

Αλγόριθμοι εύρεσης ιδιοτιμών

Χαλκής Απόστολος

Διπλωματική Εργασία

Επιβλέπων καθηγητής:
Κωνσταντίνος Χρυσάφινος



Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Εθνικό Μετσόβιο Πολυτεχνείο

Περιεχόμενα

Πρόλογος	i
1 Δείκτες κατάστασης και ευστάθεια	1
I Δείκτες κατάστασης	1
I.I Κατάσταση προβλήματος	1
I.II Απόλυτος δείκτης κατάστασης	2
I.III Σχετικός δείκτης κατάστασης	2
I.IV Κατάσταση ενός πίνακα	5
I.V Δείκτης κατάστασης ενός πίνακα	7
II Αριθμητική κινητής υποδιαστολής	8
II.I Όρια ψηφιακής αναπαράστασης	8
II.II Αριθμοί κινητής υποδιαστολής	9
II.III Το έψιλον μηχανής	9
II.IV Αριθμητική κινητής υποδιαστολής	10
II.V Σχετικά με το έψιλον μηχανής	11
III Ευστάθεια	12
III.I Αλγόριθμοι	12
III.II Ακρίβεια	13
III.III Ευστάθεια	13
III.IV Προς τα πίσω ευστάθεια	14
III.V Η σημασία του $O(\epsilon_{machine})$	14
III.VI Ευστάθεια στην αριθμητική κινητής υποδιαστολής	16
III.VII Ένας ασταθής αλγόριθμος	16
III.VIII Ακρίβεια προς τα πίσω ευσταθών αλγορίθμων	19
III.IX Ευστάθεια της Householder τριγωνοποίησης	20
IV Βασικά στοιχεία της θεωρίας διαταραχών	23
2 Εύρεση Ιδιοτιμών	27
I Το πρόβλημα των ιδιοτιμών	27
I.I Ιδιοτιμές και Ιδιοδιανύσματα	27
I.II Διαγωνιοποίηση Πίνακα	30
I.III Ορθομοναδία διαγωνιοποίηση	31
I.IV Schur παραγοντοποίηση	32
I.V Συμπεράσματα	33

II	Επισκόπηση αλγορίθμων	34
II.I	Προφανείς αλγόριθμοι: Μειονεκτήματα	34
II.II	Μία θεμελιώδης δυσκολία	35
II.III	Schur παραγοντοποίηση και διαγωνιοποίηση	36
II.IV	Οι δύο φάσεις του υπολογισμού ιδιοτιμών	37
III	Μετασχηματισμός σε μορφή Hessenberg	39
III.I	Μία κακή ιδέα	39
III.II	Μία καλή ιδέα	40
III.III	Χρόνος εκτέλεσης	41
III.IV	Η περίπτωση ερμιτιανού πίνακα	43
III.V	Ευστάθεια	43
IV	Πηλίκο Rayleigh και αντίστροφη επανάληψη	44
IV.I	Αρχική σύμβαση-αυτοπεριορισμός	44
IV.II	Πηλίκο Rayleigh	45
IV.III	Η μέθοδος των δυνάμεων	46
IV.IV	Αντίστροφη επανάληψη	48
IV.V	Επανάληψη ηλίκου Rayleigh	50
IV.VI	Ταχύτητα σύγκλισης	52
V	Αλγόριθμος QR χωρίς εναλλαγές	53
V.I	Ο αλγόριθμος QR	53
V.II	Μη κανονικοποιημένες ταυτόχρονες επαναλήψεις	54
V.III	Ταυτόχρονες Επαναλήψεις	57
V.IV	Ισοδυναμία αλγόριθμου QR και ταυτόχρονων επαναλήψεων	58
V.V	Σύγκλιση του αλγόριθμου QR	59
VI	Αλγόριθμος QR με εναλλαγές	61
VI.I	Σύνδεση με την αντίστροφη επανάληψη	61
VI.II	Σύνδεση με την αντίστροφη επανάληψη με εναλλαγές	62
VI.III	Σύνδεση με την επανάληψη ηλίκου Rayleigh	63
VI.IV	Εναλλαγή Wilkinson	64
VI.V	Ευστάθεια και ακρίβεια	65
VII	Άλλοι αλγόριθμοι	67
VII.I	Αλγόριθμος Jacobi	67
VII.II	Αλγόριθμος διχοτόμησης	70
VII.III	Αλγόριθμος διαίρει και βασίλευε	73
VIII	Παραγοντοποίηση SVD	79
VIII.I	Στοιχεία θεωρίας	79
VIII.II	Οι ιδιοτιμές του A^*A και παραγοντοποίηση SVD	81
VIII.III	Μια διαφορετική αναγωγή σε πρόβλημα ιδιοτιμών	82
VIII.IV	Δύο φάσεις	83
VIII.V	Διδιαγωνιοποίηση Golub-Kahan	84
VIII.VI	Ταχύτερες μέθοδοι για την πρώτη φάση	85
VIII.VII	Δεύτερη φάση	87
3	Υλοποίηση στο MATLAB	89
I	Μετασχηματισμός σε μορφή Hessenberg	89

II	Πηλίκο Rayleigh και αντίστροφη επανάληψη	92
II.I	Μέθοδος των δυνάμεων	92
II.II	Αλγόριθμος αντίστροφης επανάληψης	93
II.III	Επανάληψη πηλίκου Rayleigh	94
III	Αλγόριθμος QR χωρίς εναλλαγές	96
IV	Αλγόριθμος QR με εναλλαγές	97
V	Άλλοι αλγόριθμοι	107
V.I	Αλγόριθμος Jacobi	107
V.II	Αλγόριθμος διχοτόμησης	111
V.III	Αλγόριθμος διαίρει και βασίλευε	119

Πρόλογος

Στην παρούσα εργασία θα μελετήσουμε αλγόριθμους για την εύρεση ιδιοτιμών και ιδιοδιανυσμάτων τυχαίων πινάκων, καθώς και θα παρουσιάσουμε κάποια πολύ εισαγωγικά στοιχεία για τις μεθόδους εύρεσης ιδιοτιμών.

Στο πρώτο κεφάλαιο κάνουμε μια παρουσίαση στα βασικά στοιχεία θεωρίας που θα χρειαστούμε για τη μελέτη οποιουδήποτε θέματος στην αριθμητική γραμμική άλγεβρα και την αριθμητική ανάλυση. Δίνουμε τον ορισμό του προβλήματος και των δεικτών κατάστασής του και κάποια χαρακτηριστικά παραδείγματα προβλημάτων καλής και κακής κατάστασης. Στη συνέχεια παρουσιάζουμε κάποια βασικά στοιχεία της αριθμητικής κινητής υποδιαστολής, όπως η ψηφιακή αναπαράσταση και το έψιλον μηχανής. Τέλος δίνουμε τον ορισμό του αλγόριθμου μαζί με τις έννοιες της ακρίβειας, της ευστάθειας και της προς τα πίσω ευστάθειας και μελετάμε τον αλγόριθμο εύρεσης ιδιοτιμών μέσω των συντελεστών του χαρακτηριστικού πολυωνύμου προκειμένου να βγάλουμε κάποια αρχικά συμπεράσματα.

Στο δεύτερο κεφάλαιο, αρχικά, παραθέτουμε κάποια βασικά στοιχεία θεωρίας και θεωρήματα της γραμμικής άλγεβρας για τη διαγωνιοποίηση, τη Schur παραγοντοποίηση, αλλά και βασικά θεωρήματα για τα χαρακτηριστικά μεγέθη ενός πίνακα. Στη συνέχεια δείχνουμε κάποιες βασικές δυσκολίες προκειμένου να κατασκευαστούν οι ζητούμενοι αλγόριθμοι, βγάζουμε κάποια συμπεράσματα για το τι μορφή πρέπει να έχουν και αναφέρουμε το βασικό τους στόχο και δομή. Συνεχίζοντας παρουσιάζουμε μια θεωρητική μελέτη αλγορίθμων για την εύρεση ιδιοτιμών και ιδιοδιανυσμάτων. Συγκεκριμένα αναπτύσσονται οι αλγόριθμοι:

1. Μέθοδος των δυνάμεων
2. Μέθοδος αντίστροφης επανάληψης
3. Επανάληψη πηλίκου Rayleigh
4. Απλός αλγόριθμος QR χωρίς εναλλαγές
5. Πρακτικός αλγόριθμος QR με εναλλαγές
6. Αλγόριθμος ταυτόχρονων επαναλήψεων

7. Αλγόριθμος Jacobi
8. Αλγόριθμος διχοτόμησης
9. Αλγόριθμος διαίρει και βασίλευε

ενώ αναφορά γίνεται και στη μέθοδο διδιαγωνιοποίησης Golub-Kahan.

Στο τρίτο κεφάλαιο υλοποιούμε στο MATLAB όλους τους παραπάνω αλγόριθμους εκτός από τη μέθοδο διδιαγωνιοποίησης Golub-Kahan. Επιπρόσθετα σε κάποιους αλγόριθμους αποσαφηνίζουμε κάποια θεωρητικά και πρακτικά ζητήματα, ενώ συγκρίνουμε ταχύτητες σύγκλισης και παραθέτουμε τα αντίστοιχα διαγράμματα.

Κεφάλαιο 1

Δείκτες κατάστασης και ευστάθεια

I Δείκτες κατάστασης

Σε αυτό το κεφάλαιο θα παρουσιάσουμε δύο θεμελιώδη ζητήματα της αριθμητικής ανάλυσης. Το πρώτο είναι το ζήτημα της κατάστασης που έχει να κάνει με τη συμπεριφορά διαταραχών ενός μαθηματικού προβλήματος και το δεύτερο είναι η ευστάθεια που έχει να κάνει με τη συμπεριφορά για μικρές διαταραχές ενός αλγορίθμου που χρησιμοποιούμε για να λύσουμε το ίδιο πρόβλημα σε υπολογιστή.

I.I Κατάσταση προβλήματος

Σε αυτή την ενότητα θεωρούμε ένα πρόβλημα ως μια συνάρτηση $f : X \rightarrow Y$ από ένα διανυσματικό χώρο X , δεδομένων X , με νόρμα σε ένα διανυσματικό χώρο λύσεων Y με νόρμα. Αυτή η συνάρτηση είναι συνήθως μη γραμμική (ακόμα και στη γραμμική άλγεβρα), αλλά τις περισσότερες φορές είναι τουλάχιστον συνεχής.

Συνήθως μελετάμε τη συμπεριφορά ενός προβλήματος σε συγκεκριμένο σημείο $x \in X$ των δεδομένων (η συμπεριφορά μπορεί να αλλάζει από σημείο σε σημείο). Ένα πρόβλημα σε ένα συγκεκριμένο σημείο x το ονομάζουμε *πρόβλημα παράδειγμα*, όμως για λόγους απλοστευσης της χρησιμοποιούμενης ορολογίας χρησιμοποιούμε τον όρο πρόβλημα και σε αυτή την περίπτωση.

Ένα καλής κατάστασης πρόβλημα (παράδειγμα) λέμε αυτό που μικρές

διαταραχές στο x έχουν σαν αποτέλεσμα μικρές αλλαγές στο $f(x)$. Ένα κακής κατάστασης πρόβλημα λέμε αυτό που μικρές διαταραχές στο x έχουν σαν αποτέλεσμα μεγάλες αλλαγές στο $f(x)$. Η σημασία του μεγάλου και το μικρού εξαρτάται από την περίπτωση. Επίσης κάποιες φορές είναι προτιμότερο να μετράμε τις διαταραχές με έναν απόλυτο τρόπο και άλλες φορές με ένα σχετικό με τη νόρμα του στοιχείου που διαταράσσεται.

I.II Απόλυτος δείκτης κατάστασης

Έστω δx μια μικρή διαταραχή του x και $\delta f = f(x + \delta x) - f(x)$. Ο απόλυτος δείκτης κατάστασης $\hat{\kappa} = \hat{\kappa}(x)$ του προβλήματος f στο x ορίζεται ως εξής

$$\hat{\kappa} = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| \leq \delta} \frac{\|\delta f\|}{\|\delta x\|} \quad (1.1)$$

Στα περισσότερα προβλήματα το όριο του κατώτατου άνω φράγματος στην παραπάνω σχέση ουσιαστικά είναι το κατώτατο άνω φράγμα για όλες τις απειροελάχιστες διαταραχές δx και για να απλουστεύσουμε τη γραφή και τους συμβολισμούς μας θα γράφουμε

$$\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} \quad (1.2)$$

με τη σύμβαση ότι τα δx και δf είναι απειροελάχιστα.

Αν η f είναι διαφορίσιμη μπορούμε να υπολογίσουμε τον απόλυτο δείκτη κατάστασης με τη βοήθεια των παραγώγων της f . Έστω $J(x)$ ο πίνακας με στοιχεία στη θέση i, j τη μερική παράγωγο $\frac{\partial f_i}{\partial x_j}$ στο x , γνωστός και ως Ιακωβιανός πίνακας της f στο x . Τότε ο απόλυτος δείκτης κατάστασης είναι

$$\hat{\kappa} = \|J(x)\| \quad (1.3)$$

όπου $\|J(x)\|$ είναι η νόρμα του $J(x)$ που εξάγεται από τις νόρμες των X και Y .

I.III Σχετικός δείκτης κατάστασης

Όταν θέλουμε να μελετήσουμε τις σχετικές αλλαγές χρειαζόμαστε την έννοια της σχετικής κατάστασης. Ο σχετικός δείκτης κατάστασης $\kappa = \kappa(x)$ ορίζεται από τη σχέση

$$\kappa = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| \leq \delta} \left(\frac{\|\delta f\|}{\|f(x)\|} / \frac{\|\delta x\|}{\|x\|} \right) \quad (1.4)$$

Αν θεωρήσουμε όπως πριν ότι τα δx και δf είναι απειροελάχιστα, θα έχουμε

$$\kappa = \sup_{\delta x} \left(\frac{\|\delta f\|}{\|f(x)\|} / \frac{\|\delta x\|}{\|x\|} \right) \quad (1.5)$$

Αν η f είναι παραγωγίσιμη τότε μπορούμε να εκφράσουμε το σχετικό δείκτη κατάστασης με τη βοήθεια του Ιακωβιανού πίνακα

$$\kappa = \frac{\|J(x)\|}{\|f(x)\|/\|x\|} \quad (1.6)$$

Και οι δύο δείκτες που ορίσαμε παραπάνω έχουν τη χρησιμότητά τους όμως ο σχετικός δείκτης κατάστασης είναι αρκετά πιο σημαντικός στην αριθμητική ανάλυση. Ένα πρόβλημα λέγεται καλής κατάστασης αν το κ είναι μικρό (π.χ. 1, 10, 10^2) και κακής κατάστασης αν το κ είναι μεγάλο (π.χ. 10^6 , 10^{16}).

Παρακάτω θα παραθέσουμε κάποια χαρακτηριστικά παραδείγματα.

Παράδειγμα I.1. [1] παράδειγμα 12.1

Έστω το πρόβλημα υπολογισμού της \sqrt{x} για $x > 0$. Ο Ιακωβιανός πίνακας της $f : x \mapsto \sqrt{x}$ είναι η παράγωγος $J = f' = 1/(2\sqrt{x})$ έτσι έχουμε

$$\kappa = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{1/2}{(x/2)/x} = 1$$

Επομένως το πρόβλημα είναι καλής κατάστασης.

Παράδειγμα I.2. [1] παράδειγμα 12.3

Έστω το πρόβλημα υπολογισμού του βαθμωτού μεγέθους $f(x) = x_1 - x_2$ από το διάνυσμα $x = (x_1, x_2) \in \mathbb{C}^2$. Για λόγους απλούστευσης θα χρησιμοποιήσουμε την ∞ -νόρμα στον διανυσματικό χώρο \mathbb{C}^2 . Ο Ιακωβιανός πίνακας είναι

$$J = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} = [1 \quad -1]$$

με $\|J\|_\infty = 2$. Ο σχετικός δείκτης κατάστασης είναι

$$\kappa = \frac{\|J\|_\infty}{\|f(x)\|/\|x\|} = \frac{2}{|x_1 - x_2|/\max\{|x_1|, |x_2|\}}$$

Η παραπάνω ποσότητα είναι μεγάλη όταν $|x_1 - x_2| \approx 0$, έτσι το πρόβλημα είναι κακής κατάστασης όταν $x_1 \approx x_2$.

Το επόμενο παράδειγμα είναι χαρακτηριστικό και πολύ χρήσιμο για τα ζητήματα που θα αναπτύξουμε στα επόμενα κεφάλαια (τον υπολογισμό ιδιοτιμών).

Παράδειγμα I.3. [1] παράδειγμα 12.5

Η εύρεση των ριζών ενός πολυωνύμου μέσω των συντελεστών του είναι ένα

κλασικό πρόβλημα κακής κατάστασης. Έστω το $x^2 - 2x + 1 = (x - 1)^2$ που έχει διπλή ρίζα το $x = 1$. Μια μικρή διαταραχή στους συντελεστές μπορεί να έχει σαν αποτέλεσμα μεγάλες αλλαγές στις ρίζες. Για παράδειγμα $x^2 - 2x + 0.9999 = (x - 0.99)(x - 1.01)$. Στην πραγματικότητα οι ρίζες μπορούν να μεταβληθούν με ανάλογο τρόπο με την τετραγωνική ρίζα των μεταβολών των συντελεστών, έτσι σε αυτή την περίπτωση ο Ιακωβιανός πίνακας είναι άπειρος, επομένως $\kappa = \infty$.

Η εύρεση των ριζών ενός πολυωνύμου είναι συνήθως πρόβλημα κακής κατάστασης ακόμα και στις περιπτώσεις που δεν έχουμε πολλαπλότητα σε κάποια ρίζα. Έστω ο συντελεστής α_i του πολυωνύμου $\pi(x)$ και η απειροελάχιστη διαταραχή του συντελεστή $\delta\alpha_i$. Τότε η διαταραχή της j -οστής ρίζας x_j θα είναι $\delta x_j = (\delta\alpha_i)x_j^i / \pi'(x_j)$, όπου το π' είναι η παράγωγος του π . Ο σχετικός δείκτης κατάστασης του x_j ως προς τις διαταραχές του συντελεστή α_i θα είναι

$$\kappa = \frac{|\delta x_j|}{|x_j|} \bigg/ \frac{|\delta\alpha_i|}{\alpha_i} = \frac{|\alpha_i x_j^{i-1}|}{|\pi'(x_j)|}$$

Αυτός ο αριθμός είναι πολλές φορές πολύ μεγάλος. Για παράδειγμα έστω το πολυώνυμο του Wilkinson

$$p(x) = \prod_{i=1}^{20} (x - i) = \alpha_0 + \alpha_1 x + \dots + \alpha_{19} x^{19} + x^{20} \quad (1.7)$$

Η πιο ευαίσθητη σε διαταραχές ρίζα είναι η $x = 15$ για τον συντελεστή $\alpha_{15} \approx 1.67 \times 10^9$. Ο σχετικός δείκτης κατάστασης είναι

$$\kappa \approx \frac{1.67 \times 10^9 \cdot 15^{14}}{5! 14!} \approx 5.1 \times 10^{13}$$

Το σχήμα (1.1) απεικονίζει γραφικά την κακή κατάσταση του προβλήματος εύρεσης των ριζών του πολυωνύμου $p(x) = \prod_{i=1}^{20} (x - i)$.

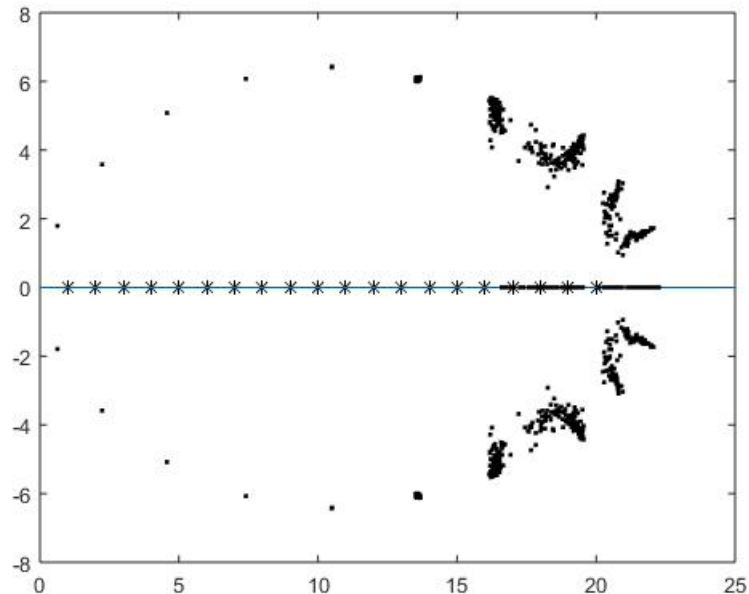
Επομένως όπως θα δούμε και στις επόμενες ενότητες πιο λεπτομερώς το να υπολογίσουμε ιδιοτιμές πινάκων μέσω των ριζών των χαρακτηριστικών τους πολυωνύμων δεν είναι πολύ καλή ιδέα καθώς για διαταραχές συντελεστών που είναι αδύνατον να μη συμβούν έχουμε πολύ μεγάλες μεταβολές στο αποτέλεσμα.

Παράδειγμα 1.4. [1] παράδειγμα 12.6

Το πρόβλημα υπολογισμού ιδιοτιμών ενός μη συμμετρικού πίνακα είναι πολλές φορές κακής κατάστασης. Αυτό μπορούμε να το διαπιστώσουμε υπολογίζοντας τις ιδιοτιμές των παρακάτω πινάκων

$$\begin{bmatrix} 1 & 1000 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1000 \\ 0.001 & 1 \end{bmatrix}$$

των οποίων οι ιδιοτιμές είναι $\{1, 1\}$ και $\{0, 2\}$ αντίστοιχα. Παρόλα αυτά αν ένας πίνακας A είναι συμμετρικός (αν είναι κανονικός γενικότερα) οι



Σχήμα 1.1: Το κλασικό παράδειγμα του Wilkinson. Τα '*' είναι οι ρίζες του πολυωνύμου 1.7. Οι τελείες είναι οι μιγαδικές ρίζες των 100 πολυωνύμων που προκύπτουν από τυχαίες διαταραχές του πολυωνύμου 1.7 όπου οι συντελεστές των πολυωνύμων είναι $\tilde{\alpha}_k = \alpha_k(1 + 10^{-10}r_k)$, όπου r_k αριθμοί που προέρχονται από την κατανομή $N(0, 1)$.

ιδιοτιμές του είναι καλής κατάστασης. Μπορεί ναδειχθεί ότι αν λ και $\lambda + \delta\lambda$ είναι οι ιδιοτιμές των πινάκων A και $A + \delta A$ αντίστοιχα, τότε $|\delta\lambda| \leq \|\delta A\|_2$ με την ισότητα να ισχύει αν ο δA είναι πολλαπλάσιο του ταυτοτικού πίνακα. Έτσι ο απόλυτος δείκτης κατάστασης μιας ιδιοτιμής ενός συμμετρικού πίνακα-προβλήματος ιδιοτιμών είναι $\hat{\kappa} = 1$ αν θεωρήσουμε σαν νόρμα των διαταραχών την 2-νόρμα, και ο σχετικός δείκτης κατάστασης είναι $\kappa = \|A\|_2/|\lambda|$.

I.IV Κατάσταση ενός πίνακα

Ο δείκτης κατάστασης ενός πίνακα είναι κάτι ιδιαίτερα σημαντικό και θεμελιακό για την αριθμητική ανάλυση. Έστωσαν ο πίνακας $A \in \mathbb{C}^{m \times n}$ και το πρόβλημα υπολογισμού του γινομένου Ax για το όρισμα x . Θα βρούμε τους δείκτες κατάστασης θεωρώντας διαταραχές του x αλλά όχι του πίνακα A . Παίρνοντας τον ορισμό του κ και θεωρώντας $\|\cdot\|$, μια αυθαίρετη νόρμα

διανυσμάτων και την αντίστοιχα εξαγόμενη νόρμα πινάκων, έχουμε

$$\kappa = \sup_{\delta x} \left(\frac{\|A(x + \delta x) - Ax\|}{\|Ax\|} \bigg/ \frac{\|\delta x\|}{\|x\|} \right) = \sup_{\delta x} \frac{\|Ax\|}{\|x\|}$$

και άρα

$$\kappa = \|A\| \frac{\|x\|}{\|Ax\|} \quad (1.8)$$

Το παραπάνω αποτέλεσμα είναι μια ακριβής σχέση ως προς τα A και x . Έστω ότι ο πίνακας παραπάνω είναι τετράγωνος και αντιστρέψιμος. Τότε χρησιμοποιώντας τη σχέση $\|x\|/\|Ax\| \leq \|A^{-1}\|$ μπορούμε να απλοποιήσουμε τη σχέση 1.8

$$\kappa \leq \|A\| \|A^{-1}\| \quad (1.9)$$

όπου καταλήξαμε να έχουμε άνω φράγμα ανεξάρτητο του x . Ισοδύναμα μπορούμε να γράψουμε

$$\kappa = \alpha \|A\| \|A^{-1}\| \quad (1.10)$$

όπου

$$\alpha = \frac{\|x\|}{\|Ax\|} \bigg/ \|A^{-1}\| \quad (1.11)$$

Για συγκεκριμένες επιλογές του x έχουμε $\alpha = 1$ και συνεπώς $\kappa = \|A\| \|A^{-1}\|$. Αν $\|\cdot\| = \|\cdot\|_2$ η σχέση $\kappa = \|A\| \|A^{-1}\|$ ισχύει όταν το x είναι πολλαπλάσιο του δεξιού ιδιάζοντος διανύσματος που αντιστοιχεί στη μικρότερη ιδιάζουσα τιμή του A .

Τι θα συμβεί αν θεωρήσουμε το αντίστροφο πρόβλημα: για δοθέντα πίνακα A να υπολογίσετε το $A^{-1}b$ με όρισμα το b ; Από μαθηματικής σκοπιάς το πρόβλημα αυτό ταυτίζεται με το προηγούμενο με τη διαφορά ότι αντί για τον πίνακα A έχουμε τον A^{-1} .

Θεώρημα 1.1. Έστωσαν $A \in \mathbb{C}^{m \times m}$ ένας αντιστρέψιμος πίνακας και η εξίσωση $Ax = b$. Το πρόβλημα υπολογισμού του x , δεδομένου του b , έχει σχετικό δείκτη κατάστασης

$$\kappa = \|A\| \frac{\|x\|}{\|b\|} \leq \|A\| \|A^{-1}\| \quad (1.12)$$

θεωρώντας διαταραχές του x . Το πρόβλημα υπολογισμού του x , δεδομένου του b , έχει σχετικό δείκτη κατάστασης

$$\kappa = \|A^{-1}\| \frac{\|b\|}{\|x\|} \leq \|A\| \|A^{-1}\| \quad (1.13)$$

θεωρώντας διαταραχές του b . Αν $\|\cdot\| = \|\cdot\|_2$ η ισότητα στη σχέση 1.12 ισχύει αν το x είναι πολλαπλάσιο του δεξιού ιδιάζοντος διανύσματος που αντιστοιχεί στη μικρότερη ιδιάζουσα τιμή σ_m του A και η ισότητα στη σχέση 1.13 ισχύει αν το b είναι πολλαπλάσιο του αριστερού ιδιάζοντος διανύσματος που αντιστοιχεί στη μεγαλύτερη ιδιάζουσα τιμή σ_1 .

Περισσότερες λεπτομέρειες για το θεώρημα I.1 στο [1] θεώρημα 12.1.

I.V Δείκτης κατάστασης ενός πίνακα

Το γινόμενο $\|A\| \|A^{-1}\|$ εμφανίζεται τόσο συχνά που έχει το δικό του όνομα: είναι ο δείκτης κατάστασης ενός πίνακα A και το συμβολίζουμε $\kappa(A)$

$$\kappa(A) = \|A\| \|A^{-1}\| \quad (1.14)$$

Επομένως σε αυτή την περίπτωση ο όρος δείκτης κατάστασης σχετίζεται με ένα πίνακα και όχι με κάποιο πρόβλημα. Αν το $\kappa(A)$ είναι μικρό λέμε ότι ο πίνακας A είναι καλής κατάστασης ενώ αν είναι μεγάλο λέμε ότι είναι κακής κατάστασης. Αν ο A είναι μη αντιστρέψιμος συνήθως γράφουμε $\kappa(A) = \infty$.

Σημειώνουμε ότι αν $\|\cdot\| = \|\cdot\|_2$ είναι $\|A\| = \sigma_1$ και $\|A^{-1}\| = 1/\sigma_m$. Οπότε

$$\kappa(A) = \frac{\sigma_1}{\sigma_m} \quad (1.15)$$

είναι ο τύπος που συνήθως χρησιμοποιούμε για να υπολογίσουμε το δείκτη κατάστασης ενός πίνακα με τη 2-νόρμα. Ο τύπος 1.15 είναι ιδιαίτερα χρήσιμος καθώς ισχύει και για μη τετραγωνικούς πίνακες $A \in \mathbb{C}^{m \times n}$ μέγιστου βαθμού και $m \geq n$ όταν χρησιμοποιούμε τη 2-νόρμα

$$\kappa(A) = \frac{\sigma_1}{\sigma_n} \quad (1.16)$$

Διαφορετικά δείκτης κατάστασης ορίζεται με τη βοήθεια του ψευδοαντίστροφου και είναι $\kappa(A) = \|A\| \|A^+\|$.

Θεώρημα I.2. Έστω το πρόβλημα υπολογισμού $x = A^{-1}b$ με δοσμένο b και όπου A ένας τετραγωνικός αντιστρέψιμος πίνακας. Ο σχετικός δείκτης κατάστασης του προβλήματος θεωρώντας διαταραχές του A είναι

$$\kappa = \|A\| \|A^{-1}\| = \kappa(A) \quad (1.17)$$

Περισσότερες λεπτομέρειες για το θεώρημα I.2 στο [1] θεώρημα 12.2.

Τα θεωρήματα I.1 και I.2 είναι θεμελιακά για την αριθμητική γραμμική άλγεβρα καθώς καθορίζουν με πόση ακρίβεια μπορεί να λύσει κάποιος γραμμικά συστήματα εξισώσεων. Αν ένα πρόβλημα $Ax = b$ περιέχει έναν κακής κατάστασης πίνακα A αναμένεται να έχουμε απώλεια $\log_{10}(\kappa(A))$ ψηφία.

II Αριθμητική κινητής υποδιαστολής

Η αριθμητική κινητής υποδιαστολής είναι η ψηφιακή αναπαράσταση των πραγματικών αριθμών σε μια ψηφιακή μηχανή (π.χ. έναν υπολογιστή).

II.1 Όρια ψηφιακής αναπαράστασης

Από τη στιγμή που μια ψηφιακή αναπαράσταση σε έναν υπολογιστή ενός πραγματικού αριθμού χρησιμοποιεί πεπερασμένο αριθμό ψηφίων είναι λογικό να μπορούμε να αναπαραστήσουμε μόνο ένα υποσύνολο των πραγματικών αριθμών (και των μιγαδικών). Αυτός ο περιορισμός παρουσιάζει δύο δυσκολίες. Η πρώτη είναι ότι οι αριθμητικές αναπαραστάσεις δεν μπορούν να είναι αυθαίρετα μεγάλοι ή μικροί. Και η δεύτερη ότι πρέπει να υπάρχουν κενά μεταξύ τους.

Οι σύγχρονοι υπολογιστές αναπαριστούν αριθμούς αρκετά μεγάλες και μικρές έτσι ώστε αυτοί οι περιορισμοί να μη δημιουργούν συνήθως δυσκολίες. Για παράδειγμα το ευρέως χρησιμοποιούμενο αριθμητικό πρωτόκολλο IEEE επιτρέπει την αναπαράσταση αριθμών μεγάλων όσο το 1.79×10^{308} και μικρών όσο ο 2.23×10^{-308} που είναι ένα εύρος αρκετά μεγάλο για τη τεράστια πλειοψηφία των προβλημάτων.

Σε αντίθεση με τα παραπάνω το πρόβλημα των κενών μεταξύ των αριθμητικών αναπαραστάσεων είναι μια μόνιμη ανησυχία στην επιστήμη των υπολογιστών. Για παράδειγμα στο πρωτόκολλο IEEE το διάστημα $[1, 2]$ αναπαριστάται με το διακριτό υποσύνολο

$$1, \quad , 1 + 2^{-52}, \quad , 1 + 2 \times 2^{-52}, \quad , 1 + 3 \times 2^{-52}, \dots, \quad 2 \quad (1.18)$$

Το διάστημα $[2, 4]$ αναπαριστάται από τους ίδιους αριθμούς πολλαπλασιασμένους με το 2

$$2, \quad , 2 + 2^{-5}, \quad , 2 + 2 \times 2^{-51}, \quad , 2 + 3 \times 2^{-51}, \dots, 4 \quad (1.19)$$

και γενικότερα το διάστημα $[2^i, 2^{i+1}]$ αναπαριστάται όπως στην 1.18 πολλαπλασιασμένη με 2^i . Έτσι στο πρωτόκολλο IEEE τα κενά μεταξύ δύο διαδοχικών αριθμών δεν είναι ποτέ μεγαλύτερο από $2^{-52} \approx 2.22 \times 10^{-16}$. Αυτό το κενό φαίνεται αμελητέο, και έτσι είναι αν χρησιμοποιήσουμε ευσταθείς αλγόριθμους, αλλά ακόμα και έτσι πολλοί είναι και οι αλγόριθμοι που αποδεικνύονται ασταθείς λόγω διαφόρων προβλημάτων.

II.II Αριθμοί κινητής υποδιαστολής

Το πρωτόκολλο IEEE είναι ένα παράδειγμα αριθμητικού συστήματος βασισμένο στην αναπαράσταση κινητής υποδιαστολής των πραγματικών αριθμών. Σε ένα σύστημα αριθμητικής αναπαράστασης κινητής υποδιαστολής το ακέραιο μέρος ενός αριθμού αποθηκεύεται σε διαφορετική θέση από το δεκαδικό του μέρος και τα κενά μεταξύ των διαδοχικών αριθμών καθορίζονται σε σχέση με το μέγεθος των αριθμών. Αυτό είναι διαφορετικό από ένα πλήρως καθορισμένο τρόπο αναπαράστασης όπου τα κενά θα έχουν παντού το ίδιο μέγεθος.

Πιο συγκεκριμένα θα θεωρήσουμε ένα ιδεατό σύστημα αριθμητικής αναπαράστασης κινητής υποδιαστολής. Το σύστημά μας αποτελείται από ένα διακριτό υποσύνολο \mathbb{F} των πραγματικών αριθμών \mathbb{R} που καθορίζεται από έναν ακέραιο $\beta \geq 2$, που τον λέμε βάση, και έναν ακέραιο $t \geq 1$ που τον λέμε ακρίβεια (στο πρωτόκολλο IEEE είναι το 24 ή το 53 για μονή ή διπλή ακρίβεια αντίστοιχα). Τα στοιχεία του \mathbb{F} θα είναι το 0 μαζί με όλους τους αριθμούς της μορφής

$$x = \pm(m/\beta^t)\beta^e \quad (1.20)$$

όπου ο m είναι ένας ακέραιος με $1 \leq m \leq \beta^t$ και ο e είναι ένας αυθαίρετα επιλεγμένος ακέραιος. Ισοδύναμα μπορούμε να περιορίσουμε το $\beta^{t-1} \leq m \leq \beta^t - 1$. Την ποσότητα $\pm(m/\beta^t)$ την λέμε μάντισσα του x και το e το λέμε εκθέτη του x . Βέβαια το ιδεατό σύστημα που ορίσαμε δεν μεριμνά καθόλου για τα ζητήματα της υπερχείλισης και της υποχείλισης καθώς είναι άπειρο. Αυτό δεν μπορεί να συμβεί στην πράξη.

II.III Το έφιλον μηχανής

Η ανάλυση του συνόλου \mathbb{F} παραδοσιακά συνοψίζεται με έναν αριθμό γνωστό ως έφιλον μηχανής, που θα συμβολίζουμε με $\epsilon_{machine}$. Αρχικά ας ορίσουμε τον αριθμό αυτό ως εξής

$$\epsilon_{machine} = \frac{1}{2}\beta^{1-t} \quad (1.21)$$

Αργότερα θα επανορίσουμε το $\epsilon_{machine}$. Ο παραπάνω αριθμός είναι η μισή απόσταση μεταξύ του 1 και του επόμενου μεγαλύτερου αριθμού κινητής υποδιαστολής. Γι' αυτό το $\epsilon_{machine}$ έχει την ακόλουθη ιδιότητα

$$\forall x \in \mathbb{R}, \exists x' \in \mathbb{F} : |x - x'| \leq \epsilon_{machine}|x| \quad (1.22)$$

Στο πρωτόκολλο IEEE το $\epsilon_{machine}$ είναι ίσο με $2^{-24} \approx 5.96 \times 10^{-8}$ για μονή ακρίβεια και ίσο με $2^{-53} \approx 1.11 \times 10^{-16}$.

Έστω $fl : \mathbb{R} \rightarrow \mathbb{F}$ μια συνάρτηση που δίνει για κάποιο πραγματικό αριθμό την πιο κοντινή αριθμητική αναπαράσταση στο σύστημα κινητής υποδιαστολής μας (σε περίπτωση ισαπεχουσών αριθμών επιλέγουμε αυθαίρετα). Η σχέση 1.22 μπορεί να γραφτεί ισοδύναμα με τη βοήθεια της fl

$$\forall x \in \mathbb{R}, \exists \epsilon \quad \mu\epsilon \quad |\epsilon| \leq \epsilon_{machine} : \quad fl(x) = x(1 + \epsilon) \quad (1.23)$$

Επομένως η διαφορά μεταξύ ενός πραγματικού αριθμού και της κοντινότερης σε αυτόν αριθμητικής αναπαράστασης στο σύστημα κινητής υποδιαστολής είναι πάντα με σχετικό τρόπο μικρότερη από το $\epsilon_{machine}$.

II.IV Αριθμητική κινητής υποδιαστολής

Η αναπαράσταση των πραγματικών αριθμών όπως την περιγράψαμε στις προηγούμενες υποενότητες δεν είναι αρκετή από τη στιγμή που πρέπει να κάνουμε πράξεις και υπολογισμούς με βάση αυτές τις αναπαραστάσεις. Στον υπολογιστή όλοι οι μαθηματικοί υπολογισμοί ανάγονται σε στοιχειώδης αριθμητικές πράξεις όπως είναι οι $+$, $-$, \times , $/$. Από μαθηματική σκοπιά αυτά τα σύμβολα αντιπροσωπεύουν πράξεις στο \mathbb{R} . Στον υπολογιστή υπάρχουν ανάλογες πράξεις στο \mathbb{F} για τις οποίες συνήθως χρησιμοποιούμε τα σύμβολα \oplus , \ominus , \otimes , \oslash

Έστωσαν $x, y \in \mathbb{F}$, δηλαδή δύο τυχαίοι αριθμοί κινητής υποδιαστολής και \cdot μία από τις πράξεις $+$, $-$, \times , $/$ και \odot η αντίστοιχη πράξη στο σύστημα κινητής υποδιαστολής. Τότε η πράξη $x \odot y$ ορίζεται από την παρακάτω σχέση

$$x \odot y = fl(x \cdot y) \quad (1.24)$$

Αν ισχύει η παραπάνω σχέση σε έναν υπολογιστή τότε από τις 1.23 και 1.24 συμπεραίνουμε ότι ο υπολογιστής ικανοποιεί το παρακάτω πολύ σημαντικό αξίωμα

Θεμελιώδες αξίωμα της αριθμητικής κινητής υποδιαστολής

$$\forall x, y \in \mathbb{F}, \exists \epsilon \quad \mu\epsilon \quad |\epsilon| \leq \epsilon_{machine} : \quad x \odot y = (x \cdot y)(1 + \epsilon) \quad (1.25)$$

Το παραπάνω αξίωμα ουσιαστικά δηλώνει ότι κάθε πράξη στην αριθμητική κινητής υποδιαστολής έχει σχετικό σφάλμα το πολύ $\epsilon_{machine}$.

II.V Σχετικά με το έψιλον μηχανής

Η διαδικασία στρογγυλοποίησης των σφαλμάτων μετά από υπολογισμούς, πράξεις κτλ. στις ενότητες που θα ακολουθήσουν γίνεται με βάση τις σχέσεις 1.23, 1.25 και όχι αυτά που περιγράφηκαν πιο πάνω. Αυτό σημαίνει ότι έχουμε την πολυτέλεια στην υλοποίηση μηχανής να μην έχουμε τόσο τέλεια απόδοση στους υπολογισμούς κινητής υποδιαστολής όπως ορίζει η σχέση 1.24. Σε μια τέτοια περίπτωση μηχανής οι σχέσεις 1.23, 1.25 θα ικανοποιούνται επίσης από μια μεγαλύτερη τιμή του $\epsilon_{machine}$. Για παράδειγμα αν σε έναν υπολογιστή οι ενδιάμεσες ποσότητες δεν στρογγυλοποιούνται αλλά περικλύπτονται τότε η σχέση 1.25 ικανοποιείται με το $2\epsilon_{machine}$.

Ο πιο απλός τρόπος να δώσουμε μια τέτοια ευελιξία και να επιτρέψουμε την ύπαρξη τέτοιας πολυπλοκότητας είναι να κρατήσουμε τα αξιώματα 1.23, 1.25 και να αλλάξουμε τον ορισμό του $\epsilon_{machine}$. Από εδώ και στο εξής θα θεωρούμε το $\epsilon_{machine}$ ως το μικρότερο αριθμό για τον οποίο ισχύουν οι σχέσεις 1.23, 1.25. Για τους περισσότερους υπολογιστές, συμπεριλαμβανομένων και αυτών που ακολουθούν το πρωτόκολλο IEEE, αυτή η αλλαγή στον ορισμό του $\epsilon_{machine}$ δεν επιφέρει κάποιες καθοριστικές αλλαγές στην τιμή του.

III Ευστάθεια

Το καλύτερο ενδεχόμενο θα ήταν να μπορούσαν οι αριθμητικοί αλγόριθμοι να μας δώσουν ακριβείς λύσεις στα αριθμητικά προβλήματα. Αφού όμως τα προβλήματα είναι συνεχή και οι ψηφιακοί υπολογιστές διακριτοί αυτό το ενδεχόμενο δεν είναι πιθανό. Η έννοια της ευστάθειας είναι ένας τρόπος να εκφράσουμε τι είναι πιθανό. Δηλαδή ένας τρόπος να πούμε μέσω της αριθμητικής ανάλυσης τι σημαίνει "σωστή απάντηση", ακόμα και αν δεν είναι ακριβής.

III.I Αλγόριθμοι

Στην ενότητα I.I ορίσαμε ως μαθηματικό πρόβλημα μια συνάρτηση $f : X \rightarrow Y$ από ένα διανυσματικό χώρο δεδομένων X σε ένα διανυσματικό χώρο λύσεων Y .

Ένας αλγόριθμος μπορεί να αντιμετωπιστεί επίσης ως μια συνάρτηση $\tilde{f} : X \rightarrow Y$ μεταξύ των ίδιων χώρων. Για να γίνουμε πιο ακριβείς, έστωσαν ένα πρόβλημα f , ένας υπολογιστής που ικανοποιεί την 1.25 (και όχι αναγκαστικά την 1.24), ένας αλγόριθμος για το πρόβλημα f (με τη χαλαρή έννοια του όρου) και μια υλοποίηση του αλγορίθμου στη μορφή ενός προγράμματος υπολογιστή. Έστω τώρα ότι το $x \in X$ στρογγυλοποιήθηκε ώστε να αναπαρασταθεί σαν αριθμός κινητής υποδιαστολής με τρόπο που ικανοποιεί τη σχέση 1.23 και ύστερα δόθηκε σαν όρισμα στο πρόγραμμα του υπολογιστή. Αν τρέξουμε το πρόβλημα το αποτέλεσμα θα είναι ένα σύνολο αριθμών κινητής υποδιαστολής που ανήκουν στο διανυσματικό χώρο Y (αφού το πρόγραμμα κατασκευάστηκε για να λύσει το πρόβλημα f). Ας ονομάσουμε αυτά τα υπολογισθέντα αποτελέσματα $\tilde{f}(x)$.

Αρχικά διαπιστώνουμε ότι η κατάσταση μάλλον έχει αρκετά προβλήματα. Το λιγότερο θα μπορούσε να είναι ότι το $\tilde{f}(x)$ θα έχει επηρεαστεί από στρογγυλοποίηση σφαλμάτων. Επίσης θα μπορούσε να έχει επηρεαστεί και από άλλες επιπλοκές όπως η ακρίβεια σύγκλισης ή άλλες διαδικασίες που τρέχουν στον υπολογιστή. Έτσι η $\tilde{f}(x)$ μπορεί να είναι πλειότιμη, με την έννοια ότι μπορεί να παίρνει διαφορετικές τιμές σε κάθε εκτέλεση του προγράμματος. Παρ' όλα αυτά θα δούμε ότι μπορούμε να ξεπεράσουμε αυτές τις επιπλοκές και να κάνουμε κάποιες ξεκάθαρες δηλώσεις για την $\tilde{f}(x)$ όπως για παράδειγμα για την ακρίβεια των αλγορίθμων της αριθμητικής γραμμικής άλγεβρας.

III.II Ακρίβεια

Εκτός από τετριμμένες περιπτώσεις, η \tilde{f} δεν μπορεί να είναι συνεχής. Παρ' όλα αυτά ένας καλός αλγόριθμος πρέπει να προσεγγίζει με ικανοποιητικό τρόπο το αντίστοιχο πρόβλημα f . Για να μπορέσουμε να εκφράσουμε με έναν ποσοτικό τρόπο την προηγούμενη πρόταση θα θεωρήσουμε το απόλυτο σφάλμα ενός υπολογισμού $\|\tilde{f}(x) - f(x)\|$ και το σχετικό σφάλμα

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \quad (1.26)$$

Συνήθως το σχετικό σφάλμα είναι πιο σημαντικό γι' αυτό και εμείς θα χρησιμοποιήσουμε το σχετικό σφάλμα για να μετράμε τα σφάλματα.

Αν ο \tilde{f} είναι καλός αλγόριθμος θα περιμέναμε το σχετικό σφάλμα να είναι αρκετά μικρό, στην τάξη του $\epsilon_{machine}$.

Ορισμός III.1. Έστω ένας αλγόριθμος \tilde{f} για το πρόβλημα f . Λέμε ότι ο αλγόριθμος \tilde{f} είναι ακριβής αν για κάθε $x \in X$ ισχύει

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\epsilon_{machine}) \quad (1.27)$$

Ο συμβολισμός $O(\epsilon_{machine})$ στην παραπάνω σχέση έχει την έννοια "τάξης έψιλον μηχανής". Παρ' όλα αυτά το $O(\epsilon_{machine})$ έχει επίσης και μια ακριβή σημασία που θα δούμε αργότερα.

III.III Ευστάθεια

Αν το πρόβλημα f είναι κακής κατάστασης ο στόχος της ακρίβειας όπως ορίστηκε στη σχέση 1.27 είναι υπερβολικά φιλόδοξος. Η αναπόφευκτη στρογγυλοποίηση του ορίσματος στον ψηφιακό υπολογιστή (ακόμα και αν όλες οι υποδιαδικασίες υπολογισμών γίνουν με τέλειο τρόπο) είναι μια διαταραχή που θα οδηγούσε σε μια σημαντική μεταβολή του αποτελέσματος. Αντί να έχουμε ως στόχο την ακρίβεια σε κάθε περίπτωση είναι πιο λογικό να βάλουμε ένα γενικότερο στόχο, αυτόν της ευστάθειας.

Ορισμός III.2. Έστω ένας αλγόριθμος \tilde{f} για το πρόβλημα f . Λέμε ότι ο αλγόριθμος \tilde{f} είναι ευσταθής αν για κάθε $x \in X$ ισχύει

$$\frac{\|\tilde{f}(x) - f(\tilde{x})\|}{\|f(\tilde{x})\|} = O(\epsilon_{machine}) \quad (1.28)$$

για κάποιο \tilde{x} με

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{\text{machine}}) \quad (1.29)$$

Με απλά λόγια θα μπορούσαμε να πούμε:

Ένας ευσταθής αλγόριθμος μας δίνει σχεδόν τη σωστή απάντηση σε ένα σχεδόν σωστό ερώτημα

III.IV Προς τα πίσω ευστάθεια

Πολλοί αλγόριθμοι στην αριθμητική γραμμική άλγεβρα ικανοποιούν μια συνθήκη που είναι και πιο ισχυρή και πιο απλή από τη συνθήκη της ευστάθειας.

Ορισμός III.3. Έστω ένας αλγόριθμος \tilde{f} για το πρόβλημα f . Λέμε ότι ο αλγόριθμος \tilde{f} είναι προς τα πίσω ευσταθής αν για κάθε $x \in X$ ισχύει

$$\tilde{f}(x) = f(\tilde{x}), \quad \text{για κάποιο } \tilde{x} \text{ με } \frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{\text{machine}}) \quad (1.30)$$

Πρόκειται για μια πιο αυστηρή εκδοχή της σχέσης 1.28 όπου το $O(\epsilon_{\text{machine}})$ έχει αντικατασταθεί με το μηδέν. Με απλά λόγια θα μπορούσαμε να πούμε:

Ένας προς τα πίσω ευσταθής αλγόριθμος μας δίνει ακριβώς τη σωστή απάντηση σε ένα σχεδόν σωστό ερώτημα

III.V Η σημασία του $O(\epsilon_{\text{machine}})$

Σε αυτή την υποενότητα θα δώσουμε κάποιες παραπάνω διευκρινήσεις για το $\epsilon_{\text{machine}}$.

Ορισμός III.4. Έστωσαν οι πραγματικές συναρτήσεις $\varphi(t)$ και $\psi(t)$ και η θετική σταθερά C τέτοια ώστε για όλα τα t αρκετά κοντά σε κάποιο όριο (π.χ. $t \rightarrow 0$ ή $t \rightarrow \infty$) να ισχύει

$$|\varphi(t)| \leq C\psi(t) \quad (1.31)$$

Τότε γράφουμε

$$\varphi(t) = O(\psi(t)) \quad (1.32)$$

Για παράδειγμα η σχέση $\sin t^2 = O(t^2)$, καθώς $t \rightarrow 0$ σημαίνει πως υπάρχει μια σταθερά C η οποία για όλα τα αρκετά μικρά t ικανοποιεί τη σχέση $|\sin^2 t| \leq Ct^2$.

Επίσης άλλη μία χαρακτηριστική σχέση είναι η

$$\varphi(s, t) = O(\psi(t)) \quad \text{ομοιόμορφα ως προς } s \quad (1.33)$$

Όταν λέμε ομοιόμορφα εννοούμε ότι υπάρχει μια σταθερά C όπως στην 1.31 που ικανοποιεί τη συνθήκη για όλα τα $s \in \mathbb{R}$.

Σε αυτή την εργασία όταν θα χρησιμοποιούμε το σύμβολο O θα το κάνουμε με τη σημασία του ορισμού III.4. Πιο ειδικά συχνά παρουσιάζουμε τα αποτελέσματα κάποιου αλγορίθμου ως εξής

$$\| \text{υπολογισμένη ποσότητα} \| = O(\epsilon_{\text{machine}}) \quad (1.34)$$

Ο όρος $\| \text{υπολογισμένη ποσότητα} \|$ δηλώνει τον υπολογισμό της νόρμας κάποιου αριθμού ή σύνολο αριθμών που καθορίζονται από κάποιον αλγόριθμο \tilde{f} για κάποιο πρόβλημα f και εξαρτάται από τα $x \in X$ και από το $\epsilon_{\text{machine}}$. Επίσης το όριο που υπαινίσσεται η παραπάνω συμβολική σχέση είναι το $\epsilon_{\text{machine}} \rightarrow 0$ (είναι η μεταβλητή που αντιστοιχεί στο t στη σχέση 1.33. Τέλος το "O" ισχύει ομοιόμορφα ως προς το $x \in X$ (το x είναι η μεταβλητή που αντιστοιχεί στο s). Δεν θα το αναφέρουμε αλλά πάντα θα εννοούμε την ομοιομορφία ως προς το $x \in X$.

Σε κάθε μηχανή αριθμητικών υπολογισμών ο αριθμός $\epsilon_{\text{machine}}$ είναι ένας σταθερός αριθμός. Μιλώντας για το όριο $\epsilon_{\text{machine}} \rightarrow 0$ θεωρούμε έναν ιδανικό υπολογιστή ή για την ακρίβεια μια οικογένεια ιδανικών υπολογιστών. Η σχέση 1.34 σημαίνει ότι αν τρέξουμε έναν αλγόριθμο σε έναν υπολογιστή που ικανοποιεί τα αξιώματα 1.23, 1.25 και για μια ακολουθία τιμών $\epsilon_{\text{machine}}$ που τείνει στο μηδέν τότε η $\| \text{υπολογισμένη ποσότητα} \|$ σίγουρα θα μειώνεται αναλογικά με το $\epsilon_{\text{machine}}$ ή πιο γρήγορα. Αυτοί οι ιδανικοί υπολογιστές αρκεί να ικανοποιούν τα αξιώματα 1.23, 1.25 και τίποτα άλλο.

Οι ορισμοί που δώσαμε και περιλαμβάνουν το $O(\epsilon_{\text{machine}})$ έχουν τη σημαντική ιδιότητα ότι είναι ανεξάρτητοι από τη νόρμα που επιλέγουμε (δεδομένου ότι οι διανυσματικοί χώροι X και Y είναι πεπερασμένης διάστασης).

Θεώρημα III.1. Έστω ένα τυχαίο πρόβλημα f και ο αλγόριθμός του \tilde{f} που ορίζονται σε πεπερασμένης διάστασης διανυσματικούς χώρους X και Y . Τότε οι ιδιότητες της ακρίβειας, της ευστάθειας και της προς τα πίσω ευστάθειας ισχύουν ή δεν ισχύουν ανεξάρτητα από την επιλογή των νορμών των διανυσματικών χώρων X και Y .

Απόδειξη. [1] θεώρημα 14.1

Όπως γνωρίζουμε όλες οι νόρμες σε έναν πεπερασμένης διάστασης διανυσματικό χώρο είναι ισοδύναμες. Δηλαδή έστω ότι οι $\| \cdot \|$ και $\| \cdot \|'$ είναι δύο νόρμες στον ίδιο διανυσματικό χώρο, τότε υπάρχουν θετικές σταθερές C_1 και C_2 τέτοιες ώστε να ισχύει $C_1 \|x\| \leq \|x\|' \leq C_2 \|x\|$ για όλα τα x που ανήκουν σε αυτό το διανυσματικό χώρο. Άμεση συνέπεια αυτού είναι ότι μια αλλαγή στις νόρμες μπορεί να επιφέρει αλλαγή στην τιμή της σταθεράς C αλλά όχι να ακυρώσει την ύπαρξή της. \square

III.VI Ευστάθεια στην αριθμητική κινητής υποδιαστολής

Τα τέσσερα πιο απλά προβλήματα υπολογισμού είναι τα $+$, $-$, \times , $/$. Η επιλογή των αντίστοιχων αλγορίθμων είναι αυτονόητη. Προφανώς θα χρησιμοποιήσουμε τις πράξεις κινητής υποδιαστολής \oplus , \ominus , \otimes , \oslash . Βέβαια τα αξιώματα 1.23, 1.25 υπονοούν ότι αυτές οι τέσσερις πράξεις είναι προς τα πίσω ευσταθείς, το οποίο θα δείξουμε για την αφαίρεση παρακάτω.

Στο παράδειγμα 1.2 ο χώρος δεδομένων X είναι το \mathbb{C}^2 και ο χώρος των λύσεων είναι το \mathbb{C} . Δεν χρειάζεται να συγκεκριμενοποιήσουμε τις νόρμες βάση του θεωρήματος III.1. Για $x = (x_1, x_2) \in X$ το πρόβλημα της αφαίρεσης αντιστοιχεί στη συνάρτηση $f(x_1, x_2) = x_1 - x_2$ και ο αλγόριθμος που θεωρούμε είναι ο

$$\tilde{f}(x_1, x_2) = fl(x_1) \ominus fl(x_2)$$

Η παραπάνω εξίσωση σημαίνει ότι πρώτα στρογγυλοποιούμε τα x_1, x_2 στις τιμές κινητής υποδιαστολής και μετά εφαρμόζουμε την πράξη \ominus . Λόγω της 1.23 έχουμε

$$fl(x_1) = x_1(1 + \epsilon_1), \quad fl(x_2) = x_2(1 + \epsilon_2)$$

για κάποια $|\epsilon_1|, |\epsilon_2| \leq \epsilon_{machine}$. Από την 1.25 έχουμε

$$\begin{aligned} fl(x_1) \ominus fl(x_2) &= [x_1(1 + \epsilon_1) - x_2(1 + \epsilon_2)](1 + \epsilon_3) \\ &= x_1(1 + \epsilon_1)(1 + \epsilon_3) - x_2(1 + \epsilon_2)(1 + \epsilon_3) \\ &= x_1(1 + \epsilon_4) - x_2(1 + \epsilon_5) \end{aligned}$$

για κάποια $|\epsilon_4|, |\epsilon_5| \leq 2\epsilon_{machine} + O(\epsilon_{machine}^2)$. Με άλλα λόγια το αποτέλεσμα $\tilde{f}(x) = fl(x_1) \ominus fl(x_2)$ είναι ακριβώς το ίδιο με τη διαφορά $\tilde{x}_1 - \tilde{x}_2$ όπου τα \tilde{x}_1, \tilde{x}_2 ικανοποιούν τις σχέσεις

$$\frac{|\tilde{x}_1 - x_1|}{|x_1|} = O(\epsilon_{machine}), \quad \frac{|\tilde{x}_2 - x_2|}{|x_2|} = O(\epsilon_{machine})$$

και οποιοδήποτε $C > 2$.

Όλες οι υπόλοιπες πράξεις στην αριθμητική κινητής υποδιαστολής είναι προς τα πίσω ευσταθείς και δείχνονται με παρόμοιο τρόπο.

III.VII Ένας ασταθής αλγόριθμος

Υπάρχουν διάφορα παραδείγματα ευσταθών και ασταθών αλγορίθμων. Αυτός που θα εξετάσουμε εδώ είναι αρκετά ουσιώδης, και έχει μεγάλη σχέση με τα επόμενα κεφάλαια: ο αλγόριθμος που χρησιμοποιεί το χαρακτηριστικό

πολυώνυμο ενός πίνακα για να βρει τις ιδιοτιμές του. Έστω $\pi(x)$ το χαρακτηριστικό πολυώνυμο του πίνακα A . Αν το z είναι ρίζα του πολυωνύμου τότε θα είναι και ιδιοτιμή του πίνακα A . Άρα ορίζεται η εξής μέθοδος:

1. Βρίσκουμε τους συντελεστές του χαρακτηριστικού πολυωνύμου.
2. Βρίσκουμε τις ρίζες του.

Αυτός ο αλγόριθμος όχι μόνο δεν είναι προς τα πίσω ευσταθής αλλά δεν θα πρέπει να χρησιμοποιείται καθόλου. Ακόμα και σε περιπτώσεις όπου το πρόβλημα εύρεσης ιδιοτιμών είναι καλής κατάστασης μπορεί να μας δώσει αποτελέσματα με πολύ μεγαλύτερο σφάλμα από το $\epsilon_{machine}$.

Η αστάθεια πηγάζει από το δεύτερο βήμα. Όπως είδαμε στο παράδειγμα 1.3 το πρόβλημα της εύρεσης των ριζών του πολυωνύμου μέσω των συντελεστών του είναι γενικά ένα πρόβλημα κακής κατάστασης. Αυτό σημαίνει ότι μικρές διαταραχές στους συντελεστές οδηγούν σε μεγάλες μεταβολές στις υπολογισμένες ρίζες.

Για παράδειγμα έστω ο $A = I$, ο ταυτοτικός πίνακας 2×2 . Το πρόβλημα της εύρεσης των ιδιοτιμών του σε αυτή την περίπτωση είναι καλής κατάστασης και ένας ευσταθής αλγόριθμος θα έβρισκε τις ιδιοτιμές του με σφάλματα $O(\epsilon_{machine})$. Όμως ο παραπάνω αλγόριθμος θα μας έδινε το αποτέλεσμα με σφάλματα $O(\sqrt{\epsilon_{machine}})$. Για παράδειγμα αν $\epsilon_{machine} = 10^{-16}$ οι ρίζες θα είχαν ακρίβεια της τάξης 10^{-8} . Μια τέτοια περίπτωση είδαμε στο παράδειγμα 1.3. Αν υπολογίσουμε τις ιδιοτιμές του 2×2 ταυτοτικού πίνακα δεν θα έχουμε καθόλου σφάλματα καθώς αφού οι συντελεστές του $x^2 - 2x + 1$ είναι μικροί ακέραιοι και μπορούν να αναπαρασταθούν επακριβώς. Αυτό βλέπουμε να επαληθεύεται σε μια απλή υλοποίηση στο matlab.

```
>> syms x
>> g = symfun(x^2 - 2 * x + 1, x);
>> eval(vpa(solve(g == 0)))
ans =
```

```
1
1
```

Παρόλα αυτά αν υπολογίσουμε με τον ίδιο αλγόριθμο τις ιδιοτιμές του ταυτοτικού με μικρές διαταραχές

$$A = \begin{bmatrix} 1 + 10^{-14} & 0 \\ 0 & 1 \end{bmatrix}$$

Θα πρέπει να περιμένουμε για τις ιδιοτιμές που θα υπολογίσουμε σφάλματα τάξης $\epsilon_{machine}$. Με μια απλή υλοποίηση στο matlab

```
>> f = symfun(x^2 - (2 + 10^{-14}) * x + 1 + 10^{-14}, x);
>> roots = eval(vpa(solve(f == 0)));
>> roots
```

```
1.0000000000000000e + 00 - 1.0000000000000000e - 07i
1.0000000000000000e + 00 + 1.0000000000000000e - 07i
```

```
>> sqrt(eps())
ans =
```

```
1.490116119384766e - 08
```

Βλέπουμε ότι τα σφάλματα που παίρνουμε είναι της τάξης του $\sqrt{\epsilon_{machine}}$.

Μέχρι τις αρχές τις δεκαετίας του '90 ο αλγόριθμος εύρεσης των ιδιοτιμών μέσω του χαρακτηριστικού πολυώνυμου χρησιμοποιούνταν αρκετά συχνά, ειδικά για τους μικρούς πίνακες διάστασης μέχρι 4×4 ή 5×5 . Για τον υπολογισμό των συντελεστών του χαρακτηριστικού πολυωνύμου υπάρχουν αρκετές μέθοδοι όμως αυτή που χρησιμοποιούνταν πιο συχνά ήταν η μέθοδος FADDEEV-LEVERRIER. Σε αυτή τη μέθοδο οι συντελεστές α_i υπολογίζονται από τα ίχνη μιας ακολουθίας πινάκων B_i .

Η δημιουργία της ακολουθίας και ο υπολογισμός των συντελεστών γίνεται με βάση τα παρακάτω βήματα, αν A είναι ο αρχικός μας πίνακας του οποίου θέλουμε να προσδιορίσουμε το χαρακτηριστικό πολυώνυμο.

$$B_1 = A \rightarrow \alpha_1 = tr(B_1)$$

$$B_k = A(B_{k-1} - \alpha_{k-1}I) \rightarrow \alpha_k = \frac{tr(B_k)}{k}$$

Για $k = 2, \dots, m$.

Η υλοποίηση αυτού του αλγόριθμου στο MATLAB φαίνεται παρακάτω

Εύρεση συντελεστών χαρακτηριστικού πολυωνύμου

Μέθοδος FADDEEV-LEVERRIER

Υλοποίηση στο MATLAB

```
function [pol]=fad_lev(a)
m=length(a); pol=[-1]; B=a;
for i=1:m
    p=trace(B)/i;
    pol=[pol p];
    B=a*(B-p*eye(m));
end
end
```

Για περισσότερες λεπτομέρειες πάνω στη μέθοδο FADDEEV-LEVERRIER και στον υπολογισμό ιδιοτιμών μέσω του χαρακτηριστικού πολυωνύμου στο [3] κεφάλαιο V, υποενότητα 3.1.

III.VIII Ακρίβεια προς τα πίσω ευσταθών αλγορίθμων

Έστω ο προς τα πίσω ευσταθής αλγόριθμος \tilde{f} του προβλήματος $f : X \rightarrow Y$. Ένα αυθόρμητο ερώτημα είναι αν αυτός ο αλγόριθμος είναι ακριβής. Η απάντηση εξαρτάται από το σχετικό δείκτη κατάστασης $\kappa = \kappa(x)$ του προβλήματος f . Αν το $\kappa(x)$ είναι μικρό, τα αποτελέσματα θα είναι σχετικά ακριβή, ενώ αν είναι μεγάλο τα αποτελέσματα θα έχουν ανάλογες μεταβολές.

Θεώρημα III.2. Έστω ότι χρησιμοποιούμε τον προς τα πίσω ευσταθή αλγόριθμο \tilde{f} για να λύσουμε το πρόβλημα $f : X \rightarrow Y$ με σχετικό δείκτη κατάστασης κ , σε έναν υπολογιστή που ικανοποιεί τα αξιώματα 1.23 και 1.25. Τότε για το σχετικό σφάλμα ισχύει

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\kappa(x)\epsilon_{machine}) \quad (1.35)$$

Απόδειξη. [1] θεώρημα 15.1

Από τον ορισμό III.3 της προς τα πίσω ευστάθειας θα έχουμε ότι $\tilde{f}(x) = f(\tilde{x})$ για κάποιο $\tilde{x} \in X$ με

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{machine})$$

Άρα με βάση το παραπάνω και τον ορισμό του $\kappa(x)$, έχουμε

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq (\kappa(x) + o(1)) \frac{\|\tilde{x} - x\|}{\|x\|} \quad (1.36)$$

Όπου με το $o(1)$ εννοούμε μια ποσότητα που συγκλίνει στο μηδέν καθώς $\epsilon_{machine} \rightarrow 0$. Συνδυάζοντας τις δύο τελευταίες σχέσεις προκύπτει η 1.35. \square

Η διαδικασία που ακολουθήσαμε για να αποδείξουμε το θεώρημα III.2 είναι γνωστή ως προς τα πίσω ανάλυση σφάλματος (backward error analysis). Βρίσκουμε την ακρίβεια μιας προσέγγισης με δύο βήματα. Το πρώτο βήμα είναι η ανάλυση της κατάστασης του προβλήματος και το δεύτερο είναι η ανάλυση της ευστάθειας του αλγόριθμου. Το συμπέρασμά μας είναι ότι αν κάποιος αλγόριθμος είναι ευσταθής τότε η τελική ακρίβεια αντανακλά το σχετικό δείκτη κατάστασης του προβλήματος.

Αυτή η διαδικασία είναι διαφορετική από την προς τα εμπρός ανάλυση σφάλματος, η οποία είναι ένας διαφορετικός τρόπος προσέγγισης του ζητήματος της ακρίβειας ενός αλγορίθμου. Σε αυτή τη διαδικασία θεωρούμε τα σφάλματα στρογγυλοποίησης που εισάγονται σε κάθε βήμα των υπολογισμών και με κάποιο τρόπο υπολογίζουμε το συνολικό σφάλμα.

Εν κατακλείδι, επειδή οι καλύτεροι αλγόριθμοι για τα περισσότερα προβλήματα είναι οι προς τα πίσω ευσταθείς, η προς τα πίσω ανάλυση σφάλματος θεωρείται η καταλληλότερη μέθοδος εύρεσης της ακρίβειας ενός τέτοιου αλγορίθμου. Γι' αυτό και εμείς θα επικεντρωθούμε σε αυτή από δω και πέρα.

III.IX Ευστάθεια της Householder τριγωνοποίησης

Η Householder τριγωνοποίηση είναι ένας προς τα πίσω ευσταθής αλγόριθμος για να υπολογίσουμε την παραγοντοποίηση QR. Αυτός ο ισχυρισμός είναι πολύ βασικός για την παρούσα εργασία. Την παραγοντοποίηση QR τη θεωρούμε θεμελιώδες εργαλείο στους αλγόριθμους που θα αναπτύξουμε στις επόμενες ενότητες, ενώ Householder πίνακες θα χρησιμοποιήσουμε και για τη δημιουργία πινάκων Hessenberg ως απαραίτητο βήμα, σε αρκετές περιπτώσεις, για τον υπολογισμό ιδιοτιμών. Επομένως η εξέταση της ευστάθειας του αλγορίθμου της Householder τριγωνοποίησης είναι πολύ σημαντική. Στη συνέχεια πριν δώσουμε το καθοριστικό θεώρημα θα κάνουμε ένα είδους πείραμα στο matlab με βάση το πρωτόκολλο IEEE και $\epsilon_{machine} = 2.220446049250313 \times 10^{-16}$.

» format short	Θέτουμε μονή αριθμητική ακρίβεια.
» R=triu(randn(50));	Δημιουργούμε έναν 50×50 άνω-τριγωνικό πίνακα με στοιχεία που προέρχονται από την $N(0,1)$.
» [Q,X]=qr(randn(50));	Δημιουργούμε ένα τυχαίο ορθογώνιο πίνακα Q με στοιχεία που προέρχονται από την $N(0,1)$.
» format long	Θέτουμε διπλή αριθμητική ακρίβεια.
» A=Q*R;	Θέτουμε τον πίνακα A ως το γινόμενο QR με σφάλματα στρογγυλοποίησης.
» format short	Θέτουμε μονή αριθμητική ακρίβεια.
» [Q2,R2]=qr(A);	Υπολογίζουμε την παραγοντοποίηση QR , $A \approx Q_2 R_2$ με τον αλγόριθμο της Householder τριγωνοποίησης.

Ο στόχος των παραπάνω εντολών στη Matlab είναι να κατασκευάσουμε

ένα πίνακα με γνωστή QR παραγοντοποίηση, $A = QR$, ώστε να τη συγκρίνουμε με την QR παραγοντοποίηση, $A = Q_2R_2$, που υπολογίζει ο αλγόριθμος της Householder τριγωνοποίησης. Για την ακρίβεια οι παράγοντες QR δεν είναι ακριβώς οι Q και R , λόγω των σφαλμάτων στρογγυλοποίησης, έτσι θα μπορούσαμε να πούμε ότι αυτό το γεγονός θα μας δημιουργήσει πρόβλημα όταν θα συγκρίνουμε τους πίνακες (στην πραγματικότητα δεν θα μας δημιουργούσε ιδιαίτερο πρόβλημα στα πλαίσια των στόχων του "πειράματος"). Γιαυτό το λόγο όταν υπολογίσαμε τον πίνακα $A = QR$ θέσαμε μεγαλύτερη ακρίβεια, ώστε να εξαλείψουμε αυτό το πρόβλημα. Συγκρίνουμε λοιπόν τους πίνακες όπως παρακάτω

» norm(Q2-Q) ans=0.2515	Βλέπουμε πόσο ακριβής είναι ο πίνακας Q_2
» norm(R2-R) ans=0.3512	Βλέπουμε πόσο ακριβής είναι ο πίνακας R_2

Τα παραπάνω σφάλματα είναι τεράστια! Οι υπολογισμοί μας έγιναν με ακρίβεια 16 ψηφίων και παρόλα αυτά τα τελικά αποτελέσματα δεν είχαν ακρίβεια ούτε ενός δεκαδικού. Φαίνεται ότι έχουμε χάσει ακρίβεια 16 δεκαδικών. Όμως αν κάνουμε τον παρακάτω έλεγχο

» norm(A-Q2*R2)/norm(A) ans=8.3075e-16	Βλέπουμε πόσο ακριβής είναι ο πίνακας Q_2R_2
---	--

βλέπουμε ότι το γινόμενο Q_2R_2 έχει ακρίβεια 16 ψηφίων. Έτσι τα επιμέρους σφάλματα στους πίνακες Q , Q_2 και R , R_2 εξαλείφονται με τον πολλαπλασιασμό Q_2R_2 . Για να δείξουμε πόσο ξεχωριστή είναι η ακρίβεια του γινομένου Q_2R_2 θεωρούμε μικρές διαταραχές των πινάκων αυτών.

» Q3=Q+1e-4*randn(50);	Θεωρούμε τυχαία διαταραχή του Q , που είναι πιο κοντά από τον Q_2
» R3=R+1e-4*randn(50);	Θεωρούμε τυχαία διαταραχή του R , που είναι πιο κοντά από τον R_2
» norm(A-Q3*R3)/norm(A) ans=7.8006e-04	Βλέπουμε πόσο ακριβής είναι ο πίνακας Q_3R_3

Στο παραπάνω παράδειγμα δεν μεριμνήσαμε ώστε ο R_3 να είναι άνω-τριγωνικός και ο Q_3 να είναι ορθογώνιος, όμως δεν θα είχαμε ποιοτικά μεγαλύτερες διαφορές. Βλέπουμε λοιπόν ότι το σφάλμα είναι τεράστιο. Ο Q_3 είναι καλύτερος από τον Q_2 και ο R_3 από τον R_2 όμως η ακρίβεια των δύο γινομένων δεν έχει καμία σχέση.

Τα σφάλματα στους πίνακες Q_2 , R_2 είναι προς τα εμπρός σφάλματα. Το σφάλμα του Q_2R_2 είναι το προς τα πίσω σφάλμα. Το πολύ μικρό μέγεθος

του σφάλματος αυτού υπονοεί ότι ο αλγόριθμος της Householder τριγωνοποίησης είναι προς τα πίσω ευσταθής.

Θεώρημα III.3. Έστωσαν ένας πίνακας $A \in \mathbb{C}^{m \times n}$ και ένας υπολογιστής που ικανοποιεί τα αξιώματα 1.23 και 1.25. Αν ο αλγόριθμος της Householder τριγωνοποίησης υπολογίσει την QR παραγοντοποίηση του πίνακα A και μας δώσει σαν αποτελέσματα τους πίνακες \tilde{Q} και \tilde{R} , θα ισχύει

$$\tilde{Q}\tilde{R} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}) \quad (1.37)$$

για κάποιο $\delta A \in \mathbb{C}^{m \times n}$

Περισσότερες λεπτομέρειες για το θεώρημα III.3 στο [1] θεώρημα 16.1.

IV Βασικά στοιχεία της θεωρίας διαταραχών

Σε αυτή την σύντομη ενότητα θα παρουσιάσουμε κάποια σημαντικά θεωρήματα της θεωρίας διαταραχών (Perturbation theory). Αυτά τα θεωρήματα θα μας φανούν ιδιαίτερα χρήσιμα για να δείξουμε την υψηλή ακρίβεια όταν θα υπολογίζουμε ιδιοτιμές και ιδιοδιανύσματα από προς τα πίσω ευσταθείς ή ευσταθείς αλγόριθμους στα επόμενα κεφάλαια. Έχουμε δει τα απόλυτα και σχετικά σφάλματα σε προηγούμενες υποενότητες και έχουμε αναφέρει ότι θα εξετάζουμε την ακρίβεια κυρίως από τη σκοπιά του σχετικού σφάλματος. Παρ' όλα αυτά θα παρουσιάσουμε και θεωρήματα που μας δίνουν πολύ χρήσιμες πληροφορίες και για τα απόλυτα σφάλματα. Ο βασικός στόχος μας είναι να δείξουμε ότι για μικρές διαταραχές σε συμμετρικούς τετραγωνικούς πίνακες (οι οποίες προκύπτουν είτε τεχνητά είτε λόγω αριθμητικών υπολογισμών και στρογγυλοποιήσεων) οι διαταραχές στις ιδιοτιμές και τα ιδιοδιανύσματα είναι πολύ μικρές.

Ένα πολύ βασικό θεώρημα (Bauer-Fike) μαζί με ένα λήμμα το παρουσιάζουμε στο τρίτο κεφάλαιο προκειμένου να τεκμηριώσουμε θεωρητικά την τεχνική της αναγωγής σε απλούστερο πρόβλημα (deflation), όμως ας πάρουμε τα πράγματα από την αρχή.

Έστω ότι ο τετραγωνικός πίνακας $A \in \mathbb{R}^{m \times m}$ είναι συμμετρικός, οι ιδιοτιμές του α_i και τα αντίστοιχα ιδιοδιανύσματα x_i και επίσης έστω ο τετραγωνικός, συμμετρικός πίνακας E και ο $\hat{A} = A + E$ με τις διαταραγμένες ιδιοτιμές $\hat{\alpha}_i$ και τα αντίστοιχα ιδιοδιανύσματα \hat{x}_i . Ο βασικός μας στόχος αρχικά είναι να φράξουμε τις απόλυτες διαφορές των ιδιοτιμών και των ιδιοδιανυσμάτων σε σχέση με τη νόρμα του E . Το επόμενο θεώρημα του Weyl μπορούμε να πούμε ότι συμπληρώνει το θεώρημα Bauer-Fike.

Θεώρημα IV.1. Έστωσαν οι τετραγωνικοί και συμμετρικοί πίνακες A, E και οι ιδιοτιμές $\alpha_1 \geq \dots \geq \alpha_m$ του A και $\hat{\alpha}_1 \geq \dots \geq \hat{\alpha}_m$ του $A + E$. Τότε ισχύει $|\alpha_i - \hat{\alpha}_i| \leq \|E\|_2$.

Λήμμα IV.1. Έστωσαν οι τετραγωνικοί πίνακες G, F ίδιας διάστασης όπου οι $\sigma_1 \geq \dots \geq \sigma_m$ οι ιδιάζουσες τιμές του G και $\sigma'_1 \geq \dots \geq \sigma'_m$ οι ιδιάζουσες τιμές του $G + F$. Τότε ισχύει $|\sigma_i - \sigma'_i| \leq \|F\|_2$.

Περισσότερες λεπτομέρειες για το θεώρημα IV.1 και το λήμμα IV.1 στο [7], υποενότητα 5.2.

Μπορούμε να χρησιμοποιήσουμε το θεώρημα IV.1 προκειμένου να υπολογίσουμε άνω φράγματα για τα σφάλματα, όταν υπολογίζουμε ιδιοτιμές με προς τα πίσω ευσταθείς αλγόριθμους. Ένας προς τα πίσω ευσταθής αλγόριθμος υπολογίζει τις ακριβείς ιδιοτιμές του $\hat{A} = A + E$ (βλέπε κεφάλαιο 2, θεώρημα VI.1), με $\|E\|_2 = O(\epsilon_{machine})\|A\|_2 = O(\epsilon_{machine})\max_j |\alpha_j|$. Αυτό το άνω φράγμα είναι αρκετά ικανοποιητικό, ιδιαίτερα για τις μεγάλες ιδιοτιμές αφού θα υπολογιστούν με τα περισσότερα δεκαδικά σωστά.

Ορισμός IV.1. Έστω ο τετραγωνικός πίνακας $A \in \mathbb{C}^{m \times m}$ και ο αντιστρέ-

ψιμος πίνακας $X \in \mathbb{C}^{m \times m}$. Οι πίνακες A και $B = X^T A X$ λέγονται ορθομοναδιαία όμοιοι πίνακες.

Λήμμα IV.2. Δύο ορθομοναδιαία όμοιοι πίνακες έχουν το ίδιο πλήθος θετικών, μηδενικών και αρνητικών ιδιοτιμών.

Περσσότερες λεπτομέρειες για το λήμμα IV.2 στο [7] υποενότητα 5.2.

Για να δώσουμε κάποια αντίστοιχα αποτελέσματα για τα ιδιοδιανύσματα πρέπει πρώτα να ορίσουμε την έννοια της απόστασης μιας ιδιοτιμής

Ορισμός IV.2. Έστω ο τετραγωνικός πίνακας A με ιδιοτιμές $\alpha_1 \geq \dots \geq \alpha_m$. Ονομάζουμε απόσταση μια ιδιοτιμής α_i από τις υπόλοιπες του φάσματος και τη συμβολίζουμε με $gap(i, A)$ το

$$gap(i, A) = \min_j |\alpha_i - \alpha_j|$$

Θεώρημα IV.2. Έστω $A = Q\Lambda Q^T$ μια διαγωνιοποίηση του A και $A + E = \hat{A} = \hat{Q}\hat{\Lambda}\hat{Q}^T$ μια διαταραγμένη διαγωνιοποίηση του. Έστωσαν τα x_i και \hat{x}_i , τα ιδιοδιανύσματα του A και τα διαταραγμένα ιδιοδιανύσματα αντίστοιχα. Έστω θ η οξεία γωνία των ιδιοδιανυσμάτων x_i και \hat{x}_i . Τότε ισχύει

$$\frac{1}{2} \sin 2\theta \leq \frac{\|E\|_2}{gap(i, A)}, \quad \text{Δεδομένου} \quad gap(i, A) > 0$$

Όμοια,

$$\frac{1}{2} \sin 2\theta \leq \frac{\|E\|_2}{gap(i, A + E)}, \quad \text{Δεδομένου} \quad gap(i, A + E) > 0$$

Περσσότερες λεπτομέρειες για το θεώρημα IV.2 στο [7] υποενότητα 5.2

Βλέπουμε ότι όσο μικρότερη είναι η απόσταση μιας ιδιοτιμής τόσο πιο ευαίσθητο είναι το αντίστοιχο ιδιοδιάνυσμά της. Επίσης σημειώστε ότι όταν $\theta \ll 1$, τότε $\frac{1}{2} \sin 2\theta \approx \sin \theta \approx \theta$. Το δεύτερο άνω φράγμα του θεωρήματος IV.2 έχει μεγάλη πρακτική σημασία αφού συνήθως ξέρουμε τις ιδιοτιμές του $A + E$ καθώς τις έχουμε υπολογίσει μέσω κάποιου αλγόριθμου και έτσι υπολογίζοντας το $gap(i, A + E)$ μπορούμε να εκτιμήσουμε ένα άνω φράγμα για το θ και μια εκτίμηση για το $gap(i, A)$.

Στη συνέχεια θα δώσουμε κάποια ανάλογα αποτελέσματα χρησιμοποιώντας τα σχετικά σφάλματα. Για παράδειγμα έστω το βαθμωτό μέγεθος α και η διαταραχή του $\hat{\alpha} = x^2 \alpha$, με $|x^2 - 1| \leq \epsilon$, τότε μπορούμε να πάρουμε ένα φράγμα του σχετικού σφάλματος αφού

$$\frac{|\hat{\alpha} - \alpha|}{|\alpha|} = |x^2 - 1| \leq \epsilon$$

Θα γενικεύσουμε αυτή την ιδέα και στους πίνακες φράζοντας τα σχετικά σφάλματα των ιδιοτιμών α_i του A και των ιδιοτιμών $\hat{\alpha}_i$ του $\hat{A} = X^T A X$ δεδομένου ότι $\|X^T X - I\|_2 \approx \epsilon$.

Το παρακάτω θεώρημα είναι γνωστό και ως θεώρημα Weyl.

Θεώρημα IV.3. Έστωσαν οι ιδιοτιμές α_i και $\hat{\alpha}_i$ των πινάκων A και $\hat{A} = X^T A X$ και $\|X^T X - I\|_2 = \epsilon$. Τότε ισχύει $|\hat{\alpha}_i - \alpha_i| \leq |\alpha_i| \epsilon$. Αν $|\alpha_i| \neq 0$ μπορούμε να γράψουμε

$$\frac{|\hat{\alpha}_i - \alpha_i|}{|\alpha_i|} \leq \epsilon$$

Για περισσότερες λεπτομέρειες σε σχέση με το θεώρημα IV.3 στο [7] θεώρημα 5.6.

Με βάση το θεώρημα IV.1 αν ο πίνακας X είναι ορθογώνιος θα είναι $\epsilon = \|X^T X - I\|_2 = 0$, οι πίνακες A και $\hat{A} = X^T A X$ θα έχουν τις ίδιες ιδιοτιμές και αν ο X είναι σχεδόν ορθογώνιος το ϵ θα είναι πολύ μικρό και άρα οι αντίστοιχες ιδιοτιμές θα είναι πολύ κοντά. Αυτή την ιδιότητα που απορρέει από το θεώρημα Weyl θα χρησιμοποιηθεί και στο κεφάλαιο 3, στις υλοποιήσεις αλγορίθμων στο MATLAB.

Λήμμα IV.3. Έστωσαν ο τυχαίοι πίνακες G και $\hat{G} Y^T G X$ και $\sigma_i, \hat{\sigma}_i$ οι ιδιάζουσες τιμές τους. Έστω $\epsilon = \max(\|X^T X - I\|_2, \|Y^T Y - I\|_2)$. Τότε ισχύει $|\hat{\sigma}_i - \sigma_i| \leq \epsilon \sigma_i$. Αν $\sigma_i \neq 0$ μπορούμε να γράψουμε

$$\frac{|\hat{\sigma}_i - \sigma_i|}{\sigma_i} \leq \epsilon$$

Περισσότερες λεπτομέρειες για το λήμμα IV.3 στο [7] λήμμα 5.2.

Εκτός από τις ιδιοτιμές των πινάκων A και $\hat{A} = X^T A X$ μπορούμε να εξάγουμε και ένα αντίστοιχο συμπέρασμα με αυτό του θεωρήματος IV.1 για τα ιδιοδιανύσματα τους. Για να το κάνουμε αυτό θα πρέπει πρώτα να ορίσουμε την έννοια της σχετικής απόστασης στο φάσμα ενός τετραγωνικού πίνακα.

Ορισμός IV.3. Τη σχετική απόσταση μια ιδιοτιμής α_i του πίνακα A και του υπόλοιπου φάσματος τη συμβολίζουμε με $rel_gap(i, A)$ και είναι

$$rel_gap(i, A) = \min_{j \neq i} \frac{|\alpha_j - \alpha_i|}{|\alpha_i|}$$

Θεώρημα IV.4. Έστω ο πίνακας A με τις ιδιοτιμές α_i και τα αντίστοιχα ιδιοδιανύσματα x_i και ο πίνακας $\hat{A} = X^T A X$ με τις ιδιοτιμές $\hat{\alpha}_i$ και τα ιδιοδιανύσματα \hat{x}_i . Έστωσαν θ η οξεία γωνία μεταξύ των x_i, \hat{x}_i και τα $\epsilon_1 = \|I - X^{-T} X^{-1}\|_2$, $\epsilon_2 = \|X - I\|_2$. Υπό την προϋπόθεση ότι $\epsilon_1 \leq 1$ και $rel_gap(i, X^T A X) > 0$, ισχύει

$$\frac{1}{2} \sin 2\theta \leq \frac{\epsilon_1}{1 - \epsilon_1} \cdot \frac{1}{rel_gap(i, X^T A X)} + \epsilon_2$$

Περισσότερες λεπτομέρειες για το θεώρημα IV.4 στο [7] θεώρημα 5.7.

Κεφάλαιο 2

Εύρεση Ιδιοτιμών

I Το πρόβλημα των ιδιοτιμών

Το πρόβλημα της εύρεσης ιδιοτιμών είναι ιδιαίτερα σημαντικό στην επιστήμη των υπολογιστών, ως αρκετά συνηθισμένο ενδιαμέσο πρόβλημα. Επίσης οι αλγόριθμοι εύρεσής τους είναι πολύ ισχυροί αλλά καθόλου προφανείς. Σε αυτή την ενότητα θα παραθέσουμε τη βασική θεωρία της γραμμικής άλγεβρας που χρειαζόμαστε. Οι αλγόριθμοι θα συζητηθούν αργότερα.

I.I Ιδιοτιμές και Ιδιοδιανύσματα

Έστω $A \in \mathbb{C}^{m \times m}$ ένας τετραγωνικός πίνακας. Ένα μη μηδενικό διάνυσμα $x \in \mathbb{C}^m$ λέμε ότι είναι ένα ιδιοδιάνυσμα του A και $\lambda \in \mathbb{C}$ η αντιστοιχεί ιδιοτιμή του, αν

$$Ax = \lambda x$$

Ορισμός I.1. Η εξίσωση $\det(tI - A) = 0$ λέγεται χαρακτηριστική εξίσωση του πίνακα A , ενώ το πολυώνυμο $\pi_A(t) = \det(tI - A)$ λέγεται χαρακτηριστικό πολυώνυμο του πίνακα A . Το σύνολο των ιδιοτιμών του πίνακα A λέγεται φάσμα του A και θα το συμβολίζουμε με $\sigma(A)$.

Θεώρημα I.1. Ο αριθμός $\lambda \in \mathbb{C}$ είναι ιδιοτιμή του πίνακα $A \in \mathbb{C}^{m \times m}$, αν και μόνο αν ικανοποιεί την εξίσωση (ως προς t)

$$\det(tI - A) = 0$$

Απόδειξη. Με βάση το [9] θεώρημα 11.1.1

Η απόδειξη γίνεται εύκολα αν χρησιμοποιήσουμε τον ορισμό της ιδιοτιμής:

Έστω λ μια ιδιοτιμή \Leftrightarrow υπάρχει ένα μη μηδενικό διάνυσμα x τέτοιο ώστε $\lambda x - Ax = 0 \Leftrightarrow 0$ πίνακας $\lambda I - A$ είναι μη αντιστρέψιμος $\Leftrightarrow \det(\lambda I - A) = 0 \quad \square$

Το παραπάνω θεώρημα έχει μια σημαντική συνέπεια. Ακόμη και αν ο πίνακας είναι πραγματικός κάποιες από τις ιδιοτιμές του μπορεί να είναι μιγαδικές. Στην αλγοριθμική του διάσταση αυτό σημαίνει σε ένα πρόβλημα ιδιοτιμών μπορεί το όρισμα να είναι πραγματικό αλλά το αποτέλεσμα μιγαδικό.

Ορισμός I.2. Ο υπόχωρος $V(\lambda) = \{x : Ax = \lambda x, x \in \mathbb{C}^m\}$ του \mathbb{C}^m λέγεται ιδιόχωρος του πίνακα A αντίστοιχος της ιδιοτιμής λ .

Ορισμός I.3. Η πολλαπλότητα μιας ιδιοτιμής $\lambda \in \mathbb{C}$ πίνακα $A \in \mathbb{C}^{m \times m}$ ως ρίζα της χαρακτηριστικής εξίσωσης $\pi_A = 0$ λέγεται αλγεβρική πολλαπλότητα της ιδιοτιμής λ και συμβολίζεται με $\alpha(\lambda)$.

Ορισμός I.4. Η διάσταση του ιδιοχώρου $V(\lambda)$ λέγεται γεωμετρική πολλαπλότητα της ιδιοτιμής λ και συμβολίζεται με $\gamma(\lambda)$, δηλαδή $\gamma(\lambda) = \dim V(\lambda)$.

Μπορούμε να γράψουμε το χαρακτηριστικό πολυώνυμο π_A παραγοντοποιημένο ως εξής:

$$\pi_A(x) = (x - \lambda_1)(x - \lambda_2) \dots (x - \lambda_m) \quad (2.1)$$

για κάποια $\lambda_i \in \mathbb{C}$. Προφανώς κάθε λ_i είναι ιδιοτιμή του A και όλες οι ιδιοτιμές του A εμφανίζονται στην παραπάνω παράσταση. Επομένως το χαρακτηριστικό πολυώνυμο μας δίνει τη δυνατότητα να βλέπουμε την αλγεβρική πολλαπλότητα μιας ιδιοτιμής καθώς και να μετράμε τις ιδιοτιμές ενός πίνακα. Μια ιδιοτιμή λέμε ότι είναι απλή αν η αλγεβρική της πολλαπλότητα είναι ίση με 1. Το επόμενο θεώρημα απορρέει λογικά από τα παραπάνω.

Θεώρημα I.2. Έστω πίνακας $A \in \mathbb{C}^{m \times m}$. Τότε ο A έχει m ιδιοτιμές, μετρώντας και την αλγεβρική τους πολλαπλότητα. Αν οι ρίζες του π_A είναι απλές τότε ο A έχει m διακεκριμένες ιδιοτιμές.

Περαιτέρω για το Θεώρημα I.2 στο [9] πόρισμα 11.4.3(I).

Συνέπεια του παραπάνω θεωρήματος είναι ότι ένας πίνακας $A \in \mathbb{C}^{m \times m}$ έχει τουλάχιστον μία ιδιοτιμή και το πολύ m . Πάντα η αλγεβρική πολλαπλότητα μιας ιδιοτιμής είναι μεγαλύτερη ή ίση από τη γεωμετρική της πολλαπλότητα. Για να το αποδείξουμε όμως χρειάζεται να δείξουμε κάποια πράγματα για τους μετασχηματισμούς ομοιότητας πινάκων.

Ορισμός I.5. Λέμε ότι δύο πίνακες $A, B \in \mathbb{C}^{m \times m}$ είναι όμοιοι αν υπάρχει αντιστρέψιμος πίνακας $P \in \mathbb{C}^{m \times m}$ τέτοιος ώστε $B = P^{-1}AP$.

Θεώρημα I.3. Αν οι πίνακες $A, B \in \mathbb{C}^{m \times m}$ είναι όμοιοι, δηλαδή υπάρχει πίνακας $P \in \mathbb{C}^{m \times m}$ τέτοιος ώστε $B = P^{-1}AP$, τότε έχουν τα ίδια χαρακτηριστικά πολυώνυμα, τις ίδιες ιδιοτιμές με την ίδια αλγεβρική και γεωμετρική πολλαπλότητα και τα ίδια ιδιοδιανύσματα.

Απόδειξη. Με βάση το [1] Θεώρημα 24.3 Έχουμε ότι:

$$xI - B = P^{-1}xIP - P^{-1}AP = P^{-1}(xI - A)P$$

Οπότε θα είναι:

$$\begin{aligned}\pi_B(x) &= \det(xI - B) = \det[P^{-1}(xI - A)P] \\ &= \det P^{-1} \det(xI - A) \det P = \det(xI - A) = \pi_A(x)\end{aligned}$$

Αποδείξαμε την ισότητα των χαρακτηριστικών πολυωνύμων επομένως είναι φανερό και η ταύτιση των ιδιοτιμών και των αλγεβρικών πολλαπλοτήτων τους. Για την ισότητα των γεωμετρικών πολλαπλοτήτων των ιδιοτιμών αρκεί να αποδείξουμε ότι αν $x \in \mathbb{C}^m$ είναι ιδιοδιάνυσμα του πίνακα A, αντίστοιχο της ιδιοτιμής λ , τότε το $v = P^{-1}x$ είναι ιδιοδιάνυσμα του πίνακα B αντίστοιχο της ίδιας ιδιοτιμής λ . Ισχύουν δηλαδή

$$Ax = \lambda x \neq 0 \quad \text{και} \quad BP^{-1}x = P^{-1}Ax$$

Έχουμε

$$Ax = \lambda x \Leftrightarrow P^{-1}Ax = P^{-1}\lambda x \Leftrightarrow BP^{-1}x = \lambda P^{-1}x \Leftrightarrow B(P^{-1}x) = \lambda(P^{-1}x)$$

Και το ζητούμενο αποδείχθηκε. \square

Το θεώρημα 1.3 είναι ιδιαίτερα σημαντικό αλγοριθμικά καθώς αν έχουμε σαν όρισμα ένα πίνακα όσες φορές και αν εφαρμόσουμε πάνω του ένα μετασχηματισμό ομοιότητας ο νέος πίνακας θα είναι "φορέας" των ίδιων ιδιοτιμών και ιδιοδιανυσμάτων με την ίδια αλγεβρική και γεωμετρική πολλαπλότητα.

Θεώρημα 1.4. Έστω λ ιδιοτιμή του πίνακα $A \in \mathbb{C}^{m \times m}$ με αλγεβρική πολλαπλότητα $\alpha(\lambda)$ και γεωμετρική πολλαπλότητα $\gamma(\lambda)$. Τότε ισχύει: $\gamma(\lambda) \leq \alpha(\lambda)$.

Απόδειξη. Με βάση το [1] θεώρημα 24.4

Έστω n η αλγεβρική πολλαπλότητα της ιδιοτιμής λ του πίνακα A. Κατασκευάζουμε τον $m \times n$ πίνακα V_0 του οποίου οι n στήλες είναι η ορθομοναδιαία βάση του ιδιόχωρου $\{x : Ax = \lambda x\}$. Επεκτείνουμε τον V_0 σε ένα ορθομοναδιαίο πίνακα V και θεωρούμε τον πίνακα

$$B = V^*AV = \begin{bmatrix} \lambda I & C \\ 0 & D \end{bmatrix} \quad (2.2)$$

Οι A,B είναι όμοιοι. Ο I είναι ο $n \times n$ ταυτοτικός πίνακας, ο C είναι ένας $n \times (m-n)$ πίνακας και ο D είναι ένας $(m-n) \times (m-n)$ πίνακας. Από τον ορισμό της ορίζουσας έχουμε

$$\det(xI - B) = \det(xI - \lambda I) \det(xI - D) = (x - \lambda)^n \det(xI - D).$$

Επομένως η αλγεβρική πολλαπλότητα του λ ως ιδιοτιμή του B είναι τουλάχιστον n . Αφού οι A,B είναι όμοιοι τότε το ίδιο θα ισχύει και για την αλγεβρική πολλαπλότητα του λ ως ιδιοτιμή του A. \square

I.II Διαγωνιοποίηση Πίνακα

Μια σπουδαία χρησιμότητα των ιδιοτιμών και των ιδιοδιανυσμάτων ενός πίνακα είναι η δυνατότητα που μας δίνουν να διαπιστώσουμε αν αυτός ο πίνακας είναι όμοιος με κάποιον διαγώνιο πίνακα ή όχι.

Ορισμός I.6. Μια ιδιοτιμή λ της οποίας η αλγεβρική πολλαπλότητα είναι μεγαλύτερη από τη γεωμετρική της πολλαπλότητα λέμε ότι είναι ελαττωματική ιδιοτιμή. Ένας πίνακας $A \in \mathbb{C}^{m \times m}$ ο οποίος έχει τουλάχιστον μία ελαττωματική ιδιοτιμή λέγεται ελαττωματικός.

Οποιοσδήποτε διαγώνιος πίνακας είναι μη-ελαττωματικός. Σε ένα τέτοιο πίνακα η αλγεβρική και γεωμετρική πολλαπλότητα μιας ιδιοτιμής είναι ίσες με τον αριθμό που εμφανίζεται η ιδιοτιμή λ στη διαγώνιο.

Ορισμός I.7. Ο πίνακας $A \in \mathbb{C}^{m \times m}$ είναι διαγωνιοποιήσιμος αν είναι όμοιος με ένα διαγώνιο πίνακα, δηλαδή αν υπάρχει αντιστρέψιμος πίνακας $X \in \mathbb{C}^{m \times m}$ τέτοιος ώστε

$$X^{-1}AX = \Lambda \Leftrightarrow A = X\Lambda X^{-1}$$

Όπου ο Λ είναι διαγώνιος πίνακας.

Βέβαια η διαγωνιοποίηση ενός πίνακα δεν υπάρχει πάντα. Η παραπάνω σχέση μπορεί να γραφτεί και ως

$$AX = X\Lambda$$

Άρα,

$$[A] \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_m \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

Επομένως αν x_j είναι η j -οστή στήλη του X και λ_j το j -οστό διαγώνιο στοιχείο του Λ , τότε θα ισχύει $Ax_j = \lambda_j x_j$. Επομένως η j -οστή στήλη του X είναι ιδιοδιάνυσμα του A και το j -οστό διαγώνιο στοιχείο του Λ η αντίστοιχη ιδιοτιμή.

Θεώρημα I.5. Ένας πίνακας $A \in \mathbb{C}^{m \times m}$ είναι μη-ελαττωματικός αν και μόνο αν διαγωνιοποιείται.

Απόδειξη. Με βάση το [1] θεώρημα 24.5

(\Leftarrow) Δοσμένης της διαγωνιοποίησης $A = X\Lambda X^{-1}$ ξέρουμε ότι ο Λ και ο A έχουν τις ίδιες ιδιοτιμές με τις ίδιες πολλαπλότητες καθώς είναι όμοιοι. Αφού ο Λ είναι διαγώνιος πίνακας θα είναι μη-ελαττωματικός άρα το ίδιο θα συμβαίνει και με τον A .

(\Rightarrow) Αφού ο A είναι μη-ελαττωματικός τότε θα έχει και ακριβώς m γραμμικώς ανεξάρτητα ιδιοδιανύσματα, επειδή τα ιδιοδιανύσματα διαφορετικών ιδιοτιμών αναγκαστικά είναι γραμμικά ανεξάρτητα και κάθε ιδιοτιμή θα αντιστοιχεί το πολύ σε τόσα ιδιοδιανύσματα όσα και η αλγεβρική της πολλαπλότητα. Αν με αυτά τα m γραμμικώς ανεξάρτητα ιδιοδιανύσματα κατασκευάσουμε ένα πίνακα X με στήλες αυτά τα ιδιοδιανύσματα, τότε ο πίνακας X θα είναι αντιστρέψιμος και θα ισχύει $A = X\Lambda X^{-1}$ \square

Από το θεώρημα I.5 καταλαβαίνουμε ότι η ύπαρξη m γραμμικώς ανεξάρτητων ιδιοδιανυσμάτων είναι ικανή και αναγκαία συνθήκη προκειμένου ο πίνακας A να είναι διαγωνιοποιήσιμος με $A = X\Lambda X^{-1}$ όπου οι στήλες του X θα είναι τα ιδιοδιανύσματα του A και στη διαγώνιο του Λ θα βρίσκονται οι ιδιοτιμές του A μετρώντας την αλγεβρική τους πολλαπλότητα.

Θεώρημα I.6. Η ορίζουσα και το ίχνος ενός πίνακα A είναι ίσα με το γινόμενο και το άθροισμα των ιδιοτιμών του A αντίστοιχα μετρώντας την αλγεβρική πολλαπλότητα.

$$\det(A) = \prod_{i=1}^m \lambda_i, \quad \text{tr}(A) = \sum_{i=1}^m \lambda_i \quad (2.3)$$

Απόδειξη. Με βάση το [1] θεώρημα 24.6 Από τον ορισμό I.1 και την I.1.1 έχουμε

$$\det(A) = (-1)^m \det(-A) = (-1)^m \pi_A(0) = \prod_{i=1}^m \lambda_i$$

Οπότε αποδείξαμε την πρώτη σχέση. Η δεύτερη αποδεικνύεται εύκολα καθώς από τον ορισμό I.1 προκύπτει ότι ο συντελεστής του x^{m-1} του π_A είναι το αρνητικό άθροισμα των διαγώνιων στοιχείων του A , δηλαδή $-\text{tr}(A)$. Όμως από την 2.1 προκύπτει ότι ο ίδιος συντελεστής είναι ίσος με $-\sum_{i=1}^m \lambda_i$. Επομένως $\text{tr}(A) = \sum_{i=1}^m \lambda_i$. \square

I.III Ορθομοναδιαία διαγωνιοποίηση

Μερικές φορές συμβαίνει ένας πίνακας $A \in \mathbb{C}^{m \times m}$ όχι μόνο να έχει m γραμμικώς ανεξάρτητα ιδιοδιανύσματα αλλά αυτά να επιλέγονται με τέτοιο τρόπο ώστε να είναι ορθογώνια μεταξύ τους. Σε αυτή την περίπτωση ο A είναι ορθομοναδιαία διαγωνιοποιήσιμος δηλαδή υπάρχει ένας ορθομοναδιαίος πίνακας Q τέτοιος ώστε

$$A = Q\Lambda Q^* \quad (2.4)$$

Αυτή η παραγοντοποίηση είναι παραγοντοποίηση ιδιοτιμών και παραγοντοποίηση ιδιάζουσών τιμών ταυτόχρονα. Μία κλάση πινάκων που είναι ορθομοναδιαία διαγωνιοποιήσιμη είναι αυτή των ερμιτιανών πινάκων.

Θεώρημα 1.7. Ένας ερμιτιανός πίνακας είναι ορθομοναδιαία διαγωνιοποιήσιμος και οι ιδιοτιμές του είναι πραγματικές.

Οι ερμιτιανοί πίνακες δεν είναι οι μόνοι που είναι ορθομοναδιαία διαγωνιοποιήσιμοι. Άλλα παραδείγματα τέτοιων πινάκων είναι οι ορθομοναδιαίοι πίνακες, οι κυκλικόι πίνακες, οι αντί-ερμιτιανοί και όλοι αυτοί προστιθέμενοι σε ένα πολλαπλάσιο του ταυτοτικού πίνακα.

Γενικά η κλάση των πινάκων που είναι ορθομοναδιαία διαγωνιοποιήσιμη έχει ένα ευρύτερο χαρακτηριστικό.

Ορισμός 1.8. Λέμε ότι ένας πίνακας $A \in \mathbb{C}^{m \times m}$ είναι κανονικός αν $A^*A = AA^*$

Θεώρημα 1.8. Ένας πίνακας $A \in \mathbb{C}^{m \times m}$ είναι ορθομοναδιαία διαγωνιοποιήσιμος αν και μόνο αν είναι κανονικός.

Περαιτέρω για το θεώρημα 1.8 στο [1] θεώρημα 24.8.

I.IV Schur παραγοντοποίηση

Μια τελευταία περίπτωση παραγοντοποίησης πινάκων είναι η Schur παραγοντοποίηση, η οποία είναι και η πιο σημαντική στην αριθμητική ανάλυση επειδή όλοι οι πίνακες (και οι ελαττωματικοί) μπορούν να παραγοντοποιηθούν με αυτό τον τρόπο. Η Schur παραγοντοποίηση ενός πίνακα $A \in \mathbb{C}^{m \times m}$ είναι της μορφής

$$A = QTQ^* \quad (2.5)$$

Όπου ο $Q \in \mathbb{C}^{m \times m}$ είναι ορθομοναδιαίος και ο $T \in \mathbb{C}^{m \times m}$ είναι άνω τριγωνικός με διαγώνια στοιχεία τις ιδιοτιμές του A (λαμβάνοντας υπόψη και τις πολλαπλότητες).

Θεώρημα 1.9. Κάθε τετραγωνικός πίνακας $A \in \mathbb{C}^{m \times m}$ έχει μια Schur παραγοντοποίηση.

Απόδειξη. Με βάση το [1] θεώρημα 24.9

Χρησιμοποιούμε επαγωγή για το μέγεθος m του πίνακα A . Η περίπτωση για $m=1$ είναι τετριμμένη έτσι θεωρούμε $m \geq 2$. Έστω x ένα ιδιοδιάνυσμα του πίνακα A και λ η αντίστοιχη ιδιοτιμή του. Επιλέγουμε το x ώστε να είναι μοναδιαίο και το θεωρούμε ως την πρώτη στήλη ενός ορθομοναδιαίου πίνακα U . Τότε από την 2.2 εύκολα βλέπουμε ότι το γινόμενο πινάκων U^*AU έχει τη μορφή

$$U^*AU = \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix}$$

Λόγω της επαγωγικής υπόθεσης υπάρχει μια Schur παραγοντοποίηση του πίνακα T , της μορφής VTV^* . Έτσι θεωρούμε τον πίνακα

$$Q = U \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}$$

Ο Q είναι προφανές ότι είναι ορθομοναδιαίος πίνακας και εύκολα βλέπουμε ότι το γινόμενο

$$Q^*AQ = \begin{bmatrix} \lambda & BV \\ 0 & T \end{bmatrix}$$

είναι μια Schur παραγοντοποίηση του πίνακα A . □

I.V Συμπεράσματα

Στις προηγούμενες σελίδες περιγράψαμε τρεις περιπτώσεις (ή τρόπους) μετασχηματισμού ενός πίνακα προκειμένου ύστερα από την πραγματοποίησή τους να μπορέσουμε εύκολα να βρούμε τις ιδιοτιμές τους και τα ιδιοδιανύσματά τους. Ουσιαστικά πρόκειται για τρεις μετασχηματισμούς που "αποκαλύπτουν" τις ιδιοτιμές ενός πίνακα και είναι τα βασικά αλγοριθμικά εργαλεία προκειμένου να υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα ενός πίνακα. Συνοψίζοντας αυτά είναι:

1. Η διαγωνιοποίηση $A = \lambda X X^{-1}$ υπάρχει αν και μόνο αν ο A είναι μη-ελαττωματικός.
2. Η ορθομοναδιαία παραγοντοποίηση $A = Q \Lambda Q^*$ υπάρχει αν και μόνο αν ο A είναι κανονικός.
3. Η ορθομοναδιαία τριγωνοποίηση (Schur παραγοντοποίηση) $A = QTQ^*$ υπάρχει πάντα.

Προκειμένου να βρούμε τις ιδιοτιμές ενός πίνακα πρέπει να κατασκευάσουμε κάποιες από αυτές τις παραγοντοποιήσεις. Γενικά πιο συχνά χρησιμοποιούμενος μετασχηματισμός είναι η Schur παραγοντοποίηση διότι ισχύει για όλους τους πίνακες χωρίς περιορισμούς. Επιπλέον αν χρησιμοποιήσουμε ορθομοναδιαίους μετασχηματισμούς σαν ενδιάμεσα βήματα οι αλγόριθμοι τείνουν να γίνουν αριθμητικά ευσταθείς. Αν ο A είναι κανονικός τότε η μορφή Schur έχει σαν αποτέλεσμα ένα διαγώνιο, ενώ αν ο A είναι ερμιτιανός μπορούμε να εκμεταλλευτούμε τη συμμετρία του και να τον μετασχηματίσουμε σε διαγώνιο με το μισό κόπο από ότι για κάποιον τυχαίο πίνακα.

II Επισκόπηση αλγορίθμων

Αυτή και η επόμενη ενότητα του κεφαλαίου περιγράφουν κάποιες κλασικούς αλλά και πιο μοντέρνους αλγόριθμους για τον υπολογισμό ιδιοτιμών και ιδιοδιανυσμάτων. Οι περισσότεροι αλγόριθμοι χαρακτηρίζονται από δύο φάσεις: ένα πρώτο αρχικό μετασχηματισμό σε ένα ενδιάμεσο πίνακα που φυσικά θα έχει τις ίδιες ιδιοτιμές και ιδιοδιανύσματα με τον αρχικό και δεύτερον μια επαναληπτική διαδικασία που συγκλίνει σε ένα πίνακα με τη βοήθεια του οποίου προκύπτουν άμεσα οι ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα. Αυτή η ενότητα περιγράφει τις δύο φάσεις και εξηγεί τα πλεονεκτήματά τους.

II.1 Προφανείς αλγόριθμοι: Μειονεκτήματα

Παρόλο που οι ιδιοτιμές και τα ιδιοδιανύσματα έχουν απλούς ορισμούς και χαρακτηριστικά οι καλύτεροι τρόποι να τους υπολογίσουμε δεν είναι καθόλου προφανείς.

Συνήθως η πρώτη ιδέα που έχουμε είναι να βρούμε το χαρακτηριστικό πολυώνυμο του πίνακα και μέσω των συντελεστών του να υπολογίσουμε τις ιδιοτιμές. Αυτός ο τρόπος όπως αναφέραμε και στο προηγούμενο κεφάλαιο δεν είναι καθόλου καλή ιδέα καθώς το πρόβλημα της εύρεσης ριζών ενός πολυωνύμου είναι κακής κατάστασης (έχει μεγάλο σχετικό δείκτη κατάστασης) ακόμα και αν το αρχικό πρόβλημα ιδιοτιμών είναι καλής κατάστασης, καθώς και για ακόμα ένα λόγο που θα δούμε στη συνέχεια. Γενικά η εύρεση των ριζών ενός πολυωνύμου δεν είναι καθόλου χρήσιμο εργαλείο καθώς σπάνια είναι ο καλύτερος τρόπος να λύσουμε ένα εφαρμοσμένο πρόβλημα.

Μια άλλη ιδέα είναι να εκμεταλλευτούμε το γεγονός ότι η ακολουθία

$$\frac{x}{\|x\|}, \frac{Ax}{\|Ax\|}, \frac{A^2x}{\|A^2x\|}, \frac{A^3x}{\|A^3x\|}, \dots \quad (2.6)$$

συγκλίνει (κάτω από συγκεκριμένες προϋποθέσεις) στο ιδιοδιάνυσμα που αντιστοιχεί στη μεγαλύτερη κατά απόλυτη τιμή ιδιοτιμή του πίνακα A . Αυτή η μέθοδος εύρεσης ιδιοδιανύσματος λέγεται μέθοδος των δυνάμεων. Δυστυχώς, αν και αυτή η μέθοδος είναι διάσημη δεν είναι καθόλου αποτελεσματική για γενική χρήση. Εκτός από συγκεκριμένες περιπτώσεις είναι αρκετά αργή γεγονός που δεν την καθιστά ιδιαίτερα ελκυστική.

Αντί για ιδέες όπως οι προηγούμενες η πιο αποτελεσματική είναι να υπολογίσουμε μια παραγοντοποίηση όπως αυτές που περιγράψαμε στην προηγούμενη ενότητα μέσω της οποίας οι ιδιοτιμές θα είναι στοιχεία ενός από τους πίνακες της παραγοντοποίησης. Στην πράξη οι ιδιοτιμές συνήθως υπολογίζονται όταν μέσω μετασχηματισμών ομοιότητας του αρχικού πίνακα

(επομένως μια κατάλληλη ακολουθία πινάκων) καταλήγουμε σε μία από τις τρεις παραγοντοποιήσεις που περιγράψαμε. Πιο συγκεκριμένα το ζητούμενο είναι στην ακολουθία μετασχηματισμένων πινάκων να εισάγουμε μηδενικά στα αναγκαία σημεία ώστε να μπορέσουμε να κατασκευάσουμε τη ζητούμενη παραγοντοποίηση. Αυτή η τεχνική είναι η βάση πολλών αλγορίθμων στην αριθμητική γραμμική άλγεβρα: εισαγωγή μηδενικών σε πίνακες.

II.II Μία θεμελιώδης δυσκολία

Αναφέραμε στην προηγούμενη υποενότητα ότι η προσπάθεια υπολογισμού των ιδιοτιμών ενός πίνακα A μέσω του χαρακτηριστικού πολυωνύμου δεν είναι καλή ιδέα, καθώς η εύρεση των ριζών πολυωνύμου είναι ένα πρόβλημα κακής κατάστασης. Παρ' όλα αυτά υπάρχει ακόμα το ερώτημα: Θα ήταν δυνατόν οι αλγόριθμοι εύρεσης ιδιοτιμών να βασίζονταν σε αλγεβρικού τύπου υπολογισμούς, με συγκεκριμένες και σταθερού αριθμού πράξεις και βήματα, χωρίς επαναληπτικές διαδικασίες; Η απάντηση είναι αρνητική αλλά θα πρέπει να δούμε τα πράγματα σε μεγαλύτερο βάθος.

Για να καταλάβουμε τη δυσκολία πρέπει να δούμε ότι ένα πρόβλημα ιδιοτιμών μπορεί να αναχθεί σε πρόβλημα εύρεσης ριζών ενός πολυωνύμου και το αντίστροφο, κάθε πρόβλημα εύρεσης ριζών ενός πολυωνύμου μπορεί να αναχθεί σε πρόβλημα εύρεσης ιδιοτιμών ενός τετραγωνικού πίνακα. Έστω το κανονικό πολυώνυμο (ο συντελεστής του μεγιστοβάθμιου όρου είναι ίσος με 1)

$$\pi(x) = x^m + \alpha_{m-1}x^{m-1} + \dots + \alpha_1x + \alpha_0 \quad (2.7)$$

Δεν είναι δύσκολο να δείξουμε ότι το $\pi(x)$ είναι ίσο με το γινόμενο του παράγοντα $(-1)^m$ επί την ορίζουσα του $m \times m$ πίνακα

$$\begin{bmatrix} -x & & & & & -\alpha_0 \\ 1 & -x & & & & -\alpha_1 \\ & 1 & -x & & & -\alpha_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & -x & -\alpha_{m-2} \\ & & & & 1 & (-x - \alpha_{m-1}) \end{bmatrix} \quad (2.8)$$

Άρα οι ρίζες του $\pi(x)$ είναι ίσες με τις ιδιοτιμές του πίνακα

$$A = \begin{bmatrix} 0 & & & & -\alpha_0 \\ 1 & 0 & & & -\alpha_1 \\ & 1 & 0 & & -\alpha_2 \\ & & 1 & \ddots & \vdots \\ & & & \ddots & 0 & -\alpha_{m-2} \\ & & & & 1 & -\alpha_{m-1} \end{bmatrix} \quad (2.9)$$

Ο πίνακας A ονομάζεται συντροφικός πίνακας του πολυώνυμου p .

Τώρα η δυσκολία είναι εμφανής. Όπως είναι γνωστό δεν υπάρχει κάποιος τυποποιημένος τρόπος να εκφράσεις τις ρίζες ενός τυχαίου πολυωνύμου σε συνάρτηση με τους συντελεστές του. Το θεώρημα του Abel το 1824 μας λέει ότι δεν μπορεί να υπάρξει ανάλογη τυποποίηση της λύσης του τριωνύμου για πολυώνυμα βαθμού μεγαλύτερου ή ίσου του 5.

Θεώρημα II.1. Για κάθε $m \geq 5$, υπάρχει ένα πολυώνυμο $p(x)$ βαθμού m με ρητούς συντελεστές το οποίο έχει μια πραγματική ρίζα $p(r) = 0$ με την ιδιότητα ότι το r δεν μπορεί να εκφραστεί σαν αριθμητική παράσταση ρητών αριθμών (παράσταση με προσθέσεις, αφαιρέσεις, πολλαπλασιασμούς, διαιρέσεις και n -οστές ρίζες).

Η αλγοριθμική αξία αυτού του θεωρήματος είναι ότι ακόμα και αν μπορούσαμε να δουλέψουμε με ακριβείς αριθμητικούς υπολογισμούς, δεν θα υπήρχε κανένα υπολογιστικό πρόγραμμα που θα μπορούσε να βρει τις ρίζες ενός τυχαίου πολυωνύμου σε πεπερασμένο αριθμό βημάτων. Το ίδιο προφανώς ισχύει και στο πρόβλημα εύρεσης ιδιοτιμών πινάκων. Αυτό δεν σημαίνει ότι δεν μπορούμε να βρούμε καλούς αλγόριθμους για το πρόβλημά μας αλλά ουσιαστικά σημαίνει ότι

*Κάθε αλγόριθμος υπολογισμού ιδιοτιμών
πρέπει να είναι επαναληπτικός*

Ο στόχος μας είναι η δημιουργία μιας ακολουθίας αριθμών που θα συγκλίνει γρήγορα στις ιδιοτιμές του πίνακα. Η ανάγκη να κατασκευάσουμε επαναληπτικούς αλγόριθμους μπορεί αρχικά να φαίνεται αποθαρρυντικός όμως υπάρχει η δυνατότητα κατασκευής αλγορίθμων που συγκλίνουν πάρα πολύ γρήγορα. Στις περισσότερες περιπτώσεις είναι δυνατό να υπολογίζουμε ακολουθίες αριθμών που διπλασιάζουν ή τριπλασιάζουν την ακρίβεια σε κάθε βήμα.

II.III Schur παραγοντοποίηση και διαγωνιοποίηση

Οι περισσότεροι αλγόριθμοι υπολογισμού ιδιοτιμών ενός τυχαίου πίνακα βρίσκουν τη Schur παραγοντοποίηση για τους λόγους που προαναφέραμε. Υπολογίζουμε τη Schur παραγοντοποίηση $A = QTQ^*$ μέσω μιας ακολουθίας στοιχειωδών ορθομοναδιαίων μετασχηματισμών ομοιότητας του A , $X \mapsto Q_j^* X Q_j$, ώστε το γινόμενο

$$Q_j^* \cdots Q_2^* Q_1^* A Q_1 Q_2 \cdots Q_j \quad (2.10)$$

με $Q^* = Q_j^* \cdots Q_2^* Q_1^*$ και $Q = Q_1 Q_2 \cdots Q_j$, να συγκλίνει σε ένα άνω τριγωνικό πίνακα T καθώς $j \rightarrow \infty$.

Αν ο A είναι πραγματικός αλλά όχι συμμετρικός τότε πιθανόν να έχει μιγαδικές, συζυγείς ιδιοτιμές, επομένως και η μορφή Schur να είναι μιγαδική. Έτσι ένας αλγόριθμος που υπολογίζει μια Schur παραγοντοποίηση θα πρέπει να είναι ικανός να παράγει μιγαδικά αποτελέσματα ενώ θα έχει πραγματικό όρισμα. Εναλλακτικά υπάρχει τρόπος να κάνει όλους τους υπολογισμούς στο σύνολο των πραγματικών αν υπολογίζει την πραγματική Schur παραγοντοποίηση.

Αν υποθέσουμε ότι ο A είναι ερμιτιανός τότε ο $Q_j^* \dots Q_2^* Q_1^* A Q_1 Q_2 \dots Q_j$ είναι επίσης ερμιτιανός πίνακας και επομένως το όριο της συγκλίνουσας ακολουθίας είναι ταυτόχρονα τριγωνικός και ερμιτιανός πίνακας, άρα διαγωνίος. Αυτό σημαίνει ότι ο ίδιος αλγόριθμος που υπολογίζει μία ορθομοναδιαία τριγωνοποίηση (δηλαδή Schur παραγοντοποίηση) ενός τυχαίου πίνακα υπολογίζει και μία ορθομοναδιαία διαγωνιοποίηση ενός ερμιτιανού πίνακα. Στην πράξη έτσι χειριζόμαστε τους ερμιτιανούς πίνακες, αν και υπάρχουν διάφορες μετατροπές προκειμένου να εκμεταλλευτούμε τη συμμετρικότητα σε κάθε βήμα.

II.IV Οι δύο φάσεις του υπολογισμού ιδιοτιμών

Είτε ο πίνακας $A \in \mathbb{C}^{m \times m}$ είναι ερμιτιανός είτε όχι, η ακολουθία 2.10 συνήθως σπάει σε δύο κομμάτια. Στην πρώτη φάση εφαρμόζεται μια άμεση μέθοδος που μετασχηματίζει τον πίνακα μας μέσω μετασχηματισμών ομοιότητας σε ένα άνω-Hessenberg πίνακα H , ο οποίος έχει μηδενικά κάτω από την υποδιαγώνιό του. Στην δεύτερη φάση δημιουργούμε μια τυπικά άπειρη ακολουθία Hessenberg πινάκων η οποία συγκλίνει σε ένα άνω τριγωνικό πίνακα. Η διαδικασία σχηματικά είναι η παρακάτω:

$$\begin{array}{ccc}
 \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} & \xrightarrow{\text{1η Φάση}} & \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{bmatrix} \\
 A \neq A^* & & H \\
 & & \xrightarrow{\text{2η Φάση}} \\
 & & \begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{bmatrix} \\
 & & T
 \end{array}$$

Η πρώτη φάση του μετασχηματισμού σε άνω-Hessenberg πίνακα απαιτεί $O(m^3)$ πράξεις. Η δεύτερη επαναληπτική φάση θεωρητικά δεν σταματάει ποτέ και αν την αφήσουμε να τρέχει για πάντα θα χρειαστεί άπειρο αριθμός

πράξεων. Στην πράξη όμως επιτυγχάνεται σύγκλιση με ακρίβεια μηχανής σε $O(m)$ επαναλήψεις. Κάθε επανάληψη απαιτεί $O(m^2)$ πράξεις, επομένως η δεύτερη φάση απαιτεί συνολικά $O(m^3)$ πράξεις. Η τελευταία παρατήρηση καταδεικνύει τη σημαντικότητα της πρώτης φάσης καθώς χωρίς αυτό το αρχικό βήμα κάθε επανάληψη της δεύτερης φάσης θα περιελάμβανε ένα "γεμάτο" πίνακα και θα απαιτούσε $O(m^3)$ πράξεις και άρα η δεύτερη φάση θα απαιτούσε συνολικά $O(m^4)$ πράξεις (ή και παραπάνω αφού η σύγκλιση κάποιες φορές απαιτεί παραπάνω από $O(m)$ επαναλήψεις).

Αν ο πίνακας A είναι ερμιτιανός τότε η όλη διαδικασία γίνεται ακόμα πιο γρήγορα. Ο ενδιαμέσος, μετά την πρώτη φάση, πίνακας θα είναι ένας ερμιτιανός Hessenberg πίνακας, άρα τριδιαγώνιος. Σχηματικά η διαδικασία θα έχει την παρακάτω μορφή:

$$\begin{array}{ccc}
 \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} & \xrightarrow{1\text{η Φάση}} & \begin{bmatrix} x & x & & & \\ x & x & x & & \\ & x & x & x & \\ & & x & x & x \\ & & & x & x \end{bmatrix} \\
 A=A^* & & T \\
 & & \\
 & & \xrightarrow{2\text{η Φάση}} \begin{bmatrix} x & & & & \\ & x & & & \\ & & x & & \\ & & & x & \\ & & & & x \end{bmatrix} \\
 & & D
 \end{array}$$

Επίσης στην περίπτωση που ο A είναι ερμιτιανός πίνακας αν ζητούνται μόνο οι ιδιοτιμές και όχι τα ιδιοδιανύσματα τότε κάθε επανάληψη της δεύτερης φάσης απαιτεί μόλις $O(m)$ πράξεις που σημαίνει ότι η δεύτερη φάση απαιτεί συνολικά $O(m^2)$ πράξεις. Έτσι στα "ερμιτιανά" προβλήματα ιδιοτιμών συναντάμε την παράδοξη κατάσταση το "άπειρο" μέρος του αλγόριθμου να είναι στην πράξη όχι μόνο τόσο γρήγορο όσο το "πεπερασμένο" αλλά μιας τάξης γρηγορότερο.

III Μετασχηματισμός σε μορφή Hessenberg

Σε εαυτή την ενότητα θα περιγράψουμε την πρώτη φάση των αλγορίθμων υπολογισμού ιδιοτιμών και ιδιοδιανυσμάτων: Το μετασχηματισμό του αρχικού μας πίνακα σε άνω-Hessenberg με τη βοήθεια ερμιτιανών μετασχηματισμών ομοιότητας. Αν ο αρχικός μας πίνακας είναι ερμιτιανός το αποτέλεσμα είναι ένας τριδιαγώνιος πίνακας.

III.1 Μία κακή ιδέα

Μια πρώτη ιδέα που εκ πρώτης όψεως φαίνεται καλή είναι η εξής: Για να υπολογίσουμε τη Schur παραγοντοποίηση $A = QTQ^*$ πρέπει να εφαρμόσουμε μια σειρά ορθομοναδιαίων μετασχηματισμών ομοιότητας στον πίνακα A έτσι ώστε να εισάγουμε μηδενικά κάτω από τη διαγώνιό του. Επομένως αυτό που πρέπει να κάνουμε είναι με τη βοήθεια των διανυσμάτων και των αντίστοιχων πινάκων Householder να εισάγουμε μηδενικά στον αρχικό μας πίνακα και να έχουμε σαν αποτέλεσμα μια άμεση τριγωνοποίηση του A .

Ο πρώτος πίνακας Householder Q_1^* θα πολλαπλασιάσει τον πίνακα A από τα αριστερά του και έτσι θα εισάγει μηδενικά κάτω από τη διαγώνιο στην πρώτη στήλη. (Τα στοιχεία του πίνακα που αλλάζουν είναι με bold)

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \xrightarrow{Q_1^*} \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & x & x & x & x \\ \mathbf{0} & x & x & x & x \\ \mathbf{0} & x & x & x & x \\ \mathbf{0} & x & x & x & x \end{bmatrix}$$

A Q_1^*A

Δυστυχώς όμως για να ολοκληρώσουμε το μετασχηματισμό ομοιότητας θα πρέπει να πολλαπλασιάσουμε τον A με τον Q_1 και από δεξιά:

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \xrightarrow{\cdot Q_1} \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix}$$

Q_1^*A $Q_1^*AQ_1$

Αυτό όπως μπορούμε να δούμε αυτό έχει σαν αποτέλεσμα να αντικαθιστούμε κάθε στήλη του πίνακα A με ένα γραμμικό συνδυασμό όλων των στηλών του. Έτσι λοιπόν τα μηδενικά που εισήχθησαν στον πολλαπλασιασμό από τα αριστερά να χαθούν. Το τελικό αποτέλεσμα δεν έχει καμία πρόοδο σε σχέση με τον αρχικό μας πίνακα.

Βέβαια αυτή η διαδικασία ήταν αναμενόμενο να αποτύχει καθώς όπως περιγράψαμε στην υποενότητα II.Π δεν υπάρχει πεπερασμένη διαδικασία με την οποία να μπορέσουμε να υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα ενός πίνακα.

Παρόλο που αυτή η κακή ιδέα φαίνεται να μην έχει καμία χρησιμότητα έχει την ιδιότητα να μειώνει την τιμή των στοιχείων κάτω από την κύρια διαγώνιο χωρίς βέβαια να τα μηδενίζει. Θα επιστρέψουμε σε αυτή την "κακή ιδέα" όταν θα συζητήσουμε για τους QR αλγόριθμους.

III.Π Μία καλή ιδέα

Η σωστή στρατηγική στην πρώτη φάση προκειμένου να εισάγουμε μηδενικά στον πίνακά μας είναι να είμαστε λιγότερο φιλόδοξοι και να "μετασχηματίσουμε" λιγότερα στοιχεία του πίνακα ώστε να καταλήξουμε σε ένα άνω-Hessenberg πίνακα.

Έτσι λοιπόν στο πρώτο βήμα επιλέγουμε ένα πίνακα Householder Q_1^* που όταν πολλαπλασιάσουμε από αριστερά θα αφήσει την πρώτη σειρά του πίνακα μας ανέπαφη. Θα αντικαταστήσει τις σειρές 2, ..., m και θα εισάγει μηδενικά στην πρώτη στήλη από το 3ο έως και το m-οστό στοιχείο της. Έπειτα όταν θα πολλαπλασιάσουμε με τον Q_1 από τα δεξιά τον Q_1^*A η πρώτη στήλη θα μείνει ανέπαφη. Τα στοιχεία που θα μετασχηματίσει θα αντικατασταθούν με γραμμικούς συνδυασμούς των στηλών 2, ..., m και τα μηδενικά μας θα μείνουν ως έχουν:

$$\begin{array}{ccc} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} & \xrightarrow{Q_1^*} & \begin{bmatrix} x & x & x & x & x \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix} & \xrightarrow{\cdot Q_1} & \begin{bmatrix} x & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix} \\ A & & Q_1^*A & & Q_1^*AQ_1 \end{array}$$

Αυτή η διαδικασία συνεχίζεται ώστε να εισάγουμε μηδενικά και στις υπόλοιπες στήλες. Για παράδειγμα ο δεύτερος πίνακας Householder, Q_2^* , θα αφήσει ανέπαφες τις δύο πρώτες σειρές και ο Q_2 τις δύο πρώτες στήλες:

$$\begin{array}{ccc} \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & x & x & x & x \\ & x & x & x & x \end{bmatrix} & \xrightarrow{Q_2^*} & \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix} & \xrightarrow{\cdot Q_2} & \begin{bmatrix} x & x & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x & x & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & x & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ & & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix} \\ Q_1^*AQ_1 & & Q_2^*Q_1^*AQ_1 & & Q_2^*Q_1^*AQ_1Q_2 \end{array}$$

Όταν επαναλάβουμε αυτή τη διαδικασία m-2 φορές θα έχουμε σαν αποτέλεσμα ένα γινόμενο πινάκων το οποίο θα είναι το ζητούμενο αποτέλεσμα

ένας πίνακας άνω-Hessenberg:

$$Q^*AQ = H$$

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{bmatrix}$$

Όπου $Q^* = Q_{m-2}^* \cdots Q_2^* Q_1^*$ και $Q = Q_1 Q_2 \cdots Q_{m-2}$ Ο αλγόριθμος διατυπώνεται παρακάτω.

Αλγόριθμος 2.1
Householder μετασχηματισμός σε μορφή Hessenberg

for $k = 1$ **to** $m - 2$

$x = A_{k+1:m,k}$
 $v_k = \text{sign}(x_1) \|x\|_2 e_1 + x$
 $v_k = v_k / \|v_k\|_2$
 $A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^* A_{k+1:m,k:m})$
 $A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2(A_{1:m,k+1:m} v_k) v_k^*$

Βέβαια στην πρακτική υλοποίηση του αλγόριθμου συναντάμε κάποιες επιπλέον δυσκολίες που έχουν να κάνουν με την κατασκευή των πινάκων Householder, ζητήματα που θα αναπτύξουμε στο τρίτο κεφάλαιο.

III.III Χρόνος εκτέλεσης

Για να μετρήσουμε το σύνολο των πράξεων που κοστίζει ο αλγόριθμος 2.1 πρέπει να μετρήσουμε ξεχωριστά τον όγκο πράξεων των δύο διαδικασιών αλληπάλληλων πολλαπλασιασμών (των από τα αριστερά και των από τα δεξιά). Πρώτα πρέπει να δούμε ότι κάθε φορά που πολλαπλασιάζουμε ένα ερμιτιανό πίνακα Q_i^* από τα αριστερά κάθε στοιχείο που προκύπτει (εκτός από αυτά που μένουν ανέπαφα) απαιτεί τέσσερις πράξεις. Το ίδιο ισχύει και για τους διαδοχικούς πολλαπλασιασμούς των ερμιτιανών Q_i πινάκων από τα δεξιά.

Για τους από αριστερά πολλαπλασιασμούς: Κάθε επανάληψη στον αλγόριθμο 2.1 εκτελεί και ένα πολλαπλασιασμό από τα αριστερά. Ο k -οστός πίνακας Householder μετασχηματίζει τις τελευταίες $m-k$ γραμμές. Όταν ολοκληρωθεί ο πολλαπλασιασμός αυτές οι γραμμές έχουν μηδενικά στις πρώτες k στήλες, ενώ αριθμητικοί υπολογισμοί εκτελούνται στα τελευταία $m-k+1$

στοιχεία κάθε γραμμής. Άρα το σύνολο των στοιχείων που "δημιουργούνται" από τους από τα αριστερά πολλαπλασιασμούς θα είναι:

$$\begin{aligned}
 \sum_{k=1}^{m-2} (m-k+1)(m-k) &= \sum_{k=1}^{m-2} (m^2 - 2km + k^2 - k) \\
 &= m^3 + m^2 - 2m \sum_{k=1}^{m-2} k + \sum_{k=1}^{m-2} k^2 - \sum_{k=1}^{m-2} k \\
 &= m^3 + m^2 - m(m-1)(m-2) + \frac{(m-2)(m-1)(2m-1)}{6} - (m-1)(m-2) \\
 &= \frac{1}{3}m^3 - \frac{11}{6}m^2 + \frac{5}{2}m - \frac{7}{2}
 \end{aligned}$$

Επομένως όταν $m \rightarrow \infty$ το σύνολο των στοιχείων που "δημιουργούνται" από τους από τα αριστερά πολλαπλασιασμούς θα συγκλίνει στο $\frac{1}{3}m^3$. Κάθε στοιχείο όπως αναφέραμε απαιτεί τέσσερις πράξεις, επομένως το σύνολο των πράξεων για τους από τα αριστερά πολλαπλασιασμούς θα είναι $\sim \frac{4}{3}m^3$.

Κάθε επανάληψη εκτελεί επίσης και ένα πολλαπλασιασμό από τα δεξιά. Ο κ -οστός πίνακας Householder μετασχηματίζει τις τελευταίες $m-k$ στήλες. Η διαδικασία των διαδοχικών πολλαπλασιασμών από τα δεξιά περιλαμβάνει περισσότερες πράξεις καθώς δεν υπάρχουν μηδενικά που μπορούν να αγνοηθούν. Αριθμητικοί υπολογισμοί θα γίνουν σε όλα τα στοιχεία κάθε στήλης. Άρα το σύνολο των στοιχείων που θα δημιουργηθούν θα είναι:

$$\begin{aligned}
 \sum_{k=1}^{m-2} m(m-k) &= m^3 - m \sum_{k=1}^{m-2} k \\
 &= \frac{1}{2}m^3 - \frac{3}{2}m^2 + \frac{3}{2}m
 \end{aligned}$$

Επομένως όταν $m \rightarrow \infty$ το σύνολο των στοιχείων που "δημιουργούνται" από τους από τα δεξιά πολλαπλασιασμούς θα συγκλίνει στο $\frac{1}{2}m^3$. Κάθε στοιχείο όπως αναφέραμε απαιτεί τέσσερις πράξεις, επομένως το σύνολο των πράξεων για τους από τα αριστερά πολλαπλασιασμούς θα είναι $\sim 2m^3$.

Το σύνολο λοιπόν των πράξεων για τον ερμιτιανό μετασχηματισμό σε μορφή Hessenberg είναι:

Κόστος για το μετασχηματισμό σε μορφή hessenberg: $\sim \frac{10}{3}m^3$ πράξεις.

III.IV Η περίπτωση ερμιτιανού πίνακα

Αν ο A είναι ερμιτιανός τότε το αποτέλεσμα μας θα είναι ένας τρι-διαγώνιος πίνακας, καθώς θα είναι ερμιτιανός και Hessenberg. Σε αυτή την περίπτωση μηδενικά θα εισάγονται και στις σειρές εκτός από τις στήλες και έτσι αρκετοί αριθμητικοί υπολογισμοί μπορούν να αγνοηθούν.

Πρακτικά αυτό σημαίνει ότι όσες πράξεις απαιτούνται για τους από τα αριστερά πολλαπλασιασμούς τόσες ακριβώς θα απαιτηθούν τελικά και για τους από τα δεξιά πολλαπλασιασμούς. Δηλαδή το σύνολο των πράξεων των από τα δεξιά πολλαπλασιασμών μειώνονται από $2m^3$ στις $\frac{4}{3}m^3$. Επομένως το σύνολο των πράξεων όλης της διαδικασίας θα είναι $\frac{8}{3}m^3$. Βέβαια στην πραγματικότητα επειδή κάθε πίνακας που υπολογίζεται και στους από αριστερά και στους από δεξιά πολλαπλασιασμούς είναι ερμιτιανός θα έχουμε επιπλέον οικονομία με το συνολικό όγκο των πράξεων, χωρίς να αναφέρουμε τις λεπτομέρειες, να είναι τελικά:

Κόστος για το μετασχηματισμό σε μορφή hessenberg: $\sim \frac{4}{3}m^3$ πράξεις.

III.V Ευστάθεια

Όπως και ο αλγόριθμος Householder για την QR παραγοντοποίηση έτσι και ο αλγόριθμος 2.1 είναι προς τα πίσω ευσταθής. Για κάθε $A \in \mathbb{C}^{m \times n}$ ο αλγόριθμος Householder για την QR παραγοντοποίηση υπολογίζει έναν ερμιτιανό πίνακα \tilde{Q} καθώς και έναν άνω τριγωνικό πίνακα \tilde{R} , τέτοιους ώστε

$$\tilde{Q}\tilde{R} = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine})$$

Ακριβώς το ίδιο σφάλμα μπορεί να αποδειχθεί ότι ισχύει και για τον αλγόριθμο 2.1. Έστω ότι ο πίνακας \tilde{H} είναι ο πίνακας Hessenberg που υπολογίστηκε με αριθμητική κινητής υποδιαστολής από τον αλγόριθμο 2.1 και ο πίνακας \tilde{Q} ο ερμιτιανός πίνακας που αντιστοιχεί στα διανύσματα v_k και υπολογίστηκε επίσης από τον αλγόριθμο 2.1 με αριθμητική κινητής υποδιαστολής. Το παρακάτω θεώρημα μπορεί να αποδειχθεί.

Θεώρημα III.1. Έστω ο μετασχηματισμός Hessenberg $A = QHQ^*$ του πίνακα $A \in \mathbb{C}^{m \times m}$ που υπολογίστηκε από τον αλγόριθμο 2.1 σε έναν υπολογιστή που ικανοποιεί τα αξιώματα 1.23, 1.25 και έστωσαν οι πίνακες \tilde{Q} και \tilde{H} που περιγράψαμε παραπάνω. Τότε ισχύει

$$\tilde{Q}\tilde{H}\tilde{Q}^* = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}), \quad \text{για κάποιο } \delta A \in \mathbb{C}^{m \times m}$$

Περισσότερες λεπτομέρειες για το θεώρημα III.1) βλ. [1] θεώρημα 26.1

IV Πηλίκο Rayleigh και αντίστροφη επανάληψη

Σε αυτή την ενότητα θα παρουσιάσουμε κάποιους κλασικούς αλγόριθμους. Μεμονωμένα το κάθε ένα από αυτά τα εργαλεία είναι χρήσιμα υπό συγκεκριμένες συνθήκες ιδιαίτερα ο αλγόριθμος της αντίστροφης επανάληψης που είναι ο βασικός αλγόριθμος υπολογισμού ενός ιδιοδιανύσματος αν ξέρουμε την αντίστοιχη ιδιοτιμή. Παράλληλα πρόκειται και για τα στοιχειώδη εργαλεία του QR αλγόριθμου, που θα αναπτύξουμε σε επόμενη ενότητα.

IV.I Αρχική σύμβαση-αυτοπεριορισμός

Αν και οι περισσότερες μέθοδοι εφαρμόζονται και σε αυθαίρετους και σε ερμιτιανούς πίνακες, όπως αναφέραμε και στις προηγούμενες ενότητες, όταν ο αρχικός πίνακας είναι ερμιτιανός συνήθως έχουμε διάφορα πλεονεκτήματα που έχουν να κάνουν με την ταχύτητα σύγκλισης και την απλούστερη διαδικασία μετασχηματισμού του αρχικού μας πίνακα. Στις επόμενες ενότητες θα επιχειρήσουμε να απλουστεύσουμε την κατάσταση θεωρώντας ότι έχουμε σαν αρχικό όρισμα πραγματικό και συμμετρικό πίνακα καθώς επίσης και ότι $\|\cdot\| = \|\cdot\|_2$.

Έτσι λοιπόν πιο συγκεκριμένα θεωρούμε: $A = A^T \in \mathbb{R}^{m \times m}$, $x \in \mathbb{R}^m$, $x^* = x^T$, $\|x\| = \sqrt{x^T x}$. Αυτή η σύμβαση βέβαια σημαίνει ότι ο A έχει πραγματικές ιδιοτιμές και m ορθογώνια γραμμικώς ανεξάρτητα ιδιοδιανύσματα:

$$\begin{aligned} \text{πραγματικές ιδιοτιμές: } & \lambda_1, \dots, \lambda_m, \\ \text{ορθομοναδιαία ιδιοδιανύσματα: } & x_1, \dots, x_m \end{aligned}$$

Τα ιδιοδιανύσματα θεωρούμε ότι επιλέγονται κανονικοποιημένα, ώστε $\|x_i\| = 1$, και οι ιδιοτιμές διατάσσονται με αύξουσα σειρά.

Οι ιδέες που θα αναπτύξουμε στις επόμενες υποενότητες διατηρούν τις βασικές ιδέες των δύο φάσεων που περιγράψαμε στην υποενότητα II.IV. Αυτό σημαίνει ότι όταν θα εφαρμόσουμε τις επόμενες μεθόδους ο πίνακας A δεν θα είναι απλώς στη μορφή Hessenberg αλλά θα είναι τριδιαγώνιος. Αυτή η τριδιαγώνια μορφή είναι ιδιαίτερα σημαντική από μαθηματική και αλγοριθμική σκοπιά είτε για την επιλογή εναλλαγών στον QR αλγόριθμο είτε για τη μείωση των συνολικών βημάτων και πράξεων που απαιτεί η υλοποίηση ενός αλγορίθμου.

IV.II Πηλίκο Rayleigh

Το πηλίκο Rayleigh ενός διανύσματος $x \in \mathbb{R}^m$ είναι το βαθμωτό μέγεθος

$$r(x) = \frac{x^T A x}{x^T x} \quad (2.11)$$

Σημειώστε ότι αν το x είναι ιδιοδιάνυσμα ενός πίνακα τότε το $r(x) = \frac{x^T A x}{x^T x} = \lambda$ είναι η αντίστοιχη ιδιοτιμή. Ένας τρόπος να καταλήξουμε στο παραπάνω συμπέρασμα είναι να θέσουμε το ερώτημα: Δεδομένου ενός διανύσματος x ποιο βαθμωτό μέγεθος α συμπεριφέρεται σαν ιδιοτιμή αυτού με την έννοια ότι ελαχιστοποιεί το $\|Ax - \alpha x\|_2$; Αυτό είναι ένα $m \times 1$ πρόβλημα ελαχίστων τετραγώνων. Αν γράψουμε τις ομαλές εξισώσεις για το πρόβλημα καταλήγουμε στη λύση: $\alpha = r(x)$. Έτσι το βαθμωτό μέγεθος $r(x)$ είναι μια εκτίμηση της ιδιοτιμής αν θεωρήσουμε ότι το x είναι κοντά αλλά όχι ίσο με ένα ιδιοδιάνυσμα.

Μπορούμε επίσης να εξάγουμε χρήσιμα συμπεράσματα αν θεωρήσουμε την r ως συνάρτηση $\mathbb{R}^m \rightarrow \mathbb{R}$. Μας ενδιαφέρει η τοπική συμπεριφορά της $r(x)$ όταν το x είναι κοντά σε ένα ιδιοδιάνυσμα. Επομένως θα εξετάσουμε τις μερικές παραγώγους της $r(x)$ ως προς τις συντεταγμένες x_j :

$$\begin{aligned} \frac{\partial r(x)}{\partial x_j} &= \frac{\frac{\partial}{\partial x_j}(x^T A x)}{x^T x} - \frac{(x^T A x) \frac{\partial}{\partial x_j}(x^T x)}{(x^T x)^2} \\ &= \frac{2(Ax)_j}{x^T x} - \frac{(x^T A x) 2x_j}{(x^T x)^2} = \frac{2}{x^T x} (Ax - r(x)x)_j \end{aligned}$$

Αν βάλουμε όλες αυτές τις μερικές παραγώγους σε ένα διάνυσμα θα έχουμε υπολογίσει την κλίση της $r(x)$, δηλαδή το $\nabla r(x)$, επομένως δείξαμε ότι:

$$\nabla r(x) = \frac{2}{x^T x} (Ax - r(x)x) \quad (2.12)$$

Επομένως παρατηρούμε ότι αν το x είναι ιδιοδιάνυσμα του πίνακα A τότε η κλίση της $r(x)$ θα είναι το μηδενικό διάνυσμα και αντίστροφα, αν $\nabla r(x) = 0$, με $x \neq 0$, τότε το x είναι ιδιοδιάνυσμα και το $r(x)$ η αντίστοιχη ιδιοτιμή του πίνακα A .

Εξετάζοντάς το γεωμετρικά παρατηρούμε ότι τα ιδιοδιανύσματα του A θα είναι τα στάσιμα σημεία της συνάρτησης $r(x)$, και οι ιδιοτιμές του A είναι η τιμή της $r(x)$ σε αυτά τα στάσιμα σημεία, όπως φαίνεται στο σχήμα 2.1.

Έστω x_j ένα ιδιοδιάνυσμα του A . Επειδή $\nabla r(x_j) = 0$ και η $r(x)$ είναι λεία (παντού εκτός από το $x = 0$) μπορούμε να συμπεράνουμε την εξής σημαντική σχέση:

$$r(x) - r(x_j) = O(\|x - x_j\|^2) \quad \text{καθώς} \quad x \rightarrow x_j \quad (2.13)$$



Σχήμα 2.1: Το πηλίκο Rayleigh $r(x)$ είναι μια συνεχής συνάρτηση πάνω στη σφαίρα $\|x\| = 1$ στο \mathbb{R}^m και τα στάσιμα σημεία της $r(x)$ είναι τα κανονικοποιημένα ιδιοδιανύσματα του πίνακα A . Σε αυτό το παράδειγμα με $m = 3$, υπάρχουν τρία ορθογώνια στάσιμα σημεία.

Έτσι το πηλίκο Rayleigh είναι μια εκτίμηση τετραγωνικής τάξεως ακρίβειας μιας ιδιοτιμής. Αυτή είναι και η ουσία της σημαντικότητάς της.

Ένας τρόπος για να εξάγουμε το αποτέλεσμα 2.13 είναι να αναπτύξουμε το x σαν ένα γραμμικό συνδυασμό των ιδιοδιανυσμάτων x_1, \dots, x_m του πίνακα A . Αν $x = \sum_{i=1}^m \alpha_i x_i$, τότε $r(x) = \frac{\sum_{i=1}^m \alpha_i^2 \lambda_i}{\sum_{i=1}^m \alpha_i^2}$. Οπότε το $r(x)$ είναι ο σταθμικός αριθμητικός μέσος των ιδιοτιμών του A . Έτσι δεν είναι εύκολο να δούμε ότι αν $|\alpha_j/\alpha_J| \leq \varepsilon$ για όλα τα $j \neq J$, τότε $r(x) - r(x_j) = O(\varepsilon^2)$.

IV.III Η μέθοδος των δυνάμεων

Το πηλίκο Rayleigh όπως είδαμε και στην προηγούμενη υποενότητα έχει μεγάλη αλγοριθμική αξία σαν εργαλείο εκτίμησης ιδιοτιμής για κάποιο $x \in \mathbb{R}^m$ κοντά σε ένα ιδιοδιάνυσμα του πίνακα A . Έστω $v^{(0)} \in \mathbb{R}^m$ ένα διάνυσμα με $\|v^{(0)}\| = 1$. Η επόμενη αλγοριθμική διαδικασία αναφέρθηκε ως μία όχι και τόσο καλή ιδέα στην ενότητα II.I. Μπορεί υπό συγκεκριμένες προϋποθέσεις να παράξει μια ακολουθία $v^{(i)}$ που θα συγκλίνει σε ένα ιδιοδιάνυσμα αντίστοιχο της μέγιστης κατά απόλυτη τιμή ιδιοτιμής του πίνακα A .

Αλγόριθμος 2.2

Μέθοδος των δυνάμεων

$v^{(0)}$ = κάποιο διάνυσμα τέτοιο ώστε $\|v^{(0)}\| = 1$

for $k = 1, 2, \dots$

$w = Av^{(k-1)}$

$v^k = w/\|w\|$

$\lambda^{(k)} = (v^{(k)})^T Av^{(k)}$

χρησιμοποιούμε τον A που έχουμε ως όρισμα
κανονικοποιημένες

υπολογίζουμε το πηλίκο Rayleigh

Στον παραπάνω αλγόριθμο δεν δίνουμε λεπτομέρειες για τη συνθήκη

τερματισμού της διαδικασίας. Αυτό θα το εξετάσουμε στο επόμενο κεφάλαιο που θα υλοποιήσουμε τον αλγόριθμο στο MATLAB.

Μπορούμε να αναλύσουμε τον αλγόριθμο των δυνάμεων εύκολα. Γράφουμε το διάνυσμα $v^{(0)}$ ως γραμμικό συνδυασμό των ορθομοναδιαίων ιδιοδιανυσμάτων x_i ως εξής:

$$v^{(0)} = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_m x_m$$

Αφού το $v^{(k)}$ είναι πολλαπλάσιο του $A^k v^{(0)}$, τότε για κάποιες σταθερές c_k θα έχουμε

$$\begin{aligned} v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k (\alpha_1 \lambda_1^k x_1 + \alpha_2 \lambda_2^k x_2 + \dots + \alpha_m \lambda_m^k x_m) \\ &= c_k \lambda_1^k (\alpha_1 x_1 + \alpha_2 (\lambda_2/\lambda_1)^k x_2 + \dots + \alpha_m (\lambda_m/\lambda_1)^k x_m) \end{aligned} \quad (2.14)$$

Έτσι μπορούμε να βγάλουμε το επόμενο συμπέρασμα

Θεώρημα IV.1. Έστωσαν $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_m| \geq 0$ και $x_1^T v^{(0)} \neq 0$. Τότε οι επαναλήψεις του αλγόριθμου 2.2 ικανοποιούν τις εξής σχέσεις:

$$\|v^{(k)} - (\pm x_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \quad (2.15)$$

καθώς $k \rightarrow \infty$. Το πρόσημο \pm σημαίνει ότι σε κάθε βήμα γίνεται η μία ή η άλλη επιλογή.

Απόδειξη. Με βάση το συλλογισμό από το [1] θεώρημα 27.1

Η πρώτη σχέση είναι άμεση συνέπεια της 2.14 αφού $\alpha_1 = x_1^T v^{(0)} \neq 0$ από υπόθεση. Η δεύτερη σχέση προκύπτει από πάλι από την 2.14 και την 2.13. Αν $\lambda_1 > 0$, τότε τα \pm πρόσημα είναι όλα + ή όλα - και αν $\lambda_1 < 0$ το αντίθετο. \square

Τα πρόσημα \pm στην 2.15 δεν είναι και πολύ ελκυστικά. Είναι όμως ένας κομψός τρόπος να αποφύγουμε τις περιπλοκές συζητώντας για σύγκλιση υπόχωρων. Αυτό το ζήτημα δεν θα το αναπτύξουμε καθώς δεν χρειάζεται να μπούμε σε λεπτομέρειες για το πώς η σύγκλιση υπόχωρων μπορεί να γίνει ακριβής.

Από μόνη της η αλγοριθμική αξία της μεθόδου των δυνάμεων έχει σαφή όρια για διάφορους λόγους. Πρώτον μπορεί να υπολογίσει μόνο το ιδιοδιάνυσμα που αντιστοιχεί στην μεγαλύτερη κατά απόλυτη τιμή ιδιοτιμή. Δεύτερον, η σύγκλιση είναι γραμμική μειώνοντας το σφάλμα σε κάθε επανάληψη μόνο κατά $\approx |\lambda_2/\lambda_1|$. Τέλος ο παραπάνω παράγοντας εξαρτάται καθοριστικά από το αν η ιδιοτιμή λ_1 είναι αρκετά μεγαλύτερη από τις υπόλοιπες. Αν οι τιμές των δύο μεγαλύτερων ιδιοτιμών είναι κοντά τότε η σύγκλιση είναι πολύ αργή.

Το πρώτο πρόβλημα που έχει να κάνει με τον υπολογισμό μόνο της μεγαλύτερης κατά απόλυτη τιμή ιδιοτιμής έχει μία σχετικά απλή λύση αν

χρησιμοποιήσουμε την τεχνική της υποτίμησης. Έστω ο συμμετρικός πίνακας $A \in \mathbb{R}^{m \times m}$ με τις πραγματικές ιδιοτιμές

$$\lambda_1 > \lambda_2 > \lambda_3 \geq \lambda_4 \geq \dots \geq \lambda_m.$$

και τα αντίστοιχα ορθομοναδιαία ιδιοδιανύσματα x_i , $i = 1, \dots, m$. Ας υποθέσουμε ότι με τη μέθοδο των δυνάμεων υπολογίσαμε μια προσέγγιση της ιδιοτιμής λ_1 και του ιδιοδιανύσματος x_1 . Έτσι κατασκευάζουμε τον πίνακα

$$A_1 = A - \lambda_1 x_1 x_1^T.$$

Ο πίνακας A_1 έχει ιδιοτιμές τις λ_i με τα αντίστοιχα ιδιοδιανύσματα x_i με $i = 2, 3, \dots, m$ και την 0 με ιδιοδιάνυσμα το x_1 . Πράγματι

$$A_1 x_i = (A - \lambda_1 x_1 x_1^T) x_i = A x_i - \lambda_1 x_1 x_1^T x_i = \lambda_i x_i,$$

αφού $x_1 x_1^T = 0$. Επίσης

$$A - 1x_1 = (A - \lambda_1 x_1 x_1^T) x_1 = A x_1 - \lambda_1 x_1 = 0.$$

Άρα αν εφαρμόσουμε τη μέθοδο των δυνάμεων στον A_2 θα υπολογίσουμε μια προσέγγιση της ιδιοτιμής λ_2 και του ιδιοδιανύσματος x_2 . Φυσικά μπορούμε να συνεχίσουμε τη διαδικασία και να κατασκευάσουμε τον πίνακα $A_2 = A_1 - \lambda_2 x_2 x_2^T$ και να υπολογίσουμε την ιδιοτιμή λ_3 και το ιδιοδιάνυσμα x_3 κοκ.

Αυτή η τεχνική μπορεί να εφαρμοστεί ακόμη και αν η ιδιοτιμή λ_i είναι πολλαπλή και θα έχει σαν αποτέλεσμα τον υποβιβασμό της πολλαπλότητας κατά ένα σε κάθε βήμα. Για παράδειγμα αν η λ_1 είναι πολλαπλή ιδιοτιμή και κατασκευάσουμε τον πίνακα A_2 τότε αυτός θα έχει σαν ιδιοτιμή τη 0 και αντίστοιχο ιδιοδιάνυσμα το x_1 , ενώ αν εφαρμόσουμε τη μέθοδο των δυνάμεων θα υπολογίσουμε πάλι την ιδιοτιμή λ_1 και ένα διαφορετικό ιδιοδιάνυσμα κάθετο με το x_1 . Άρα η τεχνική της υποτίμησης οδηγεί πάντα στον υπολογισμό όλων των ιδιοτιμών και των ιδιοδιανυσμάτων του πίνακα A . Πρέπει να σημειώσουμε όμως ότι πρέπει να δοθεί βάρος στην ακρίβεια των υπολογισμών καθώς τα σφάλματα από τις στρογγυλοποιήσεις συσσωρεύονται από βήμα σε βήμα.

Περαιτέρω λεπτομέρειες για την τεχνική της υποτίμησης στο [8] υποενότητα 8.3.2.

IV.IV Αντίστροφη επανάληψη

Το βασικό πρόβλημα της μεθόδου των δυνάμεων το αναφέραμε στο τέλος τις προηγούμενης υποενότητας και έχει να κάνει με το πόσο μεγαλύτερη είναι η ιδιοτιμή λ_1 από τις υπόλοιπες και ειδικά από τη λ_2 . Προκειμένου να απεμπλακούμε από αυτό το αλγοριθμικό βαρίδι θα χρησιμοποιήσουμε μια ιδιότητα πολύ χρήσιμη στην περίπτωσή μας:

Για κάθε $\mu \in \mathbb{R}$ που δεν είναι ιδιοτιμή του πίνακα A , τα ιδιοδιανύσματα του πίνακα $(A - \mu I)^{-1}$ ταυτίζονται με τα ιδιοδιανύσματα του A και οι αντίστοιχες ιδιοτιμές είναι οι $\{(\lambda_i - \mu)^{-1}\}$ όπου $\{\lambda_i\}$ είναι οι ιδιοτιμές του πίνακα A .

Η παραπάνω πρόταση αυτόματα εισάγει μια ιδέα. Έστω ότι το μ είναι κοντά σε μια ιδιοτιμή λ_J του πίνακα A . Τότε το $(\lambda_J - \mu)^{-1}$ θα είναι πολύ μεγαλύτερο από το $(\lambda_j - \mu)^{-1}$ για κάθε $j \neq J$. Έτσι αν εφαρμόσουμε τη μέθοδο των δυνάμεων στον πίνακα $(A - \mu I)^{-1}$ η διαδικασία υπολογισμού θα συγκλίνει πολύ γρήγορα. Αυτή η ιδέα είναι η βάση υλοποίησης του αλγόριθμου της αντίστροφης επανάληψης.

Αλγόριθμος 2.3

Αντίστροφη επανάληψη

$v^{(0)}$ = κάποιο διάνυσμα τέτοιο ώστε $\|v^{(0)}\| = 1$

for $k = 1, 2, \dots$

$(A - \mu I)w = v^{(k-1)}$

λύνουμε ως προς w

$v_k = w/\|w\|$

κανονικοποιούμε

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$

υπολογίζουμε το πηλίκο Rayleigh

Η αντίστροφη επανάληψη πετυχαίνει μόνο γραμμική σύγκλιση όπως και η μέθοδος των δυνάμεων έχει όμως ισχυρά πλεονεκτήματα. Σε αυτή την αλγοριθμική διαδικασία μπορούμε να επιλέξουμε το ιδιοδιάνυσμα που θέλουμε να υπολογίσουμε, επιλέγοντας μια εκτίμηση της αντίστοιχης ιδιοτιμής. Επιπλέον ο βαθμός της γραμμικής σύγκλισης μπορεί να ελεγχθεί από την ποιότητα του μ , δηλαδή το πόσο κοντά θα βρίσκεται στην ιδιοτιμή που αντιστοιχεί στο ιδιοδιάνυσμα που θέλουμε να υπολογίσουμε. Αν η το μ βρίσκεται πολύ κοντά στη ζητούμενη ιδιοτιμή τότε η μεγαλύτερη κατά απόλυτη τιμή ιδιοτιμή του πίνακα $(A - \mu I)^{-1}$ θα είναι πολύ μεγαλύτερη από τις υπόλοιπες. Επομένως σύμφωνα με αυτά που αναπτύξαμε και στην προηγούμενη υποενότητα η σύγκλιση θα είναι πολύ πιο γρήγορη.

Θεώρημα IV.2. Έστω ότι η ιδιοτιμή λ_J είναι η κοντινότερη στο μ και η λ_K είναι η δεύτερη κοντινότερη έτσι ώστε $|\mu - \lambda_J| < |\mu - \lambda_K| \leq |\mu - \lambda_j|$ για κάθε $j \neq J$. Επιπλέον έστω $x_J^T v^{(0)} \neq 0$. Τότε οι επαναλήψεις του αλγόριθμου 2.3 ικανοποιούν τις εξής σχέσεις:

$$\|v^{(k)} - (\pm x_J)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \quad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right) \quad (2.16)$$

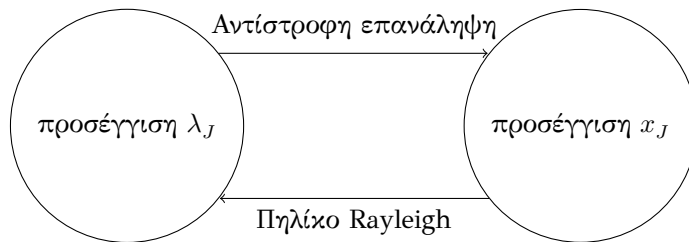
καθώς $k \rightarrow \infty$, και τα πρόσημα \pm έχουν την ίδια σημασία με του Θεωρήματος IV.1.

Για περισσότερες λεπτομέρειες στο [1] θεώρημα 27.2

Η αντίστροφη επανάληψη είναι από τα πιο σημαντικά εργαλεία της αριθμητικής γραμμικής άλγεβρας και αυτό γιατί μπορεί να υπολογίσει όσα ιδιοδιανύσματα ενός πίνακα θέλουμε αρκεί να ξέρουμε τις αντίστοιχες ιδιοτιμές.

IV.V Επανάληψη πηλίκου Rayleigh

Μέχρι στιγμής σε αυτή την ενότητα έχουμε παρουσιάσει δύο μεθόδους. Η πρώτη (πηλίκου Rayleigh) υπολογίζει μια ιδιοτιμή από μια προσέγγιση του αντίστοιχου ιδιοδιανύσματος ενώ η δεύτερη μέθοδος (αντίστροφη επανάληψη) υπολογίζει ένα ιδιοδιάνυσμα από μια προσέγγιση της αντίστοιχης ιδιοτιμής. Ο συνδυασμός αυτών των δύο μεθόδων έχει μεγάλα πλεονεκτήματα και είναι σχετικά εύκολο να επιτευχθεί.



Το παραπάνω σχήμα είναι υπεραπλουστευμένο καθώς για μια προσέγγιση της ιδιοτιμής λ_J απαιτείται μια αρχική προσέγγιση του αντίστοιχου ιδιοδιανύσματος και το αντίστροφο, παρόλα αυτά μας δίνει μια ποιοτική εικόνα του αλγορίθμου που θα παρουσιάσουμε. Η βασική ιδέα είναι να χρησιμοποιήσουμε μια συνεχώς βελτιωμένη προσέγγιση της ζητούμενης ιδιοτιμής σε κάθε επανάληψη και να αυξήσουμε το βαθμό σύγκλισης της αντίστροφης επανάληψης σε κάθε επανάληψη. Αυτός ο αλγόριθμος ονομάζεται *Επανάληψη πηλίκου Rayleigh*.

Αλγόριθμος 2.4

Επανάληψη πηλίκου Rayleigh

$v^{(0)}$ = κάποιο διάνυσμα τέτοιο ώστε $\|v^{(0)}\| = 1$

$\lambda^{(0)}$ = $(v^{(0)})^T A v^{(0)}$ → Το αντίστοιχο πηλίκου Rayleigh

for $k = 1, 2, \dots$

$(A - \lambda^{(k-1)} I)w = v^{(k-1)}$

λύνουμε ως προς w

$\begin{aligned} v_k &= w/\ w\ \\ \lambda^{(k)} &= (v^{(k)})^T A v^{(k)} \end{aligned}$	κανονικοποιούμε υπολογίζουμε το πηλίκο Rayleigh
--	--

Η σύγκλιση αυτού του αλγόριθμου είναι θεαματική: Τριπλασιάζει τον αριθμό των ψηφίων της ακρίβειας σε κάθε επανάληψη.

Θεώρημα IV.3. Ο αλγόριθμος 2.4 συγκλίνει σε ένα ζεύγος ιδιοτιμής-ιδιοδιανύσματος για όλα τα αρχικά διανύσματα $v^{(0)}$ εκτός από τα μη-δενικού μέτρου. Όταν συγκλίνει, η σύγκλιση είναι κυβική με την έννοια ότι αν το λ_J είναι μία ιδιοτιμή του πίνακα A και το διάνυσμα $v^{(0)}$ είναι αρκετά κοντά στο αντίστοιχο ιδιοδιάνυσμα x_J , τότε

$$\|v^{(k+1)} - (\pm x_J)\| = O(\|v^{(k)} - (\pm x_J)\|^3) \quad (2.17)$$

και

$$|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3) \quad (2.18)$$

Καθώς $k \rightarrow \infty$. Τα πρόσημα \pm δεν είναι απαραίτητα τα ίδια και στα δύο σκέλη της σχέσης 2.17.

Απόδειξη. Με βάση το [1] θεώρημα 27.3

Δεν θα αποδείξουμε τον ισχυρισμό της σύγκλισης για σχεδόν όλα τα αρχικά διανύσματα $v^{(0)}$. Θα αποδείξουμε ότι αν η σύγκλιση επιτευχθεί τότε είναι κυβική. Υποθέτουμε χωρίς βλάβη της γενικότητας ότι η ιδιοτιμή λ_J είναι απλή. Αν $\|v^{(k)} - x_J\| \leq \epsilon$ για αρκετά μικρό ϵ , τότε από την 2.13 προκύπτει ότι το πηλίκο Rayleigh μας δίνει μια προσέγγιση $\lambda^{(k)}$ της ιδιοτιμής με $|\lambda^{(k)} - \lambda_J| = O(\epsilon^2)$. Σύμφωνα με το θεώρημα IV.2 αν υλοποιήσουμε ένα βήμα της αντίστροφης επανάληψης για να πάρουμε το $v^{(k+1)}$ θα έχουμε:

$$\|v^{(k+1)} - x_J\| = O(|\lambda^{(k)} - \lambda_J| \|v^{(k)} - x_J\|) = O(\epsilon^3)$$

Επιπλέον οι σταθερές των συμβόλων O είναι ίδιες σε όλες τις αρκετά μικρές περιοχές των λ_J και x_J . Έτσι η σύγκλιση σχηματικά έχει ως εξής:

$$\begin{array}{l} \|v^{(k)} - (\pm x_J)\| \quad |\lambda^{(k)} - \lambda_J| \\ \epsilon \rightarrow O(\epsilon^2) \\ \downarrow \checkmark \\ O(\epsilon^3) \rightarrow O(\epsilon^6) \\ \downarrow \checkmark \\ O(\epsilon^9) \rightarrow O(\epsilon^{18}) \\ \vdots \quad \quad \quad \vdots \end{array}$$

Και έτσι προκύπτουν οι σχέσεις του Θεωρήματος. □

Παράδειγμα 2.1. Η κυβική σύγκλιση είναι τόσο γρήγορη που θα δώσουμε ένα αριθμητικό παράδειγμα. Έστω ο πίνακας

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix}$$

και έστω $v^{(0)} = (1, 1, 1)^T / \sqrt{3}$ η αρχική τιμή προσέγγισης του ιδιοδιανύσματος. Όταν η επανάληψη πηλίκου Rayleigh εκτελεστεί για τον πίνακα A προκύπτουν οι παρακάτω υπολογισμοί των τιμών $\lambda^{(k)}$:

$$\lambda^{(0)} = 5, \quad \lambda^{(1)} = 5.2131\dots, \quad \lambda^{(2)} = 5.214319743184\dots$$

Η ακριβής τιμή της αντίστοιχης στο πιο κοντινό στο $v^{(0)}$ ιδιοδιάνυσμα, ιδιοτιμής είναι $\lambda = 5.214319743377$. Μόλις μετά από τρεις επαναλήψεις ο αλγόριθμός μας συνέκλινε με ακρίβεια δέκα ψηφίων. Τρεις ακόμα επαναλήψεις θα αυξήσουν την ακρίβεια γύρω στα 270 ψηφία αν η ακρίβεια μηχανής είναι τόσο μεγάλη.

IV.VI Ταχύτητα σύγκλισης

Σε αυτή την υποενότητα θα δούμε το σύνολο της δουλειάς που απαιτεί κάθε επανάληψη των τριών αλγοριθμικών διαδικασιών που αναπτύξαμε στις προηγούμενες υποενότητες.

Έστω ότι ο πίνακας $A \in \mathbb{R}^{m \times m}$ είναι πλήρης. Τότε κάθε βήμα της μεθόδου των δυνάμεων περιλαμβάνει ένα πολλαπλασιασμό πίνακα με διάνυσμα, διαδικασία που απαιτεί $O(m^2)$ πράξεις. Κάθε βήμα της "αντίστροφης επανάληψης" περιλαμβάνει μια λύση ενός γραμμικού συστήματος το οποίο απαιτεί συνήθως μικρότερης τάξης πράξεις από το $O(m^3)$ αν χρησιμοποιηθούν οι μέθοδοι LU, QR παραγοντοποίηση ή κάποιες άλλες και καταλήγει να απαιτεί $O(m^2)$ πράξεις. Στην περίπτωση της "επανάληψης πηλίκου Rayleigh" ο πίνακας που αντιστρέφεται σε κάθε επανάληψη αλλάζει οπότε η μείωση του $O(m^3)$ δεν είναι δεδομένη σε κάθε επανάληψη.

Όταν ο πίνακός μας όμως είναι τριδιαγώνιος οι δουλειά που απαιτείται και στις τρεις περιπτώσεις μειώνεται αρκετά. Και οι τρεις αλγοριθμικές διαδικασίες απαιτούν $O(m)$ πράξεις σε κάθε βήμα. Όταν ο πίνακός μας είναι Hessenberg οι απαιτούμενες πράξεις αυξάνονται στις $O(m^2)$ σε κάθε βήμα για κάθε αλγοριθμική διαδικασία.

V Αλγόριθμος QR χωρίς εναλλαγές

Ο αλγόριθμος QR συναντάται για πρώτη φορά στις αρχές της δεκαετίας του 60' και μας έδωσε τη δυνατότητα να προσεγγίσουμε προβλήματα ιδιοτιμών που μέχρι τότε θεωρούνταν απλησίαστα. Εδώ θα τον παρουσιάσουμε στην πιο απλή μορφή του. Μπορεί επίσης να εξετασθεί από τη σκοπιά μιας ευσταθούς διαδικασίας υπολογισμού της QR παραγοντοποίησης των δυνάμεων του αρχικού πίνακα A, A^2, A^3, \dots .

V.I Ο αλγόριθμος QR

Η πιο απλή εκδοχή του αλγόριθμου QR μοιάζει ιδιαίτερα απλή.

Αλγόριθμος 2.5

Αλγόριθμος QR

$$A^{(0)} = A$$

for $k = 1, 2, \dots$

$$Q^{(k)} R^{(k)} = A^{(k-1)}$$

$$A^{(k)} = R^{(k)} Q^{(k)}$$

QR παραγοντοποίηση του $A^{(k-1)}$

Αλλάζουμε σειρά στους παράγοντες

Αυτό που κάνει ο παραπάνω αλγόριθμος είναι να πολλαπλασιάζει τους παράγοντες Q και R με την αντίστροφη σειρά RQ και να επαναλαμβάνει αυτή τη διαδικασία. Κάτω από συγκεκριμένες προϋποθέσεις, αυτός ο απλός αλγόριθμος συγκλίνει στη μορφή Schur του A (άνω-τριγωνικό αν ο A είναι τυχαίος, διαγώνιος αν ο A είναι ερμιτιανός). Σε αυτή την ενότητα θα συνεχίσουμε κάτω από την υπόθεση ότι ο A είναι πραγματικός, συμμετρικός, με πραγματικές ιδιοτιμές και ορθομοναδιαία ιδιοδιανύσματα. Γι' αυτό το ενδιαφέρον μας στρέφεται στην σύγκλιση της ακολουθίας των πινάκων $A^{(k)}$ σε διαγώνια μορφή.

Φυσικά προκειμένου να εξάγουμε τις ιδιοτιμές από τη διαγώνια τελική μορφή θα πρέπει ο συνολικός μετασχηματισμός να είναι μετασχηματισμός ομοιότητας. Αυτό δείχνεται εύκολα ως εξής: έχουμε $A^{(k)} = R^{(k)} Q^{(k)}$, όμως $R^{(k)} Q^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$, άρα και $A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$. Αυτό ίσως μας θυμίζει την κακή ιδέα που είδαμε στην ενότητα III περί αδιαμεσολάβητου μετασχηματισμού ομοιότητας σε μορφή Schur όμως εδώ έχουμε μια επαναληπτική διαδικασία και όχι ένα άμεσο καθορισμένο απόλυτα βήμα μετασχηματισμού του A σε μορφή Schur. Στην πράξη πρόκειται για μια αρκετά ισχυρή βάση για επαναληπτικές διαδικασίες.

Όπως και η μέθοδος επανάληψης πηλίκου Rayleigh, ο αλγόριθμος QR για πραγματικούς συμμετρικούς πίνακες συγκλίνει κυβικά. Για να το πετύχουμε

αυτό όμως πρέπει να τροποποιήσουμε με κατάλληλο τρόπο τον αλγόριθμό μας. Η βασική τροποποίηση είναι να εισάγουμε εναλλαγές σε κάθε βήμα ώστε σε συνδυασμό με άλλες δύο τροποποιήσεις να έχουμε ένα πρακτικό αλγόριθμο κρατώντας την αρχική ιδέα:

1. Πριν ξεκινήσουμε την επανάληψη ο A μετασχηματίζεται σε τριδιαγώνιο όπως περιγράφουμε στην ενότητα III.
2. Αντί για τον $A^{(k)}$ σε κάθε βήμα παραγοντοποιούμε τον $A^{(k)} - \mu^{(k)}I$.
3. Όποτε είναι δυνατό, και συγκεκριμένα όταν βρίσκουμε ιδιοτιμή το πρόβλημα ανάγεται σε απλούστερα σπάζοντας τον $A^{(k)}$ σε υποπίνακες.

Αν εφαρμόσουμε αυτές τις αλλαγές ο αλγόριθμος QR θα πάρει την παρακάτω μορφή.

Αλγόριθμος 2.6

Πρακτικός αλγόριθμος QR

$(Q^{(0)})^T A^{(0)} Q^{(0)} = A$ ο $A^{(0)}$ η τριδιαγωνοποίηση του A

for $k = 1, 2, \dots$

Επιλέγουμε μια εναλλαγή $\mu^{(k)}$ Π.χ. επιλέγουμε $\mu^{(k)} = A_{mm}^{(k-1)}$

$Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)}I$ QR παραγοντοποίηση του $A^{(k-1)} - \mu^{(k)}I$

$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)}I$ αλλάζουμε σειρά στους παράγοντες

Αν κάποιο μη-διαγώνιο στοιχείο του $A_{j,j+1}^{(k)}$ είναι αρκετά κοντά στο μηδέν, θέτουμε $A_{j,j+1} = A_{j+1,j} = 0$ και έχουμε

$$A^{(k)} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$$

εκτελούμε τον αλγόριθμο QR για τους πίνακες A_1, A_2

Ο παραπάνω αλγόριθμος με καλά επιλεγμένες εναλλαγές ήταν η βασική μέθοδος υπολογισμού ιδιοτιμών ενός πίνακα από τις αρχές της δεκαετίας του 60' μέχρι τη δεκαετία του 90' όπου εμφανίστηκε ο αλγόριθμος *Διαίρει και βασίλευε* που θα παρουσιάσουμε σε επόμενη ενότητα.

Προς το παρόν θα επικεντρωθούμε στον απλό αλγόριθμο QR και θα δούμε πώς υπολογίζει ιδιοτιμές.

V.II Μη κανονικοποιημένες ταυτόχρονες επαναλήψεις

Η προσέγγιση που θα ακολουθήσουμε είναι να μπορέσουμε να συσχετίσουμε τον αλγόριθμο QR με μια άλλη μέθοδο που έχει πιο προφανή συμπεριφορά και λέγεται *Μη κανονικοποιημένες ταυτόχρονες επαναλήψεις*.

Η ιδέα της μεθόδου των μη κανονικοποιημένων ταυτόχρονων επαναλήψεων είναι να εφαρμόσει τη μέθοδο των δυνάμεων σε πολλά διανύσματα με τη μία. Έστω ότι ξεκινάμε με ένα σύνολο n γραμμικών ανεξάρτητων διανυσμάτων $v^{(0)}, \dots, v_n^{(0)}$. Είναι λογικό καθώς το $A^k v_1^{(0)}$ συγκλίνει όταν $k \rightarrow \infty$ (κάτω από συγκεκριμένες προϋποθέσεις) στο αντίστοιχο ιδιοδιάνυσμα της μεγαλύτερης κατά απόλυτη τιμή ιδιοτιμή του A , ο χώρος $\langle A^k v_1^{(0)}, \dots, A^k v_n^{(0)} \rangle$ θα συγκλίνει (κάτω από συγκεκριμένες προϋποθέσεις) στο χώρο $\langle x_1, \dots, x_n \rangle$ που παράγεται από τα ιδιοδιανύσματα x_1, \dots, x_n του A που αντιστοιχούν στις n μεγαλύτερες κατά απόλυτη τιμή ιδιοτιμές του.

Για να δούμε και το παραπάνω σχηματικά, θεωρούμε $V^{(0)} \in \mathbb{R}^{m \times n}$ τον αρχικό μας πίνακα

$$V^{(0)} = \left[\begin{array}{c|c|c} v_1^{(0)} & \dots & v_n^{(0)} \end{array} \right] \quad (2.19)$$

Και έστω $V^{(k)}$ είναι το αποτέλεσμα μετά από k πολλαπλασιασμούς του A :

$$V^{(k)} = A^k V^{(0)} = \left[\begin{array}{c|c|c} v_1^{(k)} & \dots & v_n^{(k)} \end{array} \right] \quad (2.20)$$

Από τη στιγμή που το ενδιαφέρον μας έχει να κάνει με το χώρο που παράγουν οι στήλες του $V^{(k)}$, μπορούμε να εξάγουμε μια βάση για αυτό το χώρο αν υπολογίσουμε την απλή παραγοντοποίηση QR του πίνακα $V^{(k)}$:

$$\hat{Q}^{(k)} \hat{R}^{(k)} = V^{(k)} \quad (2.21)$$

Οι πίνακες $\hat{Q}^{(k)}$ και $\hat{R}^{(k)}$ είναι διαστάσεων $m \times n$ και $n \times n$ αντίστοιχα. Φαίνεται λογική η υπόθεση ότι καθώς $k \rightarrow \infty$ οι διαδοχικές στήλες του πίνακα $\hat{Q}^{(k)}$ συγκλίνουν στα ιδιοδιανύσματα $\pm x_1, \pm x_2, \dots, \pm x_n$ (κάτω από συγκεκριμένες προϋποθέσεις).

Αυτή η εκτίμηση μπορεί ναδειχθεί ότι ισχύει αν ακολουθήσουμε την ανάλυση του προβλήματος που κάναμε και στην προηγούμενη ενότητα. Αναπτύσσουμε τα $v_j^{(0)}, v_j^{(0)}$ σαν γραμμικό συνδυασμό των ιδιοδιανυσμάτων του A και έτσι έχουμε:

$$\begin{aligned} v_j^{(0)} &= \alpha_{1j} x_1 + \dots + \alpha_{mj} x_m \\ v_j^{(k)} &= \lambda_1^k \alpha_{1j} x_1 + \dots + \lambda_m^k \alpha_{mj} x_m \end{aligned}$$

Η σύγκλιση της παραπάνω διαδικασίας επιτυγχάνεται αν ικανοποιούνται δύο προϋποθέσεις. Η πρώτη προϋπόθεση είναι ότι οι πρώτες $n+1$ μεγαλύτερες κατά απόλυτη τιμή ιδιοτιμές είναι διακεκριμένες:

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \dots \geq |\lambda_m| \quad (2.22)$$

Η δεύτερη προϋπόθεση είναι ο $n \times m$ πίνακας των συντελεστών α_{ij} να είναι κατά μία έννοια αντιστρέψιμος και θα πούμε αμέσως τι εννοούμε με αυτό. Έστω $\hat{Q} \in \mathbb{R}^{m \times n}$ ο πίνακας του οποίου οι στήλες είναι τα ιδιοδιανύσματα x_1, x_2, \dots, x_n . (Επομένως ο πίνακας \hat{Q} των ιδιοδιανυσμάτων είναι τελείως διαφορετικός από τον $\hat{Q}^{(k)}$ που είναι ένας παράγοντας μια απλής QR παραγοντοποίησης). Η δεύτερη προϋπόθεση, λοιπόν, είναι η ακόλουθη:

$$\text{Όλοι οι οδηγιοί ελάχιστονες πίνακες του } \hat{Q}^T V^{(0)} \text{ να είναι αντιστρέψιμοι} \quad (2.23)$$

Όταν λέμε "οδηγιοί ελάχιστονες πίνακες του $\hat{Q}^T V^{(0)}$ " εννοούμε τους άνω-αριστερά υποπίνακες διαστάσεων $1 \times 1, 2 \times 2, \dots, n \times n$ του $\hat{Q}^T V^{(0)}$.

Θεώρημα V.1. Έστω ότι η επαναληπτική διαδικασία 2.19-2.21 εκτελείται και οι προϋποθέσεις 2.22 και 2.23 ικανοποιούνται. Τότε καθώς $k \rightarrow \infty$ οι στήλες των πινάκων $\hat{Q}^{(k)}$ συγκλίνουν γραμμικά στα ιδιοδιανύσματα του A :

$$\|x_j^{(k)} - (\pm x_j)\| = O(C^k) \quad (2.24)$$

για κάθε $1 \leq j \leq n$, όπου $C < 1$ είναι η σταθερά $C = \max_{1 \leq k \leq n} (|\lambda_{k+1}|/|\lambda_k|)$. Όπως και στα Θεωρήματα της προηγούμενης ενότητας, τα πρόσημα \pm υπονοούν ότι σε κάθε k -οστό βήμα η μία ή η άλλη επιλογή γίνεται και το παραπάνω φράγμα ισχύει.

Απόδειξη. Με βάση το [1] θεώρημα 28.1

Επεκτείνουμε τον \hat{Q} σε ένα πλήρη $m \times m$ ορθογώνιο πίνακα Q των ιδιοδιανυσμάτων του A και έστω ότι ο Λ είναι ο αντίστοιχος διαγώνιος πίνακας των ιδιοτιμών, έτσι ώστε $A = Q\Lambda Q^T$. Παρά τηρούμε ότι ο πίνακας \hat{Q} είναι ο οδηγός $m \times n$ άνω-αριστερά υποπίνακας του Q και ο (διαγώνιος) $\hat{\Lambda}$ είναι ο οδηγός $n \times n$ άνω-αριστερά υποπίνακας του Λ . Έτσι έχουμε:

$$V^{(k)} = A^k V^{(0)} = Q\Lambda^k Q^T V^{(0)} = \hat{Q}\hat{\Lambda}^k \hat{Q}^T V^{(0)} + O(|\lambda_{n+1}|^k)$$

Καθώς $k \rightarrow \infty$. Αν η ισχύει η 2.23 τότε ο $\hat{Q}^T V^{(0)}$ είναι αντιστρέψιμος και έτσι μπορούμε να πολλαπλασιάσουμε τον όρο $O(|\lambda_{n+1}|^k)$ από τα δεξιά με $(\hat{Q}^T V^{(0)})^{-1} \hat{Q}^T V^{(0)}$ και να μετασχηματίσουμε την παραπάνω σχέση ως εξής:

$$V^{(k)} = \left(\hat{Q}\hat{\Lambda}^k + O(|\lambda_{n+1}|^k) \right) \hat{Q}^T V^{(0)}$$

Αφού ο $\hat{Q}^T V^{(0)}$ είναι αντιστρέψιμος, ο χώρος που παράγουν οι στήλες του θα είναι ίδιος με το χώρο που παράγουν οι στήλες του

$$\hat{Q}\hat{\Lambda}^k + O(|\lambda_{n+1}|^k)$$

Από τη μορφή του $\hat{Q}\hat{\Lambda}^k$ και την προϋπόθεση 2.22 εύκολα προκύπτει ότι αυτός ο χώρος συγκλίνει γραμμικά σε αυτόν που παράγουν οι στήλες του

\hat{Q} . Δεν θα παρουσιάσουμε τις λεπτομέρειες αυτής της σύγκλισης μεταξύ των δύο χώρων.

Η προϋπόθεση 2.23 δεν ισχύει μόνο για τον πίνακα $\hat{Q}^T V^{(0)}$ αλλά και για όλους τους οδηγούς ελάσσονες πίνακες του. Επομένως το παραπάνω επιχείρημα περί σύγκλισης των χώρων που παράγονται από τις στήλες των εν λόγω πινάκων ισχύει και για τα οδηγιά υποσύνολα των στηλών των $V^{(k)}$ και \hat{Q} : Για τις πρώτες τους στήλες, τις πρώτες και τις δεύτερες, τις πρώτες και τις δεύτερες και τις τρίτες κτλ. Σε κάθε περίπτωση διαπιστώνουμε ότι ο χώρος που παράγεται από διαδοχικές στήλες του $V^{(k)}$ συγκλίνει στο χώρο που παράγουν οι αντίστοιχες στήλες του \hat{Q} . Από το παραπάνω και τον ορισμό της QR παραγοντοποίησης προκύπτουν τα 2.21 και 2.24. \square

V.III Ταυτόχρονες Επαναλήψεις

Καθώς $k \rightarrow \infty$ τα διανύσματα $v_1^{(k)}, \dots, v_n^{(k)}$ στον αλγόριθμο 2.19-2.21 συγκλίνουν όλα σε πολλαπλάσια του "κυρίαρχου" ιδιοδιανύσματος x_1 του A . Έτσι, παρόλο που ο χώρος που παράγεται $\langle v_1^{(k)}, \dots, v_j^{(k)} \rangle$ συγκλίνει σε κάτι χρήσιμο, αυτά τα διανύσματα συνιστούν μια όχι καλής κατάστασης βάση αυτού του χώρου. Αυτό σημαίνει ότι αν εκτελέσουμε τη μέθοδο της μη κανονικοποιημένης ταυτόχρονης επανάληψης όπως περιγράφηκε στα 2.19-2.21 η επιθυμητή πληροφορία θα χαθεί στις αριθμητικές στρογγυλοποιήσεις.

Η λύση είναι απλή: Μία ορθοκανονικοποίηση σε κάθε βήμα παρά μία για όλα τα βήματα. Έτσι δεν θα κατασκευάσουμε τους πίνακες $V^{(k)}$ όπως περιγράψαμε προηγουμένως αλλά μια διαφορετική ακολουθία πινάκων $Z^{(k)}$ των οποίων οι στήλες παράγουν τον ίδιο χώρο.

Αλγόριθμος 2.7

Αλγόριθμος ταυτόχρονων επαναλήψεων

Διαλέγουμε $\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$ με ορθομοναδιαίες στήλες

for $k = 1, 2, \dots$

$$Z = A\hat{Q}^{(k-1)}$$

$$\hat{Q}^{(k)}\hat{R}^{(k)} = Z$$

απλή QR παραγοντοποίηση του Z

Από τη μορφή του παραπάνω αλγόριθμου εύκολα συμπεραίνουμε ότι οι χώροι που παράγουν οι στήλες των πινάκων $\hat{Q}^{(k)}$ και $Z^{(k)}$ ταυτίζονται με το χώρο που παράγουν οι στήλες του πίνακα $A^k \hat{Q}^{(0)}$. Έτσι από μαθηματικής άποψης αυτή η νέα μορφή του αλγόριθμου ταυτόχρονων επαναλήψεων συγκλίνει κάτω από τις ίδιες προϋποθέσεις του προηγούμενου.

Θεώρημα V.2. Ο αλγόριθμος 2.7 παράγει τους ίδιους πίνακες $\hat{Q}^{(k)}$ όπως ο αλγόριθμος που ορίζεται από τις επαναλήψεις 2.19-2.21 που εξετά-

σαμε και στο Θεώρημα V.1 (θεωρώντας ότι οι αρχικοί πίνακες $\hat{Q}^{(0)}$ είναι ίδιοι), κάτω από τις ίδιες προϋποθέσεις 2.22 και 2.23 και συγκλίνει όπως περιγράφει και το Θεώρημα V.1.

Περισσότερες λεπτομέρειες για το θεώρημα V.2 στο [1] θεώρημα 28.2.

V.IV Ισοδυναμία αλγόριθμου QR και ταυτόχρονων επαναλήψεων

Τώρα μπορούμε να εξηγήσουμε περαιτέρω τη συμπεριφορά του αλγόριθμου QR. Είναι ισοδύναμος με τον αλγόριθμο των ταυτόχρονων επαναλήψεων αν επιλέξουμε σαν αρχικό πίνακα αυτόν που κατασκευάζεται από ένα πλήρες σύνολο $n = m$ διανυσμάτων, δηλαδή, τον $\hat{Q}^{(0)} = I$. Αφού οι πίνακες $\hat{Q}^{(k)}$ θα είναι τετραγωνικοί θα μπορούμε να εφαρμόζουμε την πλήρη QR παραγοντοποίηση. Έτσι θα αντικαταστήσουμε τους $\hat{R}^{(k)}$ και $\hat{Q}^{(k)}$ με τους $R^{(k)}$ και $Q^{(k)}$ αντίστοιχα ώστε να ξεχωρίζουμε τους πίνακες του αλγόριθμου των ταυτόχρονων επαναλήψεων και αυτού του αλγόριθμου QR.

Παρακάτω παραθέτουμε τρεις σχέσεις που ορίζουν τον αλγόριθμο ταυτόχρονων επαναλήψεων με $Q^{(0)} = I$ και μία τέταρτη σχέση που μας δίνει τους πίνακες $A^{(k)} \in \mathbb{R}^{m \times m}$:

Ταυτόχρονες επαναλήψεις:

$$Q^{(0)} = I \quad (2.25)$$

$$Z = A Q^{(k-1)} \quad (2.26)$$

$$Z = Q^{(k)} R^{(k)} \quad (2.27)$$

$$A^{(k)} = (Q^{(k)})^T A Q^{(k)} \quad (2.28)$$

Ακολουθούν τρεις σχέσεις που ορίζουν τον αλγόριθμο QR χωρίς εναλλαγές και μία τέταρτη σχέση που ορίζουμε τον πίνακα $Q^{(k)} \in \mathbb{R}^{m \times m}$:

Αλγόριθμος QR χωρίς εναλλαγές:

$$A^{(0)} = A \quad (2.29)$$

$$A^{(k-1)} = Q^{(k)} R^{(k)} \quad (2.30)$$

$$A^{(k)} = R^{(k)} Q^{(k)} \quad (2.31)$$

$$Q^{(k)} = Q^{(1)} Q^{(2)} \dots Q^{(k)} \quad (2.32)$$

Επιπλέον και για τους δύο αλγόριθμους μπορούμε να ορίσουμε τον πίνακα $R^{(k)} \in \mathbb{R}^{m \times m}$

$$\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \dots R^{(1)} \quad (2.33)$$

Τώρα μπορούμε σχετικά εύκολα να εξάγουμε την ισοδυναμία μεταξύ των δύο αλγορίθμων.

Θεώρημα V.3. Οι διαδικασίες 2.25-2.28 και 2.29-2.32 δημιουργούν ακριβώς τις ίδιες ακολουθίες πινάκων $\underline{R}^{(k)}, \underline{Q}^{(k)}$ και $A^{(k)}$ δηλαδή αυτές που ορίζονται από την QR παραγοντοποίηση της k -οστής δύναμης του A

$$A^k = \underline{Q}^{(k)} \underline{R}^{(k)} \quad (2.34)$$

και την ακολουθία πινάκων

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)} \quad (2.35)$$

Απόδειξη. Με βάση το [1] θεώρημα 28.3

Θα κάνουμε την απόδειξη με τη μέθοδο της επαγωγής. Η περίπτωση $k = 0$ είναι τετριμμένη. Και για τους δύο αλγόριθμους από τις σχέσεις 2.29-2.33 παίρνουμε $A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I$ και $A^{(0)} = A$ και οι 2.34 και 2.35 επαληθεύονται εύκολα.

Θεωρούμε την περίπτωση $k \geq 1$ για τον αλγόριθμο ταυτόχρονων επαναλήψεων. Η σχέση 2.35 επαληθεύεται από τη σχέση 2.28 καθώς ταυτίζονται, επομένως θα επαληθεύσουμε μόνο τη σχέση 2.34 όπως παρακάτω

$$A^k = A \underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

Η πρώτη ισότητα προκύπτει από την επαγωγική υπόθεση της 2.34, η δεύτερη από τις 2.26 και 2.27 και η τρίτη από την 2.33.

Θεωρούμε επίσης την περίπτωση $k \geq 1$ για τον αλγόριθμο QR. Μπορούμε να επαληθεύσουμε την 2.34 με τα παρακάτω

$$A^k = A \underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} A^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}$$

Η πρώτη ισότητα προκύπτει από την επαγωγική υπόθεση της 2.34, η δεύτερη από την επαγωγική υπόθεση της 2.35 και η τρίτη από τις 2.30, 2.32 και 2.33. Επίσης επαληθεύουμε και την 2.35 με τα παρακάτω

$$A^{(k)} = (\underline{Q}^{(k)})^T A^{(k-1)} \underline{Q}^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$$

Η πρώτη ισότητα προκύπτει από την 2.30 και την 2.31 και η δεύτερη από την επαγωγική υπόθεση της 2.35 \square

V.V Σύγκλιση του αλγόριθμου QR

Πλέον είμαστε στη θέση να βγάλουμε κάποια ποιοτικά συμπεράσματα για τον αλγόριθμο QR χωρίς εναλλαγές. Το κλειδί για την ποιοτική κατανόηση της συμπεριφοράς του αλγόριθμου QR είναι οι σχέσεις 2.34 και 2.35.

Η πρώτη εξηγεί γιατί ο αλγόριθμος βρίσκει ιδιοδιανύσματα: Κατασκευάζει ορθομοναδιαίες βάσεις για διαδοχικές δυνάμεις του πίνακα A . Η δεύτερη εξηγεί γιατί ο αλγόριθμος βρίσκει ιδιοτιμές. Από την 2.35 συμπεραίνουμε ότι τα διαγώνια στοιχεία του $A^{(k)}$ είναι πηλίκα Rayleigh των στηλών του $Q^{(k)}$ εφαρμόζοντάς τα για τον πίνακα A . Καθώς αυτές οι στήλες συγκλίνουν στα ιδιοδιανύσματα, τα πηλίκα Rayleigh συγκλίνουν (και μάλιστα δύο φορές πιο γρήγορα όπως έχουμε δει) στις αντίστοιχες ιδιοτιμές. Επίσης η σχέση 2.35 μας δείχνει ότι τα μη διαγώνια στοιχεία του $A^{(k)}$ συγκλίνουν στο μηδέν και αυτό για τον εξής λόγο: καθώς οι στήλες του $Q^{(k)}$ συγκλίνουν στα ορθομοναδιαία ιδιοδιανύσματα τα μη διαγώνια στοιχεία του $A^{(k)}$ αναγκαστικά συγκλίνουν στο μηδέν.

Ένα ακόμα θεώρημα που μπορούμε να πούμε ότι είναι συνέπεια του Θεωρήματος V.2 και ολοκληρώνει μια ποιοτική κατανόηση του αλγόριθμου QR είναι το παρακάτω.

Θεώρημα V.4. Έστω ότι ο αλγόριθμος 2.5 εκτελείται για ένα πραγματικό, συμμετρικό πίνακα A του οποίου οι ιδιοτιμές έχουν την ιδιότητα $|\lambda_1| > |\lambda_2| > \dots > |\lambda_m|$ και ο αντίστοιχος πίνακας ιδιοδιανυσμάτων Q του οποίου όλοι οι οδηγοί ελάσσονες πίνακες του είναι αντιστρέψιμοι. Τότε καθώς $k \rightarrow \infty$, ο $A^{(k)}$ συγκλίνει γραμμικά με σταθερά $C = \max_k (|\lambda_{k+1}|/|\lambda_k|)$ στο διαγώνιο πίνακα $\text{diag}(\lambda_1, \dots, \lambda_m)$ και ο πίνακας $Q^{(k)}$ (με τα πρόσθημα των στηλών του προσαρμοσμένα ανάλογα) συγκλίνει με τον ίδιο βαθμό στον πίνακα Q .

Περισσότερες λεπτομέρειες για το θεώρημα V.4 στο [1] θεώρημα 28.4.

VI Αλγόριθμος QR με εναλλαγές

Αυτό που κάνει τον αλγόριθμο QR πραγματικά αποδοτικό είναι η χρησιμοποίηση και κατάλληλη επιλογή εναλλαγών ($A \rightarrow A - \mu I$) σε κάθε βήμα. Σε αυτή την ενότητα θα δείξουμε πως αυτή η ιδέα μας δίνει κυβική σύγκλιση όταν συνδυαστεί και με τον αλγόριθμο πηλίκου Rayleigh.

VI.I Σύνδεση με την αντίστροφη επανάληψη

Συνεχίζουμε να έχουμε τη σύμβαση ότι $A \in \mathbb{R}^{m \times m}$ είναι πραγματικός και συμμετρικός, με πραγματικές ιδιοτιμές $\{\lambda_j\}$ και ορθομοναδιαία ιδιοδιανύσματα $\{x_j\}$.

Όπως είδαμε ο απλός αλγόριθμος QR 2.5 είναι ισοδύναμος με τον αλγόριθμο των ταυτόχρονων επαναλήψεων με αρχικό όρισμα τον ταυτοτικό πίνακα και συγκεκριμένα η πρώτη στήλη του πίνακα που παράγεται σε κάθε επανάληψη εξελίσσεται όπως η μέθοδος των δυνάμεων με αρχικό όρισμα το e_1 . Ο αλγόριθμος 2.5 είναι επίσης ισοδύναμος με τον αλγόριθμο των αντίστροφων ταυτόχρονων επαναλήψεων με αρχικό όρισμα έναν αντεστραμμένο ταυτοτικό πίνακα P και συγκεκριμένα η m -οστή στήλη του πίνακα-αποτέλεσμα εξελίσσεται όπως ο αλγόριθμος της αντίστροφης επανάληψης με αρχικό όρισμα το e_m . Αυτός ο διπλός ισχυρισμός είναι ιδιαίτερα σημαντικός και θα προσπαθήσουμε να τον δείξουμε παρακάτω.

Έστω ο πίνακας $Q^{(k)}$, που είναι όπως στην προηγούμενη ενότητα ο ορθογώνιος πίνακας μετά το k -οστό βήμα του αλγόριθμου QR. Στην προηγούμενη ενότητα δείξαμε ότι το γινόμενο της σχέσης 2.32 των πινάκων

$$\underline{Q}^{(k)} = \prod_{i=1}^k Q^{(i)} = \left[\begin{array}{c|c|c} x_1^{(k)} & x_2^{(k)} & \dots & x_m^{(k)} \end{array} \right]$$

είναι ο ίδιος ορθογώνιος πίνακας που εμφανίζεται μετά το k -οστό βήμα 2.27 του αλγόριθμου των ταυτόχρονων επαναλήψεων. Ισοδύναμο είναι να πούμε ότι ο πίνακας $\underline{Q}^{(k)}$ είναι ο ορθογώνιος παράγοντας της QR παραγοντοποίησης 2.34

$$A^k = \underline{Q}^{(k)} \underline{R}^{(k)}$$

Αν πάρουμε τους αντίστροφους πίνακες και στα δύο σκέλη της παραπάνω σχέσης θα έχουμε

$$A^{-k} = (\underline{R}^{(k)})^{-1} \underline{Q}^{(k)T} = \underline{Q}^{(k)} (\underline{R}^{(k)})^{-T} \quad (2.36)$$

Στη δεύτερη ισότητα χρησιμοποιήσαμε το γεγονός ότι ο A είναι συμμετρικός. Έστω ο ανάστροφος του ταυτοτικού $m \times m$ πίνακας P

$$P = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \ddots & & \\ 1 & & & \end{bmatrix}$$

Αφού $P^2 = I$, η 2.36 γράφεται

$$A^{-k}P = [\underline{Q}^{(k)}P][P(\underline{R}^{(k)})^{-T}P] \quad (2.37)$$

Ο παράγοντας $\underline{Q}^{(k)}P$ είναι ορθογώνιος. Ο παράγοντας $P(\underline{R}^{(k)})^{-T}P$ είναι άνω-τριγωνικός. Έτσι η 2.37 μπορεί να θεωρηθεί ως μια QR παραγοντοποίηση του πίνακα $A^{(-k)}P$. Με άλλα λόγια ισοδύναμα θα μπορούσαμε να εκτελέσουμε τον αλγόριθμο των ταυτόχρονων επαναλήψεων για τον πίνακα A^{-1} με αρχικό πίνακα τον P , που θα μπορούσαμε να τον πούμε αλγόριθμο ταυτόχρονων αντίστροφων επαναλήψεων του A . Συγκεκριμένα η πρώτη στήλη του $\underline{Q}^{(k)}P$ και η τελευταία του $\underline{Q}^{(k)}$ είναι το αποτέλεσμα k επαναλήψεων του αλγόριθμου αντίστροφων επαναλήψεων με αρχικό διάνυσμα το e_m .

VI.Π Σύνδεση με την αντίστροφη επανάληψη με εναλλαγές

Έτσι ο αλγόριθμος QR είναι μαζί αλγόριθμος ταυτόχρονων επαναλήψεων και αλγόριθμος ταυτόχρονων αντίστροφων επαναλήψεων: Η συμμετρία είναι τέλεια. Αλλά, όπως είδαμε στην ενότητα IV, υπάρχει μια τεράστια διαφορά μεταξύ του αλγόριθμου των δυνάμεων και του αλγόριθμου των αντίστροφων επαναλήψεων: ο τελευταίος είναι πολύ πιο γρήγορος με τη χρήση εναλλαγών. Όσο καλύτερη προσέγγιση δώσουμε μιας ιδιοτιμής $\mu \approx \lambda_j$ τόσο καλύτερη σύγκλιση θα έχουμε σε κάθε βήμα της αντίστροφης επανάληψης. Ο πρακτικός αλγόριθμος QR 2.6 δείχνει πως μπορούμε να εισάγουμε τη χρήση εναλλαγών σε κάθε βήμα του αλγόριθμου QR.

Έστω $\mu^{(k)}$ η προσέγγιση μιας ιδιοτιμής που επιλέχθηκε στο k -οστό βήμα του αλγόριθμου QR. Στον αλγόριθμο 2.6 η σχέση μεταξύ $(k-1)$ -οστού και k -οστού βήματος είναι

$$\begin{aligned} A^{(k-1)} - \mu^{(k)}I &= Q^{(k)}R^{(k)}, \\ A^{(k)} &= R^{(k)}Q^{(k)} + \mu^{(k)}I \end{aligned}$$

Επομένως

$$A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)} \quad (2.38)$$

Και από επαγωγή

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$$

Όπου είναι ίδια με τη σχέση 2.35. Όμως η 2.34 δεν ισχύει σε αυτή την περίπτωση. Για την ακρίβεια ισχύει η παρακάτω μορφή με εναλλαγές

$$(A - \mu^{(k)}I)(A - \mu^{(k-1)}I) \dots (A - \mu^{(1)}I) = \underline{Q}^{(k)} \underline{R}^{(k)} \quad (2.39)$$

Ουσιαστικά ο πίνακας $\underline{Q}^{(k)} = \text{prod}_{i=1}^k Q^{(i)}$ είναι μια ορθογωνιοποίηση του πίνακα $\prod_{i=k}^1 (A - \mu^{(i)})$. Η πρώτη στήλη του $\underline{Q}^{(k)}$ είναι το αποτέλεσμα της μεθόδου των δυνάμεων με αρχικό διάνυσμα το e_1 και με τις εναλλαγές $\mu^{(i)}$ και η τελευταία του στήλη είναι το αποτέλεσμα της αντίστροφης επανάληψης με τις ίδιες εναλλαγές και αρχικό διάνυσμα το e_m . Αν οι εναλλαγές είναι καλές προσεγγίσεις ιδιοτιμών αυτή η τελευταία στήλη συγκλίνει γρήγορα σε ένα ιδιοδιάνυσμα.

VI.III Σύνδεση με την επανάληψη πηλίκου Rayleigh

Ανακαλύψαμε ένα πολύ ένα αρκετά ισχυρό εργαλείο που ήταν κρυμμένο στον αλγόριθμο QR με εναλλαγές: την αντίστροφη επανάληψη με εναλλαγές. Για να ολοκληρώσουμε την ιδέα χρειαζόμαστε και έναν αποτελεσματικό τρόπο να επιλέγουμε εναλλαγές ώστε να επιτύχουμε τη γρήγορη σύγκλιση στην τελευταία στήλη του $\underline{Q}^{(k)}$

Το πηλίκo Rayleigh σίγουρα είναι μια καλή αρχή. Για να προσεγγίσουμε την ιδιοτιμή που αντιστοιχεί στο ιδιοδιάνυσμα που προσεγγίζει η τελευταία στήλη του $\underline{Q}^{(k)}$ είναι λογικό να πάρουμε το πηλίκo Rayleigh αυτής της στήλης. Έτσι θα έχουμε

$$\mu^{(k)} = \frac{(x_m^{(k)})^T A x_m^{(k)}}{(x_m^{(k)})^T x_m^{(k)}} = (x_m^{(k)})^T A x_m^{(k)} \quad (2.40)$$

Αν επιλέγεται αυτή η εναλλαγή σε κάθε βήμα, οι προσεγγίσεις των $\mu^{(k)}$ και $x_m^{(k)}$ ταυτίζονται με αυτές που υπολογίζει ο αλγόριθμος της επανάληψης πηλίκου Rayleigh έχοντας ως αρχικό διάνυσμα το e_m . Επομένως, ο αλγόριθμος QR επιτυγχάνει κυβική σύγκλιση με την έννοια ότι το $x_m^{(k)}$ συγκλίνει κυβικά σε ένα ιδιοδιάνυσμα.

Σημειώστε ότι στον αλγόριθμο QR το πηλίκo Rayleigh $r(x_m^{(k)})$ θα είναι πάντα το (m, m) στοιχείο του πίνακα $A^{(k)}$ και έτσι το επιλέγουμε χωρίς κανένα κόστος σε πράξεις. Ξεκινώντας από την (2.38) έχουμε

$$A_{mm}^{(k)} = e_m^T A^{(k)} e_m = e_m^T \underline{Q}^{(k)T} A \underline{Q}^{(k)} e_m = x_m^{(k)T} A x_m^{(k)} \quad (2.41)$$

Επομένως, η 2.40 ισοδύναμα μπορεί να γίνει $\mu^{(k)} = A_{mm}^{(k)}$. Αυτή η εναλλαγή είναι γνωστή ως εναλλαγή πηλίκου Rayleigh.

VI.IV Εναλλαγή Wilkinson

Αν και η εναλλαγή πηλίκου Rayleigh επιτυγχάνει κυβική σύγκλιση έχει το μειονέκτημα ότι δεν επιτυγχάνει πάντα σύγκλιση. Θα δούμε ένα απλό παράδειγμα. Έστω ο πίνακας

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2.42)$$

Ο αλγόριθμος QR χωρίς εναλλαγές δεν συγκλίνει καθόλου, όπως φαίνεται παρακάτω

$$A = Q^{(1)}R^{(1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$A^{(1)} = R^{(1)}Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A$$

Η εναλλαγή πηλίκου Rayleigh δεν μπορεί να επηρεάσει καθόλου την παραπάνω διαδικασία αφού $\mu = A_{mm} = 0$. Επομένως, ο αλγόριθμος QR με την επιλογή της εναλλαγής πηλίκου Rayleigh μπορεί στο χειρότερο σενάριο να αποτύχει.

Το πρόβλημα πηγάζει από τη συμμετρία των ιδιοτιμών. Η μία ιδιοτιμή είναι $+1$ και η άλλη -1 και επομένως όταν προσπαθούμε να βελτιώσουμε την εκτίμηση της ιδιοτιμής με την εναλλαγή 0 , οι δύο ιδιοτιμές ισαπέχουν από την εναλλαγή μας και έτσι δεν η προσέγγιση δεν προχωρά. Αυτό που χρειαζόμαστε είναι μια εκτίμηση ιδιοτιμής που θα σπάει τη συμμετρία. Μια τέτοια επιλογή ορίζουμε ακριβώς παρακάτω.

Έστω ότι ο πίνακας B είναι ο 2×2 κάτω-δεξιά υποπίνακας του $A^{(k)}$, έτσι ώστε

$$B = \begin{bmatrix} \alpha_{m-1} & b_{m-1} \\ b_{m-1} & \alpha_m \end{bmatrix}$$

Η εναλλαγή Wilkinson λέμε ότι είναι η ιδιοτιμή του πίνακα B που είναι πιο κοντά στο στοιχείο α_m . Στην περίπτωση ισαπεχουσών ιδιοτιμών από το α_m επιλέγουμε αυθαίρετα ανάμεσα στις δύο ιδιοτιμές. Μία αριθμητικά ευσταθής σχέση για την εναλλαγή Wilkinson είναι η παρακάτω

$$\mu = \alpha_m - \text{sign}(\delta)b_{m-1}^2 / \left(|\delta| + \sqrt{\delta^2 + b_{m-1}^2} \right) \quad (2.43)$$

όπου $\delta = (\alpha_{m-1} - \alpha_m)/2$. Αν $\delta = 0$ το $\text{sign}(\delta)$ επιλέγεται αυθαίρετα ως $+1$ ή -1 .

Στη γενική περίπτωση η εναλλαγή Wilkinson επιτυγχάνει κυβική σύγκλιση όπως και η εναλλαγή πηλίκου Rayleigh. Επίσης μπορεί να δειχθεί ότι

επιτυγχάνει τετραγωνική σύγκλιση στη χειρότερη περίπτωση. Ουσιαστικά ο αλγόριθμος QR με εναλλαγή Wilkinson συγκλίνει πάντα.

Στο παράδειγμα 2.42 η εναλλαγή Wilkinson είναι $+1$ ή -1 . Έτσι η συμμετρία σπάει και η σύγκλιση επιτυγχάνεται σε ένα βήμα.

VI.V Ευστάθεια και ακρίβεια

Οι προηγούμενες υποενότητες ολοκληρώνουν τη συζήτηση περί των τεχνικών ζητημάτων του αλγόριθμου QR, αν και πολλές πρακτικές λεπτομέρειες παραλείφθηκαν που έχουν να κάνουν με τις συνθήκες αναγωγής του προβλήματος σε απλούστερο και δευτερεύουσες στρατηγικές για την επιλογή εναλλαγών. Σε αυτή την υποενότητα θα αναφερθούμε στην ευστάθεια και την ακρίβεια του αλγορίθμου μας.

Όπως θα μπορούσαμε να περιμένουμε, λόγω της χρησιμοποίησης ορθογώνιων πινάκων, ο αλγόριθμος QR είναι προς τα πίσω ευσταθής. Ο πιο απλός τρόπος να εκφράσουμε με μαθηματικές σχέσεις τον προηγούμενο ισχυρισμό είναι να θεωρήσουμε τον διαγώνιο πίνακα $\tilde{\Lambda}$ ως το αποτέλεσμα της διαγωνιοποίησης του πίνακα A όπως υπολογίστηκε στην αριθμητική κινήτης υποδιαστολής και τον ορθογώνιο πίνακα \tilde{Q} που σχετίζεται με το γινόμενο των Householder πινάκων που χρησιμοποιήθηκαν σε όλη τη διαδικασία.

Θεώρημα VI.1. Έστω ένας πραγματικός, συμμετρικός, τριδιαγώνιος πίνακας $A \in \mathbb{R}^{m \times m}$ που διαγωνιοποιήθηκε από τον αλγόριθμο QR (αλγόριθμος 2.6) σε ένα υπολογιστή που ικανοποιεί τα αξιώματα 1.23, 1.25 και έστω οι πίνακες $\tilde{\Lambda}$ και \tilde{Q} όπως ορίστηκαν παραπάνω. Τότε θα ισχύει

$$\tilde{Q}\tilde{\Lambda}\tilde{Q}^* = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}}) \quad (2.44)$$

για κάποιο $\delta A \in \mathbb{C}^{m \times m}$.

Περισσότερες λεπτομέρειες για το θεώρημα VI.1 στο [1] θεώρημα 29.1

Επομένως ο αλγόριθμος QR μας δίνει ένα ακριβές αποτέλεσμα για μικρές διαταραχές. Συνδυάζοντας τα θεωρήματα III.1 και VI.1 βλέπουμε ότι ο τριδιαγώνιος μετασχηματισμός ακολουθούμενος από τον αλγόριθμο QR είναι ένας προς τα πίσω ευσταθής αλγόριθμος για τον υπολογισμό ιδιοτιμών πινάκων. Για να δούμε τι σημαίνουν τα παραπάνω για την ακρίβεια των υπολογισμένων ιδιοτιμών πρέπει να συνδυάσουμε το γεγονός της προς τα πίσω ευστάθειας του αλγόριθμου 2.6 και του θεωρήματος IV.3 του πρώτου κεφαλαίου (δεδομένου ότι οι συμμετρικοί πίνακες είναι ειδική περίπτωση των ομαλών πινάκων). Το συμπέρασμα για την ιδιοτιμή $\tilde{\lambda}_i$ που υπολογίστηκε

είναι το παρακάτω

$$\frac{|\tilde{\lambda}_i - \lambda_i|}{\|A\|} = O(\epsilon_{machine}) \quad (2.45)$$

Η παραπάνω ακρίβεια δεν είναι καθόλου κακή για ένα αλγόριθμο που απαιτεί μόλις $\frac{4}{3}m^3$ πράξεις (τα δύο-τρίτα των πράξεων που απαιτεί ο υπολογισμός του γινομένου δύο $m \times m$ πινάκων!).

VII Άλλοι αλγόριθμοι

Σε αυτή την ενότητα θα παρουσιάσουμε τρεις διάσημες εναλλακτικές του αλγόριθμου QR. Τον αλγόριθμο Jacobi, τον αλγόριθμο της διχοτόμησης και τον αλγόριθμο διαίρει και βασίλευε. Ο πρώτος χρησιμοποιείται για πλήρεις πίνακες και οι τελευταίοι για τριδιαγώνιους.

VII.I Αλγόριθμος Jacobi

Μία από τις παλαιότερες ιδέες για τον υπολογισμό ιδιοτιμών πινάκων είναι ο αλγόριθμος Jacobi που κατασκευάστηκε από τον Jacobi το 1845. Αυτή η μέθοδος υπήρξε πόλος έλξης του ενδιαφέροντος ειδικά από τη στιγμή που υπήρξε η δυνατότητα των παράλληλων υπολογισμών, αλλά ποτέ δεν μπόρεσε να σταθεί απέναντι στον ανταγωνισμό άλλων αλγορίθμων.

Όσο αφορά την ιδέα του αλγορίθμου. Ξέρουμε ότι για πίνακες διάστασης 5 ή και μεγαλύτερης μπορούμε να υπολογίσουμε τις ιδιοτιμές τους μόνο επαναληπτικά. Παρόλα αυτά σε πίνακες μικρότερης διάστασης μπορούμε να υπολογίσουμε τις ιδιοτιμές σε ένα βήμα. Αν διαγωνιοποιήσουμε διαδοχικά μικρούς υποπίνακες του A ελπίζοντας ότι η διαδικασία θα συγκλίνει με κάποιο τρόπο στη διαγωνιοποίηση του πλήρους πίνακα A ;

Αυτή η ιδέα δοκιμάστηκε για 4×4 υποπίνακες αλλά η βασική προσέγγιση βασίζεται σε 2×2 υποπίνακες. Ένας 2×2 συμμετρικός πίνακας μπορεί να διαγωνιοποιηθεί στη μορφή

$$J^T \begin{bmatrix} \alpha & d \\ d & b \end{bmatrix} J = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix} \quad (2.46)$$

Όπου ο J είναι ορθογώνιος πίνακας. Υπάρχουν αρκετοί τρόποι να επιλέξεις τον πίνακα J . Μία επιλογή θα μπορούσε να είναι

$$F = \begin{bmatrix} -c & s \\ s & c \end{bmatrix} \quad (2.47)$$

Όπου $s = \sin(\theta)$ και $c = \cos(\theta)$ για κάποιο θ . Για τον πίνακα F θα ισχύει $\det(F) = -1$. Εναλλακτικά θα μπορούσαμε να χρησιμοποιήσουμε τον πίνακα περιστροφής

$$J = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \quad (2.48)$$

Με $\det(J) = 1$, η οποία είναι μια κλασική επιλογή στον αλγόριθμο Jacobi και λέγεται πίνακας Ιακωβιανής περιστροφής. Μπορεί να δειχθεί ότι η πα-

τη συμμετρία και στον πίνακα $A^{(1)}$. Εύκολα λοιπόν διαπιστώνουμε ότι οι πίνακες A και $A^{(1)}$ είναι όμοιοι και έχουν μόνη διαφορά τις στήλες και γραμμές i, j .

Επομένως είμαστε σε θέση να περιγράψουμε με αρκετή σαφήνεια την επαναληπτική μέθοδο του αλγόριθμου Jacobi. Το k βήμα της διαδικασίας θα είναι το εξής:

1. Επιλέγουμε κάποιο στοιχείο A_{ij} εκτός από τα στοιχεία της κύριας διαγωνίου του πίνακα A , ώστε να μηδενιστεί μετά το τέλος του πρώτου βήματος.
2. Βρίσκουμε τη γωνία που ορίζεται από τη σχέση 2.53 και κατασκευάζουμε τον πίνακα R_{ij} .
3. Υπολογίζουμε τον ορθογώνιο μετασχηματισμό ομοιότητας $A^{(k)} = (R_{ij})^T A^{(k-1)} R_{ij}$.

Επαναλαμβάνουμε την παραπάνω διαδικασία για τον πίνακα $A^{(k)}$ μέχρι να προκύψει πίνακας "σχεδόν διαγώνιος". Χρησιμοποιούμε αυτόν τον αδόκιμο όρο καθώς η διαδικασία παραγωγής ενός διαγώνιου πίνακα μέσω της παραπάνω επαναληπτικής διαδικασίας δεν είναι τόσο απλή υπόθεση. Το βασικό πρόβλημα είναι ότι στην προσπάθειά μας να μηδενίσουμε το στοιχείο της i γραμμής και της j στήλης τροποποιούμε ολόκληρες τις i, j γραμμές και στήλες, με αποτέλεσμα να τροποποιούμε τα στοιχεία που μηδενίσαμε σε προηγούμενα βήματα. Έτσι από μαθηματικής άποψης δεν καταλήγουμε ποτέ σε διαγώνιο πίνακα. Όμως έχει αποδειχθεί ότι οι πίνακες $A^{(k)}$ που ορίζονται από την παραπάνω επαναληπτική διαδικασία συγκλίνουν σε ένα διαγώνιο πίνακα. Για την ακρίβεια αυτό που συμβαίνει είναι ότι σε κάθε βήμα του αλγόριθμου το άθροισμα των τετραγώνων των μη διαγώνιων στοιχείων μειώνεται κατά $2\alpha_{ij}^2$.

Επομένως γεννιέται το ερώτημα: Ποιο μη διαγώνιο στοιχείο πρέπει να μηδενίζουμε σε κάθε επανάληψη; Το γεγονός της μείωσης του αθροίσματος των τετραγώνων των μη διαγώνιων στοιχείων κατά $2\alpha_{ij}^2$ μας οδηγεί αυθόρμητα στην επιλογή σε κάθε επανάληψη του μη διαγώνιου στοιχείου με τη μεγαλύτερη απόλυτη τιμή. Μετά από $O(m^2)$ βήματα, που το καθένα απαιτεί $O(m)$ διαδικασίες, το άθροισμα των τετραγώνων θα έχει μειωθεί κατά μία σταθερά και η σύγκλιση με ακρίβεια $\epsilon_{machine}$ επιτυγχάνεται μετά από $O(m^3 \log(\epsilon_{machine}))$ διαδικασίες. Στην πραγματικότητα η σύγκλιση είναι καλύτερη από το προηγούμενο αποτέλεσμα και το κόστος διαδικασιών τελικά είναι $O(m^3 \log(|\log(\epsilon_{machine})|))$.

Στην πράξη όμως η αναζήτηση του στοιχείου με τη μεγαλύτερη απόλυτη τιμή στοιχίζει σε υπολογιστικό χρόνο και έτσι επιλέγουμε τα στοιχεία προς μηδενισμό με τη σειρά: $\alpha_{12}, \alpha_{13}, \dots, \alpha_{1m}, \alpha_{23}, \dots, \alpha_{2m}, \dots, \alpha_{m-1,m}$. Σε αυτή την περίπτωση η επαναληπτική διαδικασία που προκύπτει αναφέρεται συχνά σαν ειδικός κυκλικός αλγόριθμος του Jacobi. Έτσι αποφεύγουμε το $O(m^2)$ κόστος για την αναζήτηση του μεγαλύτερου κατά απόλυτη τιμή στοιχείου, ενώ παράλληλα επιτυγχάνεται μια γρήγορη ασυμπτωτική σύγκλιση. Έπειτα

από μια προσπέλαση όλων των $m(m-1)/2$ στοιχείων η ακρίβεια γενικά βελτιώνεται περισσότερο από ένα σταθερό όρο.

Ο αλγόριθμος Jacobi είναι ελκυστικός κυρίως επειδή έχει να κάνει με ζευγάρια γραμμών και στηλών με αποτέλεσμα να μπορούμε να εκτελούμε τα βήματα παράλληλα. Επιπλέον ο πίνακας δεν χρειάζεται να τριδιαγωνιοποιηθεί (αφού ο αλγόριθμος θα κατέστρεφε τα μηδενικά στοιχεία). Η σύγκλιση για τυπικά προβλήματα διάστασης $m \leq 100$ συνήθως επιτυγχάνεται σε λιγότερες από δέκα προσπελάσεις όλων των μη διαγώνιων στοιχείων και η τελική σύγκλιση είναι καλύτερης ακρίβειας από του αλγόριθμου QR. Δυστυχώς ακόμα και με χρήση παράλληλων υπολογισμών ο αλγόριθμος Jacobi δεν είναι τόσο γρήγορος όσο οι αλγόριθμοι QR ή ο διαίρει και βασίλευε αφού πρώτα τριδιαγωνιοποιήσουμε τον πίνακά μας.

VII.Π Αλγόριθμος διχοτόμησης

Ο αλγόριθμος της διχοτόμησης έχει μεγάλη πρακτική αξία. Αφού ένας συμμετρικός πίνακας μετασχηματίζεται σε τριδιαγώνιο αυτός ο αλγόριθμος ορίζει το επόμενο βήμα αν δεν θέλουμε το σύνολο των ιδιοτιμών αλλά ένα υποσύνολο. Για παράδειγμα μπορεί να βρει τις μεγαλύτερες 10% του συνόλου των ιδιοτιμών ή τις τριάντα μικρότερες ή όλες τις ιδιοτιμές στο διάστημα $[1, 2]$. Όταν βρεθούν οι επιθυμητές ιδιοτιμές μπορούμε να υπολογίσουμε τα αντίστοιχα ιδιοδιανύσματα με ένα βήμα του αλγορίθμου της αντίστροφης επανάληψης (Αλγόριθμος 2.3).

Η αφετηρία μας είναι στοιχειώδης με την έννοια ότι αφού ο πίνακας είναι πραγματικός και συμμετρικός και οι ιδιοτιμές μας πραγματικές αρκεί να ψάξουμε τον άξονα των πραγματικών αριθμών για τις ρίζες του χαρακτηριστικού πολυωνύμου $p(x) = \det(A - xI)$. Αυτή η ιδέα θυμίζει αυτή που αναφέραμε ως κακή στην ενότητα III για την εύρεση ιδιοτιμών μέσω της προσέγγισης ριζών ενός πολυωνύμου. Αυτή η διαδικασία γενικά δεν είναι αποτελεσματική καθώς είναι αρκετά μη ευσταθής. Η διαφορά όμως στην περίπτωσή μας είναι ότι δεν αναφερόμαστε στην ιδέα να βρούμε ρίζες πολυωνύμου χρησιμοποιώντας τους συντελεστές του. Η ιδέα είναι να υπολογίσουμε τις ρίζες του $p(x)$ υπολογίζοντας την τιμή του για διάφορα x , χωρίς ποτέ να ανατρέξουμε στους συντελεστές του και να εκτελέσουμε τη μέθοδο διχοτόμησης για μη γραμμικές συναρτήσεις. Αυτό θα μπορούσε να γίνει, για παράδειγμα, με τη βοήθεια της μεθόδου απαλοιφής του Gauss και ο τελικός αλγόριθμος να είναι αρκετά ευσταθής.

Όλα αυτά ακούγονται χρήσιμα όμως όχι και κάτι που κάνει κάποιο ποιοτικό άλμα σε σχέση με ότι έχουμε δει μέχρι τώρα. Αυτό που δίνει τη δυναμική και καθιστά ελκυστικό τον αλγόριθμο είναι κάποιες επιπλέον ιδιότητες των ιδιοτιμών και των οριζουσών που δεν είναι άμεσα ορατές.

Έστωσαν ένας συμμετρικός πίνακας $A \in \mathbb{R}^{m \times m}$ και οι άνω-αριστερά τε-

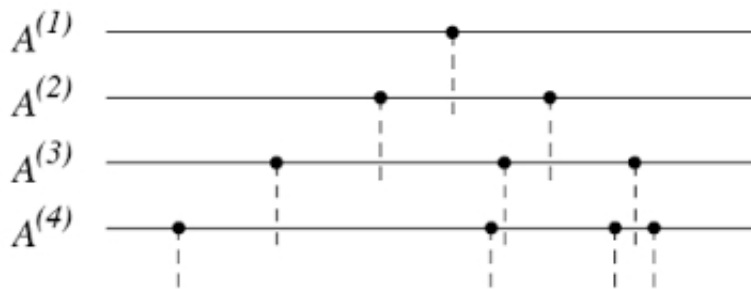
τράγωνοι πίνακες $A^{(1)}, \dots, A^{(m)}$ διαστάσεων $1, \dots, m$. Μπορεί να αποδειχθεί ότι οι ιδιοτιμές αυτών των πινάκων παρεμβάλλονται μεταξύ τους. Πριν περιγράψουμε με αυστηρότητα τι εννοούμε μπορούμε να το παρουσιάσουμε σχηματικά. Έστω ότι ο πίνακας A είναι τριδιαγώνιος και δεν μπορεί να αναχθεί, με την έννοια ότι όλα τα μη διαγώνια στοιχεία του είναι μη μηδενικά

$$A = \begin{bmatrix} \alpha_1 & b_1 & & & & \\ b_1 & \alpha_2 & b_2 & & & \\ & b_2 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & b_{m-1} & \\ & & & b_{m-1} & \alpha_m & \end{bmatrix}, \quad b_i \neq 0 \quad (2.54)$$

Αν κάποια μη διαγώνια στοιχεία είναι μηδενικά τότε το πρόβλημα μπορεί να αναχθεί σε απλούστερο όπως στον αλγόριθμο 2.6. Όπως έχουμε αναφέρει οι ιδιοτιμές κάθε πίνακα θα είναι διακριτές και έστω $\lambda_1^{(k)} < \lambda_2^{(k)} < \dots < \lambda_k^{(k)}$. Η κρίσιμη ιδιότητα που κάνει τον αλγόριθμο της διχοτόμησης πολύ δυνατό είναι ότι οι ιδιοτιμές του $A^{(k)}$ παρεμβάλλονται σε αυτές του $A^{(k+1)}$ ικανοποιώντας την ανισότητα

$$\lambda_i^{(k+1)} < \lambda_i^{(k)} < \lambda_{i+1}^{(k+1)} \quad (2.55)$$

για $k = 1, 2, \dots, m-1$ και $i = 1, 2, \dots, k-1$. Αυτή η ιδιότητα φαίνεται στο σχήμα 2.2 παρακάτω



Σχήμα 2.2: Η εικόνα της παρεμβολής των ιδιοτιμών που περιγράφεται στην 2.55 για τους άνω-αριστερά υποπίνακες $\{A^{(i)}\}$ ενός μη αναγωγίμου τριδιαγώνιου, πραγματικού και συμμετρικού πίνακα A . Οι ιδιοτιμές του πίνακα $A^{(k)}$ παρεμβάλλονται σε αυτές του πίνακα $A^{(k+1)}$. Ο αλγόριθμος της διχοτόμησης εκμεταλλεύεται αυτή την ιδιότητα.

Είναι η ιδιότητα της παρεμβολής που μας δίνει την δυνατότητα να βρούμε τον αριθμό των ιδιοτιμών που ανήκουν σε ένα συγκεκριμένο διάστημα. Για παράδειγμα έστω ο 4×4 πίνακας

$$A = \begin{bmatrix} 1 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 2 & 1 & \\ & & 1 & -1 & \end{bmatrix}$$

Με βάση τους αριθμούς

$$\det(A^{(1)}) = 1, \quad \det(A^{(2)}) = -1, \quad \det(A^{(3)}) = -3, \quad \det(A^{(4)}) = 4,$$

ξέρουμε ότι ο πίνακας $A^{(1)}$ δεν έχει καμία αρνητική ιδιοτιμή, ο $A^{(2)}$ έχει μία αρνητική ιδιοτιμή, ο $A^{(3)}$ έχει μία αρνητική ιδιοτιμή και ο $A^{(4)}$ έχει δύο αρνητικές ιδιοτιμές. Γενικά για κάθε συμμετρικό, τριδιαγώνιο πίνακα $A \in \mathbb{R}^{m \times m}$, ο αριθμός των αρνητικών ιδιοτιμών του είναι ίσος με τον αριθμό αλλαγής του πρόσημου στην παρακάτω ακολουθία

$$1, \det(A^{(1)}), \det(A^{(2)}), \dots, \det(A^{(m)}) \quad (2.56)$$

Η οποία είναι γνωστή και ως ακολουθία *Sturm*. (Η προηγούμενη ιδιότητα ισχύει ακόμα και αν μετρήσουμε τις μηδενικές ορίζουσες αν ορίσουμε σαν "αλλαγή πρόσημου" την αλλαγή από το + ή το 0 στο - ή από το - ή το 0 στο + αλλά όχι από το + ή το - στο 0). Αν πάρουμε μια εναλλαγή του A με την έννοια ότι αφαιρούμε από τον A ένα πολλαπλάσιο του ταυτοτικού πίνακα μπορούμε να βρούμε τον αριθμό των ιδιοτιμών του σε οποιοδήποτε διάστημα $[\alpha, \beta]$: θα είναι ο αριθμός των ιδιοτιμών στο $(-\infty, \beta)$ μείον τον αριθμό των ιδιοτιμών στο $(-\infty, \alpha)$.

Υπάρχουν ακόμα δύο βασικές ιδιότητες που χαρακτηρίζουν τον αλγόριθμο διχοτόμησης. Για κάθε τριδιαγώνιο πίνακα οι ορίζουσες και τα χαρακτηριστικά πολυώνυμα των πινάκων $\{A^{(k)}\}$ συνδέονται με δύο αναδρομικές σχέσεις. Αναπτύσσοντας την ορίζουσα $\det(A^{(k)})$ με την βοήθεια των υποπινάκων και θεωρώντας τα στοιχεία b_{k-1}, α_k που βρίσκονται στην k -οστή σειρά, παίρνουμε με επεξεργασία της 2.54

$$\det(A^{(k)}) = \alpha_k \det(A^{(k-1)}) - b_{k-1}^2 \det(A^{(k-2)}) \quad (2.57)$$

Επίσης για τα χαρακτηριστικά πολυώνυμα $\pi^{(k)} = \det(A^{(k)} - xI)$ ισχύει

$$\pi^{(k)}(x) = (\alpha_k - x)\pi^{(k-1)}(x) - b_{k-1}^2 \pi^{(k-2)}(x) \quad (2.58)$$

Αν ορίσουμε τα $\pi^{(-1)}(x) = 0$ και $\pi^{(0)}(x) = 1$, τότε αυτή η αναδρομή ισχύει για όλα τα $k = 1, 2, \dots, m$. Οι παραπάνω αναδρομικές σχέσεις δίνονται στο [1] υποενότητα 30.

Αν υπολογίσουμε την 2.58 για το χαρακτηριστικό πολυώνυμο του A και για διαδοχικές τιμές του x και μετρήσουμε τις αλλαγές των προσήμων τότε ο αλγόριθμος διχοτόμησης εντοπίζει ιδιοτιμές σε αρκετά μικρά διαστήματα. Το κόστος είναι $O(m)$ πράξεις για κάθε υπολογισμό της ακολουθίας και επομένως απαιτούνται $O(m \log(\epsilon_{machine}))$ πράξεις συνολικά για να υπολογίσουμε μια ιδιοτιμή με ακρίβεια σχετική με το $\epsilon_{machine}$. Αν ζητείται ένας μικρός αριθμός ιδιοτιμών τότε έχουμε μια σαφή βελτίωση από το κόστος $O(m^2)$ διαδικασιών του απαιτεί ο αλγόριθμος QR.

VII.III Αλγόριθμος διαίρει και βασιλεύει

Ο αλγόριθμος διαίρει και βασιλεύει βασίζεται σε μια αναδρομική υποδιαίρεση ενός συμμετρικού τριδιαγώνιου προβλήματος ιδιοτιμών σε προβλήματα μικρότερων διαστάσεων μέχρι τη διαίρεση του σε στοιχειώδη προβλήματα όπου η σύνθεση των λύσεων τους παράγει την ζητούμενη λύση. Ουσιαστικά πρόκειται για την πιο σημαντική εξέλιξη στους αλγόριθμους υπολογισμού ιδιοτιμών πινάκων από τη δεκαετία του '60. Ο πρώτος που τον ανέπτυξε και τον εισήγαγε ήταν ο Curren το 1981 και είναι δύο φορές πιο γρήγορος από τον αλγόριθμο QR αν ζητούνται και οι ιδιοτιμές και τα ιδιοδιανύσματα.

Προς το παρόν θα δώσουμε μόνο τη βασική ιδέα, χωρίς τις λεπτομέρειες. Όμως πρέπει να σημειώσουμε ότι οι λεπτομέρειες όπως θα δούμε στην υλοποίηση του αλγόριθμου είναι ιδιαίτερα σημαντικές γιατί αν δεν εκτελεστούν σωστά ο αλγόριθμος δεν είναι ευσταθής.

Έστω ο πραγματικός, συμμετρικός, τριδιαγώνιος και μη αναγώγιμος πίνακας (αλλιώς το πρόβλημα μπορεί να αναχθεί σε απλούστερο) $T \in \mathbb{R}^{m \times m}$ με $m \geq 2$. Τότε για κάθε $n \in \mathbb{N}$ με $1 \leq n < m$, ο T μπορεί να χωριστεί σε στους υποπίνακες όπως φαίνεται στο σχήμα 2.3

$$T = \begin{array}{|c|c|} \hline T_1 & \beta \\ \hline \beta & T_2 \\ \hline \end{array}$$

Σχήμα 2.3

Ο T_1 είναι ο $n \times n$ άνω-αριστερά υποπίνακας του T και ο T_2 είναι ο $(m-n) \times (m-n)$ κάτω-δεξιά υποπίνακας του T και $\beta = t_{n+1,n} = t_{n,n+1} \neq 0$. Στη συνέχεια αντικαθιστούμε το κάτω-δεξιά, t_{nn} , στοιχείο του T_1 με το $t_{nn} - \beta$ και το άνω-αριστερά, $t_{n+1,n+1}$, στοιχείο του T_2 με το $t_{n+1,n+1} - \beta$. Έτσι λοιπόν μετασχηματίζουμε περαιτέρω τον πίνακα και θα έχουμε τη σχέση που φαίνεται στο παρακάτω σχήμα.

Το σχήμα 2.4 ουσιαστικά μας δείχνει ότι ένας τριδιαγώνιος πίνακας μπορεί να εκφραστεί σαν άθροισμα ενός σύνθετου 2×2 πίνακα με τριδιαγώνιους πίνακες-στοιχεία και ενός πίνακα-διόρθωση βαθμού ένα.

Έστω λοιπόν ότι έχουμε σπάσει τον αρχικό μας πίνακα T με $n \approx m/2$ και ότι το πρόβλημα που προκύπτει είναι στοιχειώδες, δηλαδή βρίσκουμε τη λύση του (με κάποιο τρόπο που δεν μας ενδιαφέρει προς το παρόν) και άρα ξέρουμε τις ιδιοτιμές των πινάκων \hat{T}_1 και \hat{T}_2 . Από τη στιγμή που ο πίνακας-διόρθωση είναι βαθμού ένα μπορούμε με τη βοήθεια ενός μη γραμμικού

$$T = \begin{array}{|c|c|} \hline \hat{T}_1 & 0 \\ \hline 0 & \hat{T}_2 \\ \hline \end{array} + \begin{array}{|c|c|} \hline & \beta \\ \hline \beta & \beta \\ \hline \end{array}$$

Σχήμα 2.4

αλλά γρήγορου υπολογισμού να καταλήξουμε από τις ιδιοτιμές των πινάκων \hat{T}_1, \hat{T}_2 σε αυτές του πίνακα A. Αν αναδρομικά προσπαθήσουμε να βρούμε τις ιδιοτιμές των πινάκων \hat{T}_1, \hat{T}_2 χωρίς να θεωρήσουμε κάποιο πρόβλημα στοιχειώδες θα καταλήξουμε σε ένα σύνολο 1×1 στοιχειωδών προβλημάτων ιδιοτιμών και σε ένα σύνολο πινάκων-διόρθωσης βαθμού ένα. (Στην πράξη για καλύτερη αποδοτικότητα συνήθως χρησιμοποιούμε τον αλγόριθμο QR για να λύσουμε τα στοιχειώδη προβλήματα όταν οι πίνακες που προκύπτουν από την αναδρομή είναι αρκετά μικροί).

Σε όλη αυτή τη διαδικασία υπάρχει μια μαθηματική σχέση κλειδί. Αν λοιπόν ξέρουμε τις ιδιοτιμές των πινάκων \hat{T}_1, \hat{T}_2 πώς μπορούμε να βρούμε αυτές του πίνακα T; Έστω ότι έχουμε υπολογίσει τις διαγωνιοποιήσεις

$$\hat{T}_1 = Q_1 D_1 Q_1^T, \quad \hat{T}_2 = Q_2 D_2 Q_2^T$$

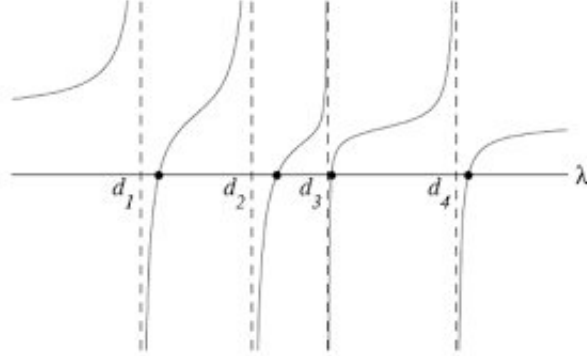
Τότε από το σχήμα 2.4 παίρνουμε ότι

$$T = \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \left(\begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} + \beta z z^T \right) \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix} \quad (2.59)$$

με $z^T = (q_1^T, q_2^T)$, όπου q_1^T η τελευταία σειρά του Q_1 και q_2^T η πρώτη σειρά του Q_2 . Αφού η σχέση 2.59 είναι ένας μετασχηματισμός ομοιότητας, ουσιαστικά έχουμε αναγάγει το πρόβλημά μας στο πρόβλημα εύρεσης των ιδιοτιμών του αθροίσματος ενός διαγώνιου πίνακα και ενός πίνακα-διόρθωση βαθμού ένα.

Θεωρούμε τον διαγώνιο πίνακα $D \in \mathbb{R}^{m \times m}$ με διακριτά στοιχεία $\{d_i\}$ στη διαγώνιό του και το διάνυσμα $w \in \mathbb{R}^m$. Αρκεί λοιπόν ισοδύναμα να βρούμε τις ιδιοτιμές του πίνακα $D + w w^T$ (Η επιλογή του πρόσημου + αντιστοιχεί για $\beta > 0$ ενώ αν είναι $\beta < 0$ τότε θα θεωρούσαμε $D - w w^T$). Θεωρούμε επίσης $w_i \neq 0$ για κάθε i διαφορετικά το πρόβλημά μας ανάγεται σε απλούστερο. Τότε οι ιδιοτιμές του πίνακα $D + w w^T$ είναι οι ρίζες της ρητής πολυωνυμικής συνάρτησης

$$f(\lambda) = 1 + \sum_{i=1}^m \frac{w_i^2}{d_i - \lambda}, \quad (2.60)$$



Σχήμα 2.5: Γραφική παράσταση της συνάρτησης 2.60 για πρόβλημα διάστασης 4. Οι ασύμπτωτες της $f(\lambda)$ είναι οι ιδιοτιμές $\{d_i\}$ του πίνακα D και οι ρίζες της $f(\lambda)$ είναι οι ιδιοτιμές του πίνακα $D + ww^T$. Η γρήγορη εύρεση των ριζών είναι η βάση κάθε αναδρομικού βήματος του αλγόριθμου διαίρει και βασίλευε.

της οποίας η γραφική παράσταση με τις ρίζες της φαίνεται στο σχήμα 2.5. Για να δείξουμε τον πιο πάνω ισχυρισμό έχουμε (με βάση το συλλογισμό στο [4] στην υποενότητα 30).

$$\begin{aligned} \text{Έστω } (D + ww^T)x &= \lambda x \quad \text{για κάποιο } x \neq 0 \Rightarrow (D - \lambda I)x + w(w^T x) = 0 \\ &\Rightarrow x + (D - \lambda I)^{-1}w(w^T x) = 0 \\ &\Rightarrow w^T x + w^T(D - \lambda I)^{-1}w(w^T x) = 0 \\ &\Rightarrow f(\lambda)(w^T x) = 0 \end{aligned}$$

Όπου το $w^T x \neq 0$ αλλιώς το x θα ήταν ιδιοδιάνυσμα του πίνακα D άρα μη μηδενικό ακριβώς ένα στοιχείο του (αφού ο D είναι διαγώνιος) και επομένως θα ήταν $w^T x \neq 0$ άτοπο. Επομένως αν το x είναι είναι ιδιοδιάνυσμα του πίνακα $D + ww^T$ με αντίστοιχη ιδιοτιμή τη λ , τότε το $f(\lambda)$ θα είναι 0 και το αντίστροφο ισχύει διότι η $f(\lambda)$ έχει ακριβώς m ρίζες.

Σε κάθε αναδρομικό βήμα του αλγόριθμου διαίρει και βασίλευε οι ρίζες της συνάρτησης 2.60 υπολογίζονται με μια γρήγορη επαναληπτική διαδικασία με βάση τη μέθοδο Newton-Raphson. Μόνο $O(1)$ επαναλήψεις απαιτούνται για κάθε ρίζα και άρα $O(m)$ πράξεις για κάθε ρίζα ενός $m \times m$ πίνακα, δηλαδή $O(m^2)$ πράξεις συνολικά για να βρούμε τις ζητούμενες ιδιοτιμές. Αν θεωρήσουμε μια αναδρομή όπου ο πίνακας διάστασης m χωρίζεται ακριβώς στη μέση σε κάθε βήμα το συνολικό κόστος για να βρούμε τις ιδιοτιμές ενός τριδιαγώνιου πίνακα με τον αλγόριθμο διαίρει και βασίλευε θα είναι:

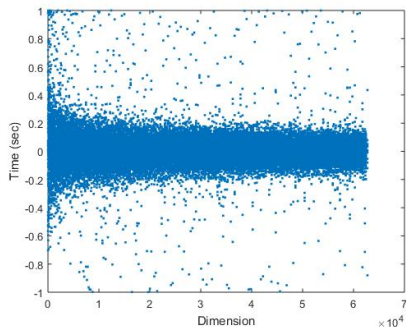
$$O\left(m^2 + 2\left(\frac{m}{2}\right)^2 + 4\left(\frac{m}{4}\right)^2 + \dots + m\left(\frac{m}{m}\right)^2\right), \quad (2.61)$$

Μία σειρά που συγκλίνει στο $O(m^2)$ (και όχι στο $O(m^2 \log m)$) λόγω των τετραγώνων στους παρονομαστές. Έτσι το συνολικό κόστος μετρήθηκε ίδιο με αυτό του αλγόριθμου QR.

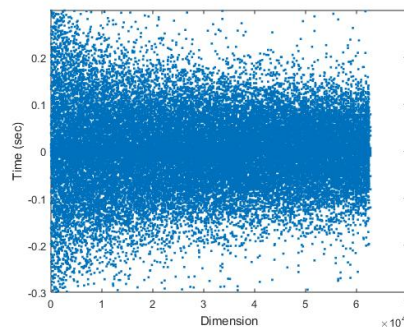
Μέχρι στιγμής δεν φαίνεται κάποιος λόγος ώστε ο αλγόριθμος διαίρει και βασίλευε να θεωρηθεί ανώτερος από τον αλγόριθμο QR. Αφού ο μετασχηματισμός ενός πλήρη πίνακα σε τριδιαγώνια μορφή (όπως είδαμε στην ενότητα III) απαιτεί $4m^3/3$ πράξεις (υποενότητα III.IV) είναι προφανές ότι όποια βελτίωση έχουμε στο κόστος $O(m^2)$ της διαγωνιοποίησης (δεύτερη φάση) είναι πολύ σημαντική. Όμως τα συνολικά κόστη σε πράξεις και βήματα αλλάζουν όταν μαζί με τις ιδιοτιμές υπολογίζουμε και τα ιδιοδιανύσματα. Σε αυτή την περίπτωση η πρώτη φάση απαιτεί $8m^3/3$ πράξεις αλλά η δεύτερη φάση και πάλι απαιτεί $O(m^3)$ πράξεις και μάλιστα στον αλγόριθμο QR η δεύτερη φάση απαιτεί $\approx 6m^3$. Ο αλγόριθμος διαίρει και βασίλευε βελτιώνει αυτό το συνολικό κόστος για τη δεύτερη φάση καθώς οι μη γραμμικές επαναλήψεις του εμπεριέχουν μόνο την πραγματική συνάρτηση 2.60 και όχι τους ορθογώνιους πίνακες Q_i όπου ο αλγόριθμος QR πρέπει να υπολογίζει σε κάθε επαναληπτικό βήμα.

Αν μετρήσουμε το συνολικό κόστος θα δούμε ότι το κομμάτι υπολογισμού που απαιτεί $O(m^3)$ της διαίρει και βασίλευε είναι οι πολλαπλασιασμοί με τους πίνακες Q_i και Q_i^T στην σχέση 2.59. Το συνολικό κόστος της αναδρομής σε όλα τα βήματα είναι $4m^3/3$ πράξεις το οποίο είναι μια σημαντική βελτίωση σε σχέση με το $\approx 6m^3$. Αν προσθέσουμε τις $8m^3/3$ πράξεις που απαιτεί η πρώτη φάση έχουμε μια συνολική μείωση του κόστους από τις $\approx 9m^3$ στις $4m^3$ πράξεις.

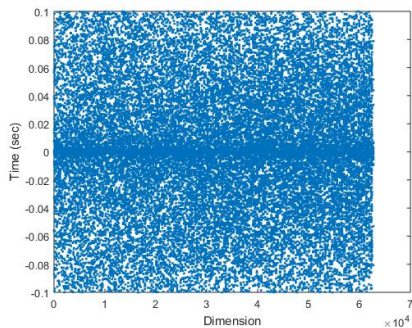
Επίσης ο αλγόριθμος διαίρει και βασίλευε συνήθως αποδίδει καλύτερα και από το προαναφερθέν κόστος για ένα λόγο που δεν είναι δομικός. Για τους περισσότερους πίνακες A πολλά από τα διανύσματα z και τους πίνακες Q_i που εμφανίζονται στην 2.59 ισχύει ότι είναι αριθμητικά αραιά με την έννοια ότι πολλά στοιχεία τους είναι πολύ κοντά στο μηδέν ώστε να τα θέσουμε ίσα με το μηδέν. Αυτή η αραιότητα επιτρέπει μια διαδικασία αριθμητικής αναγωγής μέσω της οποίας διαδοχικά τριδιαγώνια προβλήματα ιδιοτιμών ανάγονται σε ανεξάρτητα προβλήματα μικρότερης διάστασης. Σε μία τυπική τέτοια περίπτωση η μείωση του κόστους της δεύτερης φάσης θα είναι μικρότερης τάξης από ότι οι m^3 πράξεις μειώνοντας συνολικά το κόστος για την πρώτη και δεύτερη φάση μαζί στις $8m^3/3$ πράξεις. Για τον υπολογισμό μόνο των ιδιοτιμών η 2.61 δεν αντιστοιχεί στο νέο συνολικό κόστος καθώς η δεύτερη φάση μειώνεται σε τάξη μικρότερης των m^2 πράξεων.



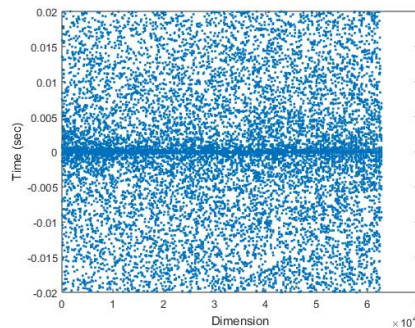
Σχήμα 2.6: Όλα τα στοιχεία των πρώτων και τελευταίων σειρών



Σχήμα 2.7: $-0.3 \leq Q(1, :)_i \leq 0.3$ ή $-0.3 \leq Q(m, :)_i \leq 0.3$



Σχήμα 2.8: $-0.1 \leq Q(1, :)_i \leq 0.1$ ή $-0.1 \leq Q(m, :)_i \leq 0.1$



Σχήμα 2.9: $-0.02 \leq Q(1, :)_i \leq 0.02$ ή $-0.02 \leq Q(m, :)_i \leq 0.02$

Στη MATLAB κατασκευάσαμε και τρέξαμε το παρακάτω script

```
clear all; vec=[]; sum=0;
for i=4:250
    A=cr_sym(randn(i)); A=red_hessenberg(A);
    [Q,~]=eig(A); nrm=norm(A); dl=eps()*nrm;
    sum=sum+length(find(Q(1,:)<=dl | Q(i,:)<=dl));
    vec=[vec Q(1,:) Q(i,:)];
end
pososto=sum/length(vec);
plot(1:length(vec),vec, '.')
```

Το παραπάνω script κατασκευάζει 247 τυχαίους, πραγματικούς και συμμετρικούς πίνακες διάστασης 4×4 έως και 250×250 , τους τριδιαγωνιοποιεί και υπολογίζει τους πίνακες ιδιοδιανυσμάτων κάθε τριδιαγωνίου πίνακα. Στη

συνέχεια επιλέγει την πρώτη και την τελευταία σειρά κάθε πίνακα ιδιοδιανυσμάτων και υπολογίζει το πλήθος των στοιχείων που είναι $\leq \epsilon_{machine} \|A_i\|_2$, ενώ αθροίζει το σύνολο αυτών των στοιχείων για όλους τους πίνακες ιδιοδιανυσμάτων.

Στην εκτέλεση αυτού του script από τα 62738 στοιχεία των πρώτων και τελευταίων σειρών των πινάκων ιδιοδιανυσμάτων τα 27098 ήταν "πολύ κοντά στο μηδέν", δηλαδή το 43.19% των στοιχείων! Αυτό το φαινόμενο είναι η βάση της μεγάλης ταχύτητας του αλγόριθμου διαίρει και βασίλευε.

VIII Παραγοντοποίηση SVD

Ο υπολογισμός της παραγοντοποίησης SVD ενός τυχαίου πίνακα μπορεί να αναχθεί στον υπολογισμό της διαγωνιοποίησης ενός ερμιτιανού, τετράγωνου πίνακα αλλά ο πιο προφανής τρόπος να το κάνουμε αυτό δεν είναι ευσταθής. Έτσι ο πιο αποδοτικός τρόπος για να υπολογίζουμε την παραγοντοποίηση SVD είναι η αναγωγή του προβλήματος σε διαφορετική ερμιτιανή μορφή. Για μεγαλύτερη ταχύτητα ο πίνακας αρχικά διδιαγωνιοποιείται ορθομοναδιαία.

VIII.1 Στοιχεία θεωρίας

Θεώρημα VIII.1. Έστω ο πίνακας $A \in \mathbb{C}^{m \times n}$. Τότε υπάρχουν ορθομοναδιαίοι πίνακες $U \in \mathbb{C}^{m \times m}$ και $V \in \mathbb{C}^{n \times n}$ τέτοιοι ώστε

$$A = U \text{diag}\{s_1, s_2, \dots, s_{\min(n,m)}\} V^*$$

Όπου $s_1 \geq s_2 \geq \dots \geq s_{\min(n,m)} \geq 0$.

Ο διαγώνιος πίνακας $\text{diag}\{s_1, s_2, \dots, s_{\min(n,m)}\}$ συμβολίζεται με Σ και οι τιμές

$$s_1 \geq s_2 \geq \dots \geq s_{\min(n,m)} \geq 0$$

καλούνται *ιδιάζουσες τιμές* του πίνακα A . Οι στήλες των πινάκων U και V λέγονται *αριστερά και δεξιά ιδιάζοντα διανύσματα* που αντιστοιχούν στην ιδιάζουσα τιμή s_i . (Περισσότερες λεπτομέρειες για το θεώρημα VIII.1 στο [4] θεώρημα 4.1.1).

Υπάρχουν θεμελιώδεις διαφορές μεταξύ της παραγοντοποίησης SVD και της διαγωνιοποίησης (χρησιμοποιώντας αναγκαστικά τις ιδιοτιμές και τα ιδιοδιανύσματα). Η μία είναι ότι η SVD χρησιμοποιεί δύο βάσεις (τα δεξιά και αριστερά ιδιάζουσα διανύσματα) ενώ η διαγωνιοποίηση μία (τα ιδιοδιανύσματα). Μία δεύτερη είναι ότι ο SVD χρησιμοποιεί ορθομοναδιαίες βάσεις ενώ η διαγωνιοποίηση μια βάση που δεν είναι αναγκαστικά ορθογώνια. Μία τρίτη είναι ότι διαγωνιοποίηση δεν υπάρχει για όλους τους πίνακες (ακόμα και τους τετράγωνους) ενώ η SVD υπάρχει για όλους (ακόμα και τους μη τετράγωνους πίνακες). Σε εφαρμογές οι ιδιοτιμές ενός πίνακα τείνουν να σχετίζονται με προβλήματα που περιλαμβάνουν τη συμπεριφορά επαναληπτικών μορφών του πίνακα A , όπως οι δυνάμεις του πίνακα A^k και η εκθετική του μορφή e^{tA} , ενώ τα ιδιάζοντα διανύσματα τείνουν να σχετίζονται με προβλήματα που περιλαμβάνουν τη συμπεριφορά του ίδιου του A ή του αντιστρόφου του A^{-1} .

Θεώρημα VIII.2. Ο βαθμός του πίνακα A είναι ίσος με τον αριθμό των μη μηδενικών ιδιζουσών τιμών του A

Απόδειξη. Με βάση το [1] θεώρημα 5.1.

Ο βαθμός ενός διαγώνιου πίνακα είναι ίσος με τον αριθμό των μη μηδενικών του στοιχείων. Στην παραγοντοποίηση $A = U\Sigma V^*$, οι U, V είναι πλήρης πίνακες. Επομένως $\text{rank}(A) = \text{rank}(\Sigma) = r$. \square

Θεώρημα VIII.3. Ο πυρήνας του πίνακα A είναι $\text{null}(A) = \langle v_{r+1}, \dots, v_n \rangle$ και η εικόνα του A είναι $\text{range}(A) = \langle u_1, \dots, u_r \rangle$

Απόδειξη. Με βάση το [1] θεώρημα 5.2.

Το θεώρημα είναι συνέπεια του γεγονότος ότι $\text{range}(\Sigma) = \langle e_1, \dots, e_r \rangle \subseteq \mathbb{C}^m$ και ότι $\text{null}(\Sigma) = \langle e_{r+1}, \dots, e_n \rangle \subseteq \mathbb{C}^m$. \square

Θεώρημα VIII.4. $\|A\|_2 = \sigma_1$ και $\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2}$

Απόδειξη. Με βάση το [1] θεώρημα 5.3.

Η απόδειξη της πρώτης σχέσης προκύπτει από το θεώρημα VIII.1. Αφού $A = U\Sigma V^*$ με ορθομοναδιαίους τους πίνακες U και V θα είναι $\|A\|_2 = \|\Sigma\|_2 = \max(|\sigma_i|) = \sigma_1$. Για τη δεύτερη σχέση αρκεί να πούμε ότι η νόρμα Frobenius είναι αναλλοίωτη όταν υπόκειται σε ορθομοναδιαίους μετασχηματισμούς, άρα $\|A\|_F = \|\Sigma\|_F$. \square

Θεώρημα VIII.5. Οι μη μηδενικές ιδιάζουσες τιμές του πίνακα A είναι οι τετραγωνικές ρίζες των μη μηδενικών ιδιοτιμών του πίνακα A^*A ή του AA^* . (Αυτοί οι πίνακες έχουν τις ίδιες μη μηδενικές ιδιοτιμές).

Απόδειξη. Με βάση το [1] θεώρημα 5.4.

Έχουμε

$$A^*A = (U\Sigma V^*)^*(U\Sigma V^*) = V\Sigma^*U^*U\Sigma V^* = V(\Sigma^*\Sigma)V$$

Άρα ο A^*A είναι όμοιος με τον $\Sigma^*\Sigma$ και έτσι έχουν τις ίδιες ιδιοτιμές. Οι ιδιοτιμές του πίνακα $\Sigma^*\Sigma$ είναι οι $\sigma_1^2, \sigma_2^2, \dots, \sigma_p^2$, όπου p είναι ο αριθμός των μη αρνητικών στοιχείων του πίνακα Σ και ακόμα $n - p$ μηδενικές ιδιοτιμές αν $n > p$. Για τον πίνακα AA^* κάνουμε τον αντίστοιχο υπολογισμό. \square

Λήμμα VIII.1. Τα αριστερά ιδιάζοντα διανύσματα του A είναι τα ιδιοδιανύσματα του AA^* και τα δεξιά ιδιάζοντα διανύσματα του A είναι τα ιδιοδιανύσματα του A^*A . (Περισσότερα στο [4] Κεφάλαιο 4).

Θεώρημα VIII.6. Αν $A = A^*$, τότε οι ιδιάζουσες τιμές του πίνακα A είναι οι απόλυτες τιμές των ιδιοτιμών του A .

Απόδειξη. Με βάση το [1] θεώρημα 5.5.

Όπως ξέρουμε ένας ερμιτιανός πίνακας έχει ένα πλήρες σύνολο ορθογώνιων ιδιοδιανυσμάτων και όλες οι ιδιοτιμές του είναι πραγματικές. Έτσι έχουμε

$$A = Q\Lambda Q^* = Q|\Lambda| \text{sign}(\Lambda) Q^* \quad (2.62)$$

Όπου $|\Lambda|$ ο διαγώνιος πίνακας με στοιχεία τις διαγωνίου του τα $|\lambda_i|$ και όπου $\text{sign}(\Lambda)$ τον διαγώνιο πίνακα με στοιχεία της διαγωνίου του τα $\text{sign}(\lambda_i)$. Αφού ο Q είναι ορθομοναδιαίος τότε και ο $\text{sign}(\Lambda)Q^*$ θα είναι ορθομοναδιαίος, επομένως η σχέση 2.62 ορίζει μια SVD παραγοντοποίηση του A με ιδιάζουσες τιμές τα διαγώνια στοιχεία του $|\Lambda|, |\lambda_i|$. \square

Θεώρημα VIII.7. Έστω πίνακας $A \in \mathbb{C}^{m \times m}$, τότε $|\det(A)| = \prod_{i=1}^m \sigma_i$

Απόδειξη. Με βάση το [1] θεώρημα 5.6.

Η ορίζουσα ενός γινομένου τετράγωνων πινάκων είναι ίση με το γινόμενο των ορίζουσών των πινάκων. Επιπλέον λόγω των ιδιοτήτων $U^*U = I$ και $\det(U^*) = (\det(U))^*$, η απόλυτη τιμή της ορίζουσας ενός ορθογώνιου πίνακα είναι πάντα ίση με το 1. Επομένως θα έχουμε

$$|\det(A)| = |\det(USV^*)| = |\det(U)||\det(\Sigma)||\det(V^*)| = |\det(\Sigma)| = \prod_{i=1}^m \sigma_i$$

\square

VIII.II Οι ιδιοτιμές του A^*A και παραγοντοποίηση SVD

Όπως αναφέραμε και στο θεώρημα VIII.5 η παραγοντοποίηση SVD του $m \times n$ πίνακα $A (m \geq n)$, $A = USV^*$ σχετίζεται με τη διαγωνιοποίηση του πίνακα A^*A ως εξής

$$A^*A = V(\Sigma^*\Sigma)V^*$$

Επομένως από μαθηματικής άποψης μπορούμε να υπολογίσουμε την παραγοντοποίηση SVD του A αν ακολουθήσουμε τα παρακάτω βήματα

1. Παίρνουμε τον πίνακα A^*A
2. Υπολογίζουμε τη διαγωνιοποίηση του $A^*A = V\Lambda V^*$
3. Έστω Σ ο $m \times n$ μη αρνητικός διαγώνιος πίνακας με στοιχεία του τις τετραγωνικές ρίζες των στοιχείων του Λ .
4. Λύνουμε το σύστημα $U\Sigma = AV$ και βρίσκουμε τον ορθομοναδιαίο πίνακα U .

Ο πίνακας A^*A είναι γνωστός και ως πίνακας συνδιακύμανσης του πίνακα A και έχει διάφορες εφαρμογές στη στατιστική και σε άλλα πεδία. Παρόλα αυτά ο αλγόριθμος που ορίζεται από τα παραπάνω βήματα είναι ασταθής κυρίως επειδή ανάγει το SVD πρόβλημα σε ένα πρόβλημα ιδιοτιμών που συνήθως είναι πιο ευαίσθητο σε διαταραχές.

Είδαμε ότι όταν θεωρούμε μια διαταραχή δB του ερμιτιανού πίνακα A^*A οι απόλυτες διαφορές σε κάθε ιδιοτιμή φράζονται από την ευκλείδεια νόρμα της διαταραχής, δηλαδή $|\lambda_k(A^*A + \delta B) - \lambda_k(A^*A)| \leq \|\delta B\|_2$. Όπως συμπεραίνουμε από την 2.64 παρακάτω ένα παρόμοιο άνω φράγμα υπάρχει

και για τις ιδιάζουσες τιμές του A , δηλαδή $|\sigma_k(A + \delta A) - \sigma_k(A)| \leq \|\delta A\|_2$. Έτσι ένας προς τα πίσω ευσταθής αλγόριθμος υπολογισμού των ιδιάζουσών τιμών θα υπολόγιζε τα $\tilde{\sigma}_k$ για τα οποία θα ισχύει

$$\tilde{\sigma}_k = \sigma_k(A + \delta A), \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}) \quad (2.63)$$

Επομένως

$$|\tilde{\sigma}_k - \sigma_k| = O(\epsilon_{machine})$$

Έστω ότι υπολογίζουμε το $\lambda_k(A^*A)$ με ένα ευσταθή τρόπο. Επομένως θα περιμένουμε σφάλματα τάξης

$$|\tilde{\lambda}_k - \lambda_k| = O(\epsilon_{machine}\|A^*A\|) = O(\epsilon_{machine}\|A\|^2)$$

Άρα για τα σ_k θα έχουμε

$$|\tilde{\sigma}_k - \sigma_k| = O(|\tilde{\lambda}_k - \lambda_k|/\sqrt{\lambda_k}) = O(\epsilon_{machine}\|A\|^2/\sigma_k)$$

Το παραπάνω αποτέλεσμα για τα σ_k είναι χειρότερο από το προηγούμενο κατά τον παράγοντα $O(\|A\|/\sigma_k)$. Αυτό δεν είναι πρόβλημα για τις ιδιάζουσες τιμές με $\sigma_k \approx \|A\|$ αλλά είναι μεγάλο πρόβλημα για τις ιδιάζουσες τιμές για τις οποίες θα είναι $\sigma_k \ll \|A\|$. Για τη μικρότερη ιδιάζουσα τιμή πρέπει να περιμένουμε μια απώλεια ακρίβειας τάξης $\kappa(A)$.

VIII.III Μια διαφορετική αναγωγή σε πρόβλημα ιδιοτιμών

Υπάρχει ένας εναλλακτικός, ευσταθής τρόπος να ανάγουμε το πρόβλημα παραγοντοποίησης SVD σε ένα πρόβλημα ιδιοτιμών. Υποθέτουμε ότι ο πίνακας A είναι τετράγωνος (δηλαδή $m = n$). Δεν πρόκειται για κάποιο περιορισμό που βλάπτει τη γενικότητα καθώς θα δούμε ότι και μη τετραγωνικά προβλήματα SVD μπορούν να αναχθούν σε τετράγωνα. Έστω ο $2m \times 2m$ ερμιτιανός πίνακας

$$H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix} \quad (2.64)$$

Αφού η σχέση $A = U\Sigma V^*$ συνεπάγεται τις $AV = U\Sigma$ και $A^*U = V\Sigma^* = V\Sigma$ θα ισχύει

$$\begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix} \begin{bmatrix} V & V \\ U & -U \end{bmatrix} = \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix} \quad (2.65)$$

Σχέση η οποία αντιστοιχεί σε μια διαγωνιοποίηση του πίνακα H . Έτσι βλέπουμε ότι οι ιδιάζουσες τιμές του A είναι οι απόλυτες τιμές των ιδιοτιμών του H και τα ιδιάζοντα διανύσματα του A είναι τα ιδιοδιανύσματα του H .

Έτσι θα μπορούσαμε να υπολογίσουμε την παραγοντοποίηση SVD αν κατασκευάζαμε τον πίνακα H και υπολογίζαμε τη διαγωνιοποίησή του (άρα τις ιδιοτιμές και τα ιδιοδιανύσματά του). Σε αντίθεση με τον αλγόριθμο της προηγούμενης υποενότητας αυτή η προσέγγιση είναι ευσταθής. Οι βασικοί αλγόριθμοι υπολογισμού της παραγοντοποίησης SVD βασίζονται σε αυτή την ιδέα αν και για τους μη τετραγωνικούς πίνακες θα αναγκαστούμε προκειμένου να κάνουμε τη διαδικασία πιο γρήγορη να ορίσουμε δύο φάσεις. Η πρώτη φάση είναι ένα αρχικό βήμα ώστε να μετασχηματίσουμε τον πίνακά μας σε διδιαγώνιο.

VIII.IV Δύο φάσεις

Είδαμε ότι τα ερμιτιανά προβλήματα ιδιοτιμών συνήθως λύνονται με έναν υπολογισμό που χωρίζεται σε δύο φάσεις: η πρώτη του μετασχηματισμού του αρχικού μας πίνακα σε τριδιαγώνιο και η δεύτερη της διαγωνιοποίησης του τριδιαγώνιου πίνακα. Τη δεκαετία του '60 οι Golub και Kahan περιέγραψαν μια ανάλογη διαδικασία δύο φάσεων για τον υπολογισμό της παραγοντοποίησης SVD. Η πρώτη φάση μετατρέπει τον πίνακά μας σε διδιαγώνιο και η δεύτερη διαγωνιοποιεί τον τελευταίο.

$$\begin{array}{ccc}
 \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} & \xrightarrow{\text{1η Φάση}} & \begin{bmatrix} x & & & \\ & x & & \\ & & x & \\ & & & x \end{bmatrix} & \xrightarrow{\text{2η Φάση}} & \begin{bmatrix} x & & & \\ & x & & \\ & & x & \\ & & & x \end{bmatrix} \\
 A & & B & & \Sigma
 \end{array}$$

Η πρώτη φάση περιλαμβάνει ένα πεπερασμένο αριθμό διαδικασιών, $O(mn^2)$ πράξεων. Η δεύτερη φάση θεωρητικά απαιτεί άπειρο αριθμό διαδικασιών αλλά ο βασικός αλγόριθμος συγκλίνει υπεργραμμικά και έτσι απαιτούνται μόλις $O(n \log(|\log(\epsilon_{machine})|))$ επαναλήψεις για σύγκλιση τάξης $\epsilon_{machine}$. Στην πράξη θεωρούμε το $\epsilon_{machine}$ ως μεταβλητή και επομένως η σύγκλιση θα επιτυγχάνεται σε $O(n)$ επαναλήψεις. Επειδή ο πίνακας πάνω στον οποίο εκτελούνται οι πράξεις είναι διδιαγώνιος κάθε μία από τις επαναλήψεις απαιτεί μόλις $O(n)$ πράξεις. Η δεύτερη φάση επομένως απαιτεί $O(n^2)$ πράξεις συνολικά (θεωρώντας ότι ζητούνται οι ιδιάζουσες τιμές και όχι τα ιδιάζοντα διανύσματα). Έτσι βλέπουμε ότι η πρώτη φάση παρόλο που είναι πεπερασμένη έχει μεγαλύτερο κόστος από τη δεύτερη που θεωρητικά είναι άπειρη (το ίδιο είδαμε ότι συμβαίνει και για τις δύο φάσεις των αλγορίθμων υπολογισμού ιδιοτιμών).

VIII.V Διδιαγωνιοποίηση Golub-Kahan

Στην πρώτη φάση λοιπόν του υπολογισμού της SVD φέρνουμε τον πίνακα A σε διδιαγώνια μορφή εκτελώντας ορθομοναδιαίους πολλαπλασιασμούς από τα αριστερά και από τα δεξιά. Η βασική διαφορά με την αντίστοιχη πρώτη φάση στον υπολογισμό ιδιοτιμών είναι ότι στον υπολογισμό ιδιοτιμών ο δεξιά πολλαπλασιασμός γινόταν βάση του ίδιου πίνακα (με τον αντίστροφο) με τον αριστερά πολλαπλασιασμό και ήταν δυνατή η εισαγωγή μηδενικών μόνο κάτω από την υπερδιαγώνιο. Στην περίπτωση του SVD μπορούμε να τριγωνοποιήσουμε πλήρως τον πίνακά μας καθώς και να εισάγουμε μηδενικά πάνω από την υπερδιαγώνιο, αφού μπορούμε να πολλαπλασιάσουμε με διαφορετικούς πίνακες από αριστερά και από δεξιά.

Ο πιο απλός τρόπος υλοποίησης του αλγόριθμου διδιαγωνιοποίησης των Golub-Kahan είναι ο πολλαπλασιασμός εναλλάξ από αριστερά και από δεξιά με κατάλληλους πίνακες Householder που θα εισάγουν μηδενικά στον πίνακα μας. Κάθε από αριστερά πολλαπλασιασμός εισάγει μια στήλη από μηδενικά κάτω από το εκάστοτε στοιχείο της διαγώνιου. Ο από δεξιά πολλαπλασιασμός που ακολουθεί εισάγει μια σειρά μηδενικών δεξιά από την πρώτη υπερδιαγώνιο αφήνοντας έτσι ανέπαφα τα μηδενικά που εισήχθησαν στις προηγούμενες στήλες. Για παράδειγμα για ένα 6×4 πίνακα τα δύο πρώτα ζευγάρια Householder πινάκων εκτελούν, σχηματικά, την εξής διαδικασία

$$\begin{array}{ccc}
 \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} & \xrightarrow{U_1^*} & \begin{bmatrix} x & x & x & x \\ \mathbf{0} & x & x & x \\ \mathbf{0} & x & x & x \\ \mathbf{0} & x & x & x \\ \mathbf{0} & x & x & x \\ \mathbf{0} & x & x & x \end{bmatrix} & \xrightarrow{\cdot V_1} & \begin{bmatrix} x & x & \mathbf{0} & \mathbf{0} \\ & x & x & x \\ & x & x & x \\ & x & x & x \\ & x & x & x \\ & x & x & x \end{bmatrix} \\
 A & & U_1^* A & & U_1^* A V_1
 \end{array}$$

$$\begin{array}{ccc}
 & \xrightarrow{U_2^*} & \begin{bmatrix} x & x & & \\ & x & x & x \\ & \mathbf{0} & x & x \\ & \mathbf{0} & x & x \\ & \mathbf{0} & x & x \\ & \mathbf{0} & x & x \end{bmatrix} & \xrightarrow{\cdot V_2} & \begin{bmatrix} x & x & & \\ & x & x & \mathbf{0} \\ & & x & x \\ & & x & x \\ & & x & x \\ & & x & x \end{bmatrix} \\
 & & U_2^* U_1^* A V_1 & & U_2^* U_1^* A V_1 V_2
 \end{array}$$

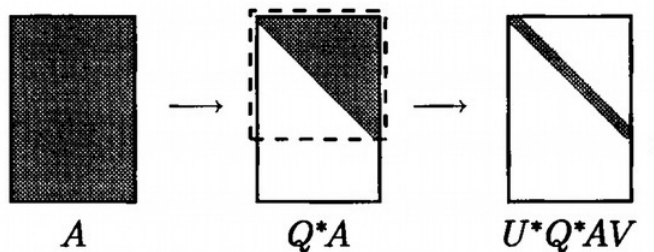
Ο από τα αριστερά πολλαπλασιασμός με τον πίνακα U_1^* μετασχηματίζει τις σειρές 1 έως και τη m , εισάγοντας μηδενικά στην πρώτη στήλη κάτω από τη διαγώνιο. Ο από τα δεξιά πολλαπλασιασμός με τον πίνακα V_1 μετασχηματίζει τις στήλες 2 έως n , εισάγοντας μηδενικά στην πρώτη σειρά μετά το στοιχείο της πρώτης υπερδιαγώνιου, αφήνοντας αναλλοίωτα τα μηδενικά που εισήγαγε ο U_1^* . Η διαδικασία συνεχίζεται με τους επόμενους πολλα-

πλασιασμούς να μετασχηματίζουν τις σειρές 2 έως m (ο από τα αριστερά πολλαπλασιασμός) και τις στήλες 3 έως n (ο από τα δεξιά πολλαπλασιασμός). Στο τέλος της διαδικασίας n πίνακες θα έχουν εφαρμοστεί από τα αριστερά και $n - 2$ πίνακες από τα δεξιά.

$$\text{Κόστος για διδιαγωνιοποίηση Golub-Kahan:} \quad \approx 4mn^2 - \frac{4}{3}n^3 \quad \text{πράξεις} \quad (2.66)$$

VIII.VI Ταχύτερες μέθοδοι για την πρώτη φάση

Για $m \gg n$ η διαδικασία που περιγράφηκε στην προηγούμενη υποενότητα είναι αχρείαστα μεγάλη. Μόνη της η QR παραγοντοποίηση μπορεί να εισάγει μηδενικά σε όλες τις θέσεις κάτω από τη διαγώνιο και για $m \gg n$ αυτά είναι σχεδόν όλα τα μηδενικά που χρειαζόμαστε. Όμως η μέθοδος των Golub-Kahan έχει το διπλάσιο κόστος από την QR παραγοντοποίηση. Αυτή η παρατήρηση μας οδηγεί στο να ανατρέξουμε σε κάποια εναλλακτική μέθοδο διδιαγωνιοποίησης για $m \gg n$. Μια τέτοια μέθοδος προτάθηκε από τους Lawson-Hanson και αργότερα βελτιώθηκε από τον Chan. Η ιδέα της μεθόδου διδιαγωνιοποίησης LHC φαίνεται σχηματικά παρακάτω.

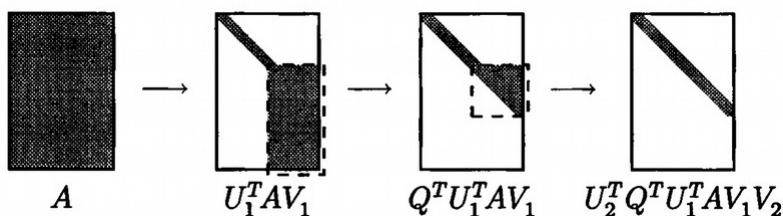


Σχήμα 2.10: Διδιαγωνιοποίηση των Lawson-Hanson-Chan

Η διαδικασία έχει ως εξής: αρχικά υπολογίζουμε την παραγοντοποίηση QR, $A = QR$. Τότε υπολογίζουμε τη διδιαγωνιοποίηση των Golub-Kahan, $B = U^*RV$ του R . Η παραγοντοποίηση QR απαιτεί $2mn^2 - \frac{2}{3}n^3$ πράξεις και η διαδικασία των Golub-Kahan, η οποία έχει πλέον να μετασχηματίσει μόνο τον άνω $n \times n$ υποπίνακα, απαιτεί $\frac{8}{3}n^3$ πράξεις. Άρα

$$\text{Κόστος για LHC διδιαγωνιοποίηση:} \quad \approx 2mn^2 + 2n^3 \quad \text{πράξεις} \quad (2.67)$$

Εύκολα μπορούμε να δείξουμε ότι η παραπάνω μέθοδος είναι πιο γρήγορη από τη μέθοδο των Golub-Kahan για $m > \frac{5}{3}n$. Η ιδέα όμως της μεθόδου μπορεί να γενικευθεί ώστε να έχουμε μια εξοικονόμηση πράξεων για κάθε $m > n$. Η τεχνική έχει να κάνει με τη χρησιμοποίηση της QR παραγοντοποίησης όχι στην αρχή του υπολογισμού αλλά σε μια κατάλληλη στιγμή στη μέση της διαδικασίας. Αυτή η ιδέα είναι αρκετά αποδοτικότερη διότι αν σε ένα πίνακα με $m > n$ εκτελέσουμε τη μέθοδο Golub-Kahan ο λόγος $\frac{m}{n}$ θα αυξάνεται σε κάθε βήμα, αφού μετά από κάθε βήμα ο λόγος θα γίνεται ίσος με το πηλίκο $(m-k)/(n-k)$. Άρα όταν αυτός ο λόγος γίνει αρκετά μεγάλος τότε εκτελούμε την παραγοντοποίηση QR και ανάγουμε το πρόβλημά μας σε διδιαγώνιο.



Σχήμα 2.11: Διδιαγωνιοποίηση των τριών βημάτων

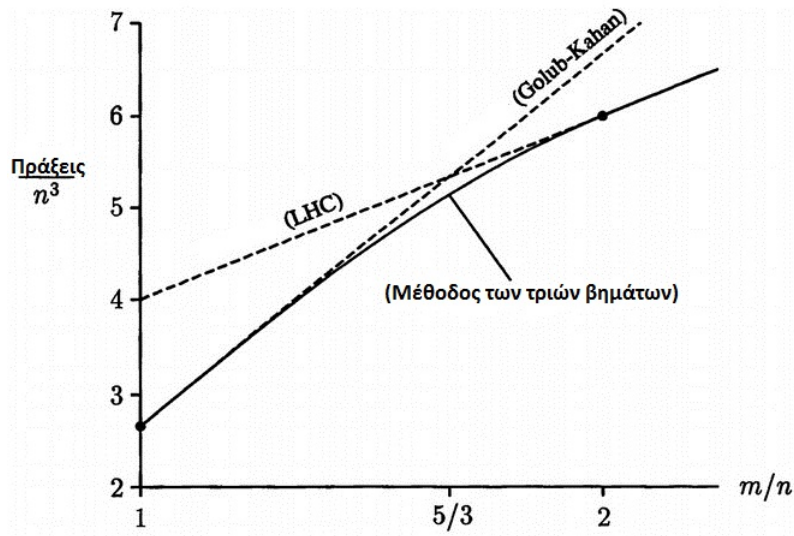
Αυτή τη διαδικασία την ονομάζουμε μέθοδο των τριών βημάτων και φαίνεται στο σχήμα 2.11.

Πότε όμως πρέπει να εκτελεστεί η παραγοντοποίηση QR. Αν το κριτήριό μας είναι αποκλειστικά να ελαχιστοποιήσουμε το κόστος η απάντηση προκύπτει εύκολα: Όταν ο λόγος γίνει $(m-k)/(n-k) = 2$. Έτσι θα έχουμε το παρακάτω συμπέρασμα

$$\begin{aligned} &\text{Κόστος για διδιαγωνιοποίηση με τη μέθοδο των τριών βημάτων:} \\ &\approx 4mn^2 - \frac{4}{3}n^3 - \frac{2}{3}(m-n)^3 \text{ πράξεις.} \end{aligned}$$

Μία όχι αμελητέα αλλά όχι και τόσο θεαματική βελτίωση σε σχέση με τις δύο προηγούμενες μεθόδους για $n < m < 2n$.

Το σύνολο του κόστους και για τις τρεις μεθόδους φαίνεται σχηματικά στο παρακάτω διάγραμμα ως συνάρτηση του λόγου m/n . Η αλήθεια είναι ότι η βελτίωση που επιτυγχάνεται με τη μέθοδο των τριών βημάτων είναι τόσο μικρή που στην πράξη άλλα ζητήματα καθορίζουν την επιλογή της μεθόδου όταν υλοποιούμε το πρώτο βήμα.



Σχήμα 2.12

VIII.VII Δεύτερη φάση

Στη δεύτερη φάση υπολογισμού της παραγοντοποίησης SVD υπολογίζουμε την SVD του διαγωνίου πίνακα. Από τη δεκαετία του '60 μέχρι και τη δεκαετία του '90 ο κύριος αλγόριθμος που χρησιμοποιούνταν ήταν μια εκδοχή του αλγόριθμου QR. Αργότερα οι διαίρει και βασίλευε αλγόριθμοι έγιναν πιο ανταγωνιστικοί. Οι λεπτομέρειες ξεφεύγουν από τα όρια της παρούσας εργασίας.

Κεφάλαιο 3

Υλοποίηση στο MATLAB

Σε αυτό το κεφάλαιο θα υλοποιήσουμε στο MATLAB τους αλγόριθμους που περιγράψαμε στο δεύτερο κεφάλαιο και θα δώσουμε κάποια χαρακτηριστικά παραδείγματα εύρεσης ιδιοτιμών, θα συγκρίνουμε αλγόριθμους μεταξύ τους και με τους αλγόριθμους του MATLAB και θα μετρήσουμε χρόνους σύγκλισης. Οδηγός μας θα είναι το δεύτερο κεφάλαιο αλλά σε κάποιες περιπτώσεις θα δώσουμε παραπάνω διευκρινήσεις που έχουν να κάνουν με την πρακτική υλοποίηση των αλγόριθμων. Σημειώνουμε ότι οι υλοποιήσεις είναι του γράφοντος με ό,τι αυτό συνεπάγεται για τα υποκειμενικά όριά τους.

I Μετασχηματισμός σε μορφή Hessenberg

Η υλοποίηση που θα παρουσιάσουμε στο MATLAB βασίζεται στον αλγόριθμο 2.1. Θα πρέπει όμως να δώσουμε κάποιες λεπτομέρειες για την κατασκευή των πινάκων Householder που χρησιμοποιούμε σε κάθε επανάληψη.

Έστω ότι είμαστε στην πρώτη επανάληψη. Τότε θα πρέπει να κατασκευάσουμε τον πίνακα Householder Q_1^* που περιγράψαμε στις υποενότητες του δεύτερου κεφαλαίου, ώστε το διάνυσμα $Q_1^* \alpha_1$ να ανήκει στο $\text{span}(e_1)$, όπου α_1 η πρώτη στήλη του πίνακα A χωρίς το στοιχείο της πρώτης γραμμής, όπως φαίνεται στο σχήμα παρακάτω:

$$A = \begin{bmatrix} x & x & x & x & x \\ \mathbf{x} & x & x & x & x \\ \mathbf{x} & x & x & x & x \\ \mathbf{x} & x & x & x & x \\ \mathbf{x} & x & x & x & x \end{bmatrix}$$

Θα βρούμε τον πίνακα Householder $\tilde{Q}_1^* \in \mathbb{C}^{(m-1) \times (m-1)}$ ώστε: $\tilde{Q}_1^* \alpha_1 \in \text{span}(e_1)$.

Άρα ο πίνακας $Q_1^* \in \mathbb{C}^{m \times m}$ θα είναι ο

$$Q_1^* = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_1^* \end{bmatrix}$$

Αφού

$$\begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_1^* \end{bmatrix} A = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}$$

Και

$$\begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_1^* \end{bmatrix} A \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_1^* \end{bmatrix}^{-1} = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{bmatrix}$$

Επομένως στο k -οστό βήμα θα κατασκευάσουμε τον ορθογώνιο πίνακα Householder $\tilde{Q}_k^* \in \mathbb{C}^{(m-k) \times (m-k)}$ ώστε $\tilde{Q}_k^* \alpha_k \in \text{span}(e_1)$. Τότε ο ζητούμενος πίνακας $Q_k^* \in \mathbb{C}^{m \times m}$ θα είναι ο

$$Q_k^* = \begin{bmatrix} I_k & 0 \\ 0 & \tilde{Q}_k^* \end{bmatrix}$$

Ενώ συνολικά θα έχουμε $k-2$ βήματα. Επομένως μια υλοποίηση του αλγόριθμου 2.1 στο MATLAB θα μπορούσε να είναι η παρακάτω.

Αλγόριθμος 3.1

Householder μετασχηματισμός σε μορφή Hessenberg
Υλοποίηση στο MATLAB

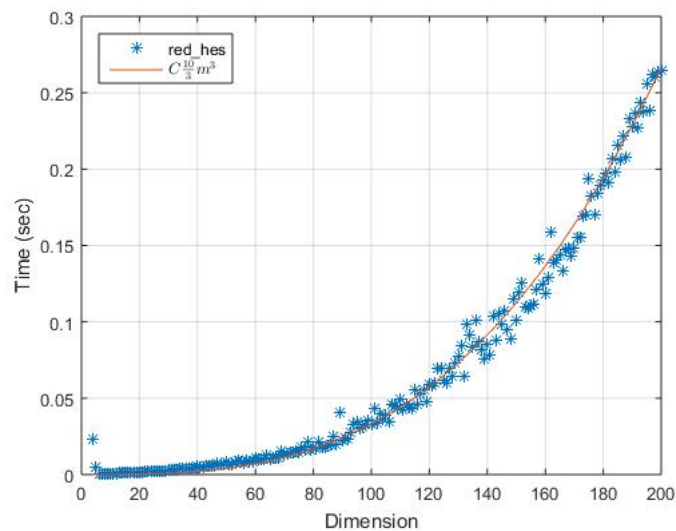
```
function [c]=red_hessenberg(a)
m=length(a);
if nargin==1
    for k=1:m-2
        w=a(k+1:m,k); e=eye(m-k);
        e1=e(:,1); u=(sign(w(1))*norm(w))*e1+w;
        u=u/norm(u); p=e-2*(u*transpose(u));
        p=blkdiag(eye(k),p);
        a=p*a; a=a*p';
        a(k+2:m,k)=0; a(k,k+2:m)=0;
    end
else
```

```

Qt=eye(m);
for k=1:m-2
    w=a(k+1:m,k); e=eye(m-k);
    e1=e(:,1); u=(sign(w(1))*norm(w))*e1+w;
    u=u/norm(u); p=e-2*(u*transpose(u));
    p=blkdiag(eye(k),p); a=p*a;
    a=a*p'; Qt=Qt*p;
    a(k+2:m,k)=0; a(k,k+2:m)=0;
end
end
c=a;
end

```

Ο αλγόριθμος 3.1 μας επιτρέπει αν δηλώσουμε και δεύτερη μεταβλητή εξόδου να πάρουμε τον από δεξιά πίνακα του μετασχηματισμού Householder, το οποίο θα μας χρειαστεί αργότερα στον αλγόριθμο QR με εναλλαγές ώστε να υπολογίσουμε τον πίνακα των ιδιοδιανυσμάτων.



Σχήμα 3.1: Οι χρόνοι εκτέλεσης των πράξεων στον αλγόριθμο της Householder μετασχηματισμού σε μορφή Hessenberg για πίνακες διάστασης 4×4 έως και 200×200 .

Στο δεύτερο κεφάλαιο δείξαμε ότι ο αλγόριθμος του Householder μετασχηματισμού σε Hessenberg μορφή απαιτεί συνολικά $\sim \frac{10}{3}m^3$ πράξεις. Έτσι αν στον κορμό του κώδικα προσθέσουμε τις παρακάτω εντολές

```

:
tic

```

```

a=p*a;
a=a*p';
t=toc;
time=time+t;
:

```

θα μετρήσουμε πόσο χρόνο κόστισε η εκτέλεση των πράξεων. Στο σχήμα 3.1 παραθέτουμε ένα διάγραμμα στο οποίο φαίνονται οι χρόνοι εκτέλεσης των πράξεων για πίνακες διάστασης 4×4 έως και 200×200 . Η συνεχής καμπύλη του διαγράμματος είναι η $10^{-8} \times \frac{10}{3} m^3$. Βλέπουμε ότι η εκτίμηση των σημείων με βάση την καμπύλη $C \frac{10}{3} m^3$ για $C \approx 10^{-8}$ (στην περίπτωση μας) είναι απόλυτα λογική, γεγονός που είναι σύμφωνο με τη θεωρία.

II Πηλίκιο Rayleigh και αντίστροφη επανάληψη

Σε αυτή την ενότητα θα υλοποιήσουμε τον αλγόριθμο των δυνάμεων, της αντίστροφης επανάληψης και της επανάληψης πηλίκιου Rayleigh με οδηγό την αντίστοιχη ενότητα του δεύτερου κεφαλαίου και στο τέλος θα συγκρίνουμε τους αλγόριθμους που υλοποιήσαμε στο MATLAB.

II.I Μέθοδος των δυνάμεων

Το συγκεκριμένο αλγόριθμο περιγράψαμε πλήρως στο δεύτερο κεφάλαιο. Επομένως η υλοποίηση στο MATLAB βασίστηκε στον αλγόριθμο 2.2. Η μόνη εκκρεμότητα που αφήσαμε για αυτή την υποενότητα είναι η επιλογή της συνθήκης τερματισμού της επαναληπτικής διαδικασίας. Θεωρητικά και πρακτικά δεν υπάρχει μία μόνο συνθήκη τερματισμού που θα μπορούσαμε να επιλέξουμε. Αυτή η συνθήκη θα πρέπει να ελέγχει το βαθμό σύγκλισης του ιδιοδιανύσματος ή της ιδιοτιμής από το $(k-1)$ στο k βήμα. Έτσι εμείς επιλέξαμε η επαναληπτική διαδικασία να τερματίζεται όταν $|1 - |v^{(k-1)T} v^k|| \leq \epsilon_{machine}$ ή όταν ο αριθμός των επαναλήψεων υπερβεί τις 400, αριθμός που θεωρείτε υπεραρκετός για την επίτευξη της σύγκλισης. Έτσι πρακτικά ο αλγόριθμος θα πάρει την παρακάτω μορφή

Αλγόριθμος 3.2

```

Μέθοδος των δυνάμεων
Υλοποίηση στο MATLAB

function [q,l]=power_iteration(A,v0)
k=2;
q=v0;
n=0;
while k>eps() && n<=400
    k=q;
    q=A*q;
    q=q/norm(q);
    k=abs(abs(k'*q)-1);
    l=q'*A*q;
    n=n+1;
end
end

```

Ουσιαστικά εκμεταλλευόμαστε το γεγονός ότι δημιουργείται μια ακολουθία κανονικοποιημένων διανυσμάτων. Όταν το $v^{(k-1)T}v^k$ είναι αρκετά κοντά στο ένα ουσιαστικά η σύγκλιση στο ιδιοδιάνυσμα που αντιστοιχεί στη μεγαλύτερη κατά απόλυτη τιμή ιδιοτιμή είναι αρκετά ικανοποιητική. Οι απόλυτες τιμές έχουν να κάνουν με την τυχαιότητα των προσήμων.

II.II Αλγόριθμος αντίστροφης επανάληψης

Και σε αυτή την περίπτωση υλοποίησης του αλγόριθμου στο MATLAB τα περισσότερα τα αναφέραμε στο δεύτερο κεφάλαιο, με μόνη εκκρεμότητα την επιλογή της συνθήκης τερματισμού. Η συνθήκη που επιλέξαμε είναι ίδια με την περίπτωση της μεθόδου των δυνάμεων. Έτσι η υλοποίηση του αλγόριθμου πήρε την παρακάτω μορφή

```

Αλγόριθμος 3.3

Αλγόριθμος αντίστροφης επανάληψης
Υλοποίηση στο MATLAB

function [q,l]=inverse_it(A,m,v0)
n=length(A);
A=A-m*eye(n);
q=v0;
k=1;
n=0;

```

```

while k>eps() && n<=400
    k1=q;
    q=A\q;
    q=q/norm(q);
    k=abs(abs(k'*q)-1);
    l=q'*A*q;
    n=n+1;
end
end

```

II.III Επανάληψη πηλίκου Rayleigh

Η υλοποίηση του αλγόριθμου σε αυτή την περίπτωση όπως και στις δύο προηγούμενες περιγράφηκε με αρκετή ακρίβεια στο δεύτερο κεφάλαιο, με εκκρεμότητα τη συνθήκη τερματισμού της επαναληπτικής διαδικασίας. Η συνθήκη τερματισμού επιλέχθηκε να είναι η ίδια με τους προηγούμενους αλγορίθμους.

Αλγόριθμος 3.4

Αλγόριθμος επανάληψης πηλίκου Rayleigh
Υλοποίηση στο MATLAB

```

function [q,l]=ray_quot(A,v0,cl)
m=length(A);
n=0;
k=1;
if nargin==3
    A1=A-cl*eye(m);
    q=A1\v0;
    q=q/norm(q);
    l=q'*A*q;
else
    q=v0;
    l=q'*A*q;
end
while k>eps() && n<=400
    k=q;
    A2=A-l*eye(m);
    q=A2\q;
    q=q/norm(q);

```

```

    k=abs(abs(k*q)-1);
    l=q'*A*q;
    n=n+1;
end
end

```

Στον αλγόριθμο επανάληψης πηλίκου Rayleigh σε κάθε βήμα χρησιμοποιούμε μια προσέγγιση της ιδιοτιμής λ_J για να υπολογίσουμε μια προσέγγιση του ιδιοδιανύσματος q_J και αντίστροφα. Βέβαια όλα αυτά τα βήματα προϋποθέτουν μια αρχική προσέγγιση είτε της ιδιοτιμής είτε του ιδιοδιανύσματος. Με τον έλεγχο [if nargin== 3 ... else] επιλέγουμε ποια θα είναι η αρχική μας προσέγγιση. Αν ο αλγόριθμος μας έχει τρία ορίσματα τότε σημαίνει ότι έχουμε μια αρχική προσέγγιση της ιδιοτιμής λ_J και ο αλγόριθμός μας θα υπολογίσει με βάση το πρώτο βήμα του αλγόριθμου της αντίστροφης επανάληψης μια πρώτη προσέγγιση του ιδιοδιανύσματος q_J . Αν τα ορίσματα είναι δύο τότε σημαίνει πως έχουμε μια αρχική προσέγγιση του ιδιοδιανύσματος q_J και άρα θα υπολογίσουμε μια προσέγγιση της ιδιοτιμής λ_J με βάση το πηλίκο Rayleigh.

Με βάση τους αλγόριθμους 3.3, 3.4 μπορούμε να κατασκευάσουμε μια υπορουτίνα (Από αυτό το σημείο θα ονομάζουμε όλους τους δευτερεύοντες αλγόριθμους ενδιάμεσων υπολογισμών των βασικών αλγορίθμων που μελετάμε και υλοποιούμε, υπορουτίνες), την οποία θα την χρησιμοποιήσουμε στις επόμενες αλγοριθμικές υλοποιήσεις ώστε αν γνωρίζουμε τις ιδιοτιμές ενός πίνακα να υπολογίζουμε τα ιδιοδιανύσματά του.

Υπορουτίνα 3.1

Υπολογισμός πίνακα ιδιοδιανυσμάτων
 Υλοποίηση στο MATLAB

```

function [Q]=givmeq(A,D)
n=length(A); I=eye(n); Q=[]; q2=zeros(1,n)';
m=D(1,1); v0=I(:,1); q=ray_quot(A,v0,m); Q=[Q q];
for i=2:n
    m=D(i,i); v0=I(:,i);
    q=ray_quot(A,v0,m);
    [ ,j]=ismember(q,Q,'rows');
    if j~ 0
        q2=inverse_it(A,m-0.001*m,v0);
        m=D(j,j); v0=I(:,j);
        q1=inverse_it(A,m-0.001*m,v0);
        Q(:,j)=q1; Q=[Q q2];
    else
        Q=[Q q];
    end
end

```

```

end
end
end

```

Στην υπορουτίνα 3.1 έχουμε ως ορίσματα έναν πίνακα A και τον διαγώνιο πίνακα των ιδιοτιμών του. Για κάθε ιδιοτιμή του A υπολογίζουμε το αντίστοιχο ιδιοδιάνυσμα με τον αλγόριθμο 3.4. Βέβαια αν δύο ιδιοτιμές του πίνακα A είναι αρκετά κοντά τότε είναι πιθανό ο αλγόριθμος 3.4 να αποτύχει να βρει τη σωστή ιδιοτιμή σε μία από τις δύο περιπτώσεις και να υπολογίσει το ίδιο ιδιοδιάνυσμα και για τις δύο ιδιοτιμές. Αυτό όμως δεν κινδυνεύουμε να συμβεί αν υπολογίσουμε με τον αλγόριθμο 3.3 τα αντίστοιχα ιδιοδιανύσματα των ιδιοτιμών. Έτσι στην υπορουτίνα 3.1 ελέγχουμε σε κάθε βήμα αν το ιδιοδιάνυσμα που υπολογίσαμε υπάρχει ήδη στον υπό κατασκευή πίνακα όλων των ιδιοδιανυσμάτων. Αν δεν υπάρχει τότε το προσθέτουμε στον πίνακα των ιδιοδιανυσμάτων. Αν υπάρχει ήδη σημαίνει ότι ο αλγόριθμος 3.4 απέτυχε να υπολογίσει το σωστό ιδιοδιάνυσμα για μία από τις δύο ιδιοτιμές και έτσι υπολογίζουμε με τον αλγόριθμο 3.3 τα δύο ιδιοδιανύσματα εκ νέου και τα τοποθετούμε στην κατάλληλη θέση-στήλη του πίνακα των ιδιοδιανυσμάτων. Η παραπάνω υπορουτίνα θα χρησιμοποιηθεί σε επόμενους αλγόριθμους αλλά έχει και τη δική της αυτοτελή αξία.

III Αλγόριθμος QR χωρίς εναλλαγές

Στο δεύτερο κεφάλαιο παρουσιάσαμε τον αλγόριθμο QR στην πιο απλή και σε μια πιο πρακτική μορφή καθώς και τον αλγόριθμο των ταυτόχρονων επαναλήψεων. Σε αυτή την ενότητα θα παρουσιάσουμε τις αντίστοιχες υλοποιήσεις στο MATLAB και θα γίνουμε λίγο πιο αναλυτικοί στην επιλογή της συνθήκης τερματισμού της επαναληπτικής διαδικασίας καθώς και στην τεχνική της αναγωγής σε ένα ισοδύναμο, απλούστερο πρόβλημα (deflation).

Η συνθήκη τερματισμού που επιλέξαμε και στους τρεις αλγόριθμους αναγκαστικά θα έπρεπε να ελέγχει το βαθμό σύγκλισης της επαναληπτικής διαδικασίας με βάση το $k-1$ και το k βήμα. Επομένως επιλέξαμε η συνθήκη να είναι

$$\max_{1 \leq i \leq m} (\text{abs}(\text{diag}(Q^{(k)T}Q^{(k-1)} - I))_i) \leq \epsilon_{\text{machine}} \quad (3.1)$$

Όπου I ο ταυτοτικός πίνακας.

Αφού οι ακολουθία των πινάκων Q συγκλίνει σε έναν πίνακα με στήλες τα ορθομοναδιαία ιδιοδιανύσματα του πίνακα A τότε είναι προφανές ότι η ακολουθία $Q^{(k)T}Q^{(k-1)}$ θα συγκλίνει στον ταυτοτικό πίνακα. Έτσι η συνθήκη (3.1) ελέγχει σε κάθε βήμα πόσο κοντά στο 1 είναι το μέγιστο κατά απόλυτη τιμή διαγώνιο στοιχείο του πίνακα $Q^{(k)T}Q^{(k-1)}$.

Επομένως η υλοποίηση της απλής μορφής του αλγόριθμου QR θα είναι η παρακάτω

Αλγόριθμος 3.5

Απλός αλγόριθμος QR

Υλοποίηση στο MATLAB

```

function [Qt,D]=QR_pure(A)
m=length(A);
if nargin==2
    m=length(A); [Q,R]=qr(A);
    Qt=Q; A=R*Q; K=1;
    while max(diag(K))>eps()
        K=Q; [Q,R]=qr(A); Qt=Qt*Q;
        A=R*Q; K=abs(Q'*K-eye(m));
    end
    D=diag(A); [D,in]=sort(D);
    D=diag(D); Qt=Qt(:,in);
else
    m=length(A); [Q,R]=qr(A);
    A=R*Q; K=1;
    while max(diag(K))>eps()
        K=Q; [Q,R]=qr(A);
        A=R*Q; K=abs(Q'*K-eye(m));
    end
    Qt=sort(diag(A));
end
end

```

Ο αλγόριθμος 3.5 μας δίνει τη δυνατότητα να υπολογίσουμε μόνο τις ιδιοτιμές του πίνακα A ή τις ιδιοτιμές και τα ιδιοδιανύσματα. Αυτό καθορίζεται από το πόσες μεταβλητές εξόδου ζητάμε. Όταν ζητάμε μία επιστρέφονται οι ιδιοτιμές σε ένα διάνυσμα με αύξουσα σειρά. Αν ζητάμε δύο επιστέφει στην πρώτη τον πίνακα των ιδιοδιανυσμάτων και στη δεύτερη τον διαγώνιο πίνακα των ιδιοτιμών σε αύξουσα σειρά. Βλέπουμε ότι και στους δύο βρόγχους ο κώδικας είναι ίδιος εκτός από τη διαφορά ότι στην περίπτωση που ζητάει και τα ιδιοδιανύσματα παράγεται ο πίνακας των ιδιοδιανυσμάτων της σχέσης 2.32.

IV Αλγόριθμος QR με εναλλαγές

Για την παρουσίαση της υλοποίησης του "πρακτικού" αλγόριθμου QR επιλέξαμε την ίδια συνθήκη τερματισμού με αυτή του αλγόριθμου (3.5).

Επίσης το ζήτημα των εναλλαγών το αναπτύξαμε επαρκώς στο δεύτερο κεφάλαιο. Όμως θα πρέπει να γίνουμε πιο αναλυτικοί όσο αφορά την τεχνική της αναγωγής σε ένα ισοδύναμο και απλούστερο πρόβλημα. Το παρακάτω λήμμα είναι πολύ βασικό.

Λήμμα IV.1. Έστω ο πίνακας $A \in \mathbb{C}^{m \times m}$ που έχει τη μορφή

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

Με $A_{11} \in \mathbb{C}^{p \times p}$, $A_{12} \in \mathbb{C}^{p \times q}$, $A_{22} \in \mathbb{C}^{q \times q}$, με $p + q = m$. Τότε ισχύει ότι $\lambda(A) = \lambda(A_{11}) \cup \lambda(A_{22})$.

Απόδειξη. Με βάση το λήμμα 7.1.2 στο [2].
Έστω ότι

$$Ax = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

όπου $x_1 \in \mathbb{C}^p$ και $x_2 \in \mathbb{C}^q$. Αν $x_2 \neq 0$ τότε $A_{22}x_2 = \lambda x_2$ και άρα $\lambda \in \lambda(A_{22})$. Αν $x_2 = 0$ τότε $A_{11}x_1 = \lambda x_1$ και άρα $\lambda \in \lambda(A_{11})$. Επομένως συνεπάγεται ότι $\lambda(A) \subset \lambda(A_{11}) \cup \lambda(A_{22})$ και αφού τα δύο σύνολα έχουν την ίδια πληθικότητα θα είναι ίσα. \square

Επομένως αν ο πίνακας A^k (υπενθυμίζουμε ότι είναι τριδιαγώνιος) του βήματος k είναι της μορφής

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad (3.2)$$

μπορούμε να υπολογίσουμε αντί για τις ιδιοτιμές του πίνακα A , τις ιδιοτιμές των πινακων A_1, A_2 . Όμως αυτό δεν συμβαίνει πολύ συχνά, καθώς σχεδόν ποτέ κάποιο μη διαγώνιο $A_{j,j+1}$ στοιχείο είναι ακριβώς μηδέν. Αυτό όμως που συμβαίνει αρκετά συχνά είναι κάποιο μη διαγώνιο στοιχείο να είναι αρκετά κοντά στο μηδέν. Έτσι το επόμενο θεώρημα των Bauer-Fike είναι αρκετά σημαντικό.

Θεώρημα IV.1. Έστω μ μία ιδιοτιμή του $A + E \in \mathbb{C}^{m \times m}$ και $X^{-1}AX = D = \text{diag}(\lambda_1, \dots, \lambda_m)$, τότε

$$\min_{\lambda \in \lambda(A)} |\lambda - \mu| \leq \kappa_p(X) \|E\|_p \quad (3.3)$$

Όπου $\|\cdot\|_p$ οποιαδήποτε p -νόρμα.

Απόδειξη. Με βάση το θεώρημα 7.2.2 στο [2].

Αν $\mu \in \lambda(A)$ τότε η απόδειξη είναι προφανής. Αλλιώς αν ο πίνακας $X^{-1}(A + E - \mu I)X$ είναι μη αντιστρέψιμος, τότε μη αντιστρέψιμος θα είναι και ο πίνακας $I + (D - \mu I)^{-1}(X^{-1}EX)$. Έτσι με βάση το λήμμα IV.2 θα ισχύει

$$1 \leq \|(D - \mu I)^{-1}(X^{-1}EX)\|_p \leq \|(D - \mu I)^{-1}\|_p \|X\|_p \|E\|_p \|X^{-1}\|_p$$

Αφού ο $(D - \mu I)^{-1}$ είναι διαγώνιος και η p-νόρμα κάθε διαγώνιου πίνακα είναι η απόλυτη τιμή του μέγιστου διαγώνιου στοιχείου, θα ισχύει

$$\|(D - \mu I)^{-1}\| = \max_{\lambda \in \lambda(A)} \frac{1}{|\lambda - \mu|}$$

Και η απόδειξη ολοκληρώθηκε. \square

Λήμμα IV.2. Αν $A \in \mathbb{R}^{m \times m}$ και $\|A\|_p < 1$, τότε ο πίνακας $I - A$ είναι αντιστρέψιμος και ισχύει

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$$

με

$$\|(I - A)^{-1}\|_p \leq \frac{1}{1 - \|A\|_p}$$

Περισσότερες λεπτομέρειες για το λήμμα IV.2 στο [2], λήμμα 2.3.3.

Η σχέση 3.3 του θεωρήματος IV.1 μας δείχνει ότι υπάρχει η περίπτωση οι ιδιοτιμές ενός πίνακα να είναι ευαίσθητες σε διαταραχές όταν ο πίνακας είναι κακής κατάστασης. Βέβαια όταν ένας πίνακας A είναι κανονικός και χρησιμοποιήσουμε τη 2-νόρμα τότε $\kappa(A) = \frac{|\lambda_{max}|}{|\lambda_{min}|}$ (αφού όπως είδαμε στο δεύτερο κεφάλαιο $\kappa(A) = \frac{\sigma_1}{\sigma_m}$). Επομένως στην περίπτωση μας, αφού ο πίνακας X είναι ορθομοναδιαίος, θα είναι και κανονικός, άρα θα είναι και καλής κατάστασης. Επίσης ο πίνακας E στην περίπτωση μας θα είναι ένας πίνακας με μηδενικά στοιχεία παντού, εκτός από δύο στοιχεία (ένα στην υπερδιαγώνιο και το συμμετρικό του) που θα είναι αρκετά κοντά στο μηδέν. Έτσι η νόρμα του E θα είναι πολύ μικρή, κάτι που σε συνδυασμό με την καλή κατάσταση του πίνακα X μας διασφαλίζει ότι οι αντίστοιχες ιδιοτιμές των πινάκων A , $A + E$ θα είναι πολύ κοντά και συνήθως με τόση ακρίβεια που δεν μας δημιουργούν κανένα πρόβλημα στους υπολογισμούς μας. Έτσι λοιπόν μπορούμε να παρουσιάσουμε την υλοποίηση του πρακτικού αλγόριθμου QR με χρήση της τεχνικής της αναγωγής σε ισοδύναμο απλούστερο πρόβλημα.

Αλγόριθμος 3.6

Πρακτικός αλγόριθμος QR
Υλοποίηση στο MATLAB

```
function [Q,D]=QR_prac(A,fake)
if nargin==1
    A=red_hessenberg(A);
end
```

```

n=length(A); t=true; m=A(n,n);
[Q, R]=qr(A-m*eye(n)); K=1;
A=R*Q+m*eye(n);
while max(diag(K))>eps()
    if length(A)>=10
        dd=diag(A,1); dd(1)=[];
        dd(length(dd))=[];
        if ~isempty(find(abs(dd)<=eps()))
            r=find(abs(dd)<=eps()+1);
            A1=A(1:r(1),1:r(1));
            A2=A(r(1)+1:n,r(1)+1:n);
            [Q1,D1]=QR_prac(A1,0);
            [Q2,D2]=QR_prac(A2,0);
            Q=blkdiag(Q1,Q2);
            D=blkdiag(D1,D2);
            t=false;
            break
        end
    end
    K=Q; m=A(n,n);
    [Q,R]=qr(A-m*eye(n));
    A=R*Q+m*eye(n); K=abs(Q'*K-eye(n));
end
if t
    D=diag(sort(diag(A)));
end
if nargin==1
    D=sort(diag(D)); D=diag(D);
    Q=givmeq(Z,D);
end
end

```

Βλέπουμε ότι ο παραπάνω αλγόριθμος στο MATLAB εμπεριέχει μια ενδεχόμενη αναδρομικότητα. Ελέγχει αν ο πίνακας A είναι διάστασης μεγαλύτερης ή ίσης του 10 προκειμένου να ελέγξει αν μπορούμε να αναγάγουμε το πρόβλημά μας σε ένα ισοδύναμο και απλούστερο πρόβλημα. Στη συνέχεια αν υπάρχει κάποιο στοιχείο της υπερδιαγωνίου που είναι αρκετά κοντά στο μηδέν, σπάμε το πρόβλημα μας σε δύο προβλήματα εύρεσης ιδιοτιμών και καλούμε τον ίδιο αλγόριθμο 3.6 για τα δύο νέα προβλήματα και στη συνέχεια συνθέτουμε τους πίνακες Q, D με βάση τις λύσεις των απλούστερων προβλημάτων.

Η τελευταία συνθήκη [if t ...] ελέγχει αν έχει γίνει αναγωγή σε απλούστερο πρόβλημα ώστε να υπολογίσει τον πίνακα D μόνο στην περίπτωση της μη αναγωγής. Αν θέλουμε να αποφύγουμε την αναγωγή σε ισοδύναμο και απλούστερο πρόβλημα, τότε αρκεί να αφαιρέσουμε από τον κώδικά μας τους δύο ελέγχους [if ...] μέσα στο βρόγχο while.

Επίσης η μεταβλητή "fake" μας βοηθάει να καταλάβουμε αν ο αλγόριθμος έχει καλεστεί για πρώτη φορά ώστε να εκτελέσουμε το μετασχηματισμό Hessenberg, καθώς αν τον εκτελούσαμε σε κάθε αναδρομική κλήση του αλγορίθμου 3.6 θα έχουμε δύο προβλήματα. Το ένα είναι ότι θα εκτελέσουμε αχρείαστες πράξεις και το άλλο ότι θα έχουμε πρόβλημα στον αριθμητικό υπολογισμό του μετασχηματισμού Hessenberg καθώς ο πίνακας A^k τείνει σε ένα διαγώνιο πίνακα και άρα τα στοιχεία της υπερδιαγωνίου και της υποδιαγωνίου τείνουν στο μηδέν. Επομένως στον υπολογισμό των Householder πινάκων θα πρέπει να διαιρέσουμε με αριθμούς που είναι πολύ κοντά στο μηδέν, οπότε ο κίνδυνος της υπερχειλίσης ή της υποχειλίσης είναι αρκετά μεγάλος.

Τέλος βλέπουμε ότι ο αλγόριθμος 3.6 μας επιστρέφει πάντα και τον πίνακα ιδιοδιανυσμάτων εκτός από τον διαγώνιο πίνακα των ιδιοτιμών.

Στον αλγόριθμο 3.6 όπως βλέπουμε επιλέγουμε σε κάθε βήμα την εναλλαγή πηλίκου Rayleigh όπως δείξαμε και στη σχέση 2.40. Αυτή η εναλλαγή επιτυγχάνει γρήγορη σύγκλιση στην τελευταία στήλη του πίνακα Q^k όμως έχει κάποια προβλήματα όπως δείξαμε στο δεύτερο κεφάλαιο. Έτσι εκτός της εναλλαγής πηλίκου Rayleigh υλοποιήσαμε και τον "πρακτικό" αλγόριθμο QR με την εναλλαγή Wilkinson. Η υλοποίηση αυτού του αλγορίθμου φαίνεται παρακάτω

Αλγόριθμος 3.7

Πρακτικός αλγόριθμος QR με εναλλαγή Wilkinson

Υλοποίηση στο MATLAB

```
function [D]=QR_wilkinson(A,fake)
if nargin==1
    A=red_hessenberg(A);
end
n=length(A); t=true;
B=A(n-1:n,n-1:n); d=(B(1)-B(4))/2;
m=B(4)-(sign(d)*B(2)*2)/(abs(d)+sqrt(d*2+B(2)*2));
[Q,R]=qr(A-m*eye(n)); K=1;
A=R*Q+m*eye(n);
while max(diag(K))>eps()
    if length(A)>=10
        dd=diag(A,1); dd(1)=[];
        dd(length(dd))=[];
        if ~isempty(find(abs(dd)<=eps()))
            r=find(abs(dd)<=eps()+1);
            A1=A(1:r(1),1:r(1));
            A2=A(r(1)+1:n,r(1)+1:n);
```

```

        D1=QR_wilkinson(A1,0);
        D2=QR_wilkinson(A2,0);
        Q=blkdiag(Q1,Q2);
        D=blkdiag(D1,D2);
        t=false;
        break
    end
end
K=Q; B=A(n-1:n,n-1:n);
d=(B(1)-B(4))/2;
m=B(4)-(sign(d)*B(2)^2)/(abs(d)+sqrt(d^2+B(2)^2));
[Q,R]=qr(A-m*eye(n)); A=R*Q+m*eye(n);
K=abs(Q*K-eye(n));
end
if t
    D=diag(sort(diag(A)));
end
if nargin==1
    D=sort(diag(D)); D=diag(D);
    Q=givmeq(Z,D);
end
end

```

Βλέπουμε, όπως αναμενόταν, ότι δεν υπάρχει καμία ουσιαστική αλλαγή από τον αλγόριθμο 3.6 καθώς αλλάξαμε μόνο τις εντολές που έχουν να κάνουν με τον υπολογισμό της εναλλαγής.

Στους αλγόριθμους 3.6 και 3.7 ελέγχουμε σε κάθε βήμα αν υπάρχει κάποιο στοιχείο της υπερδιαγωνίου που είναι αρκετά κοντά στο μηδέν ώστε να θεωρήσουμε δύο απλούστερα προβλήματα με αποτέλεσμα να σπαταλάμε αρκετό υπολογιστικό χρόνο για την εύρεση αυτών των στοιχείων του πίνακα. Επίσης δεν εκμεταλλευόμαστε τη βασική ιδιότητα του αλγόριθμου QR με εναλλαγές που είναι ότι η τελευταία στήλη του $Q^{(k)}$ συγκλίνει γρήγορα σε ένα ιδιοδιάνυσμα του A . Έτσι αρκεί να ελέγξουμε πότε το $A^{(k)}(m, m-1)$ είναι αρκετά κοντά στο μηδέν. Τότε ο $A^{(k)}$ θα είναι της μορφής $A^{(k)} = \begin{bmatrix} A_1 & 0 \\ 0 & \hat{\lambda}_j \end{bmatrix}$ και μπορούμε να θεωρήσουμε το υποπρόβλημα A_1 . Έτσι θα υπολογίσουμε τις ιδιοτιμές του A και με τη βοήθεια της υπορουτίνας 3.1 θα υπολογίσουμε τα ιδιοδιανύσματα.

Αλγόριθμος 3.8

Πρακτικός αλγόριθμος QR με εναλλαγή Wilkinson 2
Υλοποίηση στο MATLAB

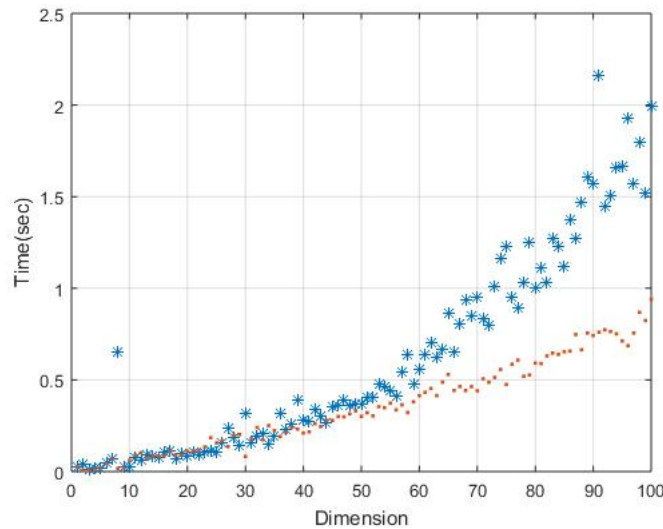
```

function [Q,D]=QR_wil(A,z)
n=length(A);
if nargin==1
    A2=A;
    nrm=norm(A);
    A=red_hes(A);
else
    nrm=z;
end
B=A(n-1:n,n-1:n); d=(B(1)-B(4))/2;
m=B(4)-(sign(d)*B(2)^2)/(abs(d)+sqrt(d^2+B(2)^2));
[Q,R]=qr(A-m*eye(n));
A=R*Q+m*eye(n); K=1;
if n>3
    while true
        B=A(n-1:n,n-1:n); d=(B(1)-B(4))/2;
        m=B(4)-(sign(d)*B(2)^2)/(abs(d)+sqrt(d^2+B(2)^2));
        [Q,R]=qr(A-m*eye(n)); A=R*Q+m*eye(n);
        if abs(A(n,n-1))<=eps()*nrm && n>=4
            D=QR_wil(A(1:n-1,1:n-1),nrm);
            D=blkdiag(D,A(n,n));
            D=diag(sort(diag(D)));
            break
        end
    end
else
    while max(diag(K))>eps()
        K=Q;
        B=A(n-1:n,n-1:n);
        d=(B(1)-B(4))/2;
        m=B(4)-(sign(d)*B(2)^2)/(abs(d)+sqrt(d^2+B(2)^2));
        [Q,R]=qr(A-m*eye(n));
        A=R*Q+m*eye(n);
        K=abs(Q'*K-eye(n));
    end
    D=diag(sort(diag(A)));
end
if nargin==1 && nargout==2
    Q=givmeq(A2,D);
else
    Q=D;
end
end

```

Στον αλγόριθμο 3.8 θεωρούμε ότι το στοιχείο $A(m, m-1)$ είναι αρκετά κοντά στο μηδέν όταν είναι μικρότερο ή ίσο με $\epsilon_{machine}\|A\|$ (λόγω του θεωρήματος

IV.1 στο πρώτο κεφάλαιο) και τότε λύνουμε το υποπρόβλημα $A^{(k)}(1:m-1, 1:m-1)$. Επίσης μας δίνει τη δυνατότητα να υπολογίσουμε μόνο τις ιδιοτιμές του και όχι τα ιδιοδιανύσματά του. Ποιος αλγόριθμος είναι πιο γρήγορος; Ο 3.7 ή ο 3.8; Βλέπουμε ξεκάθαρα ότι ο αλγόριθμος 3.8 για



Σχήμα 3.2: Οι χρόνοι εκτέλεσης για 100 πίνακες με στοιχεία από την κανονική κατανομή, διαστάσεων 4×4 έως και 103×103 , των αλγόριθμων 3.7 και 3.8. Με `**` είναι οι χρόνοι του 3.7 και με `.` οι χρόνοι του 3.8.

πίνακες διαστάσεων 50×50 και μεγαλύτερους υπερέρχει σημαντικά.

Στη συνέχεια θα υλοποιήσουμε τον αλγόριθμο QR με εναλλαγές Wilkinson χωρίς αναγωγές σε απλούστερα υποπροβλήματα με μία ακόμη διαφορά σε σχέση με τους προηγούμενους αλγόριθμους QR που έχει να κάνει με τον τρόπο υπολογισμού του πίνακα των ιδιοδιανυσμάτων. Σε αυτή την περίπτωση θα κατασκευάσουμε τον πίνακα της σχέσης 2.39. Επειδή όμως ο πρακτικός αλγόριθμος QR τριδιαγωνιοποιεί το πρόβλημά μας πριν το επιλύσει για τον υπολογισμό του πίνακα των ιδιοδιανυσμάτων θα χρειαστούμε και τους από τα δεξιά πίνακες Householder του μετασχηματισμού Hessenberg που μας δίνει ο αλγόριθμος 3.1. Ο ζητούμενος πίνακας θα είναι ο $P_1 P_2 \cdots P_{m-2} Q_1 Q_2 \cdots Q_i$, όπου P_j οι από τα δεξιά πίνακες Householder του μετασχηματισμού Hessenberg και Q_i οι πίνακες που παράγει σε κάθε βήμα ο αλγόριθμος 2.6, χωρίς την αναγωγή.

Αλγόριθμος 3.9

Πρακτικός αλγόριθμος QR με εναλλαγή Wilkinson 2

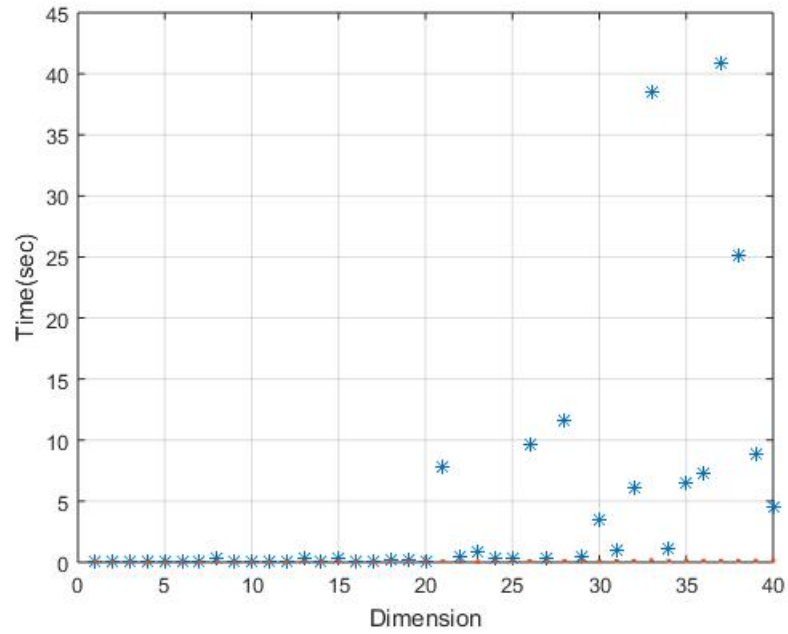
Υλοποίηση στο MATLAB

```
function [Qt,D]=QR_wil_2(A,z)
n=length(A);
if nargin==2
    Qt=eye(n);
else
    [A,Qt]=red_hessenberg(A);
end
K=1; B=A(n-1:n,n-1:n); d=(B(1)-B(4))/2;
m=B(4)-(sign(d)*B(2)^2)/(abs(d)+sqrt(d^2+B(2)^2));
[Q,R]=qr(A-m*eye(n)); Qt=Qt*Q;
A=R*Q+m*eye(n);
while max(diag(K))>eps()
    K=Q; B=A(n-1:n,n-1:n); d=(B(1)-B(4))/2;
    m=B(4)-(sign(d)*B(2)^2)/(abs(d)+sqrt(d^2+B(2)^2));
    [Q,R]=qr(A-m*eye(n)); Qt=Qt*Q;
    A=R*Q+m*eye(n); K=abs(Q'*K-eye(n));
end
D=diag(A); [D,indx]=sort(D);
D=diag(D); Qt=Qt(:,indx);
end
```

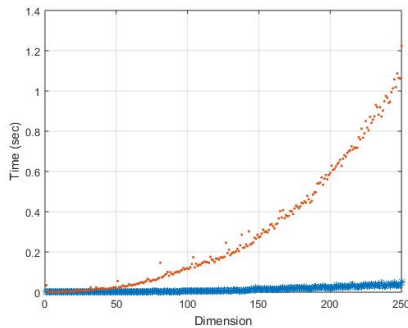
Στο σχήμα 3.3 συγκρίνουμε τους αλγόριθμους 3.8 και 3.9. Βλέπουμε ότι ο αλγόριθμος 3.8 υπερέρχει σημαντικά του 3.9 όχι όμως με τον τρόπο που υπερτερούσε του 3.7 αλλά με έναν τυχαίο σε σχέση με τη διάσταση του πίνακα τρόπο. Αυτό έχει να κάνει με το γεγονός ότι ο αλγόριθμος 3.9 δεν ανάγει το πρόβλημά μας σε υποπροβλήματα και έτσι κάθε στήλη του πίνακα Q συγκλίνει με βάσει τις εναλλαγές που από κάποιο βήμα και μετά (αρκετά μικρό) είναι ίδιες με την ιδιοτιμή που εμφανίζεται σαν το κάτω δεξιά στοιχείο του $A^{(k)}$. Έτσι τυχαίνει σε κάποιες περιπτώσεις αυτή η εναλλαγή να μην επιτυγχάνει γρήγορη σύγκλιση.

Στη συνέχεια θα συγκρίνουμε τον αλγόριθμο 3.8 με τον αλγόριθμο του MATLAB που αντιστοιχεί στην εντολή $eig()$ για προβλήματα που υπολογίζουμε μόνο ιδιοτιμές και όχι ιδιοδιανύσματα καθώς ο αλγόριθμος 3.8 είναι προς το παρόν ο ταχύτερος αλγόριθμος για τον υπολογισμό ιδιοτιμών που έχουμε υλοποιήσει.

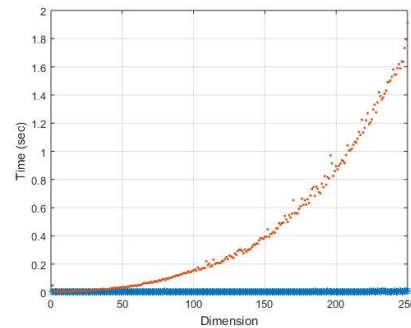
Στο σχήμα 3.4 υπολογίζουμε τις ιδιοτιμές τριδιαγώνιων πινάκων και έτσι δεν έχουμε το κόστος των πράξεων του μετασχηματισμού Hessenberg ενώ στο σχήμα 3.5 παρατηρούμε ότι το κόστος του μετασχηματισμού Hessenberg επιβαρύνει την εκτέλεση του αλγόριθμου 3.8. Είναι φανερό πως ο αλγόριθμος του MATLAB υπερέρχει του αλγόριθμου 3.8 αν και ειδικά στην πρώτη περίπτωση η διαφορά δεν ξεπερνά το ένα δευτερόλεπτο ακόμα και για πίνακες 250×250 .



Σχήμα 3.3: Οι χρόνοι εκτέλεσης για 37 πίνακες με στοιχεία από την κανονική κατανομή, διαστάσεων 4×4 έως και 40×40 , των αλγόριθμων 3.8 και 3.9. Με '*' είναι οι χρόνοι του 3.9 και με '.' οι χρόνοι του 3.8.



Σχήμα 3.4: Οι χρόνοι εκτέλεσης για 247 τυχαίους τριδιαγώνιους πίνακες, διαστάσεων 4×4 έως και 250×250 , των αλγόριθμων 3.8 και του αλγόριθμου του MATLAB που αντιστοιχεί στην εντολή `eig()`. Με '*' είναι οι χρόνοι του MATLAB και με '.' οι χρόνοι του 3.8.



Σχήμα 3.5: Οι χρόνοι εκτέλεσης για 247 πίνακες με στοιχεία από την κανονική κατανομή, διαστάσεων 4×4 έως και 250×250 , των αλγόριθμων 3.8 και του αλγόριθμου του MATLAB που αντιστοιχεί στην εντολή `eig()`. Με '*' είναι οι χρόνοι του MATLAB και με '.' οι χρόνοι του 3.8.

V Άλλοι αλγόριθμοι

Σε αυτή την ενότητα θα παρουσιάσουμε τις υλοποιήσεις των αλγορίθμων που αναπτύξαμε στο δεύτερο κεφάλαιο στην αντίστοιχη ενότητα. Δηλαδή του αλγόριθμου Jacobi, του αλγόριθμου της διχοτόμησης και του αλγόριθμου διαίρει και βασίλευε. Όπως και στις προηγούμενες ενότητες έτσι και σε αυτή θα πρέπει να γίνουμε πιο αναλυτικοί στα ζητήματα που δεν έχουμε αναπτύξει και είναι αναγκαία για την πρακτική υλοποίηση των αλγορίθμων.

V.I Αλγόριθμος Jacobi

Τον αλγόριθμο Jacobi τον παρουσιάσαμε αρκετά αναλυτικά στο δεύτερο κεφάλαιο. Οι πίνακες που πρέπει να κατασκευάσουμε σε κάθε βήμα περιγράφονται από τις σχέσεις 2.50, 2.51. Προκειμένου τα στοιχεία τις 2.52 να είναι μηδενικά πρέπει να ικανοποιείται η 2.53. Όμως για να κατασκευάσουμε τους πίνακες της σχέσης 2.50 πρέπει να υπολογίσουμε τα $\cos\theta$, $\sin\theta$, τα οποία θα υπολογίσουμε με τη βοήθεια των τριγωνομετρικών σχέσεων

$$\cos^2\theta = \frac{1}{1 + \tan^2\theta}, \quad \sin^2\theta + \cos^2\theta = 1$$

προκύπτουν οι σχέσεις

$$\cos 2\theta = \frac{1}{\sqrt{1 + \tan^2 2\theta}}, \quad \sin 2\theta = \frac{\tan 2\theta}{\sqrt{1 + \tan^2 2\theta}}$$

έτσι με τη βοήθεια των σχέσεων $\cos 2\theta = 2\cos^2\theta - 1$ και $\sin 2\theta = 2\sin\theta\cos\theta$ καταλήγουμε στις σχέσεις

$$\cos\theta = \sqrt{\frac{1}{2} + \frac{1}{2\sqrt{1 + \tan^2 2\theta}}}, \quad \sin\theta = \frac{\tan 2\theta}{2\cos\theta\sqrt{1 + \tan^2 2\theta}} \quad (3.4)$$

Επομένως η σχέση 3.4 μας επιτρέπει να υπολογίσουμε κατευθείαν τα $\cos\theta$, $\sin\theta$ χωρίς να χρειαστεί να υπολογίσουμε το θ . Επομένως πλέον μπορούμε να παρουσιάσουμε την υλοποίηση του αλγόριθμου Jacobi.

Αλγόριθμος 3.10

Αλγόριθμος Jacobi

Υλοποίηση στο MATLAB

```
function [eigv]=Jacobi_eig(q)
m=length(q);
q2=triu(q,1);
q2=q2.^2;
sum_jac=sum(sum(q2));
while sum_jac>=eps()
    for i=1:m-1
        for j=i+1:m
            B=(2*q(i,j))/(q(i,i)-q(j,j));
            costh=sqrt(0.5*(1+1/sqrt(1+B^2)));
            sinh=B/(2*costh*sqrt(1+B^2));
            R=eye(m);
            R(i,i)=costh;
            R(j,j)=R(i,i);
            R(j,i)=sinh;
            R(i,j)=-R(j,i);
            q=(R')*q*R;
        end
    end
    q2=triu(q,1);
    q2=q2.^2;
    sum_jac=sum(sum(q2));
end
eigv=sort(diag(q));
end
```

Στον αλγόριθμο 3.10 επιλέξαμε να μηδενίζουμε με τη σειρά: $\alpha_{12}, \alpha_{13}, \dots, \alpha_{1m}, \alpha_{23}, \dots, \alpha_{2m}, \dots, \alpha_{m-1,m}$ τα στοιχεία του πίνακα $A^{(k)}$. Θα μπορούσαμε,

όπως αναφέραμε και στο δεύτερο κεφάλαιο, σε κάθε βήμα να επιλέγουμε το μεγαλύτερο κατά απόλυτη τιμή μη διαγώνιο στοιχείο του πίνακα $A^{(k)}$ και να κατασκευάζουμε τον πίνακα που ορίζεται από τη σχέση 2.50. Έτσι για να υλοποιήσουμε τον αλγόριθμο Jacobi με αυτή την παραλλαγή αρκεί να τροποποιήσουμε κατάλληλα τον αλγόριθμο 3.10 όπως φαίνεται παρακάτω

Αλγόριθμος 3.11

Αλγόριθμος Jacobi (παραλλαγή)

Υλοποίηση στο MATLAB

```
function [eigv]=Jacobi_eig(q)
m=length(q);
q2=triu(q,1);
q4=q2.^2;
sum_jac=sum(sum(q4));
while sum_jac>=eps()
    q3=abs(q2);
    i=mod(find(q3==max(q3(:))),m);
    j=floor(find(q3==max(q3(:)))/m)+1;
    :
    q2=sparse(triu(q)-diag(diag(q)));
    q4=q2.^2;
    sum_jac=sum(sum(q4));
end
eigv=sort(diag(q));
end
```

Με τις εντολές

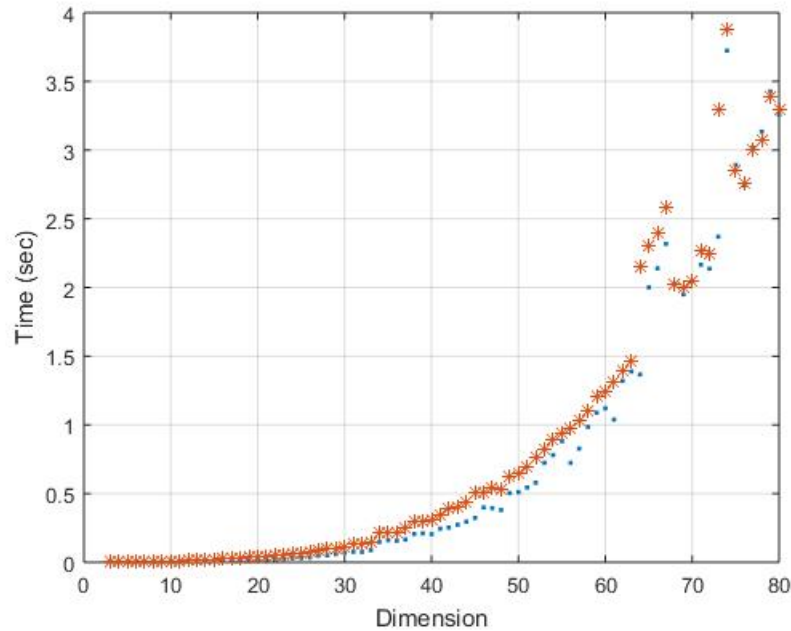
```
:
i=mod(find(q3==max(q3(:))),m);
j=floor(find(q3==max(q3(:)))/m)+1;
:
```

παίρνουμε τη σειρά και τη στήλη του μεγαλύτερου κατά απόλυτη τιμή μη διαγώνιου στοιχείου σε κάθε βήμα, ώστε να κατασκευάσουμε τον πίνακα της σχέσης 2.50. Αν προσθέσουμε και στους δύο αλγόριθμους τις εντολές

```
tic
:
toc
```

μπορούμε να μετρήσουμε το συνολικό χρόνο που απαιτείται για την εκτέλεση των αλγορίθμων για κάποιο συμμετρικό πίνακα και έτσι να έχουμε μια

πρώτη απλή σύγκριση των δύο αλγορίθμων. Το σχήμα 3.6 είναι χαρακτηρι-



Σχήμα 3.6: Οι χρόνοι που απαιτούνται για την εκτέλεση των αλγορίθμων 3.10, 3.11 για συμμετρικούς πίνακες διάστασης 4×4 έως και 83×83 . Με "." σημειώνουμε τους χρόνους εκτέλεσης του αλγόριθμου 3.10 και με "*" τους αντίστοιχους χρόνους εκτέλεσης του αλγόριθμου 3.11.

στικό. Βλέπουμε ότι ο αλγόριθμος 3.10 έχει ένα πλεονέκτημα σε σχέση με τον αλγόριθμο 3.11 με τη διαφορά να γίνεται αισθητή για πίνακα διάστασης 40×40 και μεγαλύτερης. Αυτή η διαφορά στην ταχύτητα έχει να κάνει με τον υπολογιστικό χρόνο που χρειάζεται ο αλγόριθμος 3.11 για την αναζήτηση του μέγιστου κατ' απόλυτη τιμή μη διαγώνιου στοιχείου σε κάθε βήμα. Επομένως μπορούμε να πούμε ότι προτιμότερος σε αρκετά μεγάλο βαθμό είναι ο αλγόριθμος 3.10 σε σχέση με τον αλγόριθμο 3.11.

V.II Αλγόριθμος διχοτόμησης

Στην αντίστοιχη ενότητα στο δεύτερο κεφάλαιο παρουσιάσαμε όλο το θεωρητικό υπόβαθρο που είναι αναγκαίο για την υλοποίηση του αλγορίθμου της διχοτόμησης. Οι κρίσιμες ιδιότητες είναι η παρεμβολή των ιδιοτιμών του πίνακα $A^{(k)}$ σε αυτές του $A^{(k)}$ που περιγράφεται από τη σχέση 2.55 και το σχήμα 2.2, η ακολουθία Sturm της 2.56 και οι αναδρομικές σχέσεις 2.57, 2.58. Μια πολύ σημαντική και ξεχωριστή (σε σχέση με άλλους αλγόριθμους) δυνατότητα που μας δίνει ο συγκεκριμένος αλγόριθμος είναι η εύρεση του πλήθους των ιδιοτιμών σε διαστήματα της μορφής $(-\infty, \alpha)$, (α, β) , $(\beta, +\infty)$, η οποία μας διευκολύνει στην εύρεση των ιδιοτιμών σε αυτόν αλλά και σε άλλους αλγόριθμους.

Αρχικά θα παρουσιάσουμε μια απλή υλοποίηση του αλγορίθμου η οποία δεν εκμεταλλεύεται όλες τις παραπάνω ιδιότητες και απαιτεί περισσότερο χρόνο εκτέλεσης από τη δεύτερη υλοποίηση που θα παρουσιάσουμε. Θα χρησιμοποιήσουμε μόνο την ιδιότητα της παρεμβολής των ιδιοτιμών της 2.55 και του σχήματος 2.2 και την αναδρομική σχέση 2.58 των χαρακτηριστικών πολυωνύμων. Η ιδέα είναι να υπολογίσουμε όλες τις ιδιοτιμές των άνω-αριστερά τετραγωνικών υποπινάκων $A^{(2)}$, $A^{(3)}$, ..., $A^{(m)}$, βρίσκοντας τις ρίζες των αντίστοιχων χαρακτηριστικών πολυωνύμων στα διαστήματα που ορίζουν η 2.55 και το σχήμα 2.2. Για παράδειγμα στο k -οστό βήμα θα υπολογίσουμε τις ιδιοτιμές του πίνακα $A^{(k)}$ εκτελώντας τη μέθοδο διχοτόμησης στα διαστήματα $(\lambda_j^{(k-1)}, \lambda_{j+1}^{(k-1)})$ και στα $(\alpha, \lambda_1^{(k-1)})$, $(\lambda_m^{(k-1)}, \beta)$ με κατάλληλα επιλεγμένα α, β . Αρχικά παρουσιάζουμε τον αναδρομικό αλγόριθμο που υπολογίζει τα χαρακτηριστικά πολυώνυμα από την 2.58, τον οποίο υλοποιήσαμε με τη συνάρτηση `comp_pol()` στο MATLAB.

Υπορουτίνα 3.2

Υπολογισμός χαρακτηριστικών πολυωνύμων της σχέσης (2.58)
Υλοποίηση στο MATLAB

```
function [pv]=comp_pol(a,k,deriv)
p=val_coef_p(k);
if nargin==3
    for i=1:deriv
        p=polyder(p);
    end
end
pv=@comp_par;
function [c]=val_coef_p(n)
    if n==1
        c=[-1 a(1,1)];
    elseif n==0
```

```

        c=[1];
    else
        c=conv([-1 a(n,n)],val_coef_p(a,n-1))-[0 0 a(n,n-1)^2*val_coef_p(a,n-
2)];
    end
end
function [y]=comp_par(x)
    y=polyval(p,x);
end
end

```

Η παραπάνω συνάρτηση στο MATLAB δέχεται σαν ορίσματα ένα τριδιαγώνιο πίνακα και έναν φυσικό αριθμό μικρότερο ή ίσο με τη διάσταση του πίνακα και επιστρέφει το αντίστοιχο χαρακτηριστικό πολυώνυμο της 2.58. Επίσης μας δίνει τη δυνατότητα να δώσουμε και τρίτο όρισμα ένα φυσικό αριθμό μεγαλύτερο ή ίσο του ένα δηλώνοντας ποια παράγωγο του χαρακτηριστικού πολυωνύμου θέλουμε να πάρουμε σαν αποτέλεσμα. Για την ακρίβεια επιστρέφει τη μεταβλητή *pv* που καθορίζεται (handle) από την ένθετη υποσυνάρτηση (nested function) *comp_par()*. Η τελευταία υπολογίζει την τιμή του πολυωνύμου για το όρισμα *x*, με τη βοήθεια των συντελεστών των πολυωνύμων *p* που υπολογίζονται από την κύρια συνάρτηση και της συνάρτησης του MATLAB, *polyval()*. Η υποσυνάρτηση *val_coef_p()* υλοποιεί τον αναδρομικό αλγόριθμο που προκύπτει από τη σχέση 2.58, παίρνοντας σαν ορίσματα τα ορίσματα της κύριας συνάρτησης και μας επιστρέφει τους συντελεστές του αντίστοιχου πολυωνύμου σε ένα διάνυσμα. Η εντολή *conv()* παίρνει σαν ορίσματα τους συντελεστές, σε διανύσματα, δύο πολυωνύμων που πολλαπλασιάζονται και επιστρέφει τους συντελεστές του πολυωνύμου που προκύπτει σε ένα διάνυσμα. Τέλος η εντολή *polyder()* παίρνει σαν όρισμα τους συντελεστές ενός πολυωνύμου και επιστρέφει τους συντελεστές της πρώτης παραγώγου του.

Μία απλοϊκή υλοποίηση, λοιπόν, του αλγόριθμου της διχοτόμησης είναι η παρακάτω.

Αλγόριθμος 3.12

Αλγόριθμος διχοτόμησης (απλός)
Υλοποίηση στο MATLAB

```

function [eigval]=simple_bis_eig(q)
a=red_hessenberg(q);
m=length(a); poly={};
for i=1:m
    poly{i}=comp_pol(a,i);
end

```



```

p=[1 -(a(1,1)+a(2,2)) (a(1,1)*a(2,2)-a(1,2)^2)];
r=sort(roots(p));
for i=3:m
    d=r; r=[]; p=poly{i};
    for j=1:length(d)-1
        l=bis_root(p,d(j),d(j+1));
        r=[r l];
    end
    con=length(r);
    a1=d(1); a2=d(2);
    at=d(length(d)); apt=d(length(d)-1);
    k1=true; k2=true;
    while(con>length(r)-2)
        if k1
            d=abs(a2-a1);
            a2=a1; a1=a1-d;
            if p(a1)*p(a2)<=0
                rt1=bis_root(p,a1,a2); r=[rt1 r];
                k1=false;
            end
        end
        if k2
            d=abs(at-apt);
            apt=at; at=at+d;
            if p(apt)*p(at)<=0
                rt2=bis_root(p,apt,at);
                r=[r rt2]; k2=false;
            end
        end
    end
end
eigval=r';
end

```

Σε κάθε βήμα του βρόγχου

```

for i=3:m
:
end

```

υπολογίζουμε τις ρίζες των πολυωνύμων $\pi^{(i)}(x)$ της (2.58). Αναλυτικότερα στον βρόγχο

```

for j=1:length(d)-1
:
end

```

υπολογίζουμε τις ρίζες του χαρακτηριστικού πολυωνύμου $p^i(x)$ στα διαστήματα $(\lambda_j^{(i-1)}, \lambda_{j+1}^{(i-1)})$. Στο βρόγχο

```
while
:
end
```

υπολογίζουμε τις ιδιοτιμές $\lambda_1^{(i)}, \lambda_i^{(i)}$. Για τον υπολογισμό της ιδιοτιμής $\lambda_1^{(i)}$ θεωρώ το διάστημα $(\lambda_1^{(i-1)} - d, \lambda_1^{(i-1)})$, όπου $d = \lambda_2^{(i-1)} - \lambda_1^{(i-1)}$. Αν η ρίζα δεν υπάρχει σε αυτό το διάστημα τότε θεωρούμε το $(\lambda_1^{(i-1)} - 2d, \lambda_1^{(i-1)} - d)$ και επαναλαμβάνουμε αυτή τη διαδικασία μέχρι να βρεθεί η ζητούμενη ρίζα. Την ίδια διαδικασία κάνουμε και για τον υπολογισμό της $\lambda_i^{(i)}$. Τις ρίζες υπολογίζουμε με τη μέθοδο της διχοτόμησης, την οποία υλοποιήσαμε με τον παρακάτω αναδρομικό αλγόριθμο.

Υπορουτίνα 3.3

Μέθοδος διχοτόμησης
Υλοποίηση στο MATLAB

```
function [sol]=bis_root(p,a,b)
c=(a+b)/2;
sol=c;
if abs(b-a)>eps()
    if p(a)*p(c)<0
        sol=bis_root(p,a,c);
    elseif p(b)*p(c)<0
        sol=bis_root(p,c,b);
    else
        d=[a b c];
        [~,sol]=min([abs(p(a)) abs(p(b)) abs(p(c))]);
        sol=d(sol);
    end
end
end
```

Ο αλγόριθμος 3.12 είναι βέβαια αρκετά απλοϊκός και δεν χρησιμοποιεί όλες τις ιδιότητες των τριδιαγώνιων πινάκων που αναφέραμε στο δεύτερο κεφάλαιο. Μία βασική ιδιότητα που πρέπει να εκμεταλλευθούμε είναι η ακολουθία Stourm 2.56 και η αναδρομική σχέση 2.57. Με βάση αυτές τις δύο μπορούμε να μετρήσουμε σχετικά εύκολα πόσες αρνητικές ιδιοτιμές έχει ένας τριδιαγώνιος πίνακας, άρα και ένας συμμετρικός. Αυτή την υπολογιστική διαδικασία μέτρησης των αρνητικών ιδιοτιμών ενός τριδιαγώνιου πίνακα μέσω των 2.56, 2.57 υλοποιήσαμε με την παρακάτω υπορουτίνα.

Υπορουτίνα 3.4

Μέτρηση πλήθους αρνητικών ιδιοτιμών τριδιαγώνιου πίνακα
Υλοποίηση στο MATLAB

```
function [c]=detcomp_rec(a)
m=length(a); c=0;
for i=0:m-1
    value1=det_value(i); value2=det_value(i+1);
    if value1*value2<0
        c=c+1;
    elseif value1*value2==0
        if value1==0 && value2~=0
            c=c+1;
        end
    end
end
end
function [val]=det_value(k)
if k==0
    val=1;
elseif k==1
    val=a(1,1);
elseif k==2
    val=det(a(1:2,1:2));
else
    val=a(k,k)*det_value(k-1)-a(k,k-1)^2*det_value(k-2);
end
end
end
```

Στην υπορουτίνα 3.4 έχουμε σαν όρισμα έναν τριδιαγώνιο πίνακα και σαν αποτέλεσμα ένα φυσικό αριθμό που αντιστοιχεί στο πλήθος των αρνητικών ιδιοτιμών του. Η ένθετη υποσυνάρτηση *det_value()* υλοποιεί την αναδρομική σχέση 2.57. Η κυρίως συνάρτηση υλοποιεί την ακολουθία Sturm και μετράει τις αλλαγές των προσήμων της.

Στο δεύτερο κεφάλαιο αναφέραμε ότι μπορούμε να βρούμε πόσες ιδιοτιμές έχει ένας τριδιαγώνιος πίνακας στα διαστήματα της μορφής $(-\infty, \alpha)$, (α, β) , $[\alpha, +\infty)$. Αν από ένα πίνακα A με ιδιοδιανύσματα λ_j αφαιρέσουμε το αI , όπου $\alpha \in \mathbb{R}$, θα προκύψει ο πίνακας $A - \alpha I$, του οποίου τα ιδιοδιανύσματα θα είναι τα $(\lambda_j - \alpha)^{-1}$. Έτσι τα αρνητικά ιδιοδιανύσματα του πίνακα $A - \alpha I$ θα είναι τα ιδιοδιανύσματα του A με $\lambda_j < \alpha$. Επομένως αν βρούμε το πλήθος των αρνητικών ιδιοτιμών του πίνακα $A - \alpha I$ μέσω της υπορουτίνας 3.4 θα έχουμε βρει το πλήθος των ιδιοτιμών του πίνακα A που ανήκουν στο διάστημα $(-\infty, \alpha)$. Για να βρούμε το πλήθος των ιδιοτιμών του πίνακα A

που ανήκουν στο διάστημα (α, β) βρίσκουμε το πλήθος των ιδιοτιμών στο $(-\infty, \beta)$ και αφαιρούμε το πλήθος των ιδιοτιμών που ανήκουν στο $(-\infty, \alpha)$. Την διαδικασία που μόλις περιγράψαμε την υλοποιήσαμε με την παρακάτω υπορουτίνα.

Υπορουτίνα 3.5

Μέτρηση πλήθους ιδιοτιμών τριδιαγώνιου πίνακα σε διάστημα
Υλοποίηση στο MATLAB

```
function [n]=count_eig(q,a,b)
m=length(q);
if nargin==2
    q=q-a*eye(m); n=detcomp_rec(q);
else
    q2=q-b*eye(m); q1=q-a*eye(m);
    n2=detcomp_rec(q2); n1=detcomp_rec(q1);
    n=n2-n1;
end
end
```

Η υπορουτίνα 3.5 αν έχει δύο ορίσματα σημαίνει ότι το ένα θα είναι ο τριδιαγώνιος πίνακας και το άλλο ένας πραγματικός αριθμός a και έτσι θα υπολογίσει το πλήθος των ιδιοτιμών στο διάστημα $(-\infty, a)$. Αν έχει τρία ορίσματα τότε το ένα θα είναι ο τριδιαγώνιος πίνακας και τα άλλα δύο θα είναι δύο πραγματικοί αριθμοί α, β που θα ορίζουν ένα διάστημα (α, β) και έτσι θα υπολογίσει το πλήθος των ιδιοτιμών σε αυτό το διάστημα. Οι υπορουτίνες 3.4, 3.5 μπορούν να χρησιμοποιηθούν και σε άλλες εφαρμογές ή αλγόριθμους υπολογισμού ιδιοτιμών.

Μία άλλη ιδέα για την υλοποίηση του αλγόριθμου της διχοτόμησης για τριδιαγώνιους πίνακες είναι να εντοπίσουμε το διάστημα που ανήκουν όλες οι ιδιοτιμές του πίνακα A με τη βοήθεια των υπορουτίνων 3.4, 3.5, να θεωρήσουμε κατάλληλη διαμέριση αυτού του διαστήματος και να εντοπίσουμε τις ιδιοτιμές σε αυτά τα διαστήματα. Αυτή την αλγοριθμική διαδικασία υλοποιήσαμε στο MATLAB όπως φαίνεται παρακάτω.

Αλγόριθμος 3.13

Αλγόριθμος διχοτόμησης
Υλοποίηση στο MATLAB

```
function [eigv]=bisection_eig(q)
a=red_hes(q); m=length(a); p=comp_pol(a,m);
```

```

dp=comp_pol(a,m,1); ddp=comp_pol(a,m,2);
p2=[1 -(a(1,1)+a(2,2)) (a(1,1)*a(2,2)-a(1,2)^2)];
c=sort(roots(p2)); ap=c(2)-c(1); c22=c(2)+ap; c11=c(2);
c2=c(1); c1=c2-ap;
nd=count_eig(a,c22); na=count_eig(a,c1);
while na =0 && nd~=m
    if na~=0
        ap=c2-c1; c2=c1; c1=c2-ap;
        na=count_eig(a,c1);
    end
    if nd~=m
        ap=c22-c11; c11=c22; c22=c22+ap;
        nd=count_eig(a,c22);
    end
end
step=(c22-c1)/m; eigv=[];
while length(eigv)<m
    c2=c1+step;
    if count_eig(a,c1,c2)==0
        c1=c1+step;
    else
        eigv=[eigv find_eig(p,dp,ddp,c1,c2)];
        c1=c1+step;
    end
end
eigv=eigv';
function [sol]=find_eig(p,dp,ddp,a1,b1)
    c21=(a1+b1)/2; n_eig=count_eig(a,a1,b1)
    if n_eig==1
        if p(c21)*ddp(c21)>0
            sol=newton_raphson(p,dp,c21,c21+1,0);
        else
            sol=bis_root(p,a1,b1);
        end
    elseif n_eig>1
        sol1=find_eig(p,dp,ddp,a1,c21);
        sol2=find_eig(p,dp,ddp,c21,b1);
        sol=[sol1 sol2];
    else
        sol=[];
    end
end
end
end

```

Ενώ η υπορουτίνα για τον υπολογισμό της ρίζας με τη μέθοδο Newton-Raphson είναι ο παρακάτω αναδρομικός αλγόριθμος.

Υπορουτίνα 3.6

Μέθοδος Newton-Raphson
Υλοποίηση στο MATLAB

```
function [sol]=newton_raphson(p,dp,x0,xold,n)
sol=x0;
if abs(x0-xold)>1e-14 && n<=400
    xold=x0;
    dd=p(x0)/dp(x0);
    x0=x0-dd;
    n=n+1;
    sol=newton_raphson(p,dp,x0,xold,n);
end
```

Η υπορουτίνα 3.6 δέχεται σαν όρισμα ένα πολυώνυμο και την παράγωγό του, δύο σημεία του άξονα x' και ένα φυσικό αριθμό για τη συνθήκη τερματισμού της αναδρομικής διαδικασίας. Η διαδικασία που ακολουθείται είναι η γνωστή διαδικασία σύγκλισης στη ρίζα του πολυωνύμου p σε μια κατάλληλη περιοχή του x_0 .

Στον αλγόριθμο 3.13 η αναδρομική υποσυνάρτηση $find_eig()$ ελέγχει πόσες ιδιοτιμές υπάρχουν σε ένα διάστημα (α, β) . Αν υπάρχει ακριβώς μία και ισχύει $p\left(\frac{\alpha+\beta}{2}\right)p''\left(\frac{\alpha+\beta}{2}\right) > 0$, την υπολογίζει με τη μέθοδο Newton-Raphson, καθώς αυτή θα συγκλίνει σίγουρα, διαφορετικά την υπολογίζει με τη μέθοδο της διχοτόμησης που υλοποιήσαμε στην υπορουτίνα 3.3. Αν δεν υπάρχει ιδιοτιμή μας επιστρέφει ένα κενό διάνυσμα, ενώ αν υπάρχουν πάνω από μία ιδιοτιμές θα καλεστεί η ίδια υποσυνάρτηση για τα διαστήματα $\left(\alpha, \frac{\alpha+\beta}{2}\right)$ και $\left(\frac{\alpha+\beta}{2}, \beta\right)$ μέχρι να εντοπίσει ακριβώς μία ή καμία ιδιοτιμή σε κάποιο διάστημα.

Η κύρια συνάρτηση αρχικά βρίσκει ένα διάστημα που ανήκουν όλες οι ιδιοτιμές του πίνακα που δέχεται σαν όρισμα. Αυτό το καταφέρνει ως εξής: υπολογίζει τις ρίζες $\rho_1 < \rho_2$ του χαρακτηριστικού πολυωνύμου του πίνακα $A^{(2)}$ και ορίζει το $d = \rho_2 - \rho_1$. Στη συνέχεια δημιουργεί την ακολουθία των αριθμών $\rho_1 - d, \rho_1 - 2d, \dots, \rho_1 - \nu d, \dots$ και σταματάει στον πρώτο αριθμό για τον οποίο το διάστημα $(-\infty, \rho_1 - \nu d)$ δεν περιέχει καμία ιδιοτιμή του πίνακα A . Την ίδια διαδικασία εκτελεί και "από δεξιά" δημιουργώντας την ακολουθία αριθμών $\rho_2 + d, \rho_2 + 2d, \dots, \rho_2 + \nu d, \dots$ σταματώντας στον πρώτο αριθμό για τον οποίο ισχύει ότι το διάστημα $(-\infty, \rho_2 + \nu d)$ περιέχει όλες τις ιδιοτιμές του πίνακα A . Έτσι το διάστημα $(\rho_1 - \nu_1 d, \rho_2 + \nu_2 d)$ θα περιέχει όλες τις ιδιοτιμές του πίνακα A . Στη συνέχεια ορίζει το

$$\delta = \frac{(\rho_2 + \nu_2 d) - (\rho_1 - \nu_1 d)}{m}$$

Όπου m η διάσταση του πίνακα A . Στη συνέχεια υπολογίζει τις ρίζες του χαρακτηριστικού πολυωνύμου $\pi(x)$ του πίνακα A στα διαστήματα $(\rho_1 - \nu_1 d, (\rho_1 - \nu_1 d) + \delta), \dots, ((\rho_1 - \nu_1 d) - (m - 1)\delta, \rho_2 + \nu_2 d)$ με τη βοήθεια της υποσυνάρτησης $find_eig()$.

V.III Αλγόριθμος διαίρει και βασίλευε

Στο δεύτερο κεφάλαιο περιγράψαμε σε γενικές γραμμές τον αλγόριθμο διαίρει και βασίλευε, αφήνοντας αρκετές λεπτομέρειες να αποσαφηνιστούν σε αυτή την υποενότητα. Οι αναδρομικοί αλγόριθμοι τύπου διαίρει και βασίλευε ακολουθούν μία χαρακτηριστική προσέγγιση επίλυσης του προβλήματος. **Διαιρούν** το πρόβλημα σε υποπροβλήματα. **"Κυριεύουν"** τα υποπροβλήματα επιλύοντας τα αναδρομικά μέχρι τα στοιχειώδη προβλήματα που τα λύνει απευθείας. **Συνδυάζουν** τις λύσεις των υποπροβλημάτων και συνθέτουν την τελική λύση. Τέτοια χαρακτηριστικά παραδείγματα αλγορίθμων είναι οι υποσυναρτήσεις $find_eig()$ (του αλγόριθμου 3.13), $det_value()$ (της υπορουτίνας 3.3) και η $val_coef_p()$ (της υπορουτίνας 3.2). Ο αλγόριθμος διαίρει και βασίλευε υπολογισμού των ιδιοτιμών ενός τριδιαγώνιου πίνακα είναι ένας τέτοιος αλγόριθμος. Στην υλοποίηση του αλγόριθμου στο MATLAB θα προσπαθήσουμε να υλοποιήσουμε κάθε βήμα του αλγόριθμου σε υπορουτίνες. Πριν την παρουσίαση όμως όλων των επιμέρους βημάτων (υπορουτίνων) θα παρουσιάσουμε τη συνολική υλοποίηση του αλγόριθμου, για να εξετάσουμε αργότερα την υλοποίηση κάθε βήματος. Η υλοποίηση φαίνεται παρακάτω.

Αλγόριθμος 3.14

Αλγόριθμος διαίρει και βασίλευε
Υλοποίηση στο MATLAB

```
function [q,d,t]=div_rec(a,fake)
if nargin==1
    C=a;
    a=red_hessenberg(a);
end
m=length(a);
if m<=20
    [T1,T2,b]=split_it(a);
    [Q1,D1]=QR_wil(T1);
    [Q2,D2]=QR_wil(T2);
```

```

[D,w]=compound_matrices(Q1,Q2,D1,D2,b);
d=divconrizz(a,D,w,b)
if nargin==1
    q=givmeq(C,d);
else
    q=givmeq(a,d);
end
t=sort(diag(d));
else
    [T1,T2,b]=split_it(a);
    [Q1,D1,~]=div_rec(T1,0);
    [Q2,D2,~]=div_rec(T2,0);
    [D,w]=compound_matrices(Q1,Q2,D1,D2,b);
    d=divconrizz(a,D,w,b);
    if nargin==1
        q=givmeq(C,d);
    else
        q=givmeq(a,d);
    end
    t=sort(diag(d));
end
end

```

Η ιδέα είναι να θεωρήσουμε σαν στοιχειώδες πρόβλημα έναν πίνακα διάστασης 10×10 ή μικρότερο. Αυτό το στοιχειώδες πρόβλημα το λύνουμε (διαγωνιοποιούμε δηλαδή τον πίνακα) με τον αλγόριθμο QR με την εναλλαγή Wilkinson (αλγόριθμος 3.8). Έτσι λοιπόν για ένα πίνακα A διάστασης 20×20 , ή μικρότερο, η διαδικασία επίλυσης του προβλήματος που ακολουθεί ο αλγόριθμος 3.14 είναι η εξής: Επιλέγει τους πίνακες $T1, T2$ διάστασης $\sim \frac{m}{2}$, όπου m η διάσταση του αρχικού πίνακα A , διαγωνιοποιεί τους $T1, T2$ και δημιουργεί τους πίνακες και το διάνυσμα της σχέσης 2.59. Στη συνέχεια βρίσκει τις ρίζες της συνάρτησης 2.60 εντοπίζοντας τις στα διαστήματα που περιγράφει το σχήμα 2.6, οι οποίες είναι οι ιδιοτιμές του αρχικού πίνακα A και στη συνέχεια βρίσκει τον πίνακα X της διαγωνιοποίησης του $A = X\Lambda X^{-1}$. Αν ο αρχικός πίνακας A είναι μεγαλύτερης διάστασης, τότε επιλέγει τους πίνακες $T1, T2$ και ακολουθεί την προηγούμενη διαδικασία για αυτούς μέχρι να προκύψουν κάποιο στοιχειώδες προβλήματα από το οποίο θα συνθέσει την τελική μας λύση.

Την επιλογή των πινάκων $T1, T2$ την κάνουμε με την παρακάτω υπορουτίνα.

Υπορουτίνα 3.7


```

Επιλογή πινάκων T1, T2
Υλοποίηση στο MATLAB

function[T1,T2,b]=split_it(a)
m=length(a);
if mod(m,2)==0
    n1=m/2; n2=m-n1;
else
    n1=(m+1)/2; n2=m-n1;
end
b=a(n1,n1+1); T1=a(1:n1,1:n1);
t=length(T1)+1; T2=a(t:m,t:m);
T1(n1,n1)=T1(n1,n1)-b; T2(1,1)=T2(1,1)-b;
end

```

Η υπορουτίνα 3.7 δέχεται έναν τριδιαγώνιο πίνακα A και επιστρέφει τους πίνακες $T1$, $T2$ και το στοιχείο β του σχήματος 2.4.

Η υπορουτίνα που δημιουργεί τους πίνακες και το διάνυσμα της σχέσης 2.59 είναι η παρακάτω.

Υπορουτίνα 3.8

```

Κατασκευή πινάκων και διανύσματος της (2.59)
Υλοποίηση στο MATLAB

function [D,w]=compound_matrices(Q1,Q2,D1,D2,b)
Q=blkdiag(Q1,Q2);
D=blkdiag(D1,D2);
w=[Q1(length(Q1),:) Q2(1,:)];
w=sqrt(b)*w;
end

```

Η υπορουτίνα 3.8 δέχεται σαν όρισμα τους πίνακες Q_i , D_i και το στοιχείο β του πίνακα A και επιστρέφει τους πίνακες και το διάνυσμα της σχέσης 2.59. Η εντολή `blkdiag()` δέχεται δύο πίνακες $X1$, $X2$ και δημιουργεί τον πίνακα

$$\begin{bmatrix} X1 & 0 \\ 0 & X2 \end{bmatrix}$$

Επίσης το διάνυσμα w που δημιουργεί είναι το $\sqrt{\beta}z^T$, όπου z^T το διάνυσμα της σχέσης 2.59, καθώς είναι το αντίστοιχο διάνυσμα w τα στοιχεία της οποίας ορίζουν τη συνάρτηση 2.60 μαζί με τα στοιχεία του πίνακα D . Αν το β είναι αρνητικό τότε η $\sqrt{\beta}$ θα είναι φανταστικός αριθμός και η γραφική παράσταση της συνάρτησης 2.60 δεν θα είναι όπως στο σχήμα 2.5. Οι ρίζες της συνάρτησης θα είναι στα διαστήματα (d_i, d_{i+1}) , για $i = 1, \dots, m-1$, και η m -οστή ρίζα θα βρίσκεται αριστερά του d_1 και όχι δεξιά του d_m . Αυτό θα παίζει ρόλο για τον υπολογισμό των ριζών της 2.60, όπως θα δούμε

παρακάτω. Η υπορουτίνα που επιλέγει τα άκρα των διαστημάτων που θα υπολογίσουμε τις ρίζες της 2.60 είναι η παρακάτω.

Υπορουτίνα 3.9

Επιλογή διαστημάτων για την εύρεση των ριζών της (2.60)
Υλοποίηση στο MATLAB

```
function [d]=find_eig_roots(q,D,w,bm)
m=length(D); D=full(D); l=D; D=sort(diag(D));
d=[]; t1=true; t2=true;
for i=1:m-1
    a=D(i)+eps(); b=D(i+1)-eps();
    d=[d diamr(w,l,a,b,bm,0,t1,t2)];
end
if bm>0
    ap=D(m)-D(m-1);
    a=D(m); b=D(m)+ap;
    while count_eig(q,a,b)~=1
        a=b; b=b+ap;
    end
    t2=false;
else
    ap=D(2)-D(1); b=D(1); a=D(1)-ap;
    while count_eig(q,a,b)~=1
        b=a; a=a-ap;
    end
    t1=false;
end
d=[d diamr(w,l,a,b,bm,0,t1,t2)];
d=diag(sort(d));
end
```

Η υπορουτίνα 3.9 επιλέγει τα διαστήματα (d_i, d_{i+1}) , για $i = 1, \dots, m-1$ και καλεί την υπορουτίνα *diamr* που υπολογίζει τις ρίζες της 2.60 στα διαστήματα αυτά. Αφού υπολογίσει αυτές τις ρίζες θα πρέπει να υπολογίσει ανάλογα με το πρόσημο του β και την m -οστή ρίζα. Στην περίπτωση που το β είναι θετικό ορίζει το διάστημα $(d_m, d_m + \delta)$, όπου $\delta = d_m - d_{m-1}$. Χρησιμοποιεί την υπορουτίνα 3.5 για να δει αν υπάρχει ιδιοτιμή σε αυτό το διάστημα. Αν δεν υπάρχει ορίζει το διάστημα $(d_m + \delta, d_m + 2\delta)$ και συνεχίζει αυτή τη διαδικασία μέχρι να βρει το διάστημα που περιλαμβάνει τη m -οστή ρίζα της 2.60. Αν το β είναι αρνητικό, τότε εκτελεί την ίδια διαδικασία στα αριστερά του d_1 . Βέβαια η παραπάνω διαδικασία εντοπισμού της m -οστής ρίζας μπορεί να φαίνεται θεωρητικά σωστή, υπάρχει όμως ενδεχόμενο (και μάλιστα όχι πολύ απίθανο για μεγάλης διάστασης πίνακες όπως

παρατηρήσαμε από διαδοχικές εκτελέσεις του αλγόριθμου 3.14) να παρουσιάσει πρόβλημα και να μην καταφέρει να εντοπίσει τη ζητούμενη ρίζα. Ας δώσουμε ένα παράδειγμα για να γίνει κατανοητό τι εννοούμε.

Έχουμε δημιουργήσει στο MATLAB έναν 150×150 συμμετρικό, πραγματικό πίνακα και τον έχουμε καταχωρήσει στην μεταβλητή q και εκτελέσαμε τις παρακάτω εντολές.

```
>> z=red_hessenberg(q);
>> [T1,T2,b]=split_it(z);
>> b
b =
    84.064681563073364
>> a=max([eig(T1)' eig(T2)']);
>> eigmax=max(eig(q))
eigmax =
    2.366098369006855e+02
>> a
a =
    2.366098369006859e+02
```

Βλέπουμε ότι το στοιχείο β του Hessenberg πίνακα που παράγεται από τον πίνακα q είναι θετικό. Έτσι η m -οστή ρίζα της 2.60 θα βρίσκεται δεξιά του d_{150} . Όμως βλέπουμε ότι, με βάση τους αριθμητικούς υπολογισμούς των συναρτήσεων του MATLAB, $d_{150} > \lambda_{max}$ με μια διαφοροποίηση στο 16ο δεκαδικό ψηφίο, όπου λ_{max} η μέγιστη ιδιοτιμή του πίνακα q που υπολόγισε η συνάρτηση $eig()$ του MATLAB. Αυτό θεωρητικά δεν μπορεί να συμβεί, όμως πρακτικά μετά από αρκετά βήματα αριθμητικών υπολογισμών και στρογγυλοποιήσεων, όπως στην περίπτωση μας, είναι αρκετά πιθανό να συμβεί. Έτσι αυτό σημαίνει ότι θα χρειαστεί να τροποποιήσουμε την υπορουτίνα 3.9 ώστε να παίρνουμε υπόψιν και αυτή την περίπτωση, που όπως αναφέραμε και παραπάνω, για μεγάλους πίνακες δεν είναι πολύ σπάνια. Αυτή την τροποποίηση όμως θα την κάνουμε αφού πρώτα παρουσιάσουμε και την επόμενη υπορουτίνα, που υπολογίζει τις ρίζες στα διαστήματα που επιλέγει η υπορουτίνα 3.9. Αυτή φαίνεται παρακάτω.

Υπορουτίνα 3.10

Υπολογισμός ρίζας στο διάστημα (d_i, d_{i+1})
 Υλοποίηση στο MATLAB

```
function sol=diamr(w,l,a,b,bm,n,t1,t2)
step=(b-a)/100; d=a:step:b; d2=plm2(d,w,1); d5=d2;
wh_is_inf=find(isinf(d2)); wh_is_nan=find(isnan(d2));
```

```

if t1
if ~isinf(d2(1)) && ~isnan(d2(1))
    if d2(1)*d2(2)<0
        if bm>0
            d2(1)=inf;
        else
            d2(1)=-inf;
        end
        wh_is_inf=find(isinf(d2));
    end
end
d5=d2;
end
if t2
if ~isinf(d2(length(d2))) && ~isnan(d2(length(d2)))
    if d2(length(d2))*d2(length(d2)-1)<0
        if bm>0
            d2(length(d2))=inf;
        else
            d2(length(d2))=-inf;
        end
        wh_is_inf=find(isinf(d2));
    end
end
d5=d2;
end
if bm>0
    d3=find(d5<0); d5(wh_is_inf)=-1; d5(wh_is_nan)=-1;
    d4=find(d5>0);
else
    d3=find(d2>0); d5(wh_is_inf)=1; d5(wh_is_nan)=1;
    d4=find(d5<0);
end
if length(d3)==0
    if abs(d(d4(1))-a)<10^(-13)
        b=d(d4(1)); sol=(a+b)/2;
    elseif isempty(wh_is_inf) && isempty(wh_is_nan)
        b=a; a=a-eps(); sol=(a+b)/2;
    else
        b=d(d4(1)); n=n+1; t1=true; t2=false;
        sol=diamr(w,l,a,b,bm,n,t1,t2);
    end
elseif d3(length(d3))+1<=length(d)
    if isinf(d2(d3(length(d3))+1)) || isnan(d2(d3(length(d3))+1))

```

```

    if abs(b-d(d3(length(d3))))<10^(-13)
        a=d(d3(length(d3))); sol=(a+b)/2;
    else
        a=d(d3(length(d3))); n=n+1; t1=false; t2=true;
        sol=diamr(w,l,a,b,bm,n,t1,t2);
    end
else
    x0=d(d3(length(d3)));
    sol=newr(w,l,x0,x0+1,0);
end
else
    a=b; b=b+eps(); sol=(a+b)/2;
end
end

```

Η διαδικασία υπολογισμού της ρίζας κάθε διαστήματος (d_i, d_{i+1}) περιλαμβάνει αρκετές λεπτομέρειες που πρέπει να προσέξουμε προκειμένου να υλοποιήσουμε μια αποτελεσματική αλγοριθμική διαδικασία αριθμητικής προσέγγισης των ριζών. Σημειώνουμε εδώ ότι στην υπορουτίνα 3.10 συναντήσαμε τη μεγαλύτερη δυσκολία, κυρίως στο να εντοπίσουμε που οφειλόταν κάθε φορά η αποτυχία εκτέλεσης ή εντοπισμού των ριζών. Η κεντρική ιδέα της υπορουτίνας είναι να θεωρήσουμε μια διαμέριση του διαστήματος (d_i, d_{i+1}) και να εκτελέσουμε τη μέθοδο Newton-Raphson όταν και εκεί που εντοπίσουμε αλλαγή στο πρόσημο της συνάρτησης 2.60, καθώς τότε θα έχουμε εντοπίσει το διάστημα που θα περιέχει τη ζητούμενη ρίζα.

Εδώ πριν συνεχίσουμε πρέπει να σημειώσουμε την πρώτη λεπτομέρεια. Από τα παραπάνω είναι φανερό ότι θα χρειαστούμε να υπολογίσουμε τις τιμές της 2.60 αρκετές φορές και κάποιες από αυτές θα έχουμε υπερχειλίση ή υποχειλίση καθώς η 2.60 έχει ασύμπτωτες τις ευθείες $x = d_i$, για $i = 1, \dots, m$. Αρχικά υλοποιήσαμε τη 2.60 στο MATLAB με τη βοήθεια της εντολής *stm.fun()* και μιας μεταβλητής τύπου *syms*. Αυτή η επιλογή έχει αρκετά προβλήματα διότι η 2.60 είναι αρκετά πολύπλοκη και έτσι, πρώτον αργεί στους υπολογισμούς και δεύτερον πολλές φορές αποτυγχάνει να υπολογίσει τις τιμές για διάφορα x κυρίως για αυτά κοντά στις ασύμπτωτες. Επομένως υλοποιήσαμε τη 2.60 και την παράγωγό της με ξεχωριστές υπορουτίνες που φαίνονται παρακάτω.

Υπορουτίνα 3.11

Υλοποίηση της συνάρτησης 2.60
Υλοποίηση στο MATLAB

```

function [y]=plm(x,w,D)
D=diag(D); y=1; m=length(w);

```

```

for i=1:m
    y=y+w(i)^2/(D(i)-x);
end
end

```

Υπορουτίνα 3.12

Υλοποίηση της παραγώγου της συνάρτησης 2.60
 Υλοποίηση στο MATLAB

```

function [y]=dplm(x,w,D)
y=0; m=length(w);
for i=1:m
    y=y+w(i)^2/(D(i,i)-x)^2;
end
end

```

Επίσης υλοποιήσαμε και την παρακάτω υπορουτίνα που της δίνουμε ένα διάστημα με διαμέριση και μας επιστρέφει τις τιμές της 2.60 σε αυτό το διάστημα.

Υπορουτίνα 3.13

Υλοποίηση της συνάρτησης (2.60) σε διάστημα
 Υλοποίηση στο MATLAB

```

function [r]=plm2(x,w,D)
D=diag(D); m=length(w); n=length(x); r=[];
for j=1:n
    y=1;
    for i=1:m
        y=y+w(i)^2/(D(i)-x(j));
    end
    r=[r y];
end
end

```

Επίσης η μέθοδος Newton-Raphson αναγκαστικά θα υλοποιηθεί με διαφορετικό τρόπο από ότι στην υπορουτίνα 3.6, όπως φαίνεται παρακάτω.

Υπορουτίνα 3.14

Μέθοδος Newton-Raphson για τον αλγόριθμο 3.13
 Υλοποίηση στο MATLAB

```

function [sol]=newr(w,l,x0,xold,n)
sol=x0; tol=1e-14;
if abs(x0-xold)>tol && n<=400
    xold=x0; x0=x0-plm(x0,w,l)/dplm(x0,w,l);
    n=n+1; sol=newr(w,l,x0,xold,n);
end
end

```

Τώρα μπορούμε να επιστρέψουμε στην υπορουτίνα 3.10. Θεωρούμε για λόγους απλοποίησης ότι το στοιχείο β είναι θετικό καθώς στην αντίθετη περίπτωση η διαδικασία είναι ίδια. Ο βασικός λόγος που επιλέξαμε να θεωρήσουμε μια διαμέριση κάθε διαστήματος και να εντοπίσουμε την αλλαγή του πρόσημου είναι διότι η μέθοδος Newton-Raphson δεν συγκλίνει προφανώς για όλα τα x_0 στο κάθε διάστημα. Επειδή η 2.60 δεν είναι συνεχής σε όλο το \mathbb{R} είναι πιθανό κάποιο x_n που θα υπολογίσει η μέθοδος Newton-Raphson να μην ανήκει στο αρχικό διάστημα που ανήκει το x_0 ή η μέθοδος να αποτύχει να συγκλίνει για κάποιον άλλο λόγο επειδή το x_0 δεν ήταν αρκετά κοντά στη ρίζα. Έτσι αν θεωρήσουμε μια διαμέριση $d_i + \epsilon_{machine}, d_i + \epsilon_{machine} + \delta_i, d_i + \epsilon_{machine} + 2\delta_i, \dots, d_{i+1} - \epsilon_{machine}$, όπου $\delta_i = \frac{d_{i+1} - \epsilon_{machine} - (d_i + \epsilon_{machine})}{10}$, και υπολογίσουμε τις τιμές της 2.60 σε αυτά τα σημεία όταν θα εντοπίσουμε μια αλλαγή στην ακολουθία των προσήμων της 2.60, θα έχουμε εντοπίσει το διάστημα που ανήκει η ρίζα και έτσι θα εκτελέσουμε τη μέθοδο Newton-Raphson για κάποιο x_0 που ανήκει σε αυτό το διάστημα.

Στην περίπτωση που το πρόσημο της 2.60 παραμένει σταθερό και αρνητικό σε όλα τα σημεία εκτός φυσικά του $d_{i+1} - \epsilon_{machine}$ τότε η ρίζα θα βρίσκεται στο διάστημα $(d_i + \epsilon_{machine} + 9\delta_i, d_{i+1} - \epsilon_{machine})$ και έτσι η υπορουτίνα 3.10 ξανά καλεί τον εαυτό της για αυτό το διάστημα. Αν το πρόσημο της 2.60 παραμένει σταθερό και θετικό τότε η ρίζα θα βρίσκεται στο διάστημα $(d_i + \epsilon_{machine}, d_i + \epsilon_{machine} + \delta_i)$, οπότε η υπορουτίνα 3.10 θα καλέσει τον εαυτό της για αυτό το διάστημα. Αν και στα νέα διαστήματα το πρόσημο της 2.60 παραμένει σταθερό τότε συνεχίζουμε την ίδια διαδικασία μέχρι να έχουμε ένα τόσο μικρό διάστημα ώστε να εκτιμήσουμε τη ρίζα μας με ακρίβεια 14 (μπορούμε να επιλέξουμε και περισσότερα) δεκαδικών ψηφίων.

Οι έλεγχοι $[if \ t1]$ και $[if \ t2]$ και οι έλεγχοι που περιλαμβάνουν υπάρχουν για τον εξής λόγο: Συνήθως οι ακραίες τιμές της 2.60 στα διαστήματα που αναφέραμε έχουν την τιμή inf , καθώς στον υπολογισμό της 2.60 έχει συμβεί υπερχείλιση ή υποχείλιση, λόγω της μορφής της 2.60. Σε κάποιες περιπτώσεις όμως αυτό δεν συμβαίνει και απλώς η 2.60 έχει μια μεγάλη αριθμητική τιμή σε αυτά τα σημεία. Τότε αν για παράδειγμα το πρόσημο της 2.60 παραμένει σταθερό και αρνητικό στο διάστημα (εκτός της δεξιάς ακραίας τιμής), η υπορουτίνα 3.10 θα εντοπίσει μια αλλαγή πρόσημου μεταξύ των σημείων $(d_i + \epsilon_{machine} + 9\delta_i, d_{i+1} - \epsilon_{machine})$ και θα εκτελέσει τη μέθοδο Newton-Raphson η οποία όμως δεν θα συγκλίνει καθώς δεν βρίσκεται αρκετά κοντά στη ρίζα. Το ίδιο θα συμβεί και στην περίπτωση που το

πρόσημο της 2.60 παραμένει σταθερό και θετικό και δεν έχει συμβεί υποχειλίση στο σημείο $d_i + \epsilon_{machine}$. Για να αποφύγουμε αυτό το ενδεχόμενο ελέγχουμε τις δύο ακραίες τιμές της 2.60 και αν δεν έχει συμβεί υπερχειλίση ή υποχειλίση και υπάρχει αλλαγή πρόσημου με το γειτονικό τους σημείο τότε θέτουμε τις ακραίες τιμές της 2.60 ίσες με inf . Οι λογικές μεταβλητές $t1, t2$ ορίζουν το αν θα κάνουμε τους ελέγχους για τις δύο ακραίες τιμές της 2.60 καθώς σε κάποια αναδρομικά βήματα που έχουμε αλλάξει τα άκρα του διαστήματος (είτε το δεξί είτε το αριστερό, όπως είδαμε προηγουμένως) δεν θέλουμε να γίνει ο έλεγχος. Επίσης γίνεται και έλεγχος στην περίπτωση που η 2.60 συντηρεί το θετικό πρόσημο σε όλα τα σημεία της διαμέρισης (και τα ακραία). Τότε η ρίζα θα βρίσκεται στο διάστημα $(d_i, d_i + \epsilon_{machine})$, οπότε θεωρούμε σαν ρίζα το μέσο αυτού του διαστήματος με αποτέλεσμα να έχουμε μια πολύ καλή προσέγγιση της ρίζας.

Η μόνη εκκρεμότητα που έχουμε είναι η διόρθωση της υπορουτίνας 3.9 στην περίπτωση αποτυχίας εντοπισμού της ιδιοτιμής στα δεξιά (ή στα αριστερά) του d_m , για $\beta > 0$ (ή στα αριστερά του d_1 για $\beta < 0$). Η υπορουτίνα μετά τη διόρθωση θα είναι ως εξής:

Υπορουτίνα 3.15

Επιλογή διαστημάτων για την εύρεση των ριζών της (2.60) [Διόρθωση]
Υλοποίηση στο MATLAB

```
function [d]=find_eig_roots(q,D,w,bm)
m=length(D); D=full(D); l=D; D=sort(diag(D));
d=[]; opa=0; t1=true; t2=true;
for i=1:m-1
    a=D(i)+eps(); b=D(i+1)-eps();
    d=[d diamr(w,l,a,b,bm,0,t1,t2)];
end
last=true;
if bm>0
    ap=D(m)-D(m-1); a=D(m); b=D(m)+ap;
    if count_eig(q,a,b)~=1
        a2=(a+d(length(d)))/2;
        if count_eig(q,a2,b)==1
            f=a-5*10^(-16);
            d=[d f]; last=false;
        else
            a=b; b=b+ap;
            while count_eig(q,a,b)~=1
                a=b; b=b+ap;
            end
        end
    end
end
```



```

        t2=false;
    end
end
else
ap=D(2)-D(1); b=D(1); a=D(1)-ap;
if count_eig(q,a,b)~=1
    b2=(b+d(1))/2;
    if count_eig(q,a,b2)==1
        f=b+5*10^(-16);
        d=[d f]; last=false;
    else
        b=a; a=a-ap;
        while count_eig(q,a,b)~=1
            b=a; a=a-ap;
        end
        t1=false;
    end
end
end
if last
    d=[d diamr(w,l,a,b,bm,0,t1,t2)];
end
d=diag(sort(d));
end

```

Ουσιαστικά αλλάξαμε τον κώδικα μέσα στον έλεγχο [if $\beta > 0$] (θεωρούμε για απλούστευση ότι $\beta > 0$). Προσθέσαμε τον έλεγχο για το αν η m -οστή ρίζα της 2.60 βρίσκεται πριν το d_m και για την ακρίβεια στο διάστημα $(\frac{\rho_{m-1} + d_m}{2}, d_m)$, όπου ρ_{m-1} η $(m-1)$ -οστή ρίζα της 2.60. Τότε θέτουμε σαν m -οστή ρίζα της 2.60 την τιμή $\rho_m = d_m - 5 \cdot 10^{-16}$, καθώς η m -οστή ρίζα θα βρίσκεται τόσο κοντά στο d_m με ακρίβεια 15 δεκαδικών ψηφίων. Επίσης κάνουμε τον αντίστοιχο έλεγχο αν $\beta < 0$.

Σημείωση: Για να είναι ο αλγόριθμος 3.14 λειτουργικός θα πρέπει να αλλάξουμε την υπορουτίνα 3.4 καθώς ο υπολογισμός της 2.57 με την αναδρομική υποσυνάρτηση *det_value()* είναι αρκετά αργός με αποτέλεσμα ο αλγόριθμος 3.14 για μεγάλους πίνακες πρακτικά να αχρηστεύεται. Η επαναληπτική υλοποίηση της 2.57 όπως φαίνεται παρακάτω είναι αρκετά πιο γρήγορη σε σχέση με την αναδρομική.

Υπορουτίνα 3.16

Επαναληπτική υλοποίηση της (2.57) και εύρεση

πλήθους αρνητικών ιδιοτιμών από την ακολουθία Sturm
Υλοποίηση στο MATLAB

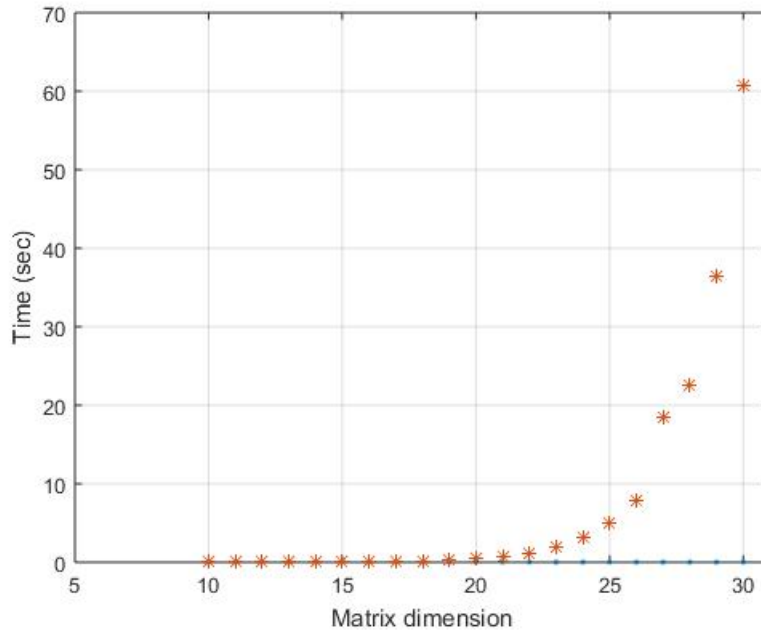
```
function [c]=detcomp(a)
y=[1; c=0; m=length(a); detpin=[a(1,1),det(a(1:2,1:2))];
for i=3:m
    dt=a(i,i)*detpin(i-1)-a(i,i-1)^2*detpin(i-2);
    detpin=[detpin dt];
end
y=[y detpin];
for i=1:m
    if y(i)*y(i+1)<0
        c=c+1;
    elseif y(i)*y(i+1)==0
        if y(i)==0 && y(i+1)~=0
            c=c+1;
        end
    end
end
end
end
```

Στο παρακάτω διάγραμμα φαίνονται οι χρόνοι που απαιτούνται για την εκτέλεση των υπορουτίνων 3.4 και 3.16. Η διαφορά είναι τεράστια. Η υπορουτίνα 3.16 είναι πολύ πιο γρήγορη από την 3.3.

Στη συνέχεια θα συγκρίνουμε τον αλγόριθμο 3.14 και τον αλγόριθμο 3.8 όταν θέλουμε να υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα.

Βλέπουμε στο σχήμα 3.8 ότι παρά τις δυσκολίες που αντιμετωπίσαμε στην υλοποίηση και τα όρια του αλγόριθμου 3.14 που θα δείξουμε στη συνέχεια το αποτέλεσμά μας συμφωνεί με τη θεωρία.

Ποια είναι τα όρια όμως του αλγόριθμου 3.14; Αν παρατηρήσουμε το σχήμα 3.8 θα δούμε ότι επιλύσαμε προβλήματα διάστασης 85×85 το πολύ. Κάναμε αυτή την επιλογή διότι ο αλγόριθμος 3.14 είναι αρκετά ασταθής για προβλήματα διάστασης περίπου 80×80 ή μεγαλύτερα με αποτέλεσμα έως και κάποιες ιδιοτιμές να τις υπολογίζει με ακρίβεια μόνο ενός δεκαδικού. Για παράδειγμα παρακάτω παραθέτουμε τις λύσεις δύο τυχαίων συμμετρικών πινάκων, με στοιχεία από την κανονική κατανομή (ο ένας 20×20 και ο άλλος 40×40) με τον αλγόριθμο 3.14.



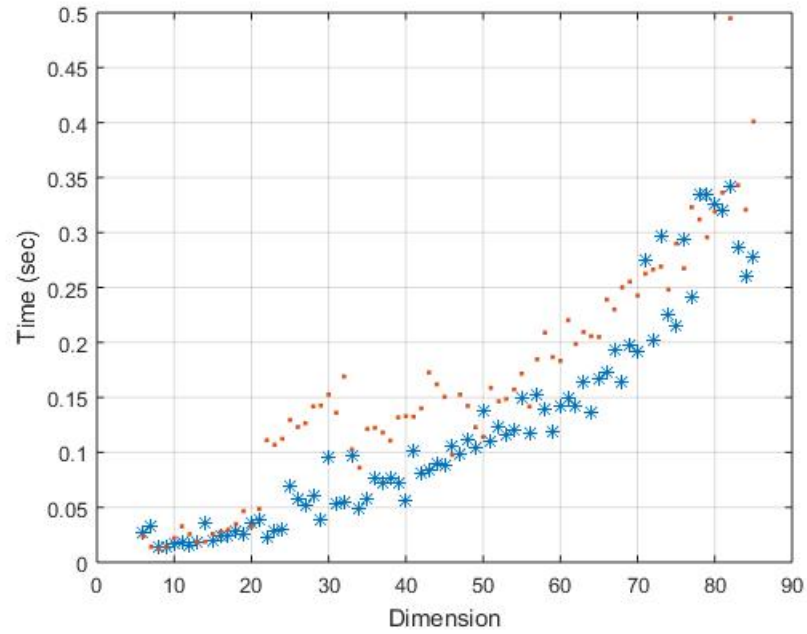
Σχήμα 3.7: Οι χρόνοι που απαιτούνται για την εκτέλεση των υπορουτίνων 3.4 και 3.16 για τριδιαγώνιους πίνακες διάστασης 10×10 έως 30×30 . Με “.” σημειώνονται οι χρόνοι εκτέλεσης της υπορουτίνας 3.16 και με “*” οι χρόνοι εκτέλεσης της υπορουτίνας 3.4.

Οι ιδιοτιμές του συμμετρικού 20×20 πίνακα που υπολόγισε ο αλγόριθμος 3.14

-71.4476	-30.4167	-2.2999	33.0342
-61.9168	-26.6868	8.3391	37.1086
-59.1500	-20.0003	17.4995	45.4294
-53.6856	-11.8087	19.3043	61.4814
-51.1162	-10.7198	28.4832	75.3099

Η απόλυτη διαφορά των ιδιοτιμών του συμμετρικού 20×20 που υπολόγισε ο αλγόριθμος 3.14 από αυτές που υπολόγισε ο αλγόριθμος του MATLAB (eig()).

1.0e-12 *



Σχήμα 3.8: Οι χρόνοι που απαιτούνται για την εκτέλεση των αλγορίθμων 3.8 και 3.14 για τυχαίους, συμμετρικούς και πραγματικούς πίνακες διάστασης 6×6 έως 85×85 . Με "." σημειώνονται οι χρόνοι εκτέλεσης του αλγορίθμου 3.8 και με "*" οι χρόνοι εκτέλεσης του αλγορίθμου 3.14.

0.0568	0.0107	0.0075	0.0284
0.0284	0.0178	0.0053	0.0355
0.0355	0	0.0142	0.0142
0.0284	0.0160	0	0.0426
0.0213	0.0018	0.0355	0.2558

Οι ιδιοτιμές του συμμετρικού 40×40 πίνακα που υπολόγισε ο αλγόριθμος 3.14

-121.9969			
-102.1165	-48.4199	-0.5877	54.3579
-100.5851	-45.6110	1.2806	62.0208
-94.3954	-43.6976	6.8191	63.9526
-83.6886	-31.8089	12.2712	67.2498
-75.2800	-28.4723	15.7702	74.7042
-72.3115	-20.3814	19.5624	82.3835
-66.5275	-18.5340	26.5001	85.8178
-62.6911	-7.1562	34.3068	97.0866
-58.0836	-6.4293	39.3712	113.6750
	-1.9964	52.1753	122.8374

Η απόλυτη διαφορά των ιδιοτιμών του συμμετρικού 40×40 πίνακα που υπολόγισε ο αλγόριθμος 3.14 από αυτές που υπολόγισε ο αλγόριθμος του MATLAB (eig()).

1.0e-12 *				
	0.0142	0.0142	0.0008	0.0426
0.0568		0.0284	0.0262	0.0071
0.0711		0.0497	0.0533	0.0924
0		0.0746	0.0195	0.0711
0.0426		0.0284	0.0195	0.1279
0.0284		0.0320	0.0107	0.0284
0.0284		0.0142	0.0249	0.1421
0.0568		0.0240	0.0639	0.1563
0.0071		0.0027	0.0284	0.0142
0.0711		0.0298	0.0071	0.1563

Οι ιδιοτιμές του συμμετρικού 60×60 πίνακα που υπολόγισε ο αλγόριθμος 3.14

-159.6152	-73.9708	-19.8592	22.1228	75.9343
-151.2216	-73.0562	-14.2062	24.1505	79.8497
-136.1900	-66.8747	-13.0259	31.3758	86.2017
-123.5143	-58.5112	-7.3269	35.8857	91.1487
-113.1117	-55.1247	-4.5547	39.2489	95.1556
-110.2846	-54.5246	-3.6760	44.9729	103.9082
-105.9982	-48.0284	2.2972	50.4521	105.2139
-96.2045	-40.9705	5.3777	53.1519	108.7072
-92.3846	-37.0537	10.3151	59.4586	115.2512
-86.5683	-29.8334	14.0317	66.4072	122.4749
-83.4481	-25.2156	17.2166	68.8294	131.1924
-78.7566	-24.5877	19.9450	69.9488	131.9081

Η απόλυτη διαφορά των ιδιοτιμών του συμμετρικού 60×60 πίνακα που υπολόγισε ο αλγόριθμος 3.14 από αυτές που υπολόγισε ο αλγόριθμος του MATLAB (eig()).

1.0e-12 *					
	0.0284	0.0568	0.0178	0.0107	0.1421
0		0.0142	0.0941	0.0178	0.0711
0.0284		0.0142	0.0195	0.1137	0.0142
0.0284		0	0.0258	0.0426	0.0995
0.1137		0.0853	0.0027	0.0924	0.0284
0.1137		0.0142	0.0511	0.0142	0.0995
0		0	0.0835	0.0142	0.0284
0.0711		0.0071	0.0107	0.1634	0.0853
0.0568		0.0142	0.0586	0.0497	0.0284
0.0426		0.0213	0.0675	0.0284	0.0711
0.0142		0.0178	0.0213	0.0142	0.1137
0.0284		0.0391	0.1350	0.0995	0.0284

Οι ιδιοτιμές του συμμετρικού 80×80 πίνακα που υπολόγισε ο αλγόριθμος 3.14

-169.0993	-84.2437	-30.4010	27.6074	90.9403
-160.5152	-76.6495	-27.0064	32.4881	92.8449
-148.7645	-75.9557	-22.0131	39.5688	97.0123
-146.3828	-74.2140	-16.3725	41.7617	106.0569
-142.0676	-71.3784	-14.5581	46.0844	108.5449
-132.3301	-63.7687	-12.4217	49.7348	110.9613
-127.1680	-59.4459	-10.0285	52.7151	118.0900
-125.7683	-57.3288	-5.0196	56.2156	121.6676
-116.6153	-55.7299	0.6310	59.9465	127.2115
-113.9749	-53.4267	2.5885	62.0984	131.1216
-108.4585	-50.7063	5.9418	64.3105	132.9242
-104.4445	-50.0625	10.1998	68.3183	141.4426
-99.5139	-40.7949	19.3224	77.6319	150.4414
-95.2178	-40.1378	20.4566	80.0800	159.4394
-93.5354	-37.0544	21.7380	84.1735	167.6244
-89.4115	-32.7153	25.1599	84.8476	167.9470

Η απόλυτη διαφορά των ιδιοτιμών του συμμετρικού 80×80 πίνακα που υπολόγισε ο αλγόριθμος 3.14 από αυτές που υπολόγισε ο αλγόριθμος του MATLAB (eig()).

1.0e-12 *

0.1421	0	0.0355	0.0213	0.0426
0.5116	0.0142	0.0284	0.0355	0.0284
0.0284	0.0426	0.0284	0.0355	0.0568
0.0284	0.0142	0.0497	0.0213	0.0284
0.0284	0.0426	0.0426	0.0284	0.0568
0.0284	0.0711	0.0941	0.0142	0
0.0711	0.0426	0.0160	0.0497	0.0711
0.0142	0.0497	0.0533	0.0071	0.0284
0.0853	0.0142	0.0543	0.0142	0
0.0568	0.0497	0.0200	0.0071	0.0568
0.0284	0.0497	0.0240	0.0142	0.0284
0.0426	0	0.0622	0.0426	0.1137
0.0426	0.1137	0.0604	0.0142	0.0284
0.0142	0.0568	0.0462	0.0426	0.1421
0	0.0497	0.0284	0.0284	0.0568
0.0142	0.0142	0.0213	0.0142	0.1137

Οι ιδιοτιμές του συμμετρικού 100×100 πίνακα που υπολόγισε ο αλγόριθμος 3.14

-189.8154	-94.4745	-30.8854	30.7849	102.7530
-181.3803	-88.8909	-29.1189	33.7336	105.4511
-169.1745	-87.6202	-26.6750	36.3913	106.6800
-165.8685	-83.7205	-21.8028	39.1835	112.2692
-159.0783	-80.2228	-15.8510	41.9921	115.7383
-155.9293	-79.5059	-14.2096	43.1879	120.9475
-152.0097	-78.0545	-11.1777	47.7853	123.3941
-147.1453	-75.2665	-7.4309	51.1341	126.1283
-142.7394	-73.0918	-2.9657	55.7402	128.1400
-139.6552	-67.3711	-1.2027	60.0339	133.7705
-136.4146	-64.1290	3.9154	61.8913	136.8847
-135.3176	-59.8226	6.5174	68.0896	139.8166
-130.0635	-55.8200	7.2814	71.6331	150.7086
-126.9086	-51.8423	10.8227	73.7693	153.2964
-120.6771	-50.4838	15.8130	75.5632	158.4013
-117.3463	-48.8101	17.1791	81.1876	161.0981
-112.2038	-44.2901	19.3004	83.8744	163.4770
-108.0343	-41.3729	21.4077	85.7963	170.1004
-101.9330	-38.9984	24.7188	86.8240	176.0748
-97.6133	-32.7768	29.9303	98.0786	186.6808

Η απόλυτη διαφορά των ιδιοτιμών του συμμετρικού 100×100 πίνακα που

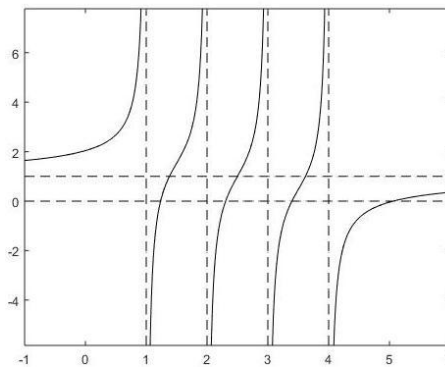
υπολόγισε ο αλγόριθμος 3.14 από αυτές που υπολόγισε ο αλγόριθμος του MATLAB (eig()).

0.0000	0.0000	0.0001	0.0252	0.3894
0.0000	0.0008	0.0002	0.0002	0.1946
0.0000	0.0131	0.0014	0.0169	0.0058
0.0000	0.0087	0.0073	0.0684	0.4876
0.0000	0.0001	0.0000	0.0614	0.5598
0.0000	0.0000	0.0113	0.0031	4.0097
0.0000	0.0014	0.0091	0.0396	1.4847
0.0000	0.0146	0.0003	0.0194	1.1742
0.0005	0.0001	0.0031	0.0065	0.2187
0.0000	0.0008	0.0046	0.0130	1.4964
0.0005	0.0000	0.0093	0.0119	0.6723
0.0003	0.0143	0.0001	0.1676	3.3225
0.0000	0.0036	0.0402	0.0251	0.0013
0.0005	0.0027	0.0021	0.0037	0.0005
0.0006	0.0016	0.0076	0.1451	0.0007
0.0000	0.0045	0.0338	0.0012	0.0000
0.0028	0.0026	0.0009	0.0055	0.0000
0.0053	0.0204	0.0114	0.0199	0.0000
0.0174	0.0041	0.0380	0.0004	0.0000
0.0011	0.0104	0.0374	0.2271	0.0000

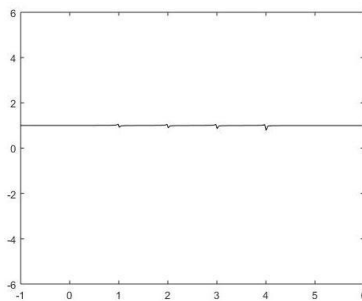
Παρατηρούμε ότι μέχρι και για τους 80×80 συμμετρικούς πίνακες ο αλγόριθμος 3.14 υπολογίζει με αρκετά καλή ακρίβεια τις ιδιοτιμές μας. Οι περισσότερες υπολογισθέντες ιδιοτιμές έχουν 13 δεκαδικά ψηφία ακρίβεια σε σχέση με αυτές που υπολογίζει η MATLAB, κάποιες είναι ακριβώς ίδιες με της MATLAB και κάποιες λίγες έχουν 12 δεκαδικά ψηφία ακρίβεια. Μπορούμε να πούμε ότι μέχρι και για συμμετρικούς πίνακες περίπου 80×80 ο αλγόριθμος 3.14 είναι αρκετά ικανοποιητικός. Επίσης τα αναδρομικά βήματα εκτελούνται με ευσταθή τρόπο χωρίς να χάνεται ακρίβεια.

Για μεγαλύτερους πίνακες δεν έχουμε την ίδια εικόνα. Όπως βλέπουμε και στο παράδειγμά μας για ένα συμμετρικό 100×100 πίνακα η ακρίβεια πολλών ιδιοτιμών είναι αρκετά κακή. Οι ιδιοτιμές που έχουν ακρίβεια 12 δεκαδικά ψηφία είναι πολύ λίγες, οι περισσότερες έχουν ακρίβεια 3 ή 4 δεκαδικών ψηφίων, ενώ κάποιες ιδιοτιμές δεν υπολογίζονται καθόλου. Ουσιαστικά ο αλγόριθμος 3.14 αχρηστεύεται για πίνακες μεγαλύτερης διάστασης από περίπου 80×80 .

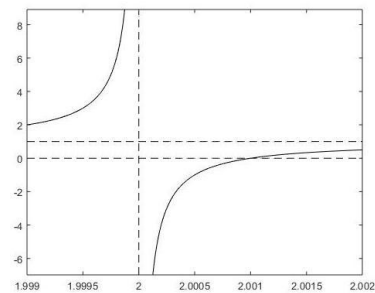
Αυτό το πρόβλημα πηγάζει κυρίως από το φαινόμενο που είδαμε στο δεύτερο κεφάλαιο και χαρακτηρίζει τα στοιχεία των πινάκων των ιδιοδιανυσμάτων των τριδιαγώνιων πινάκων που προκύπτουν κατά την εκτέλεση του αλγόριθμου 3.14. Όπως είδαμε πολλά από τα στοιχεία των πρώτων και των τελευταίων σειρών αυτών των πινάκων είναι πολύ κοντά στο μηδέν με



Σχήμα 3.9: Η γραφική παράσταση της
 $f_1(\lambda) = 1 + \frac{0.5}{1-\lambda} + \frac{0.5}{2-\lambda} + \frac{0.5}{3-\lambda} + \frac{0.5}{4-\lambda}$



Σχήμα 3.10: $f_2(\lambda) = 1 + \frac{10^{-3}}{1-\lambda} + \frac{10^{-3}}{2-\lambda} + \frac{10^{-3}}{3-\lambda} + \frac{10^{-3}}{4-\lambda}$



Σχήμα 3.11: Η γραφική παράσταση της $f_2(\lambda)$ σε μία περιοχή του d_2 .

αποτέλεσμα να επηρεάζουν καθοριστικά τη μορφή της συνάρτησης 2.60 και τη σύγκλιση της μεθόδου Newton-Raphson.

Όπως βλέπουμε στα σχήματα 3.10 και 3.11 όταν το w_i^2 είναι σχετικά μεγάλο η μέθοδος Newton-Raphson δεν θα αντιμετωπίσει κάποιο πρόβλημα στη σύγκλιση της, η οποία μάλιστα θα είναι αρκετά γρήγορη, αν ξεκινήσει από κάποιο κατάλληλο σημείο. Αντίθετα όταν το w_i^2 είναι αρκετά μικρό η γραφική παράσταση της συνάρτησης 2.60 γίνεται σχεδόν παράλληλη με τον άξονα $x'x$ στο διάστημα (d_i, d_{i+1}) , ενώ η ρίζα βρίσκεται αρκετά κοντά στο d_i . Αυτό έχει σαν αποτέλεσμα η μέθοδος Newton-Raphson στον αλγόριθμο 3.14 είτε να αποτύχει στη σύγκλιση της είτε να συγκλίνει αρκετά αργά στην ακρίβεια που θέλουμε.

Επιπλέον μία ιδιότητα που δεν εκμεταλλεύεται ο αλγόριθμος 3.14 έχει να κάνει με τον υπολογισμό του πίνακα ιδιοδιανυσμάτων του πίνακα T της σχέσης 2.59. Προφανώς αν Q' είναι ο πίνακας ιδιοδιανυσμάτων του πίνακα

$D + \beta z z^T$ τότε ο πίνακας ιδιοδιανυσμάτων Q του T θα είναι ο

$$Q = \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \cdot Q'$$

Επομένως το ερώτημα είναι αν μπορούμε να υπολογίσουμε τον πίνακα Q' πιο εύκολα από ότι με τη βοήθεια της επανάληψης πηλίκου Rayleigh ή της αντίστροφης επανάληψης. Το παρακάτω λήμμα φαίνεται να δίνει την απάντηση.

Λήμμα V.1. Έστωσαν ο διαγώνιος πίνακας $D \in \mathbb{R}^{m \times m}$, το διάνυσμα $w \in \mathbb{R}^m$ και ο πραγματικός αριθμός β . Αν το α είναι ιδιοτιμή του πίνακα $D + \beta w w^T$ τότε το $(D - \alpha I)^{-1}$ είναι το αντίστοιχο ιδιοδιάνυσμα της.

Απόδειξη. Με βάση το [7] λήμμα 5.2

Έχουμε

$$\begin{aligned} (D + \beta w w^T)[(D - \alpha I)^{-1}z] &= (D - \alpha I + \alpha I + \beta z z^T)[(D - \alpha I)^{-1}z] \\ &= z + \alpha(D - \alpha I)^{-1}z + z[\beta z^T(D - \alpha I)^{-1}z] \\ &= z + \alpha(D - \alpha I)^{-1}z - z \quad [\text{αφού } \beta z^T(D - \alpha I)^{-1}z + 1 = f(\alpha) = 0] \\ &= \alpha[(D - \alpha I)^{-1}z] \end{aligned}$$

που είναι το ζητούμενο. \square

Μάλιστα αφού ο $D - \alpha I$ είναι διαγώνιος ο υπολογισμός του $(D - \alpha I)^{-1}$ απαιτεί μόλις $O(m)$ πράξεις και επομένως ο υπολογισμός και των m ιδιοδιανυσμάτων απαιτούν $O(m^2)$ πράξεις.

Όμως παρόλο που το λήμμα V.1 φαίνεται να μας λύνει τα χέρια η πραγματικότητα είναι διαφορετική καθώς ο υπολογισμός των ιδιοδιανυσμάτων με αυτό τον τρόπο δεν είναι πάντα αριθμητικά ευσταθής. Στην περίπτωση που έχουμε δύο ιδιοτιμές α_i, α_{i+1} πολύ κοντά ο υπολογισμός των ιδιοδιανυσμάτων με βάση το λήμμα V.1 μπορεί να έχει ως αποτέλεσμα δύο μη ορθογώνια ιδιοδιανύσματα x_i, x_{i+1} .

Αυτό το πρόβλημα μαζί με το πρόβλημα που αναφέραμε πιο πάνω για το μέγεθος των w_i^2 ήταν τα δύο βασικά προβλήματα που καθυστέρησαν την πρακτική υλοποίηση του αλγόριθμου διαίρει και βασίλευε μία δεκαετία από τη στιγμή που θεμελιώθηκε θεωρητικά.

Περισσότερα για τον αλγόριθμο διαίρει και βασίλευε στο [7] υποενότητα 5.3.3.

Βιβλιογραφία

- [1] Lloyd N. Trefethen and David Bau, III.
Numerical Linear Algebra.
Siam, Philadelphia 1997.
- [2] Gene H. Golub and Charles F. Van Loan.
Matrix Computations.
Johns Hopkins University Press 1996.
- [3] Δημήτρης Κυτάγιας - Λάζαρος Βρυζίδης.
ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΛΥΣΗ. Αλγοριθμική προσέγγιση.
Εκδόσεις ΙΩΝ 1991.
- [4] Παναγιώτης Ι. Ψαρράκος.
Θέματα Ανάλυσης πινάκων.
- [5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
Εισαγωγή στους αλγόριθμους.
Μετάφραση Ιωάννης Παπαδόγγονας.
Πανεπιστημιακές εκδόσεις Κρήτης 2015
- [6] Γ.Σ. Παπαγεωργίου, Χ.Γ. Τσίτουρας
Αριθμητική Ανάλυση, με εφαρμογές σε Matlab και Mathematica.
Εκδόσεις ΣΥΜΕΩΝ 2008
- [7] James W. Demmel.
Applied Numerical Linear Algebra.
Siam, Philadelphia 1997.
- [8] Β. Δούγαλης, Δ. Νούτσος, Α. Χατζηδής.
Σημειώσεις Αριθμητικής Γραμμικής Άλγεβρας.
Βόλος, Αύγουστος 2007.
- [9] Ανάργυρος Γ. Φελλούρης
Γραμμική άλγεβρα και αναλυτική γεωμετρία.
Αθήνα, Σεπτέμβριος 2006.