Maple

Ashan J

2024-07-05

Contents

Chapter 1

Matrices

There are several ways to define a matrix in Maple. Use the help command to find the general definition of a matrix. To work with Matrices first load the package "***linalg***" Which include most linear algebra related commands.

1.1 Defining matrices

There are several ways to define Matrix. Let's look at them through examples.

You can use maple help.

```
> ? matrix
```

```
> restart;
> with(linalg):
> M1 :=matrix (3, 3, [1,-2,-3,2,-2,2,3,-3,3]);
```

```
> restart;
> with(linalg):
> M1 :=matrix (3, 3, [1,-2,-3,2,-2,2,3,-3,3]);
```

$$M1 := \begin{bmatrix} 1 & -2 & -3 \\ 2 & -2 & 2 \\ 3 & -3 & 3 \end{bmatrix}$$

```
>
```

```
> M2:= matrix(3,4,[1,1/2,-1/3,1/4,2,1/2,-3/4,5/4,3,3/5,3/7,3/8]);
```

```
> M2:= matrix(3,4,[1,1/2,-1/3,1/4,2,1/2,-3/4,5/4,3,3/5,3/7,3/8]);
```

$$M2 := \begin{bmatrix} 1 & \frac{1}{2} & \frac{-1}{3} & \frac{1}{4} \\ 2 & \frac{1}{2} & \frac{-3}{4} & \frac{5}{4} \\ 3 & \frac{3}{5} & \frac{3}{7} & \frac{3}{8} \end{bmatrix}$$

```
>
```

```
[> A:= matrix(3,3,[[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

```
> A:= matrix(3,3,[[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

$$A := \begin{bmatrix} 1 & -3 & -2 \\ 2 & -3 & 5 \\ -4 & 2 & 6 \end{bmatrix}$$

```
[> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

```
> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

$$B := \begin{bmatrix} 1 & -3 & -2 \\ 2 & -3 & 5 \\ -4 & 2 & 6 \end{bmatrix}$$

```
>
```

```
[> Matrix (2,2,fill=a);
```

```
> Matrix (2,2,fill=a);
```

Uppercase

$$\begin{bmatrix} a & a \\ a & a \end{bmatrix}$$

```
> |
```

```
[> Matrix (2,2,symbol=a);
```

```
> Matrix (2,2,symbol=a);
```

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$$

```
>
```

```
[> f:=i->x*i-1;
[> M:=Matrix(2,f);
```

```
> f:=i->x*i-1;
```

$$f := i \to x\,i - 1$$

```
> M:=Matrix(2,f);
```

$$M := \begin{bmatrix} x-1 & x-1 \\ 2x-1 & 2x-1 \end{bmatrix}$$

```
>
```

```
[> g:=(i,j)->x*(i+j-1);
[> M1:=Matrix(3,g);
```

```
> g:=(i,j)->x*(i+j-1);
```

$$g := (i,j) \to x\,(i+j-1)$$

```
> M1:=Matrix(3,g);
```

$$M1 := \begin{bmatrix} x & 2x & 3x \\ 2x & 3x & 4x \\ 3x & 4x & 5x \end{bmatrix}$$

```
> |
```

To define a diagonal matrix.

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

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

$$C := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

```
[> diag(1,1,1);
```

```
> diag(1,1,1);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To define lower triangular matrix.

```
[> M_1:=Matrix(3,[[x],[y,y],[z,z,z]],shape=triangular[lower]);
```

```
> M_1:=Matrix(3,[[x],[y,y],[z,z,z]],shape=triangular[lower]);
```

$$M\_1 := \begin{bmatrix} x & 0 & 0 \\ y & y & 0 \\ z & z & z \end{bmatrix}$$

```
[> M_2:=Matrix(3,[[1,2,3],[4,5,6],[7,8,9]],shape=triangular[upper]);
```

```
> M_2:=Matrix(3,[[1,2,3],[4,5,6],[7,8,9]],shape=triangular[upper]);
```

$$M\_2 := \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \\ 0 & 0 & 9 \end{bmatrix}$$

To define zero matrix

```
[> M_3:=Matrix(4,4,shape=zero);
```

```
> M_3:=Matrix(4,4,shape=zero);
```

$$M\_3 := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

To define identity matrix

```
[> M_3:=Matrix(3,3,shape=identity);
```

```
> M_3:=Matrix(3,3,shape=identity);
```

$$M\_3 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

To find an entry of a matrix, follow the name of the matrix by indices inside a square bracket.

