5
Γραμμικά Συστήματα με το MATLAB

Δημήτριος Χριστόπουλος[1,2]

[1]Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Οικονομικών Επιστημών
[2]dchristop@econ.uoa.gr

Άνοιξη 2011

Σημειώσεις Εργαστηρίου Γραμμικών Μαθηματικών[3].

[3]Οι ηλεκτρονικές σημειώσεις που ακολουθούν περιέχουν υπερσυνδέσεις, με ένα απλό κλικ, εσωτερικά ή εξωτερικά του κειμένου.

Περιεχόμενα

1 Γραμμικά Συστήματα με το MATLAB

Θα εξετάσουμε την γενική μορφή ενός γραμμικού συστήματος $m \times n$, δηλ. m εξισώσεων με n αγνώστους:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{m,n}x_n &= b_m \end{aligned} \tag{1}$$

το οποίο σαν εξίσωση πινάκων γράφεται ως $Ax = b$ ή πιο αναλυτικά:

$$\underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}}_{b} \tag{2}$$

Θα δούμε εάν μπορούμε να λάβουμε την λύση του συστήματος 1 ή 2 με το MATLAB άμεσα ή έστω σε πεπερασμένο αριθμό βημάτων που εμείς θα ορίσουμε. Η γενική θεωρία των γραμμικών συστημάτων προκειμένου να εξετάσει εάν το σύστημα 1 έχει λύση εξετάζει εάν το διάνυσμα $b \in R^m$ ανήκει στον υπόχωρο του R^m που παράγουν οι στήλες του πίνακα A, δηλ. θεωρεί το σύστημα σαν την ακόλουθη διανυσματική αναπαράσταση:

$$\underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}}_{b} = x_1 \cdot \underbrace{\begin{pmatrix} a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{m,1} \end{pmatrix}}_{A_1} + x_2 \cdot \underbrace{\begin{pmatrix} a_{1,2} \\ a_{2,2} \\ \vdots \\ a_{m,2} \end{pmatrix}}_{A_2} + \cdots + x_n \cdot \underbrace{\begin{pmatrix} a_{1,n} \\ a_{2,n} \\ \vdots \\ a_{m,n} \end{pmatrix}}_{A_n} \tag{3}$$

Επομένως όταν γνωρίζουμε τον χώρο στηλών ενός πίνακα μπορούμε και να εξετάσουμε εάν το διάνυσμα στήλη των δεξιών μερών των εξισώσεων ανήκει στον υπόχωρο του R^m ο οποίος παράγεται από αυτές τις στήλες. Στην πράξη βέβαια τα πράγματα είναι πολύ πιο απλά. Το μόνο που έχουμε να κάνουμε είναι να ορίσουμε τον *επαυξημένο πίνακα*:

$$A|b = \left(\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & b_m \end{array}\right) \tag{4}$$

Κατόπιν βρίσκουμε την ανηγμένη κλιμακωτή μορφή αυτού και τότε βλέπουμε πως αναπαρίσταται το b ως γραμμικός συνδυασμός των στηλών $A_i, i = 1, \ldots, n$ του πίνακα A. Εάν όμως κάπου ενδιάμεσα δούμε μία γραμμή της μορφής:

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & | & \chi \end{pmatrix} \tag{5}$$

με $\chi \neq 0$, τότε το σύστημα είναι αδύνατο. Επίσης κάθε φορά που βλέπουμε μία μηδενική γραμμή, συτό σημαίνει ότι υπάρχει ένας ελεύθερος άγνωστος.

1.1 Τετραγωνικά Συστήματα

Όταν έχουμε $m = n$ τότε το σύστημα λέγεται τετραγωνικό και ομοίως τετραγωνικός είναι και ο πίνακας A του συστήματος. Η επίλυση διευκολύνεται λόγω της ύπαρξης αρκετών παραγοντοποιήσεων για τους τετραγωνικούς πίνακες.

1.1.1 Ομογενή τετραγωνικά συστήματα

Γνωρίζουμε ότι ένα τετραγωνικό ομογενές σύστημα έχει μη μηδενική λύση μόνον όταν ο πίνακας είναι μη αντιστρέψιμος, δηλ. μόνον όταν η ορίζουσά του είναι μηδέν. Διαφορετικά το σύστημα έχει μοναδική λύση που είναι η μηδενική. Στην περίπτωση των ομογενών συστημάτων η μόνη έτοιμη εντολή που μπορούμε να χρησιμοποιήσουμε στο MATLAB είναι η εντολή null(A), η οποία δίνει μία βάση για τον πυρήνα του A. Θεωρούμε έναν πίνακα με την πρώτη και την τρίτη στήλη του ίδιες, ώστε να έχει οπωσδήποτε μηδενική ορίζουσα και υπολογίζουμε την λύση του αντίστοιχου ομογενούς συστήματος με το MATLAB:

$$\begin{aligned} 8x + y + 8z &= 0 \\ 3x + 5y + 3z &= 0 \\ 4x + 9y + 4z &= 0 \end{aligned} \tag{6}$$

Στο MATLAB γράφουμε και παίρνουμε τα ακόλουθα αποτελέσματα:

```
>> A=[8,1,8;3,5,3;4,9,4]
A =
     8     1     8
     3     5     3
     4     9     4
>> x=null(A)
x =
   0.707106781186548
                   0
  -0.707106781186548
>> A*x
```

```
ans =
  1.0e-015 *
   0.888178419700125
                   0
   0.444089209850063
```

Το MATLAB δίνει μία ορθοκανονική βάση[1] για τον πυρήνα του πίνακα A. Ένας έμπειρος αναγνώστης θα αναγνωρίσει στην ανωτέρω λύση του MATLAB το διάνυσμα:

$$x = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Αυτήν την λύση μπορούμε να λάβουμε άμεσα με το wxMaxima γράφοντας τις εντολές που ακολουθούν:

```
(%i1) A:matrix([8,1,8],[3,5,3],[4,9,4])$
      'A=A;
      Kernel('A)=nullspace(A);
      x:args(nullspace(A))[1]$
      'x=x;
      'A.'x=A.x;
      xn:x/mat_norm(x,frobenius)$
      'xn=xn;
      'A.'xn=A.xn;
```

$$(\%o2)\quad A=\begin{bmatrix} 8 & 1 & 8 \\ 3 & 5 & 3 \\ 4 & 9 & 4 \end{bmatrix}$$

$$(\%o3)\quad \mathrm{Kernel}(A)=\mathrm{span}\left(\begin{bmatrix} -37 \\ 0 \\ 37 \end{bmatrix}\right)$$

$$(\%o5)\quad x=\begin{bmatrix} -37 \\ 0 \\ 37 \end{bmatrix}$$

$$(\%o6)\quad A\ .\ x=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$(\%o8)\quad xn=\begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$(\%o9)\quad A\ .\ xn=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Σχήμα 1: Λύση ομογενούς συστήματος με το wxMaxima

[1] Ένα σύνολο διανυσμάτων $\{v_i, i = 1, \ldots, n\}$ λέγεται ορθοκανονικό όταν $v_i \cdot v_j = 0, i \neq j$ και $v_i \cdot v_i = 1$, όπου με τελίτσα συμβολίζουμε το εσωτερικό γινόμενο

