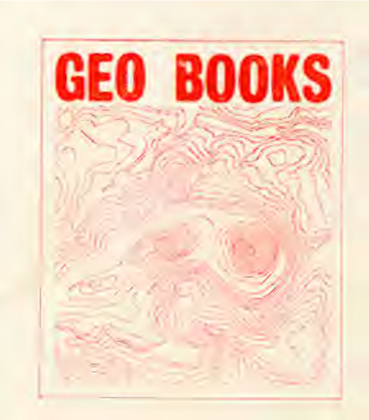


ISSN 0306 614 2
ISBN 0 86094 224 4

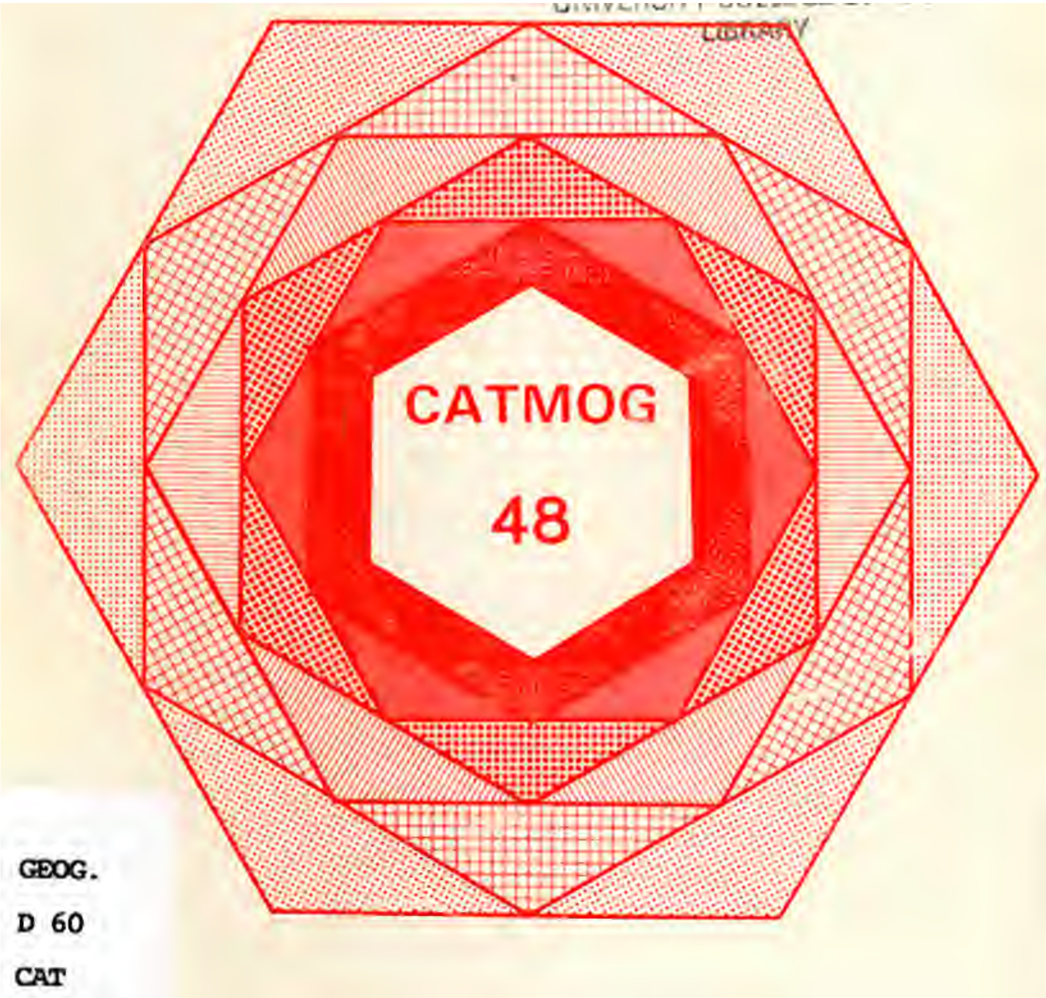
Published by Geo Books, Norwich

Printed by W,H. Hutchins & Sons, Norwich

INTRODUCTORY

MATRIX ALGEBRA

KEITH TINKLER



CATMOG **- Concepts and Techniques in Modern Geography**

CATMOG has been created to fill in a teaching need in the field of quantitative methods in undergraduate geography courses. These texts are admirable guides for teachers, yet cheap enough for student purchase as the basis of classwork. Each book is written by an author currently working with the technique or concept he describes.

1.	Introduction to Markov chain analysis	- L. Collins
2.	Distance decay in spatial interactions	- P.J. Taylor
3.	Understanding canonical correlation analysis	- D. Clark
4.	Some theoretical and applied aspects of spatial interaction shopping models	- S. Openshaw
5.	An introduction to trend surface analysis	- D. Unwin
6.	Classification in geography	- R.J. Johnston
7.	An introduction to factor analysis	- J.B. Goddard & A. Kirby
8.	Principal components analysis	- S. Daultrey
9.	Causal inferences from dichotomous variables	- N. Davidson
10.	Introduction to the use of logit models in geography	- N. Wrigley
11.	Linear programming: elementary geographical applications of the transportation problem	- A. Hay
12.	An introduction to quadrat analysis (2nd edition)	- R.W. Thomas
13.	An introduction to time-geography	N.J. Thrift
14.	An introduction to graph theoretical methods in geography	- K.J. Tinkler
15.	Linear regression in geography	- R. Ferguson
16.	Probability surface mapping. An introduction with examples and FORTRAN programs	- N. Wrigley
17.	Sampling methods for geographical research	- C.J. Dixon & B. Leach
18.	Questionnaires and interviews in geographical research	- C.J. Dixon & B. Leach
19.	Analysis of frequency distributions	- V. Gardiner & G. Gardiner
20.	Analysis of covariance and comparison of regression lines	- J. Silk
21.	An introduction to the use of simultaneous-equation regression analysis in geography	- D. Todd
22.	Transfer function modelling: relationship between time series variables	- Pong-wai Lai
43.	Stochastic processes in one dimensional series: an introduction	- K.S. Richards
24.	Linear programming: the Simplex method with geographical applications	- James E. Killen
25.	Directional statistics	- G.L. Gaile & J.E. Burt
26.	Potential models in human geography	- D.C. Rich
27.	Causal modelling: the Simon-Blalock approach	- D.G. Pringle
28.	Statistical forecasting	- R.J. Bennett
29.	The British Census	- J.C. Dewdney
30.	The analysis of variance	3. Silk
31.	Information statistics in geography	- R.W. Thomas
32.	Centrographic measures in geography	- A. Kellerman
33.	An introduction to dimensional analysis for geographers	- R. Haynes
34.	An introduction to Q-analysis	- J. Beaumont & A. Gatrell

(continued inside back cover)

ACKNOWLEDGEMENTS

I am grateful to D.J.Unwin who initiated this CATMOG during his editorship, and to A.C.Gatrell who has spent so long waiting for the finished product, who provided Appendix 2 on the use of MINITAB, and who proofread the final version.

Steven Rusk used the first half of an earlier draft and uncovered a number of errors, and an anonymous referee made several useful suggestions. Loris Gasporotto drew the diagrams and Brock University provided the computational facilities that enabled me to produce the text.

If the reader perceives aesthetic qualities in the pages which follow they are entirely due to the remarkable software of the GUTENBERG word processing program for APPLE)(series computers and the Imagewriter printer.

This CATMOG differs from most others in that it attempts to review the basic elements of a branch of mathematics that has wide applicability to problems encountered in geographical analysis. Several existing CATMOGs utilise matrix algebra to a greater or lesser extent, and this introduction should give the student a good basis for tackling more advanced topics in these and other texts. It is not possible to use a notation that is consistent with all other users; the student will have to take care to become familiar with that used by the authors being followed (I have changed my own use from the rounded brackets in CATMOG 14 to the square brackets more suited to word processing systems). Where possible I have tried to indicate different usages, and in several cases I have taken examples from other CATMOGs to rework here in the course of illustrating specific techniques.

I INTRODUCTION

(i) MATRIX ALGEBRA AND ITS USES

Just as algebra allows the manipulation of the entities that are usually called numbers, so matrix algebra permits the manipulation of whole collections, or tables of, numbers: the entities called matrices. Matrix Algebra provides a very compact way to express large numbers of linear equations and in consequence is often termed Linear Algebra. A virtue of matrix notation is that it is independent of the number of elements in the matrices, and so a solution to one size of a problem is the solution of any size. Of course the labour of solution still has to be undertaken, usually by computer, but any size of problem has the same formal solution. It will often happen that apparently very different problems have an identical matrix structure and so are solved by identical methods.

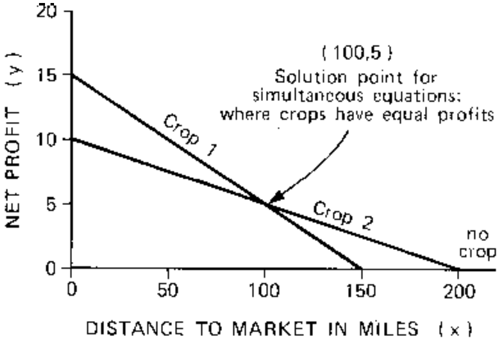


Figure 1

Two crop von Thunen model

As a motivation consider a simple two-crop Thunen model (Figure 1) represented by the simultaneous equations:

$$\begin{array}{lllll} y = 15 - 0.1x & \text{(crop 1)} & & \\ y = 10 - 0.05x & \text{(crop 2)} & \text{....} & \text{Eq 1} \end{array}$$

where the y's represent the net profits after the deduction of all the 'on the farm' costs and the cost of transporting the crop x miles to market at the rate per mile given by the coefficient attaching to the x. We would like to know the distance to the point at which the crops have identical profits, where both equations are simultaneously true. First rearrange the equations to get:

$$\begin{array}{lll} 15 = y + 0.1x & & \\ 10 = y + 0.05x & \text{.....} & \text{Eq 2} \end{array}$$

The coefficients (the numbers) in these equations can be written as follows, if the positions of y and x are taken as understood:

$$\begin{array}{llll} +15 & = & +1 & +0.10 \\ +10 & = & +1 & +0.05 \end{array}$$

Notice that the + signs attach merely to the numbers, they don't combine them as 'operators as they did in the equation form. Usually we will leave + signs as blanks. To indicate that the y and x have to be combined with these numbers in a regular way the whole problem is now written in this fashion in matrix algebra:

$$\begin{bmatrix} 15 \\ 10 \end{bmatrix} = \begin{bmatrix} 1 & 0.10 \\ 1 & 0.05 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} \quad \text{.....} \quad \text{Eq 3}$$

or $[s] = [A][x]$

where in the expanded version each set of large brackets is a **matrix.** In a short notation [s], [A] and [x] stand for the three different

of [A] and [x]

of course following the formalities of matrix **algebra such that equation** (2) is reconstituted.

Now suppose that we solve [s) = [A][x] by the **usual formalities of algebra, then** we should obtain [x] = [1/[A]][s] where we assume at the **moment that [1/[A]] is** reciprocal of CA] and is not equivalent to 0. An interesting **feature of equation** (3) is that the expression is clearly **separated** into knowns, **Is] and CA), and the unknowns,** x]. The term x]. on solution, will contain the intersection **coordinates of the two** lines specified by the original problem, equation (1). By reference **to (1) we see that** Cs] contains the net profit obtainable at the **market before the subtraction of the** transport costs. Since CA] remains fixed, once we have it we **can examine changes in** x] as a function of **changes in the net profit as the market price fluctuates. This** is a lot more convenient than **solving the** entire **equation set from scratch, which** is what we should have to **do using the methods of ordinary algebra. The solution** for this problem is discussed **in VI(i)a and the inverse is given numerically in IV(iv)d.**

An identical formal solution applies to solving the classical least squares Normal Equations. These equations arise from the problem of finding the best fitting line to a scatter of points in **a bi-variate plot, Figure** 2. **The equations are composed of** various sums constituted from the original **data; the x's and y's which define the points** in the scatter plot; full details **are given in VI(i)b below. The equations are:**

$$\begin{array}{lll} \Sigma y = aN + b\Sigma x & & \text{where } \Sigma \text{ stands for the summation operator} \\ \Sigma xy = a\Sigma x + b\Sigma x^2 & & \end{array}$$

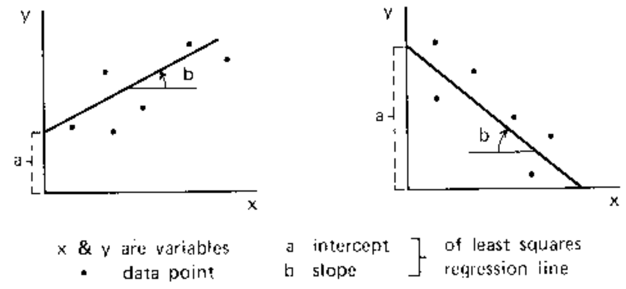


Figure 2

Typical scatter plots for least squares fit

All the summation terms and N are known and so the problem is written in matrix algebra as:

$$\begin{bmatrix} \Sigma y \\ \Sigma xy \end{bmatrix} = \begin{bmatrix} N & \Sigma x \\ \Sigma x & \Sigma x^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

or $[y] = [X][b]$

with, as before, the solution [t)] = [1/00][y], and CIA contains both a and b, respectively the intercept and slope of the best fitting line. We shall have to show that some such object as [1/[X]] exists, and how to calculate it, but assuming that it does exist (and for sensible problems it does) then matrix notation shows how a very compact solution can be written, and how it will apply however many terms there are in the Normal Equations.

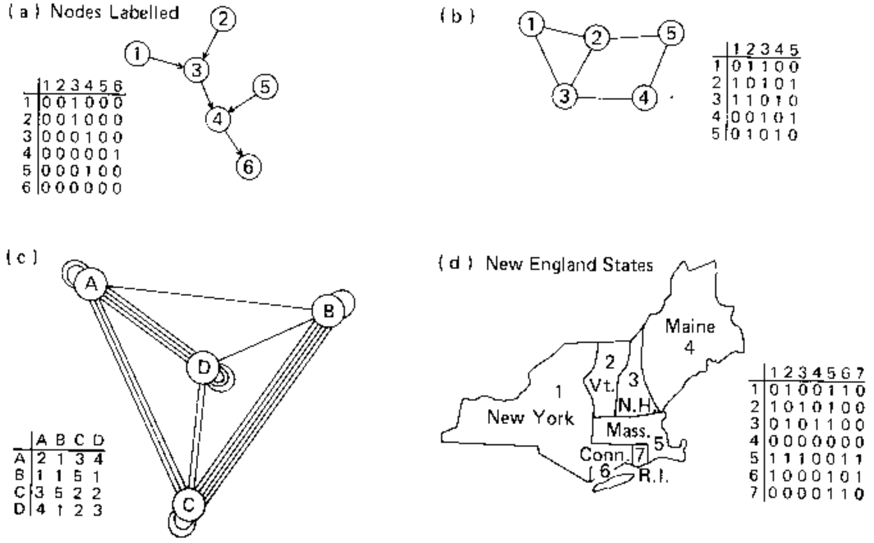
Another advantage of the inverse method is that it may be seen that it only contains products of x, and the term N in the top left-hand corner. Consequently to solve the problem for another set of y values just requires a simple operation, not the complete solution all over again. This is very convenient where a problem is repeated many times, say in the production of cubic trend surfaces of monthly rainfall totals for the same set of rainfall stations; in effect it takes advantage of the fact that the spatial location of the stations doesn't change.

As a final informal example consider the movement of flood water through a stream network caricatured as a series of connected nodes, such as is illustrated in Figure 3(a). In one time period the amount of water at a node, let us say from a storm of short duration, moves downstream to the next node. With some effort we can calculate the distribution of the water in the stream network by tracing the various paths and keeping track of the total amount of water at each of the nodes as time passes. For a complicated example, however, the labour is considerable. Matrix algebra handles this by representing the stream network as a matrix, CS], where a number one shows a downstream connection between two nodes, and zero shows no such connection. Let [w]t stand for the distribution of water at the various nodes in the system at time t. Then the simple iteration model:

$$[w]_{t+1} = [w]_t[S]$$

represents the movement of the water in the system, and where once again juxtaposition indicates the process of matrix multiplication. An essentially identical model is used

to model the growth and redistribution of population amongst regions between census periods (see section VI (iii) and Rogers 1968, 1971, 1975).



	1	2	3	4	5	6
1	0	0	1	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	0	0
4	0	0	0	0	0	1
5	0	0	0	1	0	0
6	0	0	0	0	0	0

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	0	1
3	1	1	0	1	0
4	0	0	1	0	1
5	0	1	0	1	0

	A	B	C	D
A	2	1	3	4
B	1	1	5	1
C	3	5	2	2
D	4	1	2	3

	1	2	3	4	5	6	7
1	0	1	0	0	1	1	0
2	1	0	1	0	1	0	0
3	0	1	0	1	1	0	0
4	0	0	0	0	0	0	0
5	1	1	1	0	0	1	1
6	1	0	0	0	1	0	1
7	0	0	0	0	1	1	0

Figure 3

How graphs, diagrams and maps generate matrices

In addition to the techniques discussed in this booklet there is also another group of matrix methods in which the matrix represents a spatial or problem structure and the concern is with manipulating the structure to yield the desired answer, usual]y a maximum or minimum value for some sum defined upon it, or alternatively a desired routing or selection of nodes in the structure. The elementary linear programming in the transportation model described by Hay 1977 (CATMOG 11) and the simplex model described by Killen 1979 (CATMOG 24) are such examples. The various classes of problem described by Scott (1971) are often cast in matrix notation, but their solution does not yield to the methods described here.

(ii) PREREQUISITES AND LITERATURE

Apart from an ability to do ordinary arithmetic, an acquaintance with elements of ordinary algebra, a willingness to conscientiously work through the examples and exercises, and to apply them to independent problems, the student needs no special prerequisites to fallow this booklet. The aim is to give a self-contained numerical acquaintance with the main processes using necessarily small examples. Computer routines in BASIC are given in an appendix for matrix inversion and for finding the eigenfunctions of a matrix so that no undue delay need occur for those people possessing personal computers, but whose software packages fail to isolate these fundamental elements of matrix arithmetic. In addition some worked examples using the commands of MINITAB have been included in an appendix for users with access to this user-friendly mainframe package for matrix manipulation.

No attempt is made to give formal proofs of the results given here. Proofs may be sought in standard textbooks such as Hohn (1972), Hadley (1961), Varga (1965) and Searle (1966). In a more geographical vein texts by Rogers (1971, i975), and Wilson and Rees (1977) all discuss their topics with a heavy emphasis on matrix algebra. In particular Rogers (1971) gives good numerical examples of his procedures. There is a voluminous literature on matrix methods and no writer can do more than indicate the ones he has found useful.

(iii) MATRIX OPERATIONS AS 'PROCESSES'

The great advantage of matrix notation is the compact way in which it represents unwieldy tables of data, or the structure of spatial systems. As a way of obtaining solutions to specific matrix problems there is nothing to beat it, yet I should be remiss if I failed to point out that to manipulate the symbols you must understand what happens when they are combined, in both an arithmetical and a geographical sense. For this reason most of this text will write out the numbers most explicitly, perhaps almost pedantically, but as a bonus we shall keep the matrices and/or the numbers small and simple. There will, by and large, be a marked deficit of that compact notation whose virtues r have just eulogised. Once a thorough understanding is attained, then the formal and correct manipulation of the symbols is a great reward.

Several authors have dealt with the geographical meaning of matrix manipulations. Garrison (1960), Nystuen and Dacey (1961), Pitts (1964), Gould (1967), Tinkler (1972, 1976, 1977 CATMOG 14), all give fairly thorough examples. As we shall see in a later section, Markov chains are most easily developed using matrices and in this connection the reader should consult Collins (1975 CATMOG 1).

