INDEX

CHAPTER – 01

INTRODUCTION AND HISTORICAL BACKGROUND:

The knowledge of matrices is necessary in various branches of mathematics. Matrices are one of the most powerful tools in mathematics. This mathematical tool simplifies our work to a great extent when compared with other straight forward methods. The evolution of concept of matrices is the result of an attempt to obtain compact and simple methods of solving system of linear equations. Matrices are not only used as a representation of the coefficients in system of linear equations, but utility of matrices far exceeds that use. Matrix notation and operations are used in electronic spread sheet programs for personal computer, which in turn is used in different areas of business and science like budgeting, sales projection, cost estimation, analysing the results of an experiment etc. Also, many physical operations such as magnification, rotation and reflection through a plane can be represented mathematically by matrices. Matrices are also used in cryptography. This mathematical tool is not only used in certain branches of sciences, but also in genetics, economics, sociology, modern psychology, industrial management and graph theory etc.

Historical background:

Matrices have a long history of application in solving linear equations but they were known as arrays until the 1800s. The Chinese text The Nine Chapters on the Mathematical Art written in 10th–2nd century BCE is the first example of the use of array methods to solve simultaneous equations,[102] including the concept of determinants.

Between 1700 and 1710 Gottfried Wilhelm Leibniz publicized the use of arrays for recording information or solutions and experimented with over 50 different systems of arrays. Cramer presented his rule in 1750.

The term "matrix" (Latin for "womb", derived from mater—mother) was coined by James Joseph Sylvester in 1850, who understood a matrix as an object giving rise to a number of determinants today called minors, that is to say, determinants of smaller matrices that derive from the original one by removing columns and rows. In an 1851 paper, Sylvester explains:

I have in previous papers defined a "Matrix as a rectangular array of terms, out of which different systems of determinants may be engendered as from the womb of a common parent‖.

Subsequent researchers shifted their focus from the determinant theory to the more general theory to matrices.

In mathematics the purpose of a transformation such as

$x_1=a_{11}y_1+a_{12}y_2$ and $x_2=a_{21}y_1+a_{22}y_2$ ....(1)

is to reduce an unsolved problem in variables x_1 and x_2 to a more simpler one than the original .what CAYLEY did in 1858 is as given below .in addition to the transformation (1),he considered another transformation

$y_1=b_{11}z_1+b_{12}z_2$ and $y_2=b_{21}z_1+b_{22}z_2$ ....(2)

And consequently (1) becomes as

$x_1 =a_{11}(b_{11}z_1+b_{12}z_2)+a_{12}(b_{21}z_1+b_{22}z_2)$

Therefore $x_1= (a_{11}b_{11}+a_{12}b_{21}) z_1+ (a_{11}b_{12}+a_{12}b_{22}) z_2$

Similarly, $x_2=a_{21} (b_{11}z_1+b_{12}z_2) +a_{22} (b_{21}z_1+b_{22}z_2)$

Therefore $x_2= (a_{21}b_{11}+a_{22}b_{21}) z_1+ (a_{21}b_{12}+a_{22}b_{22}) z_2$..(3)

The transformation from variables x_1 ,x_2 to variables z_1 , z_2 through y_1 , y_2 as is evident from (3) is not easy to recollect.

If, however , the coefficients in the transformations shown in equations (1) and (2) are separated and arranged schematically in terms of arrays, or matrices as Cayley called them, then everything becomes straight forward. Thus, if

A= $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, B=$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$..(4)

Then , AB $= \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21} & a_{11}b_{12}+a_{12}b_{22} \\ a_{21}b_{11}+a_{22}b_{21} & a_{21}b_{12}+a_{22}b_{22} \end{bmatrix}$......................(5)

In which the element $a_{11}b_{11} + a_{12}b_{21}$ is constituted by the first elements of A and the first column elements of B etc., and are very easy to remember.

This discovery of matrices illustrated the power and suggestiveness of a well-devised notation and paved the way for a vast and indispensible branch of mathematics with a multitude of practical applications.

BASIC PROPERTIES OF MATRICES

Definition:

A set of mn numbers either real or complex arranged in the form of rectangular array in which there are ‗m‘ rows and ‗n‘ columns ,rectangular arrangement is called a matrix of order m×n which is denoted by $[a_{\text{ij}}]_{m\times n}$

where i=1,2,3,……. J=1,2,3,………

m-represents number of rows

and n-represents number of columns

and thus the matrix of order m×n is usually written as

$$[a_{\text{ij}}]_{m\times n}=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}=(a_{ij})\in R\ m\times n.$$

We denote matrices by capital letters and its elements by small letters.

The following are some examples of matrices:

$$A=\begin{pmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 5 & 7 & 9 \end{pmatrix}\quad , \quad A=\begin{pmatrix} 3 & -1 & 0 \\ \frac{3}{2} & \frac{\sqrt{3}}{2} & 1 \\ 3 & 3 & -1 \end{pmatrix}\quad , \quad A=\begin{pmatrix} -2 & 0 & 0 \\ 3 & 5 & 0 \\ -1 & 6 & 9 \end{pmatrix}$$

In the above examples , the horizontal lines of elements are said to constitute rows of the matrix and the vertical lines of elements are said to constitute columns of the matrix. thus above matrices has three rows and three columns.

Order Of Matrix:

A matrix having m rows and n columns is called a matrix of order m×n or simply m×n matrix (read as m by n matrix). So referring to the above examples of matrices, we has A as 3×3 matrix. We observe that 3×3=9 elements.

In general, an m×n matrix has the following rectangular array :

$$A=[a_{\text{ij}}]_{m\times n}=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

]=(a_{ij})∈R m×n, 1≤ i ≤ m , 1 ≤ j ≤n , for all   i,j∈

a_{ij} is an element lying in the i^{th} row and j^{th} column .we can also call it as $(i,j)^{th}$ elements of A .the number of elements in an m×n matrix will be equal to mn.

Types of Matrices:

Matrices are distinguished on the basis of their order, elements and certain other conditions. There are different types of matrices but the most commonly used are discussed below. Let‘s find out the types of matrices in the field of mathematics.

Types of Matrices

Different types of Matrices and their forms are used for solving numerous problems. Some of them are as follows:

Row Matrix

A row matrix has only one row but any number of columns. A matrix is said to be a row matrix if it has only one row.

For example,

$$A=[-1/2 \qquad \sqrt{5}/2 \qquad 3]$$

is a row matrix of order 1 × 4. In general, A = $\left. a_{ij} \right]_{1\times n}$ is a row matrix of order 1×n.

Column Matrix

A column matrix has only one column but any number of rows. A matrix is said to be a column matrix if it has only one column.

For example,

$$A=\begin{pmatrix} 0 \\ \sqrt{3} \\ -1 \end{pmatrix}$$

is a column matrix of order 4 × 1. In general, A = $\left[b_{ij} \right]_{m\times 1}$ is a column matrix of order m × 1.

Square Matrix

A square matrix has the number of columns equal to the number of rows. A matrix in which the number of rows is equal to the number of columns is said to be a square matrix. Thus an m × n matrix is said to be a square matrix if m = n and is known as a square matrix of order ‗n‘.

For example,

$A=\begin{pmatrix} 3 & -1 & 0 \\ \frac{3}{2} & \frac{3\sqrt{}}{2} & 1 \\ 3 & 3 & -1 \end{pmatrix}$ is a square matrix of order 3. In general, A = $\left[a_{ij}\right]_{m\times m}$ is a square matrix of order m.

Rectangular Matrix

A matrix is said to be a rectangular matrix if the number of rows is not equal to the number of columns.

For example,

$$A=\begin{pmatrix} 3 & -1 & 0 \\ \frac{3}{2} & \frac{\sqrt{3}}{2} & 1 \\ 4 & 3 & -1 \\ \frac{7}{2} & 2 & -5 \end{pmatrix}$$

is a matrix of the order 4 × 3

Diagonal matrix

A square matrix A = [b$_{ij}$] m × m is said to be a diagonal matrix if all its non-diagonal elements are zero, that is a matrix A =[b$_{ij}$]m×m is said to be a diagonal matrix if bij = 0, when i ≠ j.

For example, $A=[4]$, $A=\begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix}$, $A=\begin{bmatrix} 3 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$

are diagonal matrices of order 1, 2, 3, respectively.

Scalar Matrix

A diagonal matrix is said to be a scalar matrix if all the elements in its principal diagonal are equal but not equal to one. A diagonal matrix is said to be a scalar matrix if its diagonal elements are equal, that is, a square matrix A = $\left[b_{ij}\right]_{n\times n}$ is said to be a scalar matrix if

b_{ij} = 0, when i ≠ j

b_{ij} = k, when i = j, for some constant kG 1.

For example,

$$A=\begin{bmatrix}-2 & 0 & 0\\ 0 & -5 & 0\\ 0 & 0 & 3\end{bmatrix}$$

Scalar matrices of order 3.

Zero or Null Matrix

A matrix is said to be zero matrix or null matrix if all its elements are zero.

For Example,

$$A=\left[0\right] \quad , A=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} \quad , A=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

are all zero matrices of the order 1, 2 and 3 respectively. We denote zero matrix by O.

Unit or Identity Matrix

If a square matrix has all elements 0 and each diagonal elements are non-zero, it is called identity matrix and denoted by I.

matrix A = [a_{ij}]n × n is an identity matrix if

a_{ij} = 1 if i = j

a_{ij} = 0 if i ≠ j

We denote the identity matrix of order n by In. When the order is clear from the context, we simply write it as I.

For example,

$$A=\begin{bmatrix}1\end{bmatrix}\ ,\ \ A=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\ ,\ \ A=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

are identity matrices of order 1, 2 and 3, respectively.

Upper Triangular Matrix

A square matrix in which all the elements below the diagonal are zero is known as the upper triangular matrix.

For example,

$$A=\begin{bmatrix}3 & -5 & 7\\ 0 & 4 & 0\\ 0 & 0 & 9\end{bmatrix}$$

Lower Triangular Matrix

A square matrix in which all the elements above the diagonal are zero is known as the upper triangular matrix.

For example,

$$A=\begin{bmatrix}3 & 0 & 0\\ 0 & 4 & 0\\ -5 & 7 & 0\end{bmatrix}$$

Addition Of Matrices:

Suppose A and B are two matrices of same order , then the addition of two matrices is obtained by adding the corresponding elements of A and B . it is denoted by A+B .If the order of A and B is m×n ,then the order of A+B will be m×n.

$$A+B=\begin{bmatrix} a_{11} & a_{12}\cdots a_{1n} \\ a_{21} & a_{22}\cdots a_{2n} \\ \vdots & \vdots \ \ddots \ \vdots \\ a_{m1} & a_{m2} \quad a_{mn} \end{bmatrix}+\begin{bmatrix} b_{11} & b_{12}\cdots b_{1n} \\ b_{21} & b_{22}\cdots b_{2n} \\ \vdots & \vdots \ \ddots \ \vdots \\ b_{m1} & b_{m2}\cdots b_{mn} \end{bmatrix}$$

$$=\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12}\cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22}\cdots & a_{2n}+b_{2n} \\ \vdots & \vdots \quad \ddots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & a_{mn}+b_{mn} \end{bmatrix}$$

For example:

$$A+B=\begin{bmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 2 \end{bmatrix}+\begin{bmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 1 \end{bmatrix}=\begin{bmatrix} 1+0 & 3+0 \\ 1+7 & 0+5 \\ 1+2 & 2+1 \end{bmatrix}=\begin{bmatrix} 1 & 3 \\ 8 & 5 \\ 3 & 3 \end{bmatrix}$$

Subtraction Of Matrices:

Suppose A and B are two matrices of same order ,then the subtraction of two matrices is obtained by subtracting the corresponding elements of A and B . it is denoted by A-B .If the order of A and B is m×n ,then the order of A-B will be m×n.

For example:

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

Note 1:

If the order of matrices is different, then they are not conformable for addition and subtraction.

Multiplication Of A Matrix By A Scalar:

Suppose A is a matrix of order m × n and k is a scalar, then the multiplication of A by k ,that is, kA is obtained by multiplying each element of A by k.

In general , if A = [a_{ij}] m × n is a matrix and k is a scalar, then kA is another matrix which is obtained by multiplying each element of A by the scalar k. In other words, kA = k [a_{ij}]m×n = [k (a_{ij})]m×n, that is, (i, j)th element of kA is kaij for all possible values of i and j.

For Example :

$$A=\begin{bmatrix} 0 & -1 & 5 \\ -3 & 2 & 1 \\ 2 & 0 & -4 \end{bmatrix}$$

c=-5 then cA is given by

$$cA=-5\begin{bmatrix} 0 & -1 & 5 \\ -3 & 2 & 1 \\ 2 & 0 & -4 \end{bmatrix}$$

$$=\begin{bmatrix} 0 & 5 & -25 \\ 15 & -10 & -5 \\ -10 & 0 & 20 \end{bmatrix}$$

Equality Of Matrices:

Two matrices are equal if all three of the following conditions are satisfied:

- Each matrix has the same number of rows.
- Each matrix has the same number of columns.
- Corresponding elements within each matrix are equal.

Consider the three matrices shown below.

$$A=\begin{bmatrix} 12 & x \\ y & 13 \end{bmatrix},\ B=\begin{bmatrix} 12 & 34 \\ 54 & 13 \end{bmatrix},\ C=\begin{bmatrix} o & p & q \\ l & m & n \end{bmatrix}$$

If A = B then we know that x = 34 and y = 54, since corresponding elements of equal matrices are also equal.

We know that matrix C is not equal to A or B, because C has more columns.

Properties Of Matrix Addition:

The basic properties of addition for real numbers also hold true for matrices.

