

Αριθμητική Γραμμική Άλγεβρα Project 3

Μιχαήλ Ανάργυρος Ζαμάγιας
ΤΠ5000

16 Ιουνίου 2020

Περιεχόμενα

1	Άσκηση 1	2
1.1	Ερώτημα	2
1.2	Απάντηση	3
2	Άσκηση 2	6
2.1	Ερώτημα	6
2.2	Απάντηση	7
3	Άσκηση 3	8
3.1	Ερώτημα	8
3.2	Απάντηση	9
4	Άσκηση 4	10
4.1	Ερώτημα	10
4.2	Απάντηση	11

Άσκηση 1

1.1 Ερώτημα

Φτιάξτε μια συνάρτηση σε Octave που να δέχεται έναν $n \times n$ πίνακα ως δεδομένα και να υπολογίζει το χαρακτηριστικό πολυώνυμο του πίνακα, τα ιδιοδιανύσματα και τις ιδιοτιμές του πίνακα, χρησιμοποιώντας τις έτοιμες εντολές του Octave. Η συνάρτηση θα πρέπει να παίρνει σαν είσοδο τον πίνακα και να δίνει σαν έξοδο ένα πίνακα με στήλες τα μοναδιαία ιδιοδιανύσματα, ένα οριζόντιο διάνυσμα με τις ιδιοτιμές στην ίδια σειρά των ιδιοδιανυσμάτων και το χαρακτηριστικό πολυώνυμο σε μορφή διανύσματος συντελεστών (αριθμητική μορφή πολυωνύμων στο Matlab και Octave). Η συνάρτηση αυτή δεν πρέπει να κάνει χρήση του συμβολικού πακέτου του Octave γιατί αυτό είναι πολύ πιο αργό από το αριθμητικό πακέτο. Δώστε τα αποτελέσματα της συνάρτησης σας στον πίνακα (1.1). Εδώ μ είναι το τελευταίο ψηφίο του αριθμού μητρώου σας. Επαληθεύονται οι σχέσεις του Vieta για αυτόν τον πίνακα;

$$(1.1) \quad A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & \mu \end{pmatrix}$$

1.2 Απάντηση

Δίνω δύο πίνακες στο πρόγραμμα, τους $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 0 \end{pmatrix}$ και $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$, επειδή ο πρώτος έχει

ορίζουσα διαφορετική από το μηδέν, άρα δεν είναι αντιστρέψιμος. Επομένως, για να δείξω ότι δουλεύει ο κώδικας χρησιμοποιώ τον δεύτερο πίνακα, ο οποίος έχει ορίζουσα ίση με το μηδέν.

Η έξοδος task1.txt του προγράμματος task1.m:

```
1 A =
2   1   2   3
3   2   4   5
4   3   5   0
5 Determinant of A does not equal to 0.
6
7
8 A =
9   1   2   3
10  4   5   6
11  7   8   9
12 Determinant of A equals to 0.
13 Vieta's formulae are valid for matrix A.
14 unit_eigenvectors =
15
16  -0.231971  -0.785830   0.408248
17  -0.525322  -0.086751  -0.816497
18  -0.818673   0.612328   0.408248
19
20 eigenvalues =
21
22   16   -1    0
23
24 characteristic_polynomial = 1*x^3 - 15*x^2 - 18*x^1 + 0
```

Το πρόγραμμα task1.m:

```
1 clc;
2 clear all;
3 close all;
4
5 # matrix A
6 A = [1, 2, 3; 2, 4, 5; 3, 5, 0];
7
8 [rws, cols] = size(A);
9 # if A square matrix, then do stuff
10 if (rws == cols)
11     disp('A =')
12     disp(A);
13     # if det(A) do not do stuff
14     if (det(A) != 0)
15         disp('Determinant of A does not equal to 0.')
16     else
17         disp('Determinant of A equals to 0.')
18         [unit_eigenvectors, eigenvalues, characteristic_polynomial] =
19         task1_function(A)
20     endif
21 else
22     disp('A is not square matrix.')
23 endif
24 disp("\n");
25
26 # matrix A
27 A = [1, 2, 3; 4, 5, 6; 7, 8, 9];
28
29 [rws, cols] = size(A);
30 # if A square matrix, then do stuff
31 if (rws == cols)
32     disp('A =')
33     disp(A);
34     # if det(A) do not do stuff
35     if (det(A) != 0)
36         disp('Determinant of A does not equal to 0.')
37     else
38         disp('Determinant of A equals to 0.')
39         [unit_eigenvectors, eigenvalues, characteristic_polynomial] =
40         task1_function(A)
41     endif
42 else
43     disp('A is not square matrix.')
44 endif
```

Η συνάρτηση lin_sys.m:

```
1 function [eves, evas, characteristic_polynomial] = task1_function(A)
2     # eigenvectors and eigenvalues at eves and evas respectively
3     [eves, evas] = eig(A, balanceOption='vector');
4     # convert eigenvector matrix to a single column
5     eves = eves(:);
6     # make each unit eigenvector
7     eve1 = eves(1: 3)/norm(eves(1: 3));
8     eve2 = eves(4: 6)/norm(eves(4: 6));
9     eve3 = eves(7: 9)/norm(eves(7: 9));
10    # make unit eigenvector to return
11    eves = [eve1, eve2, eve3];
12    # tolerance, to be used for converting -0 to 0
13    tolerance = 1.e-6;
14    # transpose evas
15    evas = evas';
16    # calculate vieta theorem with current state evas
17    v_trace = round(sum(evas));
18    v_det = round(prod(evas));
19    # multiply a 3x3 identity matrix with eigenvalues
20    evas = eye(3)*evas';
21    # calculate vieta theorem with altered evas
22    check_v_trace = round(sum(evas));
23    check_v_det = round(prod(evas));
24    # transpose evas to row
25    evas = evas';
26    # convert -0 to 0
27    evas(evas<0 & evas>-tolerance) = 0;
28    # round evas
29    evas = round(evas);
30    # set coefficients of characteristic polynomial of A to pre_polynomial
31    pre_polynomial = poly(A);
32    # convert any -0 to 0
33    pre_polynomial(pre_polynomial<0 & pre_polynomial>-tolerance) = 0;
34    # match coefficients with variable x at characteristic_polynomial
35    characteristic_polynomial = polyout(round(pre_polynomial), 'x');
36    # output message for vieta's formulae
37    if (v_trace == check_v_trace & v_det == check_v_det)
38        disp("Vieta's formulae are valid for matrix A.")
39    else
40        disp("Vieta's formulae are not valid for matrix A.")
41    endif
42 endfunction
```

Άσκηση 2

2.1 Ερώτημα

Χρησιμοποιώντας την συνάρτησή σας, βρείτε τις ιδιοτιμές και τα ιδιοδιανύσματα των A^2 , A^{-1} , $A^2 + 2A + I$. Πώς σχετίζονται οι ιδιοτιμές αυτών των πινάκων με τις ιδιοτιμές του A ; Πώς σχετίζονται τα ιδιοδιανύσματα; Γράψτε την σχέση που συνδέει αυτούς τους πίνακες με τον διαγώνιο πίνακα D που αντιστοιχεί στον A . Μπορείτε να χρησιμοποιήσετε αυτήν την σχέση για να ορίσετε τους πίνακες e^A και $\sin(A)$;

2.2 Απάντηση

Άσκηση 3

3.1 Ερώτημα

3.2 Απάντηση

Άσκηση 4

4.1 Ερώτημα

4.2 Απάντηση