3.0 Introduction

World is zero without mathematics, mathematics plays an important role in various subjects of all branches. It is useful to human being in daily life. Mathematics is the language of sciences. It is also called as Queen of the sciences. As such it may regard as the most fundamental branch of sciences. Role of mathematics is very important to develop the skills and logical thinking of the students as well as the teachers. Nowadays technology is getting at very rapid rate. The World Wide Web has transferred the world into global village. The internet has become the world’s biggest library and the great source of information. It is the knowledge bank for the younger generation. There are many e-resources available for learning, teaching and research activities in mathematics on the internet. There are highly technical and advanced software in mathematics such as Mathematical Laboratory, Maple, Sage, Science Laboratory, Maxima, CAS (Computer Algebra Systems) etc. There are also specific software designed for many specialized topics.

This chapter contains objective and importance of mathematical software / model, planning for study, choice of proper mathematical software, theoretical basic concepts for preparation of mathematical model, selection of topics for teaching programme and pilot study.
3.1 Objectives of Mathematical Model

Efficacy of mathematical software should focus on the following objectives:

1. Apply Theoretical Concepts

Mathematics teaching should involve the application of knowledge; use of theoretical concept and theoretical evaluation of the results obtained by the mathematical software / model. This shall enable the slow learners to achieve the following objectives:

- Define a problem to be investigated.
- Formulate objectives for possible results,
- Plan procedure to study the objectives,
- Design observation,
- Select suitable mathematical software and mathematical model.

2. Develop Manipulative Apparatus

For the learning of mathematics through mathematical software / model students shall able to:

- Develop basic information about software / model: information about instrument’s parts, how it work, how to handle it and how to operate it,
- Develop manipulative skill: setting up apparatus, describe function of each part and key buttons,
- Work according to design.

3. Develop affective skills

Developing the positive attitude towards mathematical software / model and attitude as persistence, originality and commitment to learning mathematics etc. shall be important objectives of efficacy of mathematical software / model. So mathematical software / model should help students to:
• Develop initiative and resourcefulness,
• Develop ability to co-operate to each other,
• Inculcate the value of work,
• Develop skill of communication.

This study teaching programme is developed by us by keeping all the above objectives in mind. The programme is developed to promote mathematical thinking among the slow learners. This general aim leads to the following specific objectives:

1) To develop mathematical model that is simple to understand and operate,
2) To choose mathematical software that give importance to the slow learner,
3) To choose mathematical software that is easy to operate,
4) To choose mathematical software / model that can be use by ordinary teacher in an ordinary classroom.

3.2 Importance of Mathematical Model

The traditional teaching method like lecture with chalk, talk and blackboard is used everywhere. As the result, the mathematical understanding and logical thinking are not developed among the learners. Students typically have very little time to interact with materials or the teacher and therefore they show less interest in classroom towards mathematics learning. The quality of teaching is directly proportional to student’s performance. These teaching methods fail to develop the skills such as skill of formulating, solving problems from out of syllabus and changing fields, interest in subject knowledge and flexibility across application to other subject, practices with computation, logical thinking and communication skills of spoken and written. But technology supported teaching and learning method is more beneficial to students. Here technology means that are used to support teaching and learning processes, such as computer, smart board, class pad, mathematical software / model, internet and tele-communication system. The use of technology has become increasingly popular in education system overall the past several years.
There are many e-resources available for teaching & learning and research activities in mathematics on the internet. There are highly technical and advanced software in mathematics such as MATLAB, Maple, Sage, Maxima, Scilab, CAS, fxCG-20 etc. There are also specific software designed for many specialized topics in mathematics. From these software, there are many freeware, applets and utilities available on the internet. Mathematics software / model has been viewed as potential tool for helping students increase knowledge in mathematics, motivation, develop thinking ability, gain a deeper understanding of concepts and develop better problem solving skills. The use computer, scientific calculator, smart board and mathematical software / model in education have grown tremendously over the few years.

Mathematical model is the expressions involving relationship either equations or inequalities representing variables and parameters that defines a particular phenomenon. Mathematical models are built by using symbols and numbers that can be transfer into functions, formulae, inequalities and equations. They also can be used to build much more complex model like linear programming model. Some models have been developed and used in education field for effective teaching processes. It is the process of scientific method and formal part of the curriculum for mathematics. A variety of models are possible, each of which may be use for different applications in different fields. The user can solve the mathematical model by utilizing simple techniques such as addition and multiplication or more complex techniques such as matrix algebra or Gauss-Jordan elimination. Mathematical models frequently are easy to handle and manipulate, they are appropriate for use with calculators or algorithms or computer programs.

