4. SOLVING A SYSTEM OF LINEAR EQUATIONS

Objectives of the paper:

- Recapitulation of elements regarding the solving of systems of linear equations,
- Fixing knowledge regarding the solving of systems of linear equations using the Matlab programming environment,
- Familiarization with the symbolic calculation in Matlab and with solving the linear equation systems using the symbolic method, by studying some examples and solving some problems.

It is recommended to go through Annex M4 before studying paragraphs 4.1 and 4.2.

4.1. Elements regarding the solving of systems of linear equations in Matlab

The numerical solving of systems of linear equations

In Matlab only determined compatible systems can be solved numerically and maximum two particular solutions can be found for undetermined compatible systems. When solving linear systems of p equations and n unknowns using the Matlab environment, it is assumed that these have the following matrix form:

`A·X = b`

with `A` the matrix of the coefficients, having `p` lines and `n` columns, `b` the column vector of the constant terms and `X` the column vector of the unknowns.
Alternatively, the following matrix form can also be used:

`X1·A1 = b1`

where $A1=A^T$, $b1=b^T$ and $X1=X^T$. This form uses the vectors for the constant terms and of the unknowns as line-vectors. A system of linear equations is defined in Matlab by defining the matrix `A` and the vector `b` (or matrix `A1` and vector `b1`).

a. Solving determined compatible systems

Determined compatible systems can be solved by using two methods:

1. *the inverse matrix method*:

In the case of a square system, the system solution is obtained by inverting the matrix of the coefficients and multiplying it with the vector of the constant terms:

```
X = inv(A) *b
```

respectively,

```
X1 = b1 *inv(A1)
```

2. ***the left/ right division method***:

This method uses one of the division left or right operators, depending on the matrix form used:

```
X = A \b
```

respectively,

```
X1 = b1 /A1
```

The two solving methods are based on different numerical methods. The left /right division method uses the Gauss version of the elimination method in order to solve systems of linear equations. This method does not require determining the inverse of the coefficients matrix.

b. Determining a particular solution for undetermined compatible systems

In the case of undetermined compatible systems, all solutions cannot be determined numerically, these being in infinite number. One particular solution can be determined using one of the following two methods:

1. ***the left/ right division method***:

```
X = A \b
```

respectively,

```
X1 = b1 /A1
```

In this case, this method searches for that solution of the system that minimizes in the sense of the smallest squares the Euclidean norm of the vector `A·X-b`and which has at most `rank A` non-zero components.

2. ***the pseudoinverse matrix method***:

The system solution is obtained by multiplying the Moore-Penrose pseudoinverse of the matrix of coefficients with the vector of the constant terms. The Moore-Penrose pseudoinverse is obtained by employing the Matlab function *`pinv`*:

```
X = pinv(A) *b
```

respectively,

```
X1 = b1 *pinv(A1)
```

Solving systems of linear equations the symbolic way

In Matlab following types of systems can be solved symbolically: undetermined compatible square systems, underdetermined systems and systems with parameters.

a. Symbolic computation in Matlab

For symbolic computation, Matlab offers the user the ***Symbolic Math*** toolbox. In Table 4.1. a few of the functions used in Matlab for symbolic computation are presented:

Table 4.1. Matlab functions for symbolic computation.

Function	**Use**
det	calculates the determinant of a symbolic matrix
factor	decomposes into factors a symbolic expression
inv	calculates the inverse of a symbolic matrix
rank	calculates the maximum rank of a symbolic matrix
simplify, simple	simplifies symbolic expressions
solve	solves equations and equation systems
subs	substitutes a symbol with another symbol or numeric value
sym	creates a symbolic object
syms	creates several symbolic objects

The complete list of Matlab functions for symbolic computation can be displayed by using the help associated to the Matlab *symbolic* folder. Solving a problem symbolically must begin with the definition of the symbolic objects (symbols).Then the actual implementation of the problem solving can begin.

b. Solving systems of linear equations using the Symbolic Math Toolbox

The solving methods specified at the beginning of the paper are also valid for the symbolic solving of systems of linear equations:

1. *the inverse method/ the pseudoinverse method*;
2. *the left/ right division method*.

