Linear system of equations

During applications that involve usage of differential equations, such as in modelling environmental changes, diseases, interactions, etc. often do land into a system of equations of the form $AX = B$ where X is a vector of decision variables for the given system of differential equations, while A and B are the coefficient matrix and system solution vector, respectively. For example, a system of the form

$$\begin{array}{rrrcl} 2x_1 + & x_2 - & x_3 & = & 0 \\ x_1 + & 7x_2 + & 2x_3 & = & 1 \\ 3x_1 - & 3x_2 + & 4x_3 & = & 2 \end{array}$$

can be re-arranged in form of $\boldsymbol{AX} = B$ where

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 7 & 2 \\ 3 & -3 & 4 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

In this unit, we shall then describe means of solving such given systems of linear equations in MATLAB based on the relation

$$\boldsymbol{X} = A^{-1}B$$

given that A^{-1} exists.

Since we have already looked at how to compute matrix multiplication and also inverse of a matrix, then easier it will be to compute

$$\boldsymbol{X} = A^{-1}B$$

For example, for the equations

$$\begin{array}{rrcl} x + & 2y & = & 5 \\ 3x + & 4y & = & 6 \end{array}$$

may be written in the form of $\boldsymbol{AX} = B$ as

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

where

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix}, B = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

Hence, the solution in MATLAB may be written as

```
>> A=[1 2;3 4]
A =
    1   2
    3   4
>> B=[5;6]
B =
    5
    6
>> X=inv(A)*B
X =
   -4.0000
    4.5000
```

Take note that the inverse syntax ***"X=inv(A)*B"*** can as well be written in another form as ***"X=A\B"*** where the latter syntax evaluates even if the matrix A does not have an inverse. But in actual sense, the two syntaxes should be able to give the same result. For example

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

>> X=A\B
X =
   -4.0000
    4.5000
```

For very large coefficient and ill-conditioned matrix A, this method of solving the system $AX = B$ by the operation

$$X = A^{-1}B$$

is not efficient. Thus, other methods exist which we shall discuss them as either iterative or non-iterative methods, derived based on programming techniques. However, since we have not gone into details yet of writing MATLAB programming files, in this unit, we only discuss these methods using MATLAB syntaxes.

a) LU Factorization method

This method is built on factorization of lower L and upper U triangular matrices of coefficient matrix A. Thus, LU factorization of a matrix $A \in \mathbb{R}^{n\times n}$ is given by

$$A = LU$$

where L and U are the lower and upper triangular matrices, respectively. The MATLAB syntax for LU factorization of matrix $A \in \mathbb{R}^{n\times n}$ is therefore

$$LU = lu(A)$$

For example

```
>> A=[1 2;3 4]
A =
    1    2
    3    4
>> [L U]=lu(A)
L =
   0.3333    1.0000
   1.0000         0
U =
   3.0000    4.0000

        0    0.6667
```

Or sometimes, LU factorization of matrix $A \in \mathbb{R}^{n\times n}$ is given by

$$A = LU = LDU$$

where D is the diagonal matrix. Hence, the MATLAB syntax for this is

$$[L, P, U] = lu(A)$$

For example

```
>> A=[1 2;3 4]
A =
   1    2
   3    4
>> [L,U,P]=lu(A)
L =
  1.0000        0
  0.3333   1.0000
U =
  3.0000   4.0000
       0   0.6667
P =
   0    1
   1    0
```

Hence, in order to solve the system $AX = B$ by LU factorization method, we have $AX = LUX = B$ such that

$$X = (LU)^{-1}B = L^{-1}U^{-1}B$$

Or alternatively, we evaluate $LUX = B$ in two phases

- $UX = Z$ and
- $LZ = B$

such that the unknowns of vector $\boldsymbol{Z}$ are determined by Forward substation method, while the unknowns of vector $\boldsymbol{X}$ are determined by backward substitution.

For example, using

$$\boldsymbol{Z} = L^{-1}B, \quad \boldsymbol{X} = U^{-1}Z$$

we have

```
>> A=[1 2;3 4]
A =
    1    2
    3    4
>> [L U]=lu(A)
L =
   0.3333    1.0000
   1.0000         0
U =
   3.0000    4.0000
        0    0.6667
>> B=[5;6]
B =
    5
    6
>> Z=inv(L)*B
Z =
    6
    3
>> X=inv(U)*Z
X =
  -4.0000
   4.5000
```

b) Singular Value Decomposition technique

The singular value decomposition (SVD) of a given matrix $A \in \mathbb{R}^{n \times n}$ is given by

$$A = USV^T$$

where U is an orthogonal matrix, V is an orthogonal matrix, and S is a diagonal singular matrix such that the MATLAB syntax is defined by

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

For example,

```
>> A=[1 2;3 4]
A =
   1   2
   3   4
>> [U,S,V]=svd(A)
U =
  -0.4046  -0.9145
  -0.9145   0.4046
S =
   5.4650        0
        0   0.3660
V =
  -0.5760   0.8174
  -0.8174  -0.5760
```

Activity 2 c

a) Consider the following system of linear equations

$$\begin{cases} x + 2y + 3z = 1 \\ 4x + 5y + 6z = 1 \\ 7x + 8y \qquad = 1 \end{cases}$$

   a. Derive an expression of the form $A\boldsymbol{x} = \boldsymbol{b}$ in MATLAB

   b. Hence, solve for $\boldsymbol{x}$ using inverse and $A\backslash b$ methods and compare.

b) By the system of linear equations presented in a) above, compute

   a. Inverse of its coefficient matrix

   b. Diagonal of the coefficient matrix

c) Consider the following system of linear equations

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

   a. Derive an expression of the form $A\boldsymbol{x} = \boldsymbol{b}$ in MATLAB

   b. Hence, solve for $\boldsymbol{x}$ using LU factorization method and compare with $A\backslash b$ method.

d) From each matrix A of problems a) and c), deduce U, S, V via $[U, S, V] = svd(A)$.