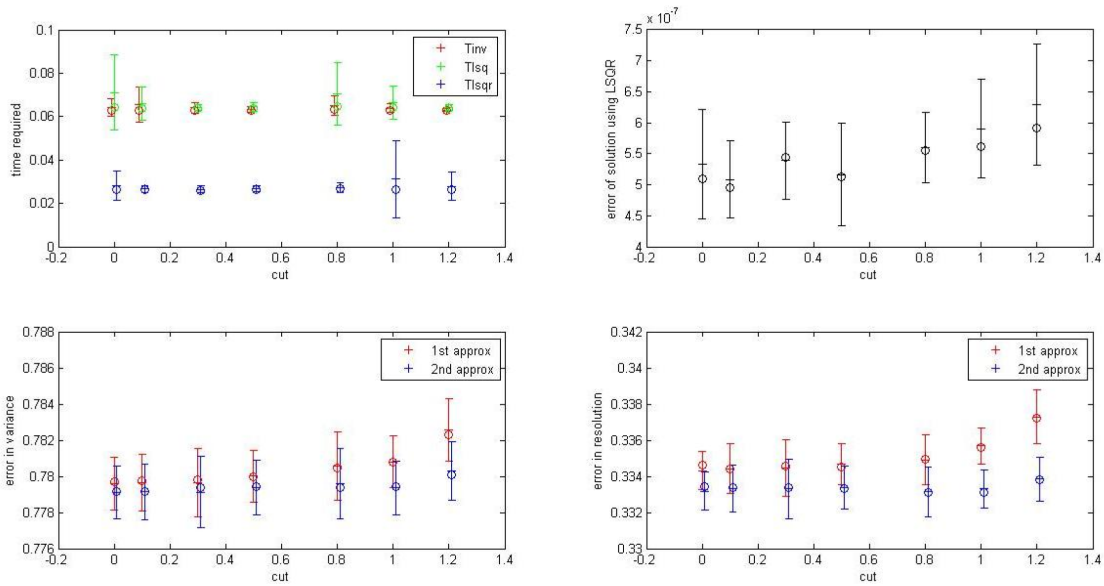
Για την τιμή 0 δεν μηδενίζεται κανένα στοιχείο του πίνακα ενώ για την τιμή 1 το 68% των στοιχείων του πίνακα είναι ίσα με μηδέν, καθώς τα στοιχεία αυτά είναι αριθμοί που προέρχονται από την κανονική κατανομή N(0,1). Για κάθε δοκιμή, δημιουργούνταν διαφορετικός τυχαίος πίνακας με τη χρήση της συνάρτησης **randn**, και δεν χρησιμοποιούνταν ο ίδιος για όλες τις δοκιμές.

Δοκιμάσαμε να χρησιμοποιήσουμε και μεγαλύτερες τιμές για το σημείο αποκοπής, όπως 1.6 και 2, αλλά οι πίνακες που προέκυπταν είχαν συντριπτικά περισσότερα στοιχεία τους ίσα με μηδέν, με αποτέλεσμα οι προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας να είναι εξαιρετικά ασταθείς. Έτσι, περιοριστήκαμε σε πιο χαμηλές τιμές του σημείου αποκοπής.

Οι δοκιμές έδειξαν ότι για τα δεδομένα του προβλήματος, δηλαδή για ένα γραμμικό σύστημα 800×400, το πλήθος των μηδενικών στοιχείων του πίνακα δεν επηρεάζει ουσιαστικά την ποιότητα της προσέγγισης των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας. Πράγματι, για τις διάφορες τιμές του **cut** πήραμε παρόμοια αποτελέσματα (ενδεικτικά βλ. Σχ.3.2).




*Σχήμα 3.2: Σύστημα 800×400 με **cut**=1, λύση ελαχίστων τετραγώνων.*




*Σχήμα 3.3: Σύγκριση των μέτρων ποιότητας, συστήματα 800×400,*
*λύση ελαχίστων τετραγώνων.*
*Με – σημειώνεται η μέση τιμή και η τυπική απόκλιση, με o σημειώνεται η διάμεσος.*

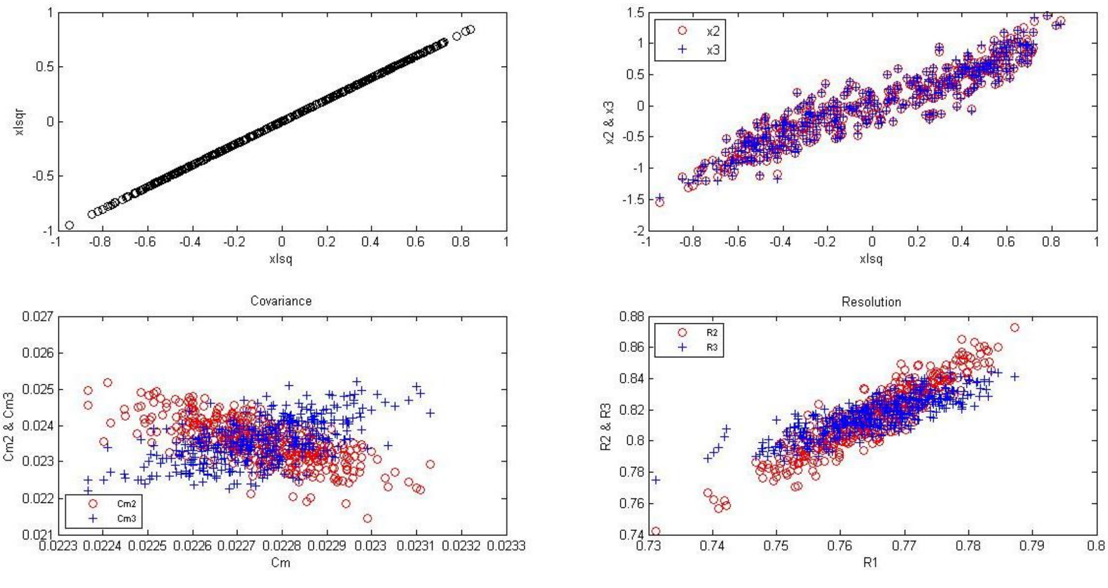
Σε πρώτη φάση, επιβεβαιώνεται η ακρίβεια της μεθόδου LSQR ενώ αντίθετα, οι προσεγγιστικες λύσεις **x2** και **x3** απέχουν πολύ από την πραγματική λύση, υπενθυμίζουμε όμως ότι αυτές δεν χρησιμοποιούνται ποτέ στην πράξη. Όσον αφορά τους πίνακες συμμεταβλητότητας, οι προσεγγίσεις δεν είναι ικανοποιητικές, καθώς διαφέρουν πολύ από τις πραγματικές τιμές των συνδιασπορών. Τα ίδια ισχύουν και για τους πίνακες διακριτικής ικανότητας, αφού οι προσεγγιστικές μορφές παίρνουν τιμές γύρω από το 0.65 για όλες τις τιμές του **cut**, δηλαδή απέχουν πολύ από τη μονάδα.

Επιπλέον, ο χρόνος επίλυσης του συστήματος δεν μεταβάλλεται ουσιαστικά με την αλλαγή του σημείου αποκοπής (βλ. Σχ.3.3). Ειδικότερα, οι χρόνοι των **xlsq** και **xinv** ταυτίζονται, υποδεικνύοντας ότι για μικρά γραμμικά συστήματα, ο υπολογισμός του αντιστρόφου δεν αυξάνει καθόλου το συνολικό υπολογιστικό χρόνο, ενώ παράλληλα η **xlsqr** εξοικονομεί περίπου τα 2/3 του χρόνου για την ακρίβεια που επιλέχθηκε στη σύγκλισή της (της τάξης του $10^{-7}$).

Τα σχετικά σφάλματα των διαγώνιων στοιχείων των προσεγγιστικών πινάκων συμμεταβλητότητας και διακριτικής ικανότητας παραμένουν επίσης αμετάβλητα από το πλήθος των σημείων αποκοπής. Το σφάλμα είναι αρκετά μεγάλο: 78% για τις διασπορές και 33.4% για τον πίνακα διακριτικής ικανότητας, γεγονός που δείχνει μία χαμηλή ποιότητα των προσεγγίσεων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας.

Στη συνέχεια εφαρμόσαμε τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου σε πίνακες 800×400 για τις ίδιες τιμές του σημείου αποκοπής. Η λύση που δίνει η LSQR είναι και πάλι ακριβής, ενώ οι λύσεις **x2** και **x3** έχουν πάλι μεγάλα σφάλματα.

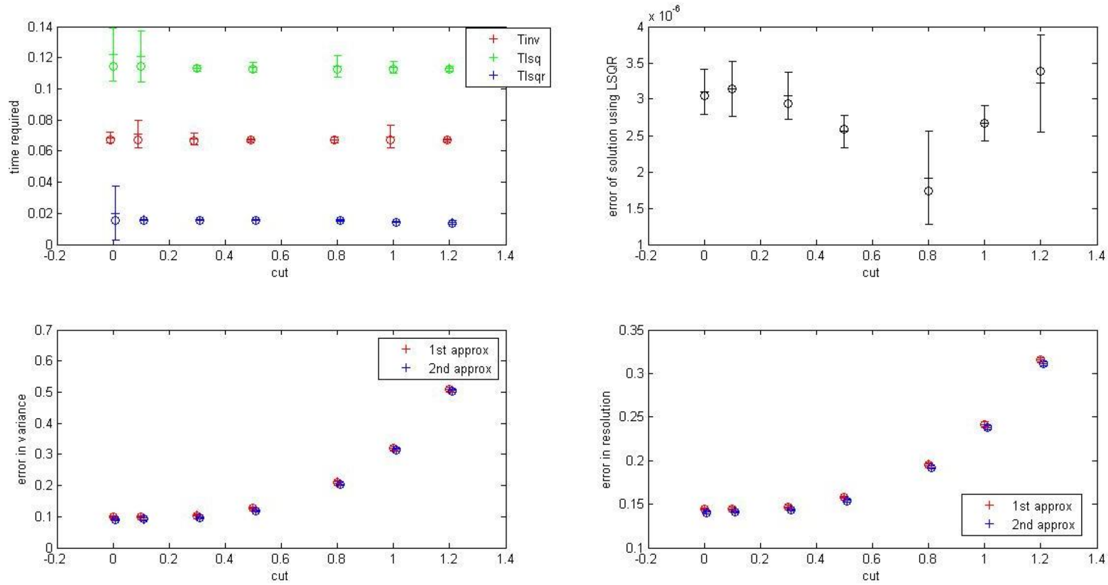
Οι προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας είναι πιο ακριβείς απ' ότι στην απλή λύση ελαχίστων τετραγώνων. Μάλιστα, η ακρίβεια είναι μεγαλύτερη όταν ο πίνακας δεν έχει μηδενικά στοιχεία (βλ. Σχ.3.4) και μειώνεται καθώς αυξάνεται το πλήθος αυτών. Το φαινόμενο αυτό, η αύξηση της διασποράς, μπορεί να ερμηνευθεί από την απώλεια πληροφοριών που επιφέρει ο τεχνητός μηδενισμός των μικρότερων στοιχείων του πίνακα.




*Σχήμα 3.4: Σύστημα 800×400 με **cut**=0, λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*

Για τον πίνακα διακριτικής ικανότητας, οι προσεγγίσεις του είναι κι αυτές πιο κοντά στις πραγματικές τιμές των διαγώνιων στοιχείων σε σχέση με την ακριβή λύση ελαχίστων τετραγώνων. Επίσης, οι προσεγγίσεις είναι καλύτερες όσο λιγότερα είναι τα μηδενικά στοιχεία του πίνακα. Η δεύτερη προσεγγιστική μορφή αποδεικνύεται λίγο πιο ακριβής από την πρώτη, τόσο για τον πίνακα συμμεταβλητότητας, όσο και για τον πίνακα διακριτικής ικανότητας.

Το σχετικό σφάλμα των προσεγγιστικών μορφών των πινάκων αυξάνεται όσο αυξάνονται τα μηδενικά στοιχεία του πίνακα, επιβεβαιώνοντας τα προηγούμενα συμπεράσματα. Πρόσθετα, τα σχετικά σφάλματα παρουσιάζουν πολύ μικρότερες διακυμάνσεις.




*Σχήμα 3.5: Σύγκριση των μέτρων ποιότητας, συστήματα 800×400, λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*

Οι υπολογιστικοί χρόνοι μένουν σταθεροί με την αύξηση του **cut**. Η LSQR απαιτεί εμφανώς λιγότερο χρόνο από τις άλλες δύο λύσεις. Η **xlsqr** και η **xinv** απαιτούν περίπου τον ίδιο χρόνο όσο και στην απλή λύση ελαχίστων τετραγώνων, ωστόσο η λύση **xlsq** απαιτεί περισσότερο χρόνο. Είναι εμφανές ότι για μικρά γραμμικά συστήματα, όπως το υπό μελέτη, η αντιστροφή του πίνακα $\mathbf{(G^T G + \varepsilon^2 I)}$ δεν μας επιβαρύνει σημαντικά ως προς τον υπολογιστικό χρόνο.

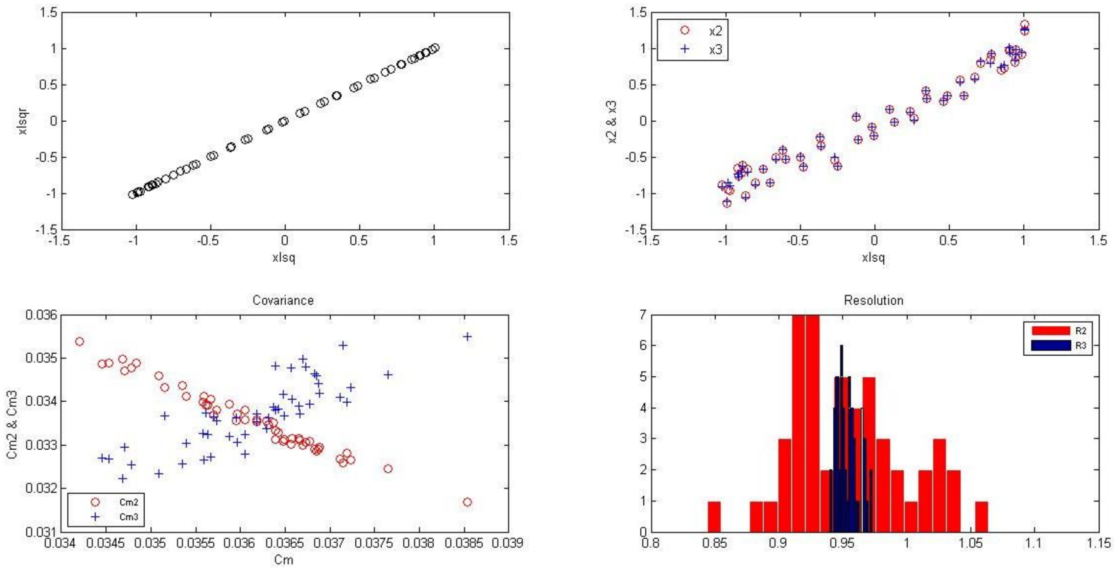
## II. Μεταβολή αριθμού αγνώστων (παραμέτρων)

Στη δεύτερη περίπτωση επιλύσαμε γραμμικά συστήματα με 1000 εξισώσεις και με το πλήθος των αγνώστων να παίρνει τις τιμές:

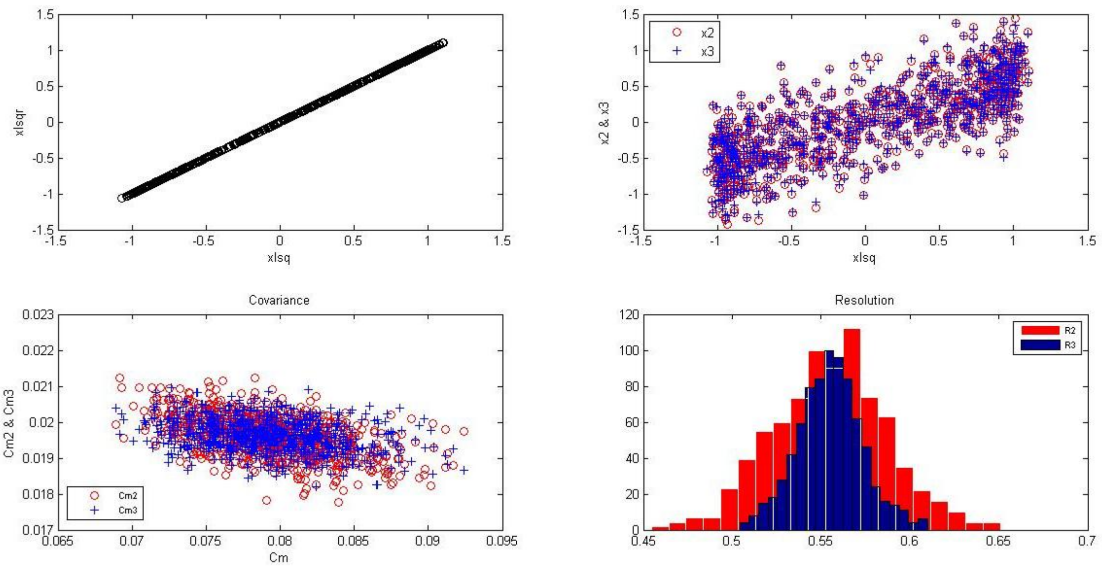
**50 100 200 400 800**

Το σημείο αποκοπής για όλες τις δοκιμές ήταν σταθερό κι ίσο με 1, έτσι ώστε οι πίνακες να έχουν πολλά μηδενικά στοιχεία, προσομοιάζοντας πραγματικά τομογραφικά προβλήματα, αλλά όχι υπερβολικά πολλά ώστε να μειωθεί η βαθμίδα του πίνακα και να μην μπορεί να εφαρμοστεί η λύση ελαχίστων τετραγώνων.

Η αύξηση του αριθμού των αγνώστων- παραμέτρων του μοντέλου- έχει ως αποτέλεσμα οι προσεγγιστικές μορφές να χάνουν σε ακρίβεια. Για συστήματα με πολλές εξισώσεις και λίγους αγνώστους (βλ. Σχ.3.6), οι προσεγγίσεις των διαγώνιων στοιχείων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας είναι αρκετά κοντά στις πραγματικές τιμές των στοιχείων αυτών. Ειδικότερα, η δεύτερη προσεγγιστική μορφή είναι λίγο ακριβέστερη και μάλιστα μειώνει σχεδόν ομοιόμορφα τις τιμές των διασπορών, ενώ η πρώτη μειώνει τις μεγάλες τιμές και αυξάνει τις μικρές. Επιπλέον, οι τιμές της δεύτερης προσέγγισης του πίνακα διακριτικής ικανότητας είναι πιο συγκεντρωμένες, ενώ της πρώτης πιο διασκορπισμένες.