```
[> A:= matrix(3,3,[[1,-3,-2],[2,-3,5],[-4,2,6]]);
[> A[3,2];
```

```
> A:= matrix(3,3,[[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

$$A := \begin{bmatrix} 1 & -3 & -2 \\ 2 & -3 & 5 \\ -4 & 2 & 6 \end{bmatrix}$$

```
> A[3,2];
```

$$2$$

```
[> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
[> B[1,2];
```

```
> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

$$B := \begin{bmatrix} 1 & -3 & -2 \\ 2 & -3 & 5 \\ -4 & 2 & 6 \end{bmatrix}$$

```
> B[1,2];
```

$$-3$$

Matrix Operation

1.1.1 Addition and Scalar Multiplication

Algebric expressions with matrices are evaluated by the command `evalm`. scalar multiplication of matrices done by the usual symbol `*`.

```
[> evalm(A+B);
```

```
> evalm(A+B);
```

$$\begin{bmatrix} 2 & -6 & -4 \\ 4 & -6 & 10 \\ -8 & 4 & 12 \end{bmatrix}$$

```
[> evalm(2*A+3*B);
```

```
> evalm(2*A+3*B);
```

$$\begin{bmatrix} 5 & -15 & -10 \\ 10 & -15 & 25 \\ -20 & 10 & 30 \end{bmatrix}$$

1.1.2 Matrix Multiplication

For matrix multiplication the symbol `&*` is used.

```
[> evalm(A&*B);
```

```
> evalm(A&*B);
```

$$\begin{bmatrix} 3 & 2 & -29 \\ -24 & 13 & 11 \\ -24 & 18 & 54 \end{bmatrix}$$

1.1.3 Transpose

To get the transpose of a matrix the command `transpose` is used.