Μπορούμε να λύσουμε ένα ομογενές σύστημα στο MATLAB κάνοντας απαλοιφή Gauss-Jordan με μερική οδήγηση γραμμών, ώστε το σύστημα κάθε φορά να είναι ισοδύναμο με το προηγούμενο. Για τον σκοπό αυτό δημιουργούμε την συνάρτηση με όνομα showechelon.m, η οποία λαμβάνει σαν όρισμα τον οποιοδήποτε πίνακα A και επιστρέφει την ανηγμένη κλιμακωτή μορφή του A. Ο πλήρης κώδικας ακολουθεί:

```
function B=showechelon(A)
%Shows all the steps of Gauss-Jordanian elimination
%with partial row pivoting
%Call: B=showechelon(A)
%Demetrios T. Christopoulos, dchristop@econ.uoa.gr, Spring 2011
format rat
format compact
more off
disp('Let the matrix be:')
A
[m,n] = size(A);
% Compute the numerical tolerance.
tol = max([m,n])*eps*norm(A,'inf');
% Loop over the entire matrix.
i = 1;
j = 1;
k = 0;
while (i <= m) && (j <= n)
   % Find the pivot element of column j.
     [mx,k] = max(abs(A(i:m,j)));
      k = k+i-1;
   if (mx <= tol)
      % The column is negligible, zero it out.
      disp([' The  column ' int2str(j) '
      is numerically negligible'])
      A(i:m,j) = zeros(m-i+1,1)
      j = j + 1;
   else
       if (i ~= k)
         % Do partial pivoting: Swap i-th and k-th rows.
         disp([' Do partial row pivoting:
         Swap rows ' int2str(i) ' and ' int2str(k) ' : ' ])
         disp([' Row(' int2str(i) ') <--> Row(' int2str(k) ') '])
         A([i k],:) = A([k i],:)
      end

      %Make the pivot element equal to unity
```

```
      %by dividing the pivot row with the pivot element.
      disp([' Make pivot element A(' int2str(i) ',' int2str(j) ')
       equal to unity:' ])
      disp([' Row(' int2str(i) ') --> Row(' int2str(i) ')
      / A(' int2str(i) ',' int2str(j) ') '])
      A(i,j:n) = A(i,j:n)/A(i,j)
      %Construct the unit vector at column j
      %by subtracting proper multiples of the pivot row
      from all other rows.
      disp([' Do  eliminations in column ' int2str(j) ' : '])
       for k = 1:m
         if k ~= i
            disp([' Row(' int2str(k) ') --> Row(' int2str(k) ')
            - A(' int2str(k) ',' int2str(j) ') * Row(' int2str(j) ')'])
            A(k,j:n) = A(k,j:n) - A(k,j)*A(i,j:n)
         end
      end
      i = i + 1;
      j = j + 1;
   end
end
B=A;
```

Σαν εφαρμογή λύνουμε το προηγούμενο πρόβλημα:

```
>> format compact
>> format rat
>> A=[8,1,8;3,5,3;4,9,4],b=zeros(3,1)
A =
       8              1              8
       3              5              3
       4              9              4
b =
       0
       0
       0
>> Ab=[A,b]
Ab =
       8              1              8              0
       3              5              3              0
       4              9              4              0
>> C=showechelon(Ab);
Let the matrix be:
A =
```

```
       8              1              8              0
       3              5              3              0
       4              9              4              0
 Make pivot element A(1,1) equal to unity:
 Row(1) --> Row(1) / A(1,1)
A =
       1             1/8             1              0
       3              5              3              0
       4              9              4              0
 Do  eliminations in column 1 :
 Row(2) --> Row(2) - A(2,1) * Row(1)
A =
       1             1/8             1              0
       0            37/8             0              0
       4              9              4              0
 Row(3) --> Row(3) - A(3,1) * Row(1)
A =
       1             1/8             1              0
       0            37/8             0              0
       0            17/2             0              0
 Do partial row pivoting: Swap rows 2 and 3 :
 Row(2) <--> Row(3)
A =
       1             1/8             1              0
       0            17/2             0              0
       0            37/8             0              0
 Make pivot element A(2,2) equal to unity:
 Row(2) --> Row(2) / A(2,2)
A =
       1             1/8             1              0
       0              1              0              0
       0            37/8             0              0
 Do  eliminations in column 2 :
 Row(1) --> Row(1) - A(1,2) * Row(2)
A =
       1              0              1              0
       0              1              0              0
       0            37/8             0              0
 Row(3) --> Row(3) - A(3,2) * Row(2)
A =
       1              0              1              0
       0              1              0              0
       0              0              0              0
```

```
 The  column 3 is numerically negligible
A =
       1             0             1             0
       0             1             0             0
       0             0             0             0
 The  column 4 is numerically negligible
A =
       1             0             1             0
       0             1             0             0
       0             0             0             0
```

Βλέπουμε ότι το αρχικό ομογενές σύστημα είναι ισοδύναμο με το άμεσα επιλύσιμο σύστημα που ακολουθεί:

$$\left\{\begin{array}{rcl} x+z & = & 0 \\ y & = & 0 \\ 0 & = & 0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{rcl} x & = & -z \\ y & = & 0 \\ z & = & z \end{array}\right\} \tag{7}$$

Συνεπώς μία λύση είναι:

$$x = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

και αν την κανονικοποιήσουμε διαιρώντας με το μήκος του διανύσματος, δηλ. με $\sqrt{2}$, λαμβάνουμε την λύση που έδωσε και το wxMaxima.

Παράδειγμα 1.1. *Να βρεθεί με την βοήθεια του MATLAB, κάνοντας απαλοιφή Gauss-Jordan με μερική οδήγηση, η γενική λύση του ομογενούς συστήματος:*

$$\left\{\begin{array}{rcl} 2x+3y-z+w & = & 0 \\ 8x+12y-9z+8w & = & 0 \\ 4x+6y+3z-2w & = & 0 \\ 2x+3y+9z-7w & = & 0 \end{array}\right\} \tag{8}$$

Λύση:

Χρησιμοποιούμε την συνάρτηση showechelon.m και λαμβάνουμε:

```
>> A=[2,3,-1,1;8,12,-9,8;4,6,3,-2;2,3,9,-7],b=zeros(4,1)
A =
       2             3            -1             1
       8            12            -9             8
       4             6             3            -2
```

```
       2             3             9            -7
b =
       0
       0
       0
       0
>> Ab=[A,b]
Ab =
       2             3            -1             1             0
       8            12            -9             8             0
       4             6             3            -2             0
       2             3             9            -7             0
>> Ab=[A,b]
Ab =
       2             3            -1             1             0
       8            12            -9             8             0
       4             6             3            -2             0
       2             3             9            -7             0
>> C=showechelon(Ab);
Let the matrix be:
A =
       2             3            -1             1             0
       8            12            -9             8             0
       4             6             3            -2             0
       2             3             9            -7             0
 Do partial row pivoting: Swap rows 1 and 2 :
 Row(1) <--> Row(2)
A =
       8            12            -9             8             0
       2             3            -1             1             0
       4             6             3            -2             0
       2             3             9            -7             0
 Make pivot element A(1,1) equal to unity:
 Row(1) --> Row(1) / A(1,1)
A =
       1             3/2          -9/8            1             0
       2             3            -1             1             0
       4             6             3            -2             0
       2             3             9            -7             0
 Do  eliminations in column 1 :
 Row(2) --> Row(2) - A(2,1) * Row(1)
A =
       1             3/2          -9/8            1             0
```

```
       0              0              5/4           -1              0
       4              6              3             -2              0
       2              3              9             -7              0
 Row(3) --> Row(3) - A(3,1) * Row(1)
A =
       1              3/2           -9/8            1              0
       0              0              5/4           -1              0
       0              0             15/2           -6              0
       2              3              9             -7              0
 Row(4) --> Row(4) - A(4,1) * Row(1)
A =
       1              3/2           -9/8            1              0
       0              0              5/4           -1              0
       0              0             15/2           -6              0
       0              0             45/4           -9              0
 The  column 2 is numerically negligible
A =
       1              3/2           -9/8            1              0
       0              0              5/4           -1              0
       0              0             15/2           -6              0
       0              0             45/4           -9              0
 Do partial row pivoting: Swap rows 2 and 4 :
 Row(2) <--> Row(4)
A =
       1              3/2           -9/8            1              0
       0              0             45/4           -9              0
       0              0             15/2           -6              0
       0              0              5/4           -1              0
 Make pivot element A(2,3) equal to unity:
 Row(2) --> Row(2) / A(2,3)
A =
       1              3/2           -9/8            1              0
       0              0              1             -4/5            0
       0              0             15/2           -6              0
       0              0              5/4           -1              0
 Do  eliminations in column 3 :
 Row(1) --> Row(1) - A(1,3) * Row(3)
A =
       1              3/2            0              1/10           0
       0              0              1             -4/5            0
       0              0             15/2           -6              0
       0              0              5/4           -1              0
 Row(3) --> Row(3) - A(3,3) * Row(3)
```

```
A =
       1              3/2            0              1/10           0
       0              0              1             -4/5            0
       0              0              0              0              0
       0              0             5/4            -1             0
 Row(4) --> Row(4) - A(4,3) * Row(3)
A =
       1              3/2            0              1/10           0
       0              0              1             -4/5            0
       0              0              0              0              0
       0              0              0              0              0
 The  column 4 is numerically negligible
A =
       1              3/2            0              1/10           0
       0              0              1             -4/5            0
       0              0              0              0              0
       0              0              0              0              0
 The  column 5 is numerically negligible
A =
       1              3/2            0              1/10           0
       0              0              1             -4/5            0
       0              0              0              0              0
       0              0              0              0              0
```