*Σχήμα 3.6: Σύστημα 1000×50, με **cut**=1, λύση ελαχίστων τετραγώνων.*

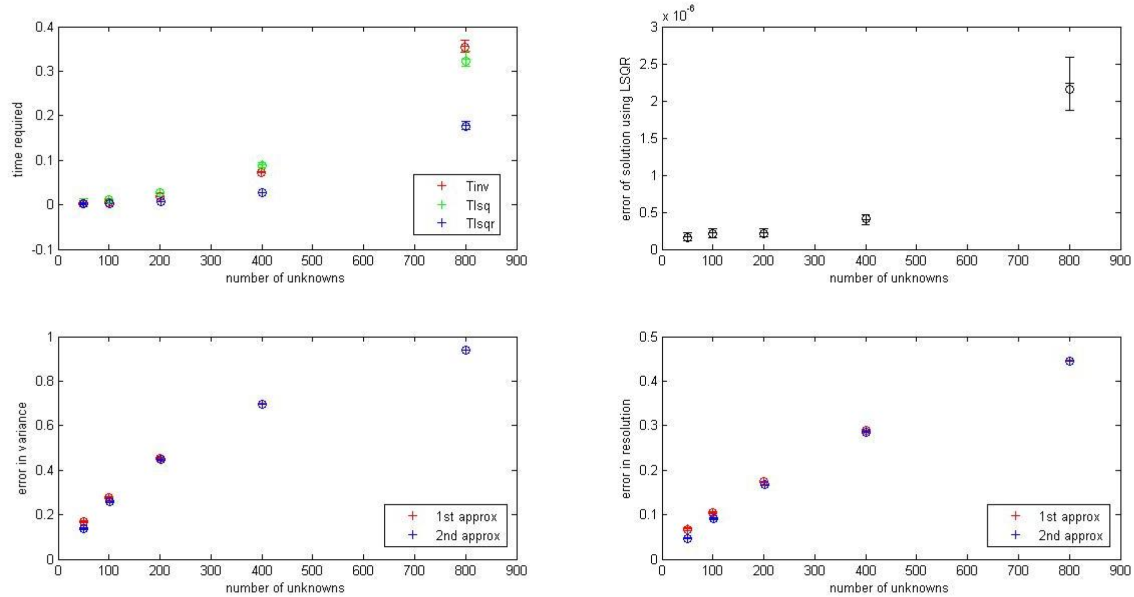



*Σχήμα 3.7: Σύστημα 1000×800, με **cut**=1, λύση ελαχίστων τετραγώνων.*

Αντίθετα, για μεγάλο αριθμό αγνώστων (βλ. Σχ.3.7) και συστήματα που πλησιάζουν προς τα τετραγωνικά, οι δύο προσεγγίσεις εμφανίζουν όμοια αποτελέσματα, τα οποία σε καμιά περίπτωση δεν είναι ικανοποιητικά. Οι προσεγγίσεις των διασπορών του μοντέλου είναι πολύ μικρότερες από τις πραγματικές, ενώ το ίδιο ισχύει και για τον πίνακα διακριτικής ικανότητας.

Τα σχετικά σφάλματα των προσεγγιστικών μορφών των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας επιβεβαιώνουν τα μέχρι τώρα συμπεράσματα, καθώς αυξάνονται ραγδαία όταν μεγαλώνει το πλήθος των αγνώστων και προσεγγίζει τον αριθμό των εξισωσεων (δεδομένων).

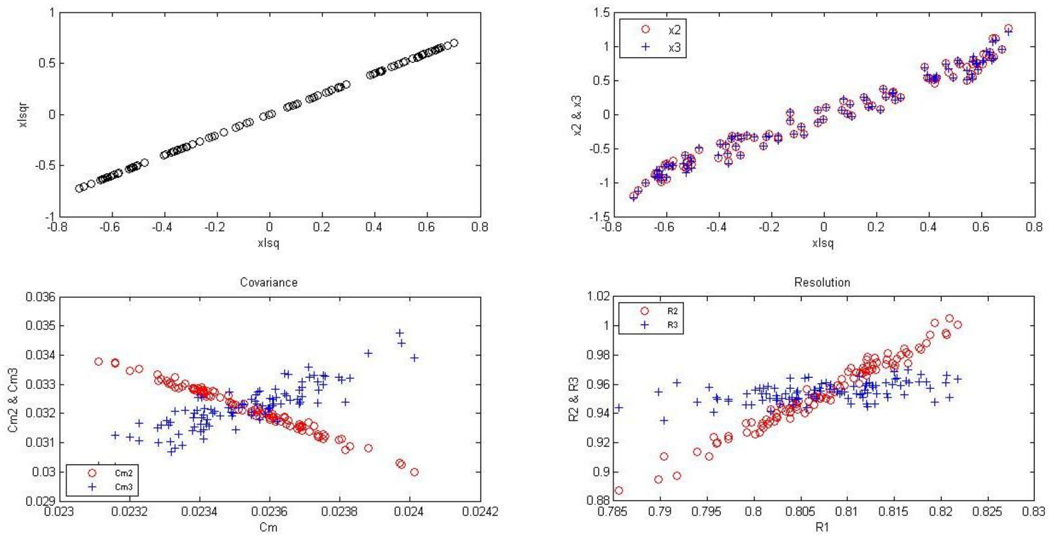
Όσον αφορά τον υπολογιστικό χρόνο, και για τις τρεις λύσεις αυξάνεται με την αύξηση του αριθμού των αγνώστων, κάτι που εξηγείται από το γεγονός ότι το υπολογιστικό έργο είναι σαφώς μεγαλύτερο. Μάλιστα, όταν το πλήθος των εξισώσεων είναι κατά πολύ μεγαλύτερο από αυτό των αγνώστων, οι τρεις μέθοδοι είναι σχεδόν ισοδύναμες. Αντίθετα, παρατηρούνται διαφοροποιήσεις όταν το σύστημα πλησιάζει προς το τετραγωνικό: η LSQR και πάλι αποδεικνύεται η ταχύτερη μέθοδος, ενώ για μεγάλα συστήματα όπως το 1000×800, η **xinv** απαιτεί λίγο περισσότερο χρόνο από την **xlsq**.



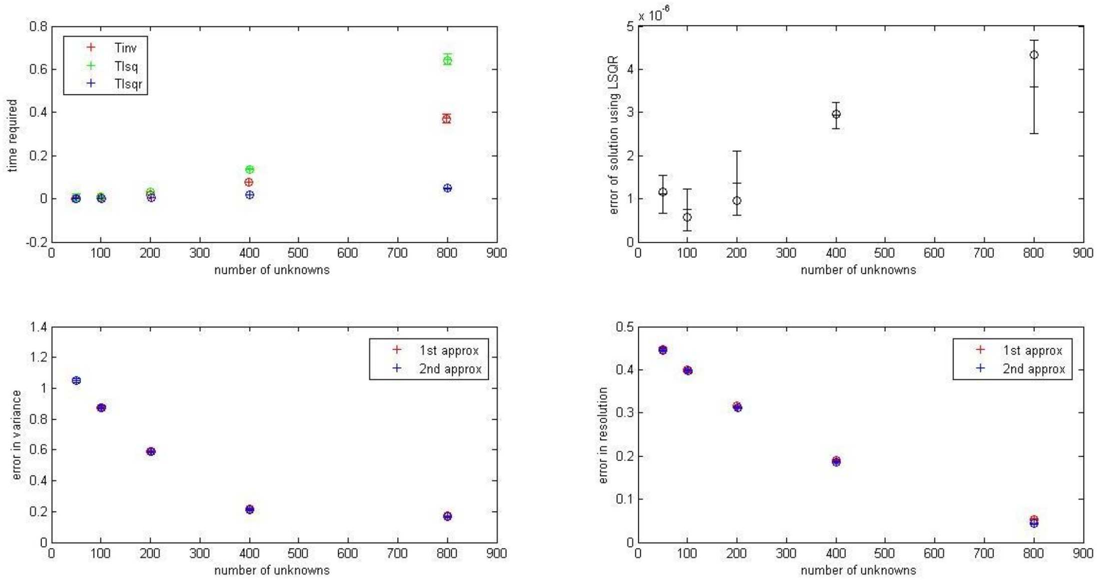

*Σχήμα 3.8: Σύγκριση των μέτρων ποιότητας, 1000 εξισώσεις,* ***cut****=1,*
*λύση ελαχίστων τετραγώνων.*

Στη συνέχεια, εφαρμόσαμε τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου σε γραμμικά συστήματα όμοιων διαστάσεων με την προηγούμενη περίπτωση, δηλαδή με 1000 εξισώσεις, σημείο αποκοπής ίσο με 1 και το πλήθος αγνώστων να παίρνει τις ίδιες τιμές με πριν.

Εδώ η αύξηση του αριθμού των αγνώστων λειτουργεί αντίθετα απ' ότι στην απλή λύση ελαχίστων τετραγώνων, δηλαδή βελτιώνει την ποιότητα της προσέγγισης των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας. Έτσι, για συστήματα με λίγους αγνώστους (π.χ. 1000×100, βλ. Σχ.3.9), οι προσεγγιστικές μορφές των πινάκων είναι πολύ ανακριβείς, καθώς οι εκτιμήσεις είναι κατά πολύ μεγαλύτερες από τις πραγματικές τιμές. Σε μεγαλύτερα γραμμικά συστήματα, οι προσεγγίσεις γίνονται καλύτερες, πλησιάζοντας τις πραγματικές τιμές των διαγώνιων στοιχείων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας. Σημειωτέον ότι σε όλα τα συστήματα, το φάσμα των τιμών των προσεγγίσεων είναι σημαντικά μεγαλύτερο από το φάσμα των πραγματικών τιμών.




*Σχήμα 3.9: Σύστημα 1000×100, με* ***cut****=1, λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*




*Σχήμα 3.10: Σύγκριση των μέτρων ποιότητας, 1000 εξισώσεις,* ***cut****=1,*
*λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*

Οι υπολογιστικοί χρόνοι των τριών μεθόδων επίλυσης του συστήματος, ακολουθούν παρόμοιο μοτίβο με την προηγούμενη οικογένεια δοκιμών. Η LSQR δεν αυξάνει σημαντικά τις απαιτήσεις της καθώς μεγαλώνει ο αριθμός των αγνώστων, γεγονός που οφείλεται στο ότι οι επιπλέον περιορισμοί (damping) εξαλείφουν ενδεχόμενη γραμμική εξάρτηση μεταξύ των γραμμών του πίνακα, η οποία καθυστερεί την LSQR. Αξιοσημείωτο είναι ότι η **xinv**, που περιλαμβάνει την εύρεση του αντιστρόφου, είναι οικονομικότερη από την **xlsq**, γεγονός που συμβαίνει μόνο στα προβλήματα με τις λύσεις ελαχίστων τετραγώνων ελαχίστου μέτρου, ενώ στις απλές λύσεις ελαχίστων τετραγώνων η **xlsq** είναι κατά πολύ λίγο ταχύτερη της **xinv**. Η εντολή **G\d** πιθανώς να καθυστερεί να λύσει το επαυξημένο σύστημα, εξαιτίας του μεγάλου βάρους (20) των περιορισμών μέτρου, που οδηγεί τις γραμμές του πίνακα στο να έχουν μέτρα διαφορετικής τάξης.

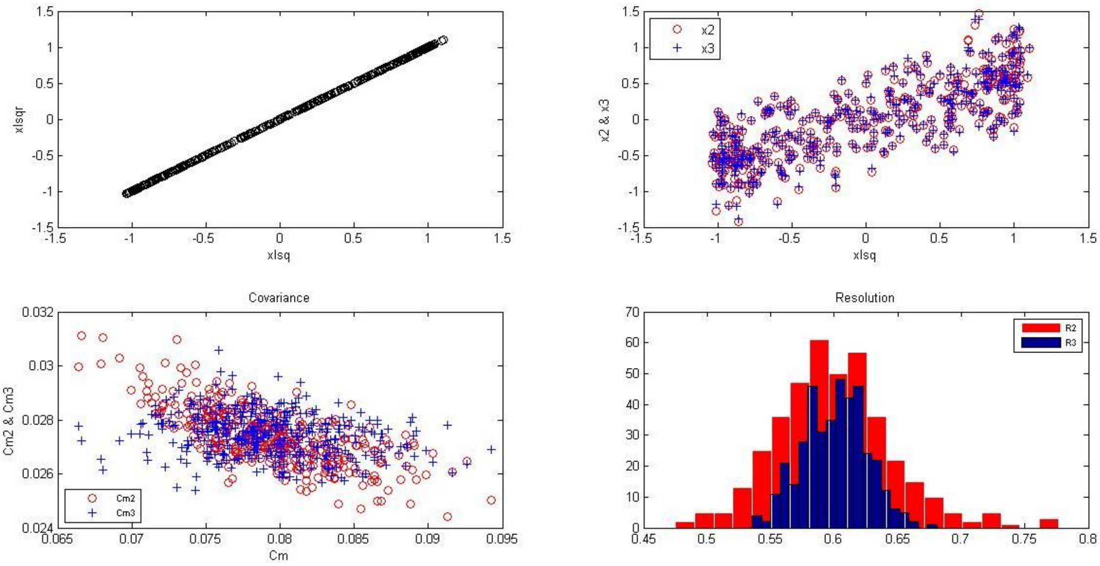
Τα σχετικά σφάλματα των διαγωνίων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας μειώνονται καθώς το σύστημα μεγαλώνει. Για συστήματα με λίγους αγνώστους συγκριτικά με τις εξισώσεις το σχετικό σφάλμα του διανύσματος των διασπορών πλησιάζει, ή και ξεπερνάει, το 100%, καθιστώντας την εκτίμηση τελείως αναξιόπιστη. Αντίθετα, σε συστήματα που πλησιάζουν τα τετραγωνικά οι δύο προσεγγίσεις είναι αρκετά ικανοποιητικές- η δεύτερη μάλιστα είναι λίγο καλύτερη- ειδικά όσον αφορά τον πίνακα διακριτικής ικανότητας, όπου το σφάλμα είναι πολύ μικρό.

## III. Μεταβολή αριθμού εξισώσεων

Στην τρίτη περίπτωση, εξετάσαμε το πώς συμπεριφέρονται οι προσεγγίσεις των ποιοτικών μέτρων της λύσης ελαχίστων τετραγώνων όταν αυξάνεται το πλήθος των εξισώσεων- δεδομένων. Έτσι, επιλύσαμε με τη μέθοδο ελαχίστων τετραγώνων τυχαία συστήματα με 400 παραμέτρους και με αριθμό δεδομένων διαδοχικά:

**600 800 1000 1200 1400**

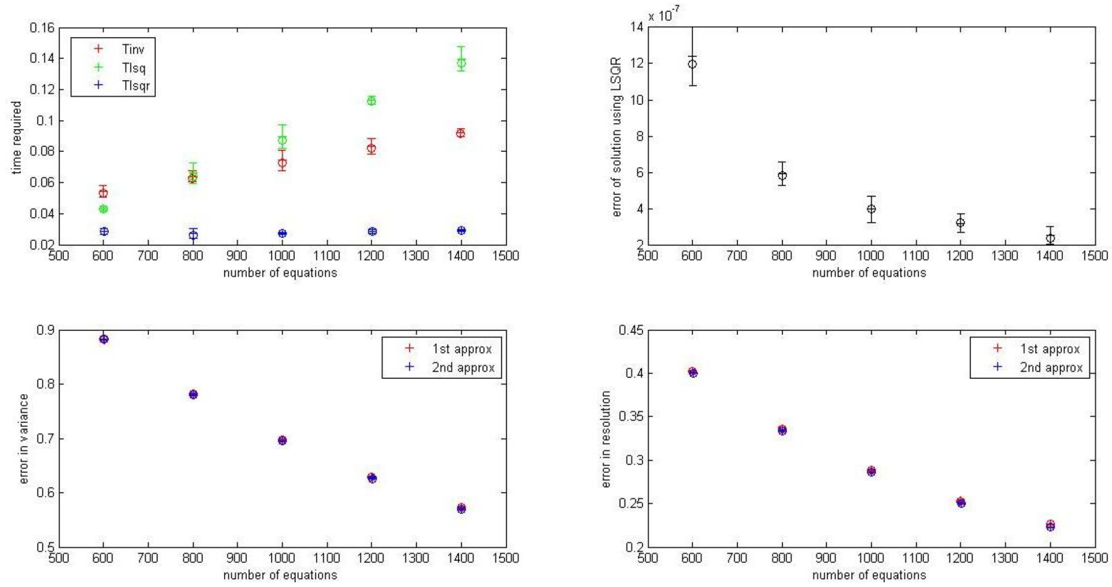
Το σημείο αποκοπής ήταν και πάλι σταθερό και ίσο με 1, έτσι ώστε να προσομοιάζει πραγματικά τομογραφικά προβλήματα. Όσο αυξανόταν το πλήθος των εξισώσεων βελτιωνόταν η ποιότητα της προσέγγισης των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας, και για τις δύο μορφές. Στο σύστημα 600×400, οι προσεγγίσεις ήταν πολύ απομακρυσμένες από τις πραγματικές τιμές (βλ. Σχ.3.11).




*Σχήμα 3.11: Σύστημα 600×400, με **cut**=1, λύση ελαχίστων τετραγώνων.*

Είναι προφανές ότι σε γραμμικά συστήματα που πλησιάζουν προς τα τετραγωνικά οι προσεγγιστικές μορφές δεν είναι ιδιαίτερα αξιόπιστες, καθώς οι τιμές τους είναι πολύ μικρότερες από τις πραγματικές. Αντίθετα, σε συστήματα με περισσότερες εξισώσεις, όπως το 1400×400, οι τιμές των διασπορών είναι πιο κοντά στις πραγματικές, ενώ και οι προσεγγίσεις του πίνακα διακριτικής ικανότητας είναι μεγαλύτερες, μεταξύ 0.75 και 0.8. Ακόμη, το φάσμα των προσεγγιστικών τιμών του πίνακα συμμεταβλητότητας είναι μικρότερο από το πραγματικό.

Τα σχετικά σφάλματα των διαγωνίων των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας (βλ. Σχ.3.12) μειώνονται με την αύξηση των εξισώσεων, επιβεβαιώνοντας ότι οι προσεγγιστικές μορφές γίνονται πιο ακριβείς για συστήματα όπου οι εξισώσεις είναι πολλαπλάσιες των αγνώστων. Ακόμα όμως και για το μεγαλύτερο σύστημα, τα σχετικά σφάλματα είναι πολύ μεγάλα: 57% για το διάνυσμα των διασπορών και 22% για τη διαγώνιο του πίνακα διακριτικής ικανότητας.