```
[> transpose(A);
```

```
> transpose(A);
```

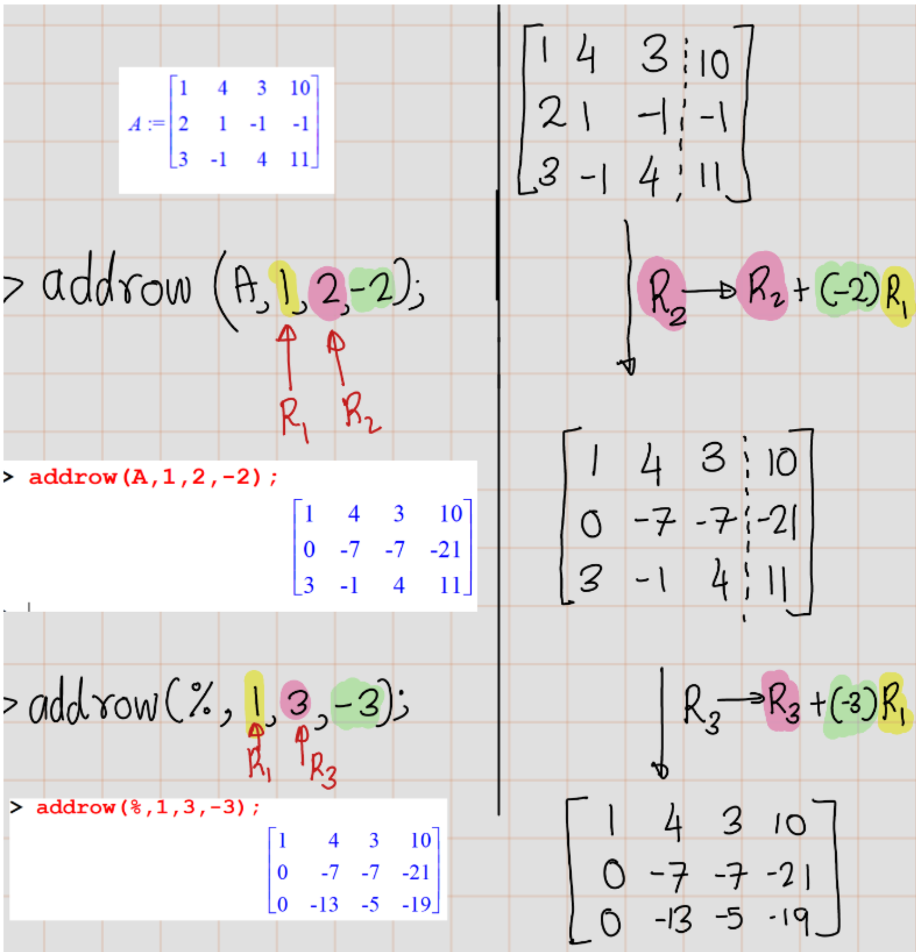
$$\begin{bmatrix} 1 & 2 & -4 \\ -3 & -3 & 2 \\ -2 & 5 & 6 \end{bmatrix}$$

1.2 Row Opreations

In your linear algebra class you will learn elementary row operations on matrices here we will use Maple to do the same thing. Let's define a new matrix A.

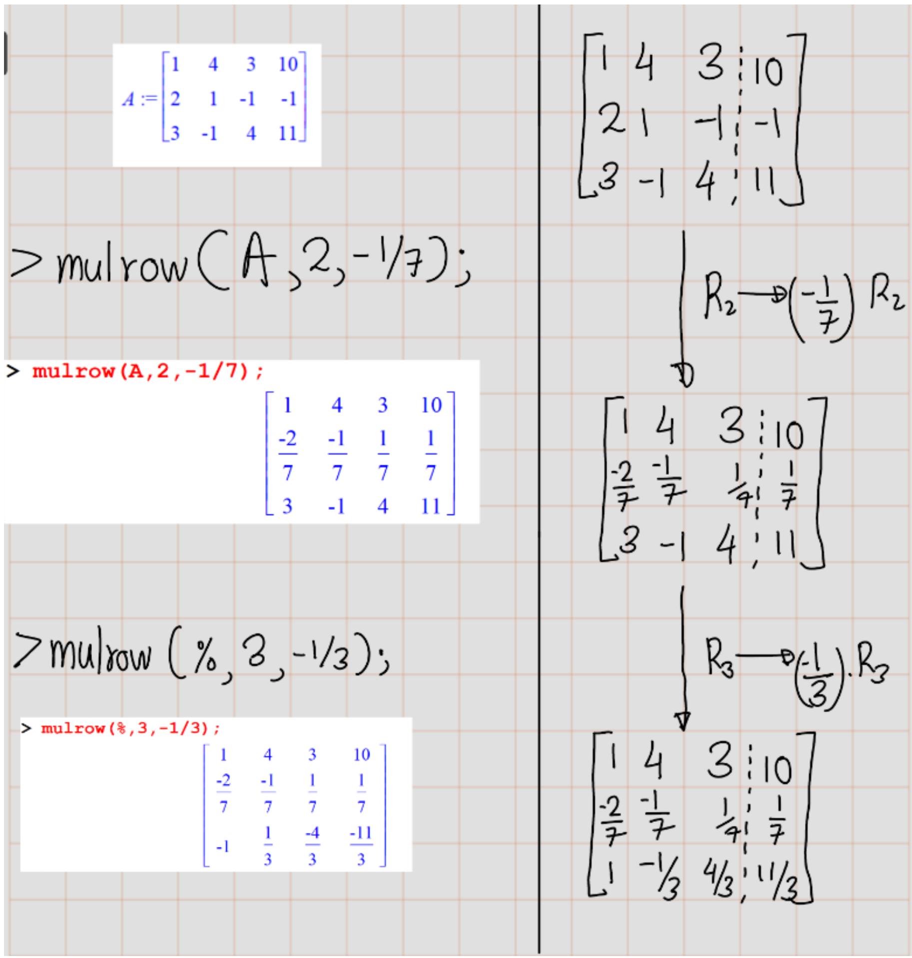
- **addrow(A,r1,r2,m)**
  Returns a copy of a matrix A in which row r2 replaced with by row(A,r2)+m*row(A,r1)