Βλέπουμε ότι το αρχικό ομογενές σύστημα είναι ισοδύναμο με το άμεσα επιλύσιμο σύστημα που ακολουθεί:

$$\left\{\begin{array}{rcc} x+\frac{3}{2}y+\frac{1}{10}w & = & 0 \\ z-\frac{4}{5}w & = & 0 \\ 0 & = & 0 \\ 0 & = & 0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{rcc} x & = & -\frac{3}{2}y-\frac{1}{10}w \\ y & = & y \\ z & = & \frac{4}{5}w \\ w & = & w \end{array}\right\} \tag{9}$$

1.1.2 Μη ομογενή τετραγωνικά συστήματα

Όταν επιπλέον ισχύει ότι $b \neq 0_{m\times m}$, τότε το σύστημα λέγεται μη ομογενές. Σ′ αυτήν την περίπτωση εάν ο πίνακας A είναι αντιστρέψιμος, μπορούμε άνετα να χρησιμοποιήσουμε το MATLAB για την επίλυση του συστήματος. Η λύση είναι μοναδική και προκύπτει ως εξής:

$$Ax = b \Rightarrow x = A^{-1}b \tag{10}$$

Η λύση με το MATLAB μπορεί να δοθεί με όλες τις παρακάτω εντολές:

```
>> x=inv(A)*b
>> x=A^(-1)*b
>> x=A^-1*b
>> x=A\b
```

Σαν παράδειγμα θεωρούμε τον ακόλουθο πίνακα:

```
>> A=magic(3)
A =
       8              1              6
       3              5              7
       4              9              2
>> b=[1;2;3]
b =
     1
     2
     3
>> x=inv(A)*b
x =
       1/20
       3/10
       1/20
>> x=A^(-1)*b
x =
       1/20
       3/10
       1/20
>> A^-1*A
ans =
       1              0              *
       0              1              0
       0              *              1
>> x=A\b
x =
       1/20
       3/10
       1/20
>> A*x-b
ans =
       0
       0
      -1/2251799813685248
>> format long e
>> A*x-b
```

```
ans =
                          0
                          0
   -4.440892098500626e-016
```

Όπου βλέπουμε την επίδραση των σφαλμάτων της αριθμητικής κινητής υποδιαστολής που χρησιμοποιεί το MATLAB. Σημειώνουμε ότι ο πίνακας έχει ορίζουσα $det(A) = -360$, όχι πολύ κοντά στο μηδέν. Επίσης ο *δείκτης κατάστασης*[2] είναι 4.33, όχι ιδιαίτερα μεγάλος. Επομένως ο πίνακας δεν είναι σε καμία περίπτωση ' περίπου ανώμαλος ', δηλ. κοντά στο να μην αντιστρέφεται.

Με την ίδια διαδικασία της απαλοιφής Gauss-Jordan με μερική οδήγηση μπορούμε να λύσουμε και ένα σύστημα στο οποίο ο πίνακας δεν αντιστρέφεται, όπως φαίνεται στα παραδείγματα που ακολουθούν.

Παράδειγμα 1.2. *Να επιλυθεί το σύστημα:*

$$\left\{\begin{array}{rcl} 5x-2y+8z+w & = & 12 \\ x+y+z-w & = & 2 \\ 3x-4y+6z+3w & = & 8 \\ 7x-7y+13z+5w & = & 18 \end{array}\right\} \tag{11}$$

Λύση:

Δημιουργούμε τον επαυξημένο πίνακα Ab και κατόπιν χρησιμοποιούμε την συνάρτηση showechelon.m για να λάβουμε τα αποτελέσματα:

```
>> A=[5,-2,8,1;1,1,1,-1;3,-4,6,3;7,-7,13,5]
A =
       5            -2             8             1
       1             1             1            -1
       3            -4             6             3
       7            -7            13             5
>> b=[12;2;8;18]
b =
      12
       2
       8
      18
>> Ab=[A,b]
Ab =
       5            -2             8             1            12
       1             1             1            -1             2
```

[2]Για τετραγωνικό πίνακα A ο δείκτης κατάστασης $\kappa(A)$ ορίζεται ως $\kappa(A) = ||A|| \cdot ||A^{-1}||$.

```
       3             -4              6              3              8
       7             -7             13              5             18
>> C=showechelon(Ab);
Let the matrix be:
A =
       5             -2              8              1             12
       1              1              1             -1              2
       3             -4              6              3              8
       7             -7             13              5             18
 Do partial row pivoting: Swap rows 1 and 4 :
 Row(1) <--> Row(4)
A =
       7             -7             13              5             18
       1              1              1             -1              2
       3             -4              6              3              8
       5             -2              8              1             12
 Make pivot element A(1,1) equal to unity:
 Row(1) --> Row(1) / A(1,1)
A =
       1             -1            13/7            5/7           18/7
       1              1              1             -1              2
       3             -4              6              3              8
       5             -2              8              1             12
 Do  eliminations in column 1 :
 Row(2) --> Row(2) - A(2,1) * Row(1)
A =
       1             -1            13/7            5/7           18/7
       0              2            -6/7          -12/7           -4/7
       3             -4              6              3              8
       5             -2              8              1             12
 Row(3) --> Row(3) - A(3,1) * Row(1)
A =
       1             -1            13/7            5/7           18/7
       0              2            -6/7          -12/7           -4/7
       0             -1             3/7            6/7            2/7
       5             -2              8              1             12
 Row(4) --> Row(4) - A(4,1) * Row(1)
A =
       1             -1            13/7            5/7           18/7
       0              2            -6/7          -12/7           -4/7
       0             -1             3/7            6/7            2/7
       0              3            -9/7          -18/7           -6/7
 Do partial row pivoting: Swap rows 2 and 4 :
```

```
 Row(2) <--> Row(4)
A =
       1             -1             13/7            5/7            18/7
       0              3             -9/7          -18/7            -6/7
       0             -1              3/7            6/7             2/7
       0              2             -6/7          -12/7            -4/7
 Make pivot element A(2,2) equal to unity:
 Row(2) --> Row(2) / A(2,2)
A =
       1             -1             13/7            5/7            18/7
       0              1             -3/7           -6/7            -2/7
       0             -1              3/7            6/7             2/7
       0              2             -6/7          -12/7            -4/7
 Do  eliminations in column 2 :
 Row(1) --> Row(1) - A(1,2) * Row(2)
A =
       1              0             10/7           -1/7            16/7
       0              1             -3/7           -6/7            -2/7
       0             -1              3/7            6/7             2/7
       0              2             -6/7          -12/7            -4/7
 Row(3) --> Row(3) - A(3,2) * Row(2)
A =
       1              0             10/7           -1/7            16/7
       0              1             -3/7           -6/7            -2/7
       0              0              0              0               *
       0              2             -6/7          -12/7            -4/7
 Row(4) --> Row(4) - A(4,2) * Row(2)
A =
       1              0             10/7           -1/7            16/7
       0              1             -3/7           -6/7            -2/7
       0              0              0              0               *
       0              0              *              0               *
 The  column 3 is numerically negligible
A =
       1              0             10/7           -1/7            16/7
       0              1             -3/7           -6/7            -2/7
       0              0              0              0               *
       0              0              0              0               *
 The  column 4 is numerically negligible
A =
       1              0             10/7           -1/7            16/7
       0              1             -3/7           -6/7            -2/7
       0              0              0              0               *
```

```
       0              0              0              0              *
 The  column 5 is numerically negligible
A =
       1              0             10/7           -1/7           16/7
       0              1             -3/7           -6/7           -2/7
       0              0              0              0              0
       0              0              0              0              0
```

Εάν δεν θέλαμε τόση πολλή λεπτομέρεια, αλλά κατευθείαν το αποτέλεσμα θα γράφαμε απλώς:

```
>> C=rref(Ab)
C =
       1              0             10/7           -1/7           16/7
       0              1             -3/7           -6/7           -2/7
       0              0              0              0              0
       0              0              0              0              0
```

Τώρα είναι φανερό ότι το αρχικό σύστημα 11 είναι ισοδύναμο με το σαφώς απλούστερο σύστημα:

$$\left\{\begin{array}{rcc} x+\frac{10}{7}z-\frac{1}{7}w & = & \frac{16}{7} \\ y-\frac{3}{7}z-\frac{6}{7}w & = & -\frac{2}{7} \\ 0 & = & 0 \\ 0 & = & 0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{rcc} x & = & \frac{16}{7}-\frac{10}{7}z+\frac{1}{7}w \\ y & = & -\frac{2}{7}+\frac{3}{7}z+\frac{6}{7}w \\ z & = & z \\ w & = & w \end{array}\right\} \tag{12}$$