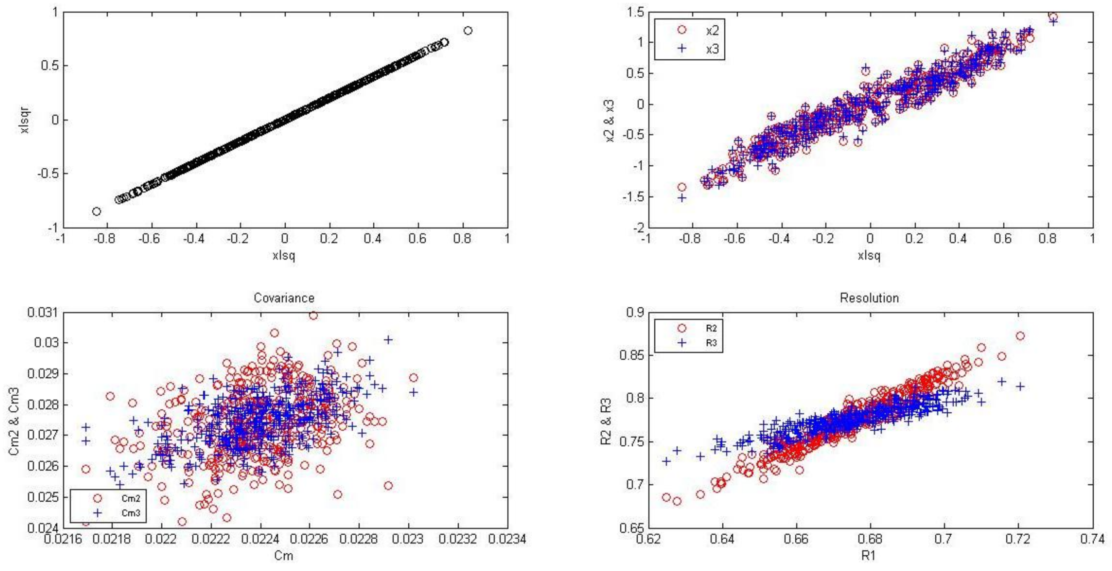



*Σχήμα 3.12: Σύγκριση των μέτρων ποιότητας, 400 άγνωστοι, **cut**=1, λύση ελαχίστων τετραγώνων.*

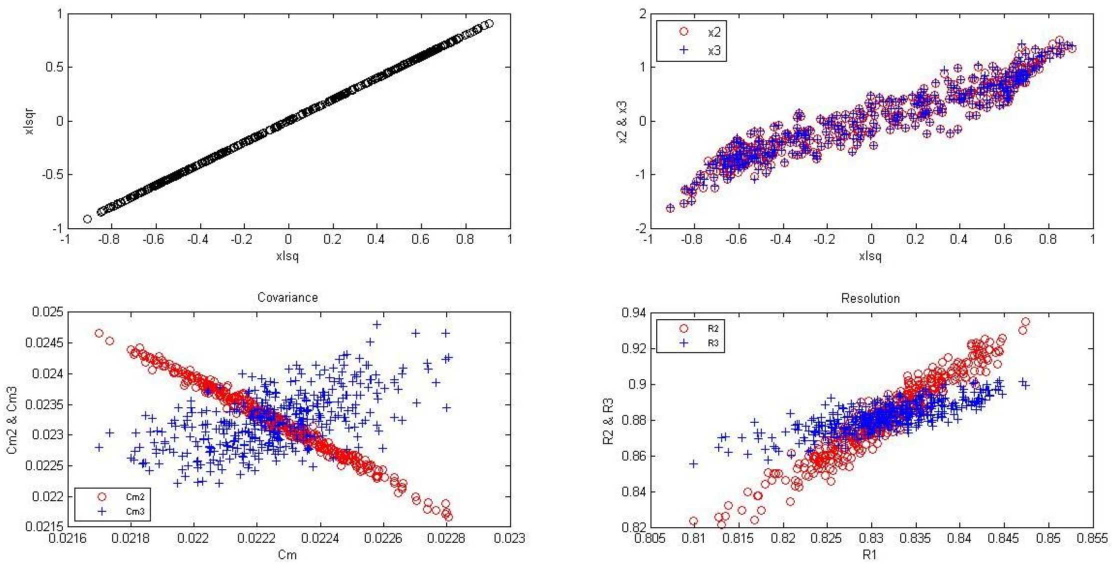
Οι υπολογιστικοί χρόνοι αυξάνονται καθώς μεγαλώνει το σύστημα, γεγονός αναμενόμενο καθώς πληθαίνουν τα στοιχεία του πίνακα, αλλά αυτή η αύξηση δεν είναι μεγάλη (στην τρίτη οικογένεια δοκιμών είχαμε πολύ μεγαλύτερη αύξηση του υπολογιστικού χρόνου). Αυτό οφείλεται στο ότι π.χ. η **xinv** απαιτεί την αντιστροφή ενός πίνακα 400×400 άσχετα με τον αριθμό των εξισώσεων. Επιπλέον, ο χρόνος εδώ φαίνεται να είναι γραμμική συνάρτηση του πλήθους των εξισώσεων, ενώ στη δεύτερη οικογένεια στην απλή λύση ελαχίστων τετραγώνων ήταν μια καμπύλη που συνέδεε το χρόνο με το πλήθος των αγνώστων. Η **xlsqr** μένει σχεδόν ανεπηρέαστη από τις αλλαγές των διαστάσεων του προβλήματος και είναι πάντα η οικονομικότερη λύση.

Στην τελευταία οικογένεια δοκιμών εφαρμόσαμε τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου σε γραμμικά συστήματα ίδιων διαστάσεων με την προηγούμενη περίπτωση, με σκοπό να δούμε πώς μεταβάλλονται οι προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας στο πρόβλημα αυτό.

Γενικά, στις δοκιμές αυτές οι προσεγγίσεις ήταν πολύ πιο κοντά στις πραγματικές τιμές συγκριτικά με την προηγούμενη οικογένεια εφαρμογών. Επιπλέον, η αύξηση του πλήθους των εξισώσεων βελτίωνε την ποιότητα των προσεγγίσεων. Σε μικρά συστήματα όπως 600×400 οι δύο προσεγγιστικές μορφές είχαν όμοια αποτελέσματα: οι τιμές τους ήταν λίγο μεγαλύτερες από τις πραγματικές τόσο για τα σφάλματα, όσο και για τα στοιχεία το πίνακα διακριτικής ικανότητας (βλ. Σχ.3.13).




*Σχήμα 3.13: Σύστημα 600×400, με **cut**=1, λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*

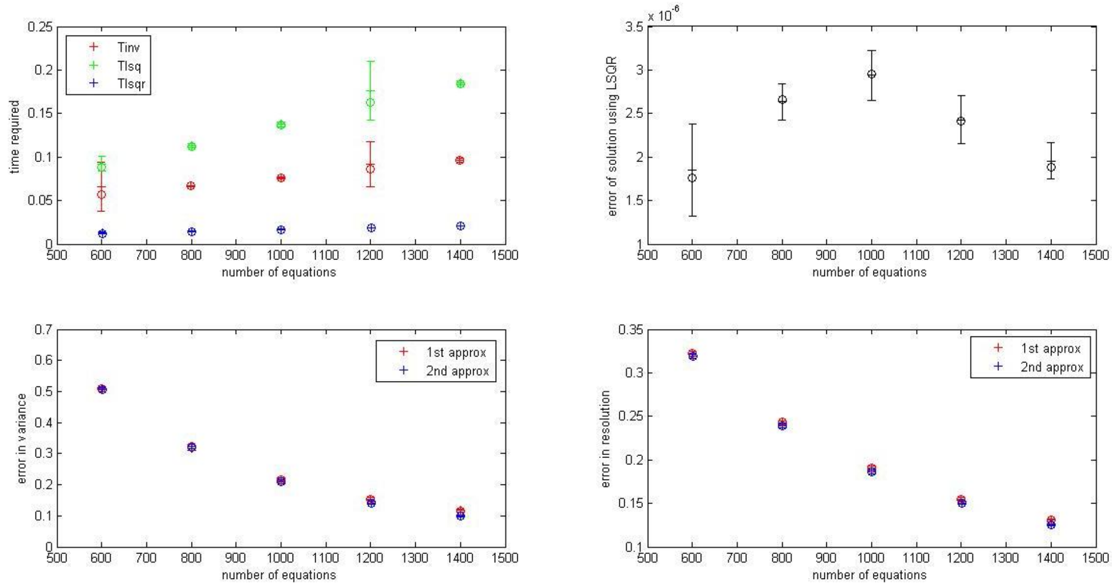



*Σχήμα 3.14: Σύστημα 1400×400, με **cut**=1, λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*

Σε μεγαλύτερα συστήματα η απόσταση των προσεγγιστικών τιμών από τις πραγματικές είναι μικρότερη και μάλιστα σε κάποιες υπάρχει σχεδόν ταύτιση (βλ. Σχ.3.14). Η δεύτερη προσεγγιστική μορφή αυξάνει σχεδόν ομοιόμορφα τις τιμές κι είναι λίγο πιο ακριβής από την πρώτη. Η πρώτη προσεγγιστική μορφή παράγει προσεγγίσεις οι οποίες βρίσκονται σε μια ευθεία σχεδόν κάθετη με τη διαγώνιο, κάτι που σημαίνει ότι παρουσιάζει αντίθετη τάση από τις πραγματικές τιμές.

Τα σχετικά σφάλματα των προσεγγιστικών μορφών είναι αρκετά μικρότερα σε σχέση με την πέμπτη οικογένεια δοκιμών. Επίσης, μειώνονται με την αύξηση του πλήθους των εξισώσεων. Είναι εμφανές ότι οι προσεγγιστικές μορφές είναι αξιόπιστες αν το πλήθος των εξισώσεων είναι σημαντικά μεγαλύτερο από αυτό των αγνώστων, αντίθετα για συστήματα που πλησιάζουν τα τετραγωνικά οι προσεγγίσεις δεν είναι ποιοτικά επαρκείς.

Οι υπολογιστικοί χρόνοι των τριών λύσεων αυξάνονται γραμμικά με τον αριθμό των εξισώσεων. Η LSQR δεν επηρεάζεται σημαντικά, ενώ οι άλλες δύο λύσεις αυξάνονται σημαντικά καθώς μεγαλώνει το σύστημα. Ωστόσο ο αριθμός των εξισώσεων επηρεάζει το χρόνο πολύ λιγότερο από ότι ο αριθμός των αγνώστων (βλ. δεύτερη οικογένεια δοκιμών). Η αντιστροφή του πίνακα που απαιτείται κατά τον υπολογισμό της **xinv** δεν είναι ιδιαίτερα επιβαρυντική, σε ότι αφορά τον υπολογιστικό χρόνο.




*Σχήμα 3.15: Σύγκριση των μέτρων ποιότητας, 400 άγνωστοι,* ***cut****=1,*
*λύση ελαχίστων τετραγώνων ελαχίστου μέτρου.*

## 3.3 Εφαρμογή σε πραγματικό πρόβλημα

Η εφαρμογή του κώδικα που αναπτύχθηκε σε κάποιο πραγματικό πρόβλημα ήταν απαραίτητη ώστε να ελεγθεί η συμπεριφορά των λύσεων και των προσεγγίσεων που μελετήθηκαν σε ένα πραγματικό πρόβλημα και να δούμε κατά πόσο τα θεωρητικά συμπεράσματα που εξήχθηκαν ανταποκρίνονται στην πραγματικότητα. Επιπλέον, θέλαμε να ελέγξουμε την αποτελεσματικότητα των τεχνικών που χρησιμοποιήθηκαν έχοντας ένα πραγματικό σημείο αναφοράς, δηλαδή ένα κατάλληλο μέτρο σύγκρισης. Επιλέγθηκε ένα γραμμικό πρόβλημα, έτσι ώστε να μην απαιτείται πολλαπλή εφαρμογή του κώδικα για την εύρεση της λύσης και να μην επηρεαστεί η ανάλθση από φαινόμενα μη γραμμικότητας. Το πρόβλημα που μελετήθηκε προέκυψε κατά της εκπόνηση της διδακτορικής διατριβής της υποψήφιας διδάκτορος Χρύσας Βεντούζη και στα πλαίσια της διατριβής είχε επιλυθεί με τη μέθοδο LSQR. Επομένως η ακρίβεια της λύσης θα μπορούσε να ελεγθεί άμεσα, σε σύγκριση με ένα συμβατικό κώδικα.

### 3.3.1 Περιγραφή του προβλήματος

Ο παράγοντας ποιότητας (Q-factor) χρησιμοποιείται για τη μελέτη της ανελαστικής απόσβεσης των σεισμικών κυμάτων, η οποία είναι από τους σημαντικότερους παράγοντες στη διάδοσή τους και παρέχει πολύτιμες πληροφορίες για τη δομή της Γης. Ο παράγοντας ποιότητας εκφράζει την απώλεια της ελαστικής ενέργειας ενός σεισμικού κύματος καθώς αυτό διαδίδεται μέσω ενός ανελαστικού μέσου. Ορίζεται ως:

$$\boldsymbol{Q = -\frac{2\pi E}{\Delta E}} \quad (3.13)$$

όπου E είναι η ενέργεια του σεισμικού κύματος και ΔΕ είναι το ποσό της ενέργεια που χάνεται σε μια περίοδο ή σε ένα μήκος κύματος. Εξαιτίας αυτής της απώλειας ενέργειας, το πλάτος ταλάντωσης των σεισμικών κυμάτων ελαττώνεται καθώς αυτά ταξιδεύουν στο εσωτερικό της Γης. Το πλάτος ταλάντωσης A ενός σεισμικού κύματος που διαδίδεται σε ένα ομογενές μέσο σε απόσταση δίνεται από τη σχέση:

$$\boldsymbol{A = \frac{A_0}{r} exp(-\frac{\pi f}{QV} r)} \quad (3.14)$$

όπου $A_0$ το πλάτος στην εστία του σεισμού, r η υποκεντρική απόσταση, f η συχνότητα και V η ταχύτητα του μέσου διάδοσης.

Ο τελεστής απόσβεσης ολόκληρης της διαδρομής t* ή αλλιώς χρόνος απόσβεσης ορίζεται ως:

$$\boldsymbol{t^* = \int_{path} \frac{dr}{QV}} \quad (3.15, \text{Kanamori, 1967})$$

και μπορεί να εκτιμηθεί τόσο για τα επιμήκη (P) όσο και για τα εγκάρσια (S) κύματα από την κλίση του φάσματος του εύρους επιτάχυνσης πάνω από τη γωνιακή συχνότητα $f_c$, εφαρμόζοντας το μετασχηματισμό Fourier (Cormier, 1982):

$$\boldsymbol{A(r,f) = A_0 exp(-\pi f t^*)} \quad (3.16)$$

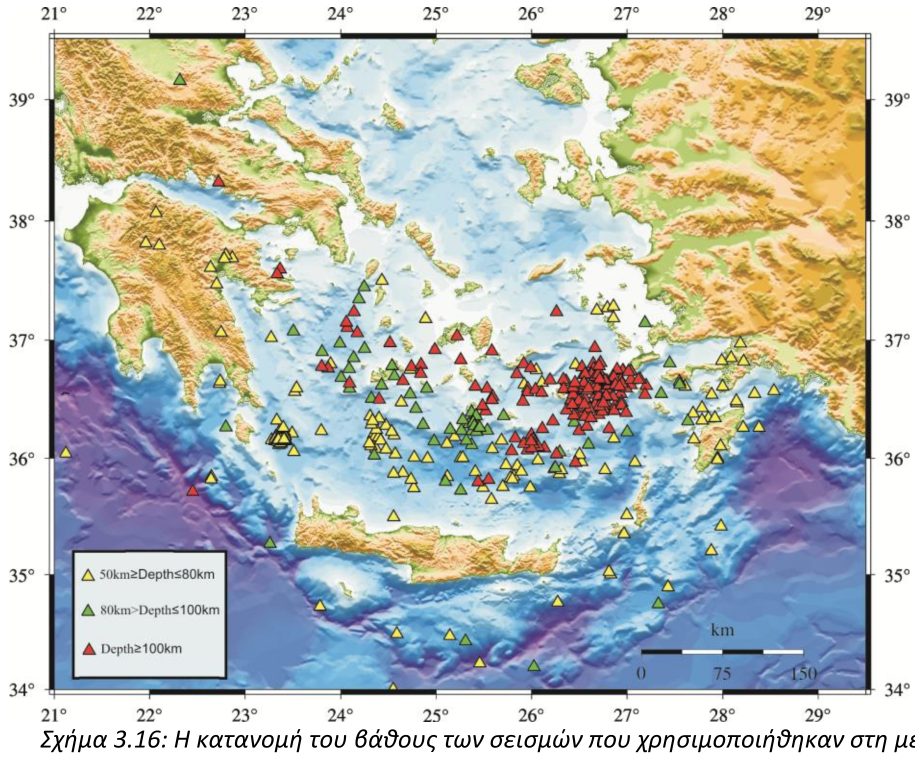
όπου $A_0$= $(2\pi f)^2 S(f)G(r,f)$ με G(r,f) να είναι ο γεωμετρικός παράγοντας διασποράς και S(f) το φάσμα μετατόπισης της πηγής, το οποίο είναι ανάλογο με τη γωνιακή συχνότητα, αν υποθέσουμε ότι η πηγή είναι τύπου Brune (Brune, 1970). Εναλλακτικά, οι χρόνοι t* μπορούν να υπολογιστούν από κατάλληλο αναλυτή. Στην προκείμενη μελέτη χρησιμοποιήθηκε ο αυτόματος τρόπος υπολογισμού. Αν θεωρήσουμε ότι ο παράγοντας ποιότητας παραμένει σταθερός σε όλο το μήκος διαδρομής της ακτίνας, η εξίσωση (3) γίνεται:

$$\boldsymbol{t^* = \frac{t}{Q}} \quad (3.17)$$

και αντικαθιστώντας στην (4) έχουμε:

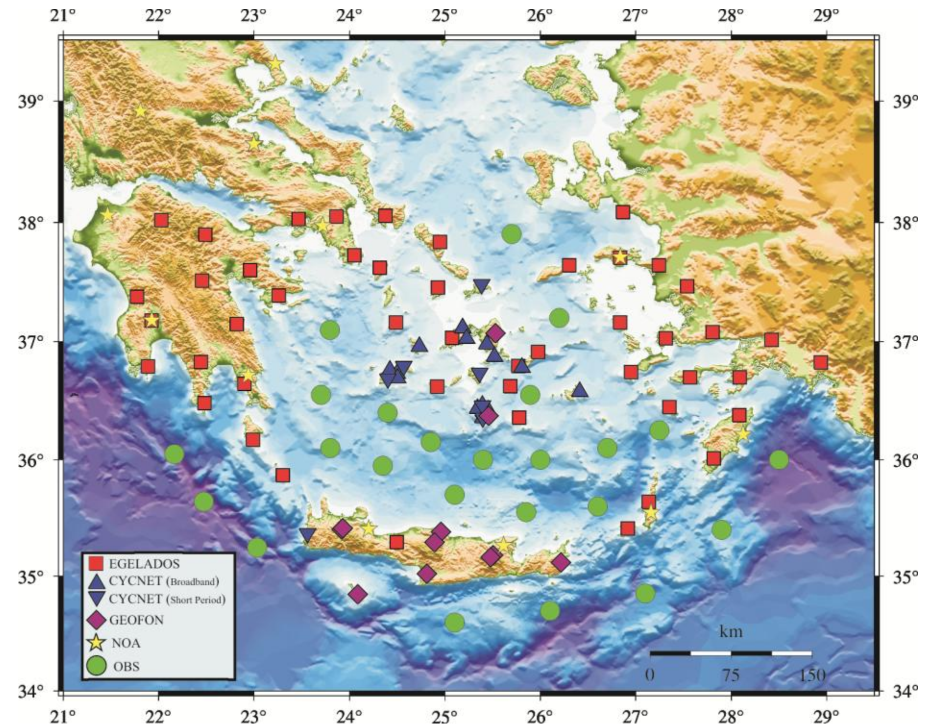
$$\boldsymbol{\ln[A(f)] = \ln A_0 - \frac{\pi f t}{Q}} \quad (3.18)$$

η οποία είναι μια γραμμική εξίσωση με κλίση πt/Q. Βρίσκοντας την κλίση, μπορεί να υπολογιστεί ο μέσος παράγοντας ποιότητας Q κατά μήκος της διαδρομής της ακτίνας, δεδομένου ότι αυτός παραμένει σταθερός σε συχνότητες μεγαλύτερες της $f_c$ για το φάσμα επιτάχυνσης και ο γεωμετρικός παράγοντας διασποράς G θεωρείται ότι μεταβάλλεται μόνο με την απόσταση.