### 3.3 Planning

In the beginning we studied the use of various advanced technology. We found that very less work has been done on use of mathematical software / model at undergraduate level in this area in India and abroad. We attended 4 days workshop on
mathematical software at Bhashkarachray Pratishtan in Pune. Even we organized two workshops on mathematical software in rural and urban areas. We also conduct some practicals by using mathematical software like Scilab and C-program for S. Y. B. sc. (Computer Science) class. So we have decided to work on mathematical software and collect the freeware as well as commercial mathematical software in our college computer laboratory, also developed new mathematical model. FxCG-20 mathematical software (Calculator) was downloaded from internet in the computer laboratory and also 23 mathematical software fxCG-20 (calculator) purchases in our computer laboratory.

3.4 Topics to Teach

The mathematical models and software has become popular tool to analyze logical thinking in mathematics to undergraduate students. Students are able to develop logical thinking and an important basic mathematical concept when they working with appropriately designed “real life” problems, and by this way their motivation for the increased in mathematics interest. Here we are interested the use of mathematical software / models to teach Linear Algebra at undergraduate slow learners. Linear Algebra has been one of the important courses in mathematics for a variety of disciplines, and this has become a compulsory subject in many syllabi at undergraduate and postgraduate level in every university. It has also been recognized that Linear Algebra is difficult to understand for most of the students at undergraduate level.

The implementation of the mathematical software / modeling approach includes in teaching learning processes that promote significant development of mathematical reasoning in meaningful situation. This research study has the potential of providing valuable insight for curriculum developers and teachers. The use of mathematical model in teaching methods should give the mathematics teachers the opportunity for the observations of how students understand and think about mathematical concepts. Since the construction of theoretical concepts in mathematics is known to be a difficult process, we consider that the use of mathematical software / model in teaching learning processes can provide the motivation for students to apply their knowledge for solving the problems from other fields and adopt the new concepts in mathematics. Use of
mathematical software / model in teaching learning processes would help the students make the necessary constructions to understand theoretical concepts from Linear Algebra.

We prepared four topics from Linear Algebra of S. Y. B. Sc. Class. These topics consists 9 subtopics and each subtopic consist different types of examples. These topics, subtopics are given below.

1. **System of linear Equations**
   
   a) Consistent and inconsistent system of equation.
   
   b) Homogenous system of linear equations
   
   c) Non homogenous system of linear equations

2. **Linear Independence**

   a) Linear combination and spanning
   
   b) Linear dependence and independence

3. **Basis and Dimensions of Vector Space**

   a) Basis of vector space
   
   b) Dimension of a vector space

4. **Row Space, Column Space, Null Space, Rank and Nullity**

   a) Row space, Column Space and Null space
   
   b) Rank and Nullity

**3.5 Format of Mathematical Software / Model**

3.5.1 **Mathematical Model-Basic Linear Algebraic software**

The modeling means the study of objects and methods in one Physical environment by using objects and methods in other physical environment. It is “the
A method of scientific inquiry” and formal part of curriculum for mathematics. Mathematical models are built by using symbols and numbers that can be transfer into functions, formulae, inequalities and equations.

1. System of linear equations

A finite set of linear equations in n unknowns $x_1, x_2, \ldots \ldots \ldots x_n$ is called system of linear equations. A sequence of numbers $s_1, s_2, \ldots \ldots \ldots s_n$ is called solution of the system if $x_1 = s_1, x_2 = s_2, \ldots \ldots \ldots x_n = s_n$. A system of linear equations that has no solutions is said to be inconsistent and there is at least one solution it is said to consistent. Every system of linear equations either consistent or inconsistent, if it is consistent then it has either unique solution or infinitely many solutions.
Consistent and inconsistent system of two linear equations in two unknowns geometrically interpreted in next chapter.

Consider general system of two equations in two variables $x$ and $y$.

\[ a_1x + b_1y = c_1 \quad (a_1, b_1 \text{ both are not zero}) \]

\[ a_2x + b_2y = c_2 \quad (a_2, b_2 \text{ both are not zero}) \]

These two equations are represented by lines $l_1$ & $l_2$ in two dimensional planes. Since a point $(x, y)$ lies on the line if and only if $x$ & $y$ satisfies the equation of the line, the solution of the system of equations will be the points of intersection of $l_1$ & $l_2$. There are three cases arises.

**Case 1:** if lines $l_1 \& l_2$ are parallel i.e. there is no intersection and so no solution to the system. Therefore system is inconsistent.

**Case 2:** If lines $l_1 \& l_2$ may intersect at only one point then system has
Unique solution. Therefore system is consistent.

**Case 3:** If lines $l_1$ & $l_2$ may coincide, then system has infinitely many solutions. Therefore system is consistent.

Any system of $m$ linear equations in $n$ variables will be written as

$$
\begin{align*}
& a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m
\end{align*}
$$

In system of linear equations (I) if $b_1 = b_2 = \ldots = b_m = 0$ then system is called homogenous system of linear equations.