το οποίο μπορεί να γραφεί επίσης διανυσματικά ως εξής:

$$\begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = \begin{pmatrix} \frac{16}{7} \\ -\frac{2}{7} \\ 0 \\ 0 \end{pmatrix} + z \cdot \begin{pmatrix} -\frac{10}{7} \\ \frac{3}{7} \\ 1 \\ 0 \end{pmatrix} + w \cdot \begin{pmatrix} \frac{1}{7} \\ \frac{6}{7} \\ 0 \\ 1 \end{pmatrix} \tag{13}$$

Εάν ψάχναμε την λύση του αντίστοιχου ομογενούς συστήματος θα βρίσκαμε τους δύο τελευταίους όρους της λύσης, ενώ ο πρωτος όρος είναι η μερική λύση του μη ομογενούς συστήματος. Επίσης πρέπει να παρατηρήσουμε ότι το MATLAB δεν κατάφερε να υπολογίσει την ορίζουσα του πίνακα A ως μηδέν, αλλά έδωσε μία πολύ μικρή τιμή γι΄ αυτήν:

```
>> det(A)
ans =
       1/257554407665418070000000000000
>> format long e
```

```
>> det(A)
ans =
    3.882674767884667e-030
```

Αυτός είναι και ο λόγος που το MATLAB μας δίνει τις εξής λύσεις:

```
>> x=A\b
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 3.589083e-018.
x =
     -36/35
      10/7
      12/5
       4/5
>> A*x-b
ans =
       0
       0
       0
       0
>> x=inv(A)*b
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 3.589083e-018.
x =
      -4
      -2
       0
      -1
>> A*x-b
ans =
     -29
      -7
     -15
     -37
>> x=pinv(A)*b
x =
      52/69
       1/23
      73/69
     -10/69
>> A*x-b
ans =
      -1/112589990684262
      -1/450359962737050
```

```
-1/160842843834661
-1/93824992236885
```

Ο αλγόριθμος που χρησιμοποιείται στην λύση `x=A\b` *είναι η παραγοντοποίηση LU και φαίνεται να επαληθεύει το σύστημα. Πράγματι μπορούμε να βρούμε ότι για τις τιμές των* $z = \frac{4}{5}, w = \frac{4}{5}$ *η γενική λύση 13 δίνει την λύση που βρήκε το MATLAB. Επίσης η λύση που βρέθηκε με τον ψευδοαντίστροφο επαληθεύει το σύστημα και μπορούμε να δείξουμε ότι προκύπτει από την γενική λύση για τις τιμές των* $z = \frac{73}{69}, w = -\frac{10}{69}$. *Η μόνη 'λύση' που δεν επαληθεύει το σύστημα είναι αυτή με την χρήση του αντιστρόφου, αφού ο πίνακας δεν έχει αντίστροφο.*

Μπορείτε να βρείτε άλλα συστήματα, τα οποία ενώ γνωρίζουμε ότι ο πίνακάς τους δεν είναι αντιστρέψιμος, εντούτοις το MATLAB δίνει λύση με χρήση αντιστρόφου; Αν ναι, μην διστάσετε να επικοινωνήσετε με τον συγγραφέα[3] των σημειώσεων αυτών.

Παράδειγμα 1.3. *Να επιλυθεί το σύστημα:*

$$\left\{\begin{array}{rcl} x-y+z & = & 1 \\ 3x+y-z & = & 2 \\ 5x-y+z & = & 4 \end{array}\right\} \tag{14}$$

Λύση:

Δημιουργούμε τον επαυξημένο πίνακα Ab και κατόπιν χρησιμοποιούμε την συνάρτηση showechelon.m για να λάβουμε τα αποτελέσματα:

```
>> A=[1,-1,1;3,1,-1;5,-1,1],b=[1;2;4]
A =
       1            -1             1
       3             1            -1
       5            -1             1
b =
       1
       2
       4
>> Ab=[A,b]
Ab =
       1            -1             1             1
       3             1            -1             2
       5            -1             1             4
>> C=showechelon(Ab);
```

[3] Δημήτριος Θ. Χριστόπουλος, dchristop@econ.uoa.gr

```
Let the matrix be:
A =
       1             -1              1             1
       3              1             -1             2
       5             -1              1             4
 Do partial row pivoting: Swap rows 1 and 3 :
 Row(1) <--> Row(3)
A =
       5             -1              1             4
       3              1             -1             2
       1             -1              1             1
 Make pivot element A(1,1) equal to unity:
 Row(1) --> Row(1) / A(1,1)
A =
       1             -1/5            1/5           4/5
       3              1             -1             2
       1             -1              1             1
 Do  eliminations in column 1 :
 Row(2) --> Row(2) - A(2,1) * Row(1)
A =
       1             -1/5            1/5           4/5
       0              8/5           -8/5          -2/5
       1             -1              1             1
 Row(3) --> Row(3) - A(3,1) * Row(1)
A =
       1             -1/5            1/5           4/5
       0              8/5           -8/5          -2/5
       0             -4/5            4/5           1/5
 Make pivot element A(2,2) equal to unity:
 Row(2) --> Row(2) / A(2,2)
A =
       1             -1/5            1/5           4/5
       0              1             -1            -1/4
       0             -4/5            4/5           1/5
 Do  eliminations in column 2 :
 Row(1) --> Row(1) - A(1,2) * Row(2)
A =
       1              0              0             3/4
       0              1             -1            -1/4
       0             -4/5            4/5           1/5
 Row(3) --> Row(3) - A(3,2) * Row(2)
A =
       1              0              0             3/4
```

```
       0             1            -1           -1/4
       0             0             0            *
 The  column 3 is numerically negligible
A =
       1             0             0            3/4
       0             1            -1           -1/4
       0             0             0            *
 The  column 4 is numerically negligible
A =
       1             0             0            3/4
       0             1            -1           -1/4
       0             0             0            0
```

Τώρα είναι φανερό ότι το αρχικό σύστημα 11 είναι ισοδύναμο με το σαφώς απλούστερο σύστημα:

$$\left\{\begin{array}{rcl} x & = & \frac{3}{4} \\ y-z & = & -\frac{1}{4} \\ 0 & = & 0 \end{array}\right\} \Leftrightarrow \left\{\begin{array}{rcl} x & = & \frac{3}{4} \\ y & = & -\frac{1}{4}+z \\ z & = & z \end{array}\right\} \tag{15}$$

το οποίο μπορεί να γραφεί επίσης διανυσματικά ως εξής:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{3}{4} \\ -\frac{1}{4} \\ 0 \end{pmatrix} + z \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tag{16}$$

1.2 Μη Τετραγωνικά Συστήματα

Η γενική περίπτωση μη τετραγωνικών συστημάτων διαφέρει στο γεγονός ότι δεν μπορούμε να κάνουμε χρήση του αντιστρόφου πίνακα. Κάνουμε κι εδώ ακριβώς την ίδια διαδικασία της απαλοιφής Gauss-Jordan με μερική οδήγηση γραμμών, όπως φαίνεται στο παράδειγμα που ακολουθεί.

