MATAB PROGRAMMING (15A04605)

LECTURE NOTES

B.TECH

III-YEAR & II-SEM

Prepared by:

Mrs. C. Geetha, Associate Professor

Department of Electronics and Communication Engineering

VEMU INSTITUTE OF TECHNOLOGY

(Approved By AICTE, New Delhi and Affiliated to JNTUA, Ananthapuramu)
Accredited By NAAC & ISO: 9001-2015 Certified Institution
Near Pakala, P. Kothakota, Chittoor- Tirupathi Highway
Chittoor, Andhra Pradesh - 517 112
Web Site: www.vemu.org

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech III-IISem. (ECE)

L	**T**	**P**	**C**
3	**1**	**0**	**3**

15A04605 MATLAB PROGRAMMING

(CBCC-I)

Objectives:

- Understand the MATLAB Desktop, Command window and the Graph Window
- Be able to do simple and complex calculation using MATLAB
- Be able to carry out numerical computations and analyses
- Understand the mathematical concepts upon which numerical methods rely
- Ensure you can competently use the MATLAB programming environment
- Understand the tools that are essential in solving engineering problems

1. **UNIT-I:Introduction to MATLAB**

MATLAB Interactive Sessions, Menus and the toolbar, computing with MATLAB, Script files and the Editor Debugger, MATLAB Help System, Programming in MATLAB.

2. **UNIT-II:Arrays**

Arrays, Multidimensional Arrays, Element by Element Operations, Polynomial Operations Using Arrays, Cell Arrays, Structure Arrays.

3. **UNIT-III:Functions & Files**

Elementary Mathematical Functions, User Defined Functions, Advanced Function Programming, Working with Data Files.

4. **UNIT-IV:Programming Techniques**

Program Design and Development, Relational Operators and Logical Variables, Logical Operators and Functions, Conditional Statements, Loops, the Switch Structure, Debugging Mat Lab Programs. Plotting :XY- plotting functions, Subplots and Overlay plots, Special Plot types, Interactive plotting, Function Discovery, Regression, 3-D plots.

5. **UNIT-V:Linear Algebraic Equations**

Elementary Solution Methods, Matrix Methods for (Linear Equations), Cramer's Method, Undet- ermined Systems, Order Systems.

TEXT BOOKS:

1. G. H. Golub and C. F. Van Loan, Matrix Computations, 3rd Ed., Johns Hopkins University Press, 1996.
2. B. N. Datta, Numerical Linear Algebra and Applications, Brooks/Cole, 1994 (out of print) 3. L. Elden, Matrix Methods in Data Mining and Pattern Recognition, SIAM Press, 2007

Misc. Useful Information:

- NA-digest, http://www.netlib.org/na-digest-html
- Society for Industrial and Applied Mathematics (SIAM), see http://www.siam.org
- Google "MATLAB Primer" or "MATLAB Tutorial" and you should be able to access lots of free MATLAB.

VEMU INSTITUTE OF TECHNOLOGY

Approved by AICTE and Permanently Affiliated to JNTUA, Ananthapuramu,

DEPARTMENT OF ELECTONICS AND COMMINICATION ENGINEERING

Course: B.TECH III Year II SEM **A.Y: (2018-2019)**

SUB: MATLAB PROGRAMMING (15A04605)

Branch: ECE **CDF** **Faculty:Mrs.C.GEETHA**

UNIT: 1

INTRODUCTION TO MATLAB

What is MATLAB:

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming. MATLAB is developed by Math Works.

platforms supported by Matlab:

Ans: MATLAB supports almost every computational platform. It supports all operating systems such as

- Windows
- AIX
- HP UX(Including UX10 and UX11)(Hewlett Packard Unix")
- IRIX(**IRIX** is a discontinued operating system developed by Silicon Graphics (SGI) to run on their MIPS workstations and servers.)
- IRIX64
- Linux
- Solaris(**Solaris** is a Unix operating system originally developed by Sun Microsystems.)
- Mac OS and
- Open VMS.(virtual memory system)

Features of mat lab:

The Most important features of Matlab are

- It is a high level programming language.
- Provides an interactive environment for exploration, design and problem solving.

