Chap. 1

Learning about Vectors and their Uses.

1. A vector can be represented by a segment of a straight line. It has a magnitude and a direction. Velocity, for example, is a vector in that it involves a magnitude, its speed, and a direction. Other vectors, such as acceleration, force, stress, electric current, etc have two parts: a magnitude and direction. A scalar, on the other hand, has magnitude only. Speed, the magnitude of a velocity, is a scalar. Some other scalars are: temperature, calories, or the quantity of any thing, numbers, as 3, 4, 5 , etc. Linear programming is involved largely with vectors.

2. Representation of a vector. Any vector as a may be graphically represented by an arrow:

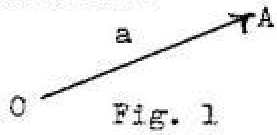

Fig. 1

Point O is its origin and A is its terminus.

3. Negative Vectors. The vector having the same magnitude as a but the opposite direction is the negative of vector a. They are correctly represented below:

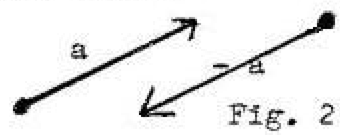

Fig. 2

These vectors do not have to have the same origin. All vectors, whatever their origins, having the same magnitude and sense are equal vectors.

4. Unit Vectors. The vector having the same sense as vector a and a magnitude 1 is a unit vector in the direction of a and we write it as a^1. We may write the vector a as:

(1) $$a = a_0 a^1$$

where a_0 is the magnitude of vector a. If our vector is a composite one as (b + c) it is written:

(2) $$(b + c) = (b + c)_0 (b + c)^1$$

5. Reciprocal Vectors. The vector with the same sense as vector a with a magnitude equal to the reciprocal of that of a is the reciprocal of a. It is written:

(1) $$a^1/a_0$$

6. An Extended Meaning of Equality

Consider two vectors a and b diagramed below. We assume that a starts from point O and that its terminus is at point B and that b starts from point B the terminus of a and has its terminus at point C. We can go from O directly to C or we can go from O to C by going to B then to C. In both cases we arrive at the same point C. Let us call the vector from O to C small c. We may then write:

$$(1) \qquad c = a + b .$$

This equation means the going along vector **c** is the same as the going along vector a added to the going along vector b for in esch case we arrive at the same point C.

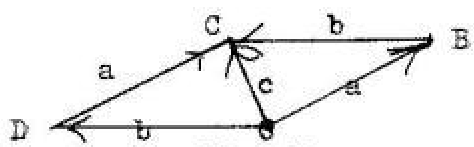

Fig. 3

Equation (1) may be written :

$$(2) \qquad c = b + a .$$

This is obvious from **Fig. 3.** Our equations, without otherwise stated, will have this extended meaning of the equality sign = .

7. Orthonormal Vectors. The orthonormal vector to vector a is written:

$$\breve{a}$$

and is sketched below.

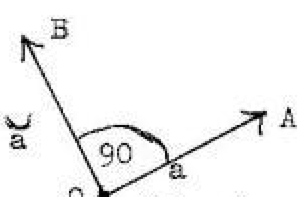

Fig. 4

It is equal to a in magnitude and makes a 90^0 angle with vector a and points in a counter clock wise sense. One can write

$$(1) \qquad a_0 = a_0$$

The orthonormal of vector (a + b) :

$$(2) \qquad (a + b)^{\smile} = \breve{a} + \breve{b} = (a + b)^{\smile}$$

and one may also write:

$$(3) \qquad (a + b)^{\smile}_0 = (a + b)_0$$

Equation (2) means that a composite orthonormal vector may be decomposed into its constituents as orthonormals, or that a sum of orthonormals may be grouped into a single orthonormal.

We shall not take time here to prove this fact but will illustrate it with some examples which will not prove it but will give credence to the fact. For simplicity we go with two vectors a and b in the plane.