System (I) of $m$ linear equations in $n$ variables can be abbreviated by writing only the rectangular array of numbers.

$$
\begin{bmatrix}
  a_{11} & a_{12} & \ldots & a_{1n} & : & b_1 \\
  a_{21} & a_{22} & \ldots & a_{2n} & : & b_2 \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
  a_{m1} & a_{m2} & \ldots & a_{mn} & : & b_m
\end{bmatrix}
$$

This is called augmented matrix for the system (I) and it is denoted by $[A:B]$.

For solving the system of linear equations (I), $AX = B$, first reduce augmented matrix $[A:B]$ into reduced row-echelon form by elementary row operations $R_{ij}$, $R_{(k)}$ & $R_{ij(k)}$. Rewrite the system of linear equations from reduced row-echelon form and by back substitution to get the solution of the system. This method is called Gauss-Jordan elimination method. If we reduced augmented matrix into row-
echelon form and use it in above method instead of reduced row-echelon form then this method is called Gaussian elimination method.

If \( AX = B \) is a system of \( n \) equations and in \( n \) variables is given then \( A \) is square matrix of order \( n \) and \( B \) is column matrix of order \( nx1 \).

If \( |A| \neq 0 \), then \( X = A^{-1}B \) has unique solution for the system \( AX = B \).

If \( |A| = 0 \) then consider

\[
(adjA)AX = (adjA)B
\]

\[
\rightarrow [(adjA)A]X = (adjA)B
\]

\[
\rightarrow (|A|I_n)X = (adjA)B, \quad \text{Since} (adjA)A = |A|I_n.
\]

\[
\rightarrow |A|(I_nX) = (adjA)B. \quad (\ast)
\]

Now two cases arise.

**Case 1:** When \((adjA)B = 0\)

In this case, equation \((\ast)\) is true for every value of \( X \). So we say that the system of equation in (I) is consistent and it has infinitely many solutions.

**Case 2:** When \((adjA)B \neq 0\).

In this case, equation \((\ast)\) does not exist because its LHS is zero and RHS is non-zero. So we say that system (I) is an inconsistent and it has no solution.

Using above theoretical concept, we developed the following algorithm (modeling) for solving the system of linear equations \( AX = B \). Where \( A \) is \( m \times n \) matrix, \( X \) is column matrix of order \( n \times 1 \), and \( B \) is column matrix of order \( m \times 1 \).
**Algorithm 1**

**Input:** Augmented matrix $[A: B]$.

$$[A: B] = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & : & b_1 \\
a_{21} & a_{22} & \cdots & a_{2n} & : & b_2 \\
a_{m1} & a_{m2} & \cdots & a_{mn} & : & b_m \\
\end{bmatrix}$$

**Step 1:** If $n > m$, that means the number of unknowns $> the number of equations, then the system is consistent and it gives infinitely many solutions.

If $n = m$, then go to step second.

**Step 2:** Reduced the augmented matrix $[A: B]$ into row-echelon form.

Locate the leftmost column that does not consist entirely of zeros.

If $a_{i1} = 0$ and $a_{i1} \neq 0$ for $i \neq 1$,

Then interchange $i^{th}$ and $1^{st}$ row (i.e. $R_{i1}$).

Take $R_{1}(-\frac{1}{a_{11}})$, (i.e. $\frac{1}{a_{11}} R_1$)

Take $R_{i}(-a_{i1})$ (i.e. $R_i - a_{i1} R_1$) $i = 2, - - m$

**Step 3:** Now cover the row in the matrix and begin again with step 1 applied to the sub matrix that remains continue in this way until the entire matrix is in row-echelon form.
Row-echelon form of matrix = 
\[
\begin{bmatrix}
    a'_{11} & a'_{12} & \cdots & a'_{1n} & : & b'_1 \\
    0 & a'_{22} & \cdots & a'_{2n} & : & b'_2 \\
    \vdots & \vdots & \ddots & \vdots & : & \vdots \\
    0 & 0 & \cdots & a'_{mn} & : & b'_m \\
\end{bmatrix}
\]

Now three cases arises

a) If \( a'_{mn} \neq 0 \) \& \( b'_m \neq 0 \), then system is consistent and gives unique solution.

b) If \( a'_{mn} = 0 \) \& \( b'_m \neq 0 \), then system is inconsistent.

c) If \( a'_{mn} = 0 \) \& \( b'_m = 0 \), then system is consistent and gives infinitely many solution.

Rewrite the equations and by back substitution we gave the solution of the system.

**Step 4**: Stop

Using above algorithm we write program in Java programming language.