Παράδειγμα 1.4. *Να επιλυθεί το σύστημα:*

$$\left\{\begin{array}{rcl} x+y & = & 2 \\ 2x-y & = & 1 \\ 4x-5y & = & -1 \\ 7x-11y & = & -4 \end{array}\right\} \tag{17}$$

Λύση:

Δημιουργούμε τον επαυξημένο πίνακα Ab και κατόπιν χρησιμοποιούμε την συνάρτηση showechelon.m για να λάβουμε τα αποτελέσματα:

```
>> A=[1,1;2,-1;4,-5;7,-11],b=[2;1;-1;-4]
A =
       1              1
       2             -1
       4             -5
       7            -11
b =
       2
       1
      -1
      -4
>> Ab=[A,b]
Ab =
       1              1              2
       2             -1              1
       4             -5             -1
       7            -11             -4
>> C=showechelon(Ab);
Let the matrix be:
A =
       1              1              2
       2             -1              1
       4             -5             -1
       7            -11             -4
 Do partial row pivoting: Swap rows 1 and 4 :
 Row(1) <--> Row(4)
A =
       7            -11             -4
       2             -1              1
       4             -5             -1
       1              1              2
 Make pivot element A(1,1) equal to unity:
 Row(1) --> Row(1) / A(1,1)
A =
       1            -11/7           -4/7
       2             -1              1
       4             -5             -1
       1              1              2
 Do  eliminations in column 1 :
 Row(2) --> Row(2) - A(2,1) * Row(1)
```

```
A =
       1            -11/7           -4/7
       0             15/7           15/7
       4             -5             -1
       1              1              2
 Row(3) --> Row(3) - A(3,1) * Row(1)
A =
       1            -11/7           -4/7
       0             15/7           15/7
       0              9/7            9/7
       1              1              2
 Row(4) --> Row(4) - A(4,1) * Row(1)
A =
       1            -11/7           -4/7
       0             15/7           15/7
       0              9/7            9/7
       0             18/7           18/7
 Do partial row pivoting: Swap rows 2 and 4 :
 Row(2) <--> Row(4)
A =
       1            -11/7           -4/7
       0             18/7           18/7
       0              9/7            9/7
       0             15/7           15/7
 Make pivot element A(2,2) equal to unity:
 Row(2) --> Row(2) / A(2,2)
A =
       1            -11/7           -4/7
       0              1              1
       0              9/7            9/7
       0             15/7           15/7
 Do  eliminations in column 2 :
 Row(1) --> Row(1) - A(1,2) * Row(2)
A =
       1              0              1
       0              1              1
       0              9/7            9/7
       0             15/7           15/7
 Row(3) --> Row(3) - A(3,2) * Row(2)
A =
       1              0              1
       0              1              1
       0              0              0
```

```
       0              15/7            15/7
 Row(4) --> Row(4) - A(4,2) * Row(2)
A =
       1               0               1
       0               1               1
       0               0               0
       0               0               0
 The  column 3 is numerically negligible
A =
       1               0               1
       0               1               1
       0               0               0
       0               0               0
```

Τώρα φαίνεται άμεσα ότι το αρχικό σύστημα 17 είναι ισοδύναμο με το σαφώς απλούστερο σύστημα:

$$\left\{\begin{matrix} x & = & 1 \\ y & = & 1 \\ 0 & = & 0 \\ 0 & = & 0 \end{matrix}\right\} \Leftrightarrow \left\{\begin{matrix} x & = & 1 \\ y & = & 1 \end{matrix}\right\} \tag{18}$$

που είναι και η λύση αυτού. Σημειώνουμε ότι η λύση μπορούσε να βρεθεί και με την χρήση του ψευδοαντίστροφου:

```
>> x=pinv(A)*b
x =
       1
       1
>> A*x-b
ans =
      -1/1125899906842624
      -1/1801439850948199
       0
       0
>> format long e
>> A*x-b
ans =
   -8.881784197001252e-016
   -5.551115123125783e-016
                         0
                         0
>> x=pinv(A)*b
x =
```

```
    9.999999999999996e-001
    9.999999999999997e-001
```

απλά δεν είναι απόλυτα ακριβής λόγω των σφαλμάτων της αριθμητικής κινητής υποδιαστολής που χρησιμοποιεί το MATLAB.

1.3 Γραμμικά συστήματα μεγάλων διαστάσεων

Μπορούμε πάντοτε να κατασκευάσουμε ένα γενικό σύστημα $m \times n$ που να έχει γνωστή εκ των προτέρων μία λύση του. Για τον σκοπό αυτό λαμβάνουμε έναν κατάλληλο πίνακα A, $m \times n$, ένα οποιοδήποτε διάνυσμα στήλη x, $n \times 1$ και επιλέγουμε το διάνυσμα των δεξιών μερών των εξισώσεων να είναι το $b = A \cdot x$. Μετά λύνουμε το σύστημα $A \cdot x = b$ με το MATLAB ή οποιοδήποτε άλλο πρόγραμμα και κάνουμε τις παρατηρήσεις μας. Μας ενδιαφέρει να δούμε τι γίνεται για *πολύ μεγάλα συστήματα*, άνω των 1000 εξισώσεων τουλάχιστον. Ορίζουμε λοιπόν τυχαίους πίνακες A, ελέγχουμε εάν έχουν βαθμό ίσο με την μικρότερη διάστασή τους και κατόπιν ορίζουμε την λύση x της αρεσκείας μας, κατασκευάζουμε το διάνυσμα b και στο τέλος βρίσκουμε την λύση με το MATLAB:

```
>> format compact
>> format long e
>> A=randn(2500);
>> tic;rank(A)
ans =
        2500
>> toc
Elapsed time is 10.197189 seconds.
>> x=ones(2500,1);
>> b=A*x;
>> tic;xmat1=A\b;
>> toc
Elapsed time is 3.017956 seconds.
>> xmat1(1:10,1)
ans =
    9.999999999997149e-001
    9.999999999982921e-001
    1.000000000001966e+000
    1.000000000000744e+000
    1.000000000000900e+000
    9.999999999991314e-001
    9.999999999965370e-001
    1.000000000000292e+000
    9.999999999995043e-001
    1.000000000000193e+000
```

```
>> min(abs(x-xmat1)),max(abs(x-xmat1))
ans =
    4.440892098500626e-016
ans =
    5.520695012251053e-012
>> tic;xmat2=inv(A)*b;
>> toc
Elapsed time is 4.603429 seconds.
>> xmat2(1:10,1)
ans =
    9.999999999975577e-001
    9.999999999984452e-001
    1.000000000001581e+000
    1.000000000007093e+000
    1.000000000001212e+000
    9.999999999954019e-001
    9.999999999808535e-001
    1.000000000001183e+000
    9.999999999992693e-001
    9.999999999972261e-001
>> min(abs(x-xmat2)),max(abs(x-xmat2))
ans =
    1.776356839400251e-015
ans =
    4.750688731292030e-011
>> tic;xmat3=pinv(A)*b;
>> toc
Elapsed time is 16.714703 seconds.
>> xmat3(1:10,1)
ans =
    9.999999999998086e-001
    1.000000000000083e+000
    9.999999999996239e-001
    1.000000000000149e+000
    9.999999999995852e-001
    1.000000000000385e+000
    1.000000000000633e+000
    9.999999999996074e-001
    9.999999999999893e-001
    9.999999999999716e-001
>> min(abs(x-xmat3)),max(abs(x-xmat3))
ans =
     0
```

```
ans =
    1.609379296496627e-012
>> mean(x-xmat1),std(x-xmat1)
ans =
   -6.666671659161239e-014
ans =
    1.520535060612883e-012
>> mean(x-xmat2),std(x-xmat2)
ans =
   -1.362339130395185e-013
ans =
    6.302509159865352e-012
>> mean(x-xmat3),std(x-xmat3)
ans =
    1.315036968208005e-014
ans =
    2.906646963869463e-013
```

Βλέπουμε ότι *η πλέον ακριβής λύση δόθηκε με την χρήση του ψευδοαντίστροφου*, αλλά χρειάστηκε περίπου τετραπλάσιο χρόνο από τις άλλες δύο μεθόδους. Τώρα κάνουμε την ίδια εργασία για έναν τυχαίο πίνακα 1500×1000:

```
>> format compact
>> format long e
>> A=randn(1500,1000);
>> tic;rank(A)
ans =
        1000
>> toc
Elapsed time is 2.951011 seconds.
>> x=ones(1000,1);b=A*x;
>> tic;xm=pinv(A)*b;
>> toc
Elapsed time is 9.058941 seconds.
>> xm(1:10,:)
ans =
    1.000000000000003e+000
    9.999999999999867e-001
    1.000000000000004e+000
    9.999999999999990e-001
    1.000000000000000e+000
    9.999999999999980e-001
    9.999999999999993e-001
    1.000000000000004e+000
```

```
    9.999999999999909e-001
    1.000000000000001e+000
>> min(abs(x-xm)),max(abs(x-xm))
ans =
     0
ans =
    1.731947918415244e-014
>> mean(x-xm),std(x-xm)
ans =
    2.051692149507289e-016
ans =
    5.600241980340997e-015
>> tic;C=rref([A,b]);
>> toc
Elapsed time is 82.360330 seconds.
>> min(min(C)),max(max(C))
ans =
     0
ans =
    1.000000000000361e+000
>> min(C(1:1000,1001)), max(C(1:1000,1001))
ans =
    9.999999999996948e-001
ans =
    1.000000000000361e+000
>> min(min(C(1001:1500,:))), max(max(C(1001:1500,:)))
ans =
     0
ans =
     0
```