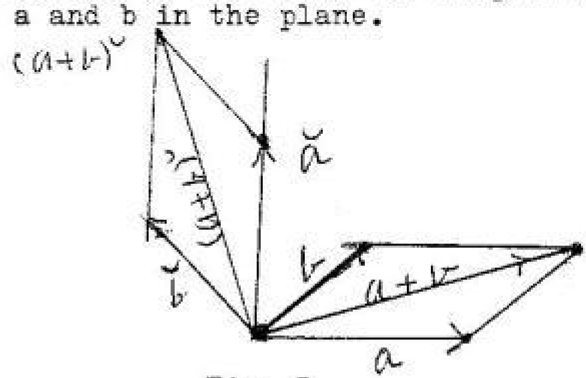

Fig. 5

We draw the orthonormal (a + b) to (a + b) and consider the new configuration as a rigid rotation thru 90 degrees of the original configuration and a rigid rotation conserves angles between the constituents; thus

$$(4) \qquad \breve{a} + \breve{b} = \breve{(a + b)}.$$

which is the same as equation (2) .

If one multiplies a vector a by a scalar n the result is a vector with the same direction as a. It may be written

$$(5) \qquad b = n a$$

Fig. 6

Fig. 6 is the sketch for n greater than 1. When n is less than 1 point B will fall between O and A. when n is negative point B will fall to the left of O.

Suppose we apply our operator ˘ twice to vector a:

$$\breve{\breve{a}} = \breve{a}^{2} = - a$$

A sketch will make this obvious or it follows from Fig. 5. Also:

$$\breve{a}^{3} = - \breve{a}$$

$$\breve{a}^{4} = a$$

8. Components of a vector. The vector a may be written:

$$(1) \qquad a = a_1 + a_2$$

Fig. 7

Here a is said to be a two dimensional vector. a_1 ane a_2 are its component vectors. For three dimensions one could write:

$$(2) \quad a = a_1 + a_2 + a_3$$

and for n dimensions one could write:

$$(3) \quad a = a_1 + a_2 + a_3 + .. a_n .$$

It is fashionable to consider a set of unit vectors mutually perpendicular to each other then one would write

$$(4) \quad a = a_1 i_1 + .. a_n i_n$$

where now the a_i are scalar components of vector a. From equation (4) we get:

$$(5) \quad \breve{a} = a_1 \breve{i}_1 + .. a_n \breve{i}_n .$$

In linear programming, in the styling of the New Science of Mutation Geometry, one makes use of such equations as (5) in dealing with polyhedrons of many dimensions. It will simply matters as we shall see. The old simplex is too combersome, with all its baggage of slack variables, expansion bases, degeneracy, cycling, and the like. Mutation Geometry offers the hope for a better day.

Henceforth we shall consider our components as rectangular. With this convention we shall omit the unit vectors from our equations. It greatly simplifies the notation and no confusion need arise. We add, subtract, and multiply corresponding components. For example we add two vectors a and b and get:

$$a = 2 + 3$$

$$b = 3 + 4$$

$$a + b = 5 + 7.$$

For hyper vectors :

$$a = 2 + 3 + 1 + 5 + 7 + 6$$

$$b = 4 + 2 + 6 + 1 + 2 - 3$$

$$a + b = 6 + 5 + 7 + 6 + 9 + 3.$$

In linear programming we shall largely be dealing with hyper vectors.

9. Scalar Product of two Vectors. For two vectors a and b:

$$a = a_1 + a_2$$

$$b = b_1 + b_2$$

$$(1) \quad P = a \cdot b = a_1 b_1 + a_2 b_2.$$

Numerically:

$$a = 2 + 3$$

$$b = 3 + 5$$

$$P = a \cdot b = 6 + 15 = 21$$

Note that 6, 15, and 21 are all scalars. Note also

$$a = 2 + 3$$

$$\breve{a} = -3 + 2$$

$$P = a \cdot \breve{a} = -6 + 6 = 0.$$ This is as it
s
should be.

10. Vector Difference The difference between two vectors a and b will be written:

$$(1) \qquad c = a - b.$$

For the diagram of its configuration see the sketch below.

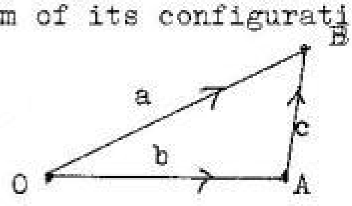

Fig. 8

We know from previous theory that

$$b + c = a$$

Transposing b we have

$$c = a - b.$$

In words the difference of a and b is the vector joining their termini and pointing from b to a. From the sketch below one sees the sum and difference of two vectors. The diagonals of the parallelogram construvted on a and b represents their sum and difference.