**Program 1**

//program to find solution of a system

function rowReduce(theMatrix, numRows, numCols) {

    var theNum;

    for (var i = 1; i <= numRows; i++) {

        var theCol = 0;

        for (var j = 1; j <= numCols; j++) {

            theNum = theMatrix[i][j];

        }
    }
}
if (Math.abs(theNum) < theSmallestNumber) theMatrix[i][j] = 0;

else {theCol = j; break}

}

if (theCol != 0) {theMatrix =

pivot(InMatrix, rows, cols, theRow, theCol) {

var thePivot = InMatrix[theRow][theCol];

for (var i = 1; i <= cols; i++)

{

InMatrix[theRow][i] = InMatrix[theRow][i]/thePivot;

}

var theNum = 1;

for (var i = 1; i <= rows; i++)

{

    if ((i != theRow) && (InMatrix[i][theCol] != 0) )

    {

        var factr = InMatrix[i][theCol];

        for (var j = 1; j <= cols; j++)

        {

            InMatrix[i][j] = InMatrix[i][j] - factr*InMatrix[theRow][j];

        }

    }

}
return(InMatrix);

}

}

}

var theRow = 1;

for (j = 1; j <= numCols; j++)
{
    for (i = theRow; i <= numRows; i++)
    {
        if( theMatrix[i][j] != 0)
        {
            if (j == theRow) {theRow++; break;}
            else { theMatrix = function swapRows(theMatrix, numRows, numCols, p,q)

                var rowHold =0;

                for(var j = 1; j <= numCols; j++)
{ rowHold = InMatrix[p][j];
    InMatrix[p][j] = InMatrix[q][j];
    InMatrix[q][j] = rowHold;
}
return(InMatrix); theRow++; break;
}
ddd = "The System has Infinitely many Solutions."

}

return(theMatrix);

}

}*/

2. Linear independence/ dependence

If S = \{v_1, v_2, \ldots, v_n\} is set of vectors in \(\mathbb{R}^n\), then the vector equation
\[k_1v_1 + k_2v_2 + \ldots + k_nv_n = 0,\]
has at least one solution, namely \(k_1 = 0, k_2 = 0, \ldots, k_n = 0\). If this is the only solution, then S is called linearly independent set. If there are other solutions, then S is called linearly dependent set.

For determining the set S = \{v_1, v_2, \ldots, v_n\} is linearly dependent or independent set in n-dimensional vector space V. Where \(v_1 = (a_{11}, a_{12}, \ldots, a_{1m}), v_2 = (a_{21}, a_{22}, \ldots, a_{2m}), \ldots, v_n = (a_{n1}, a_{n2}, \ldots, a_{nm})\).

First write these vectors in linear combination,
\[k_1v_1 + k_2v_2 + \ldots + k_nv_n = 0\]
\[k_1(a_{11}, a_{12}, \ldots, a_{1m}) + k_2(a_{21}, a_{22}, \ldots, a_{2m}) + \ldots + k_n(a_{n1}, a_{n2}, \ldots, a_{nm}) = (0, 0, \ldots, 0).\]

Equating corresponding components gives
In this system, coefficient matrix is of the form, \( A = [\mathbf{v}_1 : \mathbf{v}_2 : \ldots : \mathbf{v}_n] \) where the vectors \( \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \) are columns of the matrix \( A \).

Here two cases arises

**Case 1:** If \( n > m \) i.e. number of unknowns greater than the number of equations, then the system (II) has infinitely many nontrivial solutions. So the set \( S = \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) is linearly dependent set.

**Case 2:** If \( n = m \), then find determinant of the coefficient matrix and if determinant of coefficient matrix is nonzero then \( A^{-1} \) exist and system (II) has trivial solution, hence \( S = \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) is linearly independent. Also every system of linear equations of the type \( AX = B \) has unique solution \( X = A^{-1}B \), where \( B \) is a vector in \( V \) and hence \( S \) span \( V \). If determinant of coefficient matrix is zero then system (II) has nontrivial solution, hence \( S = \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) is linearly dependent.

For verifying the set \( S = \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) is linearly independent and spanning the vector space \( V \), by following algorithm (modeling).

**Algorithm 2**

**Input:** \( n \) vectors \( \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \)
Step 1: Write these vectors \( v_1, v_2, \ldots, v_n \) as a column of a matrix \( A \).

\[
A = [v_1; v_2; \ldots; v_n].
\]

Step 2: Find \( |A| \).

Step 3: If \( |A| \neq 0 \), then \( S = \{ v_1, v_2, \ldots, v_n \} \) is linearly independent and also span \( V \). Otherwise \( S \) is dependent and not span \( V \).

Step 4: Stop

Program 2

//program to find out linearly independent or dependent set

function det(A)
{
    var Length = A.length-1;
    if (Length == 1) return (A[1][1]);
    else
    {
        var k;
        var sum = 0;
        var factor = 1;
        for (var k = 1; k <= Length; k++)
            sum += factor * A[1][k] * det(a);
    }
if (A[1][k] != 0)
{
    minor = new makeArray2(Length-1,Length-1);
    var m;
    var n;
    var theColumn;

    for (var m = 1; m <= Length-1; m++)
    {
        if (m < k) theColumn = m;
        else theColumn = m+1;

        for (var n = 1; n <= Length-1; n++)
        {
            minor[n][m] = A[n+1][theColumn];
        }
    }

    sum = sum + A[1][k]*factor*det(minor);
}

factor = -factor;
if (sum == 0)
{
    eee = "The Vectors are Linearly Dependent.";
}

if (sum != 0)
{
    eee = "The Vectors are Linearly Independent.";
}

return(sum);