```
[> A:=matrix([[1,4,3,10],[2,1,-1,-1],[3,-1,4,11
[> addrow(A,1,2,-2);
[> addrow(%,1,3,-3);
```



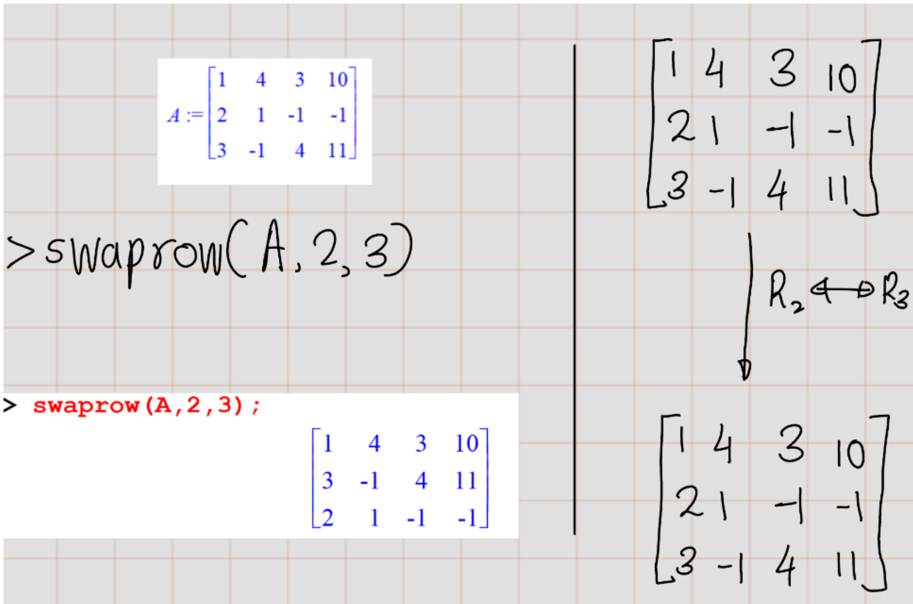
- **mulrow(A,row,expr)**
  Returns a matrix A in which has the same entries as A with the r^{th} row mutiplied by *expr*

```
[> mulrow(A,2,-1/7);
[> mulrow(%,3,-1/3);
```



- **swaprow(A,r1,r2)**
  This command interchange row r1 and r2 of A

```
[> swaprow(A,2,3);
```



Similarly, you can learn `addcol`, `mulcol`,`swapcol` commands by your self.

1.3 Determinanat of Matrix

To find the determinate of a matrix, maple has a special command `det`.

```
[> M1 :=matrix (3, 3, [1,-2,-3,2,-2,2,3,-3,3]);
[> det(M1);
```

```
> M1 :=matrix (3, 3, [1,-2,-3,2,-2,2,3,-3,3]);
```

$$M1 := \begin{bmatrix} 1 & -2 & -3 \\ 2 & -2 & 2 \\ 3 & -3 & 3 \end{bmatrix}$$

```
> det(M1);
```

$$0$$

```
[> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
[> det(B);
```

```
> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

$$B := \begin{bmatrix} 1 & -3 & -2 \\ 2 & -3 & 5 \\ -4 & 2 & 6 \end{bmatrix}$$

```
> det(B);
```

$$84$$

1.3.1 Inverse of a matrix

The `inverse` command is used to find the inverse of a square matrix, if exists.

```
[> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
[> inverse(B);
```

```
> B:=matrix([[1,-3,-2],[2,-3,5],[-4,2,6]]);
```

$$B := \begin{bmatrix} 1 & -3 & -2 \\ 2 & -3 & 5 \\ -4 & 2 & 6 \end{bmatrix}$$

```
> inverse(B);
```

$$\begin{bmatrix} \frac{-1}{3} & \frac{1}{6} & \frac{-1}{4} \\ \frac{-8}{21} & \frac{-1}{42} & \frac{-3}{28} \\ \frac{-2}{21} & \frac{5}{42} & \frac{1}{28} \end{bmatrix}$$

```
>
```

```
[> inverse([[2,-1],[3,2]]);
```

```
> inverse([[2,-1],[3,2]]);
```

$$\begin{bmatrix} \frac{2}{7} & \frac{1}{7} \\ \frac{-3}{7} & \frac{2}{7} \end{bmatrix}$$

```
>
```

1.4 Exercise

Exercise 1.1. Performs the indicated computations.