*Σχήμα 3.16: Η κατανομή του βάθους των σεισμών που χρησιμοποιήθηκαν στη μελέτη*

Για τη μελέτη της δομής απόσβεσης στη ζώνη κατάδυσης του Νοτίου Αιγαίου χρησιμοποιήθηκαν δεδομένα από 400 περίπου σεισμούς (Σχ. 3.16) ενδιαμέσου βάθους που καταγράφηκαν από τοπικά και μόνιμα δίκτυα (Σχ. 3.17). Αρχικά ο τρισδιάστατος χώρος κάτω από το Νότιο Αιγαίο χωρίστηκε σε κελιά, σε καθένα από τα οποία θα υπολογιζόταν ξεχωριστά ο παράγοντας ποιότητας, θεωρώντας ότι είναι σταθερός σε όλο το χώρο του κελιού. Κάθε σεισμική ακτίνα επηρεάζεται και μας δίνει πληροφορίες (δεδομένα) για τα κελιά από τα οποία διέρχεται. Αντίθετα, το σύνολο των ακτινών μπορεί να μας δώσει ένα σύνολο πληροφοριών για όλα τα κελιά. Για κάθε σεισμική ακτίνα, δηλαδή για κάθε διαδρομή σεισμού- σταθμού καταγραφής, δημιουργείται έτσι μια εξίσωση, το σύνολο των οποίων μας οδηγεί στο γραμμικό σύστημα που πρέπει να επιλύσουμε για τον υπολογισμό της δομής απόσβεσης. Το τελικό σύστημα αποτελείται από 16750 εξισώσεις (μαζί με τους περιορισμούς) και 5280 αγνώστους.



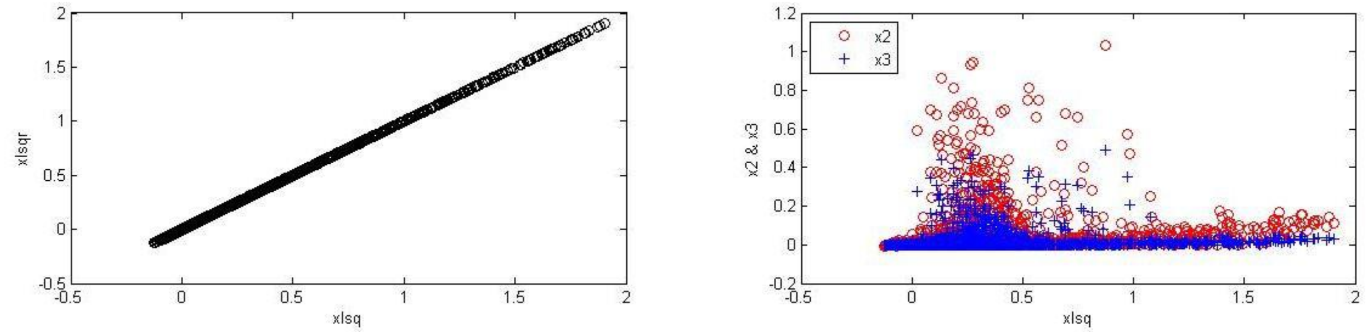

*Σχήμα 3.17: Η διανομή των σταθμών όλων των δικτύων που χρησιμοποιήθηκαν*

Ο πίνακας του συστήματος ήταν πολύ σποραδικός, λόγω της φύσης των δεδομένων. Πράγματι, δεδομένου ότι μια μεμονωμένη σεισμική ακτίνα διέρχεται από συγκεκριμένο αριθμό κελιών, μόνο τα στοιχεία του πίνακα που βρίσκονται στις αντίστοιχες στήλες είναι μη μηδενικά, για τη γραμμή που αντιστοιχεί στην επιλεγμένη ακτίνα.

## 3.3.2 Επίλυση του προβλήματος

Αρχικά έπρεπε να τροποποιήσουμε τον κώδικα ώστε να προσαρμόζεται στα νέα ζητούμενα. Ο πίνακας είχε δοθεί σε σποραδική μορφή, επομένως έπρεπε να μετατραπεί και να αποθηκευτεί σε κανονική, κάτι που πραγματοποιήθηκε στη MATLAB, και πιο συγκεκριμένα στο παράθυρο εργασίας χωρίς να γραφτεί κάποιος σχετικός κώδικας ώστε η διαδικασία να ελέγχεται συνεχώς. Πρόσθετα, ο κώδικας διαφοροποιήθηκε ώστε να διαβάζει τον πίνακα και το διάνυσμα των δεδομένων από τα αντίστοιχα αρχεία, αντί να παράγει τυχαίους πίνακες. Δεδομένου ότι ο πίνακας είχε ενσωματωμένους περιορισμούς, όχι μόνο για το μέτρο της λύσης (damping), αλλά και για την ομαλότητα (smoothing) αυτής, επιλέξαμε να επικεντρωθούμε στον πίνακα συμμεταβλητότητας, αγνοώντας αυτόν της διακριτικής ικανότητας. Γι’ αυτό το λόγο χρησιμοποιήθηκε ο κώδικας για την απλή λύση ελαχίστων τετραγώνων, με τις αλλαγές που προαναφέρθηκαν.

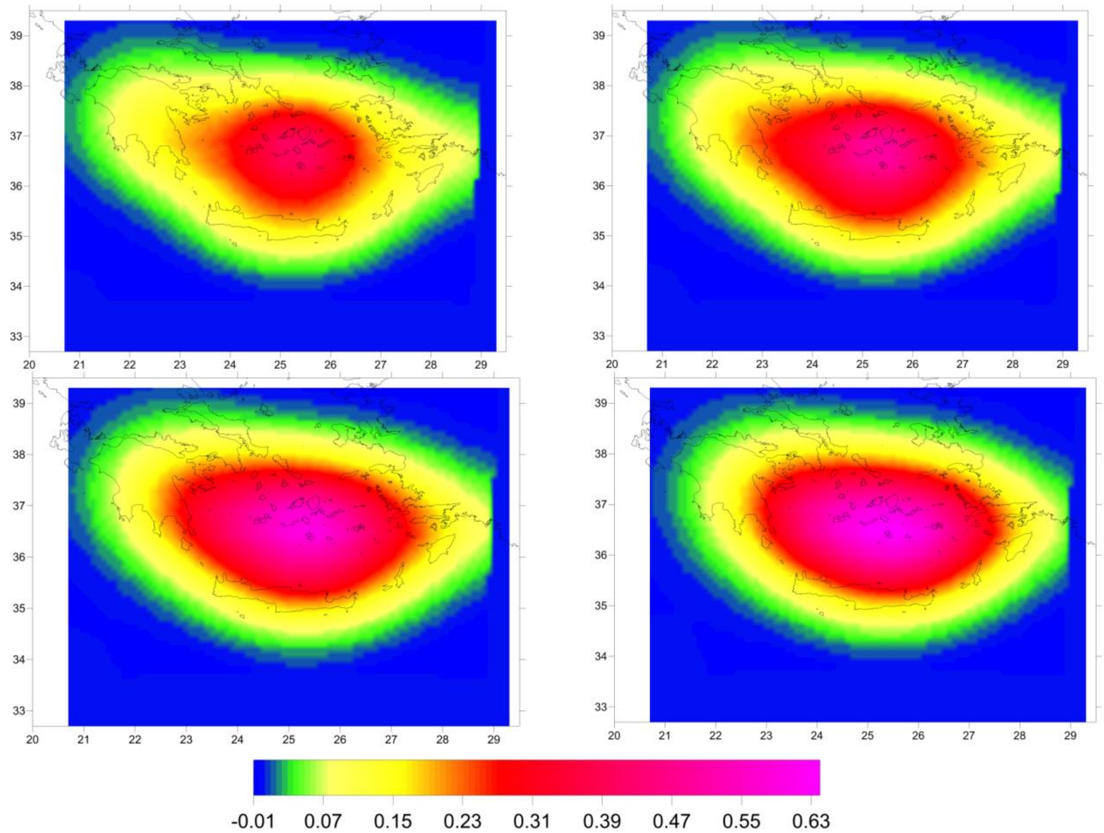
Ξεκινώντας από τη λύση του γραμμικού συστήματος, μεταξύ των τριών λύσεων (με την πράξη της διαίρεσης, με αναλυτικό υπολογισμό και με τη μέθοδο LSQR) δεν παρατηρήθηκαν σημαντικές διαφορές. Αντίθετα, υπήρχαν σημαντικές διαφορές μεταξύ των λύσεων αυτών και της αρχικής λύσης που είχε ήδη υπολογιστεί πριν από την εφαρμογή του κώδικα, και την οποία έπρεπε να προσεγγίζουν οι νέες λύσεις. Αυτό οφειλόταν σε κάποιες διαφορές μεταξύ των δύο κωδίκων της LSQR, αυτού που χρησιμοποιήσαμε εμείς κι αυτού που χρησιμοποιήθηκε στα πλαίσια της διδακτορικής διατριβής, όπως προαναφέρθηκε. Ειδικότερα, ο τελευταίος κώδικας περιείχε σύμφυτο damping (περιορισμούς για το μέτρο της λύσης), ενώ και οι επαναλήψεις (50) σταμάτησαν πολύ πριν τη μαθηματική σύγκλιση των προσεγγίσεων. Η σύγκλιση αυτή επετεύχθη από τον κώδικα που χρησιμοποιήσαμε, πραγματοποιώντας όμως περισσότερες επαναλήψεις (87). Εφαρμόζοντάς τον ξανά για μεγαλύτερες τιμές του damping, οδηγηθήκαμε σε νέες λύσεις, οι οποίες δεν διέφεραν σημαντικά από τη ζητούμενη.



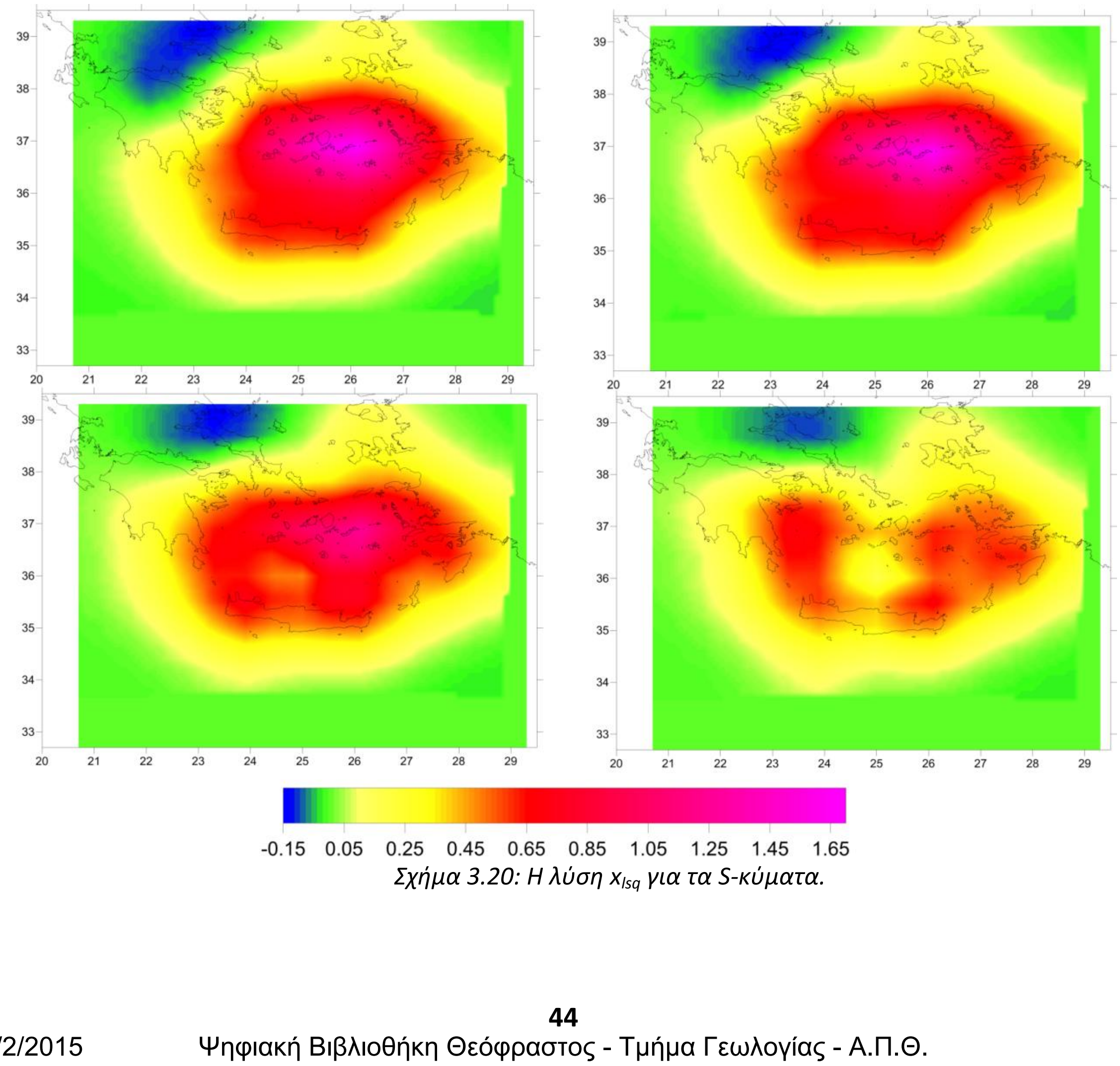

*Σχήμα 3.18: Εκτίμηση της εγγύτητας των λύσεων*

Σε σχέση με τις λύσεις που πήραμε με την εφαρμογή του κώδικα, οι λύσεις που βασίζονται στις προσεγγιστικές μορφές του πίνακα συμμεταβλητότητας παρουσιάζουν μεγάλες διαφορές από την πραγματική λύση, υπενθυμίζεται όμως ότι αυτές δεν χρησιμοποιούνται ποτέ στην πράξη. Όσον αφορά τον υπολογιστικό χρόνο που απαιτεί κάθε λύση, η LSQR αποδείχθηκε η οικονομικότερη μέθοδος, ενώ η διαίρεση ήταν πολύ πιο χρονοβόρα από τον αναλυτικό υπολογισμό της λύσης.

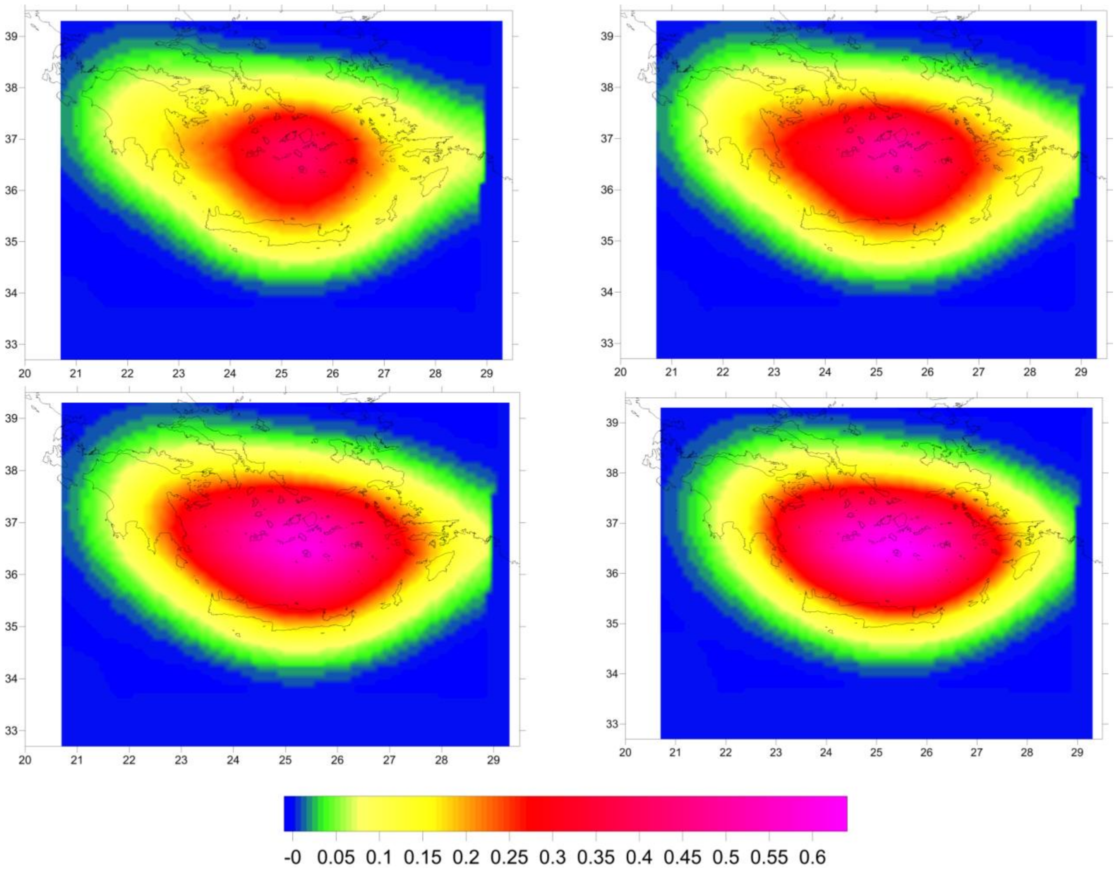
Ο σχεδιασμός της λύσης ωστόσο παρουσιάζει ορισμένες ιδιαιτερότητες, καθώς η επίλυση του συστήματος μας δίνει την τρισδιάστατη δομή απόσβεσης. Η οπτικοποίηση και στη συνέχεια η σύγκριση τρισδιάστατων δεδομένων δεν μπορεί να πραγματοποιηθεί στο σύνολό τους. Για τις ανάγκες της παρούσας εργασίας θα χρησιμοποιηθούν για τη σύγκριση, τόσο των λύσεων όσο και των πινάκων συμμεταβλητότητας, οριζόντιες τομές των αποτελεσμάτων στα βάθη 0, 20, 40 και 60 χιλιομέτρων. Τα γραφήματα που παρουσιάζονται βασίζονται στη χωρική αναπαράσταση με τη χρήση χρωματικής κλίμακας για τη σύγκριση των τιμών απόσβεσης στα διάφορα σημεία.



