Experiment1: Linear equations

1 Linear equations

A **linear equation** is an equation in which each term is either a constant or the product of a constant and a **first order power variable**. The simplest example of a linear equation is with *one variable*, x, and can be written as $ax + b = 0$, where a and b are constants. The constants might be numbers or even non-linear functions of parameters, but for now let us stick to numbers. It is also possible to consider a series of linear equations. In general, we can define n equations with m variables ($n \neq m$). Let the variables be denoted $x_1, x_2, x_3.....x_m$ then these equations can be written as

$$\begin{aligned}
a_{11}x_1 &+ a_{12}x_2 &+ a_{13}x_3.....a_{1m}x_n &= b_1 \\
a_{21}x_1 &+ a_{22}x_2 &+ a_{23}x_3.....a_{2m}x_n &= b_2 \\
a_{31}x_1 &+ a_{32}x_2 &+ a_{33}x_3.....a_{3m}x_n &= b_3 \\
. \\
. \\
. \\
. \\
a_{n1}x_1 &+ a_{n2}x_2 &+ a_{n3}x_3.....a_{nm}x_n &= b_n
\end{aligned} \tag{1}$$

Here, the a's and b's represent the constants. If all b's are zero then this is called a **homogeneous system**, which has a trivial solution of all variables being zero. On the other hand, if one of the b's is non-zero, then a non-trivial solution exists and this is then called a **non-homogeneous system**. As far as this lab goes, we will like to set up systems such that $m = n$ so that we can solve these equations to obtain solutions.

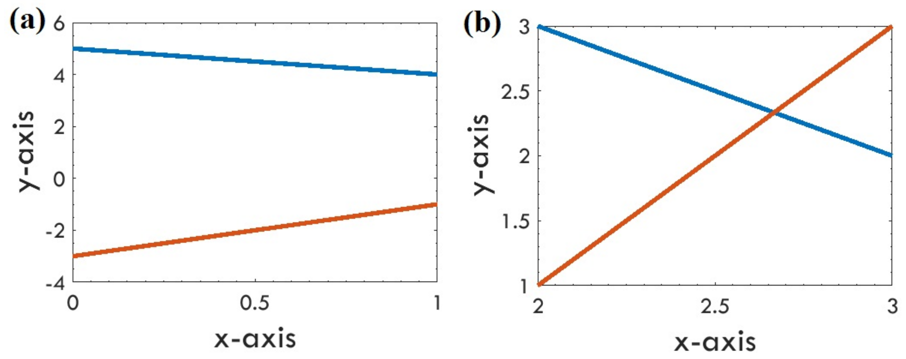

Figure 1: Plot of the two equations (a) between [0,1] and (b) [2,3]. The intersection point can be seen in (b).

2 Solving linear equations

For simplicity, consider a set of two linear equations with two variables, x and y. We will consider a situation where an unique solution exists

$$\begin{aligned} x + y &= 5 \\ 2x - y &= 3 \end{aligned} \tag{2}$$

There are primarily three ways to solve this set.

1. **Gauss elimination**: This technique best for systems with small number of variables.

2. **Graphical representation**: This works best for two variable systems. For practical solutions, it is necessary to have some idea of the range of the values for these variables.

3. **Matrix representation**: This is a more general form of setting up and solving a system of linear equations.

Consider the above example. For **Gauss elimination**, it is easy to write y in terms of x, using the first equation ($y = 5 - x$) and then substitute and solve in the second equation ($x = \frac{8}{3}$ and $y = \frac{7}{3}$). In the case of graphical representation, both equations can be plotted (in MATLAB this would be the *plot* command) and the intersection point is the solution. However, some idea of the range is need to obtain the intersection point. This can be seen in the figure 1 below. The graphical approach is suited for situations where visualization of the lines are helpful. For linear systems with more than two variables and for general situations the matrix representation is the more suited approach.