Let A, B and C be a matrices of order m x n then

1.A + B = B + A commutative law holds good

2.A + (B + C) = (A + B) + C associative law holds good

3.There is a unique m x n matrix O with

A + O = A = 0 + A additive identity

4.For any m x n matrix A there is –A is the negative of A order m×n

-A +A = O (Null matrix) additive inverse

5.For any A, B and C matrices of order m x n then

(a). A + B = A + C ⟹ B = C [Left cancellation law]

(b). B + A = C +A ⟹ B = C [Right cancellation law]

Properties Of Multiplication Of Matrix By A Scalar:

If A and B are two m × n matrices (matrices of the same order) and k, a and b are the numbers (scalars). Then the following results are obvious.

I. k(A + B) = kA + kB [Distrubutive law]

II. (a + b)A = aA + bA

III. a(bA) = (ab)A

IV. (-k)A =-(kA)

Properties Of Matrix Multiplication:

If X , Y and Z be the matrices of order m×n , n×p , and p×q then

(1) Associative property:

(XY)Z=X(YZ)

(2) Distributive property:

X(Y+Z)=XY+XZ

Special Types Of Matrices :

(1) Hermitian Matrix:

A matrix A is said to be Hermitian matrix if (A^{θ}) $=$ A , where $A^{\theta} = \overline{(A)}^{T}$.

OR

If the elements of a matrix A are complex, then replacing these elements with their conjugates, a matrix known as the conjugate of A is obtained,and is denoted by A .The transpose of $\bar{A}$ *is* denoted by($\bar{A}$)^{T} reffered to as the transposed conjugate or tranjugate of A.

Example:

A=$\begin{bmatrix} 1 & 1-\mathrm{i}4 \\ 1+\mathrm{i}4 & 2 \end{bmatrix}$ and $\bar{A} = \begin{bmatrix} 1 & 1+\mathrm{i}4 \\ 1-\mathrm{i}4 & 2 \end{bmatrix}$

$$(\bar{A})^{T} = \begin{bmatrix} 1 & 1-\mathrm{i}4 \\ 1+\mathrm{i}4 & 2 \end{bmatrix} = A.$$

(2)Skew-Hermitian Matrix:

A matrix A is said to be Skew-Hermitian matrix if (A^{θ}) $= -$ A , where $A^{\theta} = \overline{(A)}^{T}$.

Example: A=$\begin{bmatrix} 3\mathrm{i} & 2+\mathrm{i} \\ -2+\mathrm{i} & \mathrm{i} \end{bmatrix}$ and $\bar{A} = \begin{bmatrix} -3\mathrm{i} & 2-\mathrm{i} \\ -2-\mathrm{i} & -\mathrm{i} \end{bmatrix}$

$$\overline{(A)}^{T} = \begin{bmatrix} -3\mathrm{i} & -2-\mathrm{i} \\ 2-\mathrm{i} & -\mathrm{i} \end{bmatrix}$$

$$= -\begin{bmatrix} 3\mathrm{i} & 2+\mathrm{i} \\ -2+\mathrm{i} & \mathrm{i} \end{bmatrix}$$

$$\overline{(A)}^{T} = -\text{ A.}$$

Note 2:

It can be easily verified that the diagonal elements of a Hermitian matrix are real and those in a Skew-Hermitian are imaginary (observe the above problem). The symmetric matrix is a particular case of Hermitian matrix if all the elements are real.

(3)Orthogonal Matrix:

A matrix A is said to be orthogonal if A^TA = I

Where I is unit matrix of order same as of order A.

(4)Unitary Matrix:

A square matrix A is said to be unitary matrix if $A^\theta A = I.$

Where I is unit matrix of order same as of order A.

Where $A^\theta = \overline{(A)}^T.$

Example:

If A=$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$, Then $A^T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

$$A^T A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$=\begin{bmatrix} \cos\theta^2 + \sin\theta^2 & 0 \\ 0 & \cos\theta^2 + \sin\theta\ ^2 \end{bmatrix}$$

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

$\Longrightarrow$ A is orthogonal matrix.

Example:

If A = $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$, *then* $\bar{A}$ = $\begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}$

$$A^\theta = \overline{(A)}^T = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$A^\theta A = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}=I$$

$\Longrightarrow$ A is unitary matrix.

Inverse matrices:

A system of linear equations can be written as AX = B in the matrix form. This form resembles one of the simplest linear equations in one variable ax = b whose solution is simply $x = a^{-1} b$ when $a \neq 0$. Thus it is tempting to write the solution of system as $X = A^{-1} b$. However, in the case of matrices we first have to assign a meaning to A^{-1}. To discuss this we begin with the following definition.

Definition:

For an m × n matrix A, an n × m B is called a left inverse of A if

BA = I, and an n × m matrix C is called a right inverse of A if AC = I.

Lemma 1: If an n × n square matrix A has a left inverse B and a right inverse C, then B and C are equal, that is B = C.

Proof: Since B is the inverse of A

AB =BA=I

Since C is also the inverse of A

AC=CA=I

Thus

B = B I = B(AC) = (BA)C = I C = C.

By lemma, one can say that if a matrix A has both left and right inverses, then any two left inverses must be both equal to a right inverse C, and hence to each other. By the same reason, any two right two inverses must be both equal to a left inverse B, and hence to each other. So there exists only one left and only one right inverse which must be equal.

Definition: An n × n square matrix A is said to be invertible (or non-singular) if their exist a square matrix B of the same size such that

$$AB = I = BA.$$

Such a matrix B is called the inverse of A, and is denoted by A^{-1}. A matrix A is said to singular if it is not invertible.

The above lemma implies that the inverse of a square matrix is unique

For instance ,consider 2×2 matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

If ad – bc ≠ 0, then it is easy to verify that

$$A^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$
$$= \begin{bmatrix} d/ad-bc & -b/ad-bc \\ -c/ad-bc & a/ad-bc \end{bmatrix}$$

Since A A^{-1} = I = A^{-1} A. Note that any zero matrix is singular.

Invertible Matrices:

Definition: If A is a square matrix of order m, and if there exists another square matrix B of the same order m, such that AB =BA=I, then B is called the inverse matrix of A and it is denoted by A^{-1}. In that case A is said to be invertible.

For Example, Let $A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} and\ B = \begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix}$ be two matrices.

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

Now
$$= \begin{bmatrix} 4-3 & -6+6 \\ 2-2 & -3+4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$$BA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Thus B is the inverse of A, in other words B=A^{-1} and A is inverse of B, i.e.,a=B^{-1} .

Note 3:

1. A rectangular matrix does not possess inverse matrix , since for products BA and AB to be defined and to be equal , it is necessary that matrices A and B should be square matrices of the same order.
2. If B is the inverse of A, then A is the inverse of B.

Theorem 1: The product of invertible matrices is also invertible, whose inverse is the product of the individual inverse in reversed order:

$$\mathbf{(AB)^{-1} = B^{-1}A^{-1}.}$$

Proof: Suppose that A and B are invertible matrices of the same size.

Then (AB) $(B^{-1}A^{-1})$ = (AB) $(B^{-1}\ A^{-1})$ $=AIA^{-1} = AA^{-1} = I$ and similarly $(B^{-1}A^{-1})$ (AB) = I.

Thus AB has the inverse $B^{-1}\ A^{-1}$.

The inverse of A is written as ‗A to the power -1‘. So one can give the meaning of Ak for any integer k. Let A be a square matrix. Then, for any positive integer k, we define the power Ak of A inductively as

$A^K = A\ (A^{K-1})$

Moreover, if A is invertible, then the negative integer power is defined as

$A^{-K} = (A^{-1})^K$for k>0.

Theorem 2: For any square matrix A with real number entries, A+ A' is a symmetric matrix and A- A' is a skew symmetric matrix.

Proof: Let B=A+ A', then

$$\begin{aligned} B' &= (A+A')' \\ &= A' + (A')' \quad (as\,(A+B)' = A' + B') \\ &= A' + A \qquad (as\,(A')' = A) \\ &= A + A' \qquad (as\ A+B = B+A) \\ &= B \end{aligned}$$

Therefore $B = A + A' is\, a\, symmertic\, matrix$

Nowlet $C = A - A'$

$$C' = (A - A') = A' - (A')'$$
$$= A' - A$$
$$= -(A - A') = -C$$

Therefore $C = A - A' is\, a\, skewsymmetric\, matrix$

Theorem 3: Any square matrix can be expressed as the sum of a symmetric and a skew symmetric matrix.

Proof: Let A be a square matrix, then we can write

$$A = \frac{1}{2}(A + A') + \frac{1}{2}(A - A')$$

From the above theorem 2, we know that $(A + A')$ is a symmetric matrix and is a skew symmetric matrix. Since for any matrix A, $(kA)' = kA'$, it follows that $\frac{1}{2}(A + A')$ is symmetric matrix and

$\frac{1}{2}(A - A')$ is a skew symmetric matrix. Thus, any square matrix can be expressed as the sum of a symmetric and a skew symmetric matrix .

Example: Express the matrix $B = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix}$ as the sum of a symmetric and a skew symmetric matrix.

Solution: Here $B' = \begin{bmatrix} 2 & -1 & 1 \\ -2 & 3 & -2 \\ -4 & 4 & -3 \end{bmatrix}$

$$Let \qquad P' = \frac{1}{2}(B+B') = \frac{1}{2}\begin{bmatrix} 4 & -3 & -3 \\ -3 & 6 & 2 \\ -3 & 2 & -6 \end{bmatrix} = \begin{bmatrix} 2 & \frac{-3}{2} & \frac{-3}{2} \\ \frac{-3}{2} & 3 & 1 \\ \frac{-3}{2} & 1 & -3 \end{bmatrix}$$

$$Now \qquad P' = \begin{bmatrix} 2 & \frac{-3}{2} & \frac{-3}{2} \\ \frac{-3}{2} & 3 & 1 \\ \frac{-3}{2} & 1 & -3 \end{bmatrix} = P$$

$$Thus \qquad P = \frac{1}{2}(B+B')\, is\, a\, symmetricmatrix.$$

$$Also, let \quad Q = \frac{1}{2}(B-B') = \frac{1}{2}\begin{bmatrix} 0 & -1 & -5 \\ 1 & 0 & 6 \\ 5 & -6 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{2} & \frac{-5}{2} \\ \frac{1}{2} & 0 & 3 \\ \frac{5}{2} & -3 & 0 \end{bmatrix}$$

$$Thus \qquad Q = \frac{1}{2}(B-B')\, is\, a\, skewsymmetricmatrix,$$

$$Now \qquad P+Q = \begin{bmatrix} 2 & \frac{-3}{2} & \frac{-3}{2} \\ \frac{-3}{2} & 3 & 1 \\ \frac{-3}{2} & 1 & -3 \end{bmatrix} + \begin{bmatrix} 0 & \frac{-1}{2} & \frac{-5}{2} \\ \frac{1}{2} & 0 & 3 \\ \frac{5}{2} & -3 & 0 \end{bmatrix} = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix} = B$$

Thus, B is represented as the sum of a symmetric and a skew symmetric matrix.

Elementary Operation (Transformation) of a Matrix:

There are six operations (transformations0 on a matrix, three of which are due to rows and three due to columns, which are known as elementary operations or transformations.

(i) The interchange of any two rows or two columns.
Symbolically the interchange of i^{th} and j^{th} rows is denoted by $R_i \leftrightarrow R_j$ and interchange of i^{th} and j^{th} column is denoted by $C_i \leftrightarrow C_j$.

For Example, applying $R_i \leftrightarrow R_j$ to $A = \begin{bmatrix} 1 & 2 & 1 \\ -1 & \sqrt{3} & 1 \\ 5 & 6 & 7 \end{bmatrix}$, we get $\begin{bmatrix} -1 & \sqrt{3} & 1 \\ 1 & 2 & 1 \\ 5 & 6 & 7 \end{bmatrix}$

(ii) The multiplication of the elements of any row or column by a non zero number.
Symbolically, the multiplication of each element of the i^{th} row by k, where $k \neq 0$ is denoted by $R_i \to kR_i$.

The corresponding column operation is denoted by $C_i \to kC_i$.

For Example, applying $C_3 \to \frac{1}{7} C_3, to\, B = \begin{bmatrix} 1 & 2 & 1 \\ -1 & \sqrt{3} & 1 \end{bmatrix}, we\, get \begin{bmatrix} 1 & 2 & \frac{1}{7} \\ -1 & \sqrt{3} & \frac{1}{7} \end{bmatrix}$

(iii) The additional to the elements of any row or column, the corresponding elements of any other row or column multiplied by any non zero number.
Symbolically, the addition to the elements of i^{th} row, the corresponding elements of j^{th} row multiplied by k is denoted by $R_i \to R_i + kR_j$.

The corresponding column operation is denoted by $C_i \to C_i + kC_j$.

For Example, applying $R_2 \to R_2 - 2R_1, to\, C = \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}, we\, get \begin{bmatrix} 1 & 2 \\ 0 & -5 \end{bmatrix}$.

Inverse of a matrix by elementary operation:

Let X, A and B be matrices of the same order such that X=AB. In order to apply a sequence of elementary row operation on the matrix equation X=AB, we will apply these row operations simultaneously on X and and on the first matrix A of the product AB on RHS.

Similarly, in the order to apply a sequence of elementary column operations on the matrix equation X=AB, we will apply, these operations simultaneously on X and on the second matrix B of the product AB on RHS.

In the view of the above discussion, we conclude that if A is a matrix such that A^{-1} exists, then to find A^{-1} using elementary row operations, write A=IA and apply a sequence of row operation on A=AI till we get I=BA. The matrix B will be the inverse of A. Similarly, if we wish to find A^{-1} using column operations, then, write A=AI and apply a sequence of column operations on A=AI till we get, I=AB.