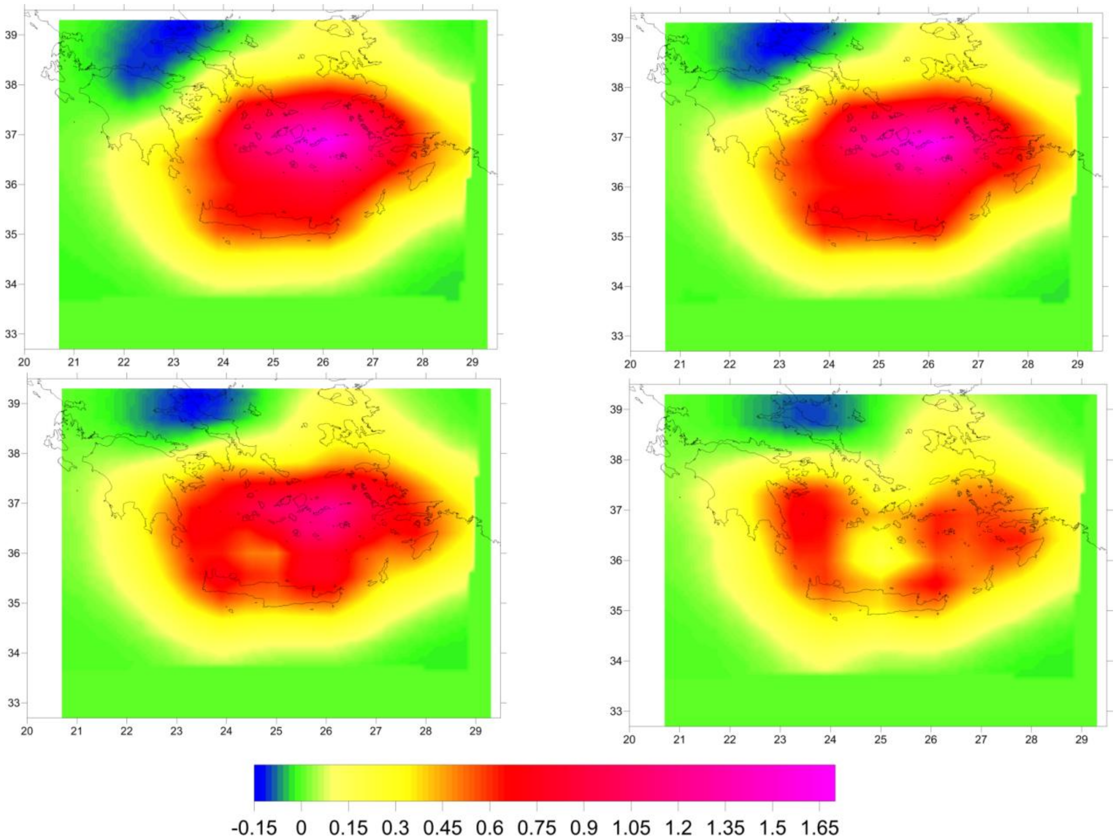

*Σχήμα 3.19: Η λύση $x_{lsq}$ για τα P-κύματα. Τα βάθη είναι τα εξής: πάνω αριστερά - 0 km, πάνω δεξιά - 20 km, κάτω αριστερά - 40 km, κάτω δεξιά - 60 km.*




*Σχήμα 3.20: Η λύση $x_{lsq}$ για τα S-κύματα.*

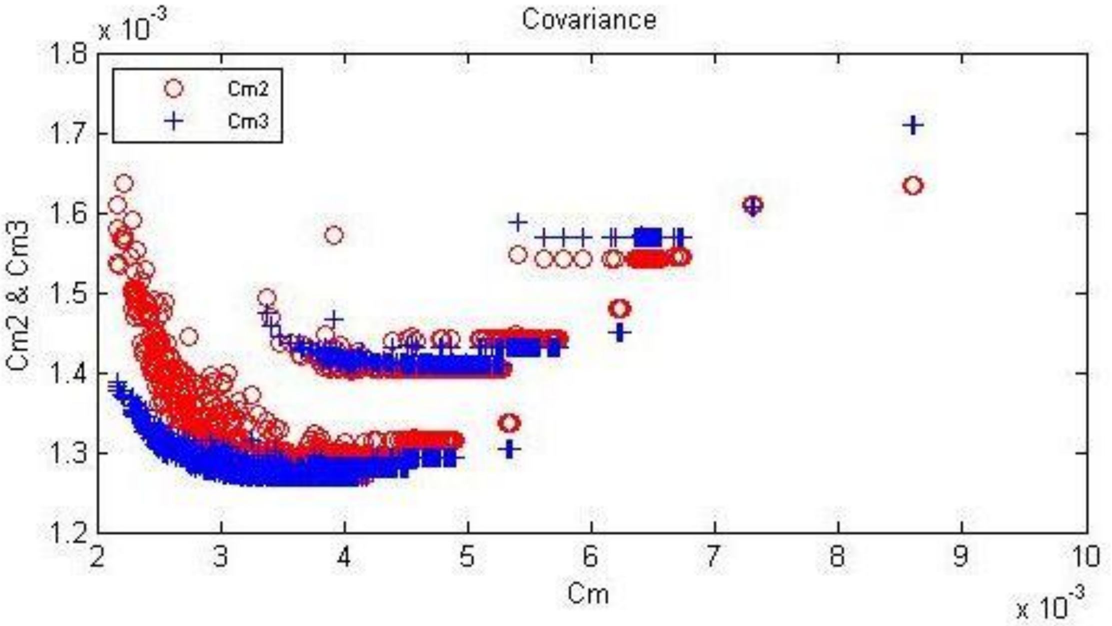



*Σχήμα 3.21: Η λύση $x_{lsqr}$ για τα P-κύματα.*




*Σχήμα 3.22: Η λύση $x_{lsqr}$ για τα S-κύματα.*

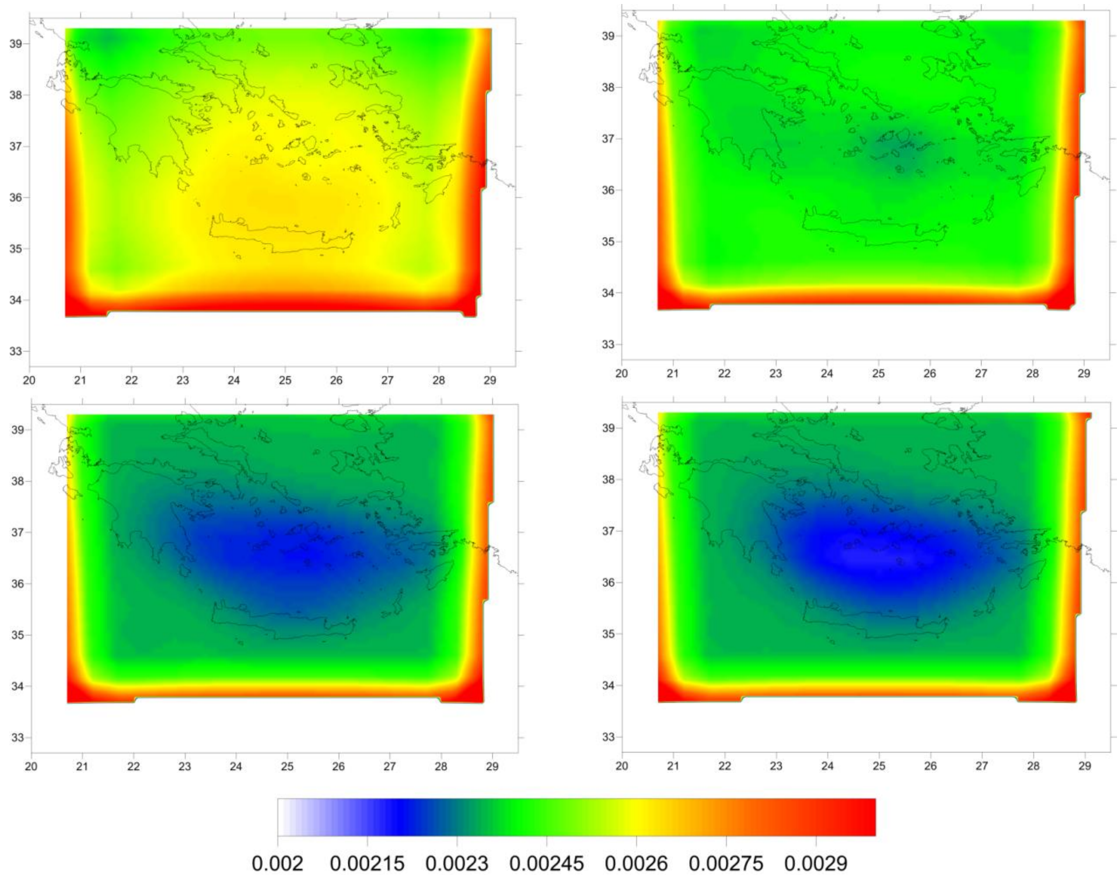
Σε ότι αφορά τον πίνακα συμμεταβλητότητας και ειδικότερα τις δύο προσεγγιστικές μορφές του, αυτές παρουσιάζουν σημαντικές διαφορές σε σχέση με την πραγματική μορφή του πίνακα (βλ. Σχ. 3.23). Οι διαφορές αυτές σημαίνουν ότι είναι αδύνατο να πάρουμε πληροφορίες για τις ακριβείς τιμές των σφαλμάτων από τις προσεγγίσεις, καθώς όλες οι προσεγγιστικές τιμές είναι πολύ μικρότερες από τις αντίστοιχες πραγματικές.



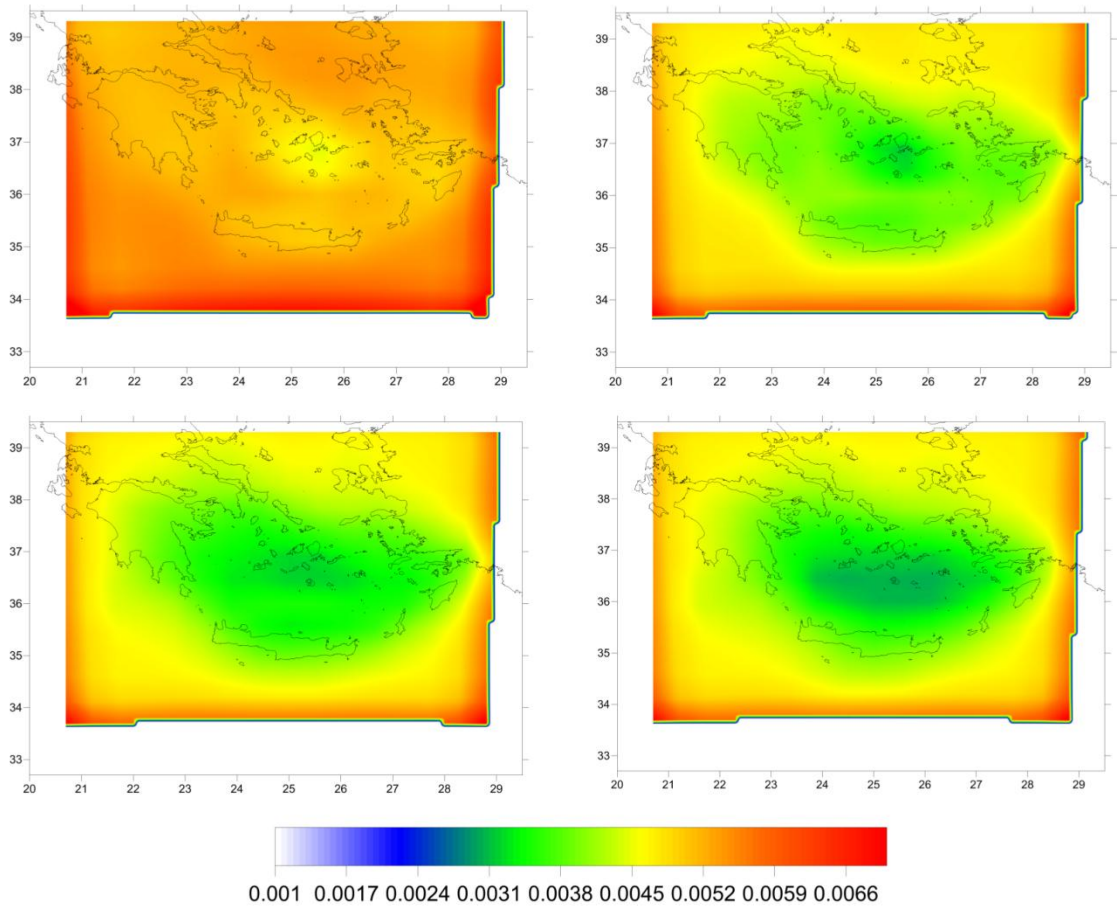

*Σχήμα 3.23: Οι προσεγγιστικές μορφές του πίνακα συμμεταβλητότητας σε σύγκριση με την ακριβή.*

Ωστόσο, δεν μπορούμε να εξαγάγουμε ούτε και σχετικά συμπεράσματα, δηλαδή να αποφανθούμε για ποιες παραμέτρους τα σφάλματα είναι μεγαλύτερα και για ποιες μικρότερα. Οι προσεγγιστικές μορφές του πίνακα συμμεταβλητότητας αποδίδουν σε παραμέτρους με μικρά πραγματικά σφάλματα άλλοτε μικρές (σχετικά) και άλλοτε μεγάλες τιμές σφαλμάτων. Το ίδιο ισχύει και για παραμέτρους με μεγάλα σφάλματα. Πιο συγκεκριμένα οι προσεγγιστικές τιμές είναι μεγαλύτερες στην επιφανειακή τομή σε σχέση με τα βαθύτερα στρώματα, κάτι που ανταποκρίνεται στην πραγματικότητα. Αντίθετα, οι προσεγγιστικές τιμές των σφαλμάτων και για τις δύο προσεγγίσεις είναι μεγαλύτερες στα κέντρα των οριζόντιων τομών δηλαδή στην περιοχή των Κυκλάδων (βλ. Σχ. 3.26–3.29), κάτι που έρχεται σε αντίθεση με τα πραγματικά σφάλματα, τα οποία είναι μεγαλύτερα στα άκρα των τομών και μικρότερα στα κέντρα.

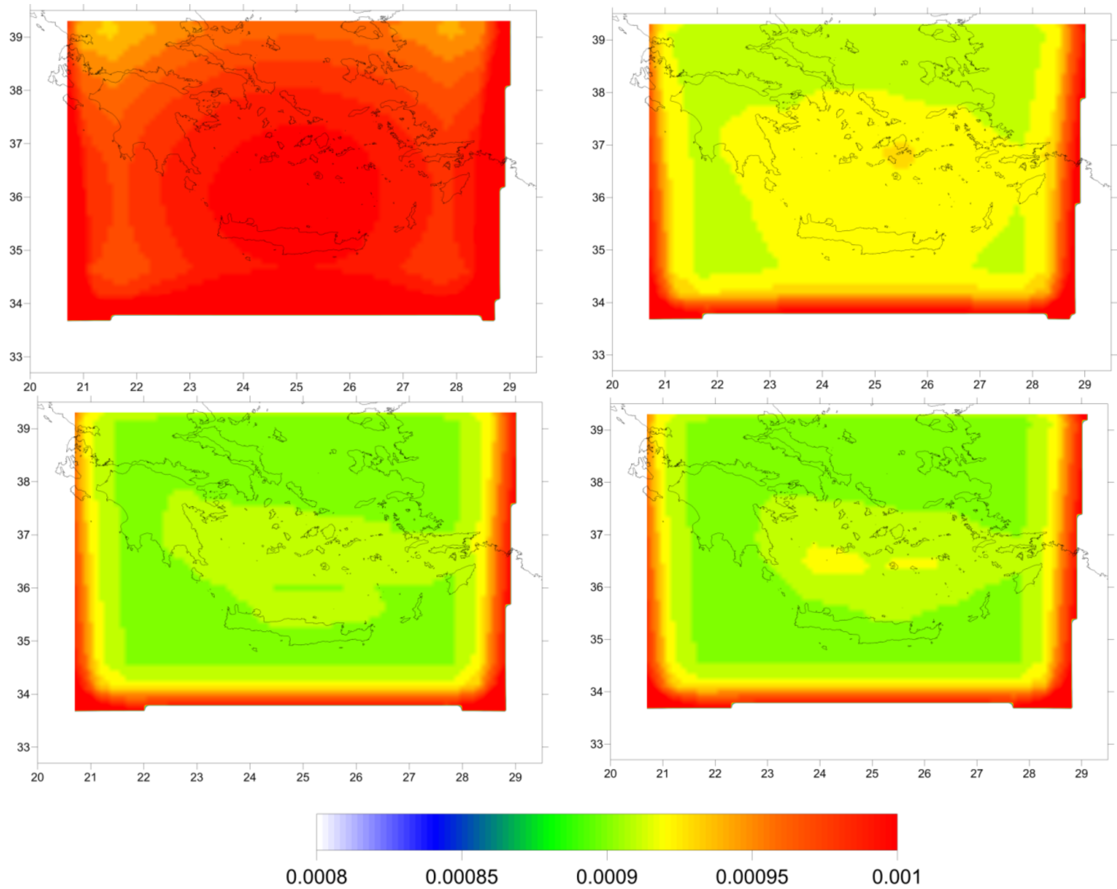
Είναι προφανές ότι οι προσεγγίσεις του πίνακα συμμεταβλητότητας δεν μπορούν στην περίπτωση αυτή να μας δώσουν αξιόπιστες πληροφορίες ούτε και για τα σχετικά σφάλματα. Αντίθετα οι προσεγγίσεις αυτές είναι «παραπλανητικές» και αν είχαμε μόνο αυτές στη διάθεσή μας θα οδηγουμασταν σε εσφαλμένα συμπεράσματα. Υπενθυμίζεται ότι οι προσεγγιστικές μορφές στήριζαν σε μεγάλο βαθμό τη θεωρητική ακρίβειά τους στη σποραδικότητα των προς επίλυση γραμμικών συστημάτων. Κάτι τέτοιο όμως δεν επιβεβαιώνεται στην πράξη, αν και ο πίνακας ο οποίος μελετήθηκε στην εργασία αυτή ήταν αρκετά σποραδικός.



