Αριθμητική Γραμμική Άλγεβρα
Project 2

Μιχαήλ Ανάργυρος Ζαμάγιας
TP5000

25 Μαΐου 2020

Περιεχόμενα

Άσκηση 1

1.1 Εκφώνηση

Δίνεται το σύστημα (1.1). Γράψτε αυτό το σύστημα στην μορφή $A\bar{x} = \bar{b}$ για κατάλληλα διανύσματα $\bar{x}$ και $\bar{b}$. Εδώ μ είναι το τελευταίο ψηφίο του αριθμού μητρώου σας. Φτιάξτε ένα πρόγραμμα που να δέχεται ως δεδομένα έναν $3x3$ πίνακα A και ένα $3x1$ διάνυσμα $\bar{b}$ και να είναι σε θεση να μας πει αν το σύστημα είναι αόριστο ή αδύνατο ή έχει μοναδική λύση. Αν το σύστημα έχει μοναδική λύση πρέπει να δίνει αυτή την λύση υπολογισμένη με την μέθοδο του επαυξημένου πίνακα (απαλοιφή Gauss). Δώστε τα αποτελέσματα του προγράμματός σας στο σύστημα εξισώσεων (1.1) καθώς και στο σύστημα (1.2).

$$\begin{aligned} 3x + 4y + 2z &= 3 \\ x - 2y &= 3 \\ \mu x + 3y + z &= 2 \end{aligned} \tag{1.1}$$

$$\begin{aligned} \mu x + y + z &= 3 \\ x + 2y &= \mu \\ (\mu - 1) + 3y + z &= 2 \end{aligned} \tag{1.2}$$

1.2 Απάντηση

Το (1.1) γράφεται ως (1.3) στην μορφή $A\bar{x} = \bar{b}$. Ομοίως, το (1.2) γράφεται ως (1.4), με $\mu = 0$ για αριθμό μητρώου 5000.

$$\underbrace{\begin{pmatrix} 3 & 4 & 2 \\ 1 & -2 & 0 \\ 0 & 3 & 1 \end{pmatrix}}_{\text{A}} \underbrace{\begin{pmatrix} x \\ y \\ z \end{pmatrix}}_{\bar{x}} = \underbrace{\begin{pmatrix} 3 \\ 3 \\ 2 \end{pmatrix}}_{\bar{b}} \tag{1.3}$$

$$\underbrace{\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \\ -1 & 3 & 1 \end{pmatrix}}_{\text{A}} \underbrace{\begin{pmatrix} x \\ y \\ z \end{pmatrix}}_{\bar{x}} = \underbrace{\begin{pmatrix} 3 \\ 0 \\ 2 \end{pmatrix}}_{\bar{b}} \tag{1.4}$$

Η έξοδος του προγράμματός μου για τα συστήματα (1.3) και (1.4):

```
A =
   3   4   2
   1  -2   0
   0   3   1
b =
   3
   3
   2

The system has a unique non-trivial solution:
x = -2.0000
y = -2.5000
z =  9.5000


A =
   0   1   1
   1   2   0
  -1   3   1
b =
   3
   0
   2

The system has a unique non-trivial solution:
x =  0.50000
y = -0.25000
z =  3.2500
```

Το κύριο πρόγραμμα:

```
clc;
clear all;
close all;

#   matrix A with system coefficients
A = [3, 4, 2; 1, -2, 0; 0, 3, 1];
#   matrix b with system constants
b = [3; 3; 2];
#   print A and b
disp("A =");
disp(A);
disp("b =");
disp(b);
#   run first task's logic
function_task1(A, b)

#   add new line to seperate outputs
disp("\n")

#   matrix A with system coefficients
A = [0, 1, 1; 1, 2, 0; -1, 3, 1];
#   matrix b with system constants
b = [3; 0; 2];
#   print A and b
disp("A =");
disp(A);
disp("b =");
disp(b);
#   run first task's logic
function_task1(A, b)
```

Η *συνάρτηση* function_task1:

```
function function_task1(A, b)
    #   calulcate A's determinant
    det_A = det(A);
    #   add empty line
    disp("");
    #   check if the system has a unique non-trivial solution
    #   check if A's determinant is non-zero
    if (det_A != 0)
        disp("The system has a unique non-trivial solution:")
        #   calculate system's unique non-trivial solution
        #   using Gaussian elimination
        Axb_roots = A\b;
        #   convert negative zeros to zeros (-0 to 0)
        Axb_roots(Axb_roots == 0) = 0;
        #   diplay solution
        x = Axb_roots(1)
        y = Axb_roots(2)
        z = Axb_roots(3)
    else
        #   get the size of b matrix
        n = size(b)(1);
        #   calculate Cramer's determinants for the system
        x_up = A;
        y_up = A;
        z_up = A;
        for i = 1 : n
            x_up(i, 1) = b(i);
        end
        det_x = det(x_up)/det_A;
        for i = 1 : n
            y_up(i, 2) = b(i);
        end
        det_y = det(y_up)/det_A;
        for i = 1 : n
            z_up(i, 3) = b(i);
        end
        det_z = det(z_up)/det_A;
        #   check if all of Cramer's determinants are equal to zero,
        #   then the system has infinite solutions
        if (det_x == 0 && det_y == 0 && det_z == 0)
            disp("The system has infinite solutions.")
        #   if not all of Cramer's determinants are equal to zero,
        #   then the system has no non-trivial solutions
        else
            disp("The system has no non-trivial solutions.")
        endif
    endif
end
```