The mathematical meaning of some of the processes to be discussed below can also aid in understanding and some use has been made of the idea that, in repeated multiplication of a vector by a matrix, the matrix rotates the vector in geometric space, often towards a fixed position. Gould (1967) made use of this idea to a give a visual form to the meaning of eigenvalues and eigenvectors, and the idea is often pursued in texts on Factor Analysis for social scientists, e.g. Harman (1967) and Rummell (1970). Strictly of course, the mathematical 'meaning' of processes is usually that the numbers obtained satisfy a matrix equation defined on the system. However, this is not always helpful in understanding what these particular numbers mean in a particular application.

(iv) MATRIX ALGEBRA AND THE COMPUTER

Matrix algebra has its roots in the nineteenth century work of Cayley but was subsequently rediscovered and applied to fields as diverse as psychology and quantum mechanics from the 1920's onwards. When computers became generally acccessible by the 1960's it was clear that the vast data tabulations, which computers handle so quickly, and matrix algebra were ideal suitors. The main high level (user) languages then available such as ALGOL. and FORTRAN both included matrices as a basic type of variable, and in multidimensional forms; not just rows and columns, but stacks as well, like room levels in a sky-scraper. Matrices of more than three or four dimensions

are not easily imagined but they are useful in the manipulation of some complex problems involving data tabulations. However, they are rarely used in matrix algebra as such, and in this booklet we will not go beyond rows and columns.

Both ALGOL and FORTRAN require the programmer to manipulate the matrices in the proper way, i.e. using DO loops to combine the rows and columns in multiplication ((v) below). Thus programmers usually have a good working knowledge of matrix algebra. Some languages, e.g. APL, BASIC, and Pascal do support symbolic matrix algebra, but it is rare if not completely absent from the affordable end of the microcomputer market, where the BASIC supported omits this facility. Thus most micro users are back in the early days of mainframes in this respect.

BASIC has the quirk that the indexing of rows and columns begins at 0, rather than 1, as it does in FORTRAN. This causes no particular problem, but it may waste precious space for the unwary.

There is then a symbiotic relationship between matrices and computers. Matrix algebra provides a very convenient notation: however, when we get down to the nitty-gritty of doing the calculations, computers are essential.

(V) EMPIRICAL EXAMPLES OF MATRICES

One of the earliest examples of the use of matrices was Garrison (1960) where he represented the road network of a part of the USA as a matrix of Os and is. He then used powers of this matrix to count the available routes between places in the system. By analogy to ordinary algebra the nth power of the matrix is the matrix multiplied by itself n times. In this way the overall connectivity of any node in the system could be found, and compared to other nodes in the system. Pitts (1964) and Carter (1968) made similar uses of the technique. A network of roads, or a cellular system of areas is easily represented by a matrix of is and Os. In Figure 3 five different systems are coded. Figure 3(d) illustrates administrative systems for which cells are counted as connected, and are marked with 1 if they have a common boundary, zero otherwise. Numbering the cells in a different order will cause the matrix to 'look' different, but it will not affect the final answers. Matrices of this type can be termed STRUCTURAL MATRICES.

In a different category are FLOW MATRICES which might, for instance, be used to record inter-regional population movements from origin-destination data collected in censi or surveys (Rogers 1968), telephone calls between exchange areas (Soda 1968, Hirst 1972), or taxi flows between traffic zones based on enumeration areas (Goddard 1970). In this case the matrix entries can be larger than 1, though many may be zero, or nearly so. They nevertheless imply a spatial structure of connections, weighted by the strength of the flow, and this is true even if all possible connections have flows larger than zero. Nystuen and Dacey (1961) provide an early example of the matrix analysis of flows.

Naturally, matrices need not imply a spatial structure, although usually a tabulation will be based upon geographical spatial units. For example, many urban system analyses collect socioeconomic data on the basis of census areas at various scales, and then proceed to analyse this DATA MATRIX to extract groupings, either of areas, or of socioeconomic categories. Analysis is usually based on the eigenfunctions described in Section VII below, and emerge in the literature as Component Analysis and varieties of Factor Analysis (Hirst 1972, Goddard and Kirby 1976 CATMOG 7, Daultrey 1976 CATMOG 8). The usual first step in Component and Factor models is to obtain a CORRELATION MATRIX showing the pairwise correlation amongst the variables. Because the number of socioeconomic variables is usually less than the number of spatial collecting units the correlation matrix is usually amongst these variables, rather than between the regions. In a sense the correlation matrix can be regarded as a sort of flow matrix since the square of the correlation coefficients, the elements in the matrix, measure the amount of nonrandom common variance between each pair of variables. That is to say it can be thought of as a 'variance' flow.

TRANSITION MATRICES and INPUT-OUTPUT MATRICES are two sorts of flow which have a particular structure imposed upon them by the problem they are structured to solve. To form a transition matrix each element in any row is divided by the tota] for that row. Therefore each row can be thought of as a set of weights or probabilities controlling the probability of transition from the cell indexed by the row number to any other cell, indexed by the column. Operations on matrices of this type are described by Collins (i975 CATMOG 1).

The INPUT-OUTPUT MATRIX has entries called technological coefficients which show the amount of goods, in cents or pennies, purchased by a given industry from all the other industries, including itself and which are needed to produce one dollar or pound's worth of industrial output. Unlike the transition matrix, the rows of the input-output matrix sum to less than one (or the industry wouldn't make a profit). Manipulations on this matrix show the multiplier effect on the economy for a given demand.

These are the principal categories of empirical matrices the geographer will encounter in the literature, or construct in his research. Once analysis gets underway most empirical matrices become equivalent to flow matrices; even structural matrices imply the potential of flow. Because of this it is well to consider the strictures mentioned in Section I (iii) above: that operations on matrices always have an interpretation as movements within the implicit spatial structure of the problem, and the meaning of these interpretations must be carefully considered. The reader can follow up issues of this sort in, for example Stephenson (1974), Tinkler (1976), Garner and Street (1978) and Tinkler (1979).

The mathematical properties of some matrices also give rise to particular names needed to identify them. These are dealt with in Section III below, but first it is necessary to go through some formal definitions of the basic entities in matrix algebra.

II DEFINITIONS

(D MATRIX

A MATRIX is a rectangular array or table of numbers enclosed in either square or curved brackets; I shall use square ones. An n by m matrix has n rows and in columns. For example:

$\begin{bmatrix} 3 & 1 \\ 2 & 0 \end{bmatrix}$ is a 2 by 2 matrix, $[3]$ is 1 by 1 matrix, or just plain 3

$\begin{bmatrix} 2 & 1 & 9 \\ 10 & 4 & 2 \end{bmatrix}$ is a 2 by 3 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is a 2 by 2 matrix

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 3 \end{bmatrix} \quad \text{is a 3 by 2 matrix}$$

By strict convention, when a matrix is said to be n by m, the first number ALWAYS indicates the number of rows, and n by m is said to be the 'order of the matrix. When n = m the matrix is said to be square of order n, and this is an extremely common type of matrix, see Figure 3. Within the matrix itself, each of the n x m positions or elements must be defined, even when they are zero. It is not usual or good practice to leave elements blank. Most of the time the elements are explicit numbers, but they may be algebraic entities, in which case they are assumed to behave in the manner of ordinary numbers. For the sake of illustration the numbers used in this booklet will be small, usually integers, and often positive, but in practice they will usually be large, fractional and even negative (e.g. correlation coefficients). In written work the numbers should be well enough spaced apart from one another that the presence 'of a space serves to separate them, and so that no confusion ensues.

(ii) SCALAR

In the previous section [37 was defined as aixi matrix. The definition is needed because certain matrix operations may end with a single number (Section IV (iii)c), and clearly it should have a matrix definition. However, the 1 by l matrix is merely an ordinary number, and it behaves and is treated as such. The usual term, therefore, for a 1 by 1 matrix is SCALAR, since its usual function is to scale the elements of a matrix or vector (see below) up or down by a constant factor. It is not usually written in brackets. However, the reader should remember that if need be, and in the tradition of Humpty-Dumpty (who made words do extra work and paid them extra for the effort), its a matrix when I want it to be one.

(iii) VECTOR

In (i) above I did not illustrate, purposely, matrices of the type 1 by m or n by **I,** although they are quite valid. In the former case the matrix has just one row, in the latter just one column. Matrices of this type are so common in applications that they are universally given the name VECTORS, with the obvious sub-types ROW vector and COLUMN vector. The latter is seen in Equation 3 of I(i). It might seem unnecessary to have both types of vector but the need arises from the way they emerge naturally when certain sorts of operations are defined on matrices. It is permissib]e, but not strictly necessary, to write row vectors with the elements separated by commas if there is any danger of confusion, or to save space. However, clear spaces serve the same purpose. Therefore [3,1,2,0] is the same as [3 **1** 2 01

(iv) MATRIX DIMENSIONS

The dimensions of a matrix as defined in computer programs is equivalent to that of order defined above. In programs enough space has to be reserved, i.e. the matrices are DIMENSIONED, for the largest matrix size one might want, although the typical problem may use much less space than this. The term ARRAY is sometimes used in computing languages for the terms matrix and vector in matrix algebra. There is also a relationship to Euclidean dimensions since each row of the matrix is usually regarded as an axis orthogonal to all other axes, and Factor and Component models make explicit use of this geometric view of a matrix.

(V) SYMBOLIC REPRESENTATION

In printed works a matrix is often shown by a boldface capital letter, **B,** and a vector by a boldface lower case letter, **b,** although some books use italics. In handwritten work capitals and lower case, underlined, will suffice to separate matrices from ordinary numbers, scalars, with which they often appear in association. Brackets of various sizes, written in association with matrix symbols, are used as in ordinary algebra to modify the order of operations and to group terms as required.

In this booklet it will be convenient (because of my word processor!) to write a matrix or a vector in square brackets: the matrices with capitals, the vectors with lower case. Such a notation is also convenient for handwritten work. Hence:

[w] = [r][S].

The link between specific locations in a matrix and the general matrix symbols is achieved by defining a matrix as the assemblage of its parts:

[A] = [a(ij)]

and in this case [a(ij)] stands for all the various numbers of the matrix enclosed in their brackets, as I shall now show in detail.

(vi) LOCATING AN ELEMENT IN A MATRIX - INDEX NOTATION

A given matrix has n rows and m columns. However, to refer to arbitrary rows, columns and elements in a matrix it is usual to use the terms i and j. The index i stands for any chosen row, and is a number in the range 1 to n (the maximum number of rows.) Likewise for the columns, j lies in the range 1 to m. A 3 x 4 matrix would be written thus:

$$[A(34)] = \begin{bmatrix} a(11) & a(12) & a(13) & a(14) \\ a(21) & a(22) & a(23) & a(24) \\ a(31) & a(32) & a(33) & a(34) \end{bmatrix} = [a(ij)]$$

and more generally:

$$[A(nm)] = \begin{bmatrix} a(11) & .... & a(1m) \\ ..... & .... & .... \\ a(n1) & .... & a(nm) \end{bmatrix} = [a(ij)]$$

where [A] has dimensions, or order, n x m. Notice that in this case the element is referred to symbolically with a lower case letter. This is to indicate that any individual element in a matrix is a scalar, or ordinary number. The brackets are to indicate the complete set of a(ij)s. Written unenclosed by square brackets an a(ij) refers to an individual element on its own. Therefore:

$$[B] = \begin{bmatrix} e & 3 & 0 \\ 4 & 2.5 & z \\ x & 0 & y \end{bmatrix} = [b(ij)]$$

then b(23) = z and b(32) = O. If row and column indices are identical then the elements lie on the main diagonal running from top left to bottom right, e.g. b(11) = e, b(22) = 2.5 and b(33) = y. It is not often necessary to reference particular elements and the indices i and j are most often used running over all the available values, i.e. i in the range **1** to n, j in the range 1 to m, in order to define the various arithmetical operations discussed below in Section IV.

Finally recall (as mentioned above) that the first index ALWAYS indicates the number of rows, and the second the columns, even if the order of i and j, n and m are interchanged. This precedence enables the definition of matrix symmetry and transposition to be described.

III SPECIAL. TYPES OF MATRIX

A number of particular types of matrix is needed in the process of doing arithmetic with matrices. They arise from certain common structures with sufficient frequency to merit special names.

(1) I1A.TRIX

Matrix equalities and inequalities can be written as in ordinary algebra. Two matrices or vectors are equal if and only if ALL their corresponding terms are equal. For example [A] = if and only if a(ij) = b(ij) for all pairs of i and j. When equalities or inequalities are written with respect to a scalar then the same rule applies: all entries in the matrix must obey the requirement. So, for example, writing [b] > 0 states that all elements of [b] are larger than zero. Similarly [b] = **1** sets all elements of [b] equal to **1.** Writing IA] # 0 would indicate that no element of [A] is equal to 0.

(ii) NULL MATRIX

The NULL matrix of order n is, as the name might suggest, a square n x n matrix full of zeros. It is symbolised by On Similarly a null rectangular matrix can be defined if required. The null matrix often, but not always, acts in matrix algebra as 0 does in ordinary algebra. For example, two matrices, neither of which contains zeros may have a matrix product equal to [0]. However, it acts as a conventional 0 for matrix addition and subtraction, and as a term in a matrix product will cause a null product.

(iii) IDENTITY MATRIX

The IDENTITY or UNIT matrix (either term is common) is a square matrix of zeros except for the MAIN or PRINCIPAL diagonal of is going from the top left to the bottom right corner:

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

It is symbolised by [I] and may be written for any order. The matrix acts in matrix multiplication as a 1 does in ordinary algebra (see IV (iii) (e) below).

(iv) MATRICES AND VECTORS OF ONES

There is sometimes a need for a matrix or vector composed entirely of ones. Such forms are easily defined by equalities, for example [E] = 1, and [E] are conventional notations. In the case of a vector where the need is often greater the convention is often [x] = = 1, which defines a unit row vector. Unit column vectors are similarly defined, for convenience the transposition notation is often used (see below) where [D^{T} indicates the transpose of [D, i.e. the row vector becomes a column vector.

(v) DIAGONAL MATRICES

If the ones in the identity matrix are replaced with either a constant, k, or any set of numbers then the matrix is said to be DIAGONAL:

$$[D] = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

The label [D] can be used as a mnemonic, but is not universal. A common variation of the form is the tri-diagonal matrix:

$$[D] = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Of course, the elements marked I could be any numbers whatsoever.

(vi) UPPER AND LOWER TRIANGULAR MATRICES

In the *upper* **TRIANGULAR matrix all entries** *below* **the main diagonal are zero, and vice versa for the lower triangular matrix.**

$$\text{An upper triangular matrix} \quad \begin{bmatrix} a & b & c & d \\ 0 & e & f & g \\ 0 & 0 & h & i \\ 0 & 0 & 0 & j \end{bmatrix}$$

It is actually easier to 'see' the structure if the zeros are left blank, contrary to normal practice.

(vii) SYMMETRIC MATRICES

A very important subset of square matrices is SYMMETRIC matrices. A matrix is symmetric when mirror image positions across the main diagonal are exactly equal. For example, letting a, b, c ... be numbers then: .

$$[A] = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}$$

is a symmetric 3 x 3 matrix. It is irrelevant what the values on the main diagonal are. They may be zero, all the same, or all different: it is the off-diagonal positions that define the symmetry. A symmetric matrix is most often defined by initial definition, e.g. "Let [H] be a symmetric matrix." It can be also defined using matrix equalities, by b(ij) = b(ji) for all i and j. More compactly it may be defined using the idea of a transpose, see the next section.

In geographical examples symmetric matrices very often are defined by problems, for example correlation matrices are symmetric, and the graphs of transport networks, at least in elementary examples, usually have symmetric adjacency matrices (Fig. 3).

(viii) THE TRANSPOSE OF A MATRIX

The TRANSPOSE of a matrix is an operation on a matrix rather than a kind of matrix. However, it is so commonly used that it is convenient to define it here, The transpose of [A] is defined symbolically as $[A]^{T}$. Another occasional form is trs[A]. It is obtained by interchanging the rows and columns of [A]: the first row of [A] becomes the first column of $[A]^{T}$, the second row becomes the second column, and so on. It is defined for all types of matrix, including vectors, e.g.,

$$[A] = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 5 & 1 \end{bmatrix} \text{ so that } [A]^{T} = \begin{bmatrix} 2 & 4 \\ 1 & 5 \\ 3 & 1 \end{bmatrix}$$

and

$$[v] = [3\ 1\ 1\ {-2}] \quad \text{so that} \quad [v]^{T} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ -2 \end{bmatrix}$$

and $$[r] = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix} \quad \text{so that} \quad [r]^{T} = [2\ 3\ {-1}]$$

By definition a symmetric matrix is equal to its own transpose, [A] = [M^{T}. The purpose of the transpose is usually to switch matrices and vectors around into positions that make various operations in matrix algebra possible, i.e. computationally legal.

(ix) PERMUTATION MATRIX

A PERMUTATION matrix contains only Os and is such that each row and column contains exactly one 1. It is used in conjunction with its own transpose, and matrix multiplication to reorder a matrix. For this reason I shall delay discussion of it until section IV (iii) (f).

IV MATRIX ARITHMETIC

(i) ADDITION AND SUBTRACTION

The procedures of matrix addition and subtraction are easy to learn. However, unlike ordinary algebra it is necessary to check that two matrices are 'conformable' for addition (subtraction). To be conformable for addition, two matrices have to be of identical dimensions. For example, while two matrices of dimension 3 x 4 and 3 x 4 are conformable for addition, two matrices of dimensions 3 x 4 and 4 x 3 are not. (However, conformability for multiplication follows different rules, see (iii) below.)

Given conformability, addition and subtraction proceed by the usual process, element by element:

$$\begin{bmatrix} 3 & 1 & a \\ x & 6 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 1 & a \\ 2x & 7 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 2a \\ 3x & 13 & 3 \end{bmatrix}$$

Because individual matrix terms may have positive, zero or negative signs the usual care is necessary when combining terms:

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

Because everything proceeds term by term, and each computation is independent of every other, it is clear that the ordinary rules of arithmetic hold and we can combine any number of matrices in this fashion provided only that they are conformable. It is also clear that addition (subtraction) is commutative and associative (it is independent of the order in which the matrices are written, or the order in which the addition (subtraction) is carried out).

Formally matrix addition (subtraction) is written as:

[c(ij)] = [a(ij)] + (-) [b(ij)]

or

[C] = [A] + (-) [B]

In the first equation the term by term procedure is explicit, in the latter it is implicit.

(ii) MULTIPLICATION OR DIVISION OF A MATRIX BY A SCALAR

It is convenient at this stage to describe the adjustment of a matrix by a scalar. This involves multiplying or dividing every element in the matrix by the same number. According to your fancy a scalar is an ordinary number or a 1 x 1 matrix. If you prefer the former then I have described the procedure used to scale up or down the elements of a matrix by a constant amount. In the latter case the fancy title for the same thing is forming the KRONECKER or DIRECT PRODUCT of two matrices. In either case the result is the same. Let k = 3, then:

$$k \begin{bmatrix} 2 & 1 & 4 \\ 3 & 0 & -2 \\ 1 & -1 & 6 \end{bmatrix} = \begin{bmatrix} 6 & 3 & 12 \\ 9 & 0 & -6 \\ 3 & -3 & 18 \end{bmatrix}$$

Of course the same procedure is used whatever the numerical value of k. In reverse the procedure is used to extract a common factor from a matrix, as sometimes happens in the hand computation of inverses (section iv(c) below). The scaling of a matrix by a fraction occurs in some applications in which matrices are powered and which are discussed in later sections.