/**************************
3. Basis for Row space

For m x n matrix \( A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \), the vectors \( \mathbf{r}_1 = (a_{11}, a_{12}, \ldots, a_{1n}) \), \( \mathbf{r}_2 = (a_{21}, a_{22}, \ldots, a_{2n}) \), \ldots, \( \mathbf{r}_m = (a_{m1}, a_{m2}, \ldots, a_{mn}) \) formed from the rows of \( A \) are called row vectors of \( A \) and \( \mathbf{c}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} \), \( \mathbf{c}_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} \), \ldots, \( \mathbf{c}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} \).
formed from columns of $A$ are called column vectors of $A$. The subspace of $\mathbb{R}^n$
spanned by the row vectors is called the row space of a matrix $A$ and subspace of $\mathbb{R}^m$
spanned by the column vectors is called the column space of a matrix $A$.

- Elementary row operations do not change the row space of a matrix $A$.
- The non-zero row vectors in row-echelon form of matrix $A$ form a basis for row
  space of $A$.
- The non-zero row vectors in row-echelon form of matrix $A^t$ form a basis for
column space of $A$.

For finding basis for the space spanned by the vectors $v_1, v_2, \ldots, v_n$, first write
matrix $A$ as $v_1, v_2, \ldots, v_n$ are rows of $A$ and then reduce the matrix into row-echelon
form.

**Algorithm 3**

Finding basis for space spanned by the vectors $v_1, v_2, \ldots, v_n$.

**Input:** n vectors $v_1, v_2, \ldots, v_n$

**Step 1:** Write a matrix $A$ has rows are $v_1, v_2, \ldots, v_n$.

$$A = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

**Step 2:** Using step 2 and 3 from algorithm 1 reduce the matrix $A$ into row-echelon form.

The non-zero rows in this matrix (row echelon form) form basis for row space and
number of non-zero rows is the rank of $A$.

**Step 3:** Stop
Program 3

//program to find basis for row space of the matrix

function rowReduce(theMatrix,numRows,numCols) {

    var theNum;
    for (var i = 1; i <= numRows; i++)
    {
    
    var theCol = 0;
    for (var j = 1; j <= numCols; j++)
    {
    
    theNum = theMatrix[i][j];
    if (Math.abs(theNum) < theSmallestNumber) theMatrix[i][j] = 0;
    else {theCol = j; break}
    
    }
    
    if (theCol != 0) {theMatrix = 
    
    pivot(InMatrix,rows,cols,theRow,theCol) {
    
    var thePivot = InMatrix[theRow][theCol];
    for (var i = 1; i <= cols; i++)
    {
    
    InMatrix[theRow][k] = InMatrix[theRow][i]/thePivot;

    
}
```javascript
var theNum = 1;

for (var i = 1; i <= rows; i++)
{
    if ( (i != theRow) && (InMatrix[i][theCol] != 0) )
    {
        var factr = InMatrix[i][theCol];

        for (var j = 1; j <= cols; j++)
        {
            InMatrix[i][j] = InMatrix[i][j] - factr*InMatrix[theRow][j];
        }
    }
}

return(InMatrix);
```
for (i = theRow; i <= numRows; i++)
{
    if( theMatrix[i][j] != 0)
    {
        if (i == theRow) {theRow++; break;}
        // function swaprows
        else { theMatrix = function swapRows(theMatrix, numRows, numCols, p,q)
            var rowHold =0;
            for(var j = 1; j <= numCols; j++)
            {
                rowHold = InMatrix[p][j];
                InMatrix[p][l] = InMatrix[q][j];
                InMatrix[q][j] = rowHold;
            }
            return(InMatrix); theRow++; break}
    }
}
}
Row basis = "";

for (i = 1; i <= rank; i++)
{
    for (j = 1; j <= numCols; j++)
    {
        Row basis = Row basis + theMatrix[i][j] + " ";
    }
    Row basis = Row basis + "\n";
}

return(theMatrix);

//*

4. Basis for Column space

Algorithm 4

For finding the basis for column space of $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$
Step 1: Write these vectors $c_1, c, \ldots, c_n$ as a column of a matrix $A$.

$$A = [c_1 : c_2 : \ldots : c_n].$$

Step 2: Write $A^t$

Step 3: Using step 2 and 3 from algorithm 1 reduce the matrix $A^t$ into row-echelon form.

The non-zero rows in this matrix (row echelon form of $A^t$) form basis for column space of $A$ and number of non-zero rows is the rank of $A$.