Remark : In case, after applying one or more elementary row (column) operations on A=IA (A=AI), if we obtain all zeros in one or more rows of the matrix A on L.H.S then A^{-1} does not exist.

For Example: By using elementary operations, find the inverse of the matrix $A=\begin{bmatrix}1 & 2\\ 2 & -1\end{bmatrix}$.

Solution: In order to use elementary row operations we may write A=IA.

$$\begin{bmatrix}1 & 2\\ 2 & -1\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}A, then \begin{bmatrix}1 & 2\\ 0 & -5\end{bmatrix}=\begin{bmatrix}1 & 0\\ -2 & 1\end{bmatrix}A \ (applying\ R_2 \to R_2 - 2R_1)$$

$$\begin{bmatrix}1 & 2\\ 0 & 1\end{bmatrix}=\begin{bmatrix}1 & 0\\ \frac{2}{5} & \frac{-1}{5}\end{bmatrix}A \ (applying\ R_2 \to -\frac{1}{5}R_2)$$

$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}=\begin{bmatrix}\frac{1}{5} & \frac{2}{5}\\ \frac{2}{5} & \frac{-1}{5}\end{bmatrix}A \ (applying\ R_1 \to R_1 - 2R_2)$$

Thus $A^{-1}=\begin{bmatrix}\frac{1}{5} & \frac{2}{5}\\ \frac{2}{5} & \frac{-1}{5}\end{bmatrix}$.

HAPTER-02

APPLICATIONS

ELECTRICAL NETWORKS

The simplest electrical circuit consist of two basic components.

Electrical resources denoted by

Resistors denoted ~~by~~ /\/\/\/\/\/

Electrical resources, such as batteries, create current in an electrical circuits, Resistors, such as lightbulbs, limit the magnitudes of the currents.

The voltage is measured in volts, the resistance in ohms, and the current flow in amperes.

The flow of current in an electrical circuit is governed by three basic principles (laws).

- Ohm‘s law : The voltage drop (V) across the resistor is product of current (I) and the resistance (R);that is, V=IR
- **Kirchhoff's current law**: The sum of the currents flowing into any point equals the sum of the currents flowing out from the point.
- **Kirchhoff's voltage law**: Around any closed loop, the algebraic sum of the voltage drops zero.

Example: Find the unknown currents I_1 ,I_2 and I_3 in the circuit diagram as shown in the figure

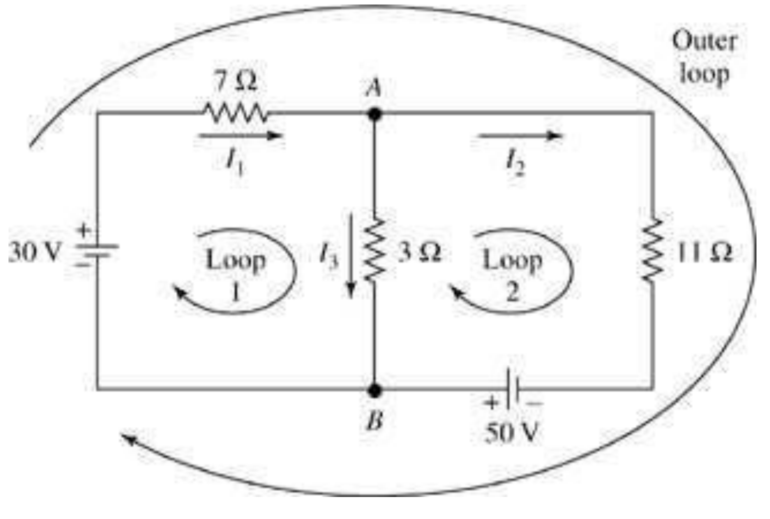


Figure 1

Solution:

The flow direction for the currents I_1 ,I_2 and I_3 (marked by the arrowheads) were picked arbitrarily. Any of these currents that turn out to be negative actually flow opposite to be direction selected.

Applying Kirchhoff‘s current law to points A and B yields

$$I_1 = I_2 + I_3 \text{ (point A)}$$

$$I_3 + I_2 = I_1 \qquad \text{(point B)}$$

Since these equations both simplify to the same linear equation

$$I_1 - I_2 - I_3 = 0 \qquad (1)$$

We still need two more equations to determine I_1 ,I_2 and I_3 uniquely .We will obtain them using Kirchhoff‘s voltage law.

To apply Kirchhoff‘s voltage law to a loop, select a positive direction around the loop (say clockwise) and make the following sign conventions:

- A current passing through a resistor produces a positive voltage drop if it flows in the positive direction of the loop and a negative voltage drop if it flows in the negative direction of the loop.
- A current passing through an electrical source produces a positive voltage drop if the positive voltage drop if the positive direction of the loop is from + to – and a negative voltage drop if the positive direction of the loop is from – to +.

Applying Kirchhoff‘s voltage law and ohm‘s to loop 1 in figure yields

$$7I_1 + 3I_3 - 30 = 0 \qquad (2)$$

And applying them to loop 2 yields

$$11I_2 - 3I_3 - 50 = 0 \qquad (3)$$

Combining 1,2 and 3 yields the linear system

$$I_1 - I_2 - I_3 = 0$$
$$7I_1 \quad + 3I_3 = 30$$
$$11I_2 - 3I_3 = 50$$

Solving this linear system yields the following values for the currents:

$$I_1 = \frac{570}{131} \text{ (A)}, \qquad I_2 = \frac{590}{131} \text{ (A)}, \qquad I_3 = -\frac{20}{131}\text{(A)}$$

I_3 is negative, which means this current flows opposite to the direction indicated in figure.

GRAPH THEORY

A graph G is a finite set of points called vertices or nodes, together with a finite set of edges, each of which joins a pair of vertices. An edge joining a vertex to itself is called a loop.

A **Directed graph** or a digraph is a finite set of points called vertices or nodes, together with a finite set of directed edges, each of which joins an ordered pair of distinct vertices. Thus a digraph contains no loops. Let us also assume that there is no multiple edges. Moreover, the directed edge P_{ij} now different from the directed edge P_iP_j. The matrix A(G), whose i,jth element is 1 if there is a directed edge from P_i to P_j and zero otherwise is called the adjacency matrix of G.

we use the notation $P_i \rightarrow P_j$ that is, (P_i connected to P_j) to indicate the directed edge (P_i, P_j) belongs to the directed graph.

Geometrically, we can visualize a directed graph by representing the vertices as the points in the plane and representing the directed edge $P_i \rightarrow P_j$ by drawing a line or arc from $P_i \rightarrow$ to P_j. If both $P_i \rightarrow P_j$ and $P_i \quad P_j$ hold (denoted by $P_i \leftrightarrow P_j$) .we draw a single line between P_i and $P_j$with two oppositely pointing arrows. Also, P_i to P_j is not permitted in a directed graph.

With a directed graph having n vertices, we may associate a n x n matrix M={m_{ij}} called the vertex matrix of the directed graph

$$m_{ij} = \{ \begin{matrix} 1 & \rightarrow \text{if } p_1 \quad p_2 \\ 0 & otherwise \end{matrix}$$

Example 1:

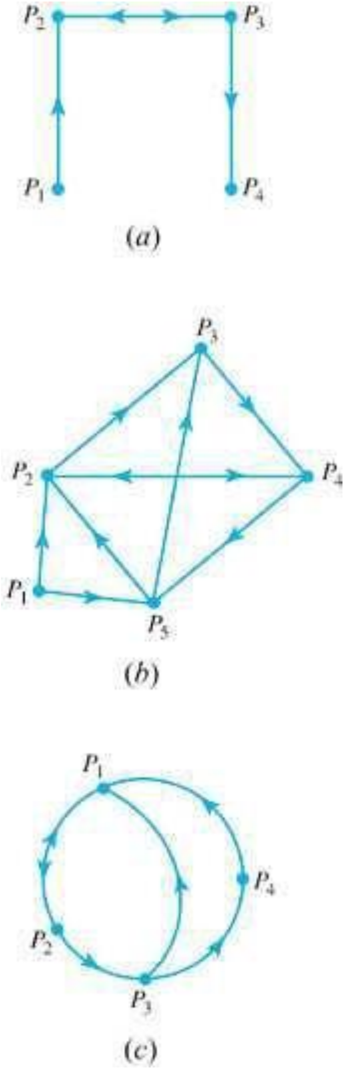


Figure 1.1

for i ,j=1 , 2, … n. For the three directed graphs in Figure 11.7.2, the corresponding vertex matrices are

Figure 1.1(a) $M= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Figure 1.1(b) $M= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$

Figure1.1 (c) $M= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

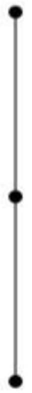
By their definition, vertex matrix have the following two properties

- All entries are either 0 or 1.
- All diagonal entries are 0.

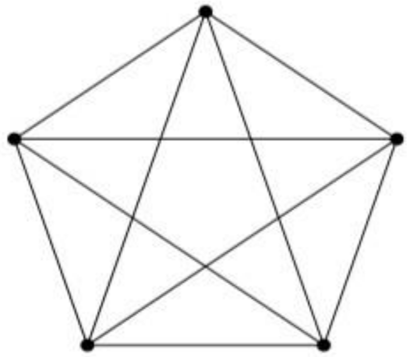
Graph Theory(Adjacency Matrix):

Graph theory is a relatively new branch of mathematics which deals with the study of objects named graphs. These types of graphs are not of the variety with an x-axis and y-axis, but rather are made up of vertices, usually represented as points, and edges, usually thought of as lines in between two vertices.

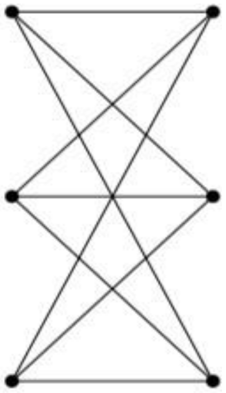
The following are examples of graphs:



P_3



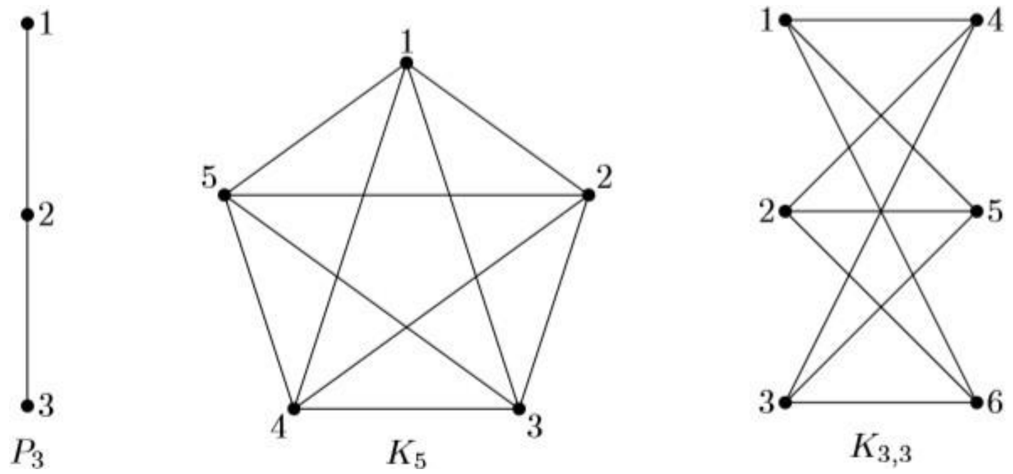
K_5



$K_{3,3}$

P_3, called the path on 3 vertices, can be generalized in the obvious way to a path on n vertices, P_n. Likewise, K_n is the complete graph on n vertices, in which all possible edges are present (NB: we do not consider graphs with more than one edge between two vertices). Lastly, Kr,s is the complete bipartite graph with parts made up of r and s vertices. In general, a bipartite graph is a graph in which the vertices can be split into two parts, where all edges of the graph are in between these two parts. That is, the two parts have no edges inside of them.

We may label the vertices of our graph in any way we would like, but, for example, we could label the vertices of the graphs above in the following ways:



with these labellings, we are ready to see a matrix that can be associated with each of these graphs. Before we do this, let us discuss a bit of notation. Now, we simply refer to a vertex in a graph by its label. Thus, we call the leftmost vertex in the representation above of P_3 simply 1. Further, if two vertices i and j are adjacent, or have an edge between them, we write $i \sim j$. Likewise, if they are not adjacent, or do not have an edge between them, we write $i \nsim j$. So, in P_3 as labelled above, $1 \sim 2$, $2 \sim 3$, but $1 \nsim 3$. Also, note that if $i \sim j$ then $j \sim i$.

We define the **adjacency matrix** of a graph G (we assume that G has n vertices) to be the n×n **matrix A(G) = (a_{ij})** with

$$(a_{ij}) = \begin{cases} 1 & \text{if } i \sim j \text{ or if vertex } v_i \text{ and } v_j \text{ are adjacent.} \\ 0 & \text{if } i \nsim j \text{ or if vertex } v_i \text{ and } v_j \text{ are not adjacent} \end{cases}$$

Thus, we can see that the adjacency matrices of the graphs as labelled above are:

$$A(P_3)=\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad A(K_5)=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$A(K_{3,3}) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

You may have already noticed some characteristics of these matrices. First of all, we have (implicitly up to this point) assumed that there are no loops, or edges which go from a vertex to itself. Thus, for any such graph, there are zeros down the diagonal entries, or $a_{ii} = 0$ for i = 1,...,n. Also, since each vertex is represented by both a row and a column, the adjacency matrix is symmetric, or has $a_{ij} = a_{ji}$ for all j = 1,...,n and i = 1,...,n.

We Have The Following Observation About The Adjacency Matrix A :

(1) The adjacency matrix is a square , symmetric and a binary matrix (it is having entries as 0 and 1) of order n.