(iii) MATRIX MULTIPLICATION

(a) *Multiplication of two matrices*

The reader will have guessed that things can't stay that simple for long, and multiplication is a little more involved, although once the procedure is routine it can be quick for hand examples. The method will be described first for two matrices, after which it will be seen that the multiplication of a vector and a matrix is a simplification of the basic method. First, however, it is necessary to check for conformability. To ensure that two matrices are conformable for multiplication write out their dimensions **IN THE ORDER IN WHICH IT IS INTENDED TO DO THE MULTIPLICATION.** So suppose [A] is 3 x 4 and [B] is 4 x 3, and we wish to compute [C] = [A] [B] then write 3 x 4, 4 x 3. The rule is that the matrices are conformable if the *inner two* numbers are identical. Since they are in this case, 4 and 4, the matrices are conformable for multiplication (although *not*, note, for addition). The rule also states that the result, [C], will have dimensions given by the order of the outer two numbers; hence 3 x 3. To take another example with the dimensions 2 x 6 and 6 x 4. From the rule they are conformable since the inner pair is 6, and the result will have dimensions 2 x 4. Clearly, the matrix result of muliplication need not be the same size as either of the original matrices. However, a simpler and very common case is that of two square matrices of identical dimensions. They will both have dimensions n x n, and so are conformable, and the result will also be n x n. This situation is encountered when a matrix is to be raised to an integer power, and is multiplied by itself.

Now to the process itself. In verbal terms, and given two conformable matrices written out in the order they are to be multiplied, we take the first row of the first matrix and the first column of the second matrix. Running along the row and column we multiply corresponding pairs and then add all the results. This gives us the element **(i,1)** in the result. In general to form the ijth element of the product matrix we combine in this fashion the ith row with the jth column. An easy mnemonic is to remember that we take a Row and a Column: matrix multiplication is Roman Catholic!

As an example

$$\begin{array}{ccccc} [A] & \text{x} & [B] & = & [C] \\ \begin{bmatrix} 3 & 1 & 2 \\ 1 & 4 & 2 \end{bmatrix} & \text{x} & \begin{bmatrix} 6 & 3 \\ 1 & 0 \\ 4 & 1 \end{bmatrix} & = & [?] \end{array}$$

Doing everything in order, the matrices are conformable because they are 2 x 3 and 3 x 2, and the result will be 2 x 2. The element (1,1) in [C] is found by taking the first *row* of [A], (3 1 2) and the first *column* of [B], (6 1 4), multiplying corresponding terms and adding:

$$\begin{array}{ccccc} \text{row} & & \text{column} & & \\ \hline 3 & \text{x} & 6 & = & 18 \\ 1 & \text{x} & 1 & = & 1 \\ 2 & \text{x} & 4 & = & 8 \\ \cline{5-5} & & & & 27 \\ \cline{5-5} \end{array}$$

Therefore c(1,1) = 27. Similarly c(1,2) is

$$\begin{array}{ccccc} \text{row} & & \text{column} & & \\ \hline 3 & \text{x} & 3 & = & 9 \\ 1 & \text{x} & 0 & = & 0 \\ 2 & \text{x} & 1 & = & 2 \\ \cline{5-5} & & & & 11 \\ \cline{5-5} \end{array}$$

Therefore c(1,2) = 11. The reader should check that c(2,1) = 18 and c(2,2) = 5. The final product is therefore:

$$[C] = \begin{bmatrix} 27 & 11 \\ 18 & 5 \end{bmatrix}$$

As an example with square matrices:

$$\begin{array}{ccc} [A]\,[B] & = & [C] \end{array}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 14 \end{bmatrix}$$

On the other hand:

$$\begin{array}{ccc} [B]\,[A] & = & [D] \end{array}$$

$$\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 13 & 20 \\ 5 & 8 \end{bmatrix}$$

and obviously [A] [B] ≠ [B] [A] because, by inspection term by term, [C] ≠ [D] (for inequalities see section III (i)).

This illustrates an important point about matrix multiplication: *the order of multiplication matters* . This follows in part from the definition of conformability since a 3 x 4, 4 x 3 pair yields 3 x 3 whereas in reverse a 4 x 3, 3 x 4 pair produces a 4 x 4 which obviously cannot be the same. However, the same is true even for square matrices, as illustrated above. The order of multiplication matters and we say that [A] and [B] in [A] [B] = [C] do not commute, i.e. cannot swop places and give the same answer.

There is one fortunate case where square matrices do commute: a matrix commutes with itself and so powers of a matrix are easily defined, Take [Al

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{then } [A][A] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

The procedure can be repeated to get higher powers:

$$[A]^4 = \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix} = \begin{bmatrix} 199 & 290 \\ 435 & 634 \end{bmatrix}$$

Because the powers commute it follows that $[A]^4\,[A]^2 = [A]^2\,[A]^4 = [A]^6$.

(b) *Multiplication of a matrix by a vector*

The same rules apply to matrix/vector multiplication as to matrix/matrix multiplication. The only difference is that the whole procedure is simpler because in one of the matrices, the vector, there is only one row. The rules of conformability remain the same so that a **1** x 4 vector is conformable with a 4 x 4 matrix, or a 4 x 7 matrix. In the former case the result is still a **1** x 4 vector, in the latter a **1** x 7 vector. It is also necessary to introduce some more terminology. If the vector comes first in the order of multiplication we say that the vector PRE-MULTIPLIES the matrix, if after, it POST-MULTIPLIES the matrix. This distinction is made because in many cases the matrix is viewed as the stable element, while the vector keeps changing its numerical contents. As an example of pre-multiplication:

$$[v]\,[A] \quad = \quad [x]$$

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 & 1 & 2 \\ 3 & 1 & 0 \\ -1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 7 & 9 & 11 \end{bmatrix}$$

The equation is conformable since the **1** x 3 vector and the 3 x 3 matrix produce a 1 x 3 row vector. Because the resulting vector has the same dimensions as the original one, the multiplication can be repeated if the result, [x], is subsituted for NI The reader should check that in this case the result is [44 38 47]. This iterative procedure is very common and is the basis of Markov Chains (See VI (ii)). Similar comments apply to post-multiplication:

$$[A]\,[v]^T \quad = \quad [x]$$

$$\begin{bmatrix} 4 & 1 & 2 \\ 3 & 1 & 0 \\ -1 & 2 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 12 \\ 5 \\ 13 \end{bmatrix}$$

Notice that the system is conformable for post-multiplication *provided that [v]* is written as a column vector. Then the dimensions are 3 x 3, 3 x **1,** with the resulting column vector [x], of dimensions 3 x **1.** Notice that although the same matrix and vector is involved the result is not the same so that even in this simplified form the system does not commute. An important exception is the case of a symmetric matrix when it *is* true that [x)[A] = [F][x].

When the row vector contains only is (see III (iv)) then the product of pre-multiplication is equivalent to adding up the elements in the columns of the matrix. Likewise, a column vector of is can be used with post-multiplication to add the elements in each row. The reader should check that this is so for the matrix CA] above and that the results of [1][A] and [A][S] are respectively: [6 4 5] and [7 4 4].

(c) *Multiplication of two vectors*

Two vectors may be combined according to the same rules of conformability. Taking the most general rule first then if the vectors have a different number of elements they will only be conformable in one of the following configurations: n x **1, 1 x m** to yield an n x m matrix, and m x 1, 1 x n to give an m x n matrix. In either case the first vector is taken to be a column vector (it has either n or m rows) and the latter to be a row vector (with m or n columns).

However, these forms are rather unusual and more common is the case of two vectors each with n elements. These follow the same rules as above but the two n x n matrices that result are merely transposes of one another. For example:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\begin{bmatrix} 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 2 \\ 6 & 2 & 4 \\ 9 & 3 & 6 \end{bmatrix} \text{ or } \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 & 9 \\ 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$$

The primary use of such a construction is to represent, and to extract, the structure due to one component, factor or eigenvector of a correlation matrix, (see V (iii)). It is used in this *way* in Component or Factor Analysis.

However, two vectors of equal length may also be combined in the form 1 x n, n x 1 to yield a *scalar* answer, a 1 x 1 matrix:

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = [11] = 11$$

This form is termed the INNER PRODUCT (or DOT PRODUCT) of two vectors and one use of it is to determine if two vectors (thought of as points connected by a line to the origin of an n dimensional Euclidean space) are at right angles - are orthogonal - or not. For example the vectors **[1** 1] and [-1/3 1/3] are orthogonal because:

$$\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} -1/3 \\ 1/3 \end{bmatrix} = [0] = 0$$

The result is the same if they are taken in reverse order (check it!). This test is of use in the definition of eigenvectors of a matrix, and is discussed in Section V (iii).

The reader will have noticed by now the considerable freedom used in decisions as to whether a given vector is to be a column or a row vector. This is an important part of matrix methodology; the crucial criterion is that the matrix process to be used is appropriate to the problem in hand. Most of the manipulation with transposes is merely to ensure that the vector and matrices (which after all are merely lists and tabulations) are in the proper position for k the same formal rules of matrix arithmetic to be used each time. This sort of problem doesn't arise in ordinary algebra. It may be taken as a general rule that if a vector, distinguished in this booklet by lower case letters, PRECEDES (pre-multiplies) a matrix then it is a *row* vector; conversely if it FOLLOWS (post-multiplies) a matrix it is a *column* vector. However, a column vector will not always have a transpose symbol attached to it.

(d) *Symbolic notation for matrix multiplication*

The rules of matrix multiplication are written out in detail for each term, element by element, as follows, and make use of summation notation:

$$c(ij) = \sum_{k=1}^{k=m} a(ik)b(kj) \quad \text{(i.e. the summation } \Sigma \text{ extends over the } k = 1 \text{ to } m \text{ terms)}$$

The i and j terms are fixed for any particular c(ij), a single term in the matrix product [C) = [A] [M. The summation of the products then runs over the available k elements as determined by the dimension of the matrices: so if [A] is n x m and [B] is m x n then k runs over the m columns of [AI (for a fixed row i) and over the m rows of [B] (for a fixed column j).

(e) *Multiplication 17,2* E03 *and 17:9* 113

I stated earlier that usually [0] acts as a zero in matrix algebra, and that [I] acts as an identity element. It is as well to test these assertions, and they are easy to check:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Check that pre-multiplication by [O] also yields a [O] matrix. Similarly with [1]:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The reader should also check that if [x] is a row vector then [x][I] = [x], and similarly in the case when [x] is a column vector: it is merely a matter of carrying out the indicated arithmetic as a check. The identity matrix is used particularly in the definition of the matrix inverse which follows in the next section.

(f) *The permutation matrix again: an example of matrix manipulation*

The permutation matrix was mentioned earlier, but illustration was delayed until matrix multiplication had been developed. The purpose of the permutation matrix is to reorder the columns and rows of a matrix without a]tering the size of the elements. Suppose we have the matrix [T] which we wish to reorder to [T]* so that the columns are arranged according to their column sums, decreasing from left to right. Then suppose that [n is:

$$\begin{array}{ll} & \begin{bmatrix} 0 & 0 & 6 & 3 \\ 3 & 0 & 4 & 3 \\ 5 & 6 & 0 & 3 \\ 4 & 2 & 3 & 0 \end{bmatrix} \\ \text{column sums} & \begin{bmatrix} 12 & 8 & 13 & 9 \end{bmatrix} \\ \text{rank order} & \begin{bmatrix} 2 & 4 & 1 & 3 \end{bmatrix} \end{array}$$

The columns sums can be found from the multiplication UHT) = [12 8 13 9] where [1] is a row vector of four is. These column sums are now ranked in descending numerical order to make a vector of ranks [2 4 1 3]. The permutation matrix [P] is constructed as follows. If the jth column of [T] has rank k then:

p(jk) = 1, and = 0 otherwise.

Verbally, the jth row of [P] has a i in the column whose number is equal to the jth element in the vector of ranks. Therefore [P] is equal to:

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

There should be only one I in any column or row. The reordering of [T] to [T]* is now accomplished by performing the following multiplication:

$$[T]^* = [P]^T[T][P]$$

The reader, working from the left, should carry out the multiplications and see that the result is [T]* equal to:

$$\begin{array}{ll} & \begin{bmatrix} 0 & 5 & 3 & 6 \\ 6 & 0 & 3 & 0 \\ 3 & 4 & 0 & 2 \\ 4 & 3 & 3 & 0 \end{bmatrix} \\ \text{column sums} & \begin{bmatrix} 13 & 12 & 9 & 8 \end{bmatrix} \\ \text{ranks} & \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \end{array}$$

and the reordering has been achieved (the rows as well as the columns have been moved). The reader can check, as an exercise, what the following multiplications accomplish: [F][T] and [T][P].

(iv) MATRIX 'DIVISION'? - THE MATRIX INVERSE

(a) *Definition*

Formal matrix division in the straightforward sense, as defined for addition, subtraction and multiplication does not exist. Instead it is necessary to take a long-winded route. If we wish to divide by the matrix [A] we must first find its inverse, written inv[A], [1/[A]] or $[A]^{-1}$. Then we must multiply by this inverse. The tedium is in obtaining $[A]^{-1}$ from [A]; the multiplication that follows is straighforward. I showed in the Introduction how one might arrive at the definition of $[A]^{-1}$ by applying traditional algebraic methods to matrix entities.

Because we have defined an identity element for multiplication, [I], it is possible to define [AT¹ as being that matrix such that:

$$[A]\,[AT^{-1} = [A]^{-1}\,[A] = [I]$$

That is, it must yield the identity matrix on pre- or post-multiplication with the original matrix, and this parallels the algebraic definition of a reciprocal in ordinary arithmetic.

(b) *Incidental comments on the inverse*

The inverse is so important to matrix computation that some comments must be made at this point that bear on both its existence, and its computation.

The problem is to find (AT' given [A]. Happily if [A] has an inverse it has only one, and is therefore unique. Less happily [A] doesn't always have an inverse, but it is some measure of comfort that all methods used to compute [AT' will fail when it fails to exist. Usually if [A] fails to have an inverse it signifies that the original problem is ill-posed in some sense so that the lack of an inverse is not usually an inconvenience - it is usually a hint that the empirical problem is somewhat deficient. It is sometimes possib]e to see in advance that [A] doesn't have an inverse since this is determined by a number called the DETERMINANT. If it is zero the inverse fai]s to exist.

Even when the inverse does exist, and even though it is unique in the mathematical sense, it is well to be warned that numerical methods may yield an answer that is incorrect due to accumulating rounding errors in the arithmetic. Whenever possible double precision variables should be used in computer routines (Microsoft Basic, MBASIC, provides double precision variables to 16 digits) and it is well to remember that the ONLY test of the accuracy of a computed inverse is by testing the result according to the definition: [AT' [A] = [D to acceptable levels of accuracy. The program in the appendix uses 1E-6 (one part in a million) as a criterion for whether a given element in [Al¹ [A] is "close enough" to M. The problem is worst for large matrices with a great variability in the size of the original numbers (Unwin 1975).

(c) *Calculating* [AP by *,formula*

For serious work [AT' should be computed by a sub-routine designed to circumvent the problems mentioned above. However, it is useful to see how for small examples a formula can be found for (AT', and especially as it demonstrates how the DETERMINANT controls the existence of the inverse. Starting from the definition: [AT' [A] = [I] it is possible to write out for a 2 x 2 matrix:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where the first matrix, [A], contains the known elements, and the second matrix, [AT', contains the unknown elements of the inverse. Then by following the rules of matrix multiplication, applied element by element:

$$\begin{array}{lcll} ae + bg & = & 1 & \text{(i)} \\ af + bh & = & 0 & \text{(ii)} \\ ce + dg & = & 0 & \text{(iii)} \\ cf + dh & = & 1 & \text{(iv)} \end{array}$$

These are four equations in four unknowns (e, f, g, h) which can be solved by standard methods of algebra. I shall illustrate the computation of the first element. Take the pair (i) and (iii) (they contain the same pair of unknowns, e and g). Multiplying (i) by c and (iii) by a gives:

$$\begin{array}{l} aeo + bgc = \\ aec + adg = 0 \end{array}$$

then:

$$\begin{array}{lcl} adg - bgc & = & -c \\ g(ad-bc) & = & -c \\ g & = & -c/(ad-bc) \end{array}$$

Similar methods are used to obtain h, e, and f. In each case the same denominator will be found: (ad-bc).

This term is called the DETERMINANT and may be written as det[A] or I A I, i.e., the matrix is enclosed by vertical bars, *not* brackets, and this implies that the calculation of the number called the DETERMINANT is to be made. Thus I A I is a *number,* unlike [A] which is an array of numbers called a matrix. Clearly, if this number, the Determinant, is zero all elements will be zero and the inverse will be the Null matrix (see III (ii)). The remaining terms of the inverse should be found to be such that:

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

Similar algebraic methods can be used to find a formula for the inverse of larger matrices, but they become very lengthy. However, in all cases a common denominator to all terms is found and may be factored out as a scalar in the manner shown above. Most users will have no practical use for the Determinant so its computation in larger examples is bypassed here. The user may request it in many statistical packages, and it is well to note that as it approaches zero the inverse will tend to be very unstable numerically since the division may result in very large values.

The inverse formula for a 3 x 3 matrix, [M, is when:

$$[A] = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$[A]^{-1} = \frac{1}{\det[A]} \begin{bmatrix} (ei-fh) & -(bi-ch) & (bf-ce) \\ -(di-fg) & (ai-cg) & -(af-cd) \\ (dh-eg) & -(ah-bg) & (ae-bd) \end{bmatrix}$$

where $\det[A] = ((aei + bfg + cdh) - (gec + hfa + idb))$.

(d) *Calculating [AP numerically*

The proof of the pudding always lies in the eating when dealing with the inverse. In this section we shall check the 2 x 2 formula by using direct calculation. Although a straightforward algorithmic routine can be applied to obtain inverses by hand, I shall not describe it since for serious work pre-packaged subroutines will be available. However, an appendix does provide a listing of such an algorithm in a minimal form of BASIC so that a personal computer user has access to a functional subroutine. Another appendix provides a worked example showing how to use commands in MINITAB to obtain the inverse. (Note that mainframe BASICs will usually provide the inverse via a single line of programming code since mainframe BASICs support symbolic matrix algebra, however if double precision is required it may be necessary to use another programming language.)

We will now check the formula numerically in the 2 x 2 case. (The reader is also invited to check the formula algebraically by multiplying out the various terms.) Suppose that [A) is:

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

then from the formula $[A]^{-1}$ is:

$$(1/(6-4)) \begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

and checking:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

This shows that a matrix and its inverse commute, i.e. the order in which they are multiplied is immaterial. Because I shall have recourse to the inverses of small matrices in the examples that follow in later sections I shall by-pass further examples at this stage. However, the reader should check that the inverse to the problem posed in I (i) is:

$$\begin{bmatrix} -1 & 2 \\ 20 & -20 \end{bmatrix}$$

WARNING: in small and exact examples like the one above there are no rounding errors, but I shall reiterate the warning given earlier - in empirical work great care must be taken and the checking computation **MUST** be performed, and in computer work use double precision arithmetic *whenever it is available.*

V EIGENFUNCTIONS

A matrix must be square to possess EIGENVALUES and EIGENVECTORS, jointly called EIGENFUNCTIONS, and in all our applications the elements of the matrix will be real numbers. In many, but not all cases (certain probability matrices are an example), the matrix will be symmetric.

There are several ways to approach an understanding of eigenfunctions and to reach the widest audience something of each wil] be indicated. The mathematical view, and clearly this subsumes all other approaches, is that they are sets of numbers satisfying certain equations defined on matrices. However, a more practica] acquaintance may be gained by taking a geometric, or what is equivalent, an iterative view of how eigenfunctions arise.