Το MATLAB βρίσκει την ακριβή λύση με σφάλμα στα όρια του *έψιλον της μηχανής*, όπως φαίνεται από τις τιμές ελαχίστου και μεγίστου σφάλματος.
Εάν θελήσουμε καλύτερη ακρίβεια θα χρησιμοποιήσουμε την εντολή rref([A,b]), αλλά θα χρειαστεί χρόνος πάνω από ένα λεπτό. Πράγματι τότε ο ανηγμένος κλιμακωτός πίνακας αποτελείται από δύο υποπίνακες:

- Έναν μοναδιαίο 1000×1000 πίνακα, έστω $I_{1000\times1000}$
- Ένα διάνυσμα στήλη 1000×1 με στοιχεία μονάδες, έστω $x_{1000\times1}$, για την ακρίβεια αριθμούς στο διάστημα

$$[9.999999999996948 \cdot 10^{-1}, 1.000000000000361 \cdot 10^{0}]$$

που είναι κοντά στις πραγματικές λύσεις

- Έναν μηδενικό πίνακα 500×1001, έστω $O_{500\times1001}$

Δηλαδή έχουμε τελικά, όπως μπορεί να δει κάποιος και με τον *editor* ότι:

$$C = \begin{pmatrix} I_{1000\times1000} & & x_{1000\times1} \\ & O_{500\times1001} & \end{pmatrix}$$

Εάν όμως δοκιμάσουμε να εργαστούμε με πίνακες που έχουν πολύ μεγάλο δείκτη κατάστασης ή πολύ μικρή ορίζουσα, τότε μόνον με την χρήση του ψευδοαντίστροφου επιτυγχάνουμε κάτι αξιόπιστο. Ένα τέτοιο παράδειγμα είναι ο πίνακας Hilbert που ορίζεται ως $h_{i,j} = \frac{1}{i+j-1}, i, j = 1, \ldots, n$. Εάν θεωρήσουμε τον πίνακα Hilbert 1000ης τάξης τότε έχουμε στο MATLAB:

```
>> A=hilb(1000);
>> tic;rank(A)
ans =
    24
>> toc
Elapsed time is 1.593930 seconds.
>> det(A)
ans =
     0
>> x=ones(1000,1);
>> b=A*x;
>> xmat1=A\b;
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 5.093585e-022.
>> xmat1(1:10,:)
ans =
    9.999941791639468e-001
    1.000823867771569e+000
    9.735646145715009e-001
    1.295309971016756e+000
    4.010338329795076e-001
   -1.222266852760478e+001
    1.246083578243988e+002
   -4.926062559835395e+002
    1.020377427038031e+003
   -9.553407337000234e+002
>> xmat1(end:-1:end-10,:)
ans =
    2.796606085889057e+003
```

```
   -1.779381861345185e+003
   -1.190212791621301e+003
   -1.930102388652020e+003
    1.915231986134224e+003
   -2.694017643689261e+003
    2.034054115380765e+003
    1.090666991604106e+003
   -6.994819998806054e+002
   -9.593164444739699e+001
    2.697303619546813e+002
>> min(abs(x-xmat1))
ans =
    5.820836053160861e-006
>> max(abs(x-xmat1))
ans =
    7.641232048806722e+003
>> mean(x-xmat1),std(x-xmat1)
ans =
   -1.435229632988921e-007
ans =
    1.758679412581018e+003
>> tic;xmat2=pinv(A)*b;
>> toc
Elapsed time is 3.194751 seconds.
>> xmat2(1:10,1)
ans =
    1.000000080923201e+000
    9.999928623437882e-001
    1.000152081251144e+000
    9.986419677734375e-001
    1.005836486816406e+000
    9.885635375976563e-001
    1.005134582519531e+000
    1.007173538208008e+000
    9.984831809997559e-001
    9.934272766113281e-001
>> xmat2(end:-1:end-10,:)
ans =
    9.978170394897461e-001
    9.976177215576172e-001
    9.979848861694336e-001
    9.982118606567383e-001
    9.978456497192383e-001
```

```
    9.979496002197266e-001
    9.981336593627930e-001
    9.984436035156250e-001
    9.983091354370117e-001
    9.984359741210938e-001
    9.985251426696777e-001
>> min(abs(x-xmat2))
ans =
    8.092320058494806e-008
>> max(abs(x-xmat2))
ans =
    1.143646240234375e-002
>> mean(x-xmat2),std(x-xmat2)
ans =
   -9.967702353606000e-007
ans =
    1.176583656753686e-003
```

Ο πίνακας έχει βαθμό ίσο με 1000 και όχι ίσο με 24. Επίσης η ορίζουσά του είναι μη μηδενική.
Εάν θελήσουμε να χρησιμοπιήσουμε την εντολή rref() η οποία έδειχνε να δουλεύει θαυμάσια για μικρότερα συστήματα το αποτέλεσμα είναι ένα παράδειγμα καταστροφής της λύσης λόγω σφαλμάτων αριθμητικής κινητής υποδιαστολής:

```
>> tic;B=rref(A);
>> toc
Elapsed time is 33.173741 seconds.
>> format rat
>> B(12,12:19)
ans =
  Columns 1 through 4
       1              0          -4978/201           0
  Columns 5 through 8
    9054/59      124587/154          0          -9057/10
```

Βλέπουμε ότι έχει δημιουργήσει μονάδες στην διαγώνιο μέχρι το στοιχείο (13,13), έχει μόνον 16 μη μηδενικές γραμμές αντί για 1000 και επίσης το κάθε διαγώνιο στοιχείο δεν έχει μηδενικά, αλλά μεγάλους αριθμούς. Κανονικά έπρεπε ο πίνακας αποτέλεσμα να είναι αυστηρά διαγώνιος με μονάδες στη διαγώνιο και τελευταίο στοιχείο κάθε γραμμής πάλι ίσο με ένα, ώστε να έχουμε βρει την πραγματική λύση. Καταλήγουμε λοιπόν στο ότι η πλέον αξιόπιστη μέθοδος για την επίλυση πολύ μεγάλων συστημάτων με το MATLAB είναι η μέθοδος του ψευδοαντίστροφου, δηλ. η χρήση της εντολής pinv(A)*b.

1.4 Σχετικά με περιορισμούς ακρίβειας στο MATLAB

Θα παρουσιάσουμε συνοπτικά αυτά που πρέπει να προσέχει κάποιος χρησιμοποιώντας το MATLAB ώστε να είναι σχετικά σίγουρος για την ακρίβεια των αποτελεσμάτων του.

1. Μην χρησιμοποιείτε κλασματικές δυνάμεις του 2.

Το MATLAB αδυνατεί να μετατρέψει αριθμούς που αναπαρίστανται ακριβώς στο πρότυπο IEEE 754 - 2008 / binary64 σε ακριβείς αριθμούς μηχανής.

Παράδειγμα 1.5. *Να εισαχθεί στο MATLAB ο πίνακας:*

$$A = \begin{pmatrix} \frac{257}{16} & \frac{4097}{256} & \frac{65537}{4096} & \frac{1048577}{65536} & \frac{16777217}{1048576} \\ \frac{513}{16} & \frac{8193}{256} & \frac{131073}{4096} & \frac{2097153}{65536} & \frac{33554433}{1048576} \\ \frac{769}{16} & \frac{12289}{256} & \frac{196609}{4096} & \frac{3145729}{65536} & \frac{50331649}{1048576} \\ \frac{1025}{16} & \frac{16385}{256} & \frac{262145}{4096} & \frac{4194305}{65536} & \frac{67108865}{1048576} \\ \frac{1281}{16} & \frac{20481}{256} & \frac{327681}{4096} & \frac{5242881}{65536} & \frac{83886081}{1048576} \end{pmatrix} \tag{19}$$

1. *Να γίνουν στοιχειώδεις πράξεις, όπως A^2, και να μελετηθεί η ακρίβειά τους*
2. *Να δημιουργηθεί το αντίστοιχο σύστημα με λύση τις μονάδες και να λυθεί. Πόσο είναι το σφάλμα;*

Λύση:

(α')Εισάγουμε στο MATLAB τον πίνακα και βλέπουμε ότι:

```
>> A=[257/16, 4097/256, 65537/4096, 1048577/65536, 16777217/1048576;
513/16, 8193/256, 131073/4096, 2097153/65536, 33554433/1048576;
769/16, 12289/256, 196609/4096, 3145729/65536, 50331649/1048576;
1025/16, 16385/256, 262145/4096, 4194305/65536, 67108865/1048576;
1281/16, 20481/256, 327681/4096, 5242881/65536, 83886081/1048576]
A =
     257/16        4097/256       65537/4096          16          16
     513/16        8193/256      131073/4096          32          32
     769/16       12289/256      196609/4096          48          48
    1025/16       16385/256      262145/4096          64          64
    1281/16       20481/256      327681/4096          80          80
```

Το σφάλμα που έχει κάνει ήδη το MATLAB κατά την εισαγωγή του πίνακα είναι:

$$A_{mat} - A = \begin{pmatrix} 0 & 0 & 0 & -\frac{1}{65536} & -\frac{1}{1048576} \\ 0 & 0 & 0 & -\frac{1}{65536} & -\frac{1}{1048576} \\ 0 & 0 & 0 & -\frac{1}{65536} & -\frac{1}{1048576} \\ 0 & 0 & 0 & -\frac{1}{65536} & -\frac{1}{1048576} \\ 0 & 0 & 0 & -\frac{1}{65536} & -\frac{1}{1048576} \end{pmatrix}$$

$$= \begin{pmatrix} 0.0 & 0.0 & 0.0 & -1.52587890625\,10^{-5} & -9.5367431640625\,10^{-7} \\ 0.0 & 0.0 & 0.0 & -1.52587890625\,10^{-5} & -9.5367431640625\,10^{-7} \\ 0.0 & 0.0 & 0.0 & -1.52587890625\,10^{-5} & -9.5367431640625\,10^{-7} \\ 0.0 & 0.0 & 0.0 & -1.52587890625\,10^{-5} & -9.5367431640625\,10^{-7} \\ 0.0 & 0.0 & 0.0 & -1.52587890625\,10^{-5} & -9.5367431640625\,10^{-7} \end{pmatrix}$$

Η τιμή του A^2 με την χρήση του MATLAB είναι:

```
>> A^2
ans =
   26923/7      76829/20      72982/19     138281/36     111393/29
   53838/7     161317/21     130580/17      53768/7       53768/7
   80753/7     149787/13      57606/5       80648/7       80648/7
  107668/7      76812/5      215057/14     107528/7      107528/7
  134583/7     192027/10      76805/4      134408/7      134408/7
```

Η πραγματική τιμή του A^2, με την χρήση του wxMaxima, είναι:

$$A^2 = \begin{pmatrix} \frac{64527554065}{16777216} & \frac{1031181525265}{268435456} & \frac{16497645064465}{4294967296} & \frac{263961061691665}{68719476736} & \frac{4223375727726865}{1099511627776} \\ \frac{129035949585}{16777216} & \frac{2062057562385}{268435456} & \frac{32990403367185}{4294967296} & \frac{527843936243985}{68719476736} & \frac{8445500462272785}{1099511627776} \\ \frac{193544345105}{16777216} & \frac{3092933599505}{268435456} & \frac{49483161669905}{4294967296} & \frac{791726810796305}{68719476736} & \frac{12667625196818705}{1099511627776} \\ \frac{258052740625}{16777216} & \frac{4123809636625}{268435456} & \frac{65975919972625}{4294967296} & \frac{1055609685348625}{68719476736} & \frac{16889749931364625}{1099511627776} \\ \frac{322561136145}{16777216} & \frac{5154685673745}{268435456} & \frac{82468678275345}{4294967296} & \frac{1319492559900945}{68719476736} & \frac{21111874665910545}{1099511627776} \end{pmatrix}$$

Το σφάλμα κατ΄ απόλυτο τιμή του MATLAB στον υπολογισμό του τετραγώνου του πίνακα, πάντα χρησιμοποιώντας το wxMaxima, κυμαίνεται από την ελάχιστη τιμή $\frac{4522001}{4294967296} = .1052860403 \cdot 10^{-2}$ έως την μέγιστη τιμή $\frac{491852049}{68719476736} = 0.7157389322 \cdot 10^{-2}$.

(β΄)Κατασκευάζουμε το διάνυσμα των μονάδων x, το διάνυσμα των δεξιών μελών $b = A \cdot x$, τον επαυξημένο πίνακα $A|b$ και βρίσκουμε τις λύσεις που μπορούμε να βρούμε με το MATLAB:

```
>> x=ones(5,1);b=A*x
```

```
b =
    1201/15
    2401/15
    3601/15
    4801/15
    6001/15
>> Ab=[A,b];
>> C=rref(Ab);
>> C(:,1:5)
ans =
       1            0          -1/16        -17/256      -273/4096
       0            1          17/16        273/256      4369/4096
       0            0            0            0            0
       0            0            0            0            0
       0            0            0            0            0
>> C(:,6)
ans =
    3295/4096
     944/225
       0
       0
       0
```

Επομένως σύμφωνα με το MATLAB το σύστημα $A \cdot x = b$ *είναι ισοδύναμο με:*

$$\left\{\begin{array}{rcl} x_1 - \frac{1}{16}\,x_3 - \frac{17}{256}\,x_4 - \frac{273}{4096}\,x_5 & = & \frac{3295}{4096} \\ x_2 + \frac{17}{16}\,x_3 + \frac{273}{256}\,x_4 + \frac{4369}{4096}\,x_5 & = & \frac{944}{225} \\ 0 & = & 0 \\ 0 & = & 0 \\ 0 & = & 0 \end{array}\right\}$$

Συνεπώς η λύση μας θα είναι:

$$\left\{\begin{array}{rcl} x_1 & = & \frac{3295}{4096} + \frac{1}{16}\,x_3 + \frac{17}{256}\,x_4 + \frac{273}{4096}\,x_5 \\ x_2 & = & \frac{944}{225} - \frac{17}{16}\,x_3 - \frac{273}{256}\,x_4 - \frac{4369}{4096}\,x_5 \\ x_3 & = & x_3 \\ x_4 & = & x_4 \\ x_5 & = & x_5 \end{array}\right\}$$

Είναι αυτή όμως η πραγματική γενική λύση;. Η απάντηση δυστυχώς είναι όχι. Εάν

εργαστούμε με το wxMaxima θα έχουμε την λύση:

$$\begin{pmatrix} 1 & 0 & -\frac{1}{16} & -\frac{17}{256} & -\frac{273}{4096} & \frac{3295}{4096} \\ 0 & 1 & \frac{17}{16} & \frac{273}{256} & \frac{4369}{4096} & \frac{17185}{4096} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Δηλαδή αναλυτικά:

$$\left\{ \begin{array}{rcc} x_1 & = & \frac{3295}{4096} + \frac{1}{16}\,x_3 + \frac{17}{256}\,x_4 + \frac{273}{4096}\,x_5 \\ x_2 & = & \frac{17185}{4096} - \frac{17}{16}\,x_3 - \frac{273}{256}\,x_4 - \frac{4369}{4096}\,x_5 \\ x_3 & = & x_3 \\ x_4 & = & x_4 \\ x_5 & = & x_5 \end{array} \right\}$$

Υπάρχει ένα απόλυτο σφάλμα της τάξης του $\frac{1}{921600} = -.1085069444 \cdot 10^{-5}$ *στην λύση του* x_2*, μάλιστα εστιάζεται στην μερική λύση του μη ομογενούς συστήματος, κάτι που δεν θα το θέλαμε για ένα τόσο απλό σύστημα* 5×5*.*

Σημείωση 1. *Σχετικά με το Octave.*
Πρέπει να σημειώσουμε εδώ ότι το Octave κατάφερε να δεχθεί τα στοιχεία του πίνακα A όπως τα εισαγάγαμε, χωρίς να τα αλλοιώσει:

```
octave-3.2.4.exe:1> format compact
octave-3.2.4.exe:2> format rat
octave-3.2.4.exe:3> A=[257/16, 4097/256, 65537/4096,
1048577/65536, 16777217/1048576;
513/16, 8193/256, 131073/4096, 2097153/65536, 33554433/1048576;
769/16, 12289/256, 196609/4096, 3145729/65536, 50331649/1048576;
1025/16, 16385/256, 262145/4096, 4194305/65536, 67108865/1048576;
1281/16, 20481/256, 327681/4096, 5242881/65536, 83886081/1048576]
A =

     257/16    4097/256           *           *           *
     513/16    8193/256           *           *           *
     769/16   12289/256           *           *           *
    1025/16   16385/256           *           *           *
    1281/16   20481/256           *           *           *

octave-3.2.4.exe:4> for j=1:5 A(3,j),A(4,j),A(5,j) end
ans = 769/16
```

```
ans = 1025/16
ans = 1281/16
ans = 12289/256
ans = 16385/256
ans = 20481/256
ans = 196609/4096
ans = 262145/4096
ans = 327681/4096
ans = 3145729/65536
ans = 4194305/65536
ans = 5242881/65536
ans = 50331649/1048576
ans = 67108865/1048576
ans = 83886081/1048576
octave-3.2.4.exe:5>
```