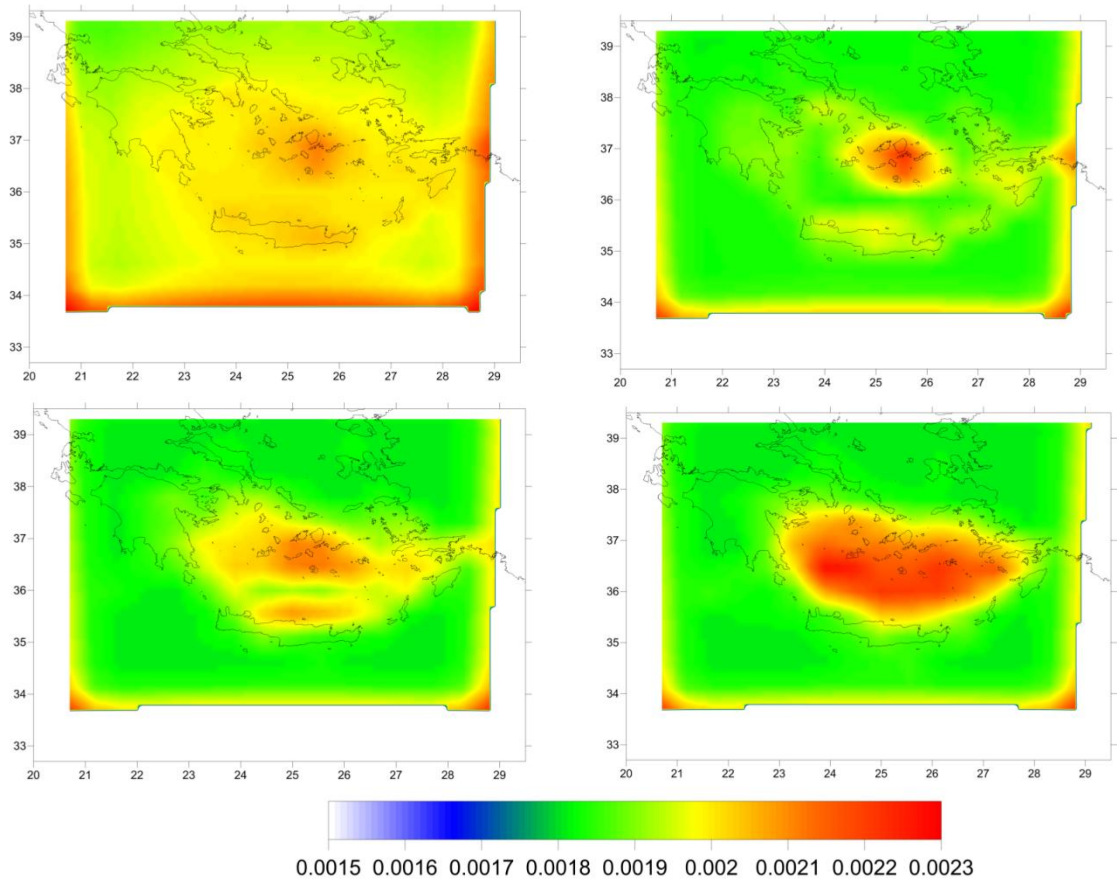

*Σχήμα 3.24: Τα σφάλματα για τα P-κύματα.*



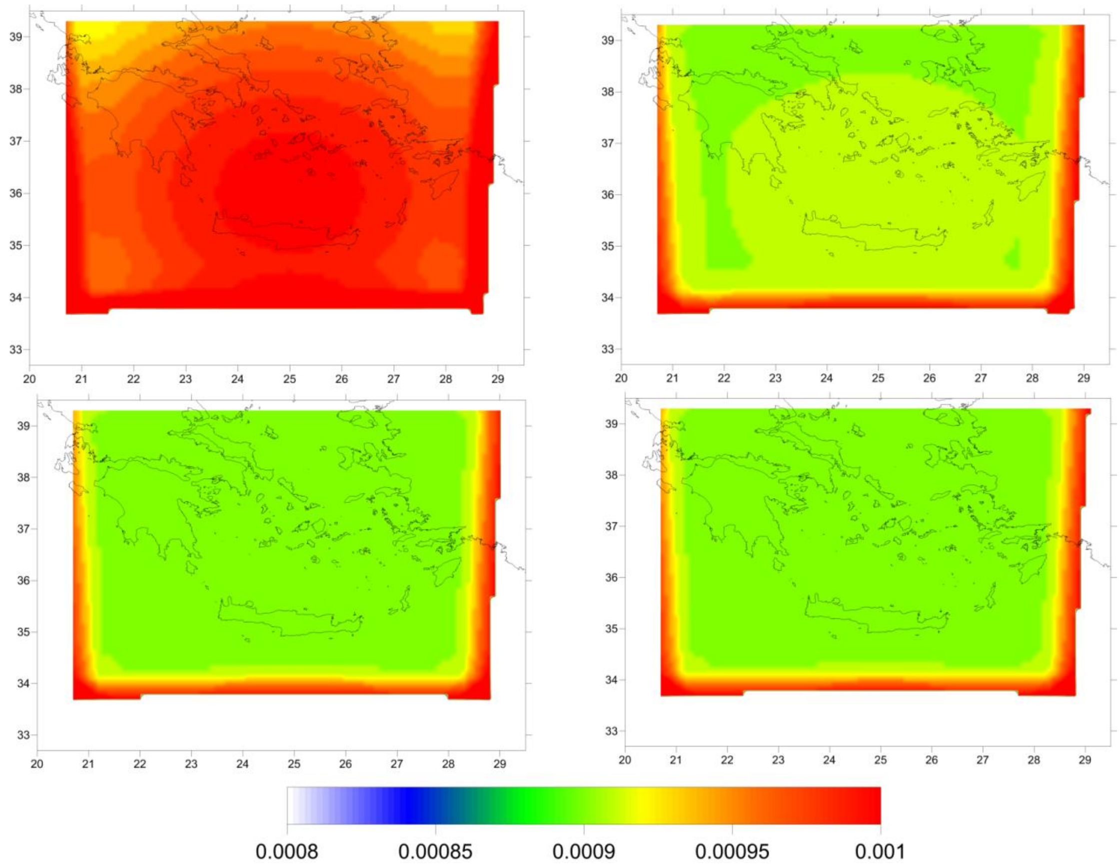

*Σχήμα 3.25: Τα σφάλματα για τα S-κύματα.*



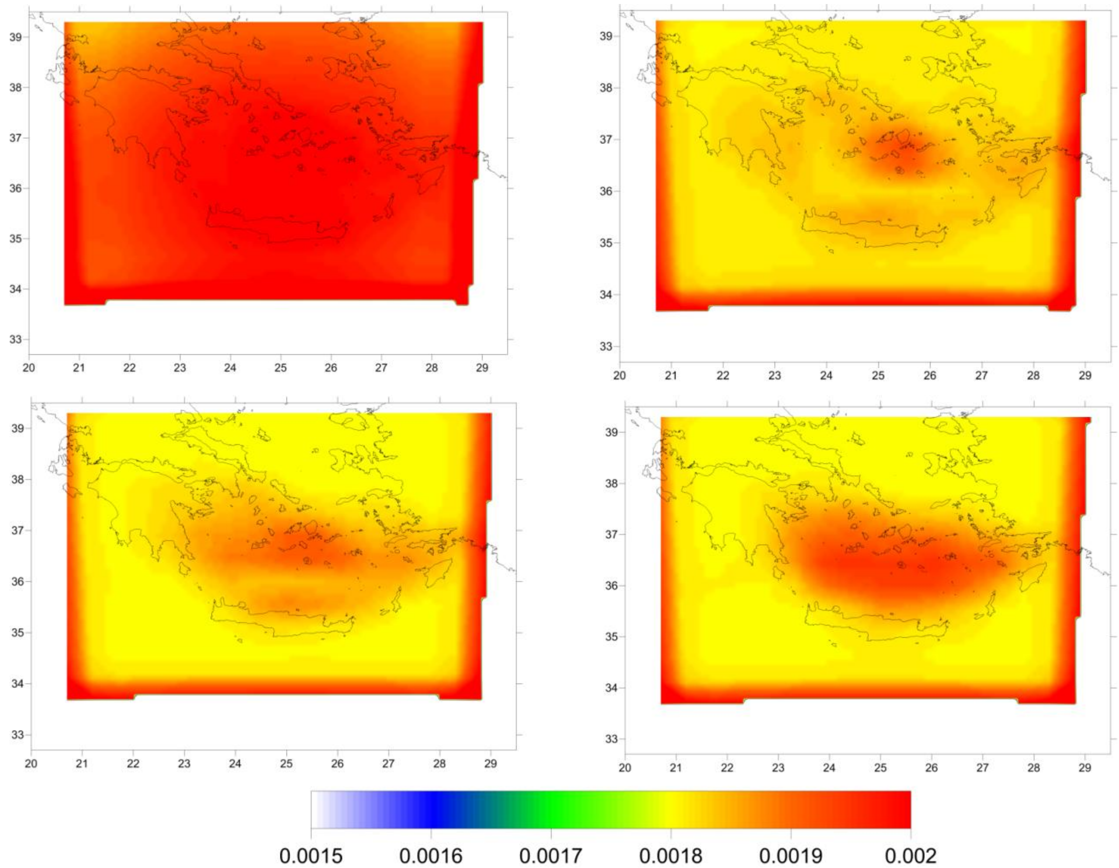
*Σχήμα 3.26: Πρώτη προσεγγιστική μορφή των σφαλμάτων για τα P-κύματα.*



*Σχήμα 3.27: Πρώτη προσεγγιστική μορφή των σφαλμάτων για τα S-κύματα.*




*Σχήμα 3.28: Δεύτερη προσεγγιστική μορφή των σφαλμάτων για τα P-κύματα.*




*Σχήμα 3.29: Δεύτερη προσεγγιστική μορφή των σφαλμάτων για τα S-κύματα.*

# 4 Συμπεράσματα

Ο βασικότερος στόχος μας μέσα από τις δοκιμές που πραγματοποιήθηκαν ήταν να διαπιστώσουμε την ακρίβεια, την αποτελεσματικότητα κι εν τέλει τη χρησιμότητα τόσο της επαναληπτικής μεθόδου LSQR, όσο και των προσεγγιστικών μορφών των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας. Η μέθοδος LSQR ήταν σε κάθε περίπτωση η πιο γρήγορη μέθοδος υπολογισμού της λύσης, δίνοντάς μας μάλιστα και την πιο υψηλή ακρίβεια, την οποία καθορίσαμε για τα προβλήματα που εξετάσαμε.

Αντίθετα, σε γενικές γραμμές, οι προσεγγίσεις των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας δεν αποδείχθηκαν αξιόπιστες, και σε κάποιες περιπτώσεις ήταν τελείως εσφαλμένες. Το σχετικό σφάλμα σε κάθε πείραμα ήταν μεγαλύτερο στον πίνακα συμμεταβλητότητας, δηλαδή οι προσεγγίσεις ήταν πιο ικανοποιητικές για τους πίνακες διακριτικής ικανότητας. Ακόμη, η δεύτερη προσεγγιστική μορφή ήταν γενικά πιο ακριβής από την πρώτη. Μάλιστα, η πρώτη προσεγγιστική μορφή απειριζόταν, δηλαδή δεν έδινε καθόλου αποτέλεσμα όταν εφαρμόστηκε σε πολύ αραιούς (sparse) πίνακες.

Όπου εφαρμόστηκε η λύση ελαχίστων τετραγώνων, οι προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας που μελετήθηκαν εμφάνισαν μεγάλα σφάλματα σε συστήματα που πλησιάζουν προς τα τετραγωνικά, όταν δηλαδή τα δεδομένα είναι κατά πολύ λίγο περισσότερα από τις παραμέτρους του μοντέλου. Αντίθετα, όταν το πλήθος των πρώτων ήταν αρκετές φορές μεγαλύτερο από τους αγνώστους, οι προσεγγίσεις πλησίαζαν αισθητά τους πραγματικούς πίνακες. Σε γενικές γραμμές φαίνεται ότι όταν εφαρμόζεται η λύση ελαχίστων τετραγώνων, το μέγεθος των σφαλμάτων δεν επηρεάζεται από το πλήθος των μηδενικών στοιχείων, αυξάνει όταν μεγαλώνει το πλήθος των αγνώστων και ελαττώνεται όταν μεγαλώνει το πλήθος των εξισώσεων.

Στα προβλήματα που επιλύθηκαν με τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου, οι προσεγγιστικές μορφές των πινάκων συμμεταβλητότητας και διακριτικής ικανότητας είναι εν γένει καλύτερες, πιο κοντά στις πραγματικές μορφές των πινάκων αυτών, συγκριτικά με την απλή λύση ελαχίστων τετραγώνων. Οι προσεγγίσεις γίνονται καλύτερες όσο οι διαστάσεις του προβλήματος γίνονται μεγαλύτερες, άσχετα αν αυτό που μεγαλώνει είναι το πλήθος των εξισώσεων ή το πλήθος των αγνώστων. Επίσης, οι προσεγγιστικές μορφές είναι πιο ακριβείς όταν ο πίνακας δεν έχει πολλά μηδενικά στοιχεία.

Η εφαρμογή του κώδικα σε πραγματικό πρόβλημα κατέδειξε την αδυναμία των προσεγγιστικών μορφών να δώσουν κάποιες ικανοποιητικές προσεγγίσεις του πίνακα συμμεταβλητότητας. Επιπλέον, οι προσεγγίσεις ήταν παραπλανητικές και δεν μας επέτρεψαν να εξαγάγουμε συμπεράσματα ούτε για το ποιες παράμετροι έχουν μικρότερα ή μεγαλύτερα σφάλματα σε σχέση με τις άλλες.

Ο υπολογιστικός χρόνος που απαιτούσε η LSQR ήταν ο μικρότερος σε σχέση με τις άλλες δύο λύσεις κι επηρεαζόταν το λιγότερο σπό τις μεταβολές στις διαστάσεις του συστήματος. Η αναλυτική λύση με την αντιστροφή του πίνακα ήταν πιο γρήγορη από τη λύση που παρείχε η διαίρεση πινάκων της MATLAB, όμως αυτό οφείλεται στο ότι τα συστήματα που μελετήσαμε ήταν σχετικά μικρά. Γενικότερα, ο υπολογιστικός χρόνος δεν φάνηκε να επηρεάζεται από το πλήθος των μηδενικών στοιχείων του πίνακα, ενώ αυξανόταν γραμμικά όσο μεγάλωνε το πλήθος των εξισώσεων, χωρίς ωστόσο να παρατηρούνται σημαντικές διαφορές. Αντίθετα, η αύξηση του πλήθους των αγνώστων είχε ως αποτέλεσμα οι υπολογιστικοί χρόνοι να αυξάνονται εκθετικά και μάλιστα σημαντικά (εκτός από την LSQR).

Τέλος, πρέπει να σημειωθεί η αποτελεσματικότητα της MATLAB σε ό,τι έχει να κάνει με τις πράξεις πινάκων, καθώς οι ρουτίνες των αντίστοιχων υπολογισμών για σποραδικούς πίνακες που υλοποιήσαμε δοκιμαστικά αποδείχθηκαν πιο αργές από τις υπορουτίνες που είναι ενσωματωμένες στη MATLAB.

# 5 Βιβλιογραφία - Αναφορές


[1] Menke, W., 1989, Geophysical data analysis: Discrete inverse theory: Revised Edition, *Academic Press Inc, San Diego.*

[2] Aki, K., Richards, P.G., 1980, Quantitative Seismology: Theory and Methods, *W. H. Freeman and Company, San Francisco.*

[3] Παπαζάχος, Β., Κ., 1996, Εισαγωγή στην εφαρμοσμένη γεωφυσική: *Εκδόσεις Ζήτη, Θεσσαλονίκη.*

[4] Γεωργίου, Γ., Ξενοφώντος, Χ., 2007, Εισαγωγή στη MATLAB, *Λευκωσία.*

[5] Paige, C.C. and Saunders, M.A., 1982, LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Soft.*, **8** 43-71 and 195-209

[6] Nolet, G., Montelli, R. and Virieux, J., 1999, Explicit, approximate expressions for the resolution and *a posteriori* covariance of massive tomographic systems, *Geophys. J. Int.,* **138**, 36-44

[7] Παπαζάχος, Β., Κ., Γεωφυσική Θεωρία Αντιστροφής, σημειώσεις μαθήματος

## ΠΑΡΑΡΤΗΜΑ

### Κώδικες

A) LSQR

Σημειώνουμε ότι αν υπάρχει το σύμβολο % στην αρχή της σειράς, η MATLAB θεωρεί ότι η σειρά είναι σχόλιο του συντάκτη του κώδικα, το χρωματίζει με πράσινο και την αγνοεί κατά την εκτέλεση του προγράμματος.

```
%                          LSQR()
%              void lsqr(int m, int n, float **A, float *u)
%              m = num of rows
%              n = num of colomns
%              A = augmented matrix (Jacobian + Smoothness)
%              u = augmented b vector
%
%              output is xx = the solution of the system
%
%    ------------------------------------------------------------
%
%            * This function performs iterative inversion with LSQR
*
%        * It solves the system      Adx=dy
%          with the constrain      Derivative x=0
%                         ---  ----          ---  ----
%                         |   A   |  --- ---   |   dy  |
%                         |       | |  x  |=   |       |
%                         | Deriv | |-- ---    |   0   |
%                         ---  ----          ---  ----
%
%    ------------------------------------------------------------
%
%    LSQR  finds a solution x to the following problems:
%
%     1. Unsymmetric equations --    solve  A*x = b
%
%     2. Linear least squares  --    solve  A*x = b
%                                   in the least-squares sense
%
%     3. Damped least squares  --    solve  (   A    )*x = ( b )
```

```
%                                  ( damp*I )     ( 0 )
%                             in the least-squares sense
%
%     where A is a matrix with m rows and n columns, b is an
%     m-vector, and damp is a scalar.  (All quantities are real.)
%     The matrix A is intended to be large and sparse.  It is
accessed
%     by means of subroutine calls of the form
%
%              aprod ( mode, m, n, x, y, UsrWrk )
%
%     which must perform the following functions:
%
%              If mode = 1, compute  y = y + A*x.
%              If mode = 2, compute  x = x + A(transpose)*y.
%
%     The vectors x and y are input parameters in both cases.
%     If  mode = 1,  y should be altered without changing x.
%     If  mode = 2,  x should be altered without changing y.
%     The parameter UsrWrk may be used for workspace as
described
%     below.
%
%     The rhs vector b is input via u, and subsequently
overwritten.
%
%
%     Note:  LSQR uses an iterative method to approximate the
solution.
%     The number of iterations required to reach a certain
accuracy
%     depends strongly on the scaling of the problem.  Poor
scaling of
%     the rows or columns of A should therefore be avoided
where
%     possible.
%
%     For example, in problem 1 the solution is unaltered by
%     row-scaling.  If a row of A is very small or large compared
to
%     the other rows of A, the corresponding row of ( A  b )
should be
```

```
%     scaled up or down.
%
%     In problems 1 and 2, the solution x is easily recovered
%     following column-scaling.  Unless better information is
known,
%     the nonzero columns of A should be scaled so that they
all have
%     the same Euclidean norm (e.g., 1.0).
%
%     In problem 3, there is no freedom to re-scale if damp is
%     nonzero.  However, the value of damp should be assigned
only
%     after attention has been paid to the scaling of A.
%
%     The parameter damp is intended to help regularize
%     ill-conditioned systems, by preventing the true solution
from
%     being very large.  Another aid to regularization is provided
by
%     the parameter acond, which may be used to terminate
iterations
%     before the computed solution becomes very large.
%
%     Note that x is not an input parameter.
%     If some initial estimate x0 is known and if damp = 0,
%     one could proceed as follows:
%
%       1. Compute a residual vector     r0 = b - A*x0.
%       2. Use LSQR to solve the system  A*dx = r0.
%       3. Add the correction dx to obtain a final solution x = x0
+ dx.
%
%     This requires that x0 be available before and after the call
%     to LSQR.  To judge the benefits, suppose LSQR takes k1
iterations
%     to solve A*x = b and k2 iterations to solve A*dx = r0.
%     If x0 is "good", norm(r0) will be smaller than norm(b).
%     If the same stopping tolerances atol and btol are used for
each
%     system, k1 and k2 will be similar, but the final solution x0
+ dx
```