- Provides a vast library of built in mathematical functions.
- Provides built-in graphic tools for creation of custom plots.
- Provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

Applications:

MATLAB is widely used in a range of applications including:

- Signal processing and Communications
- Image and video Processing
- Control systems
- Test and measurement
- Computational finance
- Computational biology

Advantages of mat lab

1. Ease of Use
2. Platform Independence
3. Predefined Functions
4. Device Independent Plotting
5. Graphical User Interface
6. Matlab Compiler

Disadvantages of mat lab

- It is an interpreted language and therefore may execute more slowly than compiled languages.
- The second disadvantage is cost
- Fortunately, there is also an inexpensive Student Edition of MATLAB, which is a great tool for students wishing to learn the language.
- The Student Edition of MATLAB is essentially identical to the full edition.

MATLAB INTERACTIVE SESSIONS

Windows of mat lab environment:

MATLAB works through three basic windows. They are

1) Matlab desktop.

2.) Editor Window.

3.) Figure Window.

Sub windows of Matlab Desktop:

The major tools within or accessible from the MATLAB desktop are

- ❖ The Command Window.
- ❖ The Command History Window.
- ❖ The Start Button.
- ❖ Workspace Browser.
- ❖ Help Browser.
- ❖ Path Browser/Current Directory/Current Folder

Command window:

The right-hand side of the default MATLAB desktop contains the Command Window. A user can enter interactive commands at the command prompt (») in the Command Window, and they will be executed on the spot.

Workspace:

A workspace is the collection of all the variables and arrays that can be used by MATLAB when a particular command, M-file, or function is executing.

Editor window:

This is where you write, edit, create, and save your own programs in files called 'M-files'.
You can use any text editor to carry out these tasks. On most systems, MATLAB provides its own built-in editor.
Number is entered as the value of a variable, MATLAB automatically sets the variable to be real (double).

Editor acts a debugger (Built in editor debugger):

In Matlab software the Matlab editor itself acts as a debugger and allows us to debug the program through the screen. In the editor while we are typing the program if we type any command erroneously, then that error will be notified with a red tricky line beneath that command and provides us a flexibility to clear those errors immediately.

Figure window:

In order to output or display the graphic related data we will use figure window. All waveforms, images and videos will be displayed in the figure window.

Script and function:

Scripts are m-files containing MATLAB statements. MATLAB ``functions'' are another type of m-file. The biggest difference between scripts and functions is that functions have input and output parameters. Script files can only operate on the variables that are hard-coded into their m-file.

Matlab file types:

MATLAB has five types of files for storing information:

1.) M-files: are standard ASCII text files, with a .m extension to the filename

2.) Mat-files: are binary data-files, with a .mat extension to the filename.

3.) Mex-files: are MATLAB-callable FORTRAN and C programs, with a .mex extension to the filename.

4.) P-files: are the compiled M-files with a '.p' extension that can be executed in MATLAB directly.

5.) Fig files: are the binary figure files with a '.fig' extension that can be opened again in Matlab as figure.

Examples:

1) firstfile.m- Example for M-File.

2) firstfile.mat-Example for Mat File.

3) firstfile.fig-Example for 'Fig' file.

4) firstfile.p-Example for P-file.

5) firstfile.mex-Example for mex file.

Menus and Tool Bars:

Matlab Editor Window contains the following menus.

File menu ,Edit Menu ,Debug ,Parallel ,Desktop ,Window ,Help

Matlab Editor Window contains the following menus.

File Menu, Edit Menu, Text Menu, Go

Matlab figure window contains the following menus.

File Menu, Edit Menu,   View Menu, Window, Help

<u>Matlab Editor Debugger:</u>

To debug your MATLAB® program graphically, use the Editor/Debugger. Alternatively, you can use debugging functions in the Command Window.

Both methods are interchangeable. Before you begin debugging, make sure that your program is saved and that the program and any files it calls exist on your search path or in the current folder.

If you run a file with unsaved changes from within the Editor, then the file is automatically saved before it runs.

<u>Matlab Help System:</u>

There are three ways to get help in MATLAB. The preferred method is to use the Help Browser. The Help Browser can be started by selecting the icon from the desktop toolbar or by typing helpdesk or helpwin in the Command Window.