Step 4: Stop

Program 4

// program to find Basis for column space of matrix

Function rowReduce(theMatrix, numRows, numCols) {

    function transpose (TheMatrix, numRows, numCols) {

        for (int i = 0; i < numRows; i++) {

            for (int j = i; j < numCols; j++) {

                TheMatrix[i][j] = TheMatrix[j][i];

            }

        }

        var theNum;

        for (var i = 1; i <= numRows; i++)

        \}
{  
  var theCol = 0;

  for (var j = 1; j <= numCols; j++)
    
    
    
    
    theNum = theMatrix[i][j];

  if (Math.abs(theNum) < theSmallestNumber) theMatrix[i][j] = 0;

  else {theCol = j; break}

}

if (theCol != 0) {theMatrix =

pivot(InMatrix,rows,cols,theRow,theCol) {

var thePivot = InMatrix[theRow][theCol];

for (var i = 1; i <= cols; i++)

  
  
  
  
  {  
    InMatrix[theRow][i] = InMatrix[theRow][i]/thePivot;

  }

  

var theNum = 1;

for (var i = 1; i <= rows; i++)

  
  
  
  
  {  
    if ( (i != theRow) && (InMatrix[i][theCol] != 0) )
{  
  var factr = InMatrix[i][theCol];

  for (var j = 1; j <= cols; j++)
  {
    InMatrix[i][j] = InMatrix[i][j] - factr*InMatrix[theRow][j];
  }
}

return(InMatrix);

}
else { theMatrix = swapRows(theMatrix, numRows, numCols, p, q)

var rowHold = 0;

for (var j = 1; j <= numCols; j++)
{

    rowHold = InMatrix[p][j];

    InMatrix[p][j] = InMatrix[q][j];

    InMatrix[q][j] = rowHold;

}

return(InMatrix); theRow++; break}

}

}

}

Column basis = "";

for (i = 1; i <= rank; i++)
{

    for (j = 1; j <= numCols; j++)
    {

        Column basis = Column basis + theMatrix[i][j] + " ";

    }

}
5. Rank

Algorithm 5

Rank of matrix $A$.

**Input:** matrix $A$.

**Step 1:** Using step 2 and step 3 from algorithm 1 reduce the matrix $A$ into row-echelon form.

**Step 2:** The number of non-zero rows in row-echelon form = rank of matrix

**Step 3:** Stop

Program 5.

//program to find rank of the matrix

function rowReduce(theMatrix,numRows,numCols) {

  var theNum;

  for (var i = 1; i <= numRows; i++)
  {
  }
var theCol = 0;

for (var j = 1; j <= numCols; j++)
{
    theNum = theMatrix[i][j];
    if (Math.abs(theNum) < theSmallestNumber) theMatrix[i][j] = 0;
    else {theCol = j; break}
}

if (theCol != 0) {theMatrix =
    pivot(InMatrix,rows,cols,theRow,theCol) {
        var thePivot = InMatrix[theRow][theCol];
        for (var i = 1; i <= cols; i++)
        {
            InMatrix[theRow][i] = InMatrix[theRow][i]/thePivot;
        }
        var theNum = 1;
        for (var i = 1; i <= rows; i++)
        {
            if ( (i != theRow) && (InMatrix[i][theCol] != 0) )
            {
            }
var factr = InMatrix[i][theCol];

for (var j = 1; j <= cols; j++)
{
    InMatrix[i][j] = InMatrix[i][j] - factr*InMatrix[theRow][j];
}

return(InMatrix);

} var theRow = 1;

for (j = 1; j <= numCols; j++)
{
    for (i = theRow; i <= numRows; i++)
    {
        if( theMatrix[i][j] != 0)
        {
            if (i == theRow) {theRow++; break;}

            // function swaprows
```javascript
else { theMatrix = function swapRows(theMatrix, numRows, numCols, p, q) {
  var rowHold = 0;
  for (var j = 1; j <= numCols; j++) {
    rowHold = InMatrix[p][j];
    InMatrix[p][j] = InMatrix[q][j];
    InMatrix[q][j] = rowHold;
  }
  return (InMatrix); theRow++; break}
}

rank = 0;
for (i = 1; i <= numRows; i++) {
  for (j = 1; j <= numCols; j++) {
    if (theMatrix[i][j] != 0)
  }
}
```
{ } 

rank++; break;
}
} 

return(theMatrix);
}

//*/ 

6. Nullity of the matrix 

Algorithm 6 

Nullity of the matrix. 

Input: matrix A. 

Step 1: Using step 2 and step 3 from algorithm 1 reduce the matrix A into row-echelon form. 

Step 2: The number of non-zero rows in row-echelon form = rank of matrix. 

Step 3: Nullity of the matrix = number of column – rank of matrix. 