(i) A GEOMETRIC VIEW AND ITERATIVE VIEW

A geometric approach is essentially visua], and it will be helpful provided it is realised that the image must be imagined as behaving in essentially the same way for matrices of n dimensions. I can only illustrate easily a two dimensional example. Suppose we have the vector [i 2]. The two elements can be thought of as a coordinate [x y] in a two dimensional Euclidean space, and can be plotted as a vector by connecting the point to the origin, [0 0], of the space. If we multiply the vector by a scalar, say 3, we get the vector [3 6]. Likewise if we add another vector to it, say [3 4] we would get [1 2] + [3 4] = [4 6]. All of these results can be plotted, (see Figure 4(a,b)). Now consider what happens if we take a matrix [R], and pre-multiply by some arbitrary vector, say [1 0]. The result is then taken back and again pre-multiplies [R], and so on. Let:

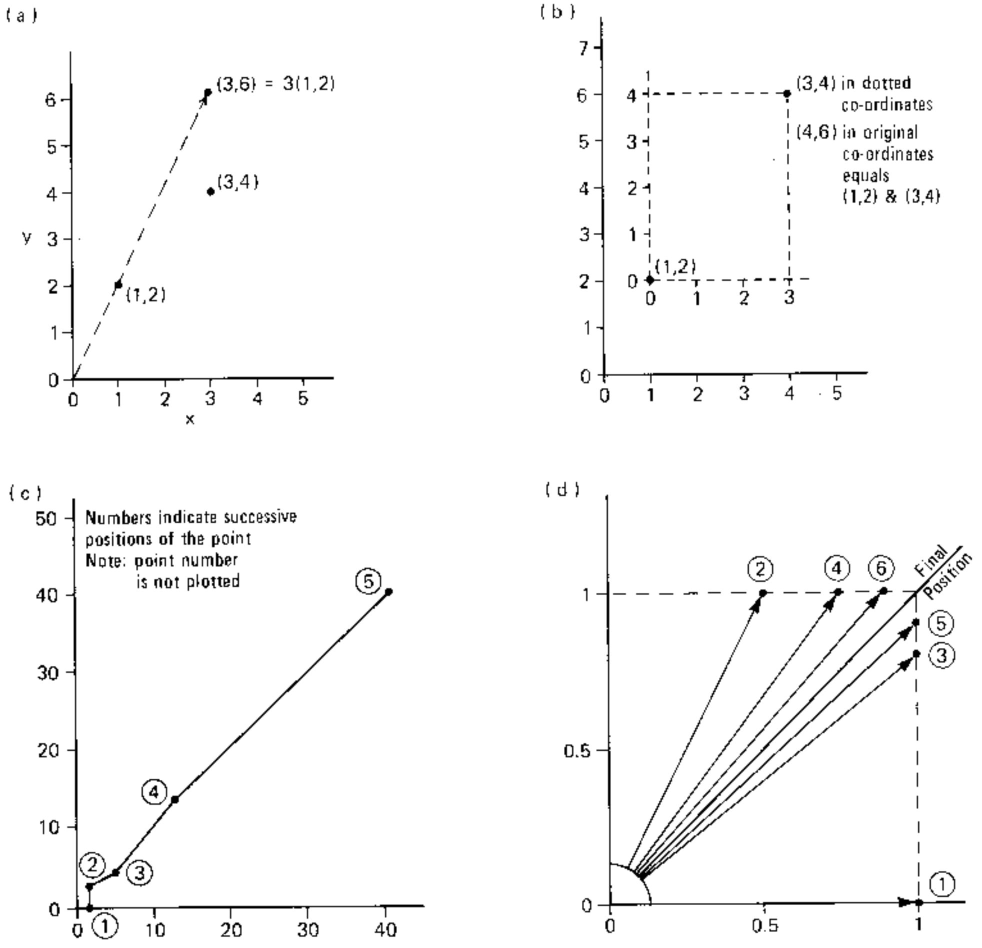


Figure **4**

Graphical illustration of eigenfunction behaviour

$$[R] = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

then

$$[1\ 0] \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = [1\ 2] \quad \text{and} \quad [1\ 2] \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = [5\ 4] \quad \text{etc.}$$

If the process continues we get the sequence of vectors:

vectors	Σ of elements
[1 0]	1
[1 2]	3
[5 4]	9
[13 14]	27
[41 40]	81

In this listing the final number is the sum of both elements in the vector and after the first step this sum inflates steadily by a factor of 3 for each step. That is to say that vector grows, or is stretched by this factor of 3, for each pre-multiplication performed. Roughly speaking this is also true of each element of the vector, although this relation takes a little longer to settle down and become exact. If we discount this inflating effect, say by dividing through after each iteration by the sum for each vector, then it also becomes apparent that the vector elements are slowly becoming equal. The last pair listed give [0.5062 0.4938] when divided by 81. Two steps later the relevant vector is [363 366] and the ratio is [0.4979 0.5021]: although the elements have reversed dominance, they are both closer to the ultimate ratio of [0.50 0.50].

In consequence it should be clear that the matrix [R] is having the effect of stretching the vector each time by a factor of three (see Figure 4(c)), and of swinging the vector to a position which is becoming 'fixed' in a relative sense, i.e. the two elements are becoming equal to each other (see Figure 4(d)). Surprisingly, this property holds *whatever* vector is supplied for repeated pre-multiplication. For this particular [R] the stretching factor will always be 3, and the ultimate ratio of the two elements will be [0,50 0.50]. The first is the principal (i.e. the largest) EIGENVALUE, and the second is its corresponding EIGENVECTOR. The set of all the pairs constitute the EIGENFUNCTIONS for the matrix in question. It will be obvious from this that the iterative approach is essentially identical to the geometric, the one numeric, the other graphic,

That the eigenvector is 'fixed' in position can be seen by pre-multiplying [R] by the vector [1 1]. The result is [3 3], from which we can see that the ratio of the elements on the vector has not changed, but their size has increased by 3, the corresponding eigenvalue.

While this iterative procedure, and its graphic counterpart, is convenient for finding the principal eigenfunction, it is more troublesome for finding others, because unless the vector used as an input is exactly at right angles (in a mathematical sense) to the principal eigenvector it will converge to the principal eigenvector. A solution to this problem is to 'extract' mathematically the effect of the first eigenfunction from the matrix, and then to proceed as before. The next step will extract the largest remaining eigenfunction, the second]argest in the original matrix. This can be done until all eigenfunctions have been extracted and is the basis of what is called the *power and deflate* method of finding eigenfunctions (and of the program in the Appendix). In a n x n matrix there wi]l be n of them. However these details make it necessary to give a mathematical definition of eigenfunctions.

(ii) MATHEMATICAL NATURE OF EIGENFUNCTIONS

An n x n real and symmetric matrix [A] has n eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$, and each of them has an associated eigenvector [x] (the notation assumes that λ is *always* paired with its corresponding eigenvector). Together, any eigenvalue and its corresponding eigenvector satisfy the equation:

$$[A][x] = [x][A] = \lambda[x]$$

In this equation, and as we saw in the previous section, λ, the eigenvalue is a *scalar* number. So the equation states that if you have the ith eigenvector of a matrix [A] then the *only effect* that [A] has on this eigenvector is to stretch it by the amount λ, the value of the corresponding eigenvalue. The first equality in the equation merely states that, since [A] is symmetric by hypothesis, the result holds for either pre- or post-multiplication of [A]. Solving the equation for all the pairs of λ and [x] is called the eigenvalue problem. While it is easy to state in principle, the numerical solution for large matrices with arbitrary entries requires great care. For this reason it is wise to rely on package programs whenever a full set of eigenfunctions is needed.

Reorganizing the equation gives:

$$[A][x] - \lambda[x] = [0].$$

We can factor out the [x] to get:

$$\big[[A] - \lambda[I]\big][x] = [0] \qquad \text{(expand, term by term, to check that this is valid)}$$

Note the appearance of the identity matrix, to keep the equation in order. The equation now has the form of an equation system [A][x] = 0, which is trivially true if [x] = 0, whatever [A]. However an eigenvector equal to zero is of little interest, so the alternative is that [A] acts multiplicatively as zero. This it will do if det[A] = 0, and then there are infinitely many solutions for [x], all multiples of each other; any particular one can be selected at will. Therefore, returning to the equation, it is required that $\det\big[[A] - \lambda[I]\big] = 0$, that is we have to find the values of λ so that the determinant is zero. Expanding this determinant (i.e. writing it laboriously out term by term!) yields an nth degree polynomial in λ called the characteristic polynomial, where n is the dimension of the matrix [A] (recall it is n x n by definition). The *roots* of this polynomial (the values of λ for which the equation equals zero) are the *eigenvalues* of [A]. Determinants were mentioned in passing in the section on the matrix inverse (IV (iv)) and a full definition is unnecessary here since the intention is merely to illustrate the method of the mathematical solution. If [A] is *say:*

$$[A] = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \quad \text{then } \big[[A] - \lambda[I]\big] = \left[\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}\right] = \begin{bmatrix} (1-\lambda) & 2 \\ 2 & (3-\lambda) \end{bmatrix}$$

then

$$\det[A] = \begin{vmatrix} (1-\lambda) & 2 \\ 2 & (3-\lambda) \end{vmatrix} = 0$$

$$\text{so that } \det[A] = (1-\lambda)(3-\lambda) - (2)(2) = 0.$$

Simplification gives $\lambda^2 - 4\lambda - 1 = 0$, which can be solved by any root finding technique to give roots of $\lambda_1 = 4.2361$ and $\lambda_2 = -0.2361$. The indices are assigned to the roots on the basis of the descending magnitude of the roots. As a check it is useful to know that the trace of a matrix, written tr[A], is the sum of the elements on its main diagonal, and this is equal to the sum of a]l the Xs. Therefore in this case tr[A] = 4, as is the sum of the two roots found, 4.2361 and -0.2361. Obviously when n is large, as in most useful examples, the procedure given above is both more lengthy and less elegant; hence the recourse to packages designed to compute the results efficiently and accurately.

There remains the problem of computing the corresponding eigenvectors. Returning to the equation which defined the eigenfunctions, we can insert the eigenvalues one at a time and solve for each of the unknown eigenvectors:

$$\begin{bmatrix} (1 - 4.2361) & 2 \\ 2 & (3 - 4.2361) \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Recalling the earlier section on notation the bracketed 1 and 2 attached to x merely indicate successive elements of the vector Ix]. Writing this out term by term as the multiplication indicates we obtain:

$$\begin{array}{rcrcl} -3.2361x(1) & + & 2x(2) & = & 0 \\ 2x(1) & - & 1.2361x(2) & = & 0 \end{array}$$

There are infinitely many solutions to these equations, all multiples of each other, so let $x(1) = 1$, then $2x(2) = 3.2361$ and so $x(2) = 1.618$. The elements on the eigenvector [x] paired with eigenvalue $\lambda_1 = 4.2361$, are therefore in the ratio [1 1.618]. A similar calculation for $\lambda_2 = -0.2361$ will give [1 -0.618], where the element 1 was, as before, arbitrarly fixed.

The calculations have been illustrated for a real symmetric matrix (albeit small) because for a great many applications, especially those involving Component or Factor Analysis, that is the typical situation. The input matrix is very often a matrix of correlation coefficients, which by definition is real and symmetric.

However, the same procedures can be carried out for non-symmetric matrices, but then the complete solutions usually involve the use of imaginary or complex numbers. In the case of transition probability matrices (see section VI (ii)), which are typically asymmetric, the principal eigenvalue is set to equal 1 by the character of the problem, and the interest lies in the elements on the fixed vector corresponding to it. Whenever the matrix is assymetric there are two eigenvectors (called conjugate) to every eigenvalue, however in the case of transition probability matrices only one of those associated with the principal eigenvalue is of interest, the other is a unit vector, all elements equa] 1. Little use is made of the other eigenfunctions.

Although the computation of the eigenfunctions is usually numerically tedious for anything but very small matrices it is possible to get some idea of the magnitude of the principal eigenvalue by inspecting the input matrix. It is known that the principal eigenvalue must lie between the limits imposed by the smallest and largest row or column sums of the input matrix. In the small example above we therefore know that it must lie between 3 and 5, which it does (more refined limits have been established but are not pursued here). If the input matrix is composed of non-negative values then it is also guaranteed that the principal eigenvector is non-negative (or more strictly has all elements the same sign), and the vector of row or columns sums (either will do if [A] is symmetric) gives a good estimate of the principal eigenvector. For example for [A] above $[1][A] = [A][1]^T = [3\ 5]$ which is linearly proportional to [1 1.667], a reasonable approximation to the exact solution [1 1.618]. However, all subsequent eigenvectors are bound to contain some negative elements. In the example above the second eigenvector was [1 -0.618]. However, it is important to realise that the signs on the terms are assigned arbitrarily and the vector is equally valid written in the form [-1 0.618] and this is true irrespective of whether the eigenvalue itself is positive or negative. When the *eigenvalue* is negative then according to the equation its effect (or equivalently that of the matrix) is to reverse the signs an its eigenvector with every iteration.

The reader should repeat this analysis on the matrix [R] given in the previous section, (i), whence the equation for the expansion of the determinant should be

$$(1-\lambda)(1-\lambda) - 4 = \lambda^2 - 2\lambda - 3 = (\lambda - 3)(\lambda + 1) = 0$$

which is to say that $\lambda_1 = 3$, and $\lambda_2 = -1$. Then the eigenvectors are found to be [1 1] and [1 -1] respectively, although they may have their signs reversed as noted above.

Daultrey (1976 CATMOG 8) gives a similar worked example for a 3 x 3 correlation matrix, apart from the actual extraction of the roots. Because of the general importance of the correlation matrix in geographical applications the next section shows how such a matrix may be teased apart into its parts by means of the eigenvalues and eigenvectors.

(iii) MORE ON THE EIGENSTRUCTURE OF REAL. SYMMETRIC MATRICES

One very useful property that eigenvectors have, and which was mentioned in an earlier section, is that of orthogonality. This is another way of saying 'at right angles' or 'normal' to each other. The property is very easily tested. If the INNER PRODUCT of two vectors is zero, then the vectors are *orthogonal* to each other. (To review the INNER PRODUCT see section IV-(iii)-c). If we take the two eigenvectors determined in the earlier section [1 1.618] and [1 -0.618] then indeed:

$$\begin{bmatrix} 1 & 1.618 \end{bmatrix} \begin{bmatrix} 1 \\ -0.618 \end{bmatrix} = 0$$

The reader can check that this does not depend on the assignment of signs on the second vector: they can be reversed to give the same result. This property holds mutually for all n eigenvectors of an n x n matrix. It is the basis of the method o2 Principal Components, whose main aim is to find the 'natural' axis system of any particular matrix, and the data set from which it was drawn. Each eigenvector is an axis. The elements of the eigenvector show how well related each original point is to this axis, and the eigenvalue is a measure of the relative strength of the axis, in terms of the original data set.

To show how this can be used to extract the 'effect' of each eigenvalue from a matrix we must first reduce the eigenvectors to the UNIT VARIANCE form. Take the eigenvector, in whatever form it was obtained, and sum the square of its elements: for the example above; 1 + 2.617924 = 3,617924; now multiply each element on the vector by the square root of this number (1.902084) to get [0.5257 0,8506]. The elements are still in the original ratio but their total variance, the sum of their squares, now equals 1.

Now that the eigenvectors have been reduced to a standard form we can produce a matrix representing the effects of that eigenvector and its eigenvalue, Daultrey's example will be used, he starts with a correlation matrix as follows:

$$[R] \quad = \quad \begin{bmatrix} 1.0 & 0.8333 & 0.5833 \\ 0.8333 & 1.0 & 0.9167 \\ 0.5833 & 0.9167 & 1.0 \end{bmatrix}$$

The eigenvalues are $\lambda_1 = 2.5636$, $\lambda_2 = 0.42140$, $\lambda_3 = 0.014980$, and the eigenvectors arranged by columns, and in unit variance form are:

e-vector 1	e-vector 2	e-vector 3
0.54116	0.76202	0.35416
0.62088	-0.079233	-0.77989
0.56623	-0.64269	0.51608

The signs are the reverse of those given in Daultrey because (a) these are the results produced by the program listed in the Appendix, and (b) it emphasizes the fact that the assignment of signs on the eigenvectors is arbitrary in the sense that reversing them on any eigenvector does not affect their orthogona] properties. The component that any eigenvalue and eiganvector contributes to the matrix [R] is now computed

$$[C1] \quad = \quad \lambda[x][x]^T$$

where $[x]^T$ is the eigenvector as a column vector, and λ is its associated eigenvalue. For the case of the first eigenvector and eigenvalue in Daultrey's example, [C1] is:

$$2.5636 \begin{bmatrix} 0.54212 \\ 0.62088 \\ 0.56623 \end{bmatrix} \begin{bmatrix} 0.54212 & 0.62088 & 0.56623 \end{bmatrix} \quad = \quad \begin{bmatrix} 0.75342 & 0.86288 & 0.78694 \\ 0.86288 & 0.98824 & 0.90127 \\ 0.78694 & 0.90127 & 0.82195 \end{bmatrix}$$

This matrix may be subtracted from [R] to produce a residual matrix containing only the effects of the second and third eigenfunctions. This produces, for this example:

$$[[R] - [C1]] \quad = \quad \begin{bmatrix} 0.24658 & -0.029581 & -0.20364 \\ -0.029581 & 0.011757 & 0.01543 \\ -0.20364 & 0.015430 & 0.17805 \end{bmatrix}$$

These matrices can be interpreted as showing, in the case of Mi], the correlations due to the first component alone, and in the case of [[R]-[C1]] the correlation matrix with the intercorrelations due to [CU removed. The same procedure can now be repeated to remove the effect of 1:02] from [[R]-[C1]]. In actual fact, and because the third eigenvalue is se small (0.014980), the residual matrix calculated above, and the matrix due to [C2], are both very similar. For the record the matrices for [C2] and [C3] are both given below. The reader should check a few selected elements to ensure that the method of construction is understood.

$$[C2] \quad = \quad \begin{bmatrix} 0.24470 & -0.025443 & -0.20638 \\ -0.025443 & 2.6455E-3 & 0.021459 \\ -0.20638 & 0.021459 & 0.17406 \end{bmatrix}$$

and

$$[C3] \quad = \quad \begin{bmatrix} 1.8790E-3 & -4.1376E-3 & 2.7380E-3 \\ -4.1376E-3 & 9.1112E-3 & -6.0291E-3 \\ 2.7380E-3 & -6.0291E-3 & 3.9896E-3 \end{bmatrix}$$

Note the use of exponential notation akin to that used in BASIC languages to allow for the small magnitude of the numbers (E-3 = 10^{-3}); in all cases figures are given rounded to 5 significant digits, and they may not agree in the]ast places with those

of Daultrey (1976). Just as the eigenvectors themselves are orthogonal, so are the matrices; that is [Ci][C2] = 0, etc. As the successive subtractions would lead us to suspect, it is the case that:

$$[R] = [a] + [C2] + [CM$$

and the reader should check this by straightforward arithmetic. For example:

$$r(1,1) = (0.75342 + 0.24470 + 0.00i8789) = 1.00000$$
$$r(3,1) = (0.78694 + (-0.20638) + 0.0027379) = 0.58330$$

to within the limits of the rounding errors at five significant digits. This is a rather different view of the matrix structure of [R] than the one usually presented, but it may nevertheless prove useful. In reverse it might be used to specify the correlation structure of [R] from hypothesized components, although it may not prove possible to ensure that such components are orthogonal to each other as they are constructed. If they are not orthogonal then a subsequent Component Analysis along the lines indicated here and in Daultrey (1976) would not recover them.