(2) The entries along the principle diagonal of A(G) are all zero's (since the graph G has no loops).

(3) The degree of vertex of a hraph G is equal to the number of 1's in the corresponding row or column of matrix.

Graph Theory(Incidence Matrix):

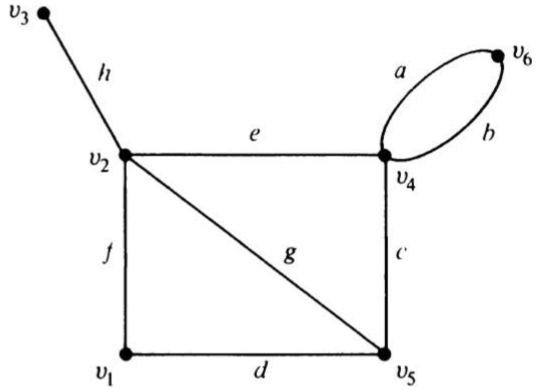
Let G be a graph with n vertices , e edges and no self-loops.

Define n by e matrix A = [a_{ij}] ,whose n rows correspond to the n vertices and the e columns correspond to the e edges , as follow

The matrix element

$$a_{ij} = \begin{cases} 1 & \text{if } edge\ e\text{j is } incident\ on\ vertex\ v\text{i } and \\ 0 & otherwise \end{cases}$$

Such a matrix A is called the **vertex – edge incidence matrix , or simply incidence matrix.** a graph G and its incidence matrix are shown in below figure.



$$
\begin{array}{c|cccccccc|}
 & a & b & c & d & e & f & g & h \\
\hline
v_1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
v_2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
v_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
v_4 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
v_5 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
v_6 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Graph And Its Incident Matrix

We Have The Following Observations About The Incidence Matrix A :

- The incident matrix contains only two elements , 0 and 1. Such a matrix is called a binary matrix or a (0, 1)-matrix.
- Since every edge is incident on exactly two vertices, each column of A has exactly two 1′ s.
- The number of 1′ s in each row equals the degree of the corresponding Vertex vi
- A row with all 0′ s, therefore, represented an isolated vertex.
- A parallel edges in a graph produce identical columns in its incidence matrix,for example , columns 1 and 2.

GEOMETRIC LINEAR PROGRAMMING

The study of linear programming theory has expanded greatly since the pioneering work of George Dantzig in the late 1940s. Today linear programming is applied to a wide variety of problems in industry and science. In this section we present a geometric approach to the solution of linear programming problems, we shall solve problems

Maximizing Sales Revenue:

Example 1: A candy manufacturer has 130 pounds of chocolate-covered cherries and 170 pounds of chocolate-covered mints in stock. He decided to sell them in the form of 2 different mixtures. One mixture will contain half cherries and half mints by weight and will sell for $ 2.00 per pound. The other mixture will contain one-third and two-third mints by weight and will sell for $ 1.25 per pound. How many pounds of each mixture should be candy manufacturer prepare in order to maximize his sales revenue?

Solution: Let the mixture of half cherries and half mints be called mixture A, and Let $x_1$be the number of pounds of this mixture to be prepared. Let the mixture of one-third cherries and two-third mints be called mixture B, and Let x_2 be the number of pounds of this mixture to be prepared. Since mixture A sells for $ 2.00 per pound and mixture B sells for $ 1.25 per pound, the total sales z (in dollars)

will be

$$z=2.00x_1 +1.25x_2$$

since each pound of mixture A contains $^1/_2$ pound of cherries and each pound of mixture B contains $^1/_3$ pound of cherries, the total number of cherries used in both mixture is

$$^1/_2\,x_1 + ^1/_3\,x_2$$

Similarly, since each pound of mixture a contains $^1/_2$ pound of mints and each pound of mixture B contains

$$^1/_2\,x_1 + ^2/_3\,x_2$$

The manufacturer can use at most 130 pounds of cherries and 170 pounds of mints, we have

$$^1/_2\,x_1 + ^1/_3\,x_2 \le 130$$

$$^1/_2\,x_1 + ^2/_3\,x_2 \le 170$$

$$x_1 \ge 0 \text{ and } x_2 \ge 0$$

The problem can therefore be formulated mathematically as follows: Find the values of x_1 and x_2 that maximize

$$z=2.00x_1 +1.25x_2$$

subjected to

$$^1/_2\,x_1 + ^1/_3\,x_2 \le 130$$

$$^1/_2\,x_1 + ^2/_3\,x_2 \le 170$$

$$x_1 \ge 0$$

$$x_2 \ge 0$$

Maximizing Annual yield

Example 2: A woman has up to \$ 10,000 to invest. Her broker suggests investing in two bonds, A and B. Bond A is a rather risky bond with an annual yield of 10%, and bond B is a rather safe bond with an annual yield of 7%. After some consideration, she decides to invest at most \$2000 in bond B, and to invest at least as much in bond A as in bond B. How should she invest her \$10,000 in order to maximize her annual yield?

Solution: Let x_1 be the number of dollars to be invested in bond A, and let x_2 be the number of dollars to be invested in bond B. Since each dollar invested in bond A earns \$ .10 per year and each dollar invested in bond B earns \$.07 per year, the total dollar amount z earned each year by both bonds is

$$z=.10x_1+.07x_2$$

The constraints imposed can be formulated mathematically as follows:

Invest no more than \$ 10,000: $x_1 + x_2 \leq 10,000$

Invest at most \$ 6000 in bond A: $x_1 \leq 6000$

Invest at least \$ 2000 in bond B: $x_2 \geq 2000$

Invest at least as much in bond A as in bond B: $x_1 \geq x_2$

We also have the implicit assumption that x_1 and x_2 are nonnegative:

$$x_1 \geq 0 \quad \text{and} \quad x_2 \geq 0$$

Thus the complete mathematical formulation of the problem is as follows: Find values of x_1 and x_2 that maximize

$$z=.10x_1+.07x_2$$

subject to

$$x_1 + x_2 \leq 10,000$$
$$x_1 \leq 6000$$
$$x_2 \geq 2000$$
$$x_1 - x_2 \geq 0$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

The feasible Region is a Line Segment

Example: Find the values of $x_1$and $x_2$that minimize

$$z = 2x_1 - x_2$$

subject to

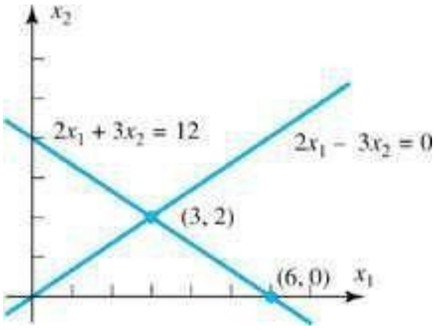
$$2x_1 + 3x_2 = 12$$

$$2x_1 - 3x_2 \geq 0$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Solution: We have drawn the feasible region of this problem. Because one of the constraints is an equality constraint, the feasible region is a straight line segment with two extreme points. The values of z at the two extreme points are given in the following table.



Figure

Extreme point (x_1, x_2)	Value of $z= 2x_1 - x_2$
(3, 2) (6, 0)	4 12

The minimum values of z is thus 4 and attained at x_1=3 and x_2=2.

Geometric Solution of Linear Programming Problems

Each of the preceding three examples is a special case of the following problem.

Problem: Find values of and that either maximize or minimize

$$z = c_1x_1 + c_2x_2 \tag{1}$$

subject to

$$a_{11}x_1 \;+\; a_{12}x_2 \;\; (\leq)\,(\geq)\,(=)\;\; b_1$$

$$a_{21}x_1 \;+\; a_{22}x_2 \;\; (\leq)\,(\geq)\,(=)\;\; b_2 \tag{2}$$

$$\vdots$$

$$a_{m1}x_1 \quad + \quad a_{m2}x_2 \;\; (\leq)\,(\geq)\,(=) \;\; b_m$$

and

$$x_1 \geq 0, \qquad x_2 \geq 0 \qquad\qquad (3)$$

In each of the *m* conditions of 2, any one of the symbols ≤, ≥, and = may be used.

The problem above is called the ***general linear programming problem*** in two variables.

The linear function *z* in 1 is called the ***objective function***. Equations 2 and 3 are called the

Constraints; in particular, the equations in 3 are called the ***nonnegativity constraints*** on the

variables x_1 and x_2.

We shall now show how to solve a linear programming problem in two variables graphically.

A pair of values $(x_1,\ x_2)$ that satisfy all of the constraints is called a ***feasible solution***. The set of all feasible solutions determines a subset of the x_1x_2-plane called the ***feasible region***. Our desire is to find a feasible solution that maximizes the objective function. Such a solution is called an ***optimal***

solution.

To examine the feasible region of a linear programming problem, let us note that each constraint of the form

$$a_{i1}x_1 + a_{i2}x_2 = b_i$$

defines a line in the x_1x_2-plane, whereas each constraint of the form

$$a_{i1}x_1 + a_{i2}x_2 \leq b_i \quad \text{or} \quad a_{i1}x_1 + a_{i2}x_2 \geq b_i$$

defines a half-plane that includes its boundary line

$$a_{i1}x_1 + a_{i2}x_2 = b_i$$

Thus the feasible region is always an intersection of finitely many lines and half-planes. For example, the four constraints

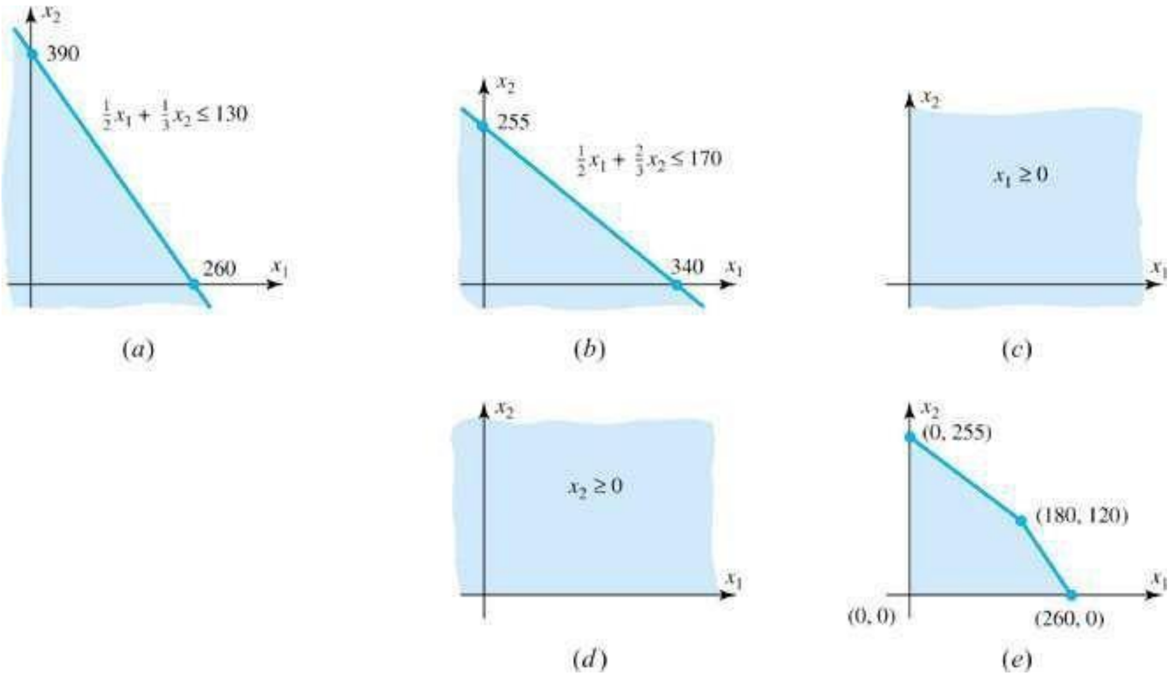
$$\tfrac{1}{2}x_1 + \tfrac{1}{3}x_2 \leq 130$$

$$\tfrac{1}{2}x_1 + \tfrac{2}{3}x_2 \leq 170$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

of Example 1 define the half-planes illustrated in parts (*a*), (*b*), (*c*), and (*d*) of Figure . The feasible region of this problem is thus the intersection of these four half-planes, which is illustrated in Figure



It can be shown that the feasible region of a linear programming problem has a boundary consisting of a finite number of straight line segments. If the feasible region can be enclosed in a sufficiently large circle, it is called ***bounded*** (Figure); otherwise,it is called ***unbounded*** (see Figure 11.3.5). If the feasible region is *empty* (contains no points), then the constraints are inconsistent and the linear programming problem has no solution (see Figure 11.3.6). Those boundary points of a feasible region that are intersections of two of the straight line boundary segments are called ***extreme points***. (They are also called *corner points* and *vertex points*.) For example, in Figure 11.3.1*e*, we see that the feasible region of Example 1 has four extreme points:

$$(0, 0), \quad (0, 25), \quad (180, 120), \quad (260, 0) \qquad (4)$$

MARKOV CHAINS

A Markov process:

Suppose a physical or mathematical system undergoes a process of change such that at any moment it can occupy one of finite number of states. For example, the weather in a certain city could be in one of three possible states: sunny, cloudy, or rainy, or an individual could be one of four possible emotional states: happy, sad, angry, or apprehensive.