3 Matrix representation

Consider again the two equations shown in (2). It is possible to write them in the form $Ax = b$, where A is a 2×2 matrix and x and b are 2×1 matrices as shown below

$$A = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix}, \text{and } b = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

The solution to this matrix equation is then just

$$X = A^{-1}b \tag{3}$$

where A^{-1} is the inverse of the 2×2 matrix A. Using MATLAB, the inverse and product of matrices can be easily performed. The advantage of this matrix method is that it allows us to solve an $n^{th}-$order linear equation system (n variables and n equations). A unique solution is possible if the equations are linearly independent, the determinant formed by the matrix is non-zero.

4 Numerical methods for solving linear equations

MATLAB can be used for solving a system of linear equations using the matrix representation. The advantage of this approach is that equations with large number of variables can be solved very quickly and efficiently. The crucial step is the inversion of the original A matrix in the system $Ax = b$, so that the solution x can be obtained by $A^{-1}b$. It would be instructive to understand the procedure for solving this system numerically, since these algorithms are implemented in packages such as MATLAB.
There are different approaches for solving a system of linear equations numerically. We will discuss primarily two techniques

1. **LU decomposition**
2. **Gauss elimination**

Both techniques are used in MATLAB to solve a system of linear equations.

4.1 LU decomposition

This technique involves breaking down a matrix A (to make life simpler let us make this a square matrix) into two matrices, L and U, which are a lower and upper triangular matrix respectively.

$$\begin{aligned} Ax &= b \\ A &= L\,U \\ L\,Ux &= b \\ Ly &= b \\ Ux &= y \end{aligned} \tag{4}$$

The advantage of this process is that these triangular matrices can be solved more easily by substitution. For example, consider the following 3×3 matrix.

$$\begin{bmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{bmatrix}$$

We can write this as the product of two matrices

$$\begin{bmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ 0 & l_{22} & l_{23} \\ 0 & 0 & l_{33} \end{bmatrix}$$

It is possible to solve these by simple inspection or by writing a suitable algorithm. In MATLAB, when a system of linear equations is solved using the matrix division symbol ($x = A\backslash b$) LU decomposition is used to calculate the values.

4.2 Gauss-Jordan elimination

Gauss-Jordan elimination is a more general route to calculate the inverse of a matrix so that the linear equations can be solved. This method is used in MATLAB to calculate the inverse of a matrix. Consider the matrix given below.

$$\begin{bmatrix} -1 & 1 & 2 \\ 3 & -1 & 1 \\ -1 & 3 & 4 \end{bmatrix}$$

To calculate its inverse, in the first step, an unit matrix is added to form what is called an **augmented matrix**. This is written as

$$\begin{bmatrix} -1 & 1 & 2 & | & 1 & 0 & 0 \\ 3 & -1 & 1 & | & 0 & 1 & 0 \\ -1 & 3 & 4 & | & 0 & 0 & 1 \end{bmatrix}$$

The next step is to convert the matrix in the left to an upper triangular matrix (by elimination) and then making all the diagonal elements equal to 1. The important point to remember is that the same operations must be carried out in the matrix on the right. This gives the following matrix

$$\begin{bmatrix} 1 & -1 & 2 & | & -1 & 0 & 0 \\ 0 & 1 & 3.5 & | & 1.5 & 0.5 & 0 \\ 0 & 0 & 1 & | & 0.8 & 0.2 & -0.2 \end{bmatrix}$$

This upper triangular matrix is then converted to a unit matrix and the matrix on the right now becomes the inverse matrix. The final answer is

$$\begin{bmatrix} 1 & 0 & 0 & | & -0.7 & 0.2 & 0.3 \\ 0 & 1 & 0 & | & -1.3 & -0.2 & 0.7 \\ 0 & 0 & 1 & | & 0.8 & 0.2 & -0.2 \end{bmatrix}$$

This can be checked by using the *inv* command in MATLAB.

References

The following source materials were used for preparing this handout

1. *Advanced Engineering Mathematics* by Kreyszig, Wiley & Sons, Inc., 8th Edition, 2001

2. *Numerical Methods for Engineers* by Chapra and Canale, Mc Graw Hill, 6th Edition, 2010

The PDF was prepared in LaTeX, using TeXstudio and MiKTeX compiler.