An altogether different method is available for reconstructing the original matrix using the eigenvectors and the eigenvalues. If [N] is a matrix (sometimes called the MODAL MATRIX) whose columns are the eigenvectors of a matrix, say [R], [λ] is a diagonal matrix (sometimes called the SPECTRAL MATRIX) whose successive terms are the eigenvalues of [R], then the following equation reconstructs [R] from its eigenfunctions:

$$[R] = [N][\lambda][N]^{-1}$$

This equation is often used in a rearranged form as an elegant statement of the basic eigenfunction problem: i.e. to find a matrix CN] such that:

$$[\lambda] = [N]^{-1}[R][N]$$

In other words, find a linear transformation, [N], that will, so to speak, squeeze the diagonal matrix of eigenvalues out of the given matrix [R]. It may be noted that in either case it does not matter what form the eigenvectors take: unit variance, all elements divided by the largest value, all elements summing to 1, or any other form: they all yield the same result. The first equation is the basis of an interpolation method in problems connected with projecting population growth in a series of regions, or in a series of age cohorts. If, for example, we take the diagonal matrix $[\lambda]^{0.5}$ which contains the square roots of the eigenvalues along the diagonal and substitute into $[N][\lambda]^{0.5}[N]^{-1}$ we get a matrix which on iteration with a vector will take two iterations to accomplish the same change that the original matrix will perform in one iteration. Hence the new matrix could be used to interpolate, or to extrapolate over a partial time period. Another use of the method would allow us to construct the correlation matrices for specific models.

As an example take first the small example that began this exposition. The eigenfunctions were $\lambda_1 = 3$ with vector $[1\ 1]$ and $\lambda_2 = -1$ with vector $[1\ -1]$.

Arranging these in column form to create the matrix [N], and using the formula given earlier for the inverse of a 2 x 2 matrix to get $[N]^{-1}$, the equation can be fleshed out as:

$$[R] \quad = \quad \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \quad = \quad \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Note that the computation proceeds from left to right, and the effect of the diagonal matrix is simply to multiply the ith column of [N] by the ith eigenvalue.

If we now take Daultrey's example with the eigenfunctions laid out as tabulated above we get a good approximation of the original [R] matrix, the numerical shortfall being due to using the figures rounded to 5 significant digits; if the fu]l accuracy of the computer solution for the eigenfunctions is used then [R] is accurately reconstructed. There is one advantage of using the eigenvectors 'in unit variance form; the inverse, $[N]^{-1}$, is merely the transpose of [N', i.e. $[N]^{-1} = [N]^{T}$, an equality that holds whenever a matrix is what is called ORTHOGONAL (ORTHONORMAL in some texts, e.g. Rummell 1970). An orthogonal matrix is one whose columns and rows all have inner products equal to zero. Written out in full $[N][\Lambda][N]^{1}$ is:

$$\begin{bmatrix} 0.54116 & 0.76202 & 0.35416 \\ 0.62088 & -0.079233 & -0.77989 \\ 0.56623 & -0.64269 & 0.51608 \end{bmatrix} \begin{bmatrix} 2.5636 & 0 & 0 \\ 0 & 0.42140 & 0 \\ 0 & 0 & 0.01498 \end{bmatrix} \begin{bmatrix} 0.54116 & 0.62088 & 0.56623 \\ 0.76202 & -0.079233 & -0.64269 \\ 0.35416 & -0.77989 & 0.51608 \end{bmatrix}$$

which computes out to, working from the left:

$$[R] = \begin{bmatrix} 0.99918 & 0.83237 & 0.58245 \\ 0.83374 & 1.00050 & 0.91715 \\ 0.58360 & 0.91704 & 1.00039 \end{bmatrix}$$

Given that correlation coefficients are usually only taken to two significant digits this is tolerable. An imaginative use of the transformation equation $[N][\Lambda][N]^{T}$ allows us to reconstruct the matrices [C1], [C2] etc, by using a diagonal matrix with λ_1 in its proper position, but the other diagonal elements set to zero, and similarly with λ_2 and λ_3, and λ_n in the n x n case. We can also alter the strengths of the eigenvalues, but in doing so it does not follow that the diagonal of [R] will be constrained to be equal to 1.0, as is appropriate for a correlation matrix.

Finally to conclude this section some minor technical matters that are useful to know. Occasionally, and especially in small examples or in structures with a lot of symmetry, repeated eigenvalues are found, *i.e. exactly* equal values. In this case

should not be taken as meaningful. Another case that may be encountered is that of a zero eigenvalue(s). Once more the corresponding eigenvector is meaningless, and the zero eigenvalue indicates that the Determinant of the original matrix is also zero since the product of all the eigenvalues of a matrix is equal to the Determinant: thus if any one is zero, so is the Determinant.

(iv) CONSTRUCTING THE INVERSE FROM THE EIGENFUNCTIONS

There is a connection between the eigenstructure of a matrix [A] and its inverse $[A]^{-1}$ which is useful to know. The inverse $[A]^{-1}$ has exactly the same set of eigenvectors as [A] itself, but the eigenvalues of $[A]^{-1}$ are the reciprocals of the eigenvalues of [A]. With this information we can construct $[A]^{-1}$ using the relation $[A]^{-1} = [N]^{-1}[\Lambda]^{-1}[N]$, where [N] and [Λ] are the relevant modal and spectral matrices.

As an example take [A] from (ii) above:

$$[A] = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$

Then

$$[\lambda] = \begin{bmatrix} 4.236067978 & 0 \\ 0 & -0.236067978 \end{bmatrix} \quad \text{and } [\lambda]^{-1} = \begin{bmatrix} 0.236067978 & 0 \\ 0 & -4.236067978 \end{bmatrix}$$

and it is a simple matter to check that $[\lambda][\lambda]^{-1} = [I]$, and to see that the diagonal elements of $[\lambda]^{-1}$ are merely the reciprocals of those in [λ]. Note that for the following calculations more exact values of the eigenvalues and eigenvectors are being used than were presented earlier.

Then the construction proceeds as follows:

$$\begin{bmatrix} 0.52573112 & 0.85065081 \\ 0.8506508 & -0.52573112 \end{bmatrix} \begin{bmatrix} 0.236067978 & 0 \\ 0 & -4.236067978 \end{bmatrix} \begin{bmatrix} 0.52573112 & 0.85065080 \\ 0.85065081 & -0.52573112 \end{bmatrix}$$

which will be found to be equal, to the first 5 significant digits, to

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

The reader should use the formula in section IV (iv) (c) to check that these values are the exact values in [A]-1 . In the expression above, $[N]^{-1} = [N]$ due to symmetry in the very small 2 x 2 problem, the matrices would be different in a larger example.

VI EXAMPLES AND APPLICATIONS

The following sections introduce many of the specific matrices and techniques discussed in the earlier sections, although in all cases greater depth will require the reader to pursue the topics in other CATMOGs or specialised texts. At all times the reader should be ready to re-read earlier sections if the approach to the materia] discussed below seems difficult or unfamiliar. As an aid the relevant sections are referenced.

(i) THE SOLUTION OF SYSTEMS OF EQUATIONS

This section will look first at exact solutions, i.e. where the number of known data points, n, exactly equals the number of coefficients required in the equation, and then at Least Squares solutions where n exceeds the number of coefficients required.

(o) Exact

Inhomogeneous systems

Systems of equations expressed in matrix form are solved by using the matrix inverse, and indeed this was one of the motivations suggested in the Introduction. To pursue that example we can take the formula for the inverse of a 2 x 2 matrix, given in section IV(iv)(d) to find that the required inverse is:

$$\begin{bmatrix} -1 & 2 \\ 20 & -20 \end{bmatrix}$$

From this the problem is solved by computing:

$$\begin{bmatrix} -1 & 2 \\ 20 & -20 \end{bmatrix}\begin{bmatrix} 15 \\ 10 \end{bmatrix} = \begin{bmatrix} 5 \\ 100 \end{bmatrix}$$

That is to say, the boundary between the two crops occurs when the price is 5 and the distance is 100km. If the market price should change to be **[16 10]** then the so]ution would become [4 120], and this solution is reached at the cost of merely one more matrix multiplication.

As another example we will take the problem of fitting a plane exactly to three data points, the simplest possible case of a linear Trend Surface (Unwin 1975 CATMOG 5). The data in the Table below show the height of the shoreline of pro-glacial Lake Iroquois, a late-glacial lake that occupied the Lake Ontario basin, at three different spatial locations. Because the shoreline is well defined the use of three heights to specify it is not as dangerous as might appear. The plane through these points defines the ancient water surface of Lake Iroquois, now tilted as a consequence of differential isostatic uplift. The data is taken from published NTS maps for Canada, the mixed units reflect that source, although I have adjusted the spatial coordinates by the removal of a constant quantity from each axis: a change in the origin of the coordinate system.

Table of data on pro-glacial Lake Iroquois

Site	z, Shoreline ht (ft)	U, Easting (km)	V, Northing (km)
Hamilton	365	0	14.8
Toronto	425	39	60.3
St Catharines	355	45.2	0

The solution required is to find z, the height, as a linear function of the spatial coordinates U and V:

$$z = a + bU + cV$$

For matrix solution this system is set up as follows, by substituting each data point into the equation:

$$\begin{array}{lllllll} 365 & = & a & + & b0 & + & c14.8 \\ 425 & = & a & + & b39 & + & c60.3 \\ 355 & = & a & + & b45.2 & + & c0 \end{array}$$

which is then reorganized using the formal rules of post-multiplication of a matrix by a vector (see IV (iii) b) as:

$$\begin{bmatrix} 365 \\ 425 \\ 355 \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 14.8 \\ 1.0 & 39.0 & 60.3 \\ 1.0 & 45.2 & 0.0 \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Notice the appearance of the column of is. These are, in effect, a specified set of coefficients attaching to the constant or intercept a, but which to save pedantry are not normally written in the usual a]gebraic presentation. However, their apppearance is essential in the matrix version to keep order and to keep actua] algebraic positions open. The system is represented compactly by the form:

$$[y] = [X][a]$$

where [y] is the vector of heights, [X] is the matrix of known spatial locations, and [a] is a vector of unknown coefficients. The solution is achieved by pre-multiplying both sides by [XT' **(see IV (iv) (a)):**

$$[X]^{-1}[y] = [X]^{-1}[X][a].$$

Since by definition MAX] = [D, (see IV (iv) (a)) and then [I][a] = [a] (see IV (iii) (e)), the solution can be expressed as:

$$[a] = [X]^{-1}[y]$$

This type of equation system, and its solution is termed INHOMOGENEOUS. With the use of the matrix inverse program given in the appendix the inverse may be found, and post multiplying by the vector [y], we get in full:

$$\begin{bmatrix} 347.6 \\ 0.164 \\ 1.178 \end{bmatrix} = \begin{bmatrix} 1.0348 & -0.25399 & 0.21915 \\ -0.022895 & 5.6193E-3 & 0.017275 \\ -2.354E-3 & 0.017162 & -0.014805 \end{bmatrix}\begin{bmatrix} 365 \\ 425 \\ 355 \end{bmatrix}$$

Thus the solution to the problem is that the plane is given by:

$$z = 347.6 + 0.164U + 1.178V$$

from which all the necessary information about the tilt of the old water surface may be extracted. **One may** argue that the solution could be reached far faster by straightforward elimination. In a particular case this may sometimes be true, but once we are in possession of the inverse (and if it is obtained by computer it is easy to store within the computer system as a **data** file) then repeated analysis for different values of the dependent variable (the left hand side of the equation) is very quick. For example, suppose that the vector of heights is re-examined and is determined instead to be [367 420 355] then a single post-multip]ication of the inverse already obtained (an option in the program provided) reveals that the new solution would be [350.9 0.09052 1.087]. In this way we can examine the sensitivity of the solution to variations in the dependent variable.

Thus the use of the inverse has some very specific virtues, and they can be seen from the very structure of the matrix solution: [X] is known to contain only the values of the independent variables U and V, and is separate from the dependent vector [yl. If [XT' is singular, i.e. the inverse fails to exist, then it is a sign of linear dependence in the original data comprising [X). This implies that the data for one location can be constructed as a linear combination of the other points: in more familiar geographical parlance it is an example of auto-correlation, albeit of an extreme kind. The solution would be to choose different points.

The solution form [y] = [X]-1[a] is extremely common in applications of matrix algebra, especially when it is realised that quite often the matrix [X] is composite, the result of various initial manipulations of the problem. Later sections will illustrate examples of this type.

Homogeneous systems

Some equation **systems are more awkward to solve** because re-organisation may involve **one side of the equation** equalling 0. For example in the case of eigenvectors the basic form of the equation is that:

$$[X][e] = \lambda[e] \qquad \text{(see V (ii))}$$

where [e] is an eigenvector and λ is its eigenvalue. If we have λ, then we obtain on reorganisation:

$$[X][e] - \lambda[e] = [0]$$

which can also be expressed as

$$[[X] - \lambda[I]][e] = [0].$$

Clearly setting $[[X]-\lambda[I]] = [A]$ and pre-multiplying by $[A]^{-1}$ would result in the trivial solution that [e] = [0], were it not for the fact that [A] is known to be singular if λ is an eigenvalue (and therefore the inverse doesn't exist in any case). As we saw in an earlier section (V (ii)), even when [e] is not equal to a zero vector there are an infinity of solutions, all multiples of each other. In this case it is necessary to fix one element of [e] at a set value. It is convenient to set the last value equal to 1, and then to obtain all the other elements as (unique) ratios with respect to it. In this instance it is necessary to reduce [A] by deleting the last (bottom) row, and to rewrite the last column (minus the bottom element) as a separate column vector, (-1)[r] = [-r], with the sign changed on all its elements, on the right hand side of the equation. Call the [A] so reduced [A-], call the last column of [M, with the signs changed and minus the bottom element, [-r], and call the first (n-1) elements of the vector [e-], then the initial system is written as:

$$[A-][e-] = [-r]$$

with solution.

$$[A-]^{-1}[-r] = [e-].$$

That is to say, the system has been reduced to an inhomogeneous set of equations that can be solved by finding an appropriate inverse. As a specific numerical example consider Daultrey's examp]e considered earlier (see V (iii)).

If we follow the equation above, for λ = 2.5636, then eliminating the bottom row, and moving the last column over to the right with a change of sign gives:

$$\begin{bmatrix} (1 - 2.5636) & 0.8333 \\ 0.8333 & (1 - 2.5636) \end{bmatrix} \begin{bmatrix} e(1) \\ e(2) \end{bmatrix} = \begin{bmatrix} -0.5833 \\ -0.9167 \end{bmatrix}$$

The use of the inverse then provides the solution

$$\begin{bmatrix} -0.89325 & -0.47605 \\ -0.47605 & -0.89326 \end{bmatrix} \begin{bmatrix} -0.5833 \\ -0.9167 \end{bmatrix} = \begin{bmatrix} 0.9574 \\ 1.0965 \end{bmatrix}$$

Recalling that the third element in the eigenvector was set to 1.0 by hypothesis then the eigenvector is [0.9574 1.0965 1.00] which may be scaled as required. The solution procedure is therefore the same as in the previous section, but the matrix is initially adjusted before the inverse is computed, and this will be typical of the use to which the inverse is put.

(b) *Least squares*

Geographical problems most often have the property that the number of data points available are in excess of the number of coefficients required in the equation that is being fitted to the data. The fit, however, will not be exact and the method of least squares is used to find a solution which assumes (amongst other things) that there is no error in the independent variables, the x's, and which minimises the sum of the squares of the differences between the actual y values and the predicted values. This is the only solution that is valid if an unbiassed estimate of y is required, but the least squares solution can also be used as a basis for computing alternative solutions that have other uses in terms of representing the general relation between the variables (Mark and Church 1977, Mark and Peucker 1978).

The final equation boils down to the same procedure as before: use the inverse to solve [y] = [[X][a] where [a] is the vector of unknown coefficients, [y] is the vector of terms including the cross products of the dependent variable y with the various independent x's, and [X] is the matrix containing the main elements of the Normal Equations. In detail, and as an example for a case where there are two independent variables, the algebraic and statistical problem is to find the coefficients in:

$$y = a + bx_1 + cx_2$$

where the number of data points is larger than 3, and where the Normal Equations are defined as:

$$\begin{aligned} \Sigma y &= aN + b\Sigma x_1 + c\Sigma x_2 \\ \Sigma yx_1 &= a\Sigma x_1 + b\Sigma x_1{}^2 + c\Sigma x_1x_2 \\ \Sigma yx_2 &= a\Sigma x_2 + b\Sigma x_1x_2 + c\Sigma x_2{}^2 \end{aligned}$$

In matrix terms these are written:

$$\begin{bmatrix} \Sigma y \\ \Sigma yx_1 \\ \Sigma yx_2 \end{bmatrix} = \begin{bmatrix} aN & b\Sigma x_1 & c\Sigma x_2 \\ a\Sigma x_1 & b\Sigma x_1{}^2 & c\Sigma x_1x_2 \\ a\Sigma x_2 & b\Sigma x_1x_2 & c\Sigma x_2{}^2 \end{bmatrix}$$

and in simple form:

$$[y] = [X][a]$$

tile will now see how matrix formalities can be used to obtain this form for solution from the initial data set. The initial data set will be set up like this, as a data matrix [D], for the subsequent matrix manipulations. I shal] use Unwin's data (CATMOG 5 1975), although the reader should note that for xi and x2 below Unwin uses x and y, and y below is Unwin's z, (the y* column is used later in this section):

	y	dummy	x_1	x_2	y* (see later)
	6	1	0	0	7
	8	1	1	8	7
	11	1	2	1	10
	12	1	3	1	13
[D] =	14	1	4	0	15
	12	1	2	2	10
	14	1	1	3	12
	12	1	0	4	13
	18	1	3	4	20
	22	1	4	4	25

The column labelled dummy is necessary to reproduce the Normal Equations. If we pre-multiply this data matrix [D] by its own transpose [M ^{T} the reader can check that the result will be to produce (in this case) a 4 x 4 matrix [S] which contains the following terms:

[[D]T[D] = [S] =

$$\begin{bmatrix} \Sigma y^2 & \Sigma y & \Sigma yx_1 & \Sigma yx_2 \\ \Sigma y & N & \Sigma x_1 & \Sigma x_2 \\ \Sigma yx_1 & \Sigma x_1 & \Sigma x_1{}^2 & \Sigma x_1x_2 \\ \Sigma yx_2 & \Sigma x_2 & \Sigma x_1x_2 & \Sigma x_2{}^2 \end{bmatrix}$$

Note that the dummy column enables the summations of the individual variables to be achieved, and in addition it produces the term N, the number of cases. Numerically the matrix is:

$$[D]^T[D] \quad = \quad [S] \quad = \begin{bmatrix} 1853 & 129 & 302 & 305 \\ 129 & 10 & 20 & 20 \\ 302 & 20 & 60 & 41 \\ 305 & 20 & 41 & 64 \end{bmatrix}$$

However, it is clear that the first row is surplus to the Normal Equations, except that the term Ey ^{2} can be used to obtain the correlation coefficient for the regression equation if that is required later. Thus the Normal equations are obtained by deleting the first row, and then writing the first column separately to the left hand side of the equals sign, as a vector [y] which contains the cross products of y with the various x's, in addition to its own summation. The matrix which remains after these alterations is [X], the terms including only the x's, and the term N, the number of cases. When, as would normally be expected, the manipulation is done by computer, it is an easy matter to delete the first row, to partition the matrix [S] into the two terms [y] and [X] as described above, and to add the vector of unknown coefficients [a]. The solution is completed in the normal fashion by finding (X]$^{-1}$ and post-multiplying by [y] to find [a].