Suppose that such a system changes with time from one state to another and at scheduled times from the state of the system is observed. If the state of the system at any observation cannot be predicted with certainty, but the probability that a given state occurs can be predicted by just knowing the state of the system at the preceding observation, then the process of change is called a ***Markov chain*** or ***Markov process*****.**

Definition

If a Markov chain has *k* possible states, which we label as 1,2,…….,*k***,** then the probability that the system is in state *i* at any observation after it was in state *j* at the preceding observation is denoted by p_{ij} and is called the ***transition probability*** from state *j* to state *i*. The matrix $P = [\ p_{ij}]$ is the ***transition matrix of any Markov chain***

For example, in a three-state Markov chain, the transition matrix has the form

$$\begin{array}{cccc} & \textbf{Preceding State} & & \\ \textbf{1} & \textbf{2} & \textbf{3} & \\ p_{11} & p_{12} & p_{13} & \\ [p_{21} & p_{22} & p_{23}] & \textbf{New state} \\ p_{31} & p_{32} & p_{33} & \end{array}$$

In this matrix, p_{32} is the probability that the system will change from state 2 to state 3, p_{11} is the probability that the system will still be in state 1 if it was previously in state 1, and so fourth.

Transition matrix of the Markov chain:

Example 1 : By reviewing its donation records, the alumni office of college finds that 80% of its alumni who contribute to the annual fund one year will also contribute the next year, and 30% of those who do not contribute one year will contribute the next. This can be viewed as a Markov chain with two states: state 1 corresponds to an alumnus giving a donation in any one year, and state 2 corresponds to the alumnus not giving a donation in that year. The transition matrix is

$$P=\begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix}$$

In the example above, the transition matrices of the Markov chains have the property that the entries in any column sum to 1. This is not accidental. If $P=[\,p_{i\ j}]$ is the transition matrix of any Markov chain with *k* states, then for each *j* we must have

$$p_{1j}+p_{2\,j}+\cdots p_{k\,j}=1 \text{ ....}(1)$$

Because if the system is in state *j* at one observation, it is certain to be in one of the *k* possible states at that next observation.

A matrix with property 1 is called a stochastic matrix, or a Markov matrix. From the preceding discussion, it follows that the transition matrix for a Markov chain must be a stochastic matrix

Theorem: If P is the transition matrix of a Markov chain and $x^{(n)}$ is the state vector at the nth observation, then $x^{(n+1)}$=P$x^{(n)}$

Proof: The proof of this theorem involves ideas from probability theory and will not be given here. From this theorem, it follows that

$$x^{(1)}=Px^{(0)}$$
$$x^{(2)}=Px^{(1)}=P^2x^{(0)}$$
$$x^{(3)}=Px^{(2)}=P^3x^{(0)}$$
$$\vdots$$
$$x^{(n)}=Px^{(n-1)}=P^nx^{(0)}$$

In this way, the initial state vector $x^{(0)}$ and the transition matrix *P* determine $x^{(n)}$ for n=1, 2, ……….

Example 2: The transition matrix in Example 1 was

$$P=\begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix}$$

We now construct the possible future donation record of a new graduate who did not give a donation in the initial year after graduation. For such a graduate the system is initially in state 2 with certainty, so the initial state vector is

$$x^{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

From theorem, we have

$$x^{(1)} = Px^{(0)} = \begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} .3 \\ .7 \end{bmatrix}$$

$$x^{(2)} = Px^{(1)} = \begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix} \begin{bmatrix} .3 \\ .7 \end{bmatrix} = \begin{bmatrix} .45 \\ .45 \end{bmatrix}$$

$$x^{(3)} = Px^{(2)} = \begin{bmatrix} .8 & .3 \\ .2 & .7 \end{bmatrix} \begin{bmatrix} .45 \\ .45 \end{bmatrix} = \begin{bmatrix} .525 \\ .475 \end{bmatrix}$$

Thus, after three years the alumnus can be expected to make a donation with probability .525. Beyond three years, we find the following state vectors

$$x^{(4)} = \begin{bmatrix} .563 \\ .438 \end{bmatrix}, \quad x^{(5)} = \begin{bmatrix} .581 \\ .419 \end{bmatrix}, \quad x^{(6)} = \begin{bmatrix} .519 \\ .409 \end{bmatrix}, \quad x^{(7)} = \begin{bmatrix} .595 \\ .405 \end{bmatrix}$$

$$x^{(8)} = \begin{bmatrix} .598 \\ .402 \end{bmatrix}, \quad x^{(9)} = \begin{bmatrix} .599 \\ .401 \end{bmatrix}, \quad x^{(10)} = \begin{bmatrix} .599 \\ .401 \end{bmatrix}, \quad x^{(11)} = \begin{bmatrix} .600 \\ .400 \end{bmatrix}$$

For beyond n 11, we have

$$x^{(11)} = \begin{bmatrix} .600 \\ .400 \end{bmatrix}$$

to three decimal places. In other words, the state vectors converge to a fixed vector as the number of observations increases.

Regular transition matrix

A transition matrix is regular if some integer power of it has all positive entries.

Thus, for a regular transition matrix P , there is some positive integer m such that all entries of p^m are positive.

A Markov chain that is governed by a regular transition matrix is called Markov chain

Behavior of p^n as n → ∞

Theorem: If P is a regular transition matrix, then as n → ∞

$$\rightarrow \quad p^n \quad \begin{bmatrix} q_1 & q_1 & \cdots & q_1 \\ q_2 & q_2 & \cdots & q_2 \\ \vdots & \vdots & & \vdots \\ q_k & q_k & \cdots & q_k \end{bmatrix}$$

Where the q_i are positive numbers such that $q_1 + q_2 + \ldots\ldots\ldots\ldots\ldots\ldots\ldots q_k = 1$

Solution: Let

$$Q = \begin{bmatrix} q_1 & q_1 & \cdots & q_1 \\ q_2 & q_2 & \ldots & q_2 \\ \vdots & \vdots & & \vdots \\ q_k & q_k & \cdots & q_k \end{bmatrix} \quad \text{and} \quad q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_k \end{bmatrix}$$

Thus, *Q* is a transition matrix, all of whose columns are equal to the probability vector q. *Q* has the property that if ***x*** is any probability vector, them

$$Qx = \begin{bmatrix} q_1 & q_1 & \cdots & q_1 \\ q_2 & q_2 & \ldots & q_2 \\ \vdots & \vdots & & \vdots \\ q_k & q_k & \cdots & q_k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} q_1x_1 + & q_1x_2 & \cdots & q_1x_k \\ q_2x_1 + & q_2x_2 & \ldots & q_2x_k \\ \vdots & \vdots & & \vdots \\ q_kx_1 + & q_kx_2 & \cdots & +q_kx_k \end{bmatrix}$$

$$= (x_1 + x_2 + \ldots\ldots\ldots\ldots\ldots x_k) \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_k \end{bmatrix} = (1)\mathbf{q} = \mathbf{q}$$

That is, *Q* transforms any probability vector ***x*** into the fixed probability vector **q**.

CRYPTOGRAPHY

Ciphers:

The study of encoding and decoding secret messages is called ***cryptography***. Although secret codes date to the earliest days of written communication, there has been a recent surge of interest in the subject because of the need to maintain the privacy of information transmitted over public lines of communication. In the language of cryptography, codes are called ***ciphers***, uncoded messages are called ***plaintext***, and coded messages are called ***ciphertext***. The process of converting from plaintext to ciphertext is called ***enciphering***, and the reverse process of converting to plaintext is called ***deciphering.***

The simplest ciphers, called substitution ciphers, are those that replace each letter of the alphabet by a different letter. For example, in the substitution cipher

Plain	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
Cipher	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*

The plaintext letter *A* is replaced by *D*, the plaintext *B* by *E*, and so forth. With this cipher the plaintext message

ROME WAS NOT BUILT IN A DAY

becomes

URPH ZDV QRW EXLOW LO D GDB

Hill Ciphers

A disadvantage of substitution ciphers is that they preserve the frequencies of individual letters, making it relatively easy to break the code by statistical methods. One way to overcome this problem is to divide the plaintext into groups of letters and encipher the plaintext group by group, rather than one letter at a time.

A system of cryptography in which the plaintext is divided into sets of *n* letters, each of which is replaced by a set of *n* cipher letters, is called a ***polygraphic system.*** In this section we will study a class of polygraphic systems based on matrix transformations.(The ciphers that we will discuss are called Hill ciphers after Lester S.Hill, who introduced them in two papers: –Cryptography in an Algebraic Alphabet‖, *American Mathematical Monthly*, 36(June-July 1929), pp. 306-312; and –Concerning Certain Linear Transformation Apparatus of Cryptography‖, *American Mathematical Monthly*, 38(March 1931), pp. 135-154)

We assume that each plaintext and ciphertext letter except Z is assigned the numerical value that specifies its position in the standard alphabet (Table 1). For reasons that will become clear letter, Z is assigned a value of zero.

Table 1

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0

In this simplest Hill ciphers, successive pairs of plaintext are transformed into ciphertext by the following procedure:

Step 1: Choose a 2×2 matrix with integer entries A=$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ to perform encoding.

Certain additional conditions on A will be imposed later.

Step 2: Group successive plaintext letters into pairs, adding an arbitrary –dummy‖ letters to fill out the last pair if the plaintext has an odd number of letters,

and replace each plaintext letter by its numerical value.

Step 3: Successively convert each plaintext pair $p_1p_2$into a column vector

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$$

and form the product A_p. We will call ***p*** a ***plaintext ve***ctor and A_p the corresponding ***ciphertext vector***.

Step 4: Convert each ciphertext vector into its alphabetic equivalent.

Hill cipher of a message

Example 1:

Use the matrix

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

to obtain the Hill cipher for the plaintext message

I AM HIDING

Solution:

If we group the plaintext into pairs and the dummy letter G to fill out the last pair, we obtain *IA* *MH* *ID* *IN* *GG*

or, equivalently, from Table 1,

9 1 13 8 9 4 9 14 7 7

To encipher the pair *IA*, we form the matrix product

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 9 \\ 1 \end{bmatrix} = \begin{bmatrix} 11 \\ 3 \end{bmatrix}$$

which, from Table 1, yields the ciphertext *KC*.

To encipher the pair *MH*, we form the product

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 29 \\ 24 \end{bmatrix} \qquad \text{……………………} (1)$$

However, there is a problem here, because the number 29 has no alphabet equivalent (Table 1). To resolve this problem, we make the following agreement:

Whenever an integer greater than 25 occurs, it will be replaced by the remainder that results when this integer is divided by 26.

Because the remainder after division by 26 is one of the integers 0, 1, 2,…,25, this procedure will always yield an integer with an alphabet equivalent.

Thus, in 1 we replace 29 bt 3, which is the remainder after dividing 29 by 26. It follows from Table 1 that the ciphertext the pair *MH* is *CX*.

The computations for the remaining ciphertext vectors are

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 9 \\ 4 \end{bmatrix} = \begin{bmatrix} 17 \\ 12 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 9 \\ 14 \end{bmatrix} = \begin{bmatrix} 37 \\ 42 \end{bmatrix} \quad or \quad \begin{bmatrix} 11 \\ 16 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 7 \\ 7 \end{bmatrix} = \begin{bmatrix} 21 \\ 21 \end{bmatrix}$$

These correspond to the ciphertext pairs *QL, KP,* and *UU* respectively.

The entire ciphertext message is

KC CX QL KP UU

which would be usually be transmitted as a single string without spaces

KCCXQLKPUU

Because the plaintext was grouped in pairs and enciphered by a 2×2 matrix, the Hill cipher in Example 1 is referred to as a ***Hill 2-cipher***. It is obviously also possible to group the plaintext in triples and encipher by a 3×3 matrix with integer entries; This is called ***Hill 3-cipher***. In general for a ***Hill n-cipher***, plaintext is grouped into sets pf *n* letters and enciphered by an *n*×*n* matrix with integer entries.

Modular Arithmetic:

In example 1, integers greater than 25 were replaced by their remainders after division by 26. This technique of working with remainders is at the core of a body of mathematics is called modular arithmetic. Because of its importance in cryptography, we will digress for a moment to touch on some of the main ideas in this area.

In modular arithmetic, we are given a positive integer *m*, called the modulus, and any two integers whose difference is an integer multiple of the modulus are regarded as –equal‖ or –equivalent‖ with respect to the modulus.

Definition:

If m is a positive integer and a and b are any integers, then we say that *a* is equivalent to *b* modulo *m*, written

$$a = b \qquad (\text{mod } m)$$

if *a-b* is an integer multiple of *m*

Residues mod 26

Example 2: Find the residue modulo 26 of (a) 87,(b) -38, and (c)-26

Solution (a)

Dividing $|87|$=87 by 26 yields a remainder of *R=9,* so *r*=9. Thus,

$$87=9 \qquad (\text{mod } 26)$$

Solution (b)

Dividing $|-38|$=38 by 26 yields a remainder of *R*=12, so *r*=26-12=14. Thus,S

$$-38 = 14 \qquad (\text{mod } 26)$$

Solution (c)

Dividing $|-26|$=26 by 26 yields a remainder of R=0, Thus,

$$-26 = 0 \qquad (\text{mod } 26)$$

LEONTIEF ECONOMIC MODELS:

Economic Systems

Matrix theory has been very successful in describing the interactions among prices, outputs and demands in economic systems. In this section we discuss some simple models based on the ideals of Nobel laureate Wassily Leontief. We examine two different but related models: the closed or input-output model, and the open or production model. In each, we are given certain economic parameters that describe the interrelations between the ―indusrties‖ in the economy under consideration. Using matrix theory, we then evaluate certain other parameters, such as prices or output levels, in order to satisfy a desired economic objective. We begin with the closed model.