Symbolic solving typically requires the study of the system and case based solving.

4.2. Examples

All examples correspond to the canonical form $A \cdot X = b$ (see paragraph 4.1).Before solving a system of linear equations it must be verified whether the system is compatible (see annex M4).

Example 4.1: Solve the system below using the inverse matrix method:

$$\begin{cases} 5x_1 + 4x_2 + x_3 = 0 \\ 6x_1 + 3x_2 + 2x_3 = 5 \\ x_1 + x_2 + x_3 = -7 \end{cases}$$

Solution: The following Matlab program sequence is executed (for example, M-file):

```
A=[5 4 1; 6 3 2; 1 1 1]; % coefficients matrix
% solving the system
if det(A)~=0 % if the system is compatible and determined
    b=[0; 5; -7]; % column-vector for the constant terms
    X=inv(A)*b
else
    disp('The system is not compatible and determined.')
end
```

After the sequence above is executed:

```
X =
   6.2500
  -6.0000
  -7.2500
```

Namely: x_1 = 6.25 ,x_2 = -6 , x_3 = –7.25.

Example 4.2: Solve the following system using the left division method:

$$\begin{cases} 2x_1 - 3x_2 = 7 \\ -6x_1 + 8x_2 - x_3 = -5 \\ 3x_2 + 4x_3 = 1 \end{cases}$$

Solution.: The following Matlab program sequence is executed (for example, M-file):

```
A=[2 -3 0; -6 8 -1; 0 3 4]; % coefficients matrix
% solving the system
if det(A)~=0
    b=[7 -5 1]'; % column-vector for the constant terms
    X=A\b
else
    disp('The system is not compatible and determined.')
end
```

After the sequence above is executed:

```
X=    -94.0000
      -65.0000
       49.0000
```

Namely: x_1 = -94, x_2 = -65 ,x_3 = 49.

Example 4.3: Determine for the following systems one or two particular solutions, using the pseudoinverse and left division methods.

a) $$\begin{cases} 2x_1 - 3x_2 = 7 \\ -6x_1 + 8x_2 - x_3 = -5 \\ 3x_2 + 4x_3 = 1 \\ -4x_1 + 8x_2 + 3x_3 = 3 \end{cases}$$

b) $$\begin{cases} 3x - y + z - 2t = 6 \\ -4x + 4y + 2z + t = 0 \end{cases}$$

Solution:

a) The following Matlab program sequence is executed (for example, M-file):

```
A=[2 -3 0; -6 8 -1; 0 3 4; -4 8 3]; % coefficients matrix
b=[7; -5; 1; 3]; % column-vector for the constant terms
% solving the system
if rank(A)==rank([A b]) % if the system is compatible
    disp('the pseudoinverse method')
    X=pinv(A)*b
    disp('the left division method')
    X=A\b
else
    disp('The system is not compatible.')
end
```

After the sequence above is executed:

```
the pseudoinverse
method X=-94.0000
      -65.0000
       49.0000
the left division
method   X=    -94.0000
    -65.0000
     49.0000
```

b) The following Matlab program sequence is executed (for example, M-file):

```
A=[3 -1 1 -2; -4 4 2 1]; % coefficients matrix
b=[6; 0]; % column-vector for the constant terms
% solving the system
if rank(A)==rank([A b])
    disp('the pseudoinverse method')
    X=pinv(A)*b
    disp('the left division method')
    X=A\b
else
    disp('The system is not compatible.')
end
```

After the sequence above is executed:

```
the pseudoinverse
method X=0.9431
         0.5418
         1.3846
        -1.1639
the left division
method   X=    1.2000
         0
         2.4000
         0
```

Comments: 1. In order to test the system compatibility, the Kronecker-Capelli theorem was used. The extended matrix was obtained in Matlab by concatenating matrix A with the column vector b, `[A b]`.

2. In the case of the first system, which is compatible and determined, obviously identical solutions were obtained by using the two methods. In the case of the second system, which is undetermined and compatible, using the two methods, two distinct solutions were obtained. The complete solution of this system appears in the example 4.4.