UNIT-2

Array:

One of the strengths of MATLAB is the capability to handle collections of items, called arrays, as if they were a single entity it is called one dimensional.

Two-Dimensional Arrays:

An array having rows and columns is a two-dimensional array that is sometimes called a matrix.

Ex: [1 2 3; 5 6 7]

Creating Matrices:

The most direct way to create a matrix is to type the matrix row by row, separating the elements in a given row with spaces or commas and separating the rows with semicolons.

Brackets are required. For example, typing

>>A = [2,4,10;16,3,7];

Ans is

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 10 \\ 16 & 3 & 7 \end{bmatrix}$$

Matrices and the Transpose Operation:

Ex:

$$\mathbf{A} = \begin{bmatrix} -2 & 6 \\ -3 & 5 \end{bmatrix} \qquad \mathbf{A}^T = \begin{bmatrix} -2 & -3 \\ 6 & 5 \end{bmatrix}$$

- Transpose operator (.’)

Colon operator:

- v(:) represents all the row or column elements of the vector v.
- v(2:5) represents the second through fth elements; that is v(2), v(3),v(4), v(5).
- A(:,3) denotes all the elements in the third column of the matrix A.
- A(3,:) denotes all the elements in the third row of A.
- A(:,2:5) denotes all the elements in the second through fth columns of A.
- A(2:3,1:3) denotes all the elements in the second and third rows that are also in the first through third columns.
- v = A(:) creates a vector v consisting of all the columns of A stacked from first to last.
- ■ A(end,:) denotes the last row in A, and A(:,end) denotes the last column.
- You can use array indices to extract a smaller array from another array. For example, if you create the array B

$$\mathbf{B} - \begin{bmatrix} 2 & 4 & 10 & 13 \\ 16 & 3 & 7 & 18 \\ 8 & 4 & 9 & 25 \\ 3 & 12 & 15 & 17 \end{bmatrix}$$

by typing

```
>>B = [2,4,10,13;16,3,7,18;8,4,9,25;3,12,15,17];
```

and then type

```
>>C = B(2:3,1:3);
```

You can produce the following array:

$$\mathbf{C} = \begin{bmatrix} 16 & 3 & 7 \\ 8 & 4 & 9 \end{bmatrix}$$

Multidimensional Numeric Arrays:

A three-dimensional array has the dimension m X n X q. A four-dimensional array has the dimension m X n X q X r, and so forth. The first two dimensions are the row and column, as with a matrix.

The higher dimensions are called pages. You can think of a three-dimensional array as layers of matrices. The first layer is page 1; the second layer is page 2, and so on. If A is a 3 X 3 X 2 array, you can access the element in row 3, column 2 of page 2 by typing A(3,2,2).

Array Addition and Subtraction:

Ex: r = [3,5,2]

v = [2,-3,1]

type w = r + v.

The result is w = [5,2,3].

Element-by-Element Multiplication:

MATLAB defines element-by-element multiplication only for arrays that are the same size.

The definition of the product x.*y, where x and y each have n elements, is x.*y = [x(1)y(1), x(2)y(2) . . . , x(n)y(n)]

polynomials.

MATLAB commands used in polynomials.

Ans: conv(a);

Polyval(a);

Roots(a);

Polyder(a);

Poly(a);

CELL ARRAYS.

MATLAB commands used in cell arrays.

celldisp(a);

Cellplot(a);

Iscell(a);

Cell2mat(a);

Mat2cell(a);

How to update the CELL

Let A= [1] [2]   [3]

‘One’ ‘two’ ‘three’

The second row is replaced by

Output:

‘first’ ‘second’ ‘third’

A= [1]   [2]   [3]

‘First’ ‘second’ ‘third’

STRUCTURE ARRAYS.