Step 4: Stop 

Program 6 

//program to find nullity of matrix
function rowReduce(theMatrix,numRows,numCols) {
    var theNum;
    for (var i = 1; i <= numRows; i++)
    {
        var theCol = 0;
        for (var j = 1; j <= numCols; j++)
        {
            theNum = theMatrix[i][j];
            if (Math.abs(theNum) < theSmallestNumber) theMatrix[i][j] = 0;
            else {theCol = j; break}
        }
        if (theCol != 0) {theMatrix =

    function pivot(InMatrix,rows,cols,theRow,theCol) {
        var thePivot = InMatrix[theRow][theCol];
        for (var i = 1; i <= cols; i++)
        {
            InMatrix[theRow][i] = InMatrix[theRow][i]/thePivot;
        }
        var theNum = 1;

    }
for (var i = 1; i <= rows; i++)
{
  if (i != theRow) && (InMatrix[i][theCol] != 0)
  {
    var factr = InMatrix[i][theCol];
    for (var j = 1; j <= cols; j++)
    {
      InMatrix[i][j] = InMatrix[i][j] - factr*InMatrix[theRow][j];
    }
  }
}
return(InMatrix);
if( theMatrix[i][j] != 0)
{
    if (i == theRow) {theRow++; break;)

    // function swaprows

    else { theMatrix = function swapRows(theMatrix, numRows, numCols, p,q)

        var rowHold =0;

        for(var j= 1; j <= numCols; j++)
        {
            rowHold = InMatrix[p][j];

            InMatrix[p][j] = InMatrix[q][j];

            InMatrix[q][j] = rowHold;

        }

        return(InMatrix); theRow++; break}

}

rank = 0;

for (i = 1; i <= numRows; i++)
for (j = 1; j <= numCols; j++)
{
    if( theMatrix[i][j] != 0)
    {
        rank++;
    break;
    }
}

Nullity = numCols - rank

return (theMatrix);

/*

Using above algorithms and programs in Java programming language, we developed following Mathematical model (Basic Linear Algebra Software).
Figure 3.1: Basic Linear Algebra Software

Mathematical model (Basic Linear Algebra Software) is developed by using Java programming language and logical concepts from above algorithms. Main two programs such as reduced matrix into row-echelon form and determinant of the matrix play an important role for developing the Mathematical model (Basic Linear Algebra Software). Apart from these the program function pivot, function swap rows and function transpose are also used for developing the software. Software is divided into three parts say upper part, middle part and lower part. Upper part is about input data and one think always in mind that input data must be in matrix form and we can take matrix of order up-to 26 x 26. Middle part about key buttons and there are six key buttons denoted by System Consistency, Linear Dependence, Rank, Column Basis, Row Basis and Nullity and each of them different work. Lower part of the Software about output screen and it display the result. This software is easily operating by any students who know the definition of matrix. For operating this software, first enter the matrix on input screen and only click
the key button from middle part which result we want. In operation of matrix in middle part matrix operation is automatically taken and stepwise result with fraction of integers display on output screen.

3.5.2 Mathematical software fx-CG20

After development of mathematical model (Basic Linear Algebra Software), we search for freeware mathematical software on internet with related to Linear Algebra topics. Among these software we select a mathematical software fxCG-20 for comparison with self development software (BLAS).

Figure 3.2: fx-CG20
Mathematical software fx-CG20 is used in two ways, first installing this software on computers and then uses it for mathematical purpose and another form of this software is available in calculator form. Calculator form is essaying to handle and essay to carry. After demonstration, every type of student is operating this fx-CG20 software. There is no need of power supply and battery back-up operating this software. It is available in calculator form so only four pencil cells are sufficient for operating this software. Teacher can use in regular classroom teaching and also in other place in urban as well as in rural areas. It is useful to all branches of education as well as many government departments, shear market and various companies.

3.5.2.1 Operations

For starting to use the fx-CG20, first use the key AC/ON and then press MENU to display the main menu, by using cursor keys left, right, upper, and lower on REPLAY key to move icon we need, and then press EXE for display mode of the screen which icon selected. We also display same mode without highlighting an icon by inputting the letter or number in the upper right corner on the icon was given. In each mode many sub modes are given.

3.5.2.2 Meaning of Icons

1. Run-Matrix: “In Run-matrix mode, to press $F_4$ (MATH) display the MATH menu. We can use this for matrix operations up-to 6 x 6 order, derivatives of first and second order at a point, integrations, absolute values, Logarithmic operations and summation of finite series. When we press OPTN (option menu) key then the following modes display:
   - LIST: list of various functions
   - MAT: operation of matrices
   - COMPLEX: calculation of complex numbers
   - CALC: analysis of functions
STAT: menu for statistical functions, mean, mode, median, distribution

CONVERT: conversion of metric

HYPERBL: hyperbolic calculation

PROB: probability/distribution

NUMERIC: numerical calculation

ANGLE: coordinate conversion, and relation

ENG_SYM: engineering-symbols

PICTURE: graph representation

FUNCMEM: function-memory

LOGIC: logical-operator

CAPTURE: about screen

FINANCE: financial calculations

Also, these modes shall display by another operations.

2. Statistics: “Using this mode we can perform mean, mode, Median, standard
deviation, regression analysis, various test (Z test, t test and CHI test etc.) and
statistical diagrams”.