Leontief closed (Input-Output) Model:

An Input-output model

Example 1: Three homeowners-a carpenter, an electrician, and a plumber-agree to make repairs in their three homes. They agree to work a total of 10 days each according to the following schedule:

	Work performed by		
	Carpenter	**electrician**	**plumber**
Days of work in Home carpenter	2	1	6
Days of work in Home Electrician	4	5	1

Days of work in Home plumber	4	4	3

For tax purposes, they must report and pay each other a reasonable daily wage, even for the work each does on his or her own home. Their normal daily wages are about $100, but they agree to adjust their respective daily wages so that each homeowner will come out even- that is, so that the total amount paid out by each is the same as the total amount each receives. We can set

p_1= daily wage of carpenter

p_2= daily wage of electrician

p_3= daily wage of plumber

To satisfy the –equilibrium‖ condition that each homeowner comes out even, we require that

Total expenditures = total income

for each of the homeowners for the 10-day period. For example, the carpenter pays a total of $2p_1 + p_2 + 6p_3$ for the repairs in his own home and receives a total income of $10p_1$ for the repairs that he performs on all three homes. Equating these two expressions then gives the first of the following three equations:

$$2p_1 + p_2 + 6p_3 = 10p_1$$

$$4p_1 + 5p_2 + p_3 = 10p_2$$

$$4p_1 + 4p_2 + 3p_3 = 10p_3$$

The remaining two equations are the equilibrium equations for the electrician and the plumber. Dividing these equations by 10 and rewriting them in matrix from yields

$$\begin{bmatrix} .2 & .1 & .6 \\ .4 & .5 & .1 \\ .4 & .4 & .3 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \qquad \text{………………..} (1)$$

Equation 1 can be written as a homogeneous system by subtracting the left side from the right side to obtain

$$\begin{bmatrix} .8 & -.1 & -.6 \\ -.4 & .5 & -.1 \\ -.4 & -.4 & .7 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The solution of this homogeneous system is found to be

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = s \begin{bmatrix} 31 \\ 32 \\ 36 \end{bmatrix}$$

where s is an arbitrary constant. This constant is a scale factor, which the homeowners may choose for their convenience.

Leontief Open (Production) Model:

In contrast with the closed model, in which the outputs of k industries are distributed only among themselves, the open model attempts to satisfy an outside demand for the outputs. Portions of these outputs may still be distributed among the industries themselves, to keep them operating, but there is to be some excess, some net production, with which to satisfy the outside demand.

In the closed model the outputs of the industries are fixed, and our objective is to determine prices for these outputs so that the equilibrium condition, that expenditures equal incomes, is satisfied. In the open model it is the prices that are fixed, and our objective is to determine levels of the outputs of the industries needed to satisfy the outside demand. We will measure the levels of the outputs in terms of their economic values using the fixed prices. To be precise, over some fixed period of time, let

x_i=monetary value of the total output of the ith industry

d_i=monetary value of the output of the *i*th industry needed to satisfy the outside demand

c_{ij} =monetary value of the output of the *i*th industry needed by the *j*th industry to procedure one unit of monetary value of its own output

With these quantities, we define the ***production vector***

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}$$

and ***the demand vector***

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \end{bmatrix}$$

and the ***consumption matrix***

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \vdots & \vdots & & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kk} \end{bmatrix}$$

By their Nature, we have that

$$\boldsymbol{x} \geq 0, \qquad \mathbf{d} \geq 0, \qquad \text{and} \qquad C \geq 0$$

From definition of c_{ij} and x_j, it can be seen that quantity

$$c_{i1}x_1 + c_{i2}x_2 + \ldots\ldots\ldots + c_{ik}x_k$$

is the value of the output of the ith industry needed by all k industries to produce a total output specified by the production vector x. Because this quantity is simply the ith entry of the column vector

$$x - Cx$$

is the value of the excess output of the ith industry available to satisfy the outside demand of the ith industry is the entry of the demand vector **d**. Consequently, we are led to the following equation

$$x - Cx = \mathbf{d}$$

or

$$(1\text{-}C)\, x = \mathbf{d} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

for the demand to be exactly met, without any surpluses or shortages. Thus, given C and $\boldsymbol{d}$, our objective is to find a production vector $x \geq 0$ that satisfies equation 2.

Production vector for a Town

Example 2: A town has three main industries: a coal-mining operation, an electric power-generating plant, and a local railroad. To mine $ 1 of coal, the mining operation must purchase $.25 of electricity to run its equipment and $.25 of transportation for its shipping needs. To produce $1 of electricity, the generating plant requires $.65 of coal for fuel, $ .05 of its own electricity to run auxiliary equipment, and transportation. To provide $1 of transportation, the railroad requires $ .55 of coal for fuel and $.10 of electricity for its auxiliary equipment. In a certain week the coal-mining operation receives orders for $50,000 of coal from outside the town, and the generating plant receives orders for $25,000 of electricity from outside. There is no outside demand for the local railroad. How much each of the three industries produce in that week to exactly satisfy their own demand and the outside demand?

Solution:

For the one-week period let

x_1= value of total output of coal-mining operation

x_2= value of total output of power-generating plant

x_3= value of total output of local railroad

From the information supplied, the consumption matrix of the system is

$$C=\begin{bmatrix} 0 & .65 & .55 \\ .25 & .05 & .10 \\ .25 & .05 & 0 \end{bmatrix}$$

The linear (1-C) x =**d**

$$\begin{bmatrix} 1.00 & -.65 & .55 \\ -.25 & .95 & .10 \\ -.25 & -.05 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 50{,}000 \\ 25{,}000 \\ 0 \end{bmatrix}$$

The coefficient matrix on the left is invertible, and the solution is given by

$$x=(1-C)^{-1}\,\mathbf{d}=\frac{1}{503}\begin{bmatrix} 756 & 542 & 470 \\ 720 & 690 & 190 \\ 200 & 170 & 630 \end{bmatrix}=\begin{bmatrix} 102{,}087 \\ 56{,}163 \\ 28{,}330 \end{bmatrix}$$

Thus, the total output of the coal-mining operation should be \$102,087, the total output of the power-generating plant should be \$56,163, and the total output of the railroad should be \$28,330.

CONSTRUCTING CURVES AND SURFACES THROUGH SPECIFIED POINTS

Theorem: A homogeneous linear system with as many equations as unknowns has a nontrivial solution if and only if the determinant of the coefficient matrix is zero.

We shall now show how this result can be used to determine equation of various curves and surfaces through specified points.

A line through Two Points

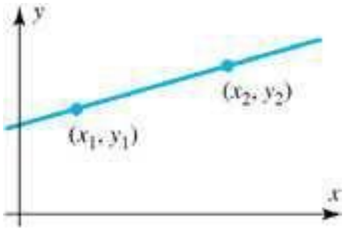
Suppose that (x_1, y_1) and (x_2, y_2) are the two distinct points in the plane. There exists a unique line

$$c_1x+c_2y+c_3=0 \qquad (1)$$

that passes through these two points (Figure 1). Note that c_1, c_2 and c_3 are not all zero and that these coefficients are unique only up to a multiplicative constant. Because (x_1, y_1) and (x_2, y_2)lie on the line, substituting them in 1 gives the two equations

$$c_1x_1+c_2y_1+c_3=0 \qquad (2)$$

$$c_1x_2+c_2y_2+c_3=0 \qquad (3)$$



Figure

The three equations 1, 2, and 3, can be grouped together and rewritten as

$$xc_1 + yc_2 + c_3 = 0$$

$$x_1c_1 + y_1c_2 + c_3 = 0$$

$$x_2c_1 + y_2c_2 + c_3 = 0$$

which is a homogeneous linear system of three equations for c_1, c_2 and c_3, Because c_1, c_2 and $c_3$are not all zero, this system has a nontrivial solution, so the determinant of the system must be zero. That is,

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0 \qquad (4)$$

Consequently, every point (x, y) on the line satisfies 4; conversely, it can be shown that every point (x, y) that satisfies 4 lies on the line.

Equation of a Line:

Example 2: Find the equation of the line that passes through the two points (2, 1) and (3, 7)

Solution: Substituting the coordinates of the two points into equation 4 gives

$$\begin{vmatrix} x & y & 1 \\ 2 & 1 & 1 \\ 3 & 7 & 1 \end{vmatrix} = 0$$

The cofactor expansion of this determinant along the first row then gives

-6x +y+11=0

A Circle through three Points:

Suppose that there are three distinct points in the plane (x_1, y_1), (x_2, y_2) and (x_3, y_3) not all lying on a straight line. From analytic geometry, we know that there is a unique, say ,

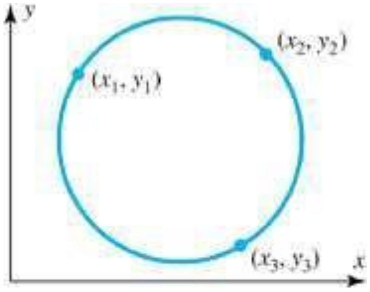
$$c_1(x^2 + y^2) + c_2x + c_3y + c_4 = 0 \qquad (5)$$

that passes through them (Figure) substituting the coordinates of three points into this equation gives

$$c_1(x_1^2+x_1^2) + c_2x_1 + c_3y_1 + c_4=0 \qquad (6)$$

$$c_1(x_2^2+x_2^2) + c_2x_2 + c_3y_2 + c_4=0 \qquad (7)$$

$$c_1(x_3^2+x_3^2) + c_2x_3 + c_3y_3 + c_4=0 \qquad (8)$$



Figure

A before equation 5 and through 8 form a homogeneous linear system with a nontrivial solution for c_1, c_2, $c_3$and c_4. Thus the determinant of the coefficient matrix is zero.

$$\begin{vmatrix} x^2+y^2 & x & y & 1 \\ x_1^2+x_1^2 & x_3 & y_1 & 1 \\ x_2^2+x_2^2 & x_2 & y_2 & 1 \\ x_3^2+x_3^2 & x_3 & y_3 & 1 \end{vmatrix} = 0 \qquad (9)$$

This is a determinant form for the equation of the circle.

Equation of a Circle

Example : Find the equation of the circle that passes through the points (1, 7), (6, 2), and

(4, 6)

Solution:

Substituting the coordinates of the three points into Equation 9 gives

$$\begin{vmatrix} x^2+y^2 & x & y & 1 \\ 50 & 1 & 7 & 1 \\ 40 & 6 & 2 & 1 \\ 52 & 4 & 6 & 1 \end{vmatrix} = 0$$

which reduces to

$10(x^2 + y^2)$ – 20x – 40y – 200=0

In standard form this is

$$(x-1)^2 + (y-2)^2 = 5^2$$

Thus, the circle has center (1, 2) and radius 5.

A General Conic Section through Five Points

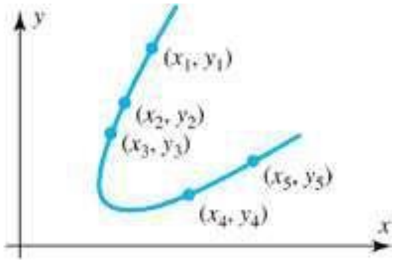
The general equation of a conic section in the plane (a parabola, hyperbola, or ellipse, or degenerate forms of these curves) is given by

$$c_1x^2 + c_2xy + c_3y^2 + c_4x+c_5y + c_6=0$$

This equation contains six coefficients, but we can reduce the number to five if we divide through by any one of them that is not zero.

Thus only five coefficients must be determined, so five distinct points in the plane are sufficient to determine the equation of the conic section (Figure). As before, the equation can be put in determinant form

$$\begin{vmatrix} x^2 & xy & y^2 & x & y & 1 \\ x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \end{vmatrix} =0$$



Equation of a Orbit:

Example : An astronomer who wants to determine the orbit of an asteroid about the sun sets up a Cartesian coordinate system in the plane of the orbit with the sun at the origin. Astronomical units of measurements are used along the axes (1 astronomical unit= mean distance of earth to sun = 93 million miles). By Kepler‘s first law, the orbit must be an ellipse, so the astronomer makes five observations of the asteroid at five different times and finds five points along the orbit be

(8.025, 8.310), (10.170, 6.355), (11.202, 3.212), (10.736, 0.375), (9.092, -2.267)

Find the equation of the orbit.

Solution:

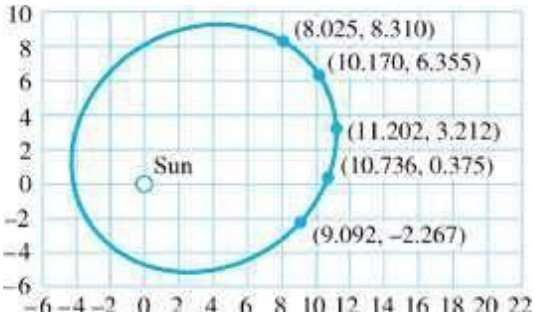
Substituting the coordinates of the five given points into 10 gives

$$\begin{vmatrix} x^2 & xy & y^2 & x & y & 1 \\ 64.401 & 66.688 & 69.056 & 8.025 & 8.310 & 1 \\ 103.429 & 64.630 & 40.386 & 10.170 & 6.355 & 1 \\ 125.485 & 35.981 & 10.317 & 11.202 & 3.212 & 1 \\ 115.262 & 4.026 & 0.141 & 10.736 & .375 & 1 \\ 82.664 & -20.612 & 5.139 & 9.092 & -2.267 & 1 \end{vmatrix} = 0$$

The cofactor expansion of this determinant along the first row is

$386.799x^2 - 102.896xy + 446.026y^2 - 2476.409x - 1427.971y - 17109.378 = 0$

Figure is an accurate of the orbit, together with the five given points.