**% should be more accurate. The only way to reduce the total work**
**% is to use a larger stopping tolerance for the second system.**
**% If some value btol is suitable for A*x = b, the larger value**
**% btol*norm(b)/norm(r0) should be suitable for A*dx = r0.**
**%**
**% Preconditioning is another way to reduce the number of iterations.**
**% If it is possible to solve a related system M*x = b efficiently,**
**% where M approximates A in some helpful way**
**% (e.g. M - A has low rank or its elements are small relative to**
**% those of A), LSQR may converge more rapidly on the system**
**% A*M(inverse)*z = b,**
**% after which x can be recovered by solving M*x = z.**
**%**
**% NOTE: If A is symmetric, LSQR should not be used!**
**% Alternatives are the symmetric conjugate-gradient method (cg)**
**% and/or SYMMLQ.**
**% SYMMLQ is an implementation of symmetric cg that applies to**
**% any symmetric A and will converge more rapidly than LSQR.**
**% If A is positive definite, there are other implementations of**
**% symmetric cg that require slightly less work per iteration**
**% than SYMMLQ (but will take the same number of iterations).**
**%**
**%**
**% Notation**
**% --------**
**%**
**% The following quantities are used in discussing the subroutine**
**% parameters:**
**%**
**% Abar = ( A ), bbar = ( b )**

```
%            ( damp*I )                 ( 0 )
%
%     r      =  b  -  A*x,          rbar  =  bbar  -  Abar*x
%
%     rnorm  =  sqrt( norm(r)**2  +  damp**2 * norm(x)**2 )
%            =  norm( rbar )
%
%     relpr  =  the relative precision of floating-point arithmetic
%            on the machine being used.  On most machines,
%            relpr is about 1.0e-7 and 1.0d-16 in single and
double
%            precision respectively.
%
%     LSQR  minimizes the function rnorm with respect to x.
%
%
%     Parameters
%     ----------
%
%     m       input      m, the number of rows in A.
%
%     n       input      n, the number of columns in A.
%
%     aprod   external   See above.
%
%     damp    input      The damping parameter for problem 3
above.
%                        (damp should be 0.0 for problems 1 and 2.)
%                        If the system A*x = b is incompatible, values
%                        of damp in the range 0 to sqrt(relpr)*norm(A)
%                        will probably have a negligible effect.
%                        Larger values of damp will tend to decrease
%                        the norm of x and reduce the number of
%                        iterations required by LSQR.
%
%                        The work per iteration and the storage needed
%                        by LSQR are the same for all values of damp.
%
%     rw      workspace  Transit pointer to user's workspace.
%                        Note:  LSQR  does not explicitly use this
%                        parameter, but passes it to subroutine aprod
for
```

```
%                    possible use as workspace.
%
%     u(m)    input      The rhs vector b.  Beware that u is
%                    over-written by LSQR.
%
%     v(n)    workspace
%
%     w(n)    workspace
%
%     x(n)    output     Returns the computed solution x.
%
%     se(*)   output     If m .gt. n  or  damp .gt. 0,  the system is
%           (maybe)    overdetermined and the standard errors
may be
%                    useful.  (See the first LSQR reference.)
%                    Otherwise (m .le. n  and  damp = 0) they do not
%                    mean much.  Some time and storage can be
saved
%                    by setting  se = NULL.  In that case, se will
%                    not be touched.
%
%                    If se is not NULL, then the dimension of se
must
%                    be n or more.  se(1:n) then returns standard
error
%                    estimates for the components of x.
%                    For each i, se(i) is set to the value
%                      rnorm * sqrt( sigma(i,i) / t ),
%                    where sigma(i,i) is an estimate of the i-th
%                    diagonal of the inverse of
Abar(transpose)*Abar
%                    and  t = 1      if  m .le. n,
%                        t = m - n  if  m .gt. n  and  damp = 0,
%                        t = m      if  damp .ne. 0.
%
%     atol    input      An estimate of the relative error in the
data
%                    defining the matrix A.  For example,
%                    if A is accurate to about 6 digits, set
%                    atol = 1.0e-6 .
%
```

```
%     btol    input      An estimate of the relative error in the
data
%                        defining the rhs vector b.  For example,
%                        if b is accurate to about 6 digits, set
%                        btol = 1.0e-6 .
%
%     conlim  input      An upper limit on cond(Abar), the
apparent
%                        condition number of the matrix Abar.
%                        Iterations will be terminated if a computed
%                        estimate of cond(Abar) exceeds conlim.
%                        This is intended to prevent certain small or
%                        zero singular values of A or Abar from
%                        coming into effect and causing unwanted
growth
%                        in the computed solution.
%
%                        conlim and damp may be used separately or
%                        together to regularize ill-conditioned systems.
%
%                        Normally, conlim should be in the range
%                        1000 to 1/relpr.
%                        Suggested value:
%                        conlim = 1/(100*relpr)  for compatible
systems,
%                        conlim = 1/(10*sqrt(relpr)) for least squares.
%
%             Note:  If the user is not concerned about the
parameters
%             atol, btol and conlim, any or all of them may be set
%             to zero.  The effect will be the same as the values
%             relpr, relpr and 1/relpr respectively.
%
%     itnlim  input      An upper limit on the number of iterations.
%                        Suggested value:
%                        itnlim = n/2   for well-conditioned systems
%                                       with clustered singular values,
%                        itnlim = 4*n   otherwise.
%
%     nout    input      File number for printed output.  If positive,
%                        a summary will be printed on file nout.
%
```

```
%     istop   output     An integer giving the reason for
termination:
%
%              0       x = 0  is the exact solution.
%                      No iterations were performed.
%
%              1       The equations A*x = b are probably
%                      compatible.  Norm(A*x - b) is sufficiently
%                      small, given the values of atol and btol.
%
%              2       damp is zero.  The system A*x = b is probably
%                      not compatible.  A least-squares solution has
%                      been obtained that is sufficiently accurate,
%                      given the value of atol.
%
%              3       damp is nonzero.  A damped least-squares
%                      solution has been obtained that is sufficiently
%                      accurate, given the value of atol.
%
%              4       An estimate of cond(Abar) has exceeded
%                      conlim.  The system A*x = b appears to be
%                      ill-conditioned.  Otherwise, there could be an
%                      error in subroutine aprod.
%
%              5       The iteration limit itnlim was reached.
%
%     itn     output     The number of iterations performed.
%
%     anorm   output     An estimate of the Frobenius norm of
Abar.
%                      This is the square-root of the sum of squares
%                      of the elements of Abar.
%                      If damp is small and if the columns of A
%                      have all been scaled to have length 1.0,
%                      anorm should increase to roughly sqrt(n).
%                      A radically different value for anorm may
%                      indicate an error in subroutine aprod (there
%                      may be an inconsistency between modes 1
and 2).
%
%     acond   output     An estimate of cond(Abar), the condition
%                      number of Abar.  A very high value of acond
```

```
%                    may again indicate an error in aprod.
%
%    rnorm   output    An estimate of the final value of
norm(rbar),
%                    the function being minimized (see notation
%                    above).  This will be small if A*x = b has
%                    a solution.
%
%    arnorm  output    An estimate of the final value of
%                    norm( Abar(transpose)*rbar ), the norm of
%                    the residual for the usual normal equations.
%                    This should be small in all cases.  (arnorm
%                    will often be smaller than the true value
%                    computed from the output vector x.)
%
%    xnorm   output    An estimate of the norm of the final
%                    solution vector x.
%
%
%    Subroutines and functions used
%    ------------------------------
%
%    USER          aprod
%    CBLAS          dcopy, dnrm2, dscal (see Lawson et al.
below)
%
%
%    References
%    ----------
%
%    C.C. Paige and M.A. Saunders,  LSQR: An algorithm for
sparse
%        linear equations and sparse least squares,
%        ACM Transactions on Mathematical Software 8, 1
(March 1982),
%        pp. 43-71.
%
%    C.C. Paige and M.A. Saunders,  Algorithm 583, LSQR:
Sparse
%        linear equations and least-squares problems,
%        ACM Transactions on Mathematical Software 8, 2
(June 1982),
```

```
%         pp. 195-209.
%
%     C.L. Lawson, R.J. Hanson, D.R. Kincaid and F.T. Krogh,
%         Basic linear algebra subprograms for Fortran usage,
%         ACM Transactions on Mathematical Software 5, 3
(Sept 1979),
%         pp. 308-323 and 324-325.
%     ---------------------------------------------------------------
%
%
%     LSQR development:
%     22 Feb 1982: LSQR sent to ACM TOMS to become
Algorithm 583.
%     15 Sep 1985: Final F66 version.  LSQR sent to "misc" in
netlib.
%     13 Oct 1987: Bug (Robert Davies, DSIR).  Have to delete
%                     if ( (one + dabs(t)) .le. one ) GO TO 200
%                  from loop 200.  The test was an attempt to reduce
%                  underflows, but caused w(i) not to be updated.
%     17 Mar 1989: First F77 version.
%     04 May 1989: Bug (David Gay, AT&T).  When the second
beta is zero,
%                  rnorm = 0 and
%                  test2 = arnorm / (anorm * rnorm) overflows.
%                  Fixed by testing for rnorm = 0.
%     05 May 1989: Sent to "misc" in netlib.
%     14 Mar 1990: Bug (John Tomlin via IBM OSL testing).
%                  Setting rhbar2 = rhobar**2 + dampsq can give
zero
%                  if rhobar underflows and damp = 0.
%                  Fixed by testing for damp = 0 specially.
%     15 Mar 1990: Converted to lower case.
%     21 Mar 1990: d2norm introduced to avoid overflow in
numerous
%                  items like  c = sqrt( a**2 + b**2 ).
%     04 Sep 1991: wantse added as an argument to LSQR, to
make
%                  standard errors optional.  This saves storage and
%                  time when se(*) is not wanted.
%     13 Feb 1992: istop now returns a value in [1,5], not [1,7].
%                  1, 2 or 3 means that x solves one of the problems
```

```
%               Ax = b,  min norm(Ax - b)  or  damped least
squares.
%               4 means the limit on cond(A) was reached.
%               5 means the limit on iterations was reached.
%    07 Dec 1994: Keep track of dxmax = max_k norm( phi_k *
d_k ).
%               So far, this is just printed at the end.
%               A large value (relative to norm(x)) indicates
%               significant cancellation in forming
%               x  =  D*f  =  sum( phi_k * d_k ).
%               A large column of D need NOT be serious if the
%               corresponding phi_k is small.
%    27 Dec 1994: Include estimate of alfa_opt in iteration log.
%               alfa_opt is the optimal scale factor for the
%               residual in the "augmented system", as described
by
%               A. Bjorck (1992),
%               Pivoting and stability in the augmented system
method,
%               in D. F. Griffiths and G. A. Watson (eds.),
%               "Numerical Analysis 1991",
%               Proceedings of the 14th Dundee Conference,
%               Pitman Research Notes in Mathematics 260,
%               Longman Scientific and Technical, Harlow, Essex,
1992.
%    14 Apr 2006: "Line-by-line" conversion to ISO C by
%               Michael P. Friedlander.
%
%
%    Michael A. Saunders                mike@sol-
michael.stanford.edu
%    Dept of Operations Research         na.Msaunders@na-
net.ornl.gov
%    Stanford University
%    Stanford, CA 94305-4022            (415) 723-1875
%-----------------------------------------------------------------------
%
%    References
%    ----------
%
%    C.C. Paige and M.A. Saunders,  LSQR: An algorithm for
sparse
```

```
%          linear equations and sparse least squares,
%          ACM Transactions on Mathematical Software 8, 1
(March 1982),
%          pp. 43-71.
%
%     C.C. Paige and M.A. Saunders,  Algorithm 583, LSQR:
Sparse
%          linear equations and least-squares problems,
%          ACM Transactions on Mathematical Software 8, 2
(June 1982),
%          pp. 195-209.
%
%     C.L. Lawson, R.J. Hanson, D.R. Kincaid and F.T. Krogh,
%          Basic linear algebra subprograms for Fortran usage,
%          ACM Transactions on Mathematical Software 5, 3
(Sept 1979),
%          pp. 308-323 and 324-325.
%
%---------------------------------------------------------------------------
%

function [lisi,it] = lssqr(m,n,A,u,cg_itr)

%   FILE *out5,*out10;
%    float s,ss,alfa,beta,b,a,c,s1,b1,rhobar,phibar0,phibar,psi,
%
rho,teta,r,phi,t1,t2,t3,tau,dnorm,dknorm,bnorm,rnorm,rnorm0,a
norm,arnorm,
%
acond,delta,gambar,rhs,zbar,xnorm,xnorm0,gamma,cs2,sn2,z,
xnorm1,res2;
% infout=fopen('lsqr_info.dat','w');
zero = 0.0;
one = 1.0;
it = 0;

% Values for Stopping Quantities
atol   = 1.0e-6;
btol   = 1.0e-6;
conlim = 1/(10*sqrt(1.0e-7));

% LSQR Iterations
```

```
%   cg_itr = min(num_mes,num_param);
% cg_itr = 200;
itn     =   0;
istop  =   0;
nstop  =   0;
maxdx  =   0;
ctol    =   zero;

if conlim > zero
    ctol = one / conlim;
end

anorm  =   zero;
acond  =   zero;
dnorm  =   zero;
dxmax  =   zero;
res2    =   zero;
psi     =   zero;
xnorm  =   zero;
xnorm1 =   zero;
cs2     = - one;
sn2     =   zero;
z        =   zero;
damp   =   zero;
%  damp    =    1.0e-7;
%   -----------------------------------------------------------------
%  Intitilize vectors
%  Set up the first vectors u and v for the bidiagonalization.
%  These satisfy  beta*u = b,  alpha*v = A(transpose)*u.
%  -----------------------------------------------------------------
xx=zeros(n,1);
vv=zeros(n,1);
alpha=zero;
beta = norm(u);   % norm of u stored in beta

if beta>zero
    u=(one / beta).*u;   % scale of u
    vv=A'*u;
    alpha = norm(vv);                  % norm of v stored in alpha
end

if alpha>zero
```

```
   vv=(one / alpha).*vv;
   ww=vv;
end
arnorm=alpha * beta;
arnorm0=alpha * beta;
if arnorm == 0
   disp('Convergence in LSQR in initialization');
   % exit(1);  ?????????????
end

rhobar  =  alpha;
phibar  =  beta;
bnorm   =  beta;
rnorm   =  beta;

% phibar0 =  phibar;

% write info in the file
%fprintf(infout,"---------------------------------------------\n");
%   fprintf(infout,"    Iteration -> %d\n",itr);
 %   fprintf(infout,"  LSQR_steps       normX       normR
CondA\n");

% printf( "step-> %d    rnomr = %f\n",it,rnorm); getch();

%
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
%   out5=fopen('lsqr_solution.d','a');
%   fprintf(out5,"****INVERSION ITERATION==>%d *****\n",itr);
%   fprintf(out5,"---------------------------------\n");
%   fprintf(out5,"  LSQR_ITR     MODEL_NORM
Non_Linear_Error     Linear_Error  \n");
%   fclose(out5);
%   ??????????????????????????????

%     start iterations

for it=1:cg_itr
 %  fprintf('      LSQR %d/%d\r',it,cg_itr);

%     -----------------------------------------------------------------
```

```
%     Perform the next step of the bidiagonalization to obtain
the
%     next  beta, u, alpha, v.  These satisfy the relations
%              beta*u  =  A*v  -  alpha*u,
%             alpha*v  =  A(transpose)*u  -  beta*v.
%    ------------------------------------------------------------

u=- alpha.*u;   %scale of u with -alpha
u=u+A*vv;
beta = norm(u);    %norm of u stored in beta

% Accumulate  anorm = || Bk || =  sqrt( sum of  alpha**2 +
beta**2 + damp**2 ).
    temp = d2norm( alpha, beta );
    temp = d2norm( temp , damp );
    anorm = d2norm( anorm, temp );

if beta > zero
   u=(one / beta).*u;  %scale of u with 1/beta
   vv=(- beta).*vv;     %scale of v with -beta
   vv=vv+A'*u;
   alpha = norm(vv);        %norm of v stored in alpha
    if alpha > zero
          vv=(one / alpha).*vv;
    end
end

%      ---------------------------------------------------------------
%      Use a plane rotation to eliminate the subdiagonal
element (beta)
%      of the lower-bidiagonal matrix, giving an upper-bidiagonal
matrix.
%      ---------------------------------------------------------------
rhbar1 = rhobar;
rho    =   d2norm( rhbar1, beta );
cs     =   rhbar1 / rho;
sn     =   beta   / rho;
theta  =   sn * alpha;
rhobar = - cs * alpha;
phi    =   cs * phibar;
phibar =   sn * phibar;
tau    =   sn * phi;
```

```
%       ---------------------------------------------------------------
%       Update  x, w  and (perhaps) the standard error estimates.
%       ---------------------------------------------------------------
t1      =   phi   / rho;
t2      = - theta / rho;
t3      =   one   / rho;
dknorm =   zero;

for i1=1:n
    t=ww(i1);
    xx(i1)=t1*t+xx(i1);
    ww(i1)=t2*t+vv(i1);
    dknorm = (t3*t)*(t3*t)+dknorm;
end

%       ---------------------------------------------------------------
%       Monitor the norm of d_k, the update to x.
%       dknorm = norm( d_k )
%       dnorm  = norm( D_k ),        where   D_k = (d_1, d_2, ..., d_k )
%       dxk    = norm( phi_k d_k ),  where new x = x_k + phi_k d_k.
%       ---------------------------------------------------------------
dknorm = sqrt( dknorm );
dnorm  = d2norm( dnorm, dknorm );
dxk    = abs( phi * dknorm );
if dxmax<dxk
  dxmax = dxk;
  maxdx = itn;
end

%       ---------------------------------------------------------------
%       Use a plane rotation on the right to eliminate the
%       super-diagonal element (theta) of the upper-bidiagonal
matrix.
%       Then use the result to estimate  norm(x).
%       ---------------------------------------------------------------
delta  =   sn2 * rho;
gambar = - cs2 * rho;
rhs    =   phi    - delta * z;
zbar   =   rhs    / gambar;
xnorm  =   d2norm( xnorm1, zbar  );
gamma  =   d2norm( gambar, theta );
cs2    =   gambar / gamma;
```

```
sn2    =  theta  / gamma;
z      =  rhs    / gamma;
xnorm1 =   d2norm( xnorm1, z     );

%       ------------------------------------------------------------------
%       Test for convergence.
%       First, estimate the norm and condition of the matrix
Abar,
%       and the norms of  rbar  and  Abar(transpose)*rbar.
%       ------------------------------------------------------------------
acond  =   anorm * dnorm;
res2   =   d2norm( res2 , psi    );
rnorm  =   d2norm( res2 , phibar );
arnorm =   alpha * abs( tau );

%       Now use these norms to estimate certain other
quantities,
%       some of which will be small near a solution.
alfopt =   sqrt( rnorm / (dnorm * xnorm) );
test1  =   rnorm /  bnorm;
test2  =   zero;
if rnorm>zero
   test2=arnorm/(anorm*rnorm);
end
test3  =   one   /  acond;
t1     =   test1 / (one  +  anorm * xnorm / bnorm);
rtol   =   btol  +  atol *  anorm * xnorm / bnorm;

%
*****************************************************************
********** */
% ------------------------------------------------------------
       % Stopping LSQR
stop1 = btol*bnorm + atol*anorm*xnorm;
stop2 = atol*anorm*rnorm;
%disp(it);disp('normr= ');disp(rnorm);disp('stop1= ');disp(stop1);
%disp('normAr= ');disp(arnorm);disp('stop2= ');disp(stop2);
%disp('condA= ');disp(acond);disp('conlim= ');disp(conlim);

if( (acond >= conlim ) || (rnorm <= stop1) || (arnorm <= stop2) )
       break
end
```

```
% Write info in the file
   %   fprintf(infout,'    %d      %.6e    %.6e
%.6e\n',it,xnorm,rnorm,acond);
   %    if it==cg_itr
   %        fprintf(infout,'-----------------------------------------');
   %    end
%disp(xx);
end
lisi=xx;
end


%
*************************************************************
*****
% d2norm  returns
%  with precautions
% to avoid overflow.
%
%
*************************************************************
*****

function metro2=d2norm(a,b)
   zero=0.0;
   scale = abs(a) + abs(b);
   if scale == zero
      metro2=zero;
   else
      metro2=scale * sqrt( (a/scale)*(a/scale) +
(b/scale)*(b/scale) );
   end
   % return (float)sqrt((a*a)+(b*b));
end
```