A structure is a data type that groups related data using data containers called fields. Each field can contain any type of data. Access the data in a field using dot(.) notation of the form

Structure name. Field name

Syntax: Structure name. field name=’fieldsdata’

- Access the particular data from the structure array by using the command ‘get field’

Syntax: getfield(structure name, ’fieldname’);

MATLAB code to display the student details using structure arrays:

```
clc;
    clear all;
    close all;
    Student.name=’venkateswarlu’;
   Student.rollnumber=’184M1A0444’;
   Student.year=’3rd’;
   Student.percent=’67%’;
   Student.mail=’rapallivenkatesh@gmail.com’;
```

Code to display bike details using struct keyword:

```
%bike details

Struct(‘bikecomp’,’royalenfield’,’cc’,350,’milage’,40,’price’,’twolakhs’,’color’,’black’,’rc’,’
AP 26 S 6100’);
```

UNIT-III FUNCTIONS & FILES

Elementary Mathematical Functions

Define a function It consists of a single MATLAB expression and any number of input and output arguments. You can define an anonymous function right at the MATLAB command line, or within a function or script. This gives you a quick means of creating simple functions without having to create a file for them each time

Run Functions in the Editor

- Create a function in a program file named myfunction.m
- View the commands available for running the function by clicking Run on the Editor tab.
- Replace the text type code to run with an expression that allows you to run the function.

Mathematical function

In mathematics, a function is a relation between a set of inputs and a set of permissible outputs with the property that each input is related to exactly one output. An example is the **function** that relates each real number x to its square x^2.

Elementary number functions

abs - absolute value of a real number or magnitude of a complex number

ceil - ceiling function: rounds up a floating-point number

floor - rounds down a floating-point number

fix - rounds a floating-point number towards zero

mod - arithmetic remainder function

rem – remainder after division

rat - rational approximation of a floating-point number

round - rounds to nearest integer

sign - sign function

sqrt - square root

User defined function

def: A user-defined function is a MATLAB program that is created by the user, saved as a function file, and then can be used like a built-in function. A function in general has input arguments (or parameters) and output variables (or parameters) that can be scalars, vectors, or matrices of any size. There can be any number of input and output parameters, including zero.

Advanced Function Programming:

Anonymous functions

Defineition

- An anonymous function is a simple, typical a single line, user-defined function that is defined and written within the computer code (not in a separate file) and is then used in the code.
- An anonymous function is created by typing the following statement:

functionName = @ (var1,var2,...) expression.

Inline functions

Similar to anonymous function, inline function is a simple single-line user-defined function that is defined without creating a separate function file (M-file). Inline functions are gradually being replaced by anonymous functions.

Ex: name = inline('mathematical expression typed as a string')

CUBE = inline('X.^3');

Recursive functions:

Functions can be recursive, that is, they can call themselves. Numerical quadratures are techniques for numerically computing the integrals of functions. I wrote two such quadratures based on the Gauss-Legendre algorithm

one using a 10-point and the other one a 12-point formula:

gauss10pts(fcn, a, b)

gauss12pts(fcn, a, b)

Each of these functions compute numerically the integral of the function fcn from a to b.

Nested functions

You can define functions within the body of another function. These are called nested functions.A nested function follows the following syntax

```
function x = A(p1, p2)
...
B(p2)
  function y = B(p3)
  ...
  end
...
end
```

Private functions:

A private function is a primary function that is visible only to a limited group of other functions. If you do not want to expose the implementation of a function(s), you can create them as private functions. Private functions reside in **subfolders** with the special name **private**. They are visible only to functions in the parent folder.

Example

Let us rewrite the *quadratic* function. This time, however, the *disc* function calculating the discriminant, will be a private function.Create a subfolder named private in working directory. Store the following function file *disc.m* in it −

```
function dis = disc(a,b,c)
%function calculates the discriminant
dis = sqrt(b^2 - 4*a*c);
end     % end of sub-function
```

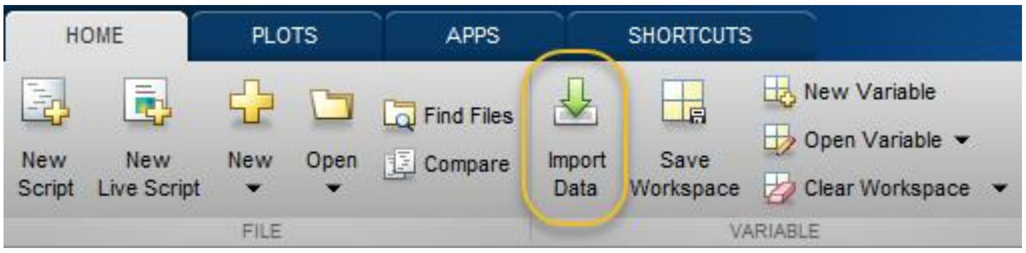
Working with Data Files:

Global Variables:

Global variables can be shared by more than one function. For this, you need to declare the variable as global in all the functions. If you want to access that variable from the base workspace, then declare the variable at the command line. The global declaration must occur before the variable is actually used in a function. It is a good practice to use capital letters for the names of global variables to distinguish them from other variables.

import and export in MATLAB:

Data import and export functions provide access to data from files, other applications, web services, and external devices. You can read popular file formats, such as Microsoft® Excel® spreadsheets, text, images, audio and video and scientific data formats. Low-level file I/O functions let you work with data files in any format.



How to create an Excel spreadsheet from a text file

In order to manipulate data and have more flexibility in formatting, it is often desired to import the data into a spreadsheet. If the data are in fixed-width columns, perform the following:

1. Open Excel
2. File – Open – Files of type – select All Files
3. Open the .txt file
4. When prompted, choose Fixed Width
5. Adjust the width of the break lines if necessary
6. Next
7. Column data format – General
8. Finish

Commands required to perform I/O operations on text files:

fopen()- Opens a text file with a file identifier FILEID and character encoding.

fscanf()- Used to read the data from a text file.

fprintf()-Used to write the data into a text file.

fclose()- Used to close the opened text file.

UNIT-04: PROGRAMMING TECHNIQUES

Program Design and Development,:

design in matlab:

A systematic hierarchical planning of operations required to be performed in the program to be developed as a step-by-step procedural algorithm is called the program design and program plan of the program under development.

development in matlab:

After completing the program design each and every step of the program design is transformed into its equivalent matlab code or instructions. This process of writing an equivalent matlab code or instructions to each and every step of the program design is called

Relational and Logical Operators

Def: The operators which are used to find the relationship between the two operands are called the relational operators. The relationship can be equal to,not equal to, less than, greater than, less than or equal to and greater than or equal to.

Relational Operators:

1.) ==(equal to)	- Determines equality.
2.) >>(greater than)	-Determines greater than.
3.) <<(less than)	-Determines less than.
4.) >=(Greater than or equal to)	- Determines greater than or equal to.
5.) <=(Less than or equal to)	- Determines less than or equal to.
6.) ~=(Not equal to)	-Determines not equal to.

Logical operators

Logical Operator		Description
1.) &(And)	-	Logical AND
2.) |(Or)	-	Logical OR
3.) ~(Not)	-	Logical NOT
4.) Xor	-	Logical XOR.

Logical variables

Logical Variable	Description
1.)all	- Determines whether all the array elements are non-zero or true.
2.)any	-Determines if any array element is non-zero or true.
3.)find	-Finds the indices and values of the non-zero array elements.
4.)false	

<u>Conditional Statements:</u>

if-conditional statements with neat syntaxes

If conditional statements are used to execute a block of instructions if the specified condition is met.If conditional statements are classified into five types they are.

1.)If statements

2.)If-else statements

3.)if-elseif statements

4.)if-elseif-else statements.

5.)Nested-if statements

syntax of switch conditional statement?

The syntax of switch conditional statement is

switch *switch_expression*

```
  case case_expression
    statements
  case case_expression
    statements
   ...
  otherwise
    statements
end
```

Syntax of nested-if structures

The syntax for a nested if statement is as follows −

```
if <expression 1>
  % Executes when the boolean expression 1 is true
  if <expression 2>
     % Executes when the boolean expression 2 is true
 end
end
```

Loops

example for for-loop:

The syntax of for-loop is

```
for index = values
  statements
end
```

Example:

```
              s = 10;
                          H = zeros(s);
                          for c = 1:s
```

```
    for r = 1:s
    H(r,c) = 1/(r+c-1);
   end
  end
```

example for while-loop:

The syntax of while loop is

```
  while expression
statements
  end
```

Example:

```
n = 10;
f = n;
while n > 1
n = n-1;
f = f*n;
end
disp(['n! = ' num2str(f)])
```

XY- plotting functions:

XY-plotting functions in matlab

XY plotting functions are used to create linear 2D plots of the linear functions of the form Y=f(X).Such linear functions of the form Y=f(X) are called the XY-plotting functions , where 'X' is the independent variable vector and Y is the dependent variable vector. The Matlab functions which are used to plot such XY-linear functions are called the XY-Plotting functions in Matlab.