Figure

A Plane through Three Points

The plane in 3-space with equation

$$c_1x + c_2y + c_3z + c_4=0$$

that passes through three non collinear points (x_1, y_1, z_1), (x_2, y_2, z_2), and (x_3, y_3, z_3) is given by the following determinant equations

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0$$

Equation of a plane

The equation of the plane that passes through the three noncollinear points (1, 1, 0), (2, 0, -1), and (2, 9, 2)

$$\begin{vmatrix} x & y & z & 1 \\ 1 & 1 & 0 & 1 \\ 2 & 0 & -1 & 1 \\ 2 & 9 & 2 & 1 \end{vmatrix}=0$$

which reduces to

2x-y-3z-1=0

A Sphere through Four Points

The sphere in 3-space with equation

$$c_1(x^2 + y^2 + z^2) + c_2x + c_3y + c_4z + c_5 = 0$$

that passes through four non coplanar points(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) and, (x_4, y_4, z_4) is given by the following determinant equation:

$$\begin{vmatrix} x^2 + y^2 + z^2 & x & y & z & 1 \\ x_1^2 + y_1^2 + z_1^2 & x_1 & y_1 & z_1 & 1 \\ x_2^2 + y_2^2 + z_2^2 & x_2 & y_2 & z_2 & 1 \\ x_3^2 + y_3^2 + z_3^2 & x_3 & y_3 & z_3 & 1 \\ x_4^2 + y_4^2 + z_4^2 & x_4 & y_4 & z_4 & 1 \end{vmatrix} = 0$$

Equation of a Sphere

Example :

The equation of the sphere that passes through the four points (0, 3, 2), (1, -1, 1), (2, 1, 0), and (5, 1, 3) is

$$\begin{vmatrix} x^2 + y^2 + z^2 & x & y & z & 1 \\ 13 & 0 & 3 & 2 & 1 \\ 3 & 1 & -1 & 1 & 1 \\ 5 & 2 & 1 & 0 & 1 \\ 35 & 5 & 1 & 3 & 1 \end{vmatrix} = 0$$

This reduces to

$$x^2 + y^2 + z^2 - 4x - 2y - 6z + 5 = 0$$

which in standard form is

$$(x - 2)^2 + (y - 1)^2 + (z - 3)^2 = 9.$$

Application To Chemistry:

(Balancing Chemical Equations Using Matrices)

Balancing equations means writing chemical equations such that the amount of stuff you start with in the reaction equals the amount of stuff you end up with as a product. In other words, if I start baking bread with 10 pounds of flour, I should end up with 10 pounds of bread, unless some is lost onto the floor or if some of it goes up in smoke!

Example :

A simple example goes a long way. We can form water by combing hydrogen gas (H_2) and oxygen (O_2) in the presence of electricity. The reaction looks like this

$$H_2 + O_2 \text{ ---> } H_2O$$

If you do some of the gram molecular weight calculations you will find this:

2 g of hydrogen + 32 g of oxygen = 18 g of water

What this says is that you start with 34 grams of stuff and end up with 18 grams of stuff. You've lost 16 grams of stuff, and in this reaction that just doesn't happen! Where did the 16 grams go?

They're not lost, we just haven't balanced the equation! You might have also noticed that there are two oxygens on the left and only one on the right! We need to get things in the correct proportions for this reaction to be balanced. The balanced reaction looks like this:

$$2\ H_2 + O_2 \text{ ---> } 2\ H_2O$$

This says that we need two hydrogen molecules to combine with one oxygen molecule to form two new water molecules. If we do the mathematically:

(2 x 2 g of hydrogen) + 32 g of oxygen = (2 x 18 g of water)

we now have 36 grams of stuff on the left and 36 grams on the right. We also now have 4 hydrogens on the left, four hydrogens on the right, two oxygens on the left, and two oxygens on the right. We can say that this equation is mass balanced.

Here the actual application of matrices will begin in balancing the chemical equation in chemistry.

Consider a reaction $H_2 + O_2 \text{ ---> } H_2O$ - not balanced(un balanced).

This chemical reaction consists of two elements :Hydroge(H) and Oxygen(O).

The equation to balance is identified our task is to assign the unknown co-efficients (a, b and c) to each chemical species. A balance equation can be written for each these elements:

$$a\ H_2 + b\ O_2 \text{ ---> } c\ H_2O$$

Two simultaneous linear equations in three unknown corresponding to each of these elements.then the algebraic representation of the balanced reaction is

Hydroge (H) : $2a + 0b = 2c$

Oxygen (O) : $0a + 2b = 1c$

Condition for each element is as follows:

We write these as homogeneous system of two equations, each having zero on its right hand side.

2 a + 0 b - 2 c = 0 ....(1)

0 a + 2 b -1 c = 0 ....(2)

It can be seen that these values can be placed into a matrix.by using row reduced echelon form (rref) technique to solve for the value of the coefficients.

First note that there are three unknowns , but only two equations. The system is solved by GAUSS-ELIMINATION method as follows:

$$\begin{bmatrix} 2 & 0 & -2 & : & 0 \\ 0 & 2 & -1 & : & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

Perform some basic operations on this like, $R_1 \rightarrow \frac{R1}{2}$ and $R_2 \rightarrow \frac{R2}{2}$

$$\Rightarrow \begin{bmatrix} 1 & 0 & -1 & : & 0 \\ 0 & 1 & -\frac{1}{2} & : & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

The last matrix is of reduced row echelon form, so we can stop, and we obtain that the solution of the system of linear equations is:

$\Rightarrow$ a – c = 0 therefore a = c.

b $- \frac{c}{2}$ = 0 therefore b = $\frac{c}{2}$

for instance letting , c = 1, we can represent the soltion set as

a = 1 , b = $\frac{1}{2}$ and c = 1 ; thus the balanced chemical reaction equation is

$1\ H_2 + \frac{1}{2} O_2$ ----> $1\ H_2O$,

But in the case of chemical equations, all chemical coefficients must be integer and hence, Multiplying both sides by 2(the least common multiple LCM), We obtain Solution with the positive integers:

$2\ H_2 + 1\ O_2$-----> $2\ H_2O$.

Or

To avoid the fractions, we can also let c = 2, so that

a = 2 and b = 1,Therefore, the chemical equation can be balanced as

CHAPTER-03

Coding Theory

Introduction to coding theory:

The study of error-control codes is called *coding theory*. This area of discrete applied mathematics includes the study and discovery of various coding schemes that are used to increase the number of errors that can be corrected during data transmission. Coding theory emerged following the publication of Claude Shannon's seminal 1948 paper, –A mathematical theory of communication‖. It is one of the fields that has a defined beginning.

Error control coding is only part of the processing done to messages that are to be transmitted
across a channel or stored on some medium. Figure 1 shows a flow chart that illustrates how error control coding fits in with the other stages of data processing.

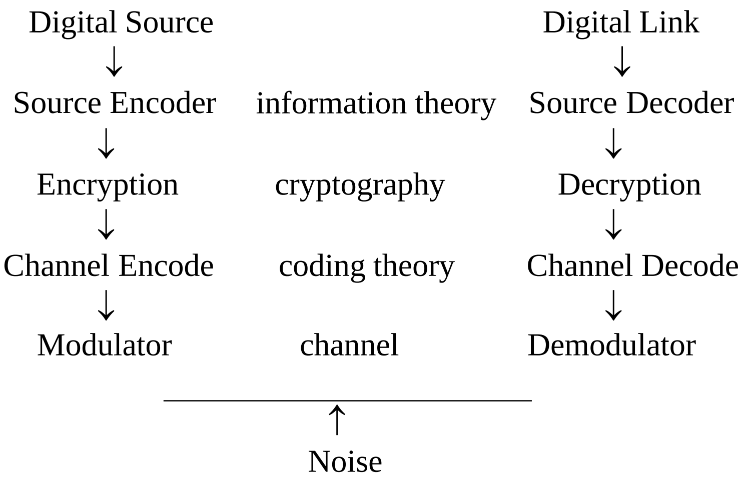


Figure 1: A Flowchart Representing the Various Stages of Data Processing

The process begins with an information source, such as a data terminal or the human voice. The source encoder transforms the source output into a sequence of symbols which we call a message m; if the information source is continuous, the source encoding involves analog-to-digital conversion. Throughout this, we usually assume that the symbols used are from the binary set {0,1} and call these symbols *bits*. If security is desired, the message would next be encrypted using a cipher, the subject of the field of cryptography. The next step is the error-control coding, also called channel coding, which involves introducing controlled redundancy into the message m. The output is a string of discrete symbols (usually binary in this book) which we call a codeword c. Next, the modulator transforms each discrete symbol in the codeword into a wavelength to be transmitted across the channel. The transmission is subject to noise of various types, and then the processes are reversed.

Historically, decreasing the error rates in data transmissions was achieved by increasing the power of the transmission. However, this is an inefficient and costly use of power. Moreover, increasing the power only works well on certain channels such as binary symmetric channels; other channels, such as fading channels, present additional challenges.

Example 1: (*The ISBN Code*) The International Standard Book Number (ISBN) Code is
used throughout the world by publishers to identify properties of each book. The first nine digits (bounded between 0 and 9, inclusive) of each ISBN represent information about the book including its language, publisher, and title. In order to guard against errors, the nine-digit ―message‖ is encoded as a ten-digit codeword. The appended tenth digit is a *check digit* chosen so that the whole ten-digit string $x_1, x_2, ..., x_{10}$ satisfies

$$\sum_{i=1}^{10} ix_i \equiv 0 \quad \text{(mod 11)}.$$

If x_{10} should be equal to 10, an `X' is used.

The ISBN code can detect any single error and any double-error created by the transposition of two digits. A great course project would be to figure out why this code has this error-detecting capability and compare this with other check-digit schemes used on airplane tickets, in bank numbers on checks, in credit card numbers, in the Universal Product Code (UPC) found on groceries, etc.

Example 2: (*The Repetition Code*) This example is a simple error-correcting code. Suppose we would like to send a 1 to signify ―yes‖, and a 0 to signify ―no‖. If we simply send one bit, then there is a chance that noise will corrupt that bit and an unintended message will be received. A simple alternative is to use a *repetition code*. Instead of sending a single bit, we send 11111 to represent 1 (or yes), and 00000 to represent 0 (or no). Since the code words consist of 0s and 1s, this is called a *binary* code. We say that the code's alphabet is the set {0,1}, with all arithmetic done modulo 2. Alternatively, we can say that this code alphabet is the *finite field* with two elements, *GF*(2).

Under certain reasonable assumptions about the channel in use, the receiver decodes a received 5-tuple using a ―majority vote‖. The received 5-tuple is decoded as the bit that occurs most frequently. This way, if zero, one, or two errors occur, we will still decode the received 5-tuple as the intended message. In other words, the receiver decodes a received 5-tuple as the ―closest' codeword. This is an example of *nearest neighbor decoding*.

Important Code Parameters:

When developing codes, there must be a way to decide which codes are ―good‖ codes. There are three main parameters used to describe and evaluate codes. The first parameter is the *code length*, *n*. In the repetition code of Example 2, the code length is 5, since the codewords 00000 and 11111 each contain 5 bits. In this book, we restrict our discussion to *block* codes, which are codes whose codewords are all of the same length. Since error-control codes build redundancy into the messages, the code length, *n*, is always greater than the original message length, *k*. The next parameter that we consider is the *total number of codewords*, *M*. In Example 2, the total number of codewords is 2. The third parameter measures the *distance* between pairs of codewords in a code. In order to explain clearly the notion of distance between codewords.

Definition : The *Hamming weight* $w(\mathrm{c})$ of a codeword c is the number of nonzero components in the codeword.

Example : $w(00000) = 0$, $w(11111) = 5$, $w(1022001) = 4$, $w(1011001) = 4$.
The Guava function WeightCodeword(C) can be used to determine the weight of a given codeword:

```
gap>  WeightCodeword(Codeword("1011001"));4
```

Definition : The *Hamming distance* between two codewords $d(\mathrm{x}; \mathrm{y})$ is the number of places in which the codewords x and y differ. In other words, $d(\mathrm{x}; \mathrm{y})$ is the Hamming weight of the vector x ¡ y, representing the component-wise difference of the vectors x and y.

Example : $d(0001; 1110) = 4$, since the two codewords differ in all four positions. $d(1202; 1211) =2$, since the two codewords differ in the last two positions.
The Guava function DistanceCodeword will return the Hamming distance of two codewords. For **example:**

```
gap> a := Codeword("01111", GF(2));;

gap> b := Codeword("11000",GF(2));;

gap> DistanceCodeword(a,b); 4
```

Definition : The *minimum (Hamming) distance* of a code C is the minimum distance between any two codewords in the code: $d(C) = \min\{d(\mathrm{x}; \mathrm{y}) / \mathrm{x} \neq \mathrm{y}, \mathrm{x},\mathrm{y} \in C\}$.

Example: The binary repetition code of length 5 has minimum distance 5 since the two codewords differ in all 5 positions.

Definition: The Hamming distance d is a *metric* on the space of all q-ary n-tuples which means that d satisfeis the following properties for any q-ary n-tuples $\mathrm{x}; \mathrm{y}; \mathrm{z}$:

1. $d(\mathrm{x}; \mathrm{y}) \geq 0$, with equality if and only if $\mathrm{x} = \mathrm{y}$.
2. $d(\mathrm{x}; \mathrm{y}) = d(\mathrm{y}; \mathrm{x})$
3. $d(\mathrm{x}; \mathrm{y}) \leq d(\mathrm{x}; \mathrm{z}) + d(\mathrm{z}; \mathrm{y})$

The third property above is called the *triangle inequality*, which should look familiar from Euclidean geometry. The notation (n,M, d) is used to represent a code with code length *n*, a total of *M* codewords, and minimum distance *d*. One of the major goals of coding theory is to develop codes that strike a balance between having small *n* (for fast transmission of messages), large *M* (to enable transmission of a wide variety of messages), and large *d* (to detect many errors).

Example: Let *C* = {0000*;* 1100*;* 0011*;* 1111}. Then *C* is a (4, 4, 2) binary code.
It is possible to construct a code in Gap by listing all of its codewords. To do this, we can use the function ElementsCode.