In effect the problem has now become an exact one: find a vector of coefficients [a] which satisfies the Normal Equations exactly, and which also has certain desirable properties with respect to the original data matrix [U Therefore, the problem is now reduced to solving:

$$[y] \quad = \quad [X][a] \quad = \begin{bmatrix} 129 \\ 302 \\ 305 \end{bmatrix} = \begin{bmatrix} 10 & 20 & 20 \\ 20 & 60 & 41 \\ 20 & 41 & 64 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

With the use of the inverse given by the program in the Appendix:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.45073 & -0.096033 & -0.079332 \\ -0.096033 & 0.050104 & -2.0877E-3 \\ -0.079332 & -2.0877E-3 & 0.041754 \end{bmatrix} \begin{bmatrix} 129 \\ 302 \\ 305 \end{bmatrix} = \begin{bmatrix} 4.946 \\ 2.106 \\ 1.871 \end{bmatrix}$$

These results agree with Unwin who gives [4.95 2.11 1.87]. As was the case with the exact solutions obtained earlier, the virtue of the inverse is that it is available for further use. In the case of the Least Squares method there is one extra step, however. Because the vector [y] contains cross products of y with the various independent variables, the x's, these terms must be computed before another set of coefficients may be calculated. For example, suppose that Unwins's spatial data represents average annual rainfall at the given spatial locations. Then suppose that we wish to compare this surface with that for one particular year. The matrix [X] contains only terms pertaining to the number of locations, N, and the various terms including only the x's, the spatial locations. Thus to reuse [X]$^{-1}$ with this problem we need only compute a new vector, [y*7, using the column labelled y* in the table giving [D]. The necessary terms are specified in the first row (or column) of [S]. They may be computed by substituting column y* for column y in the table. Then we need only compute the first row of the new [S] (i.e. a vector times a matrix operation) since the terms with N and the x's are unchanged, and [y*] is obtained easily by deleting the first element of the first row. The reader should check that therefore [132 318 367] = [y*]. Then, in the usual way [a] = [X]$^-$1[^{y*}] = [-0.157 2.49 4.19]. This solution predicts (small) negative quantities of rainfall at the spatia] origin of the grid, despite an observed value of 7 units! However, since both [y] and [y*] are fictitious in this instance we need not be too upset, and it draws attention to the fact that the solution is not guaranteed to make physical sense, and often it is the residuals, the differences between the observed and the expected values, that are most useful in analysis. However, this is straying beyond the boundaries of this CATMOG.

(ii) MARKOV CHAINS

Markov chains were the subject of the very first CATMOG (Collins 1975) and most expositions find matrix algebra a convenient notation to describe them. Although one may start with a probability or transition matrix, [P], showing the probability of moving between states of the system being modelled, in most empirical applications it is usual to begin with a tally matrix [T] which records the number of moves between states of the system. The states of the system are indexed by the rows and columns of the matrix. For example, Collins uses the example of Lever (1972) whose initial tally matrix for a system of manufacturing businesses in five states: four spatial zones, and a 'birth' 'death' and 'reservoir' state, and recorded over a ten year interval, was:

$$[T] \quad = \quad \begin{array}{ccccc} \multicolumn{4}{c}{\text{- spatial zones -}} & \text{Reservoir} \\ \left[\begin{matrix} 118 \\ 6 \\ 1 \\ 2 \\ 17 \end{matrix}\right. & \begin{matrix} 13 \\ 33 \\ 1 \\ 0 \\ 24 \end{matrix} & \begin{matrix} 4 \\ 8 \\ 68 \\ 3 \\ 17 \end{matrix} & \begin{matrix} 14 \\ 6 \\ 5 \\ 43 \\ 36 \end{matrix} & \left.\begin{matrix} 63 \\ 20 \\ 24 \\ 17 \\ 906 \end{matrix}\right] \end{array} \qquad \text{Row sums} \begin{bmatrix} 212 \\ 73 \\ 99 \\ 65 \\ 1000 \end{bmatrix}$$

$$\text{Total} \quad = \quad 1449$$

Let [1] be a column vector of is (III (iv)) then [T][1] = [r], and [r] is a vector containing the row sums of [T]. Now let [D] be a diagonal matrix (III (v)) whose successive elements are the elements in [r], then [P], the transition matrix for the system, is given by:

[FI = [D]$^{-1}$ [T]

The reader should note that [D]$^{-1}$ is merely [D] with each diagonal element changed to its reciprocal (confirm it with the inverse program!) In this case [P] is:

$$[P] = \begin{bmatrix} 0.557 & 0.0613 & 0.0189 & 0.0660 & 0.297 \\ 0.0822 & 0.452 & 0.110 & 0.0822 & 0.274 \\ 0.0101 & 0.0101 & 0.687 & 0.0505 & 0.242 \\ 0.0308 & 0.0 & 0.0462 & 0.662 & 0.262 \\ 0.0170 & 0.0240 & 0.0170 & 0.0360 & 0.906 \end{bmatrix}$$

The matrix illustrates some typical aspects of empirical transition matrices: the diagonal values are large:- most 'moves' are 'stay puts', and most off-diagonal elements are small and close to zero. The two immediate possibilities for analysis are (a) to find the equilibrium or fixed state to which the system is tending as moves take place in accordance with the transition probabilities, and (b) given an existing distribution, what wil] be the distribution in the next few steps of the process?

The section on eigenfunctions illustrated that a step by step iteration of a vector through a matrix will, given time, tend to the fixed vector: the principal eigenvector. This is exactly the behaviour observed for transition systems, Markov Chains. Thus if we take a row vector M_t at time t, the subscript, then:

$$[v]_{t+1} = [v]_t[P]$$

and the second part of the problem is solved. Repeated iteration of the process solves the first part of the problem, and provides information on the nature of the convergence to equilibrium. However, in the spirit of the previous section it is possible to find the fixed (probability) vector, [N, directly if it is so wished. In this case we should like the vector to be such that the sum of its elements equals 1.0: i.e. MINT = 1.0. Some algebraic manipulation shows that if [1²] is the initial transition matrix then

$$\{[P]^T+[E]-[I]\}^{-1}[1] = [p] = [1][[P]+[E]-[I]]^{-1}$$

solves the problem. Note that the left hand side of the equation expresses the solution in the form used in the earlier section, but it requires that [P] be in its transpose form: another example of manipulation prior to solution. In the centre is [O which is a row or column vector at whim: an example of the fact, remarked upon above, that the row or column convention is primarily for notational convenience and has little intrinsic meaning of its own. Likewise [1] acts as a column vector on the left, as a row vector on the right. Note too that all the terms within the large brackets are matrices, [P] as discussed, [M is a matrix of ones, and ID is the identity matrix.

I shall not reproduce the full inverse but the reader should use the program to check that the fixed vector is found to be:

$$[p] = [0.0445\ 0.0386\ 0.0721\ 0.107\ 0.738]$$

Eventually therefore this fixed vector would predict that the distribution of firms to be approximately [65 56 104 155 1069] (multiplying by 1449 and rounding to the nearest whole number). The initial distribution of firms may be seen from the row sums of the original tally matrix IT], so that [v]1
95 1000], this being for 1959. The distribution for 1969 is found from:

$$[v]_2 = [v]_1[P] \quad \text{or} \quad [144\ 71\ 100\ 104\ 1030]$$

Once more the number of firms has been rounded to whole numbers. The most noticeable changes are the decrease in state 1 and the increases in states 4 and 5, the latter representing 'deaths' of businesses. These tendencies can be seen to be in the same direction as the proportions on the fixed vector would predict. An important point to note is the fact that the total number of firms is not allowed to change in this

computation, and this is because the principal eigenvalue of [P] is equal to 1, a point guaranteed by the fact that the row sums of (1²] are exactly equal to 1 by definition (see V (ii)). If genuine growth and decline is to be modelled then this requires additions to the matrix, usually on the main diagonal, a topic that will be explored a little further in the next section.

Straightforward iteration of $[v]_{t+1} = [v]_t[P]$ n times will give the vector $[v]_{t+n}$. However, an alternative method is to make use of the fact that by raising [P] to the nth power (see IV (iii)-a) we can compute the vector directly, and without the intermediate vector, thus:

$$[v]_{t+n} = [v]_t[P]^n$$

This method has an associated penalty: if n is not a simple power of 2, and because there are n³ calculations for every multiplication of two n x n matrices, there may be more calculations involved than in a direct iteration. However, a compensating benefit is the fact that [P]n contains direct information on the probability of n-step transitions between states. The reader may follow up interpretations of this type in Collins (CATMOG 1 1975) for Markov Chains, and in Tinkler (CATMOG 14 1977) for the case of access in networks represented by (0,1) adjacency matrices.

Collins also shows that using a result due to Kemeny and Snell (1967) many other properties of Markov Chains can be computed from the so-called Fundamental matrix, al

$$[Z] = [[I]-[P]+[A]]^{-1} \text{ where } [A] \text{ is given by } [1]^{-1}[p], \text{ and } [1]^{1} \text{ is a column vector.}$$

The latter relation depends on computing (p] from the expression given earlier. [A] is what is called the limiting matrix of (1⁵]: the matrix whose every row is formed from (p], the fixed vector of [P].

As a final remark, note that iterating vectors with [P]-1 will reverse the entire process, although' there is no guarantee that in reverse [O will remain non-negative, as would be required for a probability vector. The principal eigenvalue of (PT' will be 1, but if (Pr' has negative elements then [v] itself may become negative eventually.

(iii) POPULATION MATRICES AND INTER - REGIONAL POPULATION MOVEMENT

In the previous section Markov Chains were used to move quantities between states of a system subject to the strict proviso that the quantity under study remained exactly conserved. It neither grew nor declined. In many systems this is not a realistic requirement, for example in population systems viewed either as cohorts moving from one censal age group to the next, or between different regions of a spatial system. The following exposition is based on various accounts by Rogers (1968, 1971, 1975).

In the first instance consider a very simple closed population system with just four age groups. The term 'closed' means that there is no migration. Establish a matrix IS] which 'survives' a population age group from one census period to the next with a proportion that is equal to, or more usually less than, 1. For pedagogical convenience and to save space assume that age groups and census periods are twenty years, rather than ten years. Transition systems in general can be written either with flows indexed from the ith row to the jth column, or in the transpose form with the flows going from the jth column to the ith row. Mathematically the results are identical and the form adopted is either a matter of convenience or convention. The transpose form often saves space, and in this case is standard in the source material. Therefore the [S] is written as:

$$[S] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1s2 & 0 & 0 & 0 \\ 0 & 2s3 & 0 & 0 \\ 0 & 0 & 3s4 & 4s4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.9870 & 0 & 0 & 0 \\ 0 & 0.9733 & 0 & 0 \\ 0 & 0 & 0.8394 & 0.1411 \end{bmatrix}$$

The terms such as 2s3 indicate the survival proportion for the group moving from group 2 to group 3 during the census period. The 4s4 element allows people in group 4 to 'survive' in that group for more than one twenty year period, i.e. its a catch-al] group including some individuals who are over eighty. The proportions to enter in this table can be computed from life tables, or taken from the census. In this instance I shall invent some plausible values (on the basis of various examples given by Rogers). However, the [S] matrix includes only half the story. The proportions, being less than 1 allow for wastage. i.e. death. Births on the other hand have to be handled differently since they originate in different proportions from different age groups. All births obviously enter age group 1, and those that do enter survive with a common survival proportion to group 2. However, different age groups have different fertility rates because fertility is age specific. Call the matrix concerned with birth On Then it is written:

$$[B] = \begin{bmatrix} 1b1 & 2b1 & 3b1 & 4b1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.250 & 1.20 & 0.0412 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

with a notation similar to that for [S]. Again I have invented some plausible figures for the right-hand matrix. The element 111 represents teenage births which obviously enter the same age group in which they originate. Presumably the element 4b1 is zero in this system: no births from females over 60! Note that the birth porportions also include a survival component: they enter the age group as an age specific proportion of the originating age group; they then survive to the next age group with a survival proportion. Both these computations are included in the proportion in the table.

Growth in any population system is obviously a balance between survival (the polite form for death!), and birth. Since [S] and [B] account for these separately we can add them to obtain the population growth matrix [[G]:

[G] = 03] + ES]

so that:

$$[G] = \begin{bmatrix} 1b1 & 2b1 & 3b1 & 4b1 \\ 1s2 & 0 & 0 & 0 \\ 0 & 2s3 & 0 & 0 \\ 0 & 0 & 3s4 & 4s4 \end{bmatrix} = \begin{bmatrix} 0.250 & 1.20 & 0.0412 & 0 \\ 0.987 & 0 & D & 0 \\ 0 & 0.9733 & D & 0 \\ 0 & 0 & D.B394 & 0.1411 \end{bmatrix}$$

One demographic problem is to calculate the intrinsic growth rate for the entire population system, and to predict the structure of the population in future age periods as a function of the existing rates (as reflected in [G]), and of the numbers presently in each **age group. The matrix gives the proportions, and it is similar to a transition probability matrix in this respect. However, its row and column sums no longer equal 1, and so its principal eigenvalue is no longer constrained to be equal to 1. It has been shown by mathematical demographers that the intrinsic growth rate is given by the principal eigenvalue of [G], and the principal right-hand eigenvector shows the stable vector of population proportions by age** group, under the assumption that the rates in [G] remain unchanged. For [G] above it may be found by iteration that the principal eigenvalue is 1.234987, and the eigenvector, expressed as proportions summing to 1 is [0.343 0.274 0.216 0.168]. Although the eigenvalue seems large, it refers to a twenty year period: taking the twentieth root yields an annual growth rate of 1.0106, a rate of only Just over 1% a year.

The problem of population distribution in the next few censal periods may be studied by repeated post-multiplication of [G], starting with the present distribution:

[G(t+1)] = [G(t)][p].

Suppose that the distribution is [45 30 40 20], with sum 135, at the present (time 0), then the next three periods will be:

time 1 = [48.90 44.41 29.20 36.40] Sum = 158.91, est λ = 1.1711
time 2 = [66.73 48.26 43.23 29.65] Sum = 187.86, est λ = 1.1822
time 3 = [74.60 65.86 46.97 40.47] Sum = 227.90, est λ = 1.2131

The vector after three time periods may be compared, as proportion's, to those predicted by the eventual fixed vector:

time 3 as proportions [0.327 0.289 0.206 0.178]
stable vector [0.343 0.274 0.216 0.168]

and it is seen that the proportions are now quite close to the stable vector. As the vectors approach the steady form it is interesting to see that the growth rate, indicated above as the estimated λ, slowly approaches the equibrium value of 1.234987.

The model given above is a single region model, but with relatively little effort it may be extended to allow for migration between different regions. In this case the matrix for each component region must be estimated. I shall term it the [B+S] matrix, which is not to be confused with the [G] = [B] + [S] matrix, because in the S components the proportions also include a reduction *due not to death, but to emigration.* However there is now a much larger matrix in which the [B+S] matrix for each region appears along the diagonal, and an [M] matrix, for migration, appears in an off-diagonal position:

$$[G] = \begin{array}{c|cc} & 1 & 2 \\ \hline 1 & [B+S] & [2M1] \\ 2 & [1M2] & [B+S] \end{array}$$

In this formulation the numbers index the separate regions being considered, and 12M1] indexes migration from region 2 to region 1, and the growth matrix [G] represents the entire inter-regional system. The general [jMi] matrix is structured as follows:

$$[jMi] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1m2 & 0 & 0 & 0 \\ 0 & 2m3 & 0 & 0 \\ 0 & 0 & 3m4 & 0 \end{bmatrix}$$

with a similar notation and format to the [S] matrix used previously but with the exception that the m indicates a migration between regions which accompanies the change between age groups: i.e. it is a survival matrix for between region use, whereas [S] acts only within the region. Since the migrants deplete the [S] matrix age groups for the region they are coming from, the elements in [S] are adjusted to allow for this emigration. Rogers (1971) gives a small concocted example involving three regions,

which will be illustrated here. Large scale examples for a full range of age groups and several regions are very space-consuming, although they are no different in principle.

In order to save space in writing the matrix a common factor, i.e. a scalar equal to (1/12) = 0.083333, has been extracted from the matrix (see II (ii)). The regional partitions have been indicated to help the reader:

$$[G] = (0.083333)\left[\begin{array}{cccc:cccc:cccc}
0 & 9 & 10 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
0 & 6 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 \\
0 & 0 & 6 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 9 & 10 & 3 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 2 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 10 & 3 \\
2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0
\end{array}\right]\left[\begin{array}{c}
72 \\ 72 \\ 72 \\ 72 \\ \hdashline 72 \\ 72 \\ 72 \\ 72 \\ \hdashline 72 \\ 72 \\ 72 \\ 72
\end{array}\right]$$

The vector at the end of the matrix shows an equal distribution of population in all age groups, in all regions, as a starting state. Obviously, repeated post-multiplication by this vector will yield future states for the hypothetical system. The principal eigenvalue may be found to be λ = 1.12091, and the principal right-hand eigenvector, the eventual stable population distribution, is found to be [0.185 0.137 0.102 0.076 ⋮ 0.092 0.069 0.051 0.038 ⋮ 0.092 0.069 0.051 0.038], in probability form (i.e. summing to 1). As might be expected, since they have identical internal and external connections, the second and third regions have identical sub-vectors.

The eigenvalue is large, as in the previous example, but as was the case before it refers, even in this concocted example, to a lengthy age group, say of the order of 20 to 25 years, so that the actual annual growth rate would be as low as 0.4% to 0.6%. Of more interest is the variation in growth rates due to the initial population distribution, which in this case is perfectly even. Because feedback effects can only take place through births, which naturally all occur into the first age group(!), it takes some time for growth to stabilise, and with it the population distribution. The sequence of growth rates for successive twenty year periods is: 1.0833, 1.0256, 1.1302, 1.1674, 1.0808, 1.1383, 1.1201, after which the rate stays within a single percentage point of the eventual equilibrium value. The very low rate at the second iteration is due to the large number of individuals in the first age group resulting from the first iteration, and who cannot give birth in that time period. However, this large young component in the population then gives rise to high growth rates in subsequent iterations, before their increasing age reduces their fertility. It is a similar effect to the well-known baby-bulge phenomenon following World War II. Thus major disturbances in population systems, such as are caused by wars, famine, and migration take a considerable time to iron themselves out, and muted effects may be felt for decades after the initial disturbance.

(j1.1) INPUT - OUTPUT ANALYSIS

When an external demand is placed upon an economy, the total resulting output by the economy is larger than external demand itself. This results from the fact that in the working of the economy, each sector requires a certain amount, both from its own sector, and from each of the other sectors merely to produce the amount the external demand has placed upon it. The process is actually recursive, and infinite, since each extra internal demand generates additional internal demands in the system. Although infinite, the process generates (luckily!) only a finite amount of extra production, but as a consequence each sector needs to produce more than that required by the external demand. This outcome is usually termed the 'multiplier effect'.

The basic input-output model is usually set up from actual or estimated tables of accounts measured in a standard monetary unit. We will assume in a first, and elementary, example a two-sector economy composed of agriculture and industry, and an external demand called 'households'. The two-sector economy is represented by a 2 x 2 matrix, [T], of coefficients, usually called 'technological coefficients.' Each coefficient is less than 1, and it measures, in fractions of the basic monetary unit, the amount that must be purchased by the jth sector from the ith sector in order for the jth sector to produce one monetary unit's worth of output. The matrix of coefficients is normally read column-wise, and each column must sum to less than 1, or that sector will be purchasing as much in material worth as it is producing; a state of affairs that is clearly unprofitable.

As a simple example, therefore, consider the following matrix:

$$\begin{array}{ccc} & \text{A} & \text{I} \\ \text{A} & 1/6 & 1/5 \\ \text{I} & 1/3 & 3/5 \end{array} \quad \text{or} \quad \begin{array}{cc} \text{A} & \text{I} \\ 0.167 & 0.20 \\ 0.333 & 0.60 \end{array} \qquad \begin{array}{c} \text{H} \\ 100 \\ 50 \end{array}$$

The column vector [h] labelled H is the household demand which will be placed on the two-sector economy.