Example 4.4: Solve the system of linear equations:

$$\begin{cases} 3x - y + z - 2t = 6 \\ -4x + 4y + 2z + t = 0 \end{cases}$$

Solution: The considered system of linear equations is an underdetermined system. It can be solved only by symbolic means. The following Matlab program sequence is executed (for example, M-file):

```
clc
% step 1: the rank of the system's matrix is
% determined
A=[3 -1 1 -2;-4 4 2 1];
r=rank(A) % 2is obtained
% therefore, 2 variables are independent, and 2
% variables dependent on the first ones

% step 2: a non-zero second order minor is searched,
% to determine the dependent variables;
% for example:
rminor=rank(A(:,[1 2])) % 2is obtained
% x,y become the dependent variables, in relation to which
% thesystem is solved;
% it is rewritten under the form:
%3x-y=6-z+2t;   -4x+4y=-2z-t;

disp('the system is undetermined and compatible')
disp(blanks(1)')

% step 3: the system, rewritten in the new form, is resolved:
% the symbolic objects are created
syms z t;
% the system's matrix
Areduced=A(:,[1 2]);
% the constant terms vector
breduced=[6-z+2*t; -2*z-t];
% solving the system with the inverse method
s=inv(Areduced)*breduced
% solving the system with the left division operator
ss=Areduced\breduced
disp(blanks(1)')
pause
disp('The system solution is:')
x=s(1)
y=s(2)
disp('z,t any real numbers')
```

Following results are obtained:

```
r=      2

rminor=      2
the system is undetermined and compatible

s =
 3-3/4*z+7/8*t
 3-
5/4*z+5/8*
t ss =
 3-3/4*z+7/8*t
 3-5/4*z+5/8*t
```

```
The system solution is:
 x =
  3-
  3/4*z+7/8
  *t y =
  3-5/4*z+5/8*t
  z,tany real numbers
```

Therefore, the obtained solution is:

$$x = 3 - \frac{3}{4}z + \frac{7}{8}t, \quad y = 3 - \frac{5}{4}z + \frac{5}{8}t, \quad z \in R, \quad t \in R$$

Comments: 1. The Matlab function *blanks* creates spaces between strings.

2. The Matlab command *pause* has the effect of temporarily suspending the execution of the program. Execution proceeds only after pressing a key.

Example 4.5: Solve the system of linear equations:

$$\begin{cases} mx - nz = q \\ nx - my = 0 \\ my + mz - 2q = 0 \end{cases}$$

, with the unknowns *x*, *y*, *z*,

with the parameters *m*, *n* and *q*, of the first two parameters, at least one being non-zero.

Solution: Given that the system is with parameters, it can only be solved using the symbolic calculation toolbox. A system compatibility study is required based on various parameter values, the system solving being done based on cases, the Matlab environment being used only to perform symbolic or numerical calculations and substitutions of parameters with particular values.

The following Matlab program sequence is executed (for example, M-file):

```
% the symbolic objects are created
m=sym('m'); n=sym('n'); q=sym('q');
% system matrix
A=[m 0 -n;n -m 0;0 m m];
% column-vector for the constant terms
b=[q; 0; 2*q];

% ** Discussion **
% determinant of the system
d=det(A)    % is obtained d = -m^3-n^2*m
factor(d)   % is obtained-m*(m^2+n^2)
% it can be noticed that d==0 if and only if m==0
% case d~=0: the system is compatible and determined
disp('Case: m~=0 -> the system is compatible and determined')
% solution calculated with the inverse matrix method
s=inv(A)*b
% solution calculated with the left division
operatorss=A\b
disp(blanks(2)')
pause
```

```
% case d==0
% substitution of m with the value 0
A=subs(A,m,0)
% it is noticeable that the last line of A contains only null
% elements, but, the last element of b is 2*q
% therefore the system is incompatible if q~=0
% and compatible andundetermined, if q==0
disp(['Case: m==0 and q~=0 -> the system'...'
'is incompatible'])
disp(blanks(2)')
pause
% case ofa compatible and undetermined system
disp(['Case: m==0,n~=0,q==0 -> system is compatible '...'
'and undetermined'])
% substitution of q with the value 0
b=subs(b,q,0);
% system becomes:
% -n*z==0; n*x==0;
% m being 0, resultsfrom thehypothesis than n is non-zero
% disp('Solution: (0,y,0) with y any real number')
```