A. $\begin{bmatrix} 1 & 0 & 3 \\ 2 & -1 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 4 \\ -2 & 5 & 8 \end{bmatrix}$

B. $5\begin{bmatrix} 2 & 1 & 3 \\ -1 & 2 & 4 \\ -6 & 1 & 5 \end{bmatrix} - 3\begin{bmatrix} -2 & 1 & 4 \\ 5 & 0 & 7 \\ 2 & -1 & 3 \end{bmatrix}$

C. $\begin{bmatrix} 2 & 3 \\ -1 & 4 \end{bmatrix}\begin{bmatrix} 5 & -1 \\ 2 & 7 \end{bmatrix}$

D. $\begin{bmatrix} 2 & 3 & 1 & 5 \\ 0 & 6 & 2 & 4 \end{bmatrix}\begin{bmatrix} 5 & 7 & 1 \\ 2 & 0 & 3 \\ 1 & 0 & 0 \\ 0 & 5 & 6 \end{bmatrix}$

E. $\begin{bmatrix} 2 & 4 & 3 \\ 1 & 3 & 5 \end{bmatrix}\begin{bmatrix} 0 & 7 & 4 \\ 2 & 3 & 0 \end{bmatrix}$

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

G. $\begin{bmatrix} 1 & 0 & 3 & -1 & 5 \\ 2 & 1 & 6 & 2 & 5 \end{bmatrix}\begin{bmatrix} 7 & 1 \\ 2 & 3 \\ -1 & 0 \\ 5 & 6 \\ 2 & 3 \end{bmatrix}$

H. $\begin{bmatrix} 1 & -1 & 2 \\ 3 & 5 & 6 \\ 2 & 4 & -1 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$

Exercise 1.2. Determine the given matrices are invertible. If they are, compute the inverse.

A. $\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$

B. $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 5 & 5 & 1 \end{bmatrix}$

C. $\begin{bmatrix} a & a \\ b & b \end{bmatrix}$

D. $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

E. $\begin{bmatrix} 5 & 7 & 0 \\ 2 & 1 & 3 \\ 1 & 0 & 0 \\ 0 & 4 & 3 \end{bmatrix}$

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

G. $\begin{bmatrix} 1 & -3 & 0 & 2 \\ 3 & -12 & -2 & -6 \\ -2 & 10 & 2 & 5 \\ -1 & 6 & 1 & 3 \end{bmatrix}$

Exercise 1.3. Compute the 3×3 matrix whåse entries are given by the function y^{ij}, Where $i = j = 1, 2, 3$.

Exercise 1.4. Compute the matrix, $\begin{bmatrix} x & x^2 \\ x^2 & x^3 \end{bmatrix}$ by defining a suitable function as in the exercise 1.4

Exercise 1.5. Which of the following matrices are skew-symmetric? (A square matrix a symmetric if $A^T = -A$, where A^T is the transpose of A.)

A. $\begin{bmatrix} 1 & -6 \\ 6 & 0 \end{bmatrix}$

B. $\begin{bmatrix} 0 & -6 \\ 6 & 0 \end{bmatrix}$

C. $\begin{bmatrix} 2 & -2 & 2 \\ 2 & 2 & -2 \\ 2 & 2 & 2 \end{bmatrix}$

D. $\begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 2 \\ 1 & -2 & 0 \end{bmatrix}$

Exercise 1.6. Convert the following matrix in to an upper triangular matrix.

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

Exercise 1.7. Convert the following matrices into a identity matrix.