The computation of the total output of the economy proceeds as follows. Each element in the computation is represented as a post-multiplication of a matrix by the vector [hi The household demand forms the first component and is computed as:

$$[I][h] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 100 \\ 40 \end{bmatrix} = \begin{bmatrix} 100 \\ 40 \end{bmatrix}$$

However, as noted above, and from the matrix CT] this demand requires that for the agricultural sector to produce 100 units it must produce for its own consumption (116)th of that output - (0.167)(100) = i6.7, and it must also supply the industrial sector with goods of va]ue equal to (1/5)th of that sector's output - (0.20)(40) = 8. Exactly the same argument goes for the internal demands necessary to meet the industrial sector's output of' 40 units. It must buy (1/3)rd of the agricultural sector's output - (0.333)(100) = 33.3, and (3/5)th of its own output - (0.6)(40) = 24. Adding these together the additional output is (16.7 + 8) = 24.7 for the agricultural sector, and (33.3 + 24) = 57.3 for the industrial sector. It will be seen that this is exactly equivalent to computing CMChi

$$[h(1)] = \begin{bmatrix} 24.7 \\ 75.3 \end{bmatrix} = \begin{bmatrix} 0.167 & 0.20 \\ 0.333 & 0.60 \end{bmatrix}\begin{bmatrix} 100 \\ 40 \end{bmatrix}$$

which I will call [h(1)]; the results of the first round of the computation:If in doubt, the reader should write out the computation given above explicity, using letters for the coefficients of [T], in order to be sure that the identifications are correct and understood.

Obviously these internal demands cannot be met out of the household demand without reducing that demand. Consequently they constitute an additional demand on the system, a demand that has to be met in an identical fashion to that created by the household demand itself. Thus the arguments in the earlier paragraph have to be repeated, except that [h(i)] is used instead of [h]. However, I showed that the earlier argument was equivalent to computing the product (TM] = [h(1)], and so therefore [T][h(1)] = [h(2)], the next term in the series, is, on substitution, equivalent to computing:

$$[h(2)] = [T][T][h] = [T]^2[h]$$

The same argument applies to the problem of meeting the demand now created by [h(2)], so that in general the nth round of such demands is computed by the term:

$$[h(n)] = [T]^n[h]$$

and in principle the process continues *and infinitum.* Therefore the final demand, or total production [p] required of the economy, can be found by computing the sum of the infinite series:

$$[p] = [I][h] + [T][h] + [T]^2[h] + [T]^3[h] + \ldots + [T]^n[h] + ..$$

Since we know in practice that economies do not produce infinitely, we would suspect, correctly, that [O has a finite value. The obvious procedure is to compute the sum term by term, carrying out the powering indicated, and with the observation that it is permissible to factor out the common post-multiplying vector [h]:

$$[p] = \Big[\, [I] + [T] + [T]^2 + [T]^3 + \ldots + [T]^n + ..\Big]\, [h]$$

Doing this we get, following the rules of matrix multiplication (IV (iii) a):

$$\begin{bmatrix}150.07\\170.65\end{bmatrix} = \left[\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} + \begin{bmatrix}0.167 & 0.20\\0.333 & 0.60\end{bmatrix} + \begin{bmatrix}0.0945 & 0.1534\\0.2554 & 0.4266\end{bmatrix} + \begin{bmatrix}0.0699 & 0.1199\\0.1847 & 0.3070\end{bmatrix}\right]\begin{bmatrix}100\\040\end{bmatrix}$$

$$\begin{bmatrix}152.07\\170.65\end{bmatrix} = \begin{bmatrix}1.3314 & 0.4733\\0.7731 & 2.3336\end{bmatrix}\begin{bmatrix}100\\40\end{bmatrix} \quad \text{or } [p] = [S][h]$$

if the sum [S] is taken to just three rounds. The problem is - how many rounds should the computation be carried to? The simple answer is, to as many as cause the final sum to converge: i.e. not to increase by a significant amount as further rounds are added in For this to happen it is obvious that some power of [T] must eventually become virtually zero. From the numerica] example above it is clear that [T] is slowly getting smaller. In fact, the principal eigenvalue of [T] controls how fast [T] (or any matrix) grows, or in this case, declines. Without actually computing the eigenvalue we know from a result given in section V (ii) that the row and/or column sums limit λ. In this case λ must lie between the limits of the column sums which are 0.50 and 0.80 (the row sums limits are 0.367 and 0.933, which are less stringent), and in fact using the program in the appendix it may be found to be 0.7204. It is also the case that if λ is the eigenvalue of a matrix [A] then λ^2 is the eigenvalue of $[A]^2$. Thus in this case, and since λ is smaller than 1, some large enough power of [T] will have an eigenvalue that is close enough to zero that the term will contribute virtually nothing to the total sum.

This convergence of the matrix sum to a finite value is useful in itself, but it enables a more direct so]ution of the problem using the matrix inverse. By analogy to the rules for the convergence of power series with a common arithmetical multiplier, it may be shown that a matrix power series also converges, given the right conditions:

$$[S] = [I] + [T] + [T]^2 + [T]^3 + \ldots + [T]^n + ..$$

$$[S][T] = [T] + [T]^2 + [T]^3 + [T]^4 + \ldots + [T]^{(n+1)} + ..$$

$$[S] - [S][T] = [I] \qquad \text{(the subtraction is allowed since } [T]^n \text{ tends to } [0] \text{ as } n \text{ tends to infinity)}$$

$$[S]\big[[I] - [T]\big] = [I]$$

and then the solution using the inverse is:

$$[S] = \{[I]-[T]\}^{-1}$$

Numerically therefore:

$$\big[[I] - [T]\big] = \begin{bmatrix}(1.0 - 0.167) & -0.20\\-0.333 & (1.0 - 0.60)\end{bmatrix} = \begin{bmatrix}0.833 & -0.20\\-0.333 & 0.40\end{bmatrix}$$

and then the inverse may be found by program or from the formula in section IV (iv):

$$[S] = \big[[I-T]\big]^{-1} = (3.75)\begin{bmatrix}0.400 & 0.200\\0.333 & 0.833\end{bmatrix} = \begin{bmatrix}1.50 & 0.750\\1.25 & 3.125\end{bmatrix}$$

The problem may now be solved completely by using the original demand vector [h] as a post-multiplier:

$$[p] = \begin{bmatrix}180\\250\end{bmatrix} = \begin{bmatrix}1.50 & 0.750\\1.25 & 3.125\end{bmatrix}\begin{bmatrix}100\\40\end{bmatrix}$$

The whole problem may now be summarised as follows:

$$[p] = \big[[I]-[T]\big]^{-1}[h]$$

where the terms have already been defined in the paragraphs above. The problem is structurally very similar to other solutions involving the inverse. However, in this instance reversing the equation (multiply both sides by [[I]-[T]]-1), to express [h] as a function of [p], we get:

$$[h] = \big[[I]-[T]\big][p]$$

which is removed from the initial definition of the problem, which involved only [T] and [h]. The difference lies in the various rounds of demand which had to be computed and which led to the definition of [I-T] as the sum of all these rounds.

We may notice that the final solution indicates that the partial solution by successive sums was a long way short of the final convergence, although the speed of the convergence depends inversely on the principal eigenvalue of [T]: the smaller it is the faster the convergence, and some estimate of this can be obtained from the row or column sums, as we noted above.

Finally, we may inspect the [S] matrix to see the details of the multiplier process. First of all we may wish to subtract [I] since this merely ensures that [h] is contained in the final production [p]. Then we can see the way in which the different sectors of the economy are stimulated, either by themselves, or by other sectors. In this instance:

$$\big[[S] - [I]\big] = \begin{bmatrix}0.50 & 0.750\\1.25 & 2.125\end{bmatrix}$$

and it is clear that the industrial sector is the strongest element in the economy, despite the fact that the direct demand for its products is much smaller than for the agricultural sector. An [S] matrix may be examined element by element, or it may be aggregated as required as a basis for comparing different sectors, or groups of sectors. Obviously sector comparisons may be made by computing [1][S] or (1][[S]-[I]], which gives the column sums, and ISM] or [[S]-[t]][1] , which gives the row sums. The one indicates total purchases into the sector, the other total sales out of the sector.

Clearly, an identical approach to this can be used however many sectors there are in the economy, provided the data are available. Similarly the sectors may be split regionally to determine regional and sectoral multipliers. If we take the same system of technological coefficients and split them between two hypothetical regions in such a way that the purchasing between the regions by the different sectors reflects the different product mix in the different regions, then we may get a basic matrix such as:

$$\begin{array}{ccl}
 & & \begin{array}{cc:cc} \multicolumn{4}{c}{\text{regions}} \\ \multicolumn{2}{c}{1} & \multicolumn{2}{c}{2} \\ A & I & A & I \end{array} \\
\text{regions} & \begin{array}{cc} 1 & A \\ & I \\ \hdashline 2 & A \\ & I \end{array} & \left[\begin{array}{cc:cc} 0.10 & 0.15 & 0.05 & 0.15 \\ 0.20 & 0.40 & 0.10 & 0.0 \\ \hdashline 0.067 & 0.05 & 0.15 & 0.05 \\ 0.133 & 0.20 & 0.20 & 0.60 \end{array}\right] \left[\begin{array}{c} 50 \\ 20 \\ \hdashline 50 \\ 20 \end{array}\right]
\end{array}$$

In splitting up the system, each sector in each region shows the dollar amount purchased has remained what it was in the single sysem: 0.50 for agriculture, 0.80 for industry. However, it has been assumed that each sector will purchase more from its own region than from the other region, and in fact industry in region 2 purchases nothing from industry in region 1, and in region 2 the mix of products that agriculture buys is split differently from that in region 1, although the total outlay, 0.50 remains the same.

The solution of this system, for the (h] vector shown, is:

$$\left[\begin{array}{c} 100.88 \\ 80.50 \\ \hdashline 81.15 \\ 164.37 \end{array}\right] = \left[\begin{array}{cc:cc} 1.333 & 0.531 & 0.265 & 0.531 \\ 0.472 & 1.881 & 0.300 & 0.214 \\ \hdashline 0.177 & 0.225 & 1.266 & 0.225 \\ 0.766 & 1.230 & 0.871 & 2.896 \end{array}\right] \left[\begin{array}{c} 50 \\ 20 \\ \hdashline 50 \\ 20 \end{array}\right]$$

The failure of industry in region 2 to purchase from industry in region 1 is reflected in the low total production for that element: 80.50 compared to more than double the value, 164.37, for industry in region 2. Likewise agriculture in region 1 out-performs region 2 because industry in both regions purchases much more from agriculture in region 1 than region 2. The anomalous value of 1.23, for purchases by region 1 industry from region 2 industry reflects the assymetry in the original inter-regional purchases, whereby region 2 industry did not purchase from region 1 industry at all.

One major use of the [S] matrix is as a tool for exploring the effect of small changes in demand, while the technological coefficients are assumed to remain stable. For example what will be the impact of increasing the demand for agriculture in region 2 by 5 units, compared to the same increase for region 1 industry?

The easiest way to look at this is to use only the incremental vectors as post-multipliers of [SL an approach that yields the marginal productivity. For example, using [0 5 0 0] (region 1 industry) will produce a marginal output of [2.66 9.41 1.12 6.15] for a total increase of 19.34 and [0 0 5 0] (region 2 agriculture) will produce [1.33 1.50 6.33 4.36], for the much smaller marginal output of 13.52. Thus it is possible to trace the impact of particular inputs sector by sector and region by region.

Finally, the astute reader may notice that the total productivity of the two region system differs from that of the one region version, even though the total demand placed on the system, Eh], is the same. This arises from the variable pattern of inter-regional purchases, cumulated over the various rounds of buying. As a result agriculture, viewed over the entire system, produces 182.03, slightly in excess of the 180 in the single region system. On the other hand, industry as a whole produces a little less: 244,87 compared to 250 in the single region system. Total productivity is therefore reduced to 426.90 compared to 430.

VII CONCLUSIONS

It is usual to conclude CATMOGs with some comments pointing out both the limitations, and the potential extensions of the technique under review. In the case of a branch of mathematics, such as Matrix Algebra, it must be remarked that the methods outlined in this booklet are merely an introduction; the full resources of this methodology may be followed up in any of the texts mentioned in the bibliography. It is probably safe to say that there can be few "processes" that a geographer might wish to model, that have not already been explored by mathematicians, and it would be most unwise to suggest mathematics has limitations, especially ones that a geographer might readily encounter. The only problem is that the appropriate methods may be buried in advanced textbooks or research journals.

Perhaps the most important caveat that must be entered is that the user of these (and indeed all) mathematical methods must provide, for self and readers, a clear interpretation of the meaning, in terms of the problem, of the matrix methods employed. Considerable emphasis has been placed upon this viewpoint in the exposition above. Attention to it will, in itself, point up the limitations of the methods employed and suggest ways in which a more sophisticated model may be built. Rather than the mathematics having intrinsic limitations, it is far more probable that it is the data, or rather the lack of it, which acts to constrain the methods that can be applied.

VIII LITERATURE CITED

Carter, F.W., 1969. An analysis of the Serbian Oecumene: a theoretical approach. *Geografiska Atwater,* 51B, 1-39

Collins, L.F., 1975. An introduction to Markov Chain analysis. *CATMOG I,* (Geo Books, Norwich)

Daultrey, S., 1976. Principal components analysis. *CATMOG 8,* (Geo Books, Norwich)

Garner, B.J., and Street, W.A., 1978. The solution matrix: alternative interpretations. *Geographical Analysis,* 10, 185-90

Garrison, W., 1960. Connectivity of the Interstate Highway System. *Papers and Proceedings of the Regional Science Association,* 6, 121-37

Goddard, J.B., 1970. Functional regions within the city centre: a study by factor analysis of taxi flows in central London. *Transactions of the Institute British Geographers,* 49, 161-182

Gould, P.R., 1967. On the geographical interpretation of eigenvalues, *Transactions of the Institute British Geographers.* 42, 53-86

Hadjey, G., 1961. *Linear algebra.* (Addison-Wesley, Reading, Mass.)

Harman, H.H., 1967. *Modern factor analysis.* (University of Chicago Press, Chicago) 2nd Ed.

Hay, A., 1977, Linear Programming: elementary geographical applications of the transportation problem. CATMOG 11 , (Geo Books, Norwich)

Hohn, F.E., 1972. *Elementary matrix algebra.* (MacMillan, New York) 3rd Ed.

Isard, W., 1960. *Methods of regional analysis.* (MIT Press)

Keyfitz, *N.*, 1968. *Introduction to the mathematics of population.* (Addison-Wesley, Reading, Mass.)

Killen, J., 1979. Linear Programming: the Simplex method with geographical applications. *CAT/106 24,* (Geo Books, Norwich)

Mark, D.H. and Church, M., 1977. On the misuse of regression in earth science. *Journal of the International Association for mathematical geology,* 9, 63-75

Mark, D.M. and Peucker, T.K., 1978. Regression analysis and geographical models. *The Canadian Geographer,* 22, 51-64

Nystuen, J. and Dacey, H.F., 1961. A graph theory interpretation of nodal regions. *Papers and Proceedings of the Regional Science Association,* 7, 29-42

Pitts, F.R., 1965. A graph theoretical approach to historical geography. *Professional Geographer,* 17(5), 15-20

Rogers, A., 1968. *A matrix analysis of inter-regional population growth and distribution.* (University of California Press, Berkeley)

Rogers, A., 1971. *Matrix methods of urban and regional analysis.* (Holden Day, San Francisco)

Rogers, A., 1975. *An introduction to multiregional and mathematical demography.* (John Wiley, New York)

Rummel, R.J., 1970. *Applied factor analysis.* (Northwestern University Press, Evanston)

Searle, S.R., 1966. *Matrix algebra for the biological sciences.* (John Wiley, New York)

Scott, A.J., 1971. *Combinatorial programming, spatial analysis, and planning.* (Methuen, London)

Soja, E., 1968. The geography of modernization in Kenya. *Syracuse Geographical Monographs,* No 2

Stephenson, L.K., 1974. On functional regions and indirect flows. *Geographical Analysis,* 6, 383-85

Tinkler, K.J., 1972. The physical interpretation of eigenfunctions of dichotomous matrices. *Institute British Geographers,* **55, 17-46**

Tinkler, K.J., 1976. On functional regions **and indirect flows.** *Geographical Analysis,* 8, 476-492

Tinkler, K.J., 1977. An introduction to **graph theoretical methods in geography.** *CATMOG 14,* (Geo Books, Norwich)

Tinkler, K.J., 1979. A comment on the solution matrix. *Geographical Analysis, 11,* **97-99**

Unwin, **D.,** 1975. An introduction to trend surface analysis. *CAT 110G 5,* (Geo **Books,** Norwich)

Varga, R.S., 1965. *Matrix iterative analysis.* (Prentice **Hall, New Jersey)**

Wilson, A.G. and Rees, P.R., 1977. *Spatial population analysis.* **(Edward Arnold, London)**

APPENDIX 1

PLEASE READ THESE NOTES

These programs are written in APPLESOFT, for an Apple 11+ with 48K, and a 40 column screen. There are no graphics, input is through the keyboard, and output is to the screen. Thus they should be easily converted to other dialects of BASIC, if the following notes are borne in mind.

IN APPLESOFT

1) **HOME** clears the screen and returns the cursor to the top left corner.
2) The colon, : , allows multiple statements on a line.
3) The semi-colon, ; , causes printing at that location and overrides the automatic tabulation of APPLESOFT. SPC(3) inserts 3 spaces in the output, etc., **SPC(n)** would insert n spaces.
4) Dimensions need not be dimensioned via the **DIM** statement unless they exceed **10.** Since indexing starts at 0 this actually provides **11** elements in each array dimension. NO dimension statements are provided in these two programs. See notes in (a) below.
5) APPLESOFT allows two-dimensional matrices. Other BASICS may not.
6) APPLESOFT is highly compatible with MBASIC running under C1^{9}111 on Apples, and these programs should run as they are printed in MBASIC.

(a) COMPOTE THE INVERSE OF A MATRIX

This program allows entry of the matrix to be inverted through the keyboard and is essentially menu-driven. A number of options are available to alter the input matrix to that needed for several of the uses to which the inverse is put. This saves you having to make the changes manually before you begin, and thereby reduces the possibility of errors. The user may wish to adapt the program to print its answers, and to read and write from disk.

A self-check of the answer is provided, i.e. [[A][$^{A]-1}$ is computed and compared to [t], the results reported, and the user may inspect both the inverse and the computed [l] matrix. Computations are in single precision so USE this check facility. Inversion may be unstable for large matrices, which is why dimensions are limited to 11. (You could easily change this, see the notes above). MBASIC in Apple CP/M is compatib]e with this program and offers double precision, see appropriate manuals.

The program allows you to enter repeated vectors to post-multiply the inverse. If this is not your wish it returns you to the matrix entry option.