Δεν κατάφερε όμως να λύσει ακριβώς το αντίστοιχο σύστημα, παρά μόνον με την χρήση ψευδοαντίστροφου:

```
octave-3.2.4.exe:5> x=ones(5,1);b=A*x;Ab=[A,b]
Ab =

     257/16    4097/256           *           *           *    1201/15
     513/16    8193/256           *           *           *    2401/15
     769/16   12289/256           *           *           *    3601/15
    1025/16   16385/256           *           *           *    4801/15
    1281/16   20481/256           *           *           *    6001/15

octave-3.2.4.exe:6> C=rref(Ab)
C =

          1           0      -1/16    -17/256  -273/4096  3295/4096
          0           1      17/16    273/256  4369/4096    944/225
          0           0          0          0          0          0
          0           0          0          0          0          0
          0           0          0          0          0          0

octave-3.2.4.exe:7> xoct=pinv(A)*b
xoct =

          1
          1
          1
          1
```

```
          1

octave-3.2.4.exe:8>
```

Είχε δηλαδή τα ίδια αποτελέσματα στην γενική λύση του συστήματος με το MATLAB (Προσέξτε την ύπαρξη του λανθασμένου όρου $\frac{944}{225}$). Κατάφερε όμως να επιτύχει καλύτερη ακρίβεια στον υπολογισμό του A^2:

```
octave-3.2.4.exe:8> A^2
ans =

    26923/7   349572/91   341863/89   138281/36   111393/29
    53838/7   291907/38   130580/17   437825/57   222753/29
    80753/7   610670/53   645187/56     80648/7   334113/29
   107668/7   276523/18   568365/37    107528/7   445473/29
   134583/7   192027/10   326421/17    134408/7   556833/29
```

Εάν ελέγξουμε το σφάλμα βλέπουμε ότι βρίσκεται εντός του διαστήματος:

$$[.1018996465 \cdot 10^{-2}, .1422409785 \cdot 10^{-2}]$$

σαφώς καλύτερο από το αντίστοιχο διάστημα του MATLAB, που ήταν:

$$[.1052860403 \cdot 10^{-2}, 0.7157389322 \cdot 10^{-2}]$$

*2. Να χρησιμοποιείτε πάντα και την εντολή x=pinv(A)*b*

Σε όλα τα προηγούμενα παραδείγματα, ειδικά μάλιστα στα μεγάλων διαστάσεων συστήματα, είδαμε ότι η πλέον αξιόπιστη λύση που μπορεί να βρεθεί με το MATLAB είναι αυτή που χρησιμοποιεί τον ψευδοαντίστροφο του πίνακα A του συστήματος, δηλ. η εντολή λύσης x=pinv(A)*b. Αυτό συμβαίνει γιατί για αυτόν τον υπολογισμό το MATLAB κάνει ουσιαστικά *ανάλυση ιδιαζουσών τιμών*[4](Singular Value Decompsition - SVD) του πίνακα, η οποία είναι μία διαδικασία που δείχνει να επιτυγχάνει πολλές φορές καλύτερα αποτελέσματα. Δεν μπορούμε όμως να βασιστουμε σε αυτήν την ανάλυση για όλα. Για παράδειγμα είδαμε ότι για τον πίνακα Hilbert διαστάσεων 1000×1000 το MATLAB έδωσε βαθμό του πίνακα ίσο με 24 και όχι 1000. Σημειώνουμε ότι το MATLAB βρίσκει τον βαθμό ενός πίνακα κάνοντας την ανωτέρω ανάλυση.
Επίσης ακόμα και αυτή η μέθοδος για ορισμένους τύπους πινάκων αποτυγχάνει πλήρως, σαν παράδειγμα μπορείτε να λύσετε την Άσκηση 2 για $\nu = 140$.

[4]Για τον $m \times n$ πίνακα A η ανάλυση ιδιαζουσών τιμών είναι $A = U \cdot \Sigma \cdot V^T$, όπου U είναι τα αριστερά και V είναι τα δεξιά ιδιοδιανύσματα του A, ενώ Σ είναι ο πίνακας με διαγώνια στοιχεία τις ιδιοτιμές του $A^T A$ ή $A^T A$.

Γενικώς ο κανόνας είναι να χρησιμοποιούμε το MATLAB για πολύ μεγάλους πίνακες με στοιχεία αρκετά μικρότερα από το $eps = 2.220446049250313 \cdot 10^{-16}$, εάν είμαστε σίγουροι ότι το πρόβλημά μας είναι *καλά τοποθετημένο* από την σκοπιά της αριθμητικής ανάλυσης.

Σε κάθε περίπτωση που δεν ισχύει κάτι από τα παραπάνω ή που δεν γνωρίζουμε σε τι εύρος θα κυμανθεί η λύση ενός προβλήματος, τότε θα πρέπει να χρησιμοποιούμε τα Συστήματα Υπολογιστικής Άλγεβρας (Computer Algebra Systems - CAS) για να είμαστε απόλυτα σίγουροι για τα αποτελέσματά μας.

1.5 Ασκήσεις

1. Να βρεθεί η γενική λύση του συστήματος:

$$\left\{\begin{array}{rcr} 3\,x + 9\,y - 3\,z & = & 12 \\ -7\,x - 23\,y + 8\,z & = & -29 \\ 2\,x - 4\,y + 3\,z & = & 3 \end{array}\right\}$$

χρησιμοποιώντας απαλοιφή Gauss-Jordan με μερική οδήγηση γραμμών στο MATLAB. Κατόπιν να προσπαθήσετε να το λύσετε με όλες τις διαθέσιμες εντολές του MATLAB. Τι παρατηρείτε ;

2. Να κατασκευαστεί και να επιλυθεί το σύστημα $A \cdot x = b$ με:

$$A = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 2 & 2^2 & 2^3 & \cdots & 2^\nu \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu & \nu^2 & \nu^3 & \cdots & \nu^\nu \end{pmatrix}, b = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ \nu \end{pmatrix}$$

για $\nu = 20, 40, 80, 100, 140$ με την χρήση όλων των διαθεσίμων μεθόδων του MATLAB και να γίνει επαλήθευση. Ποια μέθοδος έχει το μικρότερο σφάλμα; Ποια μέθοδος απέτυχε να λύσει το σύστημα με ακρίβεια αποδεκτή σε αριθμητική κινητής υποδιαστολής;

3. Να βρεθεί η γενική λύση του ομογενούς συστήματος:

$$\left\{\begin{array}{rcr} 5\,x_1 + 9\,x_2 - 3\,x_3 - 4\,x_4 - 13\,x_5 & = & 0 \\ 5\,x_1 + 5\,x_2 - x_3 + 4\,x_5 & = & 0 \\ -4\,x_1 - 2\,x_3 - 4\,x_4 - 12\,x_5 & = & 0 \\ 5\,x_1 + 10\,x_2 - 3\,x_3 - 5\,x_4 - 9\,x_5 & = & 0 \\ 3\,x_1 + 3\,x_2 + 5\,x_3 - 5\,x_5 & = & 0 \end{array}\right\}$$

4. Να βρεθεί η γενική λύση του συστήματος:

$$\left\{\begin{array}{rcr} 2\,x_1 - 3\,x_2 + 8\,x_3 + 6\,x_4 + 4\,x_5 + x_6 & = & 4 \\ 17\,x_1 - 21\,x_2 + 23\,x_3 - 15\,x_4 - 11\,x_5 + 22\,x_6 & = & 4 \\ 6\,x_1 + 7\,x_2 - 6\,x_3 - 10\,x_4 - 12\,x_5 + x_6 & = & 2 \\ -2\,x_1 - 5\,x_2 + 7\,x_3 + 8\,x_4 + 8\,x_5 & = & 1 \end{array}\right\}$$