Άσκηση 2

2.1 Εκφώνηση

Φτιάξτε ένα πρόγραμμα για την αντιστροφή ενός πίνακα A με την μέθοδο Gauss, προσαρτώντας δηλαδή τον ταυτοτικό πίνακα Α και ακολουθώντας τον αλγόριθμο της απαλοιφής Gauss. Εξηγήστε γιατί με αυτόν τον τρόπο παίρνετε τον αντίστροφο πίνακα. Το πρόγραμμά σας πρέπει να είναι σε θέση να βρίσκει αν ο πίνακας A είναι αντιστρέψιμος. Εφαρμόστε το πρόγραμμά σας στον πίνακα (2.1).

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ -1 & 1 & \mu \end{pmatrix} \tag{2.1}$$

2.2 Απάντηση

Ο πίνακας (2.1) γράφεται ως (2.2) για $\mu = 0$.

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ -1 & 1 & 0 \end{pmatrix} \tag{2.2}$$

Η έξοδος του προγράμματός μου για τον πίνακα (2.2):

```
Matrix A is non-invertible.
```

Το πρόγραμμα:

```
clc;
clear all;
close all;

#   square matrix A
A = [1, 2, 1; 2, 1, 0; -1, 1, 0];
#   make an identity matrix the size of A
I_M = eye(size(A)(1));
#   apply gaussian elimination to A and I
B = round(A\I_M);
#   convert negative zeros to zeros (-0 to 0)
B(B == 0) = 0;

#   check if A is invertible based on the relation A*A^-1=I,
#   check if A*B is equal to I, where B=A^-1
if (isequal(A*B, I_M) == 1)
    disp("Matrix A is invertible.")
    disp("A =");
    disp(A);
    disp("B = A\\I");
    disp(B);
else
    disp("Matrix A is non-invertible.")
endif
```

Άσκηση 3

3.1 Εκφώνηση

Θεωρήστε τα διανύσματα του (3.1), όπου μ είναι το τελευταίο ψηφίο του αριθμού μητρώου σας. Ελέγξτε με το χέρι αν τα παραπάνω διανύσματα είναι γραμμικά ανεξάρτητα ή όχι χωρίς την χρήση του κριτηρίου της ορίζουσας. Αποτελούν αυτά τα διανύσματα βάση του R^3; Κατόπιν φτιάξτε ένα πρόγραμμα που να αποφασίζει αν μία οποιαδήποτε τριάδα διανυσμάτων στο R^3 αποτελεί βάση του R^3 κάνοντας χρήση του κριτηρίου της ορίζουσας.

$$\begin{aligned} u &= 3i - 4j + 5k \\ v &= 2i - 3j + k \\ w &= i - j + \mu k \end{aligned} \tag{3.1}$$

3.2 Απάντηση

Το σύστημα διανυσμάτων (3.1) γράφεται ως (3.2) για $\mu = 0$.

$$\begin{aligned} u &= 3i - 4j + 5k \\ v &= 2i - 3j + k \\ w &= i - j + 0k \end{aligned} \tag{3.2}$$

Άσκηση 4

4.1 Εκφώνηση

Θεωρήστε τα διανύσματα του (4.1). Δείξτε ότι αυτά τα διανύσματα αποτελούν βάση του R^3. Εφαρμόστε με το χέρι τον αλγόριθμο Gram-Smith για την ορθοκανονικοποίηση της βάσης. Φτιάξτε ένα πρόγραμμα που θα δέχεται σαν είσοδο τρία διανύματα σαν στήλες ενός $3x3$ πίνακα, θα ελέγχει αν αυτά τα διανύσματα αποτελούν βάση του R^3, και κατόπιν θα εφαρμόζει τον αλγόριθμο Gram-Smith και θα δίνει σαν έξοδο μία ορθοκανονική βάση.

$$
\begin{aligned}
u &= i - 2j + 3k \\
v &= 2i - j + 2k \\
w &= i - 2j + k
\end{aligned} \tag{4.1}
$$

4.2 Απάντηση

Πηγές

5.1 Βιβλία

5.2 Video

- Inverse Matrix Using Gauss-Jordan
- Solving a 3x3 System Using Cramer's Rule
- Εύρεση αντίστροφου πίνακα με μέθοδο Gauss

5.3 Σύνδεσμοι

- Systems of Linear Equations
- Γραμμικά συστήματα Εξισώσεων
- 11. Solving Ax = b, Matrix division and the slash operator
- Gaussian elimination calculator
- Cramer's Rule
- Inserting code in a LaTeXdocument

5.4 Χρήσιμα αρχεία