A. $\begin{bmatrix} 2 & 7 & 3 \\ 1 & 3 & 2 \\ 3 & 7 & 9 \end{bmatrix}$

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

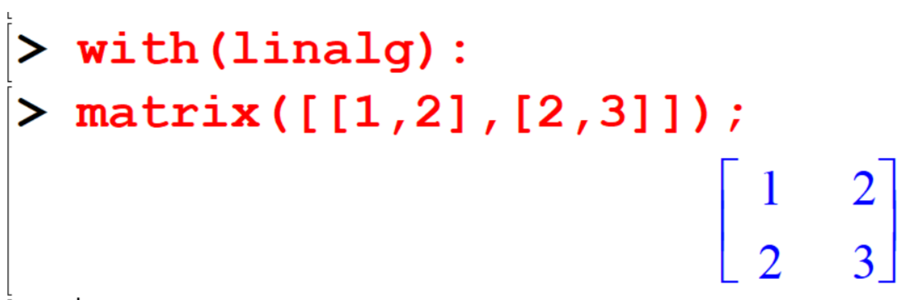
Chapter 2

System of linear equations and matrices

A linear system is a collection of first degree equations. A solution to a system consists of one or more sets of specific values that our common solutions to each of the individual equations. Here is a simple example which we can solve quite easily using the solve command. ## Augment of a matrix

```
[> with(linalg):
[> A:=matrix([[1,2],[2,3]]);
```

Output:




```
[> B:=matrix([[3,4,5],[6,7,8]]);
```

Output:

```
> B:=matrix([[3,4,5],[6,7,8]]);
```

$$B := \begin{bmatrix} 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

```
augment(A,B);
```

Output:

```
> augment(A,B);
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 6 & 7 & 8 \end{bmatrix}$$

The function augment joins two or more matrices together horizontally. The matrices and vectors must have the same number of rows.

2.1 Solving Linear Systems

$$\begin{array}{rcrcrcr} 2x & + & 5y & - & 4z & = & 9 \\ 3x & + & 5y & + & 2z & = & 12 \\ 4x & - & y & + & 5z & = & -3 \end{array}$$

$$\underbrace{\begin{bmatrix} 2 & 5 & -4 \\ 3 & 5 & 2 \\ 4 & -1 & 5 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 9 \\ 12 \\ -3 \end{bmatrix}}_{B}$$

2.1.1 Method I

A linear system is a collection of first degree equations. A solution to a system consists of one or more set of specific values that our common solution to each of the individual equations. Here is a simple example which we can solve quite easily using the solve command.

```
[> sys:={2*x+5*y-4*z=9,3*x+5*y+2*z=12,4*x-y+5*z=-3};
[> solve(sys,{x,y,z});
```

Output:

```
> sys:={2*x+5*y-4*z=9,3*x+5*y+2*z=12,4*x-y+5*z=-3};
```

$$sys := \{2\,x + 5\,y - 4\,z = 9,\ 3\,x + 5\,y + 2\,z = 12,\ 4\,x - y + 5\,z = -3\}$$

```
> solve(sys,{x,y,z});
```

$$\{x = \frac{-33}{37},\ y = \frac{99}{37},\ z = \frac{24}{37}\}$$

Maple will automatically use fractions, however you can force decimal answers using the ***evalf*** command.

```
[> evalf(%);
```

Output:

```
> evalf(%);
```

$$\{x = -0.8918918919,\ y = 2.675675676,\ z = 0.6486486486\}$$

2.1.2 Method II

We can also convert the above system of equations to a matrix system.

```
[> A:=genmatrix(sys,[x,y,z],b);
[> evalm(b);
```

Output:

```
> A:=genmatrix(sys,[x,y,z],b);
```

$$A := \begin{bmatrix} 2 & 5 & -4 \\ 3 & 5 & 2 \\ 4 & -1 & 5 \end{bmatrix}$$

```
> evalm(b);
```

$$[9,\ 12,\ -3]$$

In this case ,

- A is the coefficient matrix, and

- b is vector representing the constant values.

The command `evalm(b)` evaluated b as a matrix (a vector is a $n \times 1$ matrix). In other words, this command simply writes out what the vector b looks like.