3. E-Activity: “Using this mode we can store text or formulas of mathematics,
important theorems”.

4. Spreadsheet: “Using of this mode us performing spreadsheet calculations. Each
file contains a 999-row x 26-column spreadsheet”.

5. Graph: “Using this menu to draw graphs of the various functions”.

6. Table: “Using of this mode to generate a numerical table of different solutions”.

7. Recursion: “Using this menu bar to store recursion formulas”.

8. Conic Graphs: “Graphs of conic sections can be drawn using this mode”.

9. Equation: “Simulations system of equations up to 6 variables and high-order
differential equations up to 6th degree can be solved using this mode”.

10. Program: “We can store programs and to run programs using this mode”.

11. Financial: “Financial calculations, index price and to draw cash flow and other
type of calculations are made using this mode”.

12. Link: “We can transfer back-up data to another unit or PC using this mode”.

13. Memory: “Data can be stored in memory using memory mode”.
14. **Geometry**: “For drawing and analyzing geometric objects”.

15. **Picture Plot**: “That represents coordinates on the screen and to performing various types of analysis based on the plotted data”

### 3.5.2.3 Basic Functions

Basic calculations on `fx-CG20` are performed viz. addition, subtraction, multiplication, division of numbers, logarithmic function calculations, and exponential functions calculations. Also, we perform trigonometric, inverse trigonometric, hyperbolic, inverse hyperbolic calculations using this software. In addition, we can perform Probability distribution, random number generation in accordance to various distributions. Also, this software helpful to obtain numeric calculations viz. rounds off, GCD, LCM, mod (remainder when n divides by m) and Permutation and combination. Furthermore, we can perform coordinate conversion, sexagesimal operations (conversion of decimal value to sexagesimal value) and logical operators (AND, OR, NOT, XOR) on `fx-CG20` software. Addition of matrices, multiplication, scalar multiplication, determinant, transposition, inversion and square of matrix are perform using `fx-CG20` software. Also calculate solution of System of Linear Equations up to six unknowns and root of higher order equation $2^{nd}$ to $6^{th}$ degree using this software. This `fx-CG20` software is very useful to mathematics teachers and students because on this software some of the applications of mathematical software viz. MATLAB, Octave, Freemat and Scilab are available.

### 3.6 Pilot Study

In our research a pilot study is conducted for testing the validity of tools, to remove ambiguities, to enhance clarity and study student’s reaction. While selecting of the college for pilot study criterion is availability of well equipped computer laboratory with mathematical software and co-operation given by college authorities. Hence pilot study is carried out in Satana College This College is not considering as a sample in our present research. The pilot study of teaching programme is administered with the objectives given as follows:

1. To get an idea of the administration of mathematical software / model.
2. To check whether the reading materials and other information about research study are sufficient and useful.
3. To check whether the teaching programme is easy to understand for students.
4. To observe the reaction of students regarding the teaching programme.
5. To get idea of appropriate time duration required to complete the whole experiment.

### 3.6.1 Procedure of pilot study

For the pilot study, we select a representative sample is selected from the population viz., the students of S. Y. B. Sc. of Satana College. The strength of S. Y. B. Sc. class (those students who offered mathematic as one of the subject at second year) is 45. The sample of pilot study consists of twenty students. These twenty students are divided into two groups. One group have been chosen for teaching programme with use of mathematical model (Basic Linear Algebra Software) and the other group is exposed to traditional teaching of the content. For the pilot study we prepared two topics, system of linear equations and linear independence. During the pilot study, we made following observations.

1. Mathematical model (Basic Linear Algebra Software) / mathematical software fx-CG20 and study materials are easy to understand.
2. The teaching programme with using mathematical model (Basic Linear Algebra Software) shall be taught to any type of students, i.e. slow learner, average or brilliant students without any difficulty.
3. The students show interest in each topic.
4. Most of the students are able to think and derive conclusion.
5. In the beginning, students took more time to complete teaching programme. After some days they are able to complete teaching programme in proper time.
6. Creative and cooperative atmosphere is found while the teaching programmes duration.
Now, taking into consideration the observation during the pilot study, the necessary modification are made in the teaching programme using mathematical model (BLAS). Again it is revised, and we use it in our main study. After completion of the teaching programme, the investigator had conducted post test to both the groups. The collected data is analyzed.

### 3.7 Summary

In this chapter the objectives and importance of mathematical software and development of mathematical model have been given.

The plan of our research study is as follows:

1. Selection of topics for teaching programme.
2. Development of mathematical model (BLAS).
3. Selection of proper mathematical software (fx-CG20).
4. Pilot study is given.
5. Format of mathematical software and mathematical model is supplied, and finally, we discussed how to operate mathematical software fx-CG20, the use of mathematical software and development of mathematical model.

In the next chapter, we made testing of mathematical software and mathematical model.