The obtained results are:

```
d =-m^3-n^2*m
ans =-m*(m^2+n^2)
Case: m~=0 ->the system is compatible and
determined s =
     m/(m^2+n^2)*q+2*n/(m^2+n^2)*q
 n/(m^2+n^2)*q+2*n^2/m/(m^2+n^2)*q
    -n/(m^2+n^2)*q+2*m/(m^2+n^2)*q
ss =
     q*(m+2*n)/(m^2+n^2)
 q*n*(m+2*n)/m/(m^2+n^2)
    -q*(n-2*m)/(m^2+n^2)

A =
[  0,  0,-n]
[  n,  0,  0]
[  0,  0,  0]
Case: m==0 and q~=0 ->the system is incompatible

Case: m==0,n~=0,q==0 ->system is compatible and undetermined
Solution: (0,y,0) with y any real number
```

Example 4.6: Compare the studied methods of solving the systems of linear equations in terms of the execution time and the precision of the solution for the case of a compatible and determined square system of large linear equations.

Solution: In order to generate a large matrix, the Matlab`rand` function was used, which generates a matrix of uniformly distributed random numbers.The following Matlab program sequence is executed (for example, a script file):

```
% system matrix, exact solution, constant terms vector:
A=rand(700); x=rand(700,1); b=A*x;
%  execution times:
tic;
y=inv(A)*b;time1=toc
tic; z=A\b;time2=toc
%  precision of the calculated solutions:
n1=norm(A*y-b)
n2=norm(A*z-b)
```

After the sequence above is executed:

```
time1 =
    0.1671
time2 =
    0.0685
n1 =
  3.3005e-010
n2 =
  5.9342e-012
ans =
    55.
    618
    2
```

Since in the first case - the use of the inverse matrix method - the execution time is 0.1671 seconds and the error is $3.3005\cdot 10^{-10}$, and in the second case – the use of the left division method – the execution time is 0.0685 secondsand the error is $5.9342\cdot 10^{-12}$, it can be observed that the left division method of the matrices is more efficient than the inverse matrix method, both in terms of execution time and accuracy of the obtained solution.

4.3. Problems to solve

P4.1. Using the Matlab environment, analyze whether the following system of linear equations is compatible and determined and, if so, solve the system:

$$\begin{cases} 4x+3y-z=2 \\ -x+y+z=0 \\ x+3z=-1 \end{cases}$$

P4.2. Using the Matlab environment, determine the rank of the coefficients matrix and an approximate solution for each of the following systems of linear equations:

a) $$\begin{cases} 4x+3y-z=2 \\ -x+y+z=0 \\ x+3z=-1 \\ 4x+4y+3z=1 \end{cases}$$

b) $$\begin{cases} -6x+8y-z=-5 \\ 2x-3y=7 \end{cases}$$

P4.3. Write a program that receives as a parameter the matrix of the coefficients of an arbitrary system of linear equations and the vector of the constant terms and returns the system solution, if it is compatible and determined, or a corresponding message, if it is compatible and undetermined or incompatible.

P4.4. Using the Matlab environment, solve the system of linear equations:

$$\begin{cases} -6x+8y-z=-5 \\ 2x-3y=7 \end{cases}$$

P4.5. Using the Matlab environment, solve the following system of linear parameter equations with the unknowns x, y, z:

$$\text{a)} \begin{cases} \alpha x-\beta y+z=\gamma \\ -\gamma x+y+\beta z=\alpha \\ x+\gamma y-\alpha z=\beta \end{cases}$$

$$\text{b)} \begin{cases} ax-by=p \\ -bx+by-cz=-2q \\ cy+az=p+q \end{cases}$$

P4.6. Write a program that receives as arguments the coefficients matrix and the constant terms vector of a of an arbitrary system of linear equations and classifies the system into one of the categories: i) compatible and determined, ii) compatible and undetermined or iii) incompatible, and displays a corresponding message.

P4.7. Realize a case study similar to example 4.6. for the case of a compatible and undetermined system for which it is desired to determine a particular solution.