B) Επίλυση γραμμικού συστήματος και πίνακες συμμεταβλητότητας και διακριτικής ικανότητας

Για τους κώδικες αυτούς στα μεμονωμένα προβλήματα ο χρήστης καθόριζε την τιμή των μεταβλητών mm, nn και cut. Στα επαναληπτικά τεστ, οι αντίστοιχες σειρές του κώδικα μετατράπηκαν σε σχόλια (φαίνονται στις πρώτες σειρές με πράσινο χρώμα) και προστέθηκαν ο τίτλος function… και στο τέλος η λέξη end που δηλώνει το πέρας της συνάρτησης.

i. Λύση ελαχίστων τετραγώνων

```
function
[Tlsqr,Tlsq,Tinv,dx,dCm2,dCm3,dR2,dR3]=lssqr2(mm,nn,cut)

%clear
%clc
%mm=800;
%nn=400;
%cut=1.0;

noise=0.5;
A=randn(mm,nn);

for ii=1:mm
    for jj=1:nn
        if (abs(A(ii,jj)) < cut)
            A(ii,jj)=0;
        end
    end
end

% Solution
for kk=1:nn
xx(kk)=sin(2*kk*pi/nn);
end
xx=xx';
d=A*xx+noise*randn(mm,1);

%[mm,nn]=size(A)
    tic;[xlsqr,itnum]=lssqr(mm,nn,A,d,10000);
    Tlsqr=toc;

    tic;xlsq=A\d;
    Tlsq=toc;

    tic;L1=inv(A'*A)*A';
    xlsq1=L1*d;
```

```
Tinv=toc;

dlsq=norm(xlsq1-xlsq);
dlsqr=norm(xlsqr-xlsq);

Cm=L1*L1';
R1=L1*A;

% 1st approximate
AAt=A*A';
s=sum(AAt.^2);
D=diag(diag(AAt));
for k=1:mm
   D(k,k)=D(k,k)/s(k);
end
L2=A'*D;
Cm2=L2*L2';
R2=L2*A;
x2=L2*d;

% 2nd approximate
AtA=A'*A;
s2=sum(AtA.^2);
D2=diag(diag(AtA));
for k=1:nn
   D2(k,k)=D2(k,k)/s2(k);
end
L3=D2*A';
Cm3=L3*L3';
R3=L3*A;
x3=L3*d;
% plots
subplot(2,2,1)
plot(xlsq,xlsqr,'ko') %,xlsq,xlsqr_sp,'r+')
xlabel('xlsq')
ylabel('xlsqr')

subplot(2,2,2)
plot(xlsq,x2,'ro',xlsq,x3,'b+')
xlabel('xlsq')
ylabel('x2 & x3')
legend('x2','x3','Location','northwest')
```

```
    set(legend,'FontSize',10);

    subplot(2,2,3)

plot(sqrt(diag(Cm)),sqrt(diag(Cm2)),'ro',sqrt(diag(Cm)),sqrt(diag
(Cm3)),'b+')
    title('Covariance')
    xlabel('Cm')
    ylabel('Cm2 & Cm3')
    legend('Cm2','Cm3','Location','southwest')
    set(legend,'FontSize',7);

    subplot(2,2,4)
    hist(diag(R2),20)
    hold on
    h = findobj(gca,'Type','patch');
    set(h,'FaceColor','r','EdgeColor','w')
    hist(diag(R3),20)
    title('Resolution')
    legend('R2','R3');
    set(legend,'FontSize',7);

    % Comparison
     dx=norm(xlsqr-xlsq)/norm(xlsq);

     dx2=norm(x2-xlsq)/norm(xlsq);

     dx3=norm(x3-xlsq)/norm(xlsq);

    dCm2=norm(diag(Cm2-Cm))/norm(diag(Cm));
    dCm3=norm(diag(Cm3-Cm))/norm(diag(Cm));

    dR2= norm(diag(R2-R1))/norm(diag(R1));
    dR3= norm(diag(R3-R1))/norm(diag(R1));
end
```

ii. Λύση ελαχίστων τετραγώνων ελαχίστου μέτρου

```
function
[Tlsqr,Tlsq,Tinv,dx,dCm2,dCm3,dR2,dR3]=lssqr3(mm,nn,cut)
```

```
%clear
%clc
%mm=800;
%nn=400;
%cut=1.0;

mm1=mm;
damp=20;
damp2=damp*damp;
%cut=1.0;
A1=randn(mm,nn);

for ii=1:mm
   for jj=1:nn
      if (abs(A1(ii,jj)) < cut) A1(ii,jj)=0;
      end
   end
end

C=damp*eye(nn,nn);
A=[A1;C];
d2=zeros(nn,1);
for kk=1:nn
xx(kk)=sin(2*kk*pi/nn);
end
xx=xx';
d1=A1*xx+0.02*randn(mm,1);
d=[d1;d2];
mm=mm+nn;
%[mm,nn]=size(A)
   tic;[xlsqr,itnum]=lssqr(mm,nn,A,d,1000);
   Tlsqr=toc;

   tic;xlsq=A\d;
   Tlsq=toc;

   tic;
   L1=inv(A1'*A1+damp2*eye(nn,nn))*A1';
   xlsq1=L1*d1;
   Tinv=toc;
```

```
dlsq=norm(xlsq1-xlsq);
dlsqr=norm(xlsqr-xlsq);

Cm=L1*L1';
R1=L1*A1;

% 1st approximate
AAt=A1*A1';
s=sum(AAt.^2);
D=diag(diag(AAt));
for k=1:mm1
   D(k,k)=D(k,k)/s(k);
end
L2=A1'*D;
Cm2=L2*L2';
R2=L2*A1;
x2=L2*d1;

% 2nd approximate
AtA=A1'*A1;
s2=sum(AtA.^2);
D2=diag(diag(AtA));
for k=1:nn
   D2(k,k)=D2(k,k)/s2(k);
end
L3=D2*A1';
Cm3=L3*L3';
R3=L3*A1;
x3=L3*d1;

% plots
 subplot(2,2,1)
 plot(xlsq,xlsqr,'ko')
 xlabel('xlsq')
 ylabel('xlsqr')

 subplot(2,2,2)
 plot(xlsq,x2,'ro',xlsq,x3,'b+')
 xlabel('xlsq')
 ylabel('x2 & x3')
 legend('x2','x3','Location','northwest')
 set(legend,'FontSize',10);
```

```
subplot(2,2,3)
plot(sqrt(diag(Cm)),sqrt(diag(Cm2)),'ro',sqrt(diag(Cm)),sqrt(diag
(Cm3)),'b+')
title('Covariance')
xlabel('Cm')
ylabel('Cm2 & Cm3')
legend('Cm2','Cm3','Location','southwest')
set(legend,'FontSize',7);

subplot(2,2,4)
plot(sqrt(diag(R1)),sqrt(diag(R2)),'ro',sqrt(diag(R1)),sqrt(diag(R
3)),'b+')
title('Resolution')
xlabel('R1')
ylabel('R2 & R3')
legend('R2','R3','Location','northwest')
set(legend,'FontSize',7);

% Comparison
%X11=xcorr(xlsq,xlsqr,'coeff');
%xc=X11(nn)
dx=norm(xlsqr-xlsq)/norm(xlsq);

%X12=xcorr(xlsq,x2,'coeff');
%xc2=X12(nn)
dx2=norm(x2-xlsq)/norm(xlsq);

%X13=xcorr(xlsq,x3,'coeff');
%xc3=X13(nn)
dx3=norm(x3-xlsq)/norm(xlsq);

dCm2=norm(diag(Cm2-Cm))/norm(diag(Cm));
dCm3=norm(diag(Cm3-Cm))/norm(diag(Cm));

dR2= norm(diag(R2-R1))/norm(diag(R1));
dR3= norm(diag(R3-R1))/norm(diag(R1));
end
```
Γ) Επαναληπτικά τεστ

I. Τεστ για το σημείο αποκοπής

```
% test "cut"
clear
clc
m=800;
n=400;
cut=[-0.0001 0.1 0.3 0.5 0.8 1.0 1.2]
card=20; %sample number

Tlsqr=zeros(card,length(cut));
Tlsq=zeros(card,length(cut));
Tinv=zeros(card,length(cut));
dx=zeros(card,length(cut));
dCm2=zeros(card,length(cut));
dCm3=zeros(card,length(cut));
dR2=zeros(card,length(cut));
dR3=zeros(card,length(cut));

for ii=1:length(cut)
 for kk=1:card

[Tlsqr(kk,ii),Tlsq(kk,ii),Tinv(kk,ii),dx(kk,ii),dCm2(kk,ii),dCm3(kk
,ii),dR2(kk,ii),dR3(kk,ii)]=lssqr2(1) (800,400,cut(ii));
 end
end
% mean & std
Tlsqr1=[mean(Tlsqr);std(Tlsqr);median(Tlsqr)]
Tlsq1=[mean(Tlsq);std(Tlsq);median(Tlsq)]
Tinv1=[mean(Tinv);std(Tinv);median(Tinv)]
dx1=[mean(dx);std(dx);median(dx)]
dCm21=[mean(dCm2);std(dCm2);median(dCm2)]
dCm31=[mean(dCm3);std(dCm3);median(dCm3)]
dR21=[mean(dR2);std(dR2);median(dR2)]
dR31=[mean(dR3);std(dR3);median(dR3)]

% plots
hold off
subplot(2,2,1)
errorbar(cut-0.01,Tinv1(1,:),Tinv1(2,:),'r+')
hold on
```

```
errorbar(cut,Tlsq1(1,:),Tlsq1(2,:),'g+')
errorbar(cut+0.01,Tlsqr1(1,:),Tlsqr1(2,:),'b+')
plot(cut-
0.01,Tinv1(3,:),'ro',cut,Tlsq1(3,:),'go',cut+0.01,Tlsqr1(3,:),'bo')
legend('Tinv','Tlsq','Tlsqr')
xlabel('cut')
ylabel('time required')

subplot(2,2,2)
errorbar(cut,dx1(1,:),dx1(2,:),'k+')
hold on
plot(cut,dx1(3,:),'ko')
xlabel('cut')
ylabel('error of solution using LSQR')

subplot(2,2,3)
```

---

(1) 2 για λύση ελαχίστων τετραγώνων και 3 για τη λύση ελαχίστων τετραγώνων ελαχίστου μέτρου

```
errorbar(cut,dCm21(1,:),dCm21(2,:),'r+')
hold on
errorbar(cut+0.01,dCm31(1,:),dCm31(2,:),'b+')
plot(cut,dCm21(3,:),'ro',cut+0.01,dCm31(3,:),'bo')
legend('1st approx','2nd approx')
xlabel('cut')
ylabel('error in variance')

subplot(2,2,4)
errorbar(cut,dR21(1,:),dR21(2,:),'r+')
hold on
errorbar(cut+0.01,dR31(1,:),dR31(2,:),'b+')
plot(cut,dR21(3,:),'ro',cut+0.01,dR31(3,:),'bo')
legend('1st approx','2nd approx')
xlabel('cut')
ylabel('error in resolution')
```

II. Τεστ για το πληθος των αγνώστων

```
% test "n-unknowns"
clear
clc
```

```
m=1000;
cut=1.0;
n=[50 100 200 400 800]
card=20; %sample number

Tlsqr=zeros(card,length(n));
Tlsq=zeros(card,length(n));
Tinv=zeros(card,length(n));
dx=zeros(card,length(n));
dCm2=zeros(card,length(n));
dCm3=zeros(card,length(n));
dR2=zeros(card,length(n));
dR3=zeros(card,length(n));

for ii=1:length(n)
 for kk=1:card

[Tlsqr(kk,ii),Tlsq(kk,ii),Tinv(kk,ii),dx(kk,ii),dCm2(kk,ii),dCm3(kk
,ii),dR2(kk,ii),dR3(kk,ii)]=lssqr3(m,n(ii),cut);
 end
end

Tlsqr1=[mean(Tlsqr);std(Tlsqr);median(Tlsqr)]
Tlsq1=[mean(Tlsq);std(Tlsq);median(Tlsq)]
Tinv1=[mean(Tinv);std(Tinv);median(Tinv)]
dx1=[mean(dx);std(dx);median(dx)]
dCm21=[mean(dCm2);std(dCm2);median(dCm2)]
dCm31=[mean(dCm3);std(dCm3);median(dCm3)]
dR21=[mean(dR2);std(dR2);median(dR2)]
dR31=[mean(dR3);std(dR3);median(dR3)]

% plots
hold off
subplot(2,2,1)
errorbar(n-1,Tinv1(1,:),Tinv1(2,:),'r+')
hold on
errorbar(n,Tlsq1(1,:),Tlsq1(2,:),'g+')
errorbar(n+1,Tlsqr1(1,:),Tlsqr1(2,:),'b+')
plot(n-1,Tinv1(3,:),'ro',n,Tlsq1(3,:),'go',n+1,Tlsqr1(3,:),'bo')
legend('Tinv','Tlsq','Tlsqr')
xlabel('number of unknowns')
ylabel('time required')
```

```
subplot(2,2,2)
errorbar(n,dx1(1,:),dx1(2,:),'k+')
hold on
plot(n,dx1(3,:),'ko')
xlabel('number of unknowns')
ylabel('error of solution using LSQR')

subplot(2,2,3)
errorbar(n,dCm21(1,:),dCm21(2,:),'r+')
hold on
errorbar(n+1,dCm31(1,:),dCm31(2,:),'b+')
plot(n,dCm21(3,:),'ro',n+1,dCm31(3,:),'bo')
legend('1st approx','2nd approx')
xlabel('number of unknowns')
ylabel('error in variance')

subplot(2,2,4)
errorbar(n,dR21(1,:),dR21(2,:),'r+')
hold on
errorbar(n+1,dR31(1,:),dR31(2,:),'b+')
plot(n,dR21(3,:),'ro',n+1,dR31(3,:),'bo')
legend('1st approx','2nd approx')
xlabel('number of unknowns')
ylabel('error in resolution')
```

III. Τεστ για το πλήθος των εξισώσεων

```
% test "m-equations"
clear
clc
n=400;
cut=1.0;
m=[600 800 1000 1200 1400]
card=100; %sample number

Tlsqr=zeros(card,length(m));
Tlsq=zeros(card,length(m));
Tinv=zeros(card,length(m));
dx=zeros(card,length(m));
dCm2=zeros(card,length(m));
dCm3=zeros(card,length(m));
```

```
dR2=zeros(card,length(m));
dR3=zeros(card,length(m));

for ii=1:length(m)
 for kk=1:card

[Tlsqr(kk,ii),Tlsq(kk,ii),Tinv(kk,ii),dx(kk,ii),dCm2(kk,ii),dCm3(kk
,ii),dR2(kk,ii),dR3(kk,ii)]=lssqr3(m(ii),n,cut);
 end
end

Tlsqr1=[mean(Tlsqr);std(Tlsqr);median(Tlsqr)]
Tlsq1=[mean(Tlsq);std(Tlsq);median(Tlsq)]
Tinv1=[mean(Tinv);std(Tinv);median(Tinv)]
dx1=[mean(dx);std(dx);median(dx)]
dCm21=[mean(dCm2);std(dCm2);median(dCm2)]
dCm31=[mean(dCm3);std(dCm3);median(dCm3)]
dR21=[mean(dR2);std(dR2);median(dR2)]
dR31=[mean(dR3);std(dR3);median(dR3)]

% plots
hold off
subplot(2,2,1)
errorbar(m-1,Tinv1(1,:),Tinv1(2,:),'r+')
hold on
errorbar(m,Tlsq1(1,:),Tlsq1(2,:),'g+')
errorbar(m+1,Tlsqr1(1,:),Tlsqr1(2,:),'b+')
plot(m-1,Tinv1(3,:),'ro',m,Tlsq1(3,:),'go',m+1,Tlsqr1(3,:),'bo')
legend('Tinv','Tlsq','Tlsqr')
xlabel('number of equations')
ylabel('time required')

subplot(2,2,2)
errorbar(m,dx1(1,:),dx1(2,:),'k+')
hold on
plot(m,dx1(3,:),'ko')
xlabel('number of equations')
ylabel('error of solution using LSQR')

subplot(2,2,3)
errorbar(m,dCm21(1,:),dCm21(2,:),'r+')
hold on
```

```
errorbar(m+1,dCm31(1,:),dCm31(2,:),'b+')
plot(m,dCm21(3,:),'ro',m+1,dCm31(3,:),'bo')
legend('1st approx','2nd approx')
xlabel('number of equations')
ylabel('error in variance')

subplot(2,2,4)
errorbar(m,dR21(1,:),dR21(2,:),'r+')
hold on
errorbar(m+1,dR31(1,:),dR31(2,:),'b+')
plot(m,dR21(3,:),'ro',m+1,dR31(3,:),'bo')
legend('1st approx','2nd approx')
xlabel('number of equations')
ylabel('error in resolution')
```