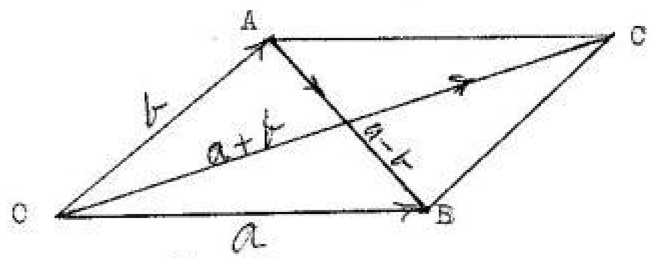

Fig. 9

11. Equation of a Straight Line. In analytic geometry the equation of a straight line may be written:

(1) $$a_1 x_1 + a_2 x_2 = b$$

This may be facrored into:

((2) $$a . r = b.$$

$$a = a_1 + a_2$$

$$r = x_1 + x_2$$

We may write (2) in the form:

(3) $$a^1 . r = (b / a_0) = p$$

Here b is a scalar and a_0 is the magnitude of vector a. p is the perpendicular distance from the origin to the line. See the sketch below.

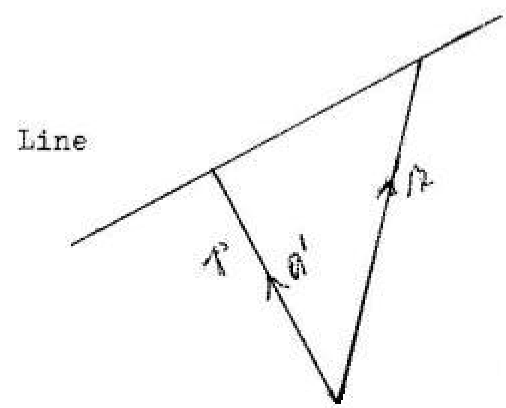

Fig. 10

The projection of r on a^1, a unit vector perpendicular to the line, is a constant p no matter where the terminus of r is on the given line.

12. Equation of a line thru a given point parallel to a given direction. Let a be the vector to the given point and vector c the given direction. See the sketch below.

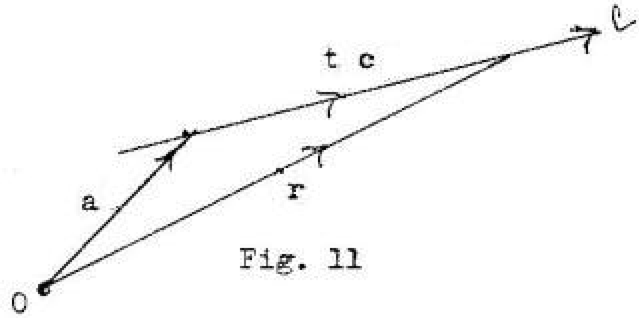

Fig. 11

From Fig. 11 we may write the equation:

(1) $r = a + t\,c$

where t ia a scalar multiplier. This last equation is very usful in linear programming Mutation - wise. Eliminating t from the last equation we get anothe form of of the equation:

(2) $c \cdot r = c \cdot a$

The equation of a hyperplane may be written:

(3) $a \cdot r = b$

where a is the normal to the hyperplane and r is a vector whose terminus is somewher in the hyperplane.

$$a = a_1 + \dots a_n$$

$$r = x_1 + \dots x_n$$

The faces of the constraint polyhedrons in linear programming are hyperplanes. Ones success there depends on how well one can manipulate these faces.

13. Solution of n linear equations in n unknowns. Consider a system of the form:

(1) $A \cdot r = b$

Here A is the matrix of the system of coefficients of the x_i and $r = x_1 + x_2 + \dots x_n$ is the n dimensional vector of the unknowns and b is an n dimensional column vector. From equation (1) we get:

(2) $r = A^{-1} \cdot b$

In conventional linear programming, using the simplex scheme, finding the inverse of a large matrix often presents a problem.

In linear programming, using the new science of Mutation

Geometry, one has no need for the inverse of a matrix for there is no expansion of vectors in terms of bases and hence no possibility of degeneracy . No slack variables are needed .

With the vexations of degeneracy, expansion bases, slack variables, cycling and other complicating factors, inherent in the simplex formulation, not present, Mutation Geometry should be able to put linear programming on a more satisfactory foundation.