For example:

```
gap> C := ElementsCode(["0000", "1100", "0011", "1111"], GF(2));;
```

constructs the code given in above Example

If we need help determining the key parameters for a code, we can use built-in functions such as:

```
gap> n := WordLength(C);4
gap> M := Size(C);4
gap> d := MinimumDistance(C);2
```

The final important code parameter is a measure of effciency:

Correcting and Detecting Errors:

Talking about the number of errors in a received codeword is equivalent to talking about the distance between the received word and the transmitted word.
Suppose a codeword $c = c_0 c_1 ... c_{n-1}$ is sent through a channel and the received vector is $r = r_0 r_1 ... r_{n-1}$. The error vector is defined as e=r-c= $e = e_0 e_1 ... e_{n-1}$. The job of the decoder is to decide which codeword was most likely transmitted, or equivalently, decide which error vector most likely occurred. Many codes use a *nearest neighbor decoding scheme* which chooses the codeword that minimizes the distance between the received vector and possible transmitted vectors. For example, the \majority vote" scheme described for our binary repetition code is an example of a nearest neighbor decoding scheme. A nearest neighbor decoding scheme for a *q*-ary code maximizes the decoder's likelihood of correcting errors provided the following assumptions are made about the channel:
1. Each symbol transmitted has the same probability *p* (< 1⁄2) of being received in error (and
1 - *p* of being received correctly)

2. If a symbol is received in error, that each of the *q* - 1 possible errors is equally likely.
Such a channel is called a *q*-ary *symmetric channel*, and we assume throughout this book that the channels involved are symmetric. This implies that error vectors of lower weight will occur with higher *probability* than error vectors of higher weight.

When analyzing codes, we often need to talk about *probabilities* of certain events occurring.

For example, we need to compare the probabilities of the occurrences of lower weight errors versus higher weight errors, and we need to ascertain the probability that a codeword will be received in error. Informally, the probability that an event occurs is a measure of the likelihood of the event occurring. Probabilities are numbers between 0 and 1, inclusive, that reflect the chances of an event occurring.

A probability near 1 reflects that the event is very likely to occur, while a probability near 0 reflects that the event is very unlikely to occur. When you flip a fair coin, there is a probability of 1⁄2 that you will get tails and a probability of 1⁄2 that you will get heads.

But what about the probability of getting three heads in three flips of a fair coin? Since each of these events is independent, meaning that each of the three flips does not depend on the other flips, we can use the *multiplicative property* of probabilities: The probability of getting three heads in the three coin tosses is $\frac{1}{2}.\frac{1}{2}.\frac{1}{2}=\frac{1}{8}$.

Example : Given the assumptions of a *q*-ary symmetric channel with symbol error probability *p*, consider the probability that no errors will occur when transmitting a *q*-ary codeword of length *n*. Since each symbol has probability (1-*p*) of being received correctly, the desired probability is $(1-p)^n$.

Theorem 1:1. A code *C* can detect up to *s* errors in any codeword if $d(C) \geq s + 1$.

2. A code *C* can correct up to *t* errors in any codeword if $d(C) \geq 2t + 1$.

Proof:1. Suppose $d(C) \geq s+1$. Suppose a codeword c is transmitted and that *s* or fewer errors occur during the transmission. Then, the received word cannot be a different codeword, since all codewords differ from c in at least $s + 1$ places. Hence, the errors are detected.

2. Suppose $d(C) \geq 2t + 1$. Suppose a codeword x is transmitted and that the received word, r, contains *t* or fewer errors. Then $d(x, r) \leq t$. Let x$'$ be any codeword other than x. Then $d(x', r) \geq t+1$, since otherwise $d(x', r) \leq t$ which implies that $d(x, x0) \cdot\leq d(x,r)+d(x', r) \leq 2t$ (by the triangle inequality), which is impossible since $d(C) \geq 2t + 1$. So x is the nearest codeword to r, and r is decoded correctly.

The remainder of this section will involve using error-correcting capabilities and code rate to compare two different codes proposed for the transmission of photographs from deep space.

Sphere-Packing Bound:

We know from Theorem that if C has minimum distance $d \geq 2t+1$, then C can correct at least t errors. We also say that C can correct all error vectors of weight t. We now develop a geometric interpretation of this result. Since each codeword in C is at least distance $2t + 1$ from any other codeword, we can picture each codeword c to be surrounded by a sphere of radius t such that the spheres are disjoint (non-overlapping). Any received vector that lies within the sphere centered at c will be decoded as c. This pictorially explains why received vectors that contain t or fewer errors are decoded correctly: They still lie within the correct sphere. We can use this picture to bound the total possible number of codewords in a code, as seen in the following theorem:

Theorem 2: (*Sphere-packing bound*) A t-error-correcting q-ary code of length n must satisfy

$$M\sum_{i=0}^{t}\binom{n}{i}(q-1)^{i} \leq q^{n}$$

In order to prove Theorem 2, we need the following lemma.

Lemma : A sphere of radius r, $0 \leq r \leq n$, in the space of all q-ary n-tuples contains exactly

$$\sum_{i=0}^{r}\binom{n}{i}(q-1)^{i} = \binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^{2}+\ldots\binom{n}{r}(q-1)^{r}$$

vectors.

Proof of Lemma*:* Let u be a fixed vector in the space of all q-ary n-tuples. Consider how many vectors v have distance exactly m from u, where $m \leq n$. The m positions in which v is to differ from u can be chosen in $\binom{n}{m}$ ways, and then in each of these m positions the entry of v can be chosen in q- 1 ways to differ from the corresponding entry of u. Hence, the number of vectors at distance exactly m from u is $\binom{n}{m}$ $(q - 1)m$.

A ball of radius r centered at u contains vectors whose distance from u ranges from 0 to r. So, the total number of vectors in a ball of radius r, centered at u, must be $\binom{n}{0}+\binom{n}{}(q-1)+\binom{n}{}(q-1)^{2}+\ldots\binom{n}{}(q-1)^{r}$.

Proof of Theorem *:* Suppose C is a t-error-correcting q-ary code of length n. Then, in order

that C can correct t errors, any two spheres of radius t centered on distinct codewords can have no vectors in common. Hence, the total number of vectors in the M spheres of radius t centered on the M codewords of C is given by $\sum_{i=0}^{t}\binom{n}{i}(q-1)^{i}$ by Lemma.

This number of vectors must be less than or equal to the total number of vectors in the space of all q-ary n-tuples, which is q^{n}.

This proves the sphere-packing bound.

Introduction to Parity Check Matrices:

Definition : Let C be an $[n, k]$ linear code. A *parity check matrix* for C is an $(n - k) \times n$ matrix H such that $c \in C$ if and only if $cH^T = 0$.

Example: If C is the code {111, 000}, then a valid parity check matrix is

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

One way to verify this is to compute the product cH^T for an arbitrary codeword c = $[x, y, z]$:

$$[xyz]\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} = [x + z, y + z]$$

So, $c = [xyz]$ is a valid codeword iff $cH^T = 0$, ie. Iff $[y + z, x + z] = [0, 0]$. Algebraic manipulations over binary show that $x = y = z$. Thus the only valid codewords are indeed {000, 111}. Notice that this is the binary repetition code of length 3.

Example: Find a parity check matrix for the (5, 4) binary parity check code, which encodes messages of length 4 to codewords of length 5 by appending a parity check bit.

The Guava function CheckMat generates a parity check matrix for any given linear code.
For example, we can generate a check matrix for a repetition code of length 5 as follows:

```
gap> C := RepetitionCode(5, GF(2));;
gap> H := CheckMat(C);;
gap>Display(H);
1  1  .  .  .
.  1 1  .  .
.  . 1  1 .
.  .  . 1 1
```

Example: Show that any codeword c in the binary repetition code of length 5 indeed satisfies the equation cH^T , where H is given by the Gap output above.

The connection drawn above between the codewords of a code C and the parity check matrix of a code C can be strengthened by considering the nullspace of the parity check matrix.

Definition : The *nullspace* of a matrix H is the set of all vectors x such that $Hx^T = 0$.

In linear algebra, we study nullspaces of matrices, often in the context of finding the *kernel* of linear mappings. However, there is a simple connection to linear codes. Given the parity check matrix *H* for a code *C*, we have 0 = cH^T for all c *2 C*, hence 0^T = (cH^T $)^T$ = *H*c^T . Therefore, the codewords of *C* comprise the nullspace of its parity check matrix *H*.

Definition : Two *q*-ary linear codes are called *equivalent* if one can be obtained from the other by a combination of operations of the following types:

1. Permutation of the positions of the codewords
2. Multiplication of the symbols appearing in a fixed position by a nonzero scalar (*i.e.* element of *GF*(*q*)).

Example : The two codes C_1 and C_2 below are equivalent since the second is merely the permutation of the first two positions in the first code.

$$C_1 = \left\{\begin{matrix} 000 \\ 101 \\ 010 \\ 111 \end{matrix}\right\} \qquad C_2 = \left\{\begin{matrix} 000 \\ 011 \\ 100 \\ 111 \end{matrix}\right\}$$

Theorem : Two $k \times n$ matrices generate equivalent [*n*, *k*] linear codes over *GF*(*q*) if one matrix can be obtained from the other by a sequence of operations of the following types:

1. Permutation of the rows
2. Multiplication of a row by a nonzero scalar
3. Addition of a scalar multiple of one row to another
4. Permutation of the columns
5. Multiplication of any column by a nonzero scalar.

Proof. The first three types of operations preserve the linear independence of the rows of a generator matrix and simply replace one basis of the code by another basis of the same code. The last two types of operations convert a generator matrix for one code into a generator matrix for an equivalent code.

Gap can be used to put parity check matrices and generator matrices into systematic or standard form. It is necessary to remember which standard form requires the identity matrix on the left. For our convention, we will put the identity on the right in generator matrices and on the left for parity check matrices.

For example, suppose I have this check parity matrix

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

We can represent this check matrix in a systematic form by using the PutStandardForm command which inputs two values, the matrix and a boolean variable, true or false. PutStandardForm(H, true) will arrange the input matrix *H* so that the identity matrix is on the left. PutStandardForm(H, false) will arrange the input matrix *H* so that the identity matrix is on the right.

```
gap> H := [[1,0,1,0], [0, 1, 1, 0], [0, 0, 0, 1]];

gap> Display(H);
[ [ 1, 0, 1, 0 ],
[ 0, 1, 1, 0 ],
[ 0, 0, 0, 1 ] ]
gap> PutStandardForm(H, true);;

gap> Display(H);
[ [ 1, 0, 0, 1 ],
[ 0, 1, 0, 1 ],
[ 0, 0, 1, 0 ] ]
```

Note that the (3,4) output indicates that the arranged matrix is 3£4, and then we use the Display command to view the standard form of *H*.

Hamming Codes:

The fact that the syndrome of a received vector is equal to the sum of the columns of the parity check matrix *H* where errors occurred gives us some insight about how to construct a parity check matrix for a binary code that will undergo coset or syndrome decoding. First, the columns of *H* should all be nonzero, since otherwise an error in the corresponding position would not affect the syndrome and would not be detected by the syndrome decoder. Second, the columns of *H* should be distinct, since if two columns were equal, then errors in those two positions would be indistinguishable. We use these observations to build a family of codes known as the binary *Hamming codes*, H_r, $r \geq 2$. Any parity check matrix for the Hamming code H_r has r rows, which implies that each column of the matrix has length r. There are precisely 2^r - 1 nonzero binary vectors of length r, and in order to construct a parity check matrix of H_r, we use all of these 2^r - 1 column vectors.

Definition : A binary *Hamming code* H_r of length $n = 2^r -1$, $r \geq 2$, has parity check matrix
H whose columns consist of all nonzero binary vectors of length r, each used once. This gives an $[n = 2^r-1, k = 2^r - 1 – r, d = 3]$ linear code.

Example : How many errors can a Hamming code correct?

One parity check matrix for the binary [7,4, 3] Hamming code H_3, where columns are taken in the natural order of increasing binary numbers is as follows:

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

We now rearrange the columns to get a parity check matrix *H* in standard form:

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

A generator matrix *G* in standard form is:

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

Example :Encode the messages 0000 and 1010 using *G*. Check that the resulting codewords are valid by using *H*.

Guava has built-in functions for Hamming codes. The [7, 4, 3] binary Hamming code is defined by the parameter *r* = 3, and is generated as follows:

```
gap>c:=Hamming code (3,GF(2));
a linear [7,4,3]1 Hamming (3,2) code over GF(2).
```

To view the full list of codewords in the [7, 4, 3] binary Hamming code, we use the following command:

```
gap>cw:=AsSortedList (c);
[[0000000],  [0001111],  [0010110],  [0011001],
 [0100101],  [0101010],  [0110011],  [0111100],
 [1000011],  [1001100],  [1010101],  [1011010],
 [1100110],  [1101001],  [1110000],  [1111111]]
```

Notice that the output is sorted. The list returned is also immutable, meaning that elements of the list cannot be changed. It will sometimes be useful to be able to access the elements in the outputted list. For example, in what follows, we will add together two of the codewords in the list and check that their sum is also a codeword (guaranteed by the linearity of the code):

```
gap> d := cw[2] + cw[3],
[0 0 1 1 0 0 1]

gap> d in cw;
true
```

Example. Write down a parity check matrix for the binary Hamming code with *r* = 4 by using the mathematical definition of Hamming codes. Check your answer with Gap.

It is easy to decode Hamming codes, which are used when we expect to have zero or one error per codeword. If we receive the vector r, compute the syndrome *S*(r). If *S*(r) = 0, then assume that r was the codeword sent. If *S*(r) ≠ 0, then, assuming a single error, *S*(r) is equal to the column of *H* that corresponds to the coordinate of r where the error occurred. Find the column of *H* that matches *S*(r), and then correct the corresponding coordinate of r. This decoding scheme is further simplified if the columns of *H* are arranged in order of increasing binary numbers.