```
[> linsolve(A,b);
```

Output:

```
> linsolve(A,b);
```

$$\left[\frac{-33}{37}, \frac{99}{37}, \frac{24}{37}\right]$$

2.1.3 Method III

Another way to solve a matrix equation $Ax = b$ is to left multiply both sides by the inverse matrix A^{-1}, if it exists, to get the solution $x = A^{-1}b$.

```
[>  inverse(A);
[> evalm(inverse(A)&*b);
```

Output:

```
> inverse(A);
```

$$\begin{bmatrix} \frac{9}{37} & \frac{-7}{37} & \frac{10}{37} \\ \frac{-7}{111} & \frac{26}{111} & \frac{-16}{111} \\ \frac{-23}{111} & \frac{22}{111} & \frac{-5}{111} \end{bmatrix}$$

```
> evalm(inverse(A)&*b);
```

$$\left[\frac{-33}{37}, \frac{99}{37}, \frac{24}{37}\right]$$

The `evalm` command forces a matrix computation and express the result.

2.2 Dependent Systems:

Some systems are dependent which means there are an infinite number of solutions. The form of these solutions will entail the use of parameters. Let's look an example using two different methods; `solve` and `linsolve`.

Example 2.1.

$$\begin{array}{rcrcrcrr} x & - & 2y & + & z & = & & 3 \\ x & + & y & - & 2z & = & - & 4 \\ 2x & - & y & - & z & = & - & 1 \end{array}$$

```
[> restart;
[> with(linalg):
[> sys:={x-2*y+z=3,x+y-2*z=-4,2*x-y-z=-1};
[> solve(sys,{x,y,z});
```

```
> restart;
> with(linalg):
> sys:={x-2*y+z=3,x+y-2*z=-4,2*x-y-z=-1};
```

$$sys := \{x - 2\,y + z = 3, x + y - 2\,z = \text{-}4, 2\,x - y - z = \text{-}1\}$$

```
> solve(sys,{x,y,z});
```

output:

$$\{x = -\frac{5}{3} + z, y = -\frac{7}{3} + z, z = z\}$$

```
[> A:=genmatrix(sys,[x,y,z],b);
[> evalm(b);
[> linsolve(A,b);
```

output:

```
> A:=genmatrix(sys,[x,y,z],b);
```

$$A := \begin{bmatrix} 1 & \text{-}2 & 1 \\ 1 & 1 & \text{-}2 \\ 2 & \text{-}1 & \text{-}1 \end{bmatrix}$$

```
> evalm(b);
```

$$[3, \text{-}4, \text{-}1]$$

```
> linsolve(A,b);
```

$$\left[-\frac{5}{3} + \_t_1, -\frac{7}{3} + \_t_1, \_t_1\right]$$

In both cases, the solution contains a parameter. The `solve` command express it in terms actual variable used,and the `linsolve` command use the funny t character to distinguish it from a variable you might have defined yourself.

2.3 Inconsistent System

An Inconsistent system has no solution. Maple generally, refers to answer questions which have no answer.

Example 2.2.

$$\begin{array}{ccccccc} x & + & y & - & 3z & = 10 \\ x & + & y & - & z & = 1 \\ x & + & y & + & z & = 8 \end{array}$$

Let's try to solve this system.

```
[> restart;
[> with(linalg):
[> sys:={x+y-3*z=10,x+y-z=1,x+y+z=8};
[> solve(sys,{x,y,z});
```

Output:

```
> restart;
> with(linalg):
> sys:={x+y-3*z=10,x+y-z=1,x+y+z=8};
```

$$sys := \{x + y - 3\,z = 10, x + y - z = 1, x + y + z = 8\}$$

```
> solve(sys,{x,y,z});
> |
```

This method does not work. It does not return anything.

Let's try `linsolve` method.

```
[> A:=genmatrix(sys,[x,y,z],b);
[> linsolve(A,b);
```

Output:

```
> A:=genmatrix(sys,[x,y,z],b);
```

$$A := \begin{bmatrix} 1 & 1 & -3 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

```
> linsolve(A,b);
>
>
```

This method also does not works.

