ENVIRONMENTAL IMPACT ANALYSIS:

THE IDENTIFICATION

OF SECONDARY IMPACTS

BY

Jonathan B Shopley

A thesis submitted in partial fulfilment of a Master of Science degree in the School of Environmental Studies, University of Cape Town,

June, 1981.

ENVIRONMENTAL IMPACT ANALYSIS : THE IDENTIFICATION OF SECONDARY IMPACTS

Table of contents

ABSTRACT

The need for a preliminary environmental impact analysis approach, able to identify secondary impacts, has been revealed by a broad literary review. Therefore, the component interaction technique has been developed which is able to structure a preliminary investigation of secondary impacts. The technique is based on a component interaction matrix. The environment is modelled as a list of environmental components, and direct dependencies between these components are then recorded. Computerized matrix powering procedures are able to structure the data to facilitate the investigation of the secondary impact potential in the system. By virtue of its construction, the technique ensures that a preliminary analysis of impacts is based on a comprehensive and structured consideration of the environmnent. The procedure can also be used to substantiate and control the subjective content of an impact study. These two attributes of the technique support its application in conjunction with other methods of impact analysis. Various extentions to the technique have also been considered.

ACKNOWLEDGEMENTS

I acknowledge with thanks the assistance of the following:

Professor RF Fuggle, who first suggested the topic, and commented on the manuscript;

Professor D Matravers, who provided advice on mathematical aspects, and commented on the manuscript;

Roy Stauth, who assisted greatly with the detailed estuarine component interaction matrix, and commented on the component interaction technique;

Anne O' Sullivan, who assisted with the typescript and proof-reading;

The "Academic Support" staff of the Computer Centre, University of Cape Town, who gave advice and assistance on computing matters;

The Council for Scientific and Industrial Research, for a post-graduate bursary for two years;

The South African Breweries, for a post-graduate bursary for two years.

PREFACE

The typescript of this report was produced on a computer (using the DOC processor of the Sperry Univac 1100 system). Much of this thesis deals with computerized approaches in environmental impact assessment, and it is held that computers can make a positive contribution to a structured analysis of environmental impacts. The scripting of this thesis is one example of the many diverse advantages which computer usage could offer to the field of impact assessment. The limitations in typographical features (e.g. no italics, underlining, or subscripts) and format control (in a few instances) are offset by the efficient means of creating, editing and reproducing the document.

1. CHAPTER 1 - INTRODUCTION

1.1. Introduction to the Topic

Environmental impacts are a by-product of human activities undertaken to meet the physical and emotional requirements of man. Although modern societies are increasingly able to manipulate the environment to meet their needs for food, shelter, and security, it is apparent that the true cost of such actions normally involves some reduction in environmental quality. Therefore the demand for commodities to support the human population conflicts with the desirability of not degrading or destroying environmental resources. Analyses of environmental impacts are performed to provide the insight required by society to resolve this conflict of interests (Matthews, 1975,p122).

A broad, but useful definition of environmental impact assessment is given by Munn (1975,p23) as "an activity designed to identify and predict the impact on man's health and well-being, of legislative proposals, policies, programs, projects and operational procedures, and to interpret and communicate information about the impacts".

As the term "environmental impact assessment" has such broad connotations, it has been redefined and other related terms have been introduced (Fuggle, 1979,p5).

Environmental impact assessment (EI assessment) is described by Fuggle (1979,p5) as the administrative process by which the environmental impact of a project is determined. EI assessment procedures, approaches, or methodologies are the conditions pertaining to the activity of conducting an environmental impact investigation (i.e. the terms of reference by which the impacts of a project are investigated, presented, and finally, considered by decision-makers). The nature of the process is normally defined by social values, administrative constraints, and legal provisions.

Environmental impact analysis (EI analysis) is a process contained in EI assessment, by which the environmental effects of a project are analysed. A method of EI analysis describes a complete activity for analysing impacts for an EI assessment. Clark et al (1980,p111) reserve the term "technique" for specialized procedures within EI analysis which evaluate (rather than identify) impacts. Generally, techniques will be taken as procedures which are only able to prescribe to some portion of an EI analysis.

The documentary reports resulting from a particular EI analysis or assessment are termed the environmental impact statement (EI statement). This component of the EI assessment projects the process into the decision-making arena.

Most EI analysis methods and techniques concentrate on impacts arising as a direct consequence of the proposed project. However, impacts indirectly caused by a project (secondary impacts) may be of equal importance, yet are rarely afforded adequate consideration. The focus of this report will be the identification of secondary impacts for EI assessment.

1.2. The Development of Environmental Impact Assessment

The field of environmental impact assessment was pioneered in the

United States of America as a response to the far reaching provisions of the National Environmental Policy Act (NEPA) of 1970 (Anderson, 1973,p49). One of the clauses of NEPA requires all federal agencies of the U.S.A. "to identify and develop methods and procedures, ... which will insure that presently unquantified environmental amenities and values may be given appropriate consideration;" (U.S.A. Congress, 1970).

Various methods and techniques have been developed to facilitate compliance with NEPA. Important examples are the map overlay method of McHarg (1971, originally 1968), the matrix method of Leopold (1971), and Sorensen's network method (1971). A number of authors (Warner and Preston, 1974,p4; Jain and Urban, 1975,p11; Munn, 1975,p117; Clark et al, 1978a,p28; Fuggle, 1979,p6), in reviewing the development of EI assessment, concur that an adequate analysis method should perform all of the following four tasks :

* impact identification
* impact measurement
* impact interpretation
* impact communication to information users

Considering the complexity of the interacting systems that constitute the environment, and the infinite variety of possible impacting actions, it seems unlikely that a single method would be able to meet all the above criteria (Jameson, 1976,p1/11; Holling, 1978,p57). The general applicability of all methods also has to be balanced against the values of the society and administrative constraints within which they are employed (Blisset, 1976,p268).

1.3. Environmental Impact Assessment in South Africa

Catlow and Thirlwall (1976,p7) have noted that methods used for EI analysis in the U.S.A. have evolved in response to the legislative requirements of NEPA, and are sensitive to the decentralized decision-making processes in that country. They contend that it would be inappropriate for the United Kingdom to adopt an American system of EI assessment. However, Holling (1978,p57) recognizes that various methods and techniques can be satisfactorily combined to perform individual EI assessments. He feels that this adaptive approach could be successfully applied in developing countries, as many existing methods could be tailored to the needs and conditions of a particular assessment (1978,p18).

In South Africa, a White Paper (Department of Water Affairs, Forestry, and Environmental Conservation, 1980,p6) has signalled parliamentary intent to incorporate some form of EI assessment into a national policy on the environment. However, South Africa has a highly centralized and strictly bureaucratic public administration system (Fuggle, 1980,p78 and 81). This creates a need for methods and techniques which can best present environmental considerations within this framework. If it is recognized that EI assessment is not a decision-making process in itself, but rather an input (by means of the EI statement) to the decision-making process (Catlow and Thirlwall, 1975,p134; Munn, 1975,p30; Fuggle, 1979,p3 and 6), the responsibility of interfacing with the bureaucratic administration makes the efficient communication of impacts an especially important criterion for any South African methodology.

EI Assessment in the U.S.A., to a large extent, is separate from the planning and development of a project (Holling, 1978,p16).

Many authors feel that a procedure which is more integrated with the design phase would be preferable (Fischer and Davies, 1973,p210; Jones, 1975,p72; Catlow and Thirlwall, 1976,p49; Clark et al, 1978b,p120; Hollick, 1981b). Environmental interests are better served by an early recognition of possible impacts. Although it has been suggested that a design interactive approach would be suitable in the South African context (Fuggle, 1979,p3), its success would rest largely upon the ability of the assessment methods used to identify impacts.

Fuggle (1979) has developed two methods for preliminary impact assessment in South Africa. These are adaptations of the Leopold matrix and McHarg map overlay systems. The strength of these methods (especially the matrix method) is that they concentrate on impact identification, impact communication, and are particularly suited for incorporation in the planning and development phases of projects. However, the recognition of impacts by all but the most sophisticated or specialized methods is complicated by the difficulty of identifying secondary impacts.

1.4. Secondary Impacts in Environmental Impact Assessment

EI Analysis has been defined as a process aimed at the recognition of causes and effects; a cause being any action of the proposed project which has an effect upon the environment. These effects are the environmental impacts of the action. Any effect in the bio-physical and socio-economic environments that arises from a cause directly related to the project is termed a first order or primary impact.

Secondary impacts are those effects on the bio-physical and

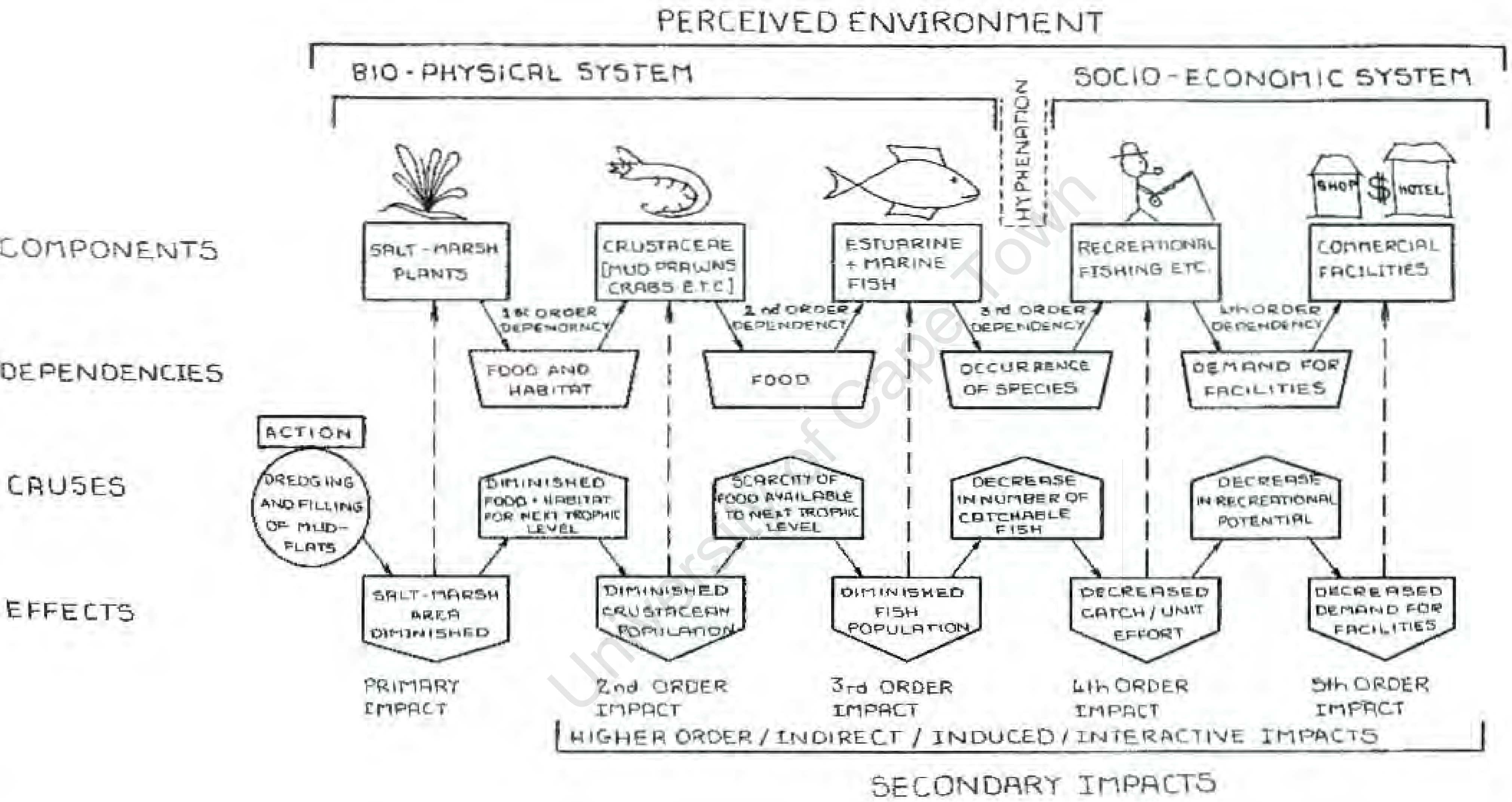


FIGURE 1. Tracing the secondary impacts which could arise from the dredging and filling of an estuarine mud-flat.

socio-economic environments which arise from an action, but which are not initiated directly by that action. Their occurrence is defined by the interdependencies which exist within and between the two systems.

Figure 1 symbolically traces the secondary impacts which could arise from the dredging and filling of an estuarine mud-flat (supposing that the proposed project was the construction of a marina). The top row of rectangles show a set of environmental components linked by various dependencies (second row of "boxes"). The way in which the primary impact of dredging and filling the mud-flats affects all the components is shown as a progression of causes and effects (in rows three and four). Note that the "commercial facilities" component of the socio-economic system has a fourth order dependency on "salt-marsh plants". Similarly the action of dredging and filling the mud-flats has a fifth order impact upon "commercial facilities".

Various authors have used different terms to describe secondary impacts (see figure 1), but most are compatible with the above definition. Jain and Webster (1977,p267), however, use the term "higher order impacts" to mean secondary impacts as given above, while reserving the terms "indirect" or "secondary" to cover impacts resulting from an induced action. While secondary impacts (as defined for this study) are propogated by existing linkages in a system, induced secondary impacts occur when a project introduces new linkages, allowing a now connected component to impact the adjoining system. For example, a new highway passing near an existing town may generate a link between highway traffic and town commerce. If town growth (induced action) were stimulated by commercial activity, the resultant effects would be part of the induced secondary impacts of the highway project.

Dependencies in the bio-physical environment are normally described by food, habitat, and other defined relationships. Therefore, the roots of induced secondary impacts are mostly confined to the socio-economic environment.

The distinction between secondary, and induced secondary impacts is important, and this thesis will adopt the term "induced action" per se, but will use "induced secondary impacts", where necessary, for impacts arising from induced actions in the socio-economic environment.

Jameson (1976,p1/4) and FitzPatrick et al (1978,p2) adopt the U.S.A. Environmental Protection Agency definition of secondary impacts: "indirect or induced changes in population and economic growth and land use, and other environmental effects resulting from these changes in land use, population, and economic growth". Emphasis is on secondary impacts and induced secondary impacts in the socio-economic sphere. Jones (1975,p71) in addressing secondary impacts of engineering projects also leans towards this interpretation.

Munn (1975,p120), Catlow and Thirlwall (1976,p16), Ross (1976), Clark et al (1978a,p28 and 1978b,p112), and Ward (1978,p1) place an emphasis on secondary impacts in the bio-physical environment, relating these to the defined interdependencies existing within the natural environment.

Isard et al (1972,p93), Jain et al (1977,p68), and Erickson (1979,p94) have recognized that dependencies exist between the socio-economic and bio-physical environments, and that these allow actions affecting the one to initiate secondary impacts in the other. These linkages between the social and natural environments have been termed hyphenations (Gold, 1978,p105).

A view common to all these authors is that secondary impacts constitute a significant proportion of the total impact of a project, and therefore need to be adequately assessed.

Although the matrix method proposed by Fuggle appears suitable for use in South Africa, it is, unfortunately, a recognized failing that such (matrix) methods are, in themselves, unable to consider secondary impacts (Munn, 1975,p5; Chase, 1976,p141; Clark et al, 1978a,p35).

1.5. Aims and Objectives

It is the aim of this thesis to address the problem of identifying secondary impacts for EI assessment. The major objective is to develop an EI analysis technique taking the following considerations into account :

- the need for an analytical technique which can identify environmental interdependencies and subsequently, secondary impacts.
- the technique should be adaptable to a wide range of EI analysis methods, but should, in particular, be compatible with the matrix method described by Fuggle.
- the technique should be simple but explicit (considering the state of the art of EI assessment in South Africa), with a minimum reliance upon research resources.
- two important functions of the technique should be the identification (as opposed to evaluation) and communication of secondary impacts.

- the technique should be available to designers and planners as an aid to the early identification of environmental impacts.

A technique developed in accordance with these criteria may not always be suitable, because, as EI assessment attains higher levels of sophistication over time (as can be expected in South Africa), concordant analytical procedures will be needed. Therefore, it is envisaged that an hierarchy of possible extentions and alternatives to the proposed technique be presented.

1.6. Approach

The chapter following (chapter two) is a review of EI methods and techniques. Attention is focused on procedures which give attention to secondary impacts. There is also a subordinate emphasis on the use of matrices in EI analysis. This is in support of the technique proposed in chapter three. The technique is an adaptation of the component interaction matrix first used by the Lands Directorate, Environment Canada (1974) for the EI assessment of the Port Nanaimo trans-shipment facility. Appendix A is a listing of the ASCII FORTRAN computer programme written to perform the mathematical functions of the technique, while Appendix B contains examples displaying the capabilities of the programme.

The applicability and efficiency of the technique is analysed in chapter four, and chapter five contains suggested extentions and alternatives, should a more comprehensive assessment of secondary impacts be required. Conclusions are drawn in chapter six.

2. CHAPTER 2 - A SELECTIVE REVIEW OF METHODS AND TECHNIQUES

2.1. Introduction

In this review, the great variety of assessment methods and techniques are divided into eight broad categories, and representative examples from each are given. Although attention is focused on approaches which assess secondary impacts, the concluding discussion investigates the possibility of extending the capabilities of methods which are not able to address higher order impacts. A case is presented for an adaptive approach to EI analysis, where some of the diverse range of methods and techniques are selectively combined to meet the conditions of particular assessments.

2.2. Review

2.2.1. Review categories

Most authors who have reviewed the development of EI analysis (Cook, 1977,p16; Jain et al, 1977,p73; Clark et al, 1978a,p27) adopt the following categorization of procedures attributed to Warner and Preston (1974,p3) :

ad hoc; checklists; matrices; networks; and overlays

Jain et al (1977,p74), have added "combination computer-aided methods" to the list; a category which is covered by "adaptive methods" in this study. As it is intended to cover both methods and techniques, two further categories are included. The first, modelling procedures, covers computerized and other types of models; the second, evaluation techniques, represents procedures which are able to aggregate a number of observations (specifically impact evaluations) into an normalized record or score.

The categories of EI analysis approaches are therefore :

1 ad hoc approaches
2 checklists
3 matrices
4 networks
5 overlays
6 modelling procedures
7 evaluation techniques
8 adaptive methods

Each of these is briefly described and discussed below.

2.2.2. Ad hoc Approaches

Ad hoc methods (perhaps the oldest and crudest approach to EI analysis) were widely used by U.S.A. Federal agencies immediately after the introduction of NEPA. The Environmental Guidelines of the Western Systems Co-ordinating Council (1971, cited by Jain et al, 1977,p90) is an example of this approach. The stipulations of NEPA are incorporated in broad guidelines which suggest areas of possible impact, without recommending specific means for their measurement or evaluation.

Although ad hoc methods do not generally address secondary impacts, the Environmental Protection Agency's "Manual for Evaluating Secondary Impacts of Wastewater Treatment Facilities" (FitzPatrick et al, 1978) is an exception. The approach is very specific to sewage projects and is structured around U.S.A. legislation. Although the manual could provide useful guidelines for studies outside of the United States, it does lack generality.

Procedures of this sort would probably not be widely used outside their originating environments.

One attribute of these early methods that has been adopted by later more rigorous methodologies is the formulation of a multidisciplinary team of experts to conduct the EI analysis.

2.2.3. Checklists

Checklists are a more formalized version of ad hoc approaches in that specific areas of potential impact are listed, and instructions are supplied for impact identification and evaluation. Where qualitative or quantitative evaluations are attempted, each impact area is associated with a list of environmental parameters (also termed characteristics, variables, attributes, or components), and parameter data is measured to reflect the degree of impact. Four classes of checklist methods have been identified by Canter (1977,p199) related to the level of impact evaluation (i.e. none; descriptive; scaling; and weight-scaling). The first two, however, are similar and may be combined (Cook, 1977,p18).

Simple and Descriptive Checklists

These differ from ad hoc methods only in that defined areas of possible impacts are listed, but no attempt is made to qualitatively or quantitatively evaluate impacts. An example cited by Canter (1977,p199) is the checklist of the U.S.A. Department of Transport (1971) in which the impacts associated with transportation projects are described.

Scaling Checklists

Scaling checklists allow listed impacts to be ranked in order of magnitude or severity. In some cases scores can be aggregated. This is permissible when impacts are scored on an interval or ratio scale (Skutsch and Flowerdew, 1976,p213).

It is difficult to meet the criteria of an interval or ratio scale when evaluating diverse impacts, even if they are associated with physical parameters. For example, an action which causes the loss of half the population of a species is not necessarily half as serious as one which leads to total extinction. In other words, although impact magnitude may be a function of some quantifiable attribute, the relationship need not be linear. Further, it is not always possible to relate physical parameters to all types of impact. The impact on a scenic view, for example, is not readily quantified in any terms. Consequently, scaling checklists tend to rely upon the subjective assignment of numerical values.

The Adkins and Burke method (of 1971, cited by Jain et al, 1977,p78) is a scaling checklist which scores impacts on a ranked scale of -5 to +5. The method assumes that the conditions of an interval scale are met, hence the number of negative scores are aggregated and average impact scores are used to compare project alternatives.

Aggregating by the arithmetic addition of impact scores affords an equal weight to each impact. However, while the impact on a species of grass may be as severe as that on a species of bird, it may not be equally important. This disparity is not accounted for by scaling checklists.

Weight-scaling Checklists

The environmental evaluation system developed by the Batelle Columbus Laboratories (Dee et al, 1973) is a weight-scaling checklist method which was favourably compared to many other methods in the Warner and Preston review (1974).

Seventy-eight environmental parameters are weighted by a multidisciplinary team using a method of fractionation (1000 units are distributed among the parameters in accordance with their relative importance). A normalized numerical value is derived for each parameter by the use of a value function which transforms parameter measures into environmental impact units (see figure 2 for examples of a scaling function). The delphi technique

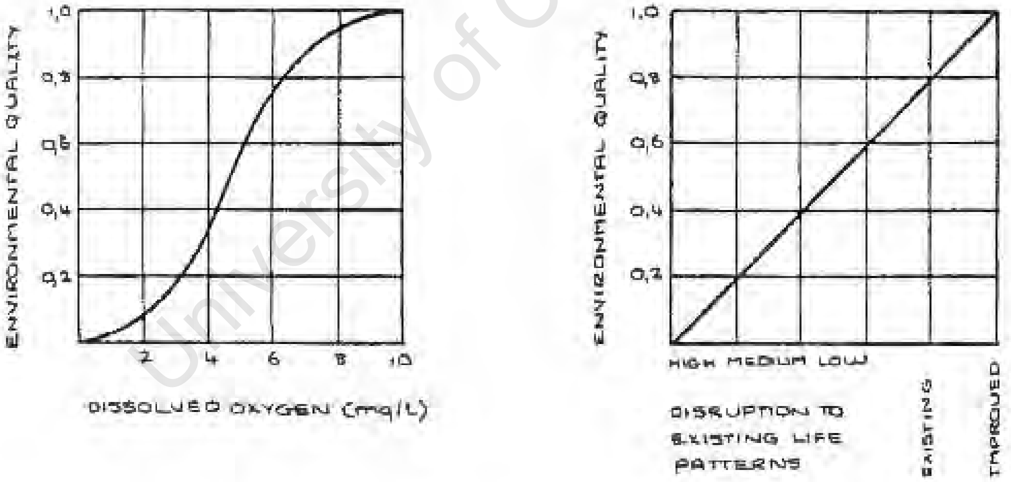


FIGURE 2. Two value functions from the environmental evaluation system of Dee et al.

(discussed under "evaluation techniques"), is used to encourage a consensus of opinion among a team of assessors on relative weighting, impact scores, and the form of the value functions. A normalized impact evaluation is obtained by multiplying the importance weight by the impact score, and summing the products over all impacts. The resultant index can be used to compare the "no project" situation with various project alternatives. The idea of high-lighting key impacts using a "red flag" system is a necessary extention to the approach, as much information is lost to decision-makers by the high degree of aggregation (Bisset, 1978,p52).

2.2.4. Matrices

The matrices most often used in EI analysis are grid diagrams in which two distinct lists are arranged along perpendicular axes. The presence of an interaction between components on opposing axes is marked and scored in some manner in the cell common to both. These scores are not normally operated upon mathematically, but there are techniques which utilize the mathematical rather than presentational characteristics of matrices.

2.2.4.1. Presentational Matrices

The best known use of presentational matrices is the EI analysis method developed by Leopold (1971) for the U.S.A. Geological Survey (Cook, 1977,p21). A matrix is used to summarize and display the interactions between a list of project actions and environmental characteristics. The lists are related on a basis of cause and effect. Where a project action is recognized to have an effect on an environmental characteristic, the appropriate matrix cell is scored for potential impact magnitude and significance (see figure 3).