ANNEX M4. ELEMENTS REGARDING THE SOLVINGOF SYSTEMS OF LINEAR EQUATIONS

M4.1. General concepts

Having the system with p linear equations and n unknowns:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{p,1}x_1 + a_{p,2}x_2 + \cdots + a_{p,n}x_n = b_p \end{cases}$$

with $a_{i,j} \in R, \quad b_i \in R, \quad i = \overline{1,p}, j = \overline{1,n}$

Introducing the notations:

$$A=\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{p,1} & a_{p,2} & \cdots & a_{p,n} \end{pmatrix}, \quad b=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix}, \quad X=\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

the matrix form of the system on linear equations is obtained:

$A \cdot X = b$

Matrix A is called ***the coefficients matrix***, matrix b ***the column vector of the constant terms*** and X ***the column vector of the unknowns of the system of linear equations***.

Matrix (A,b) with p lines and $n+1$ columns obtained by attaching to the right of the matrix A the vector b, is called ***the extended matrix of the system***.

Depending on the result of the comparison between p and n the following types of systems can be distinguished:

- ***square system***, if $p = n$;
- ***over determined system***, if $p > n$;
- ***underdetermined system***, if $p < n$.

A system of linear equations that has at least one solution is called ***a compatible system***. A compatible system that only has one solution is called ***a compatible and determined system***. A compatible system that has more than one solution is called ***a compatible undetermined system***. A system that has no solution is called ***an incompatible system***. The solution of a compatible and determined square system of linear equations can be obtained by using the relationship:

$X = A^{-1} \cdot b$

A compatible undetermined system has an infinity of solutions.

M4.2. Compatibility criteria

Kronecker – Capelli theorem: A system of linear equations is compatible if and only if the rank of the system matrix is equal with the rank of the extended system matrix.

A compatible system of linear equations is determined if and only if the rank of the system matrix is equal with the number of unknowns of the system.

A square system of linear equations is compatible and determined if and only if the determinant of the coefficients matrix is non-zero.

Checking the compatibility of a system of linear equations can be done in the following way:

a) in the case of square systems:
- if the determinant of the coefficients matrix is non-zero, the system of is compatible and determined;
- otherwise:
  - if the rank of the system matrix is equal with the rank of the extended system matrix, then the system is compatible and undetermined;
  - otherwise, the system is incompatible.

b) in the case of any systems:
- if the rank of the system matrix is equal with the rank of the extended system matrix, then the system is compatible; we distinguish two sub-cases:
  - if the rank of the system matrix is equal with the number of unknowns, then the system is compatible and determined;
  - otherwise, the system is compatible and undetermined;
- otherwise, the system is incompatible.

M4.3. Homogeneous systems of linear equations

A particular case of systems of linear equations are the homogeneous systems of linear equations, i.e. systems for which b=0.

A homogeneous system is always compatible, because it always admits the null solution: X =0.

If the rank of the homogeneous system matrix is equal with the number of unknowns, then the system is compatible and determined. Otherwise (the rank is smaller than the number of unknowns) the system is compatible and undetermined.

M4.4. The Moore-Penrose pseudoinverse of a matrix

The Moore-Penrose pseudoinverse of matrix A is a matrix B of the same dimension as A^T, that fulfills the following conditions:

(a) $A \cdot B \cdot A = A$ and $B \cdot A \cdot B = B$,

(b) $A \cdot B$ and $B \cdot A$ are Hermitian matrices.

(***A Hermitian matrix*** is a square matrix with the property that it is equal to its own conjugate transpose.)

The properties of the Moore-Penrose pseudoinverse are:

1. Any matrix has a single Moore-Penrose pseudoinverse.
2. The pseudoinverse of an invertible matrix coincides with the inverse matrix.
3. The pseudoinverse of the pseudoinverse of a matrix is the matrix itself.

Having the compatible system of linear equations with n unknowns:

$$A \cdot X = b$$

Let A^+ be the Moore-Penrose pseudoinverse of matrix A. Then:

$$Xa = A^+ \cdot b$$

is an approximate solution of the system of linear equations with the property that it has the minimal Euclidean norm of all the n-tuples X for which $\|A \cdot X - b\|^2$ is minimal, where with $\|\cdot\|$ the Euclidean norm was noted.