Subplot with syntax:

Subplot () command divides the figure window into rectangular panes that are numbered row wise. Each pane contains an axes object. Subsequent plots are output to the current pane. The syntaxes of subplot command are

Subplot(m,n,p)

Subplot(mnp)

Subplot(m,n,p,'replace')

Subplot(m,n,p,'align')

Subplot(h)

overlay plots:

Overlay plots are also called as the overlapped plots. Subplots and Overlay Plots. MATLAB can create figures that contain an array of plots, called subplots. These. are useful' when you want to compare the same data plotted with different axis. types, for example.

different methods to generate overlay plots:

There are three different ways of generating the overlay plots. They are

(i)By using a single plot() command with different data arguments.

(ii)By using hold command.

(iii) By using line command.

Special Plot types:

Special 2d functions:

2D plotting function	Description
area()	-Creates a filled area plot.
bar()	-Creates a bar graph.
barh()	-Creates a horizontal bar graph.
Comet()	-makes an animated 2D plot.
Compass()	-Creates arrow graph for complex numbers.
Contour()	-Makes contour plot.

Basic graphic command

The syntax of plot command is

plot(X,Y)

plot(X,Y,LineSpec)

plot(X1,Y1,...,Xn,Yn)

plot(X,Y) creates a 2-D line plot of the data in Y versus the corresponding values in X. If X and Y are both vectors, then they must have equal length. The plot function plots Y versus X. If X and Y are both matrices, then they must have equal size. The plot function plots columns of Y versus columns of X.

3D plotting functions

2D plotting function	**Description**
plot3()	-plots curves in space.
stem3()	-Creates a discrete data plot with stems in 3D.
bar3()	-Creates a 3D bar graph.
pie3()	-makes a 3D pie chart.
comet3()	-makes an animated 3D plot.
fill3()	-draws a filled 3D polygons.

Regression in matlab:

A data model explicitly describes a relationship between predictor and response variables. Linear **regression** fits a data model that is linear in the model coefficients. ... You also can use the **MATLAB** polyfit and polyval functions to fit your data to a model that is linear in the coefficients.

UNIT-05-Linear Algebraic Equations

Elementary Solution Methods:

Define Linear System of Equations

Solving Systems by addition. A "system" of equations is a set or collection of equations that you deal with all together at once. Linear equations (ones that graph as straight lines) are simpler than non-linear equations, and the simplest linear system is one with two equations and two variables.

Linear equation in standard form

- Find the slope using the formula.
- Write the equation into y = mx + b using y - k = m (x - h).
- Change the equation into standard form Ax + By = C.

Linear system: A linear system is a mathematical model of a system based on the use of a linear operator. Linear systems typically exhibit features and properties that are much simpler than the nonlinear case

Rank of a matrix:

The maximum number of linearly independent vectors in a matrix is equal to the number of non-zero rows in its row echelon matrix. Therefore, to find the rank of a matrix, we simply transform the matrix to its row echelon form and count the number of non-zero rows.

Using Cramer's Rule
to Solve linear Equations

Consider the following set of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

The system of equations above can be written in a matrix form as:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \; and \; [B] = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

If $D \neq 0$, then the system has a unique solution as shown below (Cramer's Rule).

$$x_1 = \frac{D_1}{D}, \quad x_2 = \frac{D_2}{D}, \quad x_3 = \frac{D_3}{D}$$

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{32} & a_{33} \end{vmatrix} \qquad D_1 = \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}$$

$$D_2 = \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{12} & b_2 & a_{23} \\ a_{13} & b_3 & a_{33} \end{vmatrix} \qquad D_3 = \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{12} & a_{22} & b_2 \\ a_{13} & a_{32} & b_3 \end{vmatrix}$$