If the matrix is SINGULAR it stops. Otherwise use CTRL C to exit.

```
4 HOME : PRINT "MATRIX INVERSION - GAUSS/JORDAN METHOD": PRINT :
      PRINT "THERE IS A CHANCE TO CHANGE THE INPUT": PRINT "MATRIX, A,
      TO THE FORM": PRINT :
      PRINT "INV((LAMBDA*I)-A) -> I=IDENTITY MATRIX"
6 PRINT : PRINT : PRINT : PRINT : PRINT
7 INPUT "(HIT RETURN TO CONTINUE)";A$
8 HOME : INPUT "HOW MANY ROWS IN THE MATRIX ? ";N
10 FOR I = 1 TO N
15 HOME
20 PRINT "ROW ";I
25 PRINT
30 FOR J = 1 TO N
40 INPUT A(I,J)
41 B(I,J) = A(I,J):AI(I,J) = 0.0
45 NEXT J:AI(I,I) = 1.0: NEXT I
50 REM NOW THE MAIN INVERSION ROUTINE
54 RV = 1
55 HOME : PRINT "DO YOU WISH TO COMPUTE THE INVERSE": PRINT "IN EITHER
      OF THE FORMS": PRINT : PRINT "   INV ((I*LAMBDA)-A) ?    PRESS 1": PRINT "
      INV (A-(LAMBDA*I)) ?    PRESS 2"
56 PRINT : PRINT "OTHERWISE PRESS RETURN": PRINT : INPUT A$: PRINT :
      IF A$ = "" THEN GOTO 100
57 IF A$ = "2" THEN RV = - 1
58 INPUT "ENTER LAMBDA ";LA
59 FOR I = 1 TO N: FOR J = 1 TO N
60 A(I,J) = - 1.0 * (A(I,J)):B(I,J) = A(I,J)
61 NEXT J: NEXT I
63 FOR I = 1 TO N
64 A(I,I) = A(I,I) + LA:B(I,I) = A(I,I)
65 NEXT I
66 FOR I = 1 TO N: FOR J = 1 TO N:A(I,J) = RV * A(I,J):B(I,J) = A(I,J):
      NEXT : NEXT
100 FOR I = 1 TO N
120 IF (A(I,I) < > 0) THEN GOTO 240
130 FOR IX = 1 TO N
150 IF (A(IX,I) = 0) THEN GOTO 180
160 IP = IX
170 GOTO 200
180 NEXT IX
190 GOTO 900: REM MATRIX IS SINGULAR
```

```
200 FOR IX = 1 TO N
210 A(I,IX) = A(I,IX) + A(IP,IX)
220 AI(I,IX) = AI(I,IX) + AI(IP,IX)
230 NEXT IX
240 P = A(I,I)
250 FOR J = 1 TO N
260 AI(I,J) = AI(I,J) / P
270 A(I,J) = A(I,J) / P
280 NEXT J
290 FOR K = 1 TO N
300 IF (K = I) THEN GOTO 360
310 XM = - A(K,I)
320 FOR IC = 1 TO N
330 A(K,IC) = A(K,IC) + XM * A(I,IC)
340 AI(K,IC) = AI(K,IC) + XM * AI(I,IC)
350 NEXT IC
360 NEXT K
370 NEXT I
380 FOR I = 1 TO N
390 FOR J = 1 TO N
400 IF (I = J) AND (A(I,J) < > 1.0) THEN GOTO 900
410 IF (I < > J) AND A(I,J) < > 0
      THEN GOTO 900
420 NEXT J: NEXT I
500 REM CHECK THAT A*AI=I
503 HOME
505 PRINT "CHECK MATRIX : SHOULD BE VERY CLOSE TO THE IDENTITY MATRIX.
      A CHECK IS MADE BY THE PROGRAM AND FOLLOWS THE CHECK MATRIX ": PRINT
510 FOR I = 1 TO N: FOR J = 1 TO N:A(I,J) = 0
530 FOR K = 1 TO N
540 A(I,J) = A(I,J) + B(I,K) * AI(K,J)
550 NEXT K: NEXT J: NEXT I
551 DF = 1E - 6:S = 0
553 FOR I = 1 TO N
554 IF ( ABS (A(I,I) - 1.0) > DF) THEN S = S + 1
555 FOR J = 1 TO N: IF J = I THEN GOTO 557
556 IF ( ABS (A(I,J)) > DF) THEN S = S + 1
557 NEXT J: NEXT I:NS = N * N
560 FOR I = 1 TO N
570 PRINT
580 PRINT : PRINT "ROW ";I
590 FOR J = 1 TO N
600 PRINT A(I,J); SPC( 1);
605 NEXT J: NEXT I
610 PRINT
611 PRINT : PRINT S" ELEMENTS OUT OF ";NS;" OR ";S * 100 / NS;"%
      DIFFER FROM THE IDENTITY MATRIX ELEMENTS BY MORE THAN ";DF: PRINT
612 INPUT "DO YOU WISH TO CONTINUE (Y/N) ? ";A$
613 IF A$ < > "Y" THEN GOTO 4
614 PRINT : INPUT "DO YOU WISH TO SEE THE INVERSE (Y/N) ? ";A$
615 IF (A$ < > "Y") THEN HOME : GOTO 690
```

```
616 HOME
630 PRINT "MATRIX INVERSE"
640 FOR I = 1 TO N
645 PRINT
650 PRINT · PRINT "ROW ";I
660 FOR J = 1 TO N
670 PRINT AI(I,J); SPC( 1);
680 NEXT J: NEXT I
685 PRINT : PRINT : INPUT "(RETURN TO CONTINUE)";A$
686 HOME
690 PRINT : INPUT "DO YOU WISH TO ENTER A VECTOR TO POST-MULTIPLY THE
      INVERSE (Y/N) ? ";A$
700 IF A$ < G "Y" THEN GOTO 4
705 PRINT : PRINT
710 PRINT "NOW ENTER THE VECTOR OF N ELEMENTS "
715 PRINT
720 FOR I = 1 TO N
730 INPUT R(I): NEXT I
740 FOR I = 1 TO N
750 V(I) = 0
760 FOR J = 1 TO N
770 V(I) = V(I) + R(J) * AI(I,J)
780 NEXT J· NEXT I:
785 HOME
890 PRINT "THE RESULT IS": PRINT
800 FOR I = 1 TO N
810 PRINT V(I); SPC( 1);
820 NEXT I
830 PRINT : PRINT : GOTO 690
800 HOME : PRINT "THE MATRIX IS SINGULAR"
810 STOP
```

(D) COMPUTE EIGENFUNCTIONS OF A REAL SYMMETRIC MATRIX

This program allows entry of the matrix to be analysed through the keyboard, with output to the screen. Write down all the answers you need as they appear on the screen, or add your own writing and/or saving routines. Although the ana]ysis is designed for a real symmetric matrix, it will find the first real (left-hand) eigenvector of a real asymmetric matrix: that is it can be used to find the fixed vector of a probability (transition) matrix.

Iteration is performed under keyboard control and may be terminated when the eigenvalue and/or eigenvector estimates displayed on the screen are stable enough. The program operates using the 'power and deflate' method. It is NOT programmed to stop after N eigenfunctions, which has the merit that the residual amount still left, and due to accumulated rounding errors in single precision, may be inspected. In APPLESOFT use CTRL C to escape.

```
4 HOME: PRINT "EIGENVALUES AND EIGENVECTORS OF A":PRINT
         "REAL SYMMETRIC MATRIX":PRINT:PRINT "CAN BE USED
         FOR FIXED VECTOR OF A":PRINT"PROBABILITY MATRIX":
```

```
         PRINT:PRINT:PRINT
5 PRINT:PRINT:PRINT:
6 INPUT "(HIT RETURN TO CONTINUE)";A$
7 HOME: INPUT"HOW MANY ROWS IN THE MATRIX ? ";N
8 FOR I=1 TO N:HOME:PRINT "ROW ";I:PRINT:FOR J = 1 TO N
10 INPUT A(I,J):NEXT:NEXT
50 S2 = 0:S1 = N
90 FOR I = 1 TO N:V1(I) = 1.0: NEXT
100 FOR I = 1 TO N:S = 0: FOR J = 1 TO N
110 S = S + V1(J) * A(J,I): NEXT
120 V2(I) = S:S2 = S2 + S: NEXT
130 ES = S2 / S1
140 HOME
150 PRINT "EIGENVALUE ESTIMATE ";ES
160 PRINT
170 FOR I = 1 TO N:V1(I) = V2(I) / ES: PRINT V1(I): NEXT
175 PRINT : PRINT
180 PRINT :S1 = N:S2 = 0
190 PRINT "ANY KEY TO CONTINUE ITERATING"
192 PRINT : PRINT "N TO TABULATE CURRENT EIGENFUNCTION"
200 GET A$
220 IF A$ < > "N" THEN GOTO 100
230 REM UNIT VARIANCE
240 S = 0
250 FOR I = 1 TO N
260 S = S + V1(I) * V1(I): NEXT
270 S = SQR (S)
280 FOR I = 1 TO N:V1(I) = V1(I) / S: NEXT
281 HOME
282 PRINT "EIGENVALUE "K,ES: PRINT
283 FOR I = 1 TO N: PRINT V1(I): NEXT
284 PRINT : PRINT : PRINT "ANY KEY TO CONTINUE, X TO STOP"
285 PRINT : PRINT "(NB X IS NEEDED EVENTUALLY)"
286 GET A$
287 IF A$ = "X" THEN STOP
290 REM EXTRACTS THIS FACTOR FROM A
300 FOR I = 1 TO N: FOR J = 1 TO N
310 A(I,J) = A(I,J) - (ES * V1(I) * V1(J))
320 NEXT : NEXT
330 HOME
331 PRINT : PRINT "MATRIX OF FACTOR ";K: PRINT
332 FOR I = 1 TO N: PRINT : FOR J = 1 TO N
335 PRINT (ES * V1(I) * V1(J)): NEXT : NEXT : PRINT
336 GET A$
337 HOME : PRINT "RESIDUAL MATRIX ": PRINT
338 FOR I = 1 TO N: PRINT : FOR J = 1 TO N: PRINT A(I,J): NEXT : NEXT
339 PRINT : GET A$
340 K = K + 1
350 GOTO 50
```

APPENDIX 2

MATRIX MANIPULATION USING MINITAB

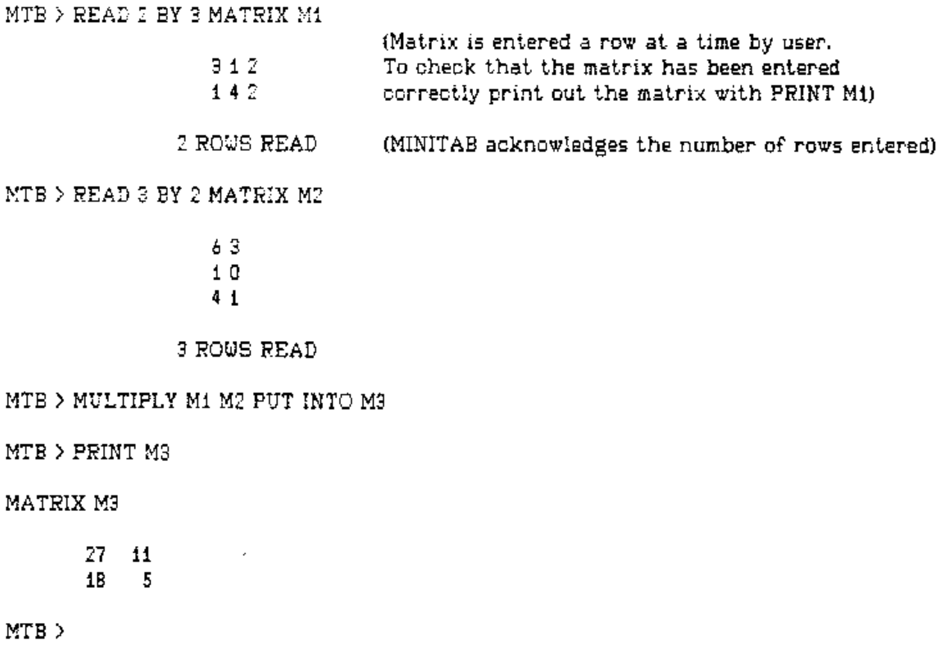
MINITAB is a very user-friendly statistical package, developed at Pennsy]vania State University. For those who have access to it on mainframe computers it may provide a useful alternative to the micro-computer programs in Appendix 1 . You should enquire of your Computer Laboratory if it is installed on your mainframe. Please note that what f allows is merely illustrative of what MINITAB can do; it does not in any sense constitute a manual on MINITAB, it is more like a typical session.

Let us assume that it is available, then if you log in and call MINITAB you can read in a matrix using the READ command. Suppose, for instance we wanted to multiply the matrices A and B (see page 16 and 17). The MINITAB prompt MTB > can be followed by the command:

REAL 2 by 3 matrix M1 (return)

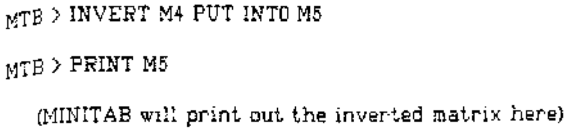
snd the two rows of three numbers entered in free format. Note that you have to give the dimensions (rows and columns) of the matrix and to label the matrix. The label must begin with M, followed by a number, Unless you are attempting something very involved it is unlikely that this number will exceed 15!

So, for the multiplication we have



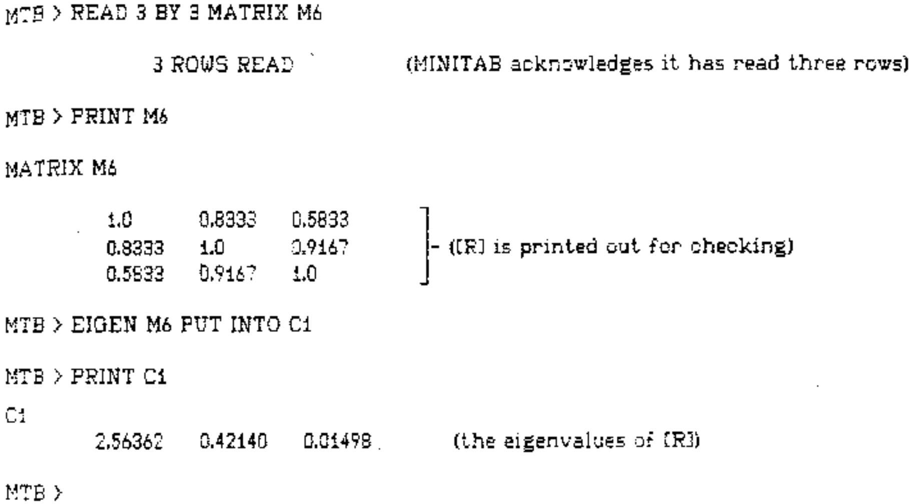
```
MTB > READ 2 BY 3 MATRIX M1
                                (Matrix is entered a row at a time by user.
            3 1 2               To check that the matrix has been entered
            1 4 2               correctly print out the matrix with PRINT M1)

          2 ROWS READ           (MINITAB acknowledges the number of rows entered)

MTB > READ 3 BY 2 MATRIX M2

            6 3
            1 0
            4 1

          3 ROWS READ

MTB > MULTIPLY M1 M2 PUT INTO M3

MTB > PRINT M3

MATRIX M3

     27   11
     18    5

MTB >
```

Inversion of a matrix is easy. Assume you have read in a matrix M4. Then



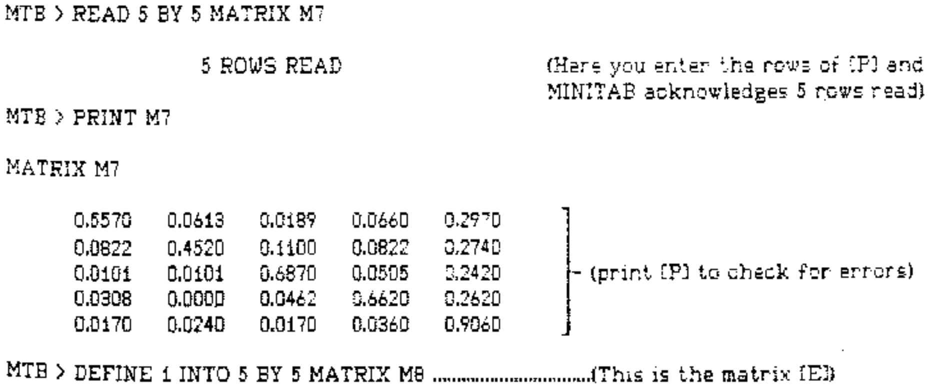
```
MTB > INVERT M4 PUT INTO M5

MTB > PRINT M5

  (MINITAB will print out the inverted matrix here)
```

To calculate eigenvalues, use the EIGEN command and store the results in a column. For instance, using the matrix [R] on page 29:



```
MTB > READ 3 BY 3 MATRIX M6

            3 ROWS READ              (MINITAB acknowledges it has read three rows)

MTB > PRINT M6

MATRIX M6

         1.0      0.8333   0.5833   ]
         0.8333   1.0      0.9167   |- ([R] is printed out for checking)
         0.5833   0.9167   1.0      ]

MTB > EIGEN M6 PUT INTO C1

MTB > PRINT C1

C1
       2.56362    0.42140    0.01498        (the eigenvalues of [R])

MTB >
```

Finally, we demonstrate how matrix manipulations in Markov Chain analysis msy be performed. Refer to the description of how to calculate the fixed probability vector (pages 39-40). In brackets below on the right are the descriptions of what is being. done.



```
MTB > READ 5 BY 5 MATRIX M7

              5 ROWS READ                    (Here you enter the rows of [P] and
                                             MINITAB acknowledges 5 rows read)
MTB > PRINT M7

MATRIX M7

      0.5570   0.0613   0.0189   0.0660   0.2970   ]
      0.0822   0.4520   0.1100   0.0822   0.2740   |
      0.0101   0.0101   0.6870   0.0505   0.2420   |- (print [P] to check for errors)
      0.0308   0.0000   0.0462   0.6620   0.2620   |
      0.0170   0.0240   0.0170   0.0360   0.9060   ]

MTB > DEFINE 1 INTO 5 BY 5 MATRIX M8 .................................(This is the matrix [E])
MTB > SET C1...........................................................(Type 5(1) in response)

MTB > DIAGONAL C1 PUT INTO M9 ....................................(This creates matrix [I])

MTB > ADD M7 TO M8 PUT INTO M10 ..................................([[P] + [E]])

MTB > SUBTRACT M9 FROM M10 PUT INTO M1..........................( [[P] + [E] - [I]] )

MTB > INVERT M1 PUT INTO M2

MTB > PRINT M2

MATRIX M2

    -1.69841    0.21032    0.70755    0.47119    0.50935   ]
     0.25165   -1.39729    0.26059    0.41187    0.67221   |
     0.55216    0.42438   -2.24229    0.56295    0.90248   |- ( [[P] + [E] - [I]]^-1 )
     0.42916    0.45759    0.57163   -2.00482    0.73683   |
     0.49874    0.34356    0.77459    0.66613   -2.08336   ]

MTB > COPY C1 INTO M4..............................................(defines a column vector of 1s)

MTB > TRANSPOSE M4 PUT INTO M5...............................(then transposes to a row vector)

MTB > MULTIPLY M5 BY M2 PUT INTO M3.........................(Multiply by inverted matrix)

MTB > PRINT M3.....................................................(print [p], the fixed vector)

MATRIX M3

      0.044513    0.038567    0.072069    0.107325    0.737505
```

MINITAB will tell you if you are attempting an inadmissable matrix operation. You can also send results to a file if you do not wish to transcribe them from the screen.

MINITAB reference manuals are available if your installation supports the package, A MINITAB Student Handbook is published by Duxbury Press.

35. The agricultural census - United Kingdom and United States - G. Clark
36. Order-neighbour analysis — G. Aplin
37. Classification using information statistics - R.J. Johnston & R.K. Semple
38. The modifiable areal unit problem - S. Openshaw
39. Survey research in underdeveloped countries - C.J. Dixon & B.E. Leach
40. Innovation diffusion: contemporary geographical approaches - G. Clark
41. Choice in field surveying - Roger P. Kirby
42. An introduction to likelihood analysis - Andrew Pickles
43. The UK census of population 1981 J.C. Dewdney
44. Geography and humanism - J. Pickles
45. Voronoi (Thiessen) polygons - B.N. Boots
46. Goodness-of-fit statistics - A. Stewart-Fotheringham & D. C. Knudsen
47. Spatial autocorrelation - Michael F. Goodchild

This series is produced by the Study Group in Quantitative methods, of the Institute of British Geographers.
For details of membership of the Study Group, write to the Institute of British Geographers, 1 Kensington Gore, London SW7 2AR, England.

The series is published by:

Geo Books,
Regency House,
34 Duke Street,
Norwich NR3 3AP,
UK,

to whom all other enquiries should be addressed.