```
[> evalm(inverse(A)&*b);
```

Output:

```
> evalm(inverse(A) &*b);
Error, (in linalg:-inverse) singular matrix
>
```

2.3.0.1 Automatic Reduction

```
[> sys:={3*x+5*y+2*z=12,2*x+5*y-4*z=9,4*x-y+5*z=-3};
[> evalm(b);
[> C:=augment(A,b);
```

Output:

```
> sys:={3*x+5*y+2*z=12,2*x+5*y-4*z=9,4*x-y+5*z=-3};
```

$$sys := \{2x + 5y - 4z = 9, 3x + 5y + 2z = 12, 4x - y + 5z = -3\}$$

```
> evalm(b);
```

$$[10, 1, 8]$$

```
> C:=augment(A,b);
```

$$C := \begin{bmatrix} 1 & 1 & -3 & 10 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 8 \end{bmatrix}$$

2.3.1 Method IV

An even faster method is simplifying let maple to do all the work for us. The `gausselim` command will perform all of the steps of Gaussian eliminations and reduce an augmented matrix to **row echelon form**.

```
[> sys:={3*x+5*y-2*z=12,2*x+5*y-4*z=9,4*x-y+5*z=-3};
[> A:=genmatrix(sys,[x,y,z],b);
[> evalm(b);
[> C:=augment(A,b);
[> gausselim(C);
```

```
> sys:={3*x+5*y-2*z=12,2*x+5*y-4*z=9,4*x-y+5*z=-3};
```

$$sys := \{2\,x+5\,y-4\,z=9,\ 3\,x+5\,y-2\,z=12,\ 4\,x-y+5\,z=-3\}$$

```
> A:=genmatrix(sys,[x,y,z],b);
```

$$A := \begin{bmatrix} 2 & 5 & -4 \\ 3 & 5 & -2 \\ 4 & -1 & 5 \end{bmatrix}$$

```
> evalm(b);
```

$$[9, 12, -3]$$

```
> C:=augment(A,b);
```

$$C := \begin{bmatrix} 2 & 5 & -4 & 9 \\ 3 & 5 & -2 & 12 \\ 4 & -1 & 5 & -3 \end{bmatrix}$$

```
> gausselim(C);
```

$$\begin{bmatrix} 2 & 5 & -4 & 9 \\ 0 & -11 & 13 & -21 \\ 0 & 0 & \frac{23}{22} & \frac{36}{11} \end{bmatrix}$$

```
> |
```

2.3.2 Method V

The command `gaussjord` does the same thing. It performs all the steps of Gauss Jordan elimination and reduces and reduces an augmented matrix into reduced row echelon form.

```
[> gaussjord(C);
```

```
> gaussjord(C);
```

$$\begin{bmatrix} 1 & 0 & 0 & \frac{-75}{23} \\ 0 & 1 & 0 & \frac{129}{23} \\ 0 & 0 & 1 & \frac{72}{23} \end{bmatrix}$$

```
>
```

2.4 Exercise

Exercise 2.1. Solve the following system of equations.

$$\begin{array}{rcrcrl} x_1 & + & 2x_2 & + & 3x_3 & = 9 \\ 2x_1 & - & x_2 & + & x_3 & = 8 \\ 3x_1 & & & - & x_3 & = 3 \end{array}$$

Exercise 2.2. A small manufacturing plant makes three types of inflatable boats: one-person, two-person, and four-person models. Each boat requires the services of three departments, as listed in the table. The cutting, assembly, and packaging departments have available a maximum of 380, 330, and 120 labor hours per week, respectively.

Department	One Person Boat	Two-Person Boat	Four-Person Boat
Cutting	0.5 hr	1.0 hr	1.5 hr
Assembling	0.6 hr	0.9 hr	1.2 hr
Packaging	0.2 hr	0.3 hr	0.5 hr

(A) How many boats of each type must be produced each week for the plant to operate at full capacity?

(B) How is the production schedule in part A affected if the packaging department is no longer used?

(C) How is the production schedule in part A affected if the four-person boat is no longer produced?