ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
Θεωρητική Πληροφορική και Θεωρία Συστημάτων Ελέγχου

ΥΠΟΛΟΓΙΣΜΟΣ ΜΗΔΕΝΙΚΩΝ ΠΟΛΥΩΝΥΜΙΚΩΝ ΠΙΝΑΚΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Τιφτίκογλου Ιορδάνης

ΕΠΙΒΛΕΠΩΝ
Νικόλαος Καραμπετάκης
Ανάπ. Καθηγητής Α.Π.Θ.

ΘΕΣΣΑΛΟΝΙΚΗ 2012

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
Θεωρητική Πληροφορική και Θεωρία Συστημάτων Ελέγχου

ΥΠΟΛΟΓΙΣΜΟΣ ΜΗΔΕΝΙΚΩΝ ΠΟΛΥΩΝΥΜΙΚΩΝ ΠΙΝΑΚΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Τιφτίκογλου Ιορδάνης

ΕΠΙΒΛΕΠΩΝ
Νικόλαος Καραμπετάκης
Ανάπ. Καθηγητής Α.Π.Θ.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή . . . .

. . . .	. . . .	. . . .
Ε. Αντωνίου	*Α. Βαρδουλάκης*	*Ν. Καραμπετάκης*
Επικ. Καθ. Α.Τ.Ε.Ι. Θεσσαλονίκης Γενικό Τμήμα	Καθηγητής	Αναπ. Καθηγητής

ΘΕΣΣΑΛΟΝΙΚΗ 2012

. . . .
Τιφτίκογλου Ιορδάνης

Περίληψη

Η μελέτη πινάκων πολυωνύμων είναι ιδιαίτερης σημασίας μιας και εμφανίζονται κατά τη μελέτη συστημάτων συνήθων διαφορικών εξισώσεων σταθερών συντελεστών. Στην εργασία αυτή μελετάμε το πρόβλημα του υπολογισμού των μηδενικών πινάκων πολυωνύμων. Πιο συγκεκριμένα, μελετάμε τις συνθήκες υπό τις οποίες το δυϊκό και το ευθύ πολυώνυμο έχουν κοινές ρίζες και παρουσιάζουμε και υλοποιούμε έναν αλγόριθμο για τον υπολογισμό των μηδενικών του οποίου βασικό συστατικό είναι η ανάλυση ιδιαζόντων τιμών.

Λέξεις κλειδιά:

Πολυωνύμικός πίνακας, μηδενικά, δυϊκό πολυώνυμο, ανάλυση ιδιαζόντων τιμών.

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια του προγράμματος μεταπτυχιακών σπουδών του Τμήματος Μαθηματικών, στην ειδίκευση Θεωρητικής Πληροφορικής και Θεωρίας Συστημάτων και Ελέγχου.

Abstract

The study of matrix polynomials is of special importance since matrix polynomials arise when studying systems of ordinary differential equations with constant coefficients. In this thesis we investigate the problem of determining the zeros of matrix polynomials. More precisely, we study the conditions under which the zeros of a polynomial coincide with those of its dual polynomial and we present and implement a singular value decomposition based algorithm for the numerical calculation of the zeros of a matrix polynomial.

Keywords:

Matrix polynomials, zeros, dual polynomial, singular value decomposition.

Ευχαριστίες

Από τη θέση αυτή θέλω να ευχαριστήσω τον επιβλέποντα μου Αναπ. Καθηγητή Ν. Καραμπετάκη για την ευκαιρία που μου παρείχε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Η επιστημονική του κατάρτιση και καθοδήγηση καθώς και η διδακτική του εμπειρία και προσέγγιση κατέστησαν δυνατή τη διεκπεραίωση της παρούσας εργασίας. Οφείλω ακόμη να ευχαριστήσω τα υπόλοιπα μέλη της τριμελούς επιτροπής Επικ. Καθηγητή Ε. Αντωνίου και Καθηγητή Α. Βαρδουλάκη για το χρόνο που αφιέρωσαν στη μελέτη και στην αξιολόγηση της εργασίας αυτής καθώς και για τις εύστοχες παρατηρήσεις τους. Τέλος, θα ήθελα να αναφερθώ στον καταλυτικό ρόλο της οικογένειας μου χωρίς τη διαρκή στήριξη της οποίας δε θα βρισκόμουνα στην ευχάριστη θέση της ολοκλήρωσης των μεταπτυχιακών μου σπουδών.

Τιφτίκογλου Ιορδάνης
Θεσσαλονίκη 2012

Αφιερώνεται στους γονείς μου

Περιεχόμενα

Περιεχόμενα **10**

1 Προαπαιτούμενες Γνώσεις **13**
1.1 Στοιχεία Γραμμικής Άλγεβρας 13
1.2 Πολυώνυμα 17
1.3 Περιγραφή της Ανάλυσης Ιδιαζόντων Τιμών 18
1.4 Βασικές Εφαρμογές της ΑΙΤ 18
1.5 Σχόλια στην Υλοποίηση της ΑΙΤ 20

2 Ανάλυση Ιδιαζόντων Τιμών **21**
2.1 Ιστορικά Στοιχεία 21
2.2 Ύπαρξη της ΑΙΤ 22
2.3 Ανάλυση QR 24
2.4 Ανάλυση QR και Πίνακας Householder 25
2.5 Αλγόριθμος Υπολογισμού της ΑΙΤ 28

3 Πολυωνυμικοί Πίνακες **35**
3.1 Στοιχεία Πολυωνυμικών Πινάκων 35
3.2 Κανονικές Μορφές, Ακολουθίες Jordan και μορφή Smith 37
3.3 Ρίζες Πολυωνυμικών Πινάκων 40

4 Υπολογισμός Μηδενικών Πινάκων Πολυωνύμων **47**
4.1 Εισαγωγικά Στοιχεία 47
4.2 Μηδενικά Πολυωνυμικών Πινάκων και ΑΙΤ 48

5 Συζήτηση και Συμπεράσματα **63**

Α΄ Εκτέλεση Αριθμητικών Πειραμάτων **65**
Α΄.1 Πειράματα στο Matlab / GNU Octave 65

Β΄ Πηγαίος Κώδικας **71**

Β′.1 Κώδικας Matlab / GNU Octave 71
Β′.2 Κώδικας C# 76

Κεφάλαιο 1

Προαπαιτούμενες Γνώσεις

Στο κεφάλαιο αυτό εισάγουμε τις βασικές έννοιες γραμμικής άλγεβρας που είναι απαραίτητες για την κατανόηση του βασικού θέματος που πραγματεύεται η παρούσα εργασία, το οποίο είναι η αλγοριθμική εύρεση των μηδενικών ενός πολυωνυμικού πίνακα.

1.1 Στοιχεία Γραμμικής Άλγεβρας

Ορισμός 1. Έστω $\mathbb{R}^{n\times n}$ το σύνολο των πραγματικών $n\times n$ πινάκων και $A \in \mathbb{R}^{n\times n}$. Ένα μη μηδενικό διάνυσμα $x \in \mathbb{R}^{n\times 1}$ ονομάζεται ιδιοδιάνυσμα του πίνακα A με αντίστοιχη ιδιοτιμή λ αν

$$Ax = \lambda x$$

Το σύνολο $\{\lambda_i\}$ όλων των ιδιοτιμών του A ονομάζεται φάσμα και συμβολίζεται με $\sigma(A)$. Παρατηρούμε ότι η εξίσωση $Ax = \lambda x$ γράφεται ισοδύναμα ως $(A-\lambda I)x = 0$ από την οποία συμπεραίνουμε ότι το λ είναι ιδιοτιμή του πίνακα A ανν ο πίνακας $A - \lambda I$ είναι ιδιάζων, δηλαδή αν $\det(A - \lambda I) = 0$. Το πολυώνυμο ϕ βαθμού n

$$\varphi(\lambda) = \det(A - \lambda I)$$

ονομάζεται χαρακτηριστικό πολυώνυμο του πίνακα A. Το χαρακτηριστικό πολυώνυμο του πίνακα A μπορεί να γραφεί στη μορφή [7]

$$\phi(\lambda) = \det(A - \lambda I) = (-1)^n(\lambda^n + c_1\lambda^{n-1} + \cdots + c_{n-1}\lambda + c_n)$$

με

$$c_k = \frac{(-1)^k}{k!} \det \begin{pmatrix} t_1 & 1 & 0 & 0 & \cdots & 0 \\ t_2 & t_1 & 2 & 0 & \cdots & 0 \\ t_3 & t_2 & t_1 & 3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{k-1} & t_{k-2} & t_{k-3} & t_{k-4} & \cdots & k-1 \\ t_k & t_{k-1} & t_{k-2} & t_{k-3} & \cdots & t_1 \end{pmatrix}$$

όπου $t_k = \mathrm{tr}(A^k)$ και $k = 1, 2, \ldots, n$. Επιπλέον, μπορεί να αποδειχτεί ότι ο k συντελεστής του χαρακτηριστικού πολυωνύμου δίνεται από το άθροισμα των $k \times k$ διαγώνιων ελασσόνων του πίνακα A. Παρατηρούμε ότι

$$c_1 = -t_1, \quad c_2 = \frac{1}{2}(t_1^2 - t_2), \ldots, c_n = (-1)^n \det(A)$$

Δύο προβλήματα ιδιοτιμών $Ax = \lambda x$ και $By = \mu y$ έχουν ίδιο σύνολο ιδιοτιμών, δηλαδή $\sigma(A) = \sigma(B)$, αν και μόνο αν τα χαρακτηριστικά τους πολυώνυμα ϕ_A, ϕ_B έχουν τους συντελεστές τους ίσους έναν προς έναν. Αυτό σημαίνει ότι

$$\mathrm{tr}(A) = \mathrm{tr}(B), \quad \mathrm{tr}(A^2) = \mathrm{tr}(B^2), \ldots, \det(A) = \det(B)$$

Δοθέντων δύο πινάκων $A, B \in \mathbb{R}^{n\times n}$ ορίζουμε ως γενικευμένο πρόβλημα ιδιοτιμών το πρόβλημα της εύρεσης ζεύγους αριθμών α, β και διανυσμάτων $x \neq 0$ τέτοια ώστε να ικανοποιείται η σχέση

$$\alpha A x = \beta B x$$

Το σύνολο των αριθμών $\lambda = \beta/\alpha$ που ικανοποιούν την παραπάνω εξίσωση ονομάζονται γενικευμένες ιδιοτιμές και στην περίπτωση που $\alpha = 0$ παίρνουμε ως ιδιοτιμή το ∞.

Ορισμός 2. Θεωρούμε το διανυσματικό χώρο V και ορίζουμε ως εσωτερικό γινόμενο την απεικόνιση $(\cdot,\cdot) : V \times V \to \mathbb{R}$ για την οποία

1. $(x, x) \geq 0$ με την ισότητα να ισχύει ανν $x = 0$.
2. $(x, y) = (y, x)$ για κάθε $x, y \in V$.
3. $(ax + by, z) = a(x, z) + b(y, z)$ για κάθε $x, y, z \in V$ και κάθε $a, b \in \mathbb{R}$.

Ένας διανυσματικός χώρος V εφοδιασμένος με εσωτερικό γινόμενο ονομάζεται χώρος εσωτερικού γινομένου. Αν ισχύει ότι $(x_i, x_j) = 0$ για κάθε $i \neq j$, τότε το σύνολο αυτό $\{x_i\}$ (καθώς και τα στοιχεία) του ονομάζεται ορθογώνιο. Ένα ορθοκανονικό σύνολο διανυσμάτων είναι ένα ορθογώνιο σύνολο διανυσμάτων νόρμας ένα, δηλαδή $\|x_i\| = \sqrt{(x_i, x_i)} = 1$, δηλαδή

$$(x_i, x_j) = \delta_{ij}$$

Δοθέντος ορθογωνίου συνόλου διανυσμάτων $\{x_i\}$ είναι δυνατό να σχηματίσουμε ένα ορθοκανονικό σύνολο $\{\tilde{x}_i\}$ με χρήση του ορισμού

$$\tilde{x}_i = \frac{x_i}{\|x_i\|}, \quad i = 1, \ldots, n$$

Επιπλέον, ένας πίνακας $A \in \mathbb{R}^{n\times n}$ ονομάζεται ορθογώνιος αν οι στήλες του σχηματίζουν ορθοκανονικό σύνολο στον $\mathbb{R}^{n\times 1}$, δηλαδή ισχύει

$$A^{\mathsf{T}} A = I, \text{ δηλαδή } A^{-1} = A^{\mathsf{T}}$$

Ορισμός 3. Ένας πίνακας $D = (d_{ij}) \in \mathbb{R}^{n\times n}$ ονομάζεται διαγώνιος αν $d_{ij} = 0$ για κάθε $i \neq j$.

Ορισμός 4. Λέμε ότι ένας πίνακας $B \in \mathbb{R}^{n\times n}$ είναι όμοιος με έναν πίνακα $A \in \mathbb{R}^{n\times n}$ αν υπάρχει μη ιδιάζων πίνακας P τέτοιος ώστε

$$B = P^{-1}AP$$

Οι όμοιοι πίνακες έχουν ίδιο σύνολο ιδιοτιμών και ίσες ορίζουσες, δηλαδή αν $By = \lambda y$ και $B = P^{-1}AP$, τότε

$$P^{-1}APy = \lambda y \Leftrightarrow APy = \lambda Py \Leftrightarrow Ax = \lambda x, \quad x = Py$$

και

$$\det B = \det P^{-1}AP = \frac{1}{\det P} \det A \det P = \det A$$

Αν ο πίνακας A είναι όμοιος με έναν διαγώνιο πίνακα D, δηλαδή αν $D = P^{-1}AP$, τότε ο πίνακας A ονομάζεται διαγωνιοποιήσιμος.

Ορισμός 5. Ως μηδενοχώρο $\mathcal{N}(\cdot)$ ενός πίνακα $A \in \mathbb{R}^{m\times n}$ ορίζουμε το σύνολο των διανυσμάτων $x \in \mathbb{R}^{n\times 1}$ τα οποία επαληθεύουν την εξίσωση $Ax = 0$, δηλαδή το σύνολο

$$\mathcal{N}(A) = \left\{x \in \mathbb{R}^{n\times 1} \mid Ax = 0\right\}$$

Αν ο πίνακας A είναι αντιστρέψιμος, $\det A \neq 0$, τότε το ομογενές σύστημα $Ax = 0$ γράφεται

$$Ax = 0 \Leftrightarrow A^{-1}Ax = 0 \Leftrightarrow Ix = 0 \Leftrightarrow x = 0$$

Ο πίνακας

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}$$

του οποίου οι στήλες είναι γραμμικά εξαρτημένες, είναι ιδιάζων αφού $\det(A) = 0$. Σε αυτή την περίπτωση έχουμε

$$Ax = 0 \Leftrightarrow \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0 \Leftrightarrow x_1 + 2x_2 = 0 \Leftrightarrow x_1 = -2x_2$$

που σημαίνει ότι κάθε στοιχείο της μορφής $\begin{pmatrix} -2 & 1 \end{pmatrix}^{\mathsf{T}} x_2$ είναι στοιχείο του μηδενοχώρου $\mathcal{N}(A)$. Σε προγράμματα εκτέλεσης αριθμητκών υπολογισμών, από το σύνολο των διανυσμάτων $\begin{pmatrix} -2 & 1 \end{pmatrix}^{\mathsf{T}} x_2$ του μηδενοχώρου $\mathcal{N}(A)$ επιλέγεται ένα από εκείνα των οποίων το μέτρο είναι ένα, δηλαδή $\|x\| = 1$ ή

$$\|x\|^2 = 1 \Leftrightarrow 4x_2^2 + x_2^2 = 1 \Leftrightarrow x_2 = \pm\sqrt{\frac{1}{5}}, \quad x_1 = \mp\sqrt{\frac{4}{5}}$$

Η διάσταση του μηδενοχώρου $\mathcal{N}(A)$ ορίζεται ως μηδενικότητα του πίνακα A και γράφουμε $\text{nullity}(A)$, δηλαδή ισχύει

$$\text{nullity}(A) = \dim(\mathcal{N}(A))$$

Ως αριστερό μηδενοχώρο $\mathcal{K}(\cdot)$ ενός πίνακα $A \in \mathbb{R}^{m\times n}$ ορίζουμε το σύνολο των διανυσμάτων $x \in \mathbb{R}^{n\times 1}$ τα οποία επαληθεύουν την εξίσωση $x^{\mathsf{T}}A = 0^{\mathsf{T}}$, δηλαδή το σύνολο

$$\mathcal{K}(A) = \left\{x \in \mathbb{R}^{n\times 1} \mid x^{\mathsf{T}}A = 0^{\mathsf{T}}\right\}$$

Εφόσον $(x^{\mathsf{T}}A)^{\mathsf{T}} = A^{\mathsf{T}}x$ ο αριστερός μηδενοχώρος ενός πίνακα A ταυτίζεται με το μηδενοχώρο του ανάστροφου του, δηλαδή

$$\mathcal{K}(A) = \mathcal{N}(A^{\mathsf{T}})$$

Ορισμός 6. Ως χώρο στηλών $\mathcal{C}(\cdot)$ ενός πίνακα $A = (a_1, a_2, \ldots a_n) \in \mathbb{R}^{m\times n}$ ορίζουμε το σύνολο των γραμμικών συνδυασμών των στηλών του, δηλαδή

$$\mathcal{C}(A) = \left\{x \in \mathbb{R}^{m\times 1} \mid x = \sum_{i=1}^{n} \lambda_i a_i,\ \lambda_i \in \mathbb{R}\right\}$$

Μία βάση του χώρου στηλών ενός πίνακα προκύπτει αν μετασχηματίσουμε τον πίνακα σε κλιμακωτό. Επιπλέον ορίζουμε ως χώρο γραμμών ενός πίνακα A το χώρο στηλών του ανάστροφου του, δηλαδή $\mathcal{C}(A^{\mathsf{T}})$. Η διάσταση του χώρου στηλών του πίνακα A ονομάζεται τάξη $\text{rank}(A)$ του πίνακα, δηλαδή $\text{rank}(A) = \dim(\mathcal{C}(A))$.

Η τάξη και η μηδενικότητα ενός πίνακα A συνδέονται με το παρακάτω θεώρημα.

Θεώρημα 1. [1] Το άθροισμα της τάξης και της μηδενικότητας ενός πίνακα $A \in \mathbb{R}^{m\times n}$ ισούται με το πλήθος των στηλών του πίνακα, δηλαδή

$$\text{rank}(A) + \text{nullity}(A) = n.$$

Απόδειξη. Αν $\text{rank}(A) = n$, τότε η μοναδική λύση του γραμμικού συστήματος $Ax = 0$ είναι η τετριμμένη λύση $x = 0$, δηλαδή $\mathcal{N}(A) = \{0\}$ και η μηδενικότητα είναι $\text{nullity}(A) = 0$.

Υποθέτουμε ότι $r = \text{rank}(A) < n$. Στην περίπτωση αυτή, η λύση του γραμμικού συστήματος $Ax = 0$ εμφανίζει $n - r$ ελεύθερες μεταβλητές $c_1, \dots, c_{n-r}$. Θεωρούμε τις λύσεις $x_1, \dots, x_{n-r}$ του γραμμικού συστήματος $Ax = 0$ οι οποίες προκύπτουν αν θέσουμε την αντίστοιχη ελεύθερη μεταβλητή $c_i = 1$ και τις υπόλοιπες $c_{j \neq i} = 0$. Τα διανύσματα του συνόλου $\{x_1, \dots, x_{n-r}\}$ είναι γραμμικά ανεξάρτητα και επιπλέον κάθε λύση του γραμμικού συστήματος $Ax = 0$ γράφεται

$$x = c_1 x_1 + \cdots + c_{n-r} x_{n-r}$$

δηλαδή το σύνολο $\{x_1, \dots, x_{n-r}\}$ παράγει το μηδενοχώρο $\mathcal{N}(A)$. Τελικά το σύνολο $\{x_1, \dots, x_{n-r}\}$ αποτελεί βάση του μηδενοχώρου $\mathcal{N}(A)$ και η μηδενικότητα του A είναι $\text{nullity}(A) = n - r$. $\square$

1.2 Πολυώνυμα

Ορισμός 7. Μία απεικόνιση $p : \mathbb{R} \to \mathbb{R}$ ονομάζεται πολυώνυμο βαθμού m αν υπάρχουν συντελεστές $a_0, \dots, a_m \in \mathbb{R}$ με $a_m \neq 0$ τέτοιοι ώστε

$$p(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_m z^m$$

για κάθε $z \in \mathbb{R}$. Το σύνολο όλων των πολυωνύμων με συντελεστές στο σύνολο $\mathbb{R}$ συμβολίζεται ως $\mathcal{P}(\mathbb{R})$ και είναι διανυσματικός χώρος. Λέμε ότι ένας αριθμός $x \in \mathbb{R}$ είναι ρίζα ενός πολυωνύμου $p \in \mathcal{P}(\mathbb{R})$ βαθμού $m \geq 1$ (ή μηδενικό της εξίσωσης $p(z) = 0$) αν και μόνο αν υπάρχει πολυώνυμο $q \in \mathcal{P}(\mathbb{R})$ βαθμού $m - 1$ τέτοιο ώστε

$$p(z) = (z - x)q(z)$$

Σημειώνουμε ότι ένα πολυώνυμο $p \in \mathcal{P}(\mathbb{R})$ βαθμού $m \geq 0$ έχει m διακριτές ρίζες στο σύνολο $\mathbb{C}$ και κάθε μη σταθερό πολυώνυμο μιγαδικών συντελεστών έχει μία τουλάχιστον ρίζα.

Ορισμός 8. Κάθε πολυώνυμο $p \in \mathcal{P}(\mathbb{R})$ βαθμού $m \geq 1$ με τιμή

$$p(z) = a_0 + a_1 z + a_2 z^2 + \cdots + z^m$$

δηλαδή με συντελεστή μεγιστοβάθμιου όρου τη μονάδα, είναι χαρακτηριστικό πολυώνυμο του πίνακα

$$A_p = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{m-1} \end{pmatrix} \in \mathbb{R}^{m\times m}$$

δηλαδή $p(\lambda) = \det(A_p - \lambda I)$. Ο πίνακας A_p ονομάζεται συνοδεύων πίνακας (companion form) του πολυωνύμου p.

1.3 Περιγραφή της Ανάλυσης Ιδιαζόντων Τιμών

Η ανάλυση ιδιαζόντων τιμών (ΑΙΤ/SVD) ενός πίνακα $A \in \mathbb{R}^{m\times n}$ έχει τη μορφή

$$A = U\Sigma V^{\mathsf{T}}$$

όπου $U \in \mathbb{R}^{m\times m}$ και $V \in \mathbb{R}^{n\times n}$ είναι ορθογώνιοι πίνακες και $\Sigma \in \mathbb{R}^{m\times n}$ ο πίνακας με στοιχεία $\sigma_{ij} = 0$ για $i \neq j$ και $\sigma_i = \sigma_{ij} \geq 0$ για $i = j$. Τα στοιχεία της διαγωνίου του πίνακα Σ ονομάζονται ιδιάζουσες τιμές του πίνακα A, διατάσσονται έτσι ώστε $\sigma_i \geq \sigma_{i+1}$, $i = 1, \ldots, n-1$ και ορίζονται [4] από τις σχέσεις

$$Av = \sigma u, \quad A^{\mathsf{T}}u = \sigma v$$

Από τις παραπάνω σχέσεις ορισμού παίρνουμε

$$Av = \sigma u \Rightarrow A^{\mathsf{T}}Av = \sigma A^{\mathsf{T}}u \Rightarrow A^{\mathsf{T}}Av = \sigma^2 v$$

και

$$A^{\mathsf{T}}u = \sigma v \Rightarrow AA^{\mathsf{T}}u = \sigma Av \Rightarrow AA^{\mathsf{T}}u = \sigma^2 u$$

Έχουμε ότι τα αριστερά ιδιάζοντα διανύσματα u ενός πίνακα A είναι τα ιδιοδιανύσματα του πίνακα AA^{T}, δηλαδή $AA^{\mathsf{T}}u = \sigma^2 u$, και τα δεξιά ιδιάζοντα διανύσματα v του ίδιου πίνακα είναι τα ιδιοδιανύσματα του πίνακα $A^{\mathsf{T}}A$, δηλαδή $A^{\mathsf{T}}Av = \sigma^2 v$. Οι στήλες των πινάκων U και V είναι τα ιδιοδιανύσματα u_i, v_i για την αντίστοιχη ιδιάζουσα τιμή σ_i.

1.4 Βασικές Εφαρμογές της ΑΙΤ

Η ΑΙΤ έχει μεγάλο πλήθος εφαρμογών [3] μεταξύ των οποίων είναι ο υπολογισμός της Ευκλείδειας νόρμας $\|\cdot\|_2$ ενός πίνακα, ο υπολογισμός του καταστατικού του αριθμού (condition number) $\kappa(\cdot)$ και η εύρεση ορθοκανονικών βάσεων των θεμελιωδών υποχωρών του.

Ευκλείδεια Νόρμα

Η νόρμα ενός πίνακα $A \in \mathbb{R}^{m\times n}$ που επάγεται από την Ευκλείδεια διανυσματική νόρμα ορίζεται ως το ελάχιστο άνω φράγμα του πηλίκου $\|Ax\|_2/\|x\|_2$. Πιο συγκεκριμένα έχουμε

$$\|A\|_2 = \sup_{x\in\mathbb{R}^{n\times 1}\setminus\{0\}} \frac{\|Ax\|_2}{\|x\|_2} = \sigma_{\max}$$

όπου η μέγιστη ιδιάζουσα τιμή $\sigma_{\max}$ μπορεί να υπολογιστεί με χρήση της ΑΙΤ. Ως εφαρμογή των παραπάνω θεωρούμε τον πίνακα

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

του οποίου η ΑΙΤ είναι

$$U = \begin{pmatrix} -0.61541 & -0.78821 \\ -0.78821 & 0.61541 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 3.6993 & 0 & 0 \\ 0 & 1.1469 & 0 \end{pmatrix}$$

και

$$V = \begin{pmatrix} -5.9250 & 3.8593 & -7.0711 \\ -5.4579 & -8.3792 & 0 \\ -5.9250 & 3.8593 & 7.0711 \end{pmatrix} 10^{-1}$$

δηλαδή $A = U\Sigma V^{\mathsf{T}}$. Η μέγιστη ιδιάζουσα τιμή είναι $\sigma_{\max} = 3.6993$, δηλαδή $\|A\|_2 = 3.6993$.

Καταστατικός Αριθμός

Ο καταστατικός αριθμός $\kappa(\cdot)$ ενός πίνακα A ορίζεται με χρήση των ιδιαζόντων τιμών του από τη σχέση

$$\kappa(A) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

Ο ορισμός αυτός είναι σύμφωνος με τον ορισμό του καταστατικού αριθμού ενός τετραγωνικού πίνακα και αποτελεί γενίκευση για πίνακα αυθαίρετης διάστασης και τάξης. Σημειώνουμε ότι με χρήση του παραπάνω ορισμού έχουμε ότι $\kappa(A) = \infty$ αν $\operatorname{rank}(A) < \min(m, n)$, εφόσον σε αυτήν την περίπτωση έχουμε $\sigma_{\min} = 0$. Ο καταστατικός αριθμός πίνακα αυθαίρετου σχήματος και τάξης αποτελεί μέτρο της πληρότητας της τάξης του πίνακα.

Βάσεις Θεμελιωδών Υποχώρων

Μία ιδιαίτερα χρήσιμη εφαρμογή της ΑΙΤ είναι η εύρεση ορθοκανονικών βάσεων των θεμελιωδών υποχώρων ενός πίνακα. Από τους πίνακες U και V μπορούμε να εξαγάγουμε ορθοκανονικές βάσεις ενός πίνακα $A \in \mathbb{R}^{m\times n}$ ως ακολούθως.

1. Βάση μηδενοχώρου: τελευταίες $n - r$ στήλες του πίνακα V,

2. βάση αριστερού μηδενοχώρου: τελευταίες $m - r$ στήλες του πίνακα U,

3. βάση χώρου στηλών: πρώτες r στήλες του πίνακα U,

4. βάση χώρου στηλων ανάστροφου: πρώτες r στήλες του πίνακα V,

όπου $r = \text{rank}(A)$ η τάξη του πίνακα A.

1.5 Σχόλια στην Υλοποίηση της ΑΙΤ

Οι ιδιάζουσες τιμές και τα ιδιάζοντα διανύσματα συνδέονται με τις ιδιοτιμές και τα ιδιοδιανύσματα ενός πίνακα. Πιο συγκεκριμένα, οι ιδιάζουσες τιμές ενός πίνακα $A \in \mathbb{R}^{m \times n}$ είναι οι μη αρνητικές τετραγωνικές ρίζες των ιδιοτιμών του πίνακα $A^{\mathsf{T}}A$. Οι στήλες των πινάκων U και V είναι τα ορθοκανονικά ιδιοδιανύσματα των πινάκων AA^{T} και $A^{\mathsf{T}}A$ αντίστοιχα. Οι ευσταθείς αλγόριθμοι για τον υπολογισμό της ΑΙΤ χρησιμοποιούν τον πίνακα A, εργάζονται δηλαδή χωρίς να σχηματίζουν κάποιον από τους πίνακες AA^{T} και $A^{\mathsf{T}}A$. Με αυτόν τον τρόπο αποφεύγουν την απώλεια πληροφοριών που ενέχει ο σχηματισμός των παραπάνω πινάκων.

Για πλήρεις πίνακες, η συνήθης προσέγγιση στον υπολογισμό της ΑΙΤ [4] χρησιμοποιεί μία παραλλαγή των επαναλήψεων QR. Αρχικά ο πίνακας A μετατρέπεται σε διδιαγώνιο με εφαρμογή πεπερασμένου πλήθους ορθογώνιων μετασχηματισμών. Έπειτα, τα υπολειπόμενα μη διαγώνια στοιχεία μηδενίζονται επαναληπτικά. Η ΑΙΤ μπορεί επίσης να υπολογιστεί με μία παραλλαγή της μεθόδου Jacobi η οποία παρουσιάζει ενδιαφέρον για την παράλληλη υλοποίηση καθώς επίσης και στην περίπτωση που ο πίνακας A έχει κάποια ιδιαίτερη δομή.

Ο συνολικός αριθμός πράξεων που απαιτείται για τον υπολογισμό της ΑΙΤ είναι ανάλογος της ποσότητας $mn^2 + n^3$ με σταθερά αναλογίας η οποία εξαρτάται από τον αλγόριθμο θα χρησιμοποιηθεί. Αν το πλήθος των στοιχείων του πίνακα A είναι μεγάλο αλλά ο πίνακας σποραδικός, τότε η διδιαγωνιποίηση πραγματοποιείται πιο αποτελεσματικά με χρήση μίας παραλλαγής του αλγόριθμου Lanczos η οποία είναι κατάλληλη στην περίπτωση που επιθυμούμε σχετικά μικρό πλήθος ιδιαζόντων τιμών και αντίστοιχων ιδιαζόντων διανυσμάτων.

Σχόλια στη Βιβλιογραφία

Επιπλέον βιβλία που χρησιμοποιήθηκαν για τη συγγραφή του κεφαλαίου αυτού είναι τα [1], [2] και [3].

Κεφάλαιο 2

Ανάλυση Ιδιαζόντων Τιμών

Ακολουθεί μία σύντομη παρουσίαση της ΑΙΤ καθώς και της ανάλυσης QR σε θεωρητικό επίπεδο καθώς και η κατασκευή ενός αλγόριθμου για τον υπολογισμό της ΑΙΤ ενός πίνακα με χρήση της ανάλυσης QR.

2.1 Ιστορικά Στοιχεία

Οι γεωμέτρες και οι αλγεβριστές του 19ου αιώνα ήθελαν να γνωρίζουν [5] πότε δύο πραγματικές διγραμμικές μορφές ϕ, ω είναι ισοδύναμες υπό ανεξάρτητες πραγματικές ορθογώνιες αντικαταστάσεις, δηλαδή $\phi(x, y) = \omega(Q_1 x, Q_2 y)$ για κάθε $x, y \in \mathbb{R}^n$. Ο E. Beltrami (1973) απέδειξε ότι υπάρχουν πάντα πραγματικοί ορθογώνιοι πίνακες Q_1, Q_2 τέτοιοι ώστε ο πίνακας

$$Q_1^{\mathsf{T}} A Q_2 = \Sigma = \operatorname{diag}(\sigma_1(A), \ldots, \sigma_n(A))$$

να είναι μη αρνητικός και διαγώνιος, όπου $\sigma_1^2(A) \geq \cdots \geq \sigma_n^2(A)$ είναι οι ιδιοτιμές του πίνακα AA^{T} καθώς και του $A^{\mathsf{T}}A$. Επιπλέον, ο Beltrami έδειξε ότι οι ορθοκανονικές στήλες των Q_1 και Q_2 είναι ιδιοδιανύσματα των πινάκων AA^{T} και $A^{\mathsf{T}}A$.

Ο C. Jordan κατέληξε στο ίδιο συμπέρασμα με τον Beltrami αλλά μελετώντας το πρόβλημα από διαφορετική οπτική. Βρήκε ότι οι ιδιοτιμές ενός $2n \times 2n$ πραγματικού και συμμετρικού πίνακα

$$\begin{pmatrix} \mathbb{O} & A \\ A^{\mathsf{T}} & \mathbb{O} \end{pmatrix}$$

εμφανίζονται ανά ζεύγη προσήμων και ότι οι n μέγιστες ιδιοτιμές είναι οι επιθυμητοί συντελεστές $\sigma_1(A), \ldots, \sigma_n(A)$ της κανονικής μορφής

$$\phi_{Q_1^{\mathsf{T}} A Q_2}(\xi, \eta) = \sum_{i=1}^{n} \sigma_i(A) \xi_i \eta_i$$

Ο J. J. Sylvester (1889, 1890) απέδειξε χωρίς γνώση της δουλειάς των Beltrami και Jordan την ανάλυση $Q_1^{\mathsf{T}} A Q_2$. Ο Sylvester ονόμασε τις ιδιοτιμές $\sigma_i(A)$ κανονικούς πολλαπλασιαστές της διγραμμικής μορφής $\phi(x, y)$.

Ο L. Autonne (1902) έδειξε ότι κάθε μη ιδιάζων μιγαδικός πίνακας A μπορεί να αναλυθεί ως $A = UP$, όπου ο πίνακας U είναι ορθογώνιος και ο P θετικά ορισμένος. Ο Autonne επανήλθε στις ιδέες του (1913 – 1915) και χρησιμοποίησε το γεγονός ότι οι πίνακες A^*A και AA^* είναι όμοιοι ώστε να δείξει ότι κάθε μιγαδικός τετραγωνικός πίνακας μπορεί να γραφεί ως $A = V\Sigma W^*$, όπου οι πίνακες V, W είναι ορθογώνιοι και ο πίνακας Σ είναι μη αρνητικά ορισμένος (θετικά ημιορισμένος) διαγώνιος. Το 1930 έγινε αναφορά στη δουλειά του Autonne από τον Browne ο οποίος χρησιμοποίησε την ανάλυση $A = V\Sigma W^*$ για να αποδείξει ανισότητες για τη φασματική ακτίνα ερμιτιανών και γενικών τετραγωνικών πινάκων.

Μετά από τις εργασίες των Williamson και Murnaghan (1930, 1935) σχετικά με την πολική ανάλυση οι Eckart και Young (1939) έδωσαν μία πλήρη και ξεκάθαρη δήλωση της ανάλυσης ιδιαζόντων τιμών για μιγαδικούς πίνακες. Ο όρος ιδιάζουσες τιμές χρησιμοποιήθηκε για πρώτη φορά από τον Pickard (1910) ο οποίος ανέπτυξε τη θεωρία του Schmidt για τους πυρήνες

$$\phi(s) = \lambda \int_a^b K(s,t)\psi(t)\mathrm{d}t, \quad \psi(s) = \lambda \int_a^b K(s,t)\phi(t)\mathrm{d}t$$

και θεώρησε ότι η διπλή χρήση του όρου ¨ιδιοτιμή¨ από τον Schmidt προκαλούσε η σύγχυση.

2.2 Ύπαρξη της ΑΙΤ

Ακολουθεί το θεώρημα της ύπαρξης της ΑΙΤ για πίνακες πραγματικών στοιχείων καθώς και η απόδειξη αυτού.

Θεώρημα 2. Για κάθε $m \times n$ πίνακα A υπάρχει η ανάλυση ιδιαζόντων τιμών, δηλαδή $A = U\Sigma V^{\mathsf{T}}$, όπου $U \in \mathbb{R}^{m\times m}$ και $V \in \mathbb{R}^{n\times n}$ είναι οι ορθογώνιοι πίνακες με στήλες τα αριστερά και δεξιά ιδιάζοντα διανύσματα και $\Sigma \in \mathbb{R}^{m\times n}$ ο πίνακας των ιδιαζόντων τιμών, [4], [3].

Απόδειξη. Εφόσον ο πίνακας $A^{\mathsf{T}}A$ είναι συμμετρικός $n \times n$ πίνακας, τότε οι ιδιοτιμές του είναι όλες πραγματικές και μη αρνητικές και επιπλέον υπάρχει η διαγωνιοποίηση του με ορθογώνιο πίνακα V. Έστω λ κάποια ιδιοτιμή του πίνακα

$A^\mathsf{T}A$ και x το αντίστοιχο ιδιοδιάνυσμα, δηλαδή

$$A^\mathsf{T}Ax = \lambda x \Leftrightarrow x^\mathsf{T}A^\mathsf{T}Ax = \lambda x^\mathsf{T}x \Leftrightarrow \|Ax\|^2 = \lambda\|x\|^2 \Leftrightarrow \lambda = \frac{\|Ax\|^2}{\|x\|^2} \geq 0$$

Υποθέτουμε ότι οι στήλες του πίνακα V έχουν διαταχθεί έτσι ώστε οι αντίστοιχες ιδιοτιμές ικανοποιούν τη σχέση

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$$

Οι ιδιάζουσες τιμές του πίνακα A δίνονται από τη σχέση $\sigma_i = \sqrt{\lambda_i}$, $i = 1, \ldots, n$. Έστω r η τάξη του πίνακα A, τότε r θα είναι και η τάξη του πίνακα $A^\mathsf{T}A$ και εφόσον ο τελευταίος είναι συμμετρικός, η τάξη του θα ισούται με το πλήθος των μη μηδενικών ιδιοτιμών του, δηλαδή

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \geq 0 \text{ και } \lambda_{r+1} = \lambda_{r+2} = \ldots = \lambda_n = 0$$

Η ίδια σχέση ισχύει για τις ιδιάζουσες τιμές

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0 \text{ και } \sigma_{r+1} = \sigma_{r+2} = \ldots = \sigma_n = 0$$

Έστω $V = \begin{pmatrix} V_1 & V_2 \end{pmatrix}$ με $V_1 = (v_1, \ldots, v_r)$, $V_2 = (v_{r+1}, \ldots, v_n)$ και

$$\Sigma_1 = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sigma_r \end{pmatrix}$$

Ο Σ_1 είναι ένας $r \times r$ διαγώνιος πίνακας με διαγώνια στοιχεία τις μη μηδενικές ιδιάζουσες τιμές σ_i. Ο $m \times n$ πίνακας Σ γράφεται

$$\Sigma = \begin{pmatrix} \Sigma_1 & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{pmatrix}$$

Οι στήλες του V_1 είναι τα μη μηδενικά ιδιοδιανύσματα του πίνακα $A^\mathsf{T}A$ και οι στήλες του V_2 είναι ιδιοδιανύσματα του πίνακα $A^\mathsf{T}A$ με ιδιοτιμή $\lambda = 0$

$$A^\mathsf{T}Av_i = \lambda v_i, \quad i = r+1, \ldots, n$$

δηλαδή οι στήλες του V_2 σχηματίζουν ορθοκανονική βάση του μηδενοχώρου $\mathcal{N}(A^\mathsf{T}A) = \mathcal{N}(A)$,

$$AV_2 = 0$$

Εφόσον ο V είναι ορθογώνιος πίνακας έχουμε ότι

$$I = VV^{\mathsf{T}} = V_1V_1^{\mathsf{T}} + V_2V_2^{\mathsf{T}}, \quad A = AI = AV_1V_1^{\mathsf{T}} + AV_2V_2^{\mathsf{T}} = AV_1V_1^{\mathsf{T}}$$

Για την ολοκλήρωση της απόδειξης πρέπει να δείξουμε ότι μπορεί να κατασκευαστεί ορθογώνιος $m \times m$ πίνακας U τέτοιος ώστε $A = U\Sigma V^{\mathsf{T}}$ ή ισοδύναμα $AV = U\Sigma$. Για το σκοπό αυτό ορίζουμε

$$u_i = \frac{1}{\sigma_i}Av_i, \quad i = 1, \ldots, r, \quad U_1 = (u_1, \ldots, u_r)$$

και έχουμε ότι

$$AV_1 = U_1\Sigma_1$$

Οι στήλες του σχηματίζουν ορθοκανονικό σύνολο εφόσον

$$u_i^{\mathsf{T}}u_j = \frac{1}{\sigma_i}v_i^{\mathsf{T}}A^{\mathsf{T}}\frac{1}{\sigma_j}Av_j = \frac{1}{\sigma_i\sigma_j}v_i^{\mathsf{T}}(A^{\mathsf{T}}Av_j) = \frac{\sigma_j}{\sigma_i}v_i^{\mathsf{T}}v_j = \delta_{ij}$$

όπου $1 \leq i \leq r$ και $1 \leq j \leq r$. Επιπλέον, κάθε u_i, $1 \leq i \leq r$, ανήκει στο χώρο $\mathcal{C}(A)$ του οποίου η διάσταση είναι r, δηλαδή οι στήλες του πίνακα U_1 σχηματίζουν ορθοκανονική βάση του $\mathcal{C}(A)$. Ο διανυσματικός χώρος $\mathcal{C}(A)^{\perp} = \mathcal{N}(A^{\mathsf{T}})$ είναι διάστασης $m - r$. Θέτουμε

$$U_2 = (u_{r+1}, \ldots, u_m), \quad U = \begin{pmatrix} U_1 & U_2 \end{pmatrix}$$

Ο πίνακας U είναι ορθογώνιος και τελικά έχουμε την ισότητα

$$U\Sigma V^{\mathsf{T}} = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{pmatrix} \begin{pmatrix} V_1^{\mathsf{T}} \\ V_2^{\mathsf{T}} \end{pmatrix} = U_1\Sigma_1V_1^{\mathsf{T}} = AV_1V_1^{\mathsf{T}} = A$$

□

2.3 Ανάλυση QR

Παραθέτουμε χωρίς απόδειξη το θεώρημα ύπαρξης της ανάλυσης QR για πίνακες πραγματικών στοιχείων

Θεώρημα 3. [3] Κάθε πίνακας $A \in \mathbb{R}^{m\times n}$ του οποίου οι στήλες είναι γραμμικά ανεξάρτητες μπορεί να παραχθεί από τον πολλαπλασιασμό ενός ορθογώνιου πίνακα $Q \in \mathbb{R}^{m\times m}$ με έναν άνω τριγωνικό πίνακα $R \in \mathbb{R}^{m\times n}$, δηλαδή

$$A = QR$$

Στη γενική περίπτωση ο $m \times n$ πίνακας A με $m \geq n$ γράφεται ως γινόμενο ενός ορθογώνιου $m \times m$ πίνακα Q με έναν $m \times n$ άνω τριγωνικό πίνακα R. Επειδή τα στοιχεία των τελευταίων $m - n$ γραμμών του πίνακα R είναι μηδενικά μπορούμε να διαμερίσουμε τους εμπλεκόμενους πίνακες ως ακολούθως.

$$A = QR = Q \begin{pmatrix} R_1 \\ \mathbb{O} \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ \mathbb{O} \end{pmatrix} = Q_1 R_1$$

όπου οι διαστάσεις των πινάκων Q_1, Q_2, R_1 είναι $m \times n$, $m \times (m - n)$ και $n \times n$ αντίστοιχα. Η ανάλυση $Q_1 R_1$ ονομάζεται λεπτή (thin) ανάλυση QR. Αν ο πίνακας A είναι πλήρους τάξης n και επιπλέον απαιτήσουμε τα διαγώνια στοιχεία του άνω τριγωνικού πίνακα R_1 να είναι θετικά (θετικά ορισμένος πίνακας), τότε μπορεί να αποδειχτεί ότι οι πίνακες Q_1 και R_1 είναι μοναδικοί και ο πίνακας R_1 ισούται με τον άνω τριγωνικό παράγοντα L^T της ανάλυσης Cholesky $A = LL^\mathsf{T}$.

2.4 Ανάλυση QR και Πίνακας Householder

Παρουσιάζουμε τη μεθοδολογία εύρεσης της ανάλυσης QR με χρήση πινάκων Householder.

Ορισμός 9. Αν $v \in \mathbb{R}^{m \times 1}$ είναι ένα μη μηδενικό διάνυσμα, ορίζουμε [4] ως πίνακα Householder τον $m \times m$ πίνακα

$$H = I_m - \frac{2}{v^\mathsf{T} v} v v^\mathsf{T}$$

Αν το διάνυσμα v είναι κανονικοποιημένο, δηλαδή αν $\|v\| = 1$, τότε το εσωτερικό γινόμεννο $v^\mathsf{T} v = 1$ και ο πίνακας Householder γράφεται στη μορφή

$$H = I_m - 2vv^\mathsf{T}$$

Ορίζουμε το διάνυσμα $u = x + ae_1$ όπου $x = \begin{pmatrix} x_1 & x_2 & \cdots & x_m \end{pmatrix}^\mathsf{T}$ και $e_1 = I_m(:, 1)$ και

$$a = \begin{cases} -\operatorname{sign}(x_1)\|x\| & , \quad x \neq 0 \\ 1 & , \quad x = 0 \end{cases}$$

και το κανονικοποιούμε, δηλαδή κατασκευάζουμε το διάνυσμα $v = u/\|u\|$ για το οποίο $\|v\| = 1$. Για το γινόμενο του πίνακα $H = I_m - 2vv^\mathsf{T}$ με το αρχικό διάνυσμα x έχουμε

$$Hx = \begin{pmatrix} a & 0 & \cdots & 0 \end{pmatrix}^\mathsf{T}$$

Εφαρμογή 1. Έστω το διάνυσμα $x = \begin{pmatrix} 1 & 2 & 2 \end{pmatrix}^{\mathsf{T}}$. Θέτουμε

$$u = x + ae_1 = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} - \|x\| \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \\ 2 \end{pmatrix}$$

Για να κανονικοποιήσουμε το διάνυσμα u υπολογίζουμε τη νόρμα του, $\|u\| = 2\sqrt{3}$. Έπειτα ορίζουμε το κανονικοποιημένο διάνυσμα

$$v = \frac{u}{\|u\|} = \frac{1}{\sqrt{3}} \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

το οποίο είναι το ζητούμενο διάνυσμα Householder. Ο πίνακας Householder είναι

$$H = I_3 - 2vv^{\mathsf{T}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \frac{2}{3} \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 & -1 \end{pmatrix}$$

ή

$$H = \frac{1}{3} \begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ 2 & -2 & 1 \end{pmatrix}$$

και παρατηρούμε ότι $Hx = \begin{pmatrix} 3 & 0 & 0 \end{pmatrix}^{\mathsf{T}}$.

Οι πίνακες Householder μπορούν να χρησιμοποιηθούν [4] για την εύρεση της ανάλυσης QR ενός πίνακα A,

$$A = QR, \quad QQ^{\mathsf{T}} = I, \quad R = (r_{ij}) \text{ με } r_{ij} = 0 \text{ όταν } i > j$$

Αναζητάμε τον πίνακα Householder $\overline{H}_1 \in \mathbb{R}^{m \times m}$ ο οποίος μετασχηματίζει την πρώτη στήλη a_1 του πίνακα A ως

$$\overline{H}_1 a_1 = \overline{a}_1$$

Αν απαιτούνται n εφαρμογές της παραπάνω διαδικασίας θέτουμε $H_n = \operatorname{diag}(I_{n-1}, \overline{H}_n)$ ώστε ο πίνακας A να μετασχηματιστεί σε άνω τριγωνικό, τότε θα έχουμε

$$H_n H_{n-1} \cdots H_1 A = R, \quad Q = H_1 \cdots H_{n-1} H_n, \quad A = QR$$

όπου για την εύρεση του πίνακα Q χρησιμοποιήσαμε τη σχέση

$$H^{-1} = H^{\mathsf{T}} = H$$

η οποία είναι αληθής επειδή οι πίνακες Householder είναι ορθογώνιοι και συμμετρικοί. Ο πίνακας Q είναι ορθογώνιος ως γινόμενο ορθογωνίων πινάκων.

Εφαρμογή 2. Έστω ο πίνακας

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{pmatrix}$$

του οποίου αναζητάμε την ανάλυση QR με χρήση μετασχηματισμών Householder. Από την προηγούμενη εφαρμογή έχουμε ότι

$$H_1 = \overline{H}_1 = \frac{1}{3} \begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ 2 & -2 & 1 \end{pmatrix}$$

αφού

$$H_1 \begin{pmatrix} 1 & 2 & 2 \end{pmatrix}^\mathsf{T} = \begin{pmatrix} 3 & 0 & 0 \end{pmatrix}^\mathsf{T}$$

Πολλαπλασιάζοντας τον πίνακα A με τον πίνακα H_1 παίρνουμε

$$H_1 A = \frac{1}{3} \begin{pmatrix} 9 & 8 \\ 0 & 4 \\ 0 & 1 \end{pmatrix}$$

Έπειτα αναζητάμε τον πίνακα $\overline{H}_2$ για το διάνυσμα $x = \begin{pmatrix} 4 & 1 \end{pmatrix}^\mathsf{T}$. Ορίζουμε το διάνυσμα $u = x - \|x\| e_1 = \begin{pmatrix} 4 - \sqrt{17} & 1 \end{pmatrix}^\mathsf{T}$ το οποίο και κανονικοποιούμε, δηλαδή $v = u/\|u\|$. Κατασκευάζουμε τον πίνακα Householder $\overline{H}_2 = I_2 - 2vv^\mathsf{T}$ τον οποίο υπολογίζουμε αριθμητικά

$$\overline{H}_2 = \begin{pmatrix} 0.97014 & 0.24254 \\ 0.24254 & 0.97014 \end{pmatrix}$$

και σχηματίζουμε τον πίνακα $H_2 = \mathrm{diag}(I_1, \overline{H}_2)$,

$$H_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.97014 & 0.24254 \\ 0 & 0.24254 & 0.97014 \end{pmatrix}$$

τον οποίο και δρούμε στον πίνακα $H_1 A$, δηλαδή

$$H_2 H_1 A = \frac{1}{3} \begin{pmatrix} 9 & 8 \\ 0 & 4.12311 \\ 0 & 0 \end{pmatrix} = R$$

Ο πίνακας Q είναι H_1H_2, δηλαδή

$$Q = H_1H_2 = \frac{1}{3}\begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ 2 & -2 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.97014 & 0.24254 \\ 0 & 0.24254 & 0.97014 \end{pmatrix}$$

ή

$$Q = \begin{pmatrix} 0.33333 & 0.80845 & -0.48507 \\ 0.66667 & 0.16169 & 0.72761 \\ 0.66667 & -0.56592 & -0.48507 \end{pmatrix}$$

2.5 Αλγόριθμος Υπολογισμού της ΑΙΤ

Στην παράγραφο αυτή παρουσιάζουμε έναν απλό αλγόριθμο για τον υπολογισμό της ΑΙΤ, τον οποίο και υλοποιούμε, θεωρώντας ότι έχουμε στη διάθεση μας μία συνάρτηση qr η οποία υλοποιεί κάποια μέθοδο ανάλυσης QR.

Αρχικά θέτουμε τους πίνακες U, V ίσους με τους μοναδιαίους πίνακες κατάλληλων διαστάσεων, υπολογίζουμε την ανάλυση QR του πίνακα $A(= QR)$ του οποίου αναζητάμε την ΑΙΤ και θέτουμε $U = UQ$. Έπειτα υπολογίζουμε την ανάλυση QR του πίνακα R^{T} και θέτουμε $V = VQ$. Επαναλαμβάνουμε την ίδια διαδικασία για τον πίνακα R^{T}. Η επαναληπτική αυτή διαδικασία τερματίζεται όταν η νόρμα του διανύσματος r_{u} με στοιχεία r_{ij} του πίνακα R για τα οποία $i < j$ είναι αγνοήσιμη σε σχέση με τη νόρμα του διανύσματος r_{d} με στοιχεία r_{ii}, δηλαδή

$$\|r_{\mathrm{u}}\| \ll \|r_{\mathrm{d}}\| \Leftrightarrow \|r_{\mathrm{u}}\| = \epsilon\|r_{\mathrm{d}}\| \Leftrightarrow \|r_{\mathrm{u}}\|/\|r_{\mathrm{d}}\| = \epsilon, \quad \epsilon \ll 1,$$

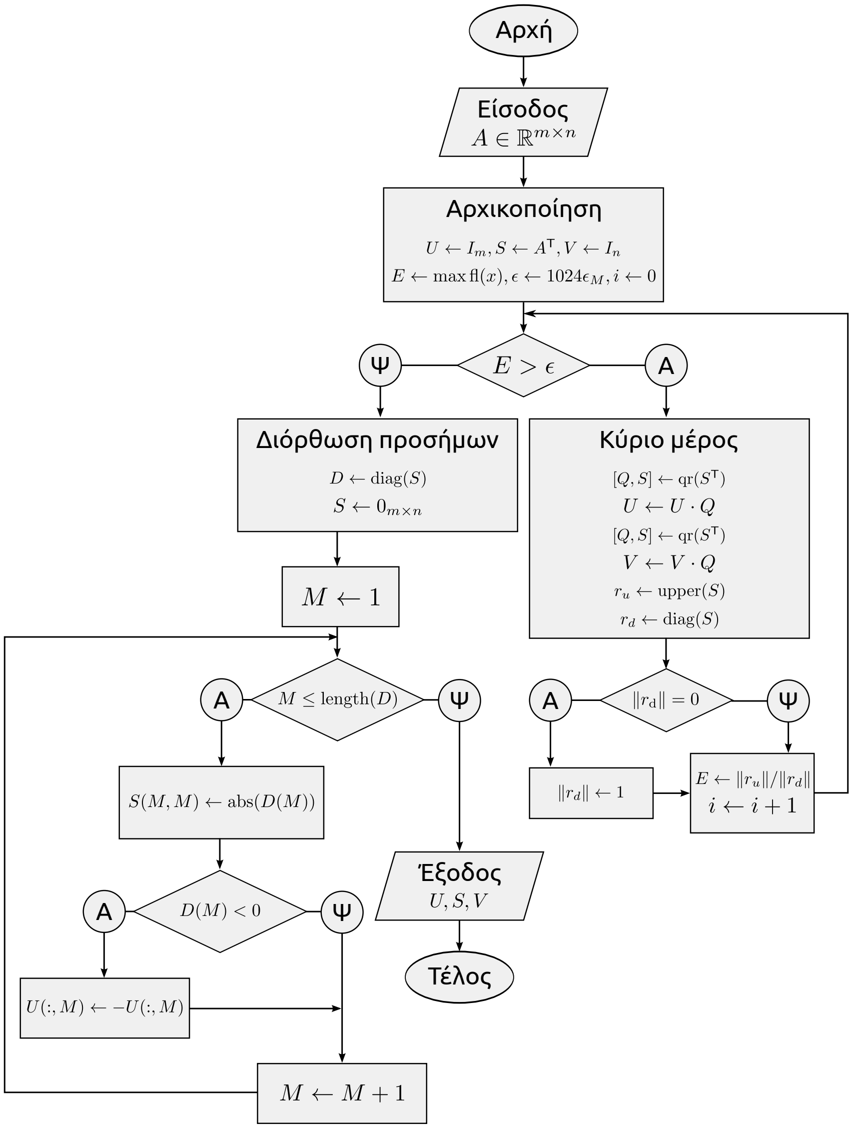
όπου ο αριθμός ανοχής ϵ καθορίζει την ακρίβεια των αποτελεσμάτων και απαιτούμε τουλάχιστον ένα από τα στοιχεία r_{ii} να είναι διάφορο του μηδενός ώστε $\|r_{\mathrm{d}}\| \neq 0$. Αν $\|r_{\mathrm{d}}\| = 0$ θέτουμε $\|r_{\mathrm{d}}\| = 1$.

Αλγόριθμος 1 (ΑΙΤ με χρήση QR)**.** Δοθέντος πίνακα $A \in \mathbb{R}^{m\times n}$ και αλγόριθμου

QR ο παρακάτω αλγόριθμος υπολογίζει την ΑΙΤ του πίνακα A.

συνάρτηση $[U, S, V] = \mathsf{qrbsvd}(A)$
$U = I_m,\ S = A^{\mathsf{T}},\ V = I_n,\ E = 10^{308},\ \epsilon = 1024\epsilon_M,\ i = 0$
όσο $E > \epsilon$
$[Q, S] = \mathrm{qr}(S^{\mathsf{T}})$
$U = UQ$
$[Q, S] = \mathrm{qr}(S^{\mathsf{T}})$
$V = VQ$
$r_{\mathrm{u}} = \mathrm{upper}(S),\ r_{\mathrm{d}} = \mathrm{diag}(S)$
$E = \|r_{\mathrm{u}}\|/\|r_{\mathrm{d}}\|$
$i = i + 1$
τέλος όσο
διόρθωση προσήμων
τέλος συνάρτησης

Επιπλέον, για να είναι σωστά τα πρόσημα των στοιχείων του πίνακα S υπεισέρχεται η ρουτίνα διόρθωσης προσήμων κατά την οποία κάθε αρνητικό στοιχείο της κύριας διαγωνίου του πίνακα S μετατρέπεται σε θετικό με ταυτόχρονη αλλαγή των προσήμων των στοιχείων της αντίστοιχης στήλης του πίνακα U ώστε να παραμένουν σωστά και τα πρόσημα των στοιχείων του πίνακα A. Ακολουθεί αναλυτικό διάγραμμα ροής του αλγόριθμου αυτού.



Εφαρμογή 3. Στο ακόλουθο παράδειγμα υπολογίζουμε την ΑΙΤ του πίνακα

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 2 \\ 2 & 1 \end{pmatrix}$$

με χρήση των ορισμών της θεωρίας της ΑΙΤ, καθώς και με τον αλγόριθμο που παρουσιάσαμε. Για να υπολογίσουμε τις ιδιάζουσες τιμές του πίνακα A βρίσκουμε τις ιδιοτιμές του πίνακα

$$A^{\mathsf{T}}A = \begin{pmatrix} 9 & 8 \\ 8 & 9 \end{pmatrix}$$

Οι ιδιοτιμές του πίνακα $A^\mathsf{T} A$ είναι οι ρίζες του χαρακτηριστικού του πολυωνύμου, δηλαδή

$$\det\left(A^\mathsf{T} A - \lambda I\right) = 0 \Leftrightarrow \lambda^2 - 18\lambda + 17 = 0 \Leftrightarrow \{\lambda_1 = 17, \lambda_2 = 1\}$$

Άρα οι ιδιάζουσες τιμές είναι $\sigma_1 = \sqrt{17}$ και $\sigma_2 = 1$ και ο πίνακας Σ γράφεται

$$\Sigma = \begin{pmatrix} \sqrt{17} & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Τα αντίστοιχα των ιδιοτιμών λ_1 και λ_2 ιδιοδιανύσματα είναι

$$v_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}^\mathsf{T}, \quad v_2 = \begin{pmatrix} 1 & -1 \end{pmatrix}^\mathsf{T}$$

όπου $\|v_1\| = \|v_2\| = \sqrt{2}$, δηλαδή ο πίνακας V είναι

$$V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Οι πρώτες δύο στήλες του πίνακα U μπορούν να υπολογιστούν από τη σχέση $Av = \sigma u$, δηλαδή

$$u_1 = \frac{1}{\sqrt{17}} A v_1, \quad u_2 = A v_2$$

Με εκτέλεση των πράξεων παίρνουμε

$$u_1 = \frac{1}{\sqrt{34}} \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}, \quad u_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Η τρίτη στήλη του πίνακα U μπορεί να υπολογιστεί από τη συνθήκη ορθοκανονικότητας

$$u_i^\mathsf{T} u_3 = \delta_{i3}, \quad i \in \{1, 2, 3\}$$

η οποία ικανοποιείται από το διάνυσμα

$$u_3 = \frac{1}{\sqrt{17}} \begin{pmatrix} 2 \\ -3 \\ 2 \end{pmatrix}$$

Τελικά, ο πίνακας U είναι

$$U = \begin{pmatrix} \frac{3}{\sqrt{34}} & -\frac{1}{\sqrt{2}} & \frac{2}{\sqrt{17}} \\ \frac{4}{\sqrt{34}} & 0 & -\frac{3}{\sqrt{17}} \\ \frac{3}{\sqrt{34}} & \frac{1}{\sqrt{2}} & \frac{2}{\sqrt{17}} \end{pmatrix}$$

Ακολουθεί μία επανάληψη του βρόγχου του αλγόριθμου υπολογισμού της ΑΙΤ κατά την οποία εκτελούμε την ανάλυση QR στο Matlab. Θέτουμε $U = I_3$ και $V = I_2$ και υπολογίζουμε την ανάλυση QR του πίνακα $S^{\mathsf{T}} = A$,

$$Q = \begin{pmatrix} -0.33333 & 0.80845 & 0.48507 \\ -0.66667 & 0.16169 & -0.72761 \\ -0.66667 & -0.56592 & 0.48507 \end{pmatrix}, \quad S = \begin{pmatrix} -3.00000 & -2.66667 \\ 0.00000 & 1.37437 \\ 0.00000 & 0.00000 \end{pmatrix}$$

και θέτουμε $U = UQ = Q$, δηλαδή

$$U = \begin{pmatrix} -0.33333 & 0.80845 & 0.48507 \\ -0.66667 & 0.16169 & -0.72761 \\ -0.66667 & -0.56592 & 0.48507 \end{pmatrix}$$

Υπολογίζουμε την ανάλυση QR του πίνακα S^{T} από την οποία λαμβάνουμε

$$Q = \begin{pmatrix} -0.74741 & -0.66436 \\ -0.66436 & 0.74741 \end{pmatrix}, \quad S = \begin{pmatrix} 4.01386 & -0.91308 & 0.00000 \\ 0.00000 & 1.02722 & 0.00000 \end{pmatrix}$$

και στη συνέχεια θέτουμε $V = VQ = Q$, δηλαδή

$$V = \begin{pmatrix} -0.74741 & -0.66436 \\ -0.66436 & 0.74741 \end{pmatrix}$$

Ορίζουμε τα διανύσματα

$$r_u = \begin{pmatrix} -0.91308 & 0.00000 & 0.00000 \end{pmatrix}^{\mathsf{T}}, \quad r_d = \begin{pmatrix} 4.0139 & 1.0272 \end{pmatrix}^{\mathsf{T}}$$

και υπολογίζουμε τη νόρμα τους,

$$\|r_u\| = 0.91308, \quad \|r_d\| = 4.1432$$

Τέλος, υπολογίζουμε το πηλίκο $E = \|r_u\|/\|r_d\| = 0.22038$ και επαναλαμβάνουμε όλη τη διαδικασία όσο η συνθήκη $E > \epsilon$ παραμένει αληθής. Τελικά η διαδικασία διακόπτεται μετά από έντεκα επαναλήψεις όταν το πηλίκο E λαμβάνει την τιμή $1.1004 \cdot 10^{-13}$ και ο αλγόριθμος επιστρέφει τους πίνακες

$$U = \frac{1}{10}\begin{pmatrix} -5.1450 & 7.0711 & 4.8507 \\ -6.8599 & \frac{8.2466}{10^{13}} & -7.2761 \\ -5.1450 & -7.0711 & 4.8507 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 4.12311 & 0.00000 \\ 0.00000 & 1.00000 \\ 0.00000 & 0.00000 \end{pmatrix}$$

και

$$V = \begin{pmatrix} -0.70711 & -0.70711 \\ -0.70711 & 0.70711 \end{pmatrix}$$

οι οποίοι αποτελούν μία ικανοποιητική προσέγγιση των αναλυτικά υπολογισμένων πινάκων U, Σ και V. Παρατηρούμε ότι παρά τη διαφορά στα πρόσημα μεταξύ της αναλυτικά και αλγοριθμικά υπολογισμένης ΑΙΤ και στις δύο περιπτώσεις οι πολλαπλασιασμοί $U\Sigma V^{\mathsf{T}}$ δίνουν το σωστό πίνακα A.

Σχόλια στη Βιβλιογραφία

Τα ιστορικά στοιχεία αποτελούν μία επιλογή όσων αναγράφονται στην αναφορά [5]. Επιπλέον, για τη θεωρία της AIT ανατρέξαμε στο άρθρο [6] και στο βιβλίο [7]. Η απόδειξη της ύπαρξης της AIT που παρουσιάσαμε είναι μετατροπή αυτής που παρουσιάζεται στην αναφορά [4]. Τα συγγράμματα που χρησιμοποιήθηκαν για τους ορισμούς είναι τα [1], [2] και [3]. Ο αλγόριθμος υπολογισμού της AIT που παρουσιάστηκε δεν βρέθηκε στη βιβλιογραφία αλλά κατασκευάστηκε με χρήση των στοιχείων που βρέθηκαν στο [4].

Κεφάλαιο 3

Πολυωνυμικοί Πίνακες

3.1 Στοιχεία Πολυωνυμικών Πινάκων

Στο κεφάλαιο αυτό εισαγάγουμε τον αναγνώστη στη θεωρία πολυωνυμικών πινάκων και μελετάμε τις συνθήκες κάτω από τις οποίες ο ευθύς πολυωνυμικός πίνακας έχει ίδια μηδενικά με το δυϊκό (dual) του.

Ορισμός 10. Κάθε απεικόνιση $C : \mathbb{R} \to \mathbb{R}^{n\times n}$ της οποίας οι τιμές δίνονται από μία σχέση της μορφής

$$C(\lambda) = \sum_{i=0}^{\ell} C_i \lambda^i$$

όπου $C_i \in \mathbb{R}^{n\times n}$ είναι σταθεροί πίνακες με $C_\ell \neq 0$ και $\lambda \in \mathbb{R}$ η ανεξάρτητη μεταβλητή, ονομάζεται *πολυωνυμικός πίνακας* βαθμού ℓ, [9]. Αν ο συντελεστής C_ℓ του μεγιστοβάθμιου όρου είναι ο μοναδιαίος πίνακας, τότε ο πολυώνυμικός πίνακας ονομάζεται *εναδικός* (monic). Επιπλέον, αν υπάρχουν μη ιδιάζοντες πίνακες $E, F : \mathbb{R} \to \mathbb{R}^{n\times n}$ σταθερής ορίζουσας (unimodular matrix), ώστε

$$A(\lambda) = E(\lambda)B(\lambda)F(\lambda)$$

γράφουμε $A \sim B$ και ονομάζουμε τους πίνακες $A, B : \mathbb{R} \to \mathbb{R}^{n\times n}$ *όμοιους*. Θεωρούμε το γραμμικό εναδικό πολυωνυμικό πίνακα $L : \mathbb{R} \to \mathbb{R}^{(n+p)\times(n+p)}$, $p \geq 0$. Το πολυώνυμο L ονομάζεται *γραμμικοποίηση* ενός εναδικού πολυωνυμικού πίνακα $C : \mathbb{R} \to \mathbb{R}^{n\times n}$ αν

$$L(\lambda) = I\lambda - L_0 \sim \begin{pmatrix} C(\lambda) & \mathbb{O} \\ \mathbb{O} & I \end{pmatrix} = \overline{C(\lambda)}$$

Από τη σχέση $L \sim \overline{C}$ παρατηρούμε ότι το πολυώνυμο $\det L$ είναι βαθμού $n\ell$ όπου ℓ ο βαθμός του πίνακα πολυωνύμων C, δηλαδή το μέγεθος της γραμμικοποίησης L είναι $n\ell$.

Θεώρημα 4. Θεωρούμε έναν εναδικό πολυωνυμικό πίνακα $C : \mathbb{R} \to \mathbb{R}^{n\times n}$ βαθμού ℓ του οποίου οι τιμές είναι

$$C(\lambda) = I\lambda^{\ell} + \sum_{i=0}^{\ell-1} C_i \lambda^i$$

και ορίζουμε $L_0 = C_{(1)}$ όπου $C_{(1)}$ ο συνοδεύων πίνακας ο οποίος είναι

$$C_{(1)} = \begin{pmatrix} \mathbb{O} & I & \mathbb{O} & \cdots & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & I & \cdots & \mathbb{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \cdots & I \\ -C_0 & -C_1 & -C_2 & \cdots & -C_{\ell-1} \end{pmatrix} \in \mathbb{R}^{n\ell\times n\ell}$$

Για το γραμμικό εναδικό πολυωνυμικό πίνακα L με τιμες $L(\lambda) = I\lambda - L_0$ έχουμε [9]

$$L(\lambda) \sim \begin{pmatrix} C(\lambda) & \mathbb{O} \\ \mathbb{O} & I \end{pmatrix}$$

Από τον ορισμό του $C_{(1)}$ παρατηρούμε ότι $\det\left(I\lambda - C_{(1)}\right) = \det C(\lambda)$. Πιο συγκεκριμένα το χαρακτηριστικό πολυώνυμο του πίνακα $C_{(1)}$ ισούται με την ορίζουσα του πίνακα $C(\lambda)$. Οποιεσδήποτε δύο γραμμικοποιήσεις ενός εναδικού πολυωνυμικού πίνακα C είναι όμοιες. Αντίστροφα, αν ο πολυωνυμικός πίνακας T είναι μία γραμμικοποίηση του C και $S \sim T$, τότε ο πολυωνυμικός πίνακας S είναι άλλη μία γραμμικοποίηση του C.

Θεώρημα 5. Ένας πολυωνυμικός πίνακας $L \in \mathbb{R}^{m\times m}$ είναι μία γραμμικοποίηση κάποιου εναδικού πολυωνυμικού πίνακα βαθμού ℓ και μεγέθους $n \times n$ ανν αληθεύουν οι ακόλουθες συνθήκες, [9].

1. $m = n\ell$,
2. $\max_{\lambda\in\mathbb{R}}(\dim(\mathcal{N}(L(\lambda)))) \leq n$

Η δεύτερη συνθήκη του παραπάνω θεωρήματος δηλώνει ότι για κάθε ιδιοτιμή λ_0 του $L(\lambda)$ ο αριθμός των στοιχειωδών διαιρετών του $L(\lambda)$ που αντιστοιχεί στην τιμή λ_0 δεν ξεπερνάει τον αριθμό n.

3.2 Κανονικές Μορφές, Ακολουθίες Jordan και μορφή Smith

Κανονικές Μορφές και Ακολουθίες Jordan

Θεώρημα 6. Κάθε πίνακας $A \in \mathbb{C}^{n \times n}$ είναι όμοιος με έναν διαγώνιο πίνακα της μορφής

$$J = \begin{pmatrix} J_1 & & \mathbb{O} \\ & \ddots & \\ \mathbb{O} & & J_p \end{pmatrix}$$

όπου ο πίνακας J_i, $i = 1, \ldots, p$ συμβολίζει τον τετραγωνικό πίνακα

$$J_i = \begin{pmatrix} \lambda_i & 1 & & \mathbb{O} \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ \mathbb{O} & & & \lambda_i \end{pmatrix}$$

ο οποίος ονομάζεται στοιχειώδης πίνακας Jordan, [2].

Αν το ελάχιστο πολυώνυμο χ ενός πίνακα A είναι γινόμενο πρωτοβάθμιων παραγόντων, δηλαδή αν

$$\chi(z) = (z - \lambda_1)^{r_1} \cdots (z - \lambda_k)^{r_k}, \quad \lambda_1 \neq \cdots \neq \lambda_k$$

τότε ο αντίστοιχος πίνακας J έχει στη διαγώνιο του τις ιδιοτιμές του πίνακα A, κάθε μία τόσες φορές όση είναι η πολλαπλότητα της ως ρίζα του χαρακτηριστικού πολυωνύμου, 1 ή 0 στην άνω δευτερεύουσα διαγώνιο και 0 οπουδήποτε αλλού. Ο πίνακας J ονομάζεται *κανονική μορφή Jordan* του πίνακα A. Ως *γενικευμένο ιδιοδιάνυσμα* ενός πίνακα A ορίζουμε το μη μηδενικό διάνυσμα v το οποίο συνδέεται με κάποια ιδιοτιμή $\lambda \in \sigma(A)$ αλγεβρικής πολλαπλότητας $r \geq 1$ μέσω της σχέσης $(A - \lambda I)^r v = 0$, δηλαδή $v \in \mathcal{N}((A - \lambda I)^r)$. Δοθείσας ιδιοτιμής $\lambda_i \in \sigma(A)$ αλγεβρικής πολλαπλότητας r_i, ο αντίστοιχος στοιχειώδης πίνακας Jordan J_i παράγει μία ακολουθία διανυσμάτων Jordan p_i μέσω του μετασχηματισμού ομοιότητας $AP = PJ$. Το οδηγό διάνυσμα της ακολουθίας Jordan είναι εκείνο για το οποίο $(A - \lambda I)^{r_i} p_{r_i} = 0$, δηλαδή $p_{r_i} \in \mathcal{N}((A - \lambda I)^{r_i})$.

Μορφή Smith

Θεώρημα 7. Για κάθε μη μηδενικό πολυωνυμικό πίνακα $A(s) \in \mathbb{R}^{m\times n}[s]$ υπάρχουν μονομετρικοί (unimodular) πίνακες $M(s) \in \mathbb{R}^{m\times m}[s]$ και $N(s) \in \mathbb{R}^{n\times n}[s]$, τέτοιοι ώστε

$$M(s)A(s)N(s) = \begin{pmatrix} S(s) & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{pmatrix} = S^{\mathbb{C}}_{A(s)}(s)$$

όπου $S(s) = \text{diag}(a_1(s), a_2(s), \dots, a_r(s))$ και τα πολυώνυμα $a_i(s)$, $i = 1, \dots, r$ όπου r ο μέγιστος ακέραιος για τον οποίο υπάρχει μη μηδενική $r \times r$ ελάσσονα, ονομάζονται *πεπερασμένοι στοιχειώδεις διαιρέτες* (finite elementary divisors), ικανοποιούν τη σχέση $a_i(s)|a_{i+1}(s)$ για κάθε $i \in \{1, \dots, r-1\}$ και δίνονται από το πηλίκο

$$a_i(s) = \frac{D_i(A(s))}{D_{i-1}(A(s))}$$

όπου $D_i(A(s))$ παριστάνει το μέγιστο κοινό διαιρέτη όλων των $i \times i$ ελασσόνων οριζουσών του πολυωνυμικού πίνακα $A(s)$. Ο πολυωνυμικός πίνακας $S^{\mathbb{C}}_{A(s)}(s)$ ονομάζεται *μορφή Smith* του πίνακα $A(s)$, [8].

Στην περίπτωση που ο $A(s)$ είναι γραμμικός πολυωνυμικός πίνακας, δηλαδή $A(s) = sI - A_0$, τότε οι μονομετρικοί πίνακες $M(s)$ και $N(s)$ της μορφής Smith είναι σταθεροί. Επιπλέον, η μορφή Smith μπορεί να χρησιμοποιηθεί για ελεγχθεί η ομοιότητα δύο πινάκων. Πιο συγκεκριμένα, δύο πίνακες A και B είναι όμοιοι ανν οι χαρακτηριστικοί τους πίνακες $sI - A$ και $sI - B$ έχουν την ίδια μορφή Smith.

Εφαρμογή 4. Θεωρούμε τους πίνακες

$$A = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & -4 \\ 1 & -1 \end{pmatrix}$$

των οποίων οι χαρακτηριστικοί πίνακες είναι

$$L_A(s) = sI - A = \begin{pmatrix} s-1 & -3 \\ 0 & s-1 \end{pmatrix} \text{ και } L_B(s) = sI - B = \begin{pmatrix} s-3 & 4 \\ -1 & s+1 \end{pmatrix}$$

αντίστοιχα. Οι στοιχειώδεις διαιρέτες για τους χαρακτηριστικούς αυτούς πίνακες είναι

$$a_1(s) = 1, \quad a_2(s) = (s-1)^2, \quad b_1(s) = 1, \quad b_2(s) = (s-3)(s+1) + 4 = (s-1)^2$$

Έχουμε αντίστοιχα τις ακόλουθες μορφές Smith

$$S^{\mathbb{C}}_{L_A(s)}(s) = S^{\mathbb{C}}_{L_B(s)}(s) = \begin{pmatrix} 1 & 0 \\ 0 & (s-1)^2 \end{pmatrix}$$

Εφόσον οι μορφές Smith είναι ίσες για κάθε $s \in \mathbb{R}$ έχουμε ότι οι πίνακες A και B είναι όμοιοι. Γνωρίζουμε ότι οι όμοιοι πίνακες έχουν ίσες ιδιοτιμές και ιδιοδιανύσματα που συνδέονται μεταξύ τους με τον πίνακα ομοιότητας, δηλαδή $\sigma(A) = \sigma(B)$ και $w = Pv$. Με άμεσο υπολογισμό των ιδιοτιμών του πίνακα A παίρνουμε $\sigma(A) = \{1, 1\}$ και παρατηρούμε ότι η δύναμη που εμφανίζεται στη μορφή Smith ισούται με την αλγεβρική πολλαπλότητα της αντίστοιχης ιδιοτιμής.

Εφαρμογή 5. Δοθέντος του πίνακα

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

αναζητάμε τη Smith μορφή του χαρακτηριστικού πίνακα $sI - A$, δηλαδή

$$B(s) = sI - A = \begin{pmatrix} s-2 & -1 & 0 & 0 & 0 \\ 0 & s-2 & 0 & 0 & 0 \\ 0 & 0 & s-2 & -1 & 0 \\ 0 & 0 & 0 & s-2 & -1 \\ 0 & 0 & 0 & 0 & s-2 \end{pmatrix}$$

Οι στοιχειώδεις διαιρέτες του πίνακα αυτού είναι

$$a_1(s) = 1,\ a_2(s) = 1,\ a_3(s) = 1,\ a_4(s) = (s-2)^2,\ a_5(s) = (s-2)^3$$

όπου, για παράδειγμα, ο στοιχειώδης διαιρέτης $a_5(s)$ υπολογίζεται ως ακολούθως. Η ορίζουσα του πίνακα $B(s)$ είναι $D_5(s) = (s-2)^5$ και οι 4×4 ελάσσονες ορίζουσες του πολυωνυμικού πίνακα $B(s)$ είναι

$$|B_{11}|(s) = (s-2)^4,\ |B_{12}|(s) = |B_{13}|(s) = |B_{14}|(s) = |B_{15}|(s) = 0,$$

$$|B_{21}(s)| = -(s-2)^3,\ |B_{22}|(s) = (s-2)^4,\ |B_{23}|(s) = |B_{24}|(s) = |B_{25}|(s) = 0,$$

$$|B_{31}|(s) = 0,\ |B_{32}|(s) = 0,\ |B_{33}|(s) = (s-2)^4,\ |B_{34}|(s) = |B_{35}|(s) = 0,$$

$$|B_{41}|(s) = |B_{42}|(s) = 0,\ |B_{43}|(s) = -(s-2)^3,\ |B_{44}|(s) = (s-2)^4,\ |B_{45}|(s) = 0,$$

$$|B_{51}|(s) = |B_{52}|(s) = 0,\ |B_{53}|(s) = (s-2)^2,\ |B_{54}|(s) = -(s-2)^3,\ |B_{55}|(s) = (s-2)^4$$

των οποίων ο μέγιστος κοινός διαιρέτης είναι $(s-2)^2$, δηλαδή $D_4(s) = (s-2)^2$. Τελικά $a_5(s) = D_5(s)/D_4(s) = (s-2)^3$. Τελικά η Smith μορφή του πίνακα $B(s) = sI - A$ είναι

$$S^{\mathbb{C}}_{B(s)}(s) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & (s-2)^2 & 0 \\ 0 & 0 & 0 & 0 & (s-2)^3 \end{pmatrix}$$

3.3 Ρίζες Πολυωνυμικών Πινάκων

Στην παράγραφο που ακολουθεί δείχνουμε ότι τα μηδενικά ενός πολυωνυμικού πίνακα ικανοποιούν ένα γενικευμένο πρόβλημα ιδιοτιμών. Επιπλέον, με χρήση του γενικευμένου προβλήματος ιδιοτιμών εξάγουμε τις συνθήκες κάτω από τις οποίες ο ευθύς πολυωνυμικός πίνακας έχει ίδια μηδενικά με τον δυϊκό του.

Ορισμός 11. Ως μηδενικά ενός $n \times n$ πολυωνυμικού πίνακα $C(\lambda)$ βαθμού ℓ ορίζουμε [15] τους μιγαδικούς αριθμούς z_i, $i = 1, \ldots, k \leq n\ell$ για τους οποίους

$$\operatorname{rank}(C(z_i)) < \operatorname{rank}(C(\lambda)) = n$$

ή ισοδύναμα τις ρίζες του βαθμωτού πολυωνύμου $p(\lambda) = \det(C(\lambda))$.

Θα δείξουμε ότι τα μηδενικά ενός πολυωνυμικού πίνακα $C(\lambda)$ είναι οι ιδιοτιμές z του γενικευμένου προβλήματος ιδιοτιμών

$$C_{(1)}v = zDv$$

όπου D είναι ο τμηματικά διαγώνιος πίνακας με όλα τα στοιχεία του ίσα με το μοναδιαίο πίνακα I και τελευταίο στοιχείο το συντελεστή του μεγιστοβάθμιου όρου, δηλαδή

$$D = \begin{pmatrix} I & & & \mathbb{O} \\ & I & & \\ & & \ddots & \\ \mathbb{O} & & & C_\ell \end{pmatrix}$$

Απόδειξη. Δοθέντος μηδενικού z ενός πολυωνυμικού πίνακα $C(s)$ βαθμού ℓ για κάθε διάνυσμα $e \in \mathcal{N}(C(z))$ έχουμε $C(z)e = 0$ ή ισοδύναμα [15]

$$(C_0 + C_1 z + \cdots + C_\ell z^\ell)e = 0$$

Ορίζοντας τα διανύσματα $e_k = ze_{k-1}$ με $k = 1, \dots, \ell - 1$ και $e_k = z^k e$ με $k = 0, \dots, \ell - 1$ η παραπάνω σχέση γράφεται

$$C_0 e_0 + C_1 e_1 + \cdots + C_{\ell-1} e_{\ell-1} + zC_\ell z^{\ell-1} e = 0 \Leftrightarrow$$

$$-C_0 e_0 - C_1 e_1 - \cdots - C_{\ell-1} e_{\ell-1} = zC_\ell e_{\ell-1}$$

Είναι προφανές ότι ισχύει η σχέση

$$\begin{pmatrix} \mathbb{O} & I & \mathbb{O} & \cdots & \mathbb{O} \\ \mathbb{O} & \mathbb{O} & I & \cdots & \mathbb{O} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \cdots & I \\ -C_0 & -C_1 & -C_2 & \cdots & -C_{\ell-1} \end{pmatrix} v = z \begin{pmatrix} I & & & & \mathbb{O} \\ & I & & & \\ & & I & & \\ & & & \ddots & \\ \mathbb{O} & & & & C_\ell \end{pmatrix} v$$

ή

$$C_{(1)} v = zDv$$

όπου

$$v = \begin{pmatrix} e_0 & e_1 & \cdots & e_{\ell-1} \end{pmatrix}^\mathsf{T}$$

Δηλαδή τα μηδενικά του πολυωνυμικού πίνακα $C(s)$ είναι οι ιδιοτιμές του παραπάνω γενικευμένου προβλήματος ιδιοτιμών. $\square$

Για έναν εναδικό (monic) πολυωνυμικό πίνακα $C(s)$ γνωρίζουμε ότι ο συντελεστής του μεγιστοβάθμιου όρου είναι $C_\ell = I$, δηλαδή έχουμε $D = I$ και το πρόβλημα ιδιοτιμών για την εύρεση των μηδενικών του πολυωνύμου ανάγεται στο $C_{(1)} v = zv$. Για να υπολογίσουμε τις ιδιοτιμές z του προβλήματος αυτού, λύνουμε την εξίσωση

$$\det(C_{(1)} - zI) = \det(zI - C_{(1)}) = 0$$

Επιπλέον, αφού η ιδιοτιμή z είναι μηδενικό του $C(s)$ έχουμε ότι $\det(C(z)) = 0$ και τελικά συμπεραίνουμε ότι

$$\det(zI - C_{(1)}) = \det(C(z))$$

Στην περίπτωση που ο πίνακας C_ℓ είναι μη ιδιάζων υπάρχει ο αντίστροφος D^{-1} του πίνακα D και το πρόβλημα είναι ισοδύναμο με ένα σύνηθες πρόβλημα ιδιοτιμών. Τα μηδενικά του $C(\lambda)$ είναι οι ιδιοτιμές του πίνακα $D^{-1}C_{(1)}$,

$$C_{(1)} v = sDv \Leftrightarrow D^{-1} C_{(1)} v = sv$$

δηλαδή $z = s$. Επιπλέον, το χαρακτηριστικό πολυώνυμο ενός $n \times n$ πίνακα A μπορεί να γραφεί στη μορφή [7]

$$p(s) = \det(A - sI) = (-1)^n (s^n + c_1 s^{n-1} + \cdots + c_{n-1} s + c_n)$$

με

$$c_k = \frac{(-1)^k}{k!} \det \begin{pmatrix} t_1 & 1 & 0 & 0 & \cdots & 0 \\ t_2 & t_1 & 2 & 0 & \cdots & 0 \\ t_3 & t_2 & t_1 & 3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{k-1} & t_{k-2} & t_{k-3} & t_{k-4} & \cdots & k-1 \\ t_k & t_{k-1} & t_{k-2} & t_{k-3} & \cdots & t_1 \end{pmatrix}$$

όπου $t_k = \mathrm{tr}(A^k)$ και $k = 1, \ldots, n$. Επιπλέον, μπορεί να αποδειχτεί ότι ο k συντελεστής του χαρακτηριστικού πολυωνύμου δίνεται από το άθροισμα των $k \times k$ διαγώνιων ελασσόνων του πίνακα A. Παρατηρούμε ότι

$$c_1 = -t_1, \quad c_2 = \frac{1}{2}(t_1^2 - t_2), \ldots, c_n = (-1)^n \det(A)$$

Δύο προβλήματα ιδιοτιμών $Ax = ax$ και $By = by$ έχουν ίδια σύνολα ιδιοτιμών, δηλαδή $\sigma(A) = \sigma(B)$, αν και μόνο αν τα χαρακτηριστικά τους πολυώνυμα

$$p(a) = \det(A - aI) = (-1)^n(a^n + p_1 a^{n-1} + \cdots + p_{n-1}a + p_n)$$

$$q(b) = \det(B - bI) = (-1)^n(b^n + q_1 b^{n-1} + \cdots + q_{n-1}b + q_n)$$

είναι ίσα, δηλαδή

$$p(s) = q(s) \Leftrightarrow p_k = q_k, k = 1, \ldots, n$$

Από όπου παρατηρούμε ότι το πλήθος n των αναγκαίων συνθηκών ισούται με το βαθμό του χαρακτηριστικού πολυωνύμου του αντίστοιχου πίνακα, δηλαδή έχουμε

$$\mathrm{tr}(A) = \mathrm{tr}(B), \quad \mathrm{tr}(A^2) = \mathrm{tr}(B^2), \ldots, \det(A) = \det(B)$$

Θεωρούμε τους πολυωνυμικούς πίνακες $C, \tilde{C} : \mathbb{R} \to \mathbb{R}^{n \times n}$ βαθμού ℓ με τιμές

$$C(s) = C_\ell s^\ell + \cdots + C_1 s + C_0, \quad \tilde{C}(s) = s^\ell C(1/s) = C_0 s^\ell + \cdots + C_{\ell-1}s + C_\ell$$

και ονομάζουμε τον πολυωνυμικό πίνακα $\tilde{C}$ *δυϊκό* (dual) του ευθέως πολυωνυμικού πίνακα C, [18]. Παρατηρούμε ότι αν ο αριθμός $z \neq 0$ είναι μηδενικό του πολυωνυμικού πίνακα C, τότε ο αριθμός $1/z$ είναι μηδενικό του δυϊκού πολυωνυμικού πίνακα $\tilde{C}$, δηλαδή έχουμε

$$\det C(z) = 0 \Leftrightarrow \det \tilde{C}(1/z) = 0$$

Έστω ότι οι C και $\tilde{C}$ έχουν μη ιδιάζοντες συντελεστές C_ℓ, C_0. Βάση των παραπάνω τα μηδενικά των πολυωνύμων $C, \tilde{C}$ είναι οι ιδιοτιμές p, q των προβλημάτων

$$D^{-1}C_{(1)}v = pv, \quad \tilde{D}^{-1}\tilde{C}_{(1)}w = qw$$

Για να είναι ίσες οι ιδιοτιμές p και q των πινάκων $D^{-1}C_{(1)}$ και $\tilde{D}^{-1}\tilde{C}_{(1)}$ πρέπει τα χαρακτηριστικά τους πολυώνυμα να είναι ίσα, δηλαδή

$$\det\left(D^{-1}C_{(1)} - pI\right) = \det\left(\tilde{D}^{-1}\tilde{C}_{(1)} - qI\right)$$

δηλαδή πρέπει να ικανοποιούνται οι προαναφερθείσες συνθήκες. Από τη συνθήκη της ισότητας του σταθερού όρου έχουμε

$$|\tilde{D}^{-1}\tilde{C}_{(1)}| = |D^{-1}C_{(1)}| \Leftrightarrow \frac{|\tilde{C}_{(1)}|}{|\tilde{D}|} = \frac{|C_{(1)}|}{|D|}$$

όπου $|\cdot| \equiv \det(\cdot)$. Επειδή

$$|\tilde{D}| = |I|\cdots|I|\cdot|C_0| = |C_0|, \quad |D| = |I|\cdots|I|\cdot|C_\ell| = |C_\ell|$$

η παραπάνω σχέση γράφεται

$$\frac{|\tilde{C}_{(1)}|}{|C_0|} = \frac{|C_{(1)}|}{|C_\ell|}$$

Γενικά έχουμε ότι

$$C_{(1)} = \begin{pmatrix} \mathbb{O} & I \\ -C_0 & R \end{pmatrix} = \begin{pmatrix} \mathbb{O} & I \\ I & \mathbb{O} \end{pmatrix} \begin{pmatrix} -C_0 & R \\ \mathbb{O} & I \end{pmatrix}$$

όπου

$$R = \begin{pmatrix} -C_1 & -C_2 & \cdots & -C_{\ell-1} \end{pmatrix}$$

Από την τελευταία σχέση με χρήση του τύπου του Leibniz παίρνουμε

$$|C_{(1)}| = \begin{vmatrix} \mathbb{O} & I \\ I & \mathbb{O} \end{vmatrix} \cdot \begin{vmatrix} -C_0 & R \\ \mathbb{O} & I \end{vmatrix} = \mathbb{I}|-C_0| = \mathbb{I}(-1)^n|C_0|, \quad |\tilde{C}_{(1)}| = \mathbb{I}(-1)^n|C_\ell|$$

όπου

$$\mathbb{I} = \det\begin{pmatrix} \mathbb{O} & I \\ I & \mathbb{O} \end{pmatrix}$$

Τελικά με χρήση όλων των παραπάνω, η συνθήκη για την ισότητα των σταθερών όρων των χαρακτηριστικών πολυωνύμων γράφεται

$$|C_0| = \pm|C_\ell|$$

Επιπλέον, από τη συνθήκη των ιχνών της πρώτης δύναμης των πινάκων παίρνουμε

$$\mathrm{tr}(D^{-1}C_{(1)}) = \mathrm{tr}(\tilde{D}^{-1}\tilde{C}_{(1)})$$

Αλλά έχουμε ότι

$$D^{-1}C_{(1)} = \begin{pmatrix} I & \mathbb{O} \\ \mathbb{O} & C_\ell^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbb{O} & & I \\ -C_0 & \cdots & C_{\ell-1} \end{pmatrix} = \begin{pmatrix} \mathbb{O} & & I \\ -C_\ell^{-1}C_0 & \cdots & -C_\ell^{-1}C_{\ell-1} \end{pmatrix}$$

και

$$\tilde{D}^{-1}\tilde{C}_{(1)} = \begin{pmatrix} \mathbb{O} & & I \\ -C_0^{-1}C_\ell & \cdots & -C_0^{-1}C_1 \end{pmatrix}$$

Τελικά η συνθήκη των ιχνών γράφεται

$$\mathrm{tr}(-C_\ell^{-1}C_{\ell-1}) = \mathrm{tr}(-C_0^{-1}C_1)$$

Παρατηρούμε ότι η αντιστροφή των πινακών $D, \tilde{D}$ προϋποθέτει την αντιστροφή των πινάκων C_ℓ και C_0 αντίστοιχα, δηλαδή $|C_\ell| \neq 0$ και $|C_0| \neq 0$. Αλλά για να είναι το $z = 0$ μηδενικό του πολυωνυμικού πίνακα $C(s)$ πρέπει ο σταθερός όρος του χαρακτηριστικού πολυωνύμου του πίνακα $D^{-1}C_{(1)}$ να είναι ίσος με μηδέν, δηλαδή

$$p(s) = 0 \Rightarrow (-1)^n(s^n + p_1 s^{n-1} + \cdots + p_{n-1}s) = 0 \Rightarrow$$
$$(-1)^n s(s^{n-1} + p_1 s^{n-2} + \cdots + p_{n-1}) = 0 \Rightarrow \{s = 0 \text{ ή } \ldots\}$$

με $p_n \propto |C_{(1)}| \propto |C_0| = 0$. Η σχέση αυτή όμως δηλώνει ότι ο πίνακας C_0 είναι ιδιάζων, το οποίο είναι άτοπο από την υπόθεση της ύπαρξης του $\tilde{D}^{-1}$, συνεπώς ο πολυωνυμικός πίνακας $C(s)$ δεν εμφανίζει μηδενικό το $z = 0$ και ως εκ τούτου ο δυϊκός πολυωνυμικός πίνακας δεν εμφανίζει μηδενικό το ∞.

Εφαρμογή 6. Θεωρούμε τον πολυωνυμικό πίνακα C με τιμές

$$C(s) = \begin{pmatrix} s^2+s+3 & 2s^2+s+1 \\ 2s^2+2s+1 & 2s^2+s+1 \end{pmatrix} = C_2 s^2 + C_1 s + C_0$$

όπου

$$C_2 = \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix},\ C_1 = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix},\ \text{και } C_0 = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}$$

Ο δυικός πολυωνυμικός πίνακας $\tilde{C}$ του C είναι

$$\tilde{C}(s) = s^2 C(1/s) = C_0 s^2 + C_1 s + C_2$$

Τα μηδενικά των πολυωνυμικών πινάκων C και $\tilde{C}$ δίνονται από τη λύση των εξισώσεων

$$\det(C(s)) = 0 \Leftrightarrow 2s^4 + 3s^3 - 2s^2 - s - 2 = 0$$

και

$$\det(\tilde{C}(s)) = 0 \Leftrightarrow 2s^4 + s^3 + 2s^2 - 3s - 2 = 0$$

αντίστοιχα. Με χρήση του Matlab υπολογίζουμε τα σύνολα Z και $\tilde{Z}$ των ριζών των παραπάνω εξισώσεων και παίρνουμε

$$Z = \{-2, 1, -0.25 \pm 0.66144\}, \quad \tilde{Z} = \{-0.5, 1, -0.5 \mp 1.32288\}$$

Παρατηρούμε ότι επειδή $z_i \neq 0$ για κάθε $i = 1, \ldots, 4$, κάθε ρίζα $z_i \in Z$ του ευθέως πολυωνυμικού πίνακα συνδέεται με τη ρίζα $\tilde{z}_i \in \tilde{Z}$ του δυϊκού πολυωνυμικού πίνακα με τη σχέση $\tilde{z}_i = 1/z$ για $i = 1, \ldots, 4$.

Εφαρμογή 7. Θεωρούμε τον πολυωνυμικό πίνακα

$$\Pi_1(s) = \begin{pmatrix} 4s^3 + 2s + 1 & 5s^3 + 4s^2 + 2 \\ 6s^3 + 4s + 3 & 7s^3 + 6s^2 + 4 \end{pmatrix}$$

ο οποίος γράφεται

$$\Pi_1(s) = \begin{pmatrix} 4 & 5 \\ 6 & 7 \end{pmatrix} s^3 + \begin{pmatrix} 0 & 4 \\ 0 & 6 \end{pmatrix} s^2 + \begin{pmatrix} 2 & 0 \\ 4 & 0 \end{pmatrix} s + \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} =$$

$$P_3 s^3 + P_2 s^2 + P_1 s + P_0$$

με ιδιάζοντες παράγοντες P_1,P_2. Επιπλέον, ορίζουμε τον δυικό πολυωνυμικό πίνακα

$$\Pi_2(s) = s^3 \Pi(1/s) = P_0 s^3 + P_1 s^2 + P_2 s + P_3$$

Με άμεσο υπολογισμό της ορίζουσας των παραπάνω πινάκων πολυωνύμων λαμβάνουμε την κοινή εξίσωση

$$\{\det(\Pi_1(s)) = 0, \det(\Pi_2(s)) = 0\} \Leftrightarrow s^6 + 3s^4 + 4s^3 + 3s^2 + 1 = 0$$

που σημαίνει ότι τα πολυώνυμα Π_1, Π_2 έχουν κοινά μηδενικά. Για να ελέγξουμε την ισχύ των κριτηρίων ισότητας των χαρακτηριστικών πολυωνύμων κατασκευάζουμε τους πίνακες $C_{(1)}$, D του ευθέως πολυωνυμικού πίνακα, καθώς και τους $\tilde{C}_{(1)}$, $\tilde{D}$, δηλαδή

$$C_{(1)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -2 & -2 & 0 & 0 & -4 \\ -3 & -4 & -4 & 0 & 0 & -6 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 5 \\ 0 & 0 & 0 & 0 & 6 & 7 \end{pmatrix}$$

και

$$\tilde{C}_{(1)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -4 & -5 & 0 & -4 & -2 & 0 \\ -6 & -7 & 0 & -6 & -4 & 0 \end{pmatrix}, \quad \tilde{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 3 & 4 \end{pmatrix}$$

Με υπολογισμό των δυνάμεων του πίνακα $D^{-1}C_{(1)}$ στο Matlab λαμβάνουμε την ισχύ των $n = 6$ απαιτούμενων συνθηκών,

$$\det P_0 = \det P_3 = -2, \quad \mathrm{tr} P_3^{-1} P_2 = \mathrm{tr} P_0^{-1} P_1 = 0,$$

$$\mathrm{tr}((D^{-1}C_{(1)})^2) = \mathrm{tr}((\tilde{D}^{-1}\tilde{C}_{(1)})^2) = -6, \quad \mathrm{tr}((D^{-1}C_{(1)})^3) = \mathrm{tr}((\tilde{D}^{-1}\tilde{C}_{(1)})^3) = -12$$

$$\mathrm{tr}((D^{-1}C_{(1)})^4) = \mathrm{tr}((\tilde{D}^{-1}\tilde{C}_{(1)})^4) = 6, \quad \mathrm{tr}((D^{-1}C_{(1)})^5) = \mathrm{tr}((\tilde{D}^{-1}\tilde{C}_{(1)})^5) = 60$$

Πιο συγκεκριμένα οι ρίζες των Π_1 και Π_2 είναι

$$\{0.62996 \pm 1.85754\mathrm{i}, -0.79370 \pm 0.60831\mathrm{i}, 0.16374 \pm 0.48281\mathrm{i}\}$$

Ως αντιπαράδειγμα θεωρούμε τον πολυωνυμικό πίνακα που προκύπτει με την εναλλαγή $P_0 \leftrightarrow P_1$, δηλαδή θα έχουμε

$$P_0 = \begin{pmatrix} 2 & 0 \\ 4 & 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

που σημαίνει ότι ελαχιστοβάθμιος συντελεστής P_0 είναι ιδιάζων, $\det(P_0) = 0$. Οι εξισώσεις που προκύπτουν με τη μέθοδο της ορίζουσας είναι

$$\{\det(\Pi_1(s)) = 0, \quad \det(\Pi_2(s)) = 0\} \Leftrightarrow$$

$$\{s^2(s^4 + 2s^2 + 6s + 3) = 0, \quad 3s^4 + 6s^3 + 2s^2 + 1 = 0\}$$

με προφανή (διπλή) ρίζα μόνο της πρώτης εξίσωσης το $s = 0$.

Κεφάλαιο 4

Υπολογισμός Μηδενικών Πινάκων Πολυωνύμων

4.1 Εισαγωγικά Στοιχεία

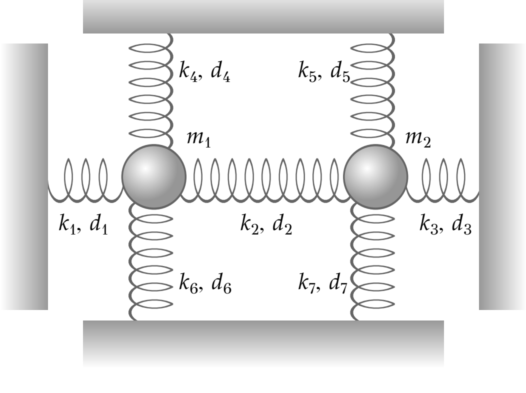
Το πρόβλημα της εύρεσης των μηδενικών ενός πολυωνύμικού πίνακα εμφανίζεται σε πληθώρα προβλημάτων. Ένα τέτοιο παράδειγμα [13] είναι η μοντελοποίηση ενός φυσικού προβλήματος με χρήση συνήθων διαφορικών εξισώσεων δεύτερης τάξης με συντελεστές πίνακες της μορφής

$$M\ddot{z} + D\dot{z} + Kz = f(t), \quad z(0) = a, \quad \dot{z}(0) = b$$

Η λύση του ομογενούς συστήματος είναι της μορφής $e^{st}u$, όπου u ένα σταθερό διάνυσμα. Αυτό οδηγεί στο πολυωνυμικό πρόβλημα ιδιοτιμών δευτέρου βαθμού

$$(Ms^2 + Ds + K)u = 0$$

Ένα τέτοιο πρόβλημα προκύπτει από σύστημα μάζών-ελατηρίων με απόσβεση όπως αυτό του παρακάτω σχήματος.



Οι πίνακες συντελεστές για το σύστημα αυτό είναι

$$M = \mathrm{diag}(m_1, m_1, m_2, m_2),$$

$$D = \mathrm{diag}(d_1 + d_2, d_4 + d_6, d_2 + d_3, d_5 + d_7)$$

και

$$K = \mathrm{diag}(k_1 + k_2, k_4 + k_6, k_2 + k_3, k_5 + k_7)$$

Πιο γενικά [9] τα προβλήματα που προκύπτουν κατά τη μελέτη συστημάτων συνήθων διαφορικών εξισώσεων τάξης $\ell > 1$ σταθερών συντελεστών,

$$\sum_{i=1}^{\ell} C_i D^i x(t)$$

όπου $D^i \equiv \mathrm{d}^i/\mathrm{d}t^i$ ο τελεστής της παραγώγισης. Η αναζήτηση εκθετικών λύσεων της μορφής

$$x(t) = x_0 e^{\lambda_0 t}$$

με $x_0 \mathbb{C}^{m\times 1}$, $\lambda_0 \in \mathbb{C}$ σταθερές οδηγεί στο πρόβλημα

$$L(\lambda_0)x_0 = 0$$

το οποίο δεν είναι παρά ένα πρόβλημα ιδιοτιμών για πολυωνυμικό πίνακα. Πιο γενικά, η συνάρτηση x με τύπο

$$x(t) = \left(\frac{t^k}{k!}x_0 + \cdots + \frac{t}{1!}x_{k-1} + x_k\right)e^{\lambda_0 t}$$

είναι λύση του διαφορικού προβλήματος αν και μόνο αν τα διανύσματα $x_0 \neq 0, x_1, \ldots, x_k$ είναι αλυσίδα Jordan, δηλαδή αν ικανοποιούν τη σχέση

$$\sum_{p=0}^{j} \frac{1}{p!} L^{(p)}(\lambda_0) x_{j-p} = 0, \quad j = 0, \ldots, k$$

4.2 Μηδενικά Πολυωνυμικών Πινάκων και ΑΙΤ

Ακολουθεί το αναγκαίο θεωρητικό υπόβαθρο για την κατανόηση του βασικού θεωρήματος της παραγράφου, το οποίο παρουσιάζει έναν αλγόριθμο για τον υπολογισμό των μηδενικών πολυωνυμικών πινάκων με χρήση της ΑΙΤ.

Λήμμα 1. Έστω δύο πίνακες $A \in \mathbb{C}^{p\times m}[s]$, $B \in \mathbb{C}^{n\times m}$ και $V \in \mathbb{C}^{m\times \ell}$ μία βάση του μηδενοχώρου $\mathcal{N}(B)$, τότε για τον πίνακα

$$C(s) = \begin{pmatrix} A(s) \\ B \end{pmatrix}$$

ισχύει ότι [14]

$$\text{nullity}(C(s)) = \text{nullity}(A(s)V)$$

Απόδειξη. Θεωρούμε μία βάση $T \in \mathbb{C}^{\ell \times k}$ του μηδενοχώρου $\mathcal{N}(A(s)V)$, δηλαδή έχουμε

$$C(s)VT = C(s)U = 0, \quad U = VT$$

και υποθέτουμε ότι υπάρχει στοιχείο x του μηδενοχώρου $\mathcal{N}(C(s))$ το οποίο δεν είναι ταυτόχρονα στοιχείο του χώρου στηλών $\mathcal{C}(U)$. Επιπλέον, επειδή ισχύει

$$\mathcal{N}(C(s)) \subseteq \mathcal{N}(B)$$

υπάρχει στοιχείο $y \in \mathbb{C}^{\ell \times 1}$ το οποίο δεν είναι στοιχείο του στηλοχώρου $\mathcal{C}(T)$ τέτοιο ώστε $x = Vy$. Τελικά εφόσον $y \notin \mathcal{C}(T)$ έχουμε ότι $y \notin \mathcal{N}(A(s)V)$, δηλαδή

$$A(s)Vy = A(s)x \neq 0$$

η οποία διαψεύδει το ότι $x \in \mathcal{N}(C(s))$. Ως βάσεις οι V, U είναι πλήρους τάξης. Τελικά εφόσον

$$\dim \mathcal{N}(C(s)) = k, \quad \dim \mathcal{N}(A(s)V) = k$$

παίρνουμε παίρνουμε τη ζητούμενη σχέση. □

Παρατήρηση 1. Συνδυάζοντας το παραπάνω λήμμα με το θεώρημα τάξης – μηδενικότητας βάση του οποίου το άθροισμα της μηδενικότητας και της τάξης ενός πίνακα ισούται με το πλήθος των στηλών του πίνακα παίρνουμε τη σχέση

$$\text{rank}(C(s)) = \text{rank}(A(s)V) + m - \ell$$

από την οποία συμπεραίνουμε ότι ένα στοιχείο $z \in \mathbb{C}$ είναι μηδενικό του πίνακα πολυωνύμων $C(s)$ αν και μόνο αν είναι μηδενικό του πίνακα $A(s)V$ επειδή

$$C(s)V = 0 \Leftrightarrow \begin{pmatrix} A(s) \\ B \end{pmatrix} V = 0$$

Εφόσον επιλέξαμε βάση V του μηδενοχώρου $\mathcal{N}(B)$ έχουμε $BV = 0$ και τελικά αρκεί και $A(s)V = 0$.

Λήμμα 2. Έστω δύο πίνακες $A \in \mathbb{C}^{m \times p}[s]$, $B \in \mathbb{C}^{m \times n}$ και $V \in \mathbb{C}^{\ell \times m}$ μία βάση του αριστερού μηδενοχώρου $\mathcal{K}(B)$, τότε κάποιο $z \in \mathbb{C}$ είναι μηδενικό του πίνακα

$$C(s) = \begin{pmatrix} A(s) & B \end{pmatrix}$$

αν και μόνο αν είναι μηδενικό του πίνακα $VA(s)$.

Απόδειξη. Θέτουμε $A_1(s) = A^\mathsf{T}(s)$, $B_1 = B^\mathsf{T}$, $V_1 = V^\mathsf{T}$ και $C_1(s) = C^\mathsf{T}(s)$. Από το προηγούμενο λήμμα της παραγράφου αυτής έχουμε ότι κάποιο $z \in \mathbb{C}$ είναι μηδενικό του $C_1(s)$ αν και μόνο αν είναι μηδενικό του $A_1(s)V_1$. Επειδή για κάθε $x \in \mathbb{C}$ έχουμε

$$\mathrm{rank}(C(s)) = \mathrm{rank}(C_1(s)), \quad \mathrm{rank}(A_1(s)V_1) = \mathrm{rank}(VA(s))$$

συμπεραίνουμε ότι το στοιχείο z είναι μηδενικό του $C(s)$ αν και μόνο αν είναι μηδενικό του $C_1(s)$ καθώς επίσης το z είναι μηδενικό του $C_1(s)$ αν και μονο αν είναι μηδενικό $A_1(s)V_1$,

$$A_1(s)V_1 = A^\mathsf{T}(s)V^\mathsf{T} = (VA(s))^\mathsf{T}$$

δηλαδή αν και μόνο αν είναι μηδενικό του $VA(s)$. Τελικά έχουμε ότι το στοιχείο z είναι μηδενικό του $C(s)$ αν και μόνο αν είναι μηδενικό του $VA(s)$. □

Λήμμα 3. Θεωρούμε τον πολυωνύμικό πίνακα

$$C(s) = \sum_{i=0}^{n} C_i s^i$$

και ορίζουμε τους πίνακες

$$E = \begin{pmatrix} C_n & \mathbb{O}_{p\times(k-m)} \\ \mathbb{O}_{(k-m)\times m} & I_{k-m} \end{pmatrix}, \quad F = \begin{pmatrix} -C_{n-1} & \cdots & -C_0 \\ I_{k-m} & & \mathbb{O}_{(k-m)\times m} \end{pmatrix}$$

Ένα στοιχείο $z \in \mathbb{C}$ είναι μηδενικό του $C(s)$ αν και μόνο αν είναι μηδενικό του

$$A(s) = Es - F$$

Απόδειξη. Αν $G \in \mathbb{R}^{p\times nm}[s]$ είναι ο πίνακας με γραμμές τις πρώτες p γραμμές του $A(s)$ και $H \in \mathbb{R}^{m(n-1)\times nm}[s]$ είναι ο πίνακας με γραμμές τις $m(n-1)$ γραμμές του πίνακα $A(s)$, δηλαδή

$$A(s) = \begin{pmatrix} G(s) \\ H(s) \end{pmatrix}$$

όπου

$$G(s) = \begin{pmatrix} C_n s + C_{n-1} & C_{n-2} & \cdots & C_0 \end{pmatrix}$$

$$H(s) = \begin{pmatrix} \mathbb{O}_{(n-1)m\times m} & I_{(n-1)m} \end{pmatrix} s - \begin{pmatrix} I_{(n-1)m} & \mathbb{O}_{(n-1)m\times m} \end{pmatrix}$$

τότε για κάθε στοιχείο $x \in \mathbb{C}$ υπάρχει μία βάση U του μηδενοχώρου $\mathcal{N}(H(s))$ της μορφής

$$U = \begin{pmatrix} x^{n-1}I_m & \cdots & xI_m & I_m \end{pmatrix}^\mathsf{T}$$

και ισχύει

$$G(s)U = C(s)$$

Από το πρώτο λήμμα της παραγράφου συμπεραίνουμε ότι

$$\text{nullity}(A(s)) = \text{nullity}(G(s)U) = \text{nullity}(C(s))$$

και το ζητούμενο έχει αποδειχτεί. $\square$

Θεώρημα 8. Δοθέντος πολυωνυμικού πίνακα $C : \mathbb{R} \to \mathbb{R}^{p\times m}[s]$ και πινάκων E_0, F_0 ο παρακάτω αλγόριθμος υπολογίζει τα μηδενικά $z_i \in \mathbb{C}$ του πίνακα $C(s)$, [14].

Αλγόριθμος zeros
  Διάβασε $C_n, C_{n-1}, \ldots, C_0$
  Κατασκεύασε E_0, F_0
  $i \leftarrow 0, \ell_0 \leftarrow p + (n-1)m, k_0 \leftarrow mn$
  Όσο $\neg\text{exists}(z)$ **επανέλαβε**
    $[U_i, S_i, V_i] \leftarrow \text{svd}(E_i)$
    $r_i \leftarrow \text{rank}(E_i)$
    Αν $r_i = 0$ **τότε**
      Αν $E_i = 0$ **τότε**
        $z \leftarrow []$
      Αλλιώς
        $z \leftarrow \text{eig}(E_i^{-1}F_i)$
      Τέλος_αν
    Αλλιώς_αν $r_i = \ell_i$ **τότε**
      Αν $r_i = k_i$ **τότε**
        Αν $E_i = 0$ **τότε**
          $z \leftarrow []$
        Αλλιώς
          $z \leftarrow \text{eig}(E_i^{-1}F_i)$
        Τέλος_αν
      Αλλιώς
        $F_i' \leftarrow (F_i V_i^{\mathsf{T}})(:, \text{last}(k_i - r_i))$
        $T_i \leftarrow \mathcal{K}(F_i')_{\text{SVD}}$
        $i \leftarrow i+1, \ell_i \leftarrow j_{i-1}, k_i \leftarrow r_{i-1}$
        $E_i \leftarrow (T_{i-1}E_{i-1}V_{i-1}^{\mathsf{T}})(:, \text{first}(r_{i-1}))$
        $F_i \leftarrow (T_{i-1}F_{i-1}V_{i-1}^{\mathsf{T}})(:, \text{first}(r_{i-1}))$

Τέλος_αν

Αλλιώς

$F_i' \leftarrow (U_i^\mathsf{T} F_i)(\text{last}(\ell_i - r_i), :)$

$W_i \leftarrow \mathcal{N}(F_i')_{\text{SVD}}$

$i \leftarrow i+1, \ell_i \leftarrow r_{i-1}, k_i \leftarrow j_{i-1}$

$E_i \leftarrow (U_{i-1}^\mathsf{T} E_{i-1} W_{i-1})(\text{first}(r_{i-1}), :)$

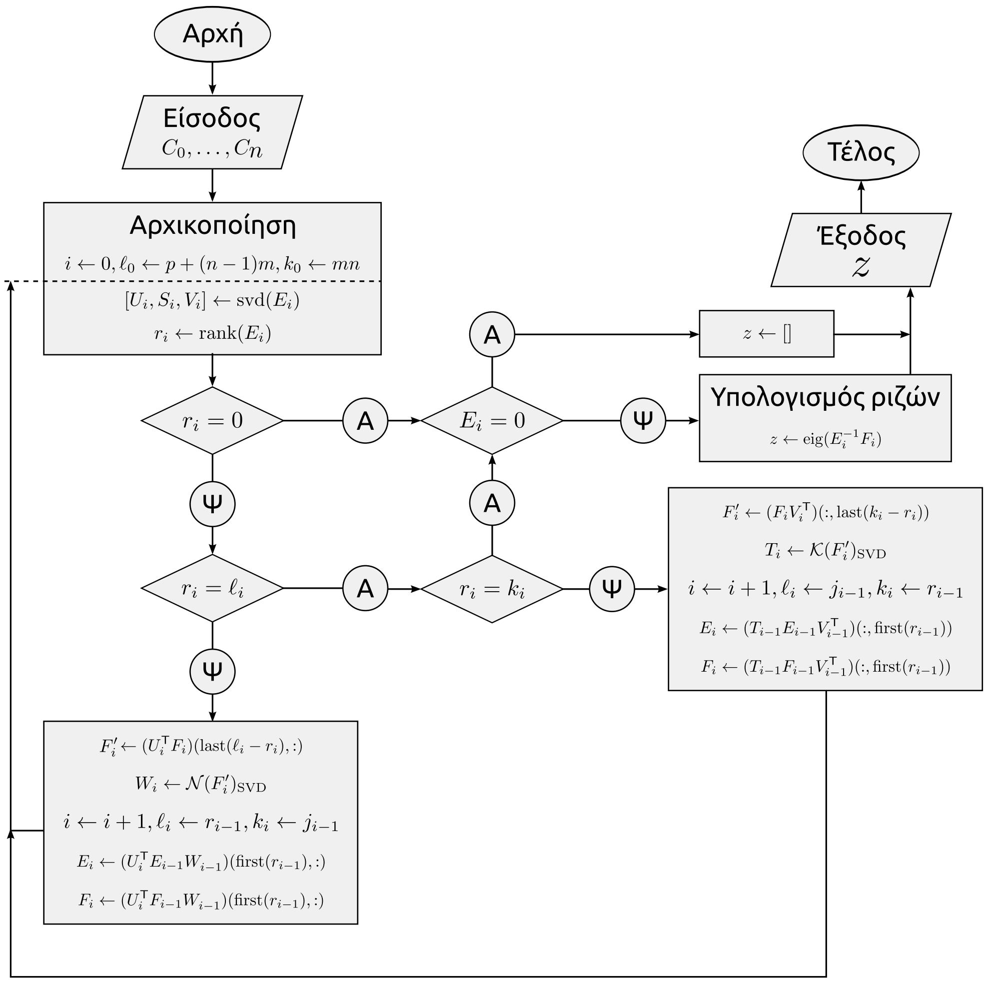
$F_i \leftarrow (U_{i-1}^\mathsf{T} F_{i-1} W_{i-1})(\text{first}(r_{i-1}), :)$

Τέλος_αν

Τέλος_επανάληψης

Τέλος zeros

ή σε διάγραμμα ροής



Απόδειξη. Ένα στοιχείο $z \in \mathbb{C}$ είναι μηδενικό του πολυωνυμικού πίνακα $C(s)$ ανν είναι μηδενικό της γραμμικοποίησης $A(s) = E_0 s - F_0$ του $C(s)$. Για την περίπτωση που $r_0 < \ell_0$ έχουμε ότι το στοιχείο z είναι μηδενικό του πολυωνυμικού πίνακα $A(s)$ ανν είναι μηδενικό του $U_0^\mathsf{T} A(z)$, δηλαδή του πίνακα

$$U_0^\mathsf{T} A(s) = U_0^\mathsf{T}(E_0 s - F_0) = \begin{pmatrix} E_0'' s - F_0'' \\ F_0' \end{pmatrix}$$

όπου ο πίνακας U_0 είναι πλήρους τάξης επειδή $U_0^\mathsf{T} U_0 = I$. Επιπλέον, αν W_0 είναι μία βάση του μηδενοχώρου $\mathcal{N}(F_0')$ έχουμε ότι το z είναι μηδενικό του $U_0^\mathsf{T} A(s)$ ανν είναι μηδενικό του πίνακα

$$(E_0'' s - F_0'')W_0 = E_1 s - F_1$$

Τελικά έχουμε ότι το στοιχείο z είναι μηδενικό του πολυωνυμικού πίνακα $C(s)$ ανν είναι μηδενικό του $E_1 z - F_1$. Αν $r_0 = \ell_0 < k_0$ τότε έχουμε ότι το z είναι μηδενικό του πολυωνυμικού πίνακα $C(s)$ ανν είναι μηδενικό του

$$A(s)V_0^\mathsf{T} = (E_0 s - F_0)V_0^\mathsf{T} = \begin{pmatrix} E_0'' s - F_0'' & F_0' \end{pmatrix}$$

Αν T_0 είναι μία βάση του αριστερού μηδενοχώρου $\mathcal{K}(F_0')$, τότε το στοιχείο z είναι μηδενικό του $A(s)V_0^\mathsf{T}$ ανν είναι μηδενικό του

$$T_0(E_0'' s - F_0'') = E_1 s - F_1$$

Γίνεται άμεσα αντιληπτό με χρήση επαγωγής ότι σε κάθε περίπτωση το στοιχείο z είναι μηδενικό του πολυωνυμικού πίνακα $C(s)$ ανν είναι μηδενικό του γραμμικού πολυωνυμικού πίνακα $E_i s - F_i = 0$. Αν $E_i = 0$ τότε δεν υπάρχουν στοιχεία του $\mathbb{C}$ ικανά να ελαττώσουν την τάξη του $E_i s - F_i$. Στην περίπτωση που $|E_i| \neq 0$ ο πίνακας E_i είναι μη ιδιάζων και το στοιχείο z είναι μηδενικό του πολυωνυμικού πίνακα $C(s)$ ανν το στοιχείο z ανήκει στο σύνολο των ιδιοτιμών του πίνακα $E_i^{-1} F_i$, δηλαδή $z \in \sigma(E_i^{-1} F_i)$. □

Παρατήρηση 2. Υπενθυμίζουμε ότι δοθέντος $m \times n$ πίνακα $A = U \Sigma V^\mathsf{T}$ τάξης r οι τελευταίες $n - r$ στήλες του πίνακα V αποτελούν βάση του μηδενοχώρου $\mathcal{N}(A)$ και οι τελευταίες $m - r$ στήλες του πίνακα U αποτελούν βάση του αριστερού μηδενοχώρου $\mathcal{K}(A)$.

Εφαρμογή 8. Θεωρούμε τον πολυωνυμικό πίνακα

$$C(s) = \begin{pmatrix} 6s^3 + 4s^2 + s + 6 & 8s^3 + 7s^2 + 3s + 5 \\ 3s^3 + 4s^2 + 4s + 8 & 4s^3 + 7s^2 + 5s + 1 \end{pmatrix}$$

του οποίου αναζητάμε τα μηδενικά. Με άμεσο υπολογισμό της ορίζουσας του παραπάνω πίνακα παίρνουμε το πολυώνυμο

$$\det(C(s)) = 5s^5 - 7s^4 - 62s^3 - 37s^2 - 13s - 34$$

και τα μηδενικά του $C(s)$ είναι

$$z_1 = 4.537, \quad z_2 - 2.433, \quad z_3 = -1.103, \quad z_{4,5} = 0.1996 \pm \mathrm{i}0.7202$$

των οποίων ο αριθμητικός υπολογισμός επιτυγχάνεται στο Matlab με χρήση των παρακάτω εντολών.

```
> p = [5, -7, -62, -37, -13, -34];
> z = roots (p)
z =

   4.53670 + 0.00000i
  -2.43297 + 0.00000i
  -1.10301 + 0.00000i
   0.19964 + 0.72020i
   0.19964 - 0.72020i
```

Για να υπολογίσουμε τα μηδενικά του πολυωνύμικού πίνακα $C(s)$ με χρήση του αλγόριθμου που παρουσιάσαμε τον γράφουμε στη μορφή πολυωνύμου με συντελεστές πίνακες, δηλαδή

$$C(s) = \begin{pmatrix} 6 & 8 \\ 3 & 4 \end{pmatrix} s^3 + \begin{pmatrix} 4 & 7 \\ 4 & 7 \end{pmatrix} s^2 + \begin{pmatrix} 1 & 3 \\ 4 & 5 \end{pmatrix} s + \begin{pmatrix} 6 & 5 \\ 8 & 1 \end{pmatrix} =$$

$$C_3 s^3 + C_2 s^2 + C_1 s + C_0$$

Ο βαθμός του πολυωνυμικού πίνακα είναι $n = 3$. Επιπλέον έχουμε $p = 2$, $m = 2$, $k = nm = 6$ και $\ell_0 = 6$. Ορίζουμε τους πίνακες

$$E_0 = \begin{pmatrix} C_3 & \mathbb{O}_{2\times 4} \\ \mathbb{O}_{4\times 2} & I_4 \end{pmatrix} = \begin{pmatrix} 6 & 8 & 0 & 0 & 0 & 0 \\ 3 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$F_0 = \begin{pmatrix} -C_2 & -C_1 & -C_0 \\ I_4 & & \mathbb{O}_{4\times 2} \end{pmatrix} = \begin{pmatrix} -4 & -7 & -1 & -3 & -6 & -5 \\ -4 & -7 & -4 & -5 & -8 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Υπολογίζουμε την ΑΙΤ του πίνακα E_0 στο Matlab και λαμβάνουμε

$$U_0 = \begin{pmatrix} -0.89443 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & -0.44721 \\ -0.44721 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.89443 \\ -0.00000 & 0.00000 & 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ -0.00000 & 0.00000 & -0.00000 & 1.00000 & 0.00000 & 0.00000 \\ -0.00000 & 0.00000 & -0.00000 & -0.00000 & 1.00000 & 0.00000 \\ -0.00000 & 1.00000 & -0.00000 & -0.00000 & -0.00000 & 0.00000 \end{pmatrix},$$

$$\Sigma_0 = \begin{pmatrix} 11.18001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

και

$$V_0 = \begin{pmatrix} -0.60000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.80000 \\ -0.80000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & -0.60000 \\ -0.00000 & 0.00000 & 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ -0.00000 & 0.00000 & -0.00000 & 1.00000 & 0.00000 & 0.00000 \\ -0.00000 & 0.00000 & -0.00000 & -0.00000 & 1.00000 & 0.00000 \\ -0.00000 & 1.00000 & -0.00000 & -0.00000 & -0.00000 & 0.00000 \end{pmatrix},$$

Επιπλέον, παρατηρούμε ότι η τάξη του πίνακα E_0 είναι $r_0 = \mathrm{rank}(E_0) = 5$ και επειδή είναι $r_0 = 5 < \ell_0 = 6$ υπολογίζουμε τον πίνακα $U_0^\top F_0$ του οποίου τις τελευταίες $l_0 - r_0 = 1$ γραμμές ορίζουμε ως F_0' και υπολογίζουμε τη βάση του W_0 με χρήση της ΑΙΤ στον F_0', δηλαδή

$$F_0' = \begin{pmatrix} -1.7889 - 3.1305 - 3.1305 - 3.1305 - 4.47211.3416 \end{pmatrix}$$

και $F_0' = U_F \Sigma_F V_F^\top$, με

$$V_F = \begin{pmatrix} -0.242536 & -0.424437 & -0.424437 & -0.424437 & -0.606339 & 0.181902 \\ -0.424437 & 0.855017 & -0.144983 & -0.144983 & -0.207119 & 0.062136 \\ -0.424437 & -0.144983 & 0.855017 & -0.144983 & -0.207119 & 0.062136 \\ -0.424437 & -0.144983 & -0.144983 & 0.855017 & -0.207119 & 0.062136 \\ -0.606339 & -0.207119 & -0.207119 & -0.207119 & 0.704115 & 0.088765 \\ 0.181902 & 0.062136 & 0.062136 & 0.062136 & 0.088765 & 0.973370 \end{pmatrix}$$

και η βάση $W_0 = \mathcal{N}(F_0')$ σχηματίζεται από τις τελευταίες $j_0 = 5$ στήλες του V_F, όπου j_0 είναι οι στήλες του F_0' ελαττωμένες κατά την τάξη του $\mathrm{rank}(F_0')$, δηλαδή

$$W_0 = \begin{pmatrix} -0.424437 & -0.424437 & -0.424437 & -0.606339 & 0.181902 \\ 0.855017 & -0.144983 & -0.144983 & -0.207119 & 0.062136 \\ -0.144983 & 0.855017 & -0.144983 & -0.207119 & 0.062136 \\ -0.144983 & -0.144983 & 0.855017 & -0.207119 & 0.062136 \\ -0.207119 & -0.207119 & -0.207119 & 0.704115 & 0.088765 \\ 0.062136 & 0.062136 & 0.062136 & 0.088765 & 0.973370 \end{pmatrix}$$

Έπειτα αυξάνουμε το μετρητή κατά μία μονάδα $i = 1$, έχουμε δηλαδή $l_1 = r_0 = 5$, $k_1 = j_0 = 5$, συνεπώς υπολογίζουμε τους πίνακες E_1, F_1 από τις πρώτες $r_0 = 5$ γραμμές των πινάκων $U_0^\top E_0 W_0$ και $U_0^\top F_0 W_0$ αντίστοιχα, δηλαδή

$$E_1 = \begin{pmatrix} -4.800289 & 4.143983 & 4.143983 & 5.919976 & -1.775993 \\ 0.062136 & 0.062136 & 0.062136 & 0.088765 & 0.973370 \\ -0.144983 & 0.855017 & -0.144983 & -0.207119 & 0.062136 \\ -0.144983 & -0.144983 & 0.855017 & -0.207119 & 0.062136 \\ -0.207119 & -0.207119 & -0.207119 & 0.704115 & 0.088765 \end{pmatrix}$$

$$F_1 = \begin{pmatrix} 3.102988 & -3.605216 & -1.369148 & -0.039296 & 7.614420 \\ -0.144983 & -0.144983 & 0.855017 & -0.207119 & 0.062136 \\ -0.424437 & -0.424437 & -0.424437 & -0.606339 & 0.181902 \\ 0.855017 & -0.144983 & -0.144983 & -0.207119 & 0.062136 \\ -0.144983 & 0.855017 & -0.144983 & -0.207119 & 0.062136 \end{pmatrix}$$

Υπολογίζουμε την ΑΙΤ του πίνακα E_1 στο Matlab και λαμβάνουμε

$$U_1 = \begin{pmatrix} -9.9878 \cdot 10^{-1} & -8.6653 \cdot 10^{-18} & 4.3326 \cdot 10^{-19} & 6.2390 \cdot 10^{-17} & -4.9436 \cdot 10^{-2} \\ 1.0308 \cdot 10^{-2} & 9.5820 \cdot 10^{-1} & 1.8921 \cdot 10^{-1} & -5.0662 \cdot 10^{-2} & -2.0826 \cdot 10^{-1} \\ -2.4052 \cdot 10^{-2} & 2.4086 \cdot 10^{-1} & -7.7207 \cdot 10^{-1} & -3.3042 \cdot 10^{-1} & 4.8594 \cdot 10^{-1} \\ -2.4052 \cdot 10^{-2} & 1.5398 \cdot 10^{-1} & -1.3363 \cdot 10^{-2} & 8.5988 \cdot 10^{-1} & 4.8594 \cdot 10^{-1} \\ -3.4361 \cdot 10^{-2} & 1.1075 \cdot 10^{-2} & 6.0657 \cdot 10^{-1} & -3.8583 \cdot 10^{-1} & 6.9420 \cdot 10^{-1} \end{pmatrix},$$

$$\Sigma_1 = \begin{pmatrix} 9.788892 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.069253 \end{pmatrix}$$

και

$$V_1 = \begin{pmatrix} 0.49129 & 0.00000 & -0.00000 & -0.00000 & -0.87100 \\ -0.42377 & 0.24086 & -0.77207 & -0.33042 & -0.23903 \\ -0.42377 & 0.15398 & -0.01336 & 0.85988 & -0.23903 \\ -0.60539 & 0.01108 & 0.60657 & -0.38583 & -0.34147 \\ 0.18162 & 0.95820 & 0.18921 & -0.05066 & 0.10244 \end{pmatrix}$$

Η τάξη του πίνακα E_1 είναι $r_1 = \mathrm{rank}(E_1) = 5$ και επειδή $r_1 = \ell_1 = 5$ και $k_1 = j_0 = 5$ υπολογίζουμε τον αντίστροφο πίνακα E_1^{-1} και στη συνέχεια τον πίνακα $E_1^{-1}F_1$ του οποίου οι ιδιοτιμές είναι τα μηδενικά του πολυωνυμικού πίνακα, δηλαδή

$$E_1^{-1}F_1 = \begin{pmatrix} 0.027972 & -6.426520 & 6.204239 & 6.216227 & 2.481099 \\ -0.307121 & -2.561613 & 1.469145 & 1.337521 & 1.224710 \\ 0.972333 & -2.282159 & 1.748599 & 1.736741 & 1.104944 \\ 0.022611 & -2.198092 & 2.560134 & 2.569824 & 1.551862 \\ -0.195262 & 0.770949 & 0.043481 & -1.040202 & -0.384782 \end{pmatrix}$$

Τελικά τα μηδενικά του πολυωνύμικού πίνακα $C(s)$ είναι οι ιδιοτιμές του πίνακα $E_1^{-1}F_1$, δηλαδή

$$z_1 = 4.537, \quad z_2 - 2.433, \quad z_3 = -1.103, \quad z_{4,5} = 0.1996 \pm \mathrm{i}0.7202$$

Παρακάτω εκτελούμε βηματικά τον αλγόριθμο που παρουσιάσαμε με χρήση του Matlab. Αφού εισάγουμε τους πίνακες συντελεστές του παραδείγματος κατασκευάζουμε τους πίνακες E_0 και F_0 (η κατασκευή των πινάκων αυτών δεν παρουσιάζεται στον αλγόριθμο για οικονομία χώρου) και υπολογίζουμε την AIT και την τάξη του πίνακα E_0.

```
> C3 = [6, 8; 3, 4];
> C2 = [4, 7; 4, 7];
> C1 = [1, 3; 4, 5];
> C0 = [6, 5; 8, 1];
> E0 = [C3, zeros(2,4); zeros(4,2), eye(4)]
E0 =
    6    8    0    0    0    0
    3    4    0    0    0    0
    0    0    1    0    0    0
    0    0    0    1    0    0
    0    0    0    0    1    0
    0    0    0    0    0    1

> F0 = [-C2, -C1, -C0; eye(4), zeros(4,2)]
F0 =
   -4   -7   -1   -3   -6   -5
   -4   -7   -4   -5   -8   -1
    1    0    0    0    0    0
    0    1    0    0    0    0
    0    0    1    0    0    0
    0    0    0    1    0    0

> [U0, S0, V0] = svd (E0);
> U0
```

```
U0 =
  -0.89443   0.00000   0.00000   0.00000   0.00000  -0.44721
  -0.44721   0.00000   0.00000   0.00000   0.00000   0.89443
  -0.00000   0.00000   1.00000   0.00000   0.00000   0.00000
  -0.00000   0.00000  -0.00000   1.00000   0.00000   0.00000
  -0.00000   0.00000  -0.00000  -0.00000   1.00000   0.00000
  -0.00000   1.00000  -0.00000  -0.00000  -0.00000   0.00000

> S0
S0 =
Diagonal Matrix

   1.1180e+01            0            0            0            0
        0
            0   1.0000e+00            0            0            0
            0
            0            0   1.0000e+00            0            0
            0
            0            0            0   1.0000e+00            0
            0
            0            0            0            0   1.0000e+00
            0
            0            0            0            0            0   5.4384e
            -17

> V0
V0 =
  -0.60000   0.00000   0.00000   0.00000   0.00000   0.80000
  -0.80000   0.00000   0.00000   0.00000   0.00000  -0.60000
  -0.00000   0.00000   1.00000   0.00000   0.00000   0.00000
  -0.00000   0.00000  -0.00000   1.00000   0.00000   0.00000
  -0.00000   0.00000  -0.00000  -0.00000   1.00000   0.00000
  -0.00000   1.00000  -0.00000  -0.00000  -0.00000   0.00000

> r0 = rank (E0)
r0 =  5
```

Επειδή $r_0 = 5 < \ell_0 = 6$ υπολογίζουμε τον πίνακα $U_0^T F_0$ του οποίου την τελευταία γραμμή θέτουμε ως F_0' και υπολογίζουμε τη βάση του W_0.

```
> U0' * F0
ans =
   5.36656   9.39149   2.68328   4.91935   8.94427   4.91935
   0.00000   0.00000   0.00000   1.00000   0.00000   0.00000
   1.00000   0.00000   0.00000   0.00000   0.00000   0.00000
   0.00000   1.00000   0.00000   0.00000   0.00000   0.00000
   0.00000   0.00000   1.00000   0.00000   0.00000   0.00000
  -1.78885  -3.13050  -3.13050  -3.13050  -4.47214   1.34164
```

```
> F0p = (U0' * F0) (6, :) % in Matlab FF=U0'*F0; F0p=FF(6,:)
F0p =
   -1.7889   -3.1305   -3.1305   -3.1305   -4.4721    1.3416

> [WU, WS, WV] = svd (F0p);
> W0 = WV(:, end - size(F0p, 2) + rank(F0p) + 1 : end)
> % equivalent to W0 = null (F0p)
W0 =
   -0.424437   -0.424437   -0.424437   -0.606339    0.181902
    0.855017   -0.144983   -0.144983   -0.207119    0.062136
   -0.144983    0.855017   -0.144983   -0.207119    0.062136
   -0.144983   -0.144983    0.855017   -0.207119    0.062136
   -0.207119   -0.207119   -0.207119    0.704115    0.088765
    0.062136    0.062136    0.062136    0.088765    0.973370
```

Έπειτα αυξάνουμε το μετρητή κατά μία μονάδα και υπολογίζουμε τους πίνακες E_1, F_1 από τους $U_0^\mathsf{T} E_0 W_0$ και $U_0^\mathsf{T} F_0 W_0$ όπως παρακάτω.

```
> E1 = (U0' * E0 * W0) (1:5, :) % in Matlab EE=U0'*E0*W0; E1=EE(1:5,:)
E1 =
   -4.800289    4.143983    4.143983    5.919976   -1.775993
    0.062136    0.062136    0.062136    0.088765    0.973370
   -0.144983    0.855017   -0.144983   -0.207119    0.062136
   -0.144983   -0.144983    0.855017   -0.207119    0.062136
   -0.207119   -0.207119   -0.207119    0.704115    0.088765

> F1 = (U0' * F0 * W0) (1:5, :) % in Matlab FF=U0'*F0*W0; F1=FF(1:5,:)
F1 =
    3.102988   -3.605216   -1.369148   -0.039296    7.614420
   -0.144983   -0.144983    0.855017   -0.207119    0.062136
   -0.424437   -0.424437   -0.424437   -0.606339    0.181902
    0.855017   -0.144983   -0.144983   -0.207119    0.062136
   -0.144983    0.855017   -0.144983   -0.207119    0.062136
```

Συνεχίζουμε υπολογίζοντας την ΑΙΤ και του πίνακα E_1, δηλαδή

```
> [U1, S1, V1] = svd (E1);
> U1
U1 =
   -9.9878e-01   -8.6653e-18    4.3326e-19    6.2390e-17   -4.9436e-02
    1.0308e-02    9.5820e-01    1.8921e-01   -5.0662e-02   -2.0826e-01
   -2.4052e-02    2.4086e-01   -7.7207e-01   -3.3042e-01    4.8594e-01
   -2.4052e-02    1.5398e-01   -1.3363e-02    8.5988e-01    4.8594e-01
   -3.4361e-02    1.1075e-02    6.0657e-01   -3.8583e-01    6.9420e-01

> S1
S1 =
```

```
Diagonal Matrix

   9.788892          0          0          0          0
          0   1.000000          0          0          0
          0          0   1.000000          0          0
          0          0          0   1.000000          0
          0          0          0          0   0.069253

> V1
V1 =
   0.49129   0.00000  -0.00000  -0.00000  -0.87100
  -0.42377   0.24086  -0.77207  -0.33042  -0.23903
  -0.42377   0.15398  -0.01336   0.85988  -0.23903
  -0.60539   0.01108   0.60657  -0.38583  -0.34147
   0.18162   0.95820   0.18921  -0.05066   0.10244

> r1 = rank (E1)
r1 =  5
```

Εφόσον $r_1 = \ell_1 = 5$ υπολογίζουμε τον αντίστροφο πίνακα E_1^{-1} και στη συνέχεια τον πίνακα $E_1^{-1}F_1$.

```
> inv (E1)
ans =
   0.571640   2.619815  -6.112902  -6.112902  -8.732717
   0.213869   0.819815  -0.912902  -1.912902  -2.732717
   0.213869   0.819815  -1.912902  -0.912902  -2.732717
   0.305527   1.171165  -2.732717  -2.732717  -2.903882
  -0.091658   0.648651   0.819815   0.819815   1.171165

> inv (E1) * F1
ans =
   0.027972  -6.426520   6.204239   6.216227   2.481099
  -0.307121  -2.561613   1.469145   1.337521   1.224710
   0.972333  -2.282159   1.748599   1.736741   1.104944
   0.022611  -2.198092   2.560134   2.569824   1.551862
  -0.195262   0.770949   0.043481  -1.040202  -0.384782
```

Τελικά τα μηδενικά του πολυωνύμικού πίνακα $C(s)$ είναι οι ιδιοτιμές του πίνακα $E_1^{-1}F_1$, δηλαδή

```
> z = eig (ans)
z =
   4.53670 + 0.00000i
  -2.43297 + 0.00000i
   0.19964 + 0.72020i
   0.19964 - 0.72020i
  -1.10301 + 0.00000i
```

οι οποίες όπως παρατηρούμε συμφωνούν με αυτές που υπολογίστηκαν στην αρχή της εφαρμογής με χρήση της ορίζουσας.

Κεφάλαιο 5

Συζήτηση και Συμπεράσματα

Το προς αντιμετώπιση πρόβλημα είναι η εύρεση των μηδενικών ενός πολυωνυμικού πίνακα. Αφού μελετήσαμε τους βασικούς αλγόριθμους της ανάλυσης QR και της ΑΙΤ της εφαρμοσμένης γραμμικής άλγεβρας και αντιλαμβανόμενοι τη σημαντικότητα του προβλήματος της εύρεσης των μηδενικών των πολυωνυμικών πινάκων στην επίλυση συνήθων διαφορικών εξισώσεων, παρουσιάσαμε έναν αλγόριθμο για την εύρεση των μηδενικών πολυωνυμικών πινάκων.

Ο αλγόριθμος που παρουσιάστηκε μπορεί να χρησιμοποιηθεί για πολυωνυμικούς πίνακες αυθαίρετου βαθμού, τάξης και διάστασης. Πιο συγκεκριμένα, η χρήση της γραμμικοποίησης του δοθέντος προβλήματος σε συνδυασμό με την ΑΙΤ μας επέτρεψε να ανάγουμε το πρόβλημα σε ένα σύνηθες πρόβλημα ιδιοτιμών για το οποίο μπορούν να χρησιμοποιηθούν οι γνωστές μέθοδοι επίλυσης. Ο αλγόριθμος που παρουσιάζουμε έχει το πλεονέκτημα ότι χρησιμοποιεί θεμελιώδεις ιδιότητες των πολυωνυμικών πινάκων, ενώ το υπολογιστικό κόστος της εύρεσης των μηδενικών καθορίζεται κυρίως από το υπολογιστικό κόστος του υπολογισμού των ιδιοτιμών καθώς και από αυτό του υπολογισμού της ΑΙΤ.

Επιπλέον, μελετήθηκαν οι συνθήκες ισότητας των μηδενικών του ευθέως πολυωνυμικού πίνακα και του δυϊκού του στην περίπτωση που ο ευθύς πολυωνυμικός πίνακας δεν εμφανίζει μηδενικό ίσο με μηδέν και εξήχθησαν αναγκαίες συνθήκες μεταξύ των συντελεστών τους ώστε τα μηδενικά αυτών να είναι ίσα μεταξύ τους.

Παρουσιάσαμε πλήθος παραδειγμάτων τα οποία στηρίζουν τα θεωρητικά μας ευρήματα και γράψαμε κώδικα σε Matlab / GNU Octave και C# για την αναπαραγωγή των πειραμάτων μας.

Παράρτημα Α΄

Εκτέλεση Αριθμητικών Πειραμάτων

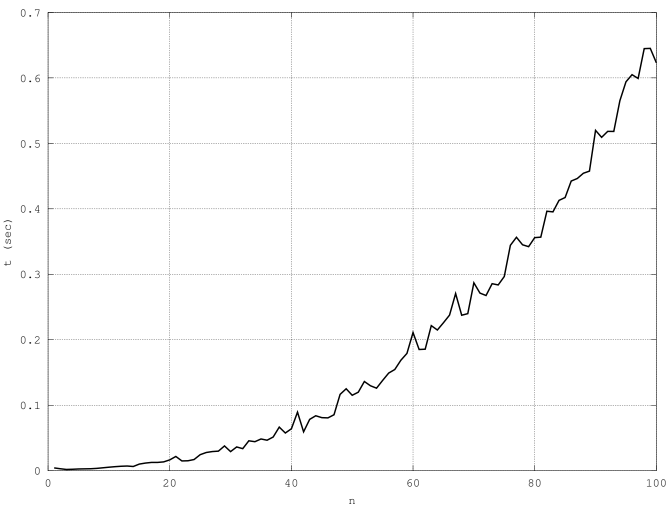
Α΄.1 Πειράματα στο Matlab / GNU Octave

```
> A = [1 , 2, 3; 4, 5, 6]
A =

   1   2   3
   4   5   6

> [U, S, V] = svd (A)
U =

  −0.38632  −0.92237
  −0.92237   0.38632

S =

Diagonal Matrix

   9.50803         0         0
         0   0.77287         0

V =

  −0.42867   0.80596   0.40825
  −0.56631   0.11238  −0.81650
  −0.70395  −0.58120   0.40825

> [U, S, V] = qrbsvd (A)
U =

```

```
  -0.38632  -0.92237
  -0.92237   0.38632

S =

   9.50803   0.00000   0.00000
   0.00000   0.77287   0.00000

V =

  -0.42867   0.80596   0.40825
  -0.56631   0.11238  -0.81650
  -0.70395  -0.58120   0.40825

> [U, S, V] = svd (A')
U =

  -0.42867   0.80596   0.40825
  -0.56631   0.11238  -0.81650
  -0.70395  -0.58120   0.40825

S =

Diagonal Matrix

   9.50803         0
         0   0.77287
         0         0

V =

  -0.38632  -0.92237
  -0.92237   0.38632

> [U, S, V] = qrbsvd (A')
U =

  -0.42867   0.80596   0.40825
  -0.56631   0.11238  -0.81650
  -0.70395  -0.58120   0.40825

S =

   9.50803   0.00000
   0.00000   0.77287
   0.00000   0.00000

V =
```

```

   −0.38632   −0.92237
   −0.92237    0.38632

> z = zersvd ([0, 1; 0, 0], [0, 1; 0, 0])
z = −1

> z = zersvd ([0, 1; 0, 0], [0, −1; 0, 0])
z =  1

> z = zersvd ([0, 1; 0, 0], [0, 7.89; 0, 0])
z = −7.8900

> C3 = [6, 8; 3, 4];
> C2 = [4, 7; 4, 7];
> C1 = [1, 3; 4, 5];
> C0 = [6, 5; 8, 1];
> z = zersvd (C3, C2, C1, C0)
z =

    4.53670 + 0.00000i
   −2.43297 + 0.00000i
    0.19964 + 0.72020i
    0.19964 − 0.72020i
   −1.10301 + 0.00000i
```

Παρακάτω παρουσιάζουμε το χρόνο εκτέλεσης της συνάρτηση zersvd σαν συνάρτηση της διάστασης n τυχαίου $n \times n$ πολυωνυμικού πίνακα βαθμού $\ell = 3$. Ο κώδικας που χρησιμοποιήθηκε για την εκτέλεση αυτού του πειράματος είναι αυτός που περιέχεται στο αρχείο perftest.m.



Επιπλέον, το πορτρέτο απόδοσης για τη μέγιστη τιμή $n = 100$ παράγεται στο GNU Octave ως ακολούθως

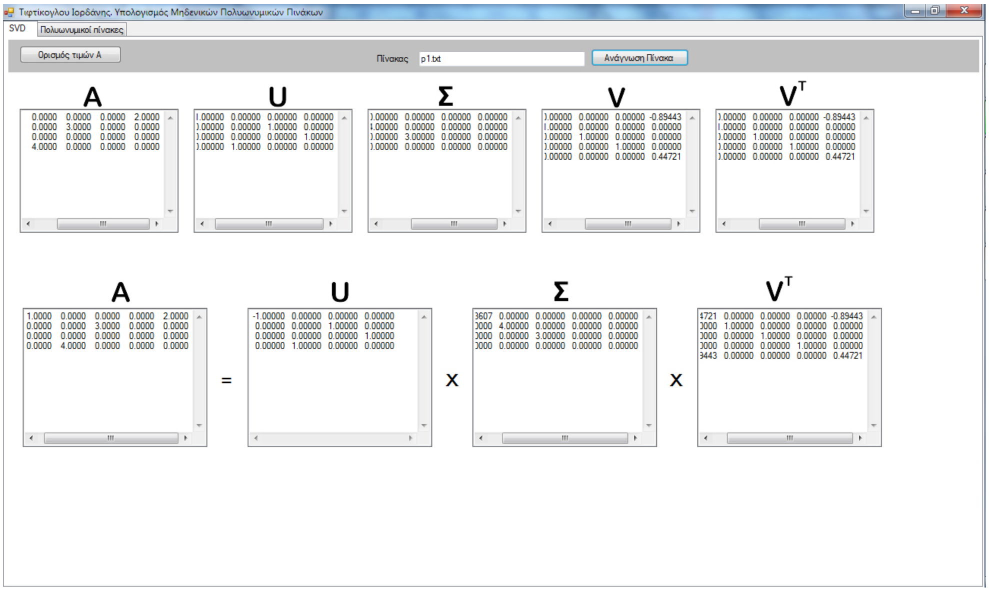
```
> n = 100;
> A = randn (n, n);
> B = randn (n, n);
> C = randn (n, n);
> D = randn (n, n);
> profile on;
> zersvd(A, B, C, D);
> profile off;
> data = profile ('info');
> profshow (data)
   #          Function  Attr       Time (s)          Calls
-----------------------------------------------------------
  38               eig              0.401              1
  25               svd              0.167              2
  37               inv              0.057              1
   5          binary *              0.032              5
   1            zersvd              0.005              1
  10          cell2mat              0.001              1
   7             zeros              0.001              3
  15         binary ==              0.001              8
  26              rank              0.000              1
  36  zersvd>casevii                0.000              1
  16           cellfun              0.000              6
  23               cat              0.000              1
  24          prefix -              0.000              1
  28               isa              0.000              1
  39           profile              0.000              1
  21          num2cell              0.000              1
  20              sort              0.000              1
   8               eye              0.000              2
   3          binary -              0.000              8
  11            nargin              0.000              4
```

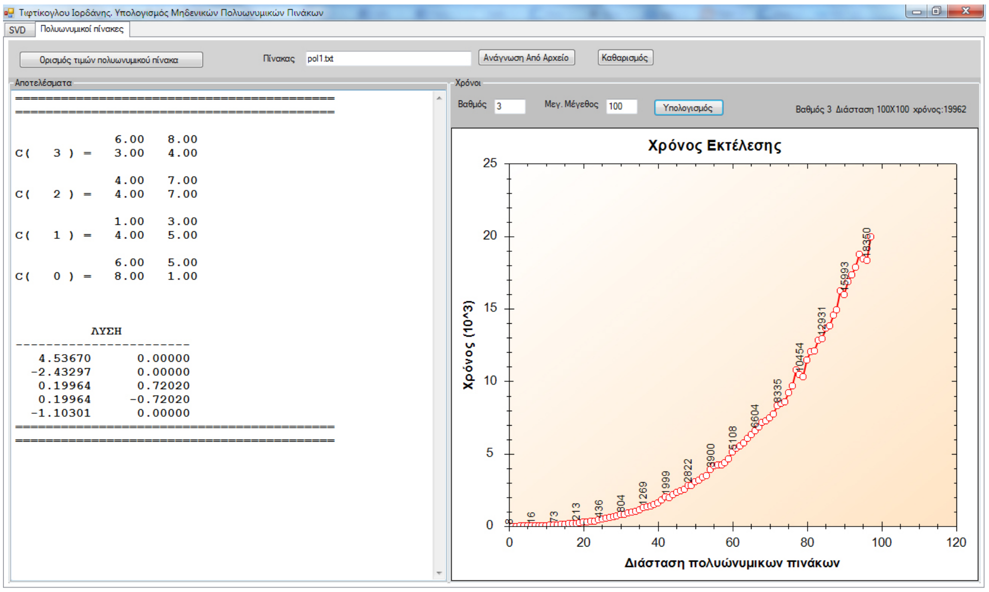
ενώ στο Matlab πληκτρολογούμε την ακολουθία εντολών

```
> profile on;
> zersvd(A, B, C, D);
> profile viewer;
```

Από το πορτρέτο αυτό παρατηρούμε ότι το μεγαλύτερο μέρος του χρόνου καταναλώνεται στις ρουτίνες eig και svd όπως ακριβώς θα περίμενε κανείς. Λόγω του ότι αυτές οι συναρτήσεις είναι υλοποιημένες στη γλώσσα C++ (με χρήση των αρχείων διασύνδεσης προγραμματισμού εφαρμογών του Octave) το έργο που απαιτείται για την εύρεση των μηδενικών θα είναι $\mathcal{O}(n^3)$.

Τα αποτελέσματα από τα προγράμματα που γράφτηκαν σε C# παρουσιάζονται στην επόμενη σελίδα. Σημειώνουμε ότι ο κώδικας αυτός υλοποιεί τους ίδιους αλγόριθμους με αυτούς που υλοποιήσαμε στο Matlab / GNU Octave και όπως αυτοί συζητήθηκαν στην εργασία αυτή.





Επιπλέον, παρακάτω παρουσιάζουμε τον τρόπο κλήσης της συνάρτησης condtest η οποία εκτελεί των έλεγχο των συνθηκών ισότητας των μηδενικών των πολυωνύμων $C(s)$, $s^{\ell}C(1/s)$. Παρατηρούμε ότι στην πρώτη περίπτωση συντελεστών λαμβάνουμε την τιμή 1 που σημαίνει ότι τα μηδενικά των πολυωνύμων $C(s)$, $s^{\ell}C(1/s)$ είναι ίσα μεταξύ τους, ενώ στη δεύτερη περίπτωση ο πίνακας P_0 είναι ιδιάζων και το πρόγραμμα επιστρέφει την τιμή 0 μιας και τα πολυώνυμα $C(s)$, $s^{\ell}C(1/s)$ δεν έχουν όλα τους τα μηδενικά ίσα.

```
> P0 = [1, 2; 3, 4];
> P1 = [2, 0; 4, 0];
> P2 = [0, 4; 0, 6];
> P3 = [4, 5; 6, 7];
> condtest(P3, P2, P1, P0)
ans =  1
> P0 = [2, 0; 4, 0];
> P1 = [1, 2; 3, 4];
> condtest(P3, P2, P1, P0)
warning: inverse: matrix singular to machine precision, rcond = 0
warning: inverse: matrix singular to machine precision, rcond = 0
warning: inverse: matrix singular to machine precision, rcond = 0
warning: inverse: matrix singular to machine precision, rcond = 0
warning: inverse: matrix singular to machine precision, rcond = 0
warning: inverse: matrix singular to machine precision, rcond = 0
ans = 0
```

Παράρτημα Β′

Πηγαίος Κώδικας

Β′.1 Κώδικας Matlab / GNU Octave

Ανάλυση Ιδιαζόντων Τιμών

Ο παρακάτω εκπαιδευτικός κώδικας υλοποιεί την απλή αλγοριθμική προσέγγιση που παρουσιάσαμε στην παράγραφο 2.9 για τον υπολογισμό της ΑΙΤ και δε σχολιάζεται περαιτέρω.

```
function [ U, S, V ] = qrbsvd (A, tol)

 if ( nargin == 1 )
    tol = 1024 * eps;
end

mdim = size (A);
maxiter = 100 * max (mdim);
count = 0;

% Initialize U, S, and V
U = eye (mdim(1));
S = A';
V = eye (mdim(2));

err = realmax;
while ( (err > tol) && (count < maxiter) )
     [Q, S] = qr (S');
     U = U * Q;


     [Q, S] = qr (S');
     V = V * Q;
     e = triu (S, 1);
     E = norm (e(:));
```

```
    F = norm (diag (S));

    if ( F == 0 )
      F = 1.0;
    end

    err = E / F;
    count = count + 1;
end

D = diag (S);
S = zeros (mdim);
for n = 1:length (D)
    Dn = D(n);
    S(n, n) = abs (Dn);
    if Dn < 0
        U(:, n) = - U(:, n);
    end
end

if (nargout <= 1)
   U = diag (S);
end
```

./code/qrbsvd.m

Υπολογισμός Μηδενικών

Στην παρακάτω συνάρτηση υλοποιούμε τον αλγόριθμο υπολογισμού των μηδενικών ενός πίνακα πολυωνύμων που παρουσιάσαμε στο κεφάλαιο 4. Η συνάρτηση zersvd δέχεται μεταβλητό πλήθος ορισμάτων εισόδου με χρήση της δομής δεδομένων varargin και επιστρέφει τα μηδενικά σε διάνυσμα z. Οι συντελεστές του πολυωνύμου εισάγονται ξεκινώντας από αυτόν του μεγιστοβάθμιου όρου. Με σκοπό να χρησιμοποιήσουμε κάθε στοιχείο της δομής δεδομένων varargin στο σχηματισμό των πινάκων E και F (βλ. γρ. 12, 13) χρησιμοποιούμε τη συνάρτηση cell2mat η οποία σχηματίζει ένα διάνυσμα γραμμή με πρώτο στοιχείο το πρώτο κελί της δομής varargin, δεύτερο στοιχείο το δεύτερο κελί της δομής varargin κ.ο.κ. Είναι προφανές ότι ο βαθμός του πολυωνύμου μπορεί να βρεθεί από το πλήθος των πινάκων εισόδου (βλ. γρ. 4). Έπειτα ξεκινάμε ένα φαινομενικά άπειρο επαναληπτικό βρόγχο while (βλ. γρ. 15, άπειρος εφόσον η συνθήκη επανάληψης είναι διαρκώς αληθής, δηλαδή 1) στον οποίο υλοποιούμε το συζητούμενο αλγόριθμο. Όλες οι πιθανές δομές ελέγχου ροής if καταλήγουν είτε στην περίπτωση r==0, είτε στην περίπτωση r==k όπου και καλούμε τη συνάρτηση casevii (βλ. γρ. 20, 24, 52) ακολουθούμενη από τη λέξη κλειδή return ώστε να διακοπεί η συνάρτηση zersvd και να λάβουμε τα μηδενικά z. Τα υπόλοιπα τμήματα του κώδικα είναι εύκολα αντιληπτά μιας και αποτελούν άμεση μετάφραση του αλγόριθμου στη γλώσσα του Matlab/GNU Octave. Τέλος, σημειώνουμε ότι η συνάρτηση zersvd δεν πραγματοποιεί έλεγχο ορθότητας των ορισμάτων της εισόδου.

```
function z = zersvd (varargin)

% polynomial degree
n = numel (varargin) - 1;

% dimensions of matrix C
[p, m] = size (varargin{1});

% E and F formation
l = p + (n - 1) * m;
k = n * m;
E = [varargin{1}, zeros(p, k - m); zeros(k - m, m), eye(k - m)];
F = [-cell2mat(varargin(2:end)); eye(k - m), zeros(k - m, m)];

while (1)
[U, S, V] = svd (E);
r = rank (E);

keyboard;

if ( r == 0 )
  z = casevii (E);
  return;
elseif ( r == l )
  if ( r == k )
    z = casevii (E, F);
    return;
  else
    f = F * V';
    Fp = f(:, end - k + r + 1 : end);
    [TU, TS, TV] = svd (Fp);
    T = TU(:, end - size(Fp, 1) + rank(Fp) + 1 : end);
    l = size (T, 1);
    k = r;
    Etmp = T * E * V';
    E = Etmp(:, 1 : r);
    Ftmp = T * F * V';
    F = Ftmp(:, 1 : r);
  end
else
  f = U' * F;
  Fp = f(end - l + r + 1: end, :);
```

```
  [WU, WS, WV] = svd (Fp);
  W = WV(:, end - size(Fp, 2) + rank(Fp) + 1 : end);
  l = r;
  k = size(W, 2);
  Etmp = U' * E * W;
  E = Etmp(1 : r, :);
  Ftmp = U' * F * W;
  F = Ftmp(1 : r, :);
end
end

function z = casevii (E, F)

if E == zeros (size (E))
  z = [];
else
  z = eig (inv(E) * F);
end
```

./code/zersvd.m

Ανάλυση Απόδοσης

Ο παρακάτω κώδικας υλοποιεί τα πειράματα απόδοσης και δε σχολιάζεται περαιτέρω.

```
N = 100; t = zeros (1, N);
for count = 1:N
  A = randn (count, count);
  B = randn (count, count);
  C = randn (count, count);
  D = randn (count, count);
  tic (); z = zersvd (A, B, C, D); t(count) = toc ();
end

plot (1:N, t, 'k', 'linewidth', 5);
xlabel 'n';
ylabel 't (sec)';
grid on;
```

./code/perftest.m

Έλεγχος Ιχνών

Ο παρακάτω κώδικας υλοποιεί απλό αλγόριθμο ο οποίος επιστρέφει 1 (αληθές) ή 0 (ψευδές) αν τα ίχνη όλων των απαιτούμενων δυνάμεων των πινάκων $D^{-1}C_{(1)}$,

$\tilde{D}^{-1}\tilde{C}_{(1)}$ είναι ίσα μεταξύ τους. Αυτό σημαίνει ότι αν η συνάρτηση επιστρέψει 1 τότε τα πολυώνυμα $C(s)$ και $s^{\ell}C(1/s)$ θα έχουν τα ίδια μηδενικά.

```
function z = condtest (varargin)

% tolerance
tol = 1e-8;

% polynomial degree
n = numel (varargin) - 1;

% dimensions of matrix C
[p, m] = size (varargin{1});

% E and F formation
l = p + (n - 1) * m;
k = n * m;

E1 = [varargin{1}, zeros(p, k - m); zeros(k - m, m), eye(k - m)];
F1 = [-cell2mat(varargin(2:end)); eye(k - m), zeros(k - m, m)];

E2 = [varargin{end}, zeros(p, k - m); zeros(k - m, m), eye(k - m)];
F2 = [-cell2mat(varargin(end-1:-1:1)); eye(k - m), zeros(k - m, m)];

sz = size (F1, 1);
p1 = zeros (1, sz);
p2 = p1;
rt = p2;

for cnt = 1:sz
  p1(cnt) = trace ( (inv (E1) * F1)^cnt );
  p2(cnt) = trace ( (inv (E2) * F2)^cnt );
  if ( abs (p1(cnt) - p2(cnt)) < tol )
    rt(cnt) = 1;
  else
    rt(cnt) = 0;
  end
end

if ( isequal (unique (rt), 1) )
  z = 1; % true: equal zeros
else
  z = 0; % false
end
```

./code/condtest.m

Β′.2 Κώδικας C#

Ανάλυση Ιδιαζόντων Τιμών

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Collections;

namespace jordan_svd
{
    /**
     * Class that solves the SVD
     */
    public class qrbsvd
    {

        public double [,] U;
        public double [,] S;
        public double[,] V;
        public double[] SA;

        // The dimensions of the input array
        protected int[] mdim;
        protected int count;

        // we define the tolerance
        protected double tol = 1.0E-13;

        protected int maxiter;

        protected double [,] qrQ;
        protected double[,] qrS;

        // The row dimension of the input matrix
        public int rowDimension;
        //The column dimension of the input matrix
        public int columnDimention;

        /**
         * The default constructor
         */
        public qrbsvd()
        {
            mdim = new int[2];
        }

        /**
```

```
         * Returns the max dimension.
         */
        protected int maxMdim()
        {
            if (mdim[0] > mdim[1])
            {
                return mdim[0];
            }
            return mdim[1];
        }

        /**
         * Creates and returns an identity matrix mXn
         */
        protected double[,] identity(int m, int n)
        {
            double[,] result = new double[m, n];
      for (int i = 0; i < m; i++)
      {
        for (int j = 0; j < n; j++)
        {
          result[i,j] = (i == j ? 1.0 : 0.0);
        }
      }
      return result;
        }

        /**
         * Returns the transpose of the specified matrix
         */
        public double[,] transpose(double[,] inm)
        {
            int n = inm.GetLength(1);
            int m = inm.GetLength(0);
            double[,] result = new double[n,m];

            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    result[j,i] = inm[i,j];
                }
            }
            return result;

        }

        /**
```

```
         * Performs the QR of the specified matrix.
         */
        public void qr(double[,] toqr)
        {
            int m = toqr.GetLength(0);
            int n = toqr.GetLength(1);
            double [] tau = new double[Math.Min(m,n)];

            alglib.ortfac.rmatrixqr(ref toqr, m, n, ref tau);
            m = toqr.GetLength(0);
            n = toqr.GetLength(1);
            int minq = Math.Min(m,n);
            qrQ = new double[minq,minq];
            alglib.ortfac.rmatrixqrunpackq ( toqr , m,n,tau, m, ref qrQ );
            qrS = new double[m, n];
            alglib.ortfac.rmatrixqrunpackr (toqr,m,n,ref qrS);

        }

        /**
         * Multiplies the specified arrays.
         */
        public double[,] mulArray(double[,] a, double[,] b)
        {
            int m = a.GetLength(0);
            int n = b.GetLength(1);
            int k = a.GetLength(1);
            double[,] c = new double[m, n];
            alglib.rmatrixgemm(m, n, k, 1, a, 0, 0, 0, b, 0, 0, 0, 0, ref c,
                0, 0);
            return c;
        }

        /**
         *  Returns the norm of the specified matrix.
         */
        protected double norm(double [] z)
        {
            double result = 0;
            for (int i = 0; i < z.Length; i++)
            {
                result = result + (z[i] * z[i]);
            }

            return Math.Sqrt(result);
        }

        /**
```

```
         * Sets and returns the diagonal of the specified matrix to 0 (triu)
         */
        protected double [,] triu (double [,] a)
        {
            int m = a.GetLength(0);
            int n = a.GetLength(1);
            double[,] result = new double[m, n];

            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    if (i == j)
                    {
                        result[i, j] = 0;
                    }
                    else
                    {
                        result[i, j] = a[i, j];
                    }
                }
            }
            return result;
        }

        /**
         * Returns matrix that contains the diagonal of the specified matrix.
         */
        public double[] diag(double[,] a)
        {
            int m = a.GetLength(0);
            int n = a.GetLength(1);
            int dg = Math.Min(m, n);
            double[] result = new double[dg];
            for (int i = 0; i < dg; i++)
            {
                result[i] = a[i, i];
            }
            return result;
        }

        /**
         * Returns the elements of the specified matrix as vector.
         */
        protected double[] eano(double[,] a)
        {
            int m = a.GetLength(0);
            int n = a.GetLength(1);
```

```
        int total = m * n;
        double[] result = new double[total];
        int k =0;
        for (int j = 0; j < n; j++)
        {
            for ( int i= 0 ; i < m ; i++) {
                result[k] = a[i, j];
                k++;
            }
        }

        return result;
    }

    /**
     * Creates and returns a zero matrix mXn
     */
    protected double[,] zeros(int m, int n)
    {
        double[,] result = new double[m,n];
        for (int i = 0; i < m; i++)
        {
            for (int j = 0; j < n; j++)
            {
                result[i, j] = 0;
            }
        }
        return result;
    }

    /**
     * Multiplies the specified column with the specified value.
     */
    protected void mulColumn(double[,] a, double val , int n)
    {
        int m = a.GetLength(0);

        for (int i = 0; i < m; i++)
        {
            a[i, n] = a[i, n] * val;
        }
    }

    /**
     * Makes canonical the S.
     */
    protected void createSFromAlg(double[] s)
    {
```

```
            S = new double[mdim[0], mdim[1]];
            for (int i = 0; i < mdim[0]; i++)
            {
                S[i, i] = s[i];
            }
        }

        /**
         * Solves the SVD using the Alg library.
         */
        public void solveAlgSvd(double[,] A)
        {
            int m = A.GetLength(0);
            int n = A.GetLength(1);

            mdim[0] = m;
            mdim[1] = n;

            double[] w;
            double[,] u;
            double[,] vt;


            alglib.rmatrixsvd(A, m, n, 2, 2, 2, out w, out u, out vt);

            this.U = u;
            this.V = this.transpose(vt);
            this.SA = w;

            createSFromAlg(SA);

        }

        /**
         * Solves the SVD.
         */
        public void solveSvd(double[,] A)
        {

            double err = 1.0E308;

            mdim[0] = A.GetLength(0);
            mdim[1] = A.GetLength(1);

            rowDimension = mdim[0];
            columnDimention = mdim[1];

            maxiter = 100 * maxMdim();
```

```
            count = 0;

            U = identity(mdim[0], mdim[0]);
            S = transpose(A);
            V = identity(mdim[1], mdim[1]);

            double[,] Q;

            while ((err > tol) && (count < maxiter))
            {
                qr(transpose(S));
                S = qrS;
                Q = qrQ;
                U = mulArray(U, Q);
                qr(transpose(S));
                S = qrS;
                Q = qrQ;
                V = mulArray(V, Q);

                double[,] e;
                e = triu(S);
                double E , F;
                E = norm(eano(e));
                F = norm(diag(S));

                if (F == 0)
                {
                    F = 1.0;
                }

                err = E / F;
                count = count + 1;
            } // end while tol and count


            double[] D;
            D = diag(S);
            S = zeros(mdim[0], mdim[1]);
            double dn;
            for (int n = 0; n < D.Length; n++)
            {
                dn = D[n];
                S[n, n] = Math.Abs(dn);
                if (dn < 0)
                {
                    mulColumn(U,(-1),n);
                }
            }
```

```

        }


        /**
         * Returns the Rank
         */
        public int Rank()
        {
            int r = 0;
            int m = S.GetLength(0);
            int n = S.GetLength(1);
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    if (S[i,j] > tol)
                    {
                        r++;
                    }

                }
            }
            return r;
        }
    }
}
```

./code/qrbsvd.cs

Υπολογισμός Μηδενικών

```
o))'Wusing System;
using System.Collections.Generic;
using System.Text;
using System.Collections;

namespace jordan_svd
{
    /**
     * Zero polynomial matrix calculation
     */
    public class ZeroPolynomialMatrixCalculation
    {
        /**
         * The input elements. An array list of input 2D arrays
         */
```

```
        public ArrayList Cn;

        /**
         * The numper of lines of the input array
         */
        public int pGrames;

        /**
         * The number of columns of the input array
         */
        public int mStiles;


        public int nValue;

        public int lamda;
        public int kapa;

        public double [,]E;
        public double[,] F;


        /**
         * The time of the algorithm execution
         */
        public long totalTime;

        /**
         * The real part of the result
         */
        public double[] resultReal;

        /**
         * The imaginary part of the result.
         */
        public double[] resultIm;

        /**
         * The default constructor
         */
        public ZeroPolynomialMatrixCalculation()
        {
        }

        /**
         * Initialises the polynomial matrix
         */
        public void setPolionimikoPinaka(ArrayList p)
```

```
        {
            Cn = p;
        }

        /**
         * Solves the polynomial matrix
         */
        public void solve()
        {
            jordan_svd.qrbsvd Fpsvd;

            // For the time calculation
            PerformanceCounter ct = new PerformanceCounter();
            ct.startCount();

            // The first array
            double[,] first;

            first = (double[,])Cn[0];

            // We get the line and the column dimension of the matrix
            pGrames = first.GetLength(0);
            mStiles = first.GetLength(1);

            nValue = Cn.Count - 1;

            lamda = pGrames + (nValue - 1) * mStiles;
            kapa = mStiles * nValue;

            // we generate the initial E and F
            generateEandF();

            jordan_svd.qrbsvd Esvd;
            int r;
            double[,] fmikro;
            double[,] Fp;


            for (;;)
            {
                Esvd = new jordan_svd.qrbsvd();
                Esvd.solveAlgSvd(E);
                r = Esvd.Rank();
                if (r == 0)
                {
                    general_case(GMatrix.GetClone(F), GMatrix.GetClone(E) ,
                        out resultReal , out resultIm);
                    break;
```

```
                }
                else if (r == lamda)
                {
                    if (r == kapa)
                    {
                        general_case(GMatrix.GetClone(F), GMatrix.GetClone(E)
                            , out resultReal, out resultIm);
                        break;

                    }
                    else // r != kapa
                    {
                        // we get the V
                        double[,] V1 = GMatrix.GetClone(Esvd.V);
                        double[,] V1Trans = GMatrix.Transpose(V1);

                        fmikro = GMatrix.Multiply(F, V1Trans);

                        int fmrowd = fmikro.GetLength(0);
                        int fmcold = fmikro.GetLength(1);
                        Fp = GMatrix.GetMatrix2(0, fmrowd, fmcold - (kapa - r
                            ), fmcold, fmikro);

                        Fpsvd = new jordan_svd.qrbsvd();
                        Fpsvd.solveAlgSvd(Fp);

                        int fprowd = Fp.GetLength(0);
                        int num = fprowd - Fpsvd.Rank();

                        double[,] TU = GMatrix.GetClone(Fpsvd.U);
                        int turowd = TU.GetLength(0);
                        int tucold = TU.GetLength(1);
                        double[,] T = GMatrix.GetMatrix2(0, turowd, (tucold -
                             num), tucold,TU);
                        lamda = T.GetLength(0);//T.RowDimension;
                        kapa = r;
                        double[,] Etmp = GMatrix.Multiply(GMatrix.Multiply(T,
                             E), V1Trans);

                        E = GMatrix.GetMatrix2(0, Etmp.GetLength(0), 0, r,
                            Etmp);

                        double[,] Ftmp = GMatrix.Multiply(GMatrix.Multiply(T,
                             F), V1Trans);
                        F = GMatrix.GetMatrix2(0, Ftmp.GetLength(0), 0, r,
                            Ftmp);
                        //sel 3
                    }
```

```
                } // end *1 r == lamda
                else  // sel 3
                {
                    double[,] U1 = GMatrix.GetClone(Esvd.U);
                    double[,] U1Trans = GMatrix.Transpose(U1);

                    fmikro = GMatrix.Multiply(U1Trans, F);

                    int fmikrorowd = fmikro.GetLength(0);
                    int fmikrocold = fmikro.GetLength(1);

                    Fp = GMatrix.GetMatrix2(fmikrorowd - (lamda - r),
                        fmikrorowd, 0, fmikrocold,fmikro);

                    Fpsvd = new jordan_svd.qrbsvd();
                    Fpsvd.solveAlgSvd(Fp);

                    int num = Fp.GetLength(1) - Fpsvd.Rank();
                    double[,] WV = GMatrix.GetClone(Fpsvd.V);

                    int wvrowd = WV.GetLength(0);
                    int wvcold = WV.GetLength(1);
                    double[,] W = GMatrix.GetMatrix2(0, wvrowd, wvcold - num,
                         wvcold,WV);

                    lamda = r;
                    kapa = W.GetLength(1);

                    double[,] Etmp = GMatrix.Multiply(GMatrix.Multiply(
                        U1Trans, E), W);

                    E = GMatrix.GetMatrix2(0, r, 0, Etmp.GetLength(1),Etmp);

                    double[,] Ftmp = GMatrix.Multiply(GMatrix.Multiply(
                        U1Trans, F), W);
                    F = GMatrix.GetMatrix2(0, r, 0, Ftmp.GetLength(1), Ftmp);
                }
            }
            // We stop the time calculation
            ct.stopCount();
            totalTime = ct.totalPref;
        }

        /**
         * Generates the first E and F
         */
        public void generateEandF()
        {
```

```
            double[,] first;
            double[,] trigonikos;
            int miden1Rows ,miden1Cols ;
            int miden2Rows , miden2Cols;
            first = (double[,])Cn[0];

            miden1Rows = pGrames;
            miden1Cols = kapa - mStiles;

            miden2Rows = kapa - mStiles;
            miden2Cols = mStiles;

            int gramesE = pGrames + miden2Rows;
            int stilesE = mStiles + miden2Rows;

            E = GMatrix.Zeroes(gramesE, stilesE);

            trigonikos = GMatrix.Identity((kapa - mStiles), (kapa - mStiles))
                ;


            // we set the Cn values
            GMatrix.SetMatrix0(0, pGrames, 0, mStiles, E, first);

            // midenikos -1
            for (int i = 0; i < pGrames; i++)
            {
                for (int j = mStiles; j < stilesE; j++)
                {
                    E[i, j] = 0.0;
                }
            }

            // midenikos 2
            for (int i = pGrames; i < gramesE; i++)
            {
                for (int j = 0; j < miden2Rows; j++)
                {
                    //E.SetElement(i, j, 0.0);
                    E[i, j] = 0.0;
                }
            }

            // we append te trigonikos
            GMatrix.SetMatrix0(pGrames, gramesE, mStiles, stilesE, E,
                trigonikos);

            double [,] cl;
```

```
            int stilesF = mStiles * (Cn.Count -1);
            int gramesF = pGrames + miden2Rows;

            //F = new GeneralMatrix(gramesF , stilesF , 0);
            F = GMatrix.Zeroes(gramesF, stilesF);

            // mul -1 and set
            for (int i = 1; i < Cn.Count; i++)
            {
                cl = GMatrix.GetClone((double[,])Cn[i]);
                for (int ix = 0; ix < pGrames; ix++)
                {
                    for (int jx = 0; jx < mStiles; jx++)
                    {
                        cl[ix, jx] = (-1) * cl[ix, jx];
                    }
                }
                // we append the cl to F
                GMatrix.SetMatrix(0, (i - 1) * mStiles, F, cl);

            }
            GMatrix.SetMatrix(pGrames, 0,F, trigonikos);
        }


        /**
         * Checks if we have result.
         * If so ... initialize the real and the imaginary part of the result
         */
        public void general_case(double[,] mF, double[,] mE , out double []
            wr, out double [] wi)
        {
            if (isZeroMatrix(mE))
            {
                wr = null;
                wi = null;
                return;
            }

            double [,] Einv = GMatrix.GetClone(mE);
            alglib.matinvreport rep;
            int info;
            // we get the inverse
            alglib.rmatrixinverse(ref Einv,out info,out rep);
            double[,] mul = GMatrix.Multiply(Einv , mF);
            int size = mul.GetLength(0);
            wr = new double[size];
            wi = new double[size];
```

```
            double [,] vl = new double[size,size];
            double [,] vr = new double[size,size];
            // Gets the Eigenvalues. The output wr , wi is our result.
            alglib.rmatrixevd( mul , size,3, out wr,out wi,out vl,out vr);
        }

        /**
         * Returns true if the specified matrix is zero matrix.
         */
        public bool isZeroMatrix(double[,] mz)
        {
            bool result = true;
            int rowd = mz.GetLength(0);
            int cold = mz.GetLength(1);
            for (int i = 0; i < rowd; i++)
            {
                for (int j = 0; j < cold; j++)
                {
                    if (mz[i, j] != 0.0) return false;
                }
            }

            return result;
        }

        public bool condtest(ArrayList varargin)
        {
            bool result = true;

            // we define the tolerance
            double tol = 1.0E-8;


            // We get the line and the column dimension of the matrix

            int n = varargin.Count - 1;

            double[,] first = (double[,])varargin[0];
            int p = first.GetLength(0); //
            int m = first.GetLength(1);

            int l = p + (n - 1) * m;
            int k = n * m;

            // define E1 , F1
            double [,] E1 = new double[l, k];
            double [,] F1 = new double[l, k];

```

```
            // first block E1
            double[,] Cl = GMatrix.GetClone((double[,])varargin[0]);
            GMatrix.SetMatrix(0, 0, E1, Cl);
            double[,] z1 = GMatrix.Zeroes(p, k - m);
            GMatrix.SetMatrix(0, m, E1, z1);

            // second block E1
            double[,] z2 = GMatrix.Zeroes(k-m, m);
            GMatrix.SetMatrix(p, 0, E1, z2);
            GMatrix.SetMatrix(p, m, E1, GMatrix.Identity(k - m, k - m));


            //first block F1
            for ( int i = 1 ; i < varargin.Count ; i++) {
                double [,] prt = (double[,])varargin[i];
                double[,] prtM = GMatrix.MulArrayWithNum(-1, prt);
                GMatrix.SetMatrix(0, (i - 1) * m, F1 , prtM);
            }

            //second block F1
            GMatrix.SetMatrix(p, 0, F1, GMatrix.Identity(k - m, k - m));
            GMatrix.SetMatrix(p, m, F1, GMatrix.Zeroes(k - m, m));

            //////////////////////////////////////

            // define E1 , F1
            double[,] E2 = new double[1, k];
            double[,] F2 = new double[1, k];


            // first block E2
            double[,] Cl0 = GMatrix.GetClone((double[,])varargin[varargin.
                Count-1]);
            GMatrix.SetMatrix(0, 0, E2, Cl0);
            double[,] ze21 = GMatrix.Zeroes(p, k - m);
            GMatrix.SetMatrix(0, m, E2, ze21);


            // second block E2
            double[,] ze22 = GMatrix.Zeroes(k - m, m);
            GMatrix.SetMatrix(p, 0, E2, ze22);
            GMatrix.SetMatrix(p, m, E2, GMatrix.Identity(k - m, k - m));


            //first block F2
            int ki = 0;
            for (int i = varargin.Count-2; i > -1; i--)
            {
```

```
                double[,] prt = (double[,])varargin[i];
                double[,] prtM = GMatrix.MulArrayWithNum(-1, prt);
                GMatrix.SetMatrix(0, ki * m, F2, prtM);
                ki++;
            }

            //second block F2
            GMatrix.SetMatrix(p, 0, F2, GMatrix.Identity(k - m, k - m));
            GMatrix.SetMatrix(p, m, F2, GMatrix.Zeroes(k - m, m));


            int sz = F1.GetLength(0);
            double [] p1 = GMatrix.ZeroesOne(sz);
            double[] p2 = GMatrix.ZeroesOne(sz);
            double[] rt = GMatrix.ZeroesOne(sz);

            for (int cnt = 0; cnt < sz; cnt++)
            {
                double [,] E1inv = GMatrix.GetClone(E1);
                alglib.matinvreport rep;
                int info;
                // we get the inverse
                alglib.rmatrixinverse(ref E1inv,out info,out rep);
                double [,] forTrace = GMatrix.Pow( GMatrix.Multiply(E1inv ,
                    F1) , cnt);
                p1[cnt] = GMatrix.Trace(forTrace);


                double[,] E2inv = GMatrix.GetClone(E2);
                // we get the inverse
                alglib.rmatrixinverse(ref E2inv, out info, out rep);
                forTrace = GMatrix.Pow(GMatrix.Multiply(E2inv, F2), cnt);
                p2[cnt] = GMatrix.Trace(forTrace);


                if (Math.Abs(p1[cnt] - p2[cnt]) < tol)
                {
                    rt[cnt] = 1;
                }
                else
                {
                    result = false;
                }
            }


            return result;
        }
```

```
    }
}
```

./code/ZeroPolynomialMatrixCalculation.cs

Ανάλυση Απόδοσης

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Collections;

namespace jordan_svd
{
    /**
     *  Class that calculates the speed of the  Zero polynomial matrix
        calculation
     */
    public class PoliSpeedAnalisys
    {
        /**
         * The random number generator.
         */
        protected Random rnd1 = new Random();

        /**
         * The results
         */
        public long [] times;

        /**
         * The default constructor.
         */
        public PoliSpeedAnalisys()
        {

        }

        /**
         * Performs the speed calculation.
         */
        public void performCalculations(int maxSize , int vathmos , System.
            Windows.Forms.Label info)
        {
            times = new long[maxSize+1];
            vathmos++;
```

```

            long ticksPerSecond;
            PerformanceCounter ct = new PerformanceCounter();
            ticksPerSecond = ct.QueryPerformanceFrequency();


            int start = 2;
            ArrayList poli;
            for (int i = start; i < maxSize+1; i++)
            {
                poli = new ArrayList();
                for (int k = 0; k < vathmos; k++)
                {
                    double[,] mat = getRandomMatrix(i);
                    poli.Add(mat);
                }
                ZeroPolynomialMatrixCalculation polionimAlgorithm = new
                    ZeroPolynomialMatrixCalculation();
                polionimAlgorithm.setPolionimikoPinaka(poli);
                polionimAlgorithm.solve();
                info.Text = "X”X•X’EX’OO•O• " + (vathmos − 1) + "
                    X”X’HX•O•O•X•O•X• " + i + "X•" + i + "  O•O•O•X•X’WO•:" + (
                    polionimAlgorithm.totalTime / (ticksPerSecond/1000) ) ;
                System.Windows.Forms.Application.DoEvents();
                times[i] = (polionimAlgorithm.totalTime / (ticksPerSecond
                    /1000));
            }
        }


        /**
         * Reutrns randon squere matrix nXn size.
         */
        public double[,] getRandomMatrix(int size)
        {
            double[,] result = new double[size, size];

            for (int i = 0; i < size; i++)
            {
                for( int j = 0 ; j < size ; j++)
                {
                    result[i, j] = rnd1.NextDouble();
                }
            }

            return result;
        }
    }
```

```
}
```

./code/PoliSpeedAnalisys.cs

Βιβλιογραφία

[1] S.Roman. *Advanced linear algebra*, ISBN 0-387-24766-1, Springer, 2005.

[2] Θ.Τζουβάρας, Κ.Τζιρώνης. *Γραμμική άλγεβρα I & II*, ISBN 960-460-786-3, Εκδόσεις Σαββάλα, 2002.

[3] G.Strang. *Linear algebra and its applications*, ISBN 0-03-010567-6, Thomson Brooks / Cole, 2006.

[4] G.Golub, C.Van Loan. *Matrix computations*, ISBN 0-8018-5413-X, The Johns Hopkins University Press, 1996.

[5] G.W.Stewart. *On the early history of the singular value decomposition*, Report, University of Maryland, 1992.

[6] G.Strang. *Computational science and engineering*, ISBN 0-9614088-1-2, Wellesley-Cambridge Press, 2007.

[7] V.Prasolov. *Problems and theorems in linear algebra*, Providence, RI, American Mathematical Society, 1994.

[8] F.Ayres. *Schaum's outline of theory and problems of matrices*, New York, Schaum, 1962.

[9] I.Gohberg, P.Lancaster, L.Rodman. *Matrix polynomials*, ISBN 978-0-898716-81-8, SIAM, 2009.

[10] G.Golub, W.Kahan. *Calculating the singular values and pseudo-inverse of a matrix*, J. SIAM Numer. Anal., Ser.B, Vol.2, No.2, 1965.

[11] P.Lancaster. *Linearization of regular matrix polynomials*, Electron. J.

Linear Algebra, Vol.17, pp.21–27, 2008.

[12] M.Berhanu. *The polynomial eigenvalue problem*, Ph.D. Thesis, University of Manchester, 2005.

[13] Won Young Yang et al.. *Applied numerical methods using Matlab*, ISBN 0-471-6983304, Wiley-Interscience, 2005.

[14] M.S.Holzel, D.S.Bemstein. *SVD-Based Computation of zeros of polynomial matrices*, ISBN 978-1-61284-800-6, Decision and Control and European Control Conference (CDC-ECC), 50th IEEE Conference on, 2011.

[15] H.Kwakernaak. *Polynomial j-spectral factorization*, IEEE Transactions on Automatic Control, Vol.39, No. 2, 1994.

[16] A.Vardulakis. *Linear multivariable control*, ISBN 978-0471928591, Wiley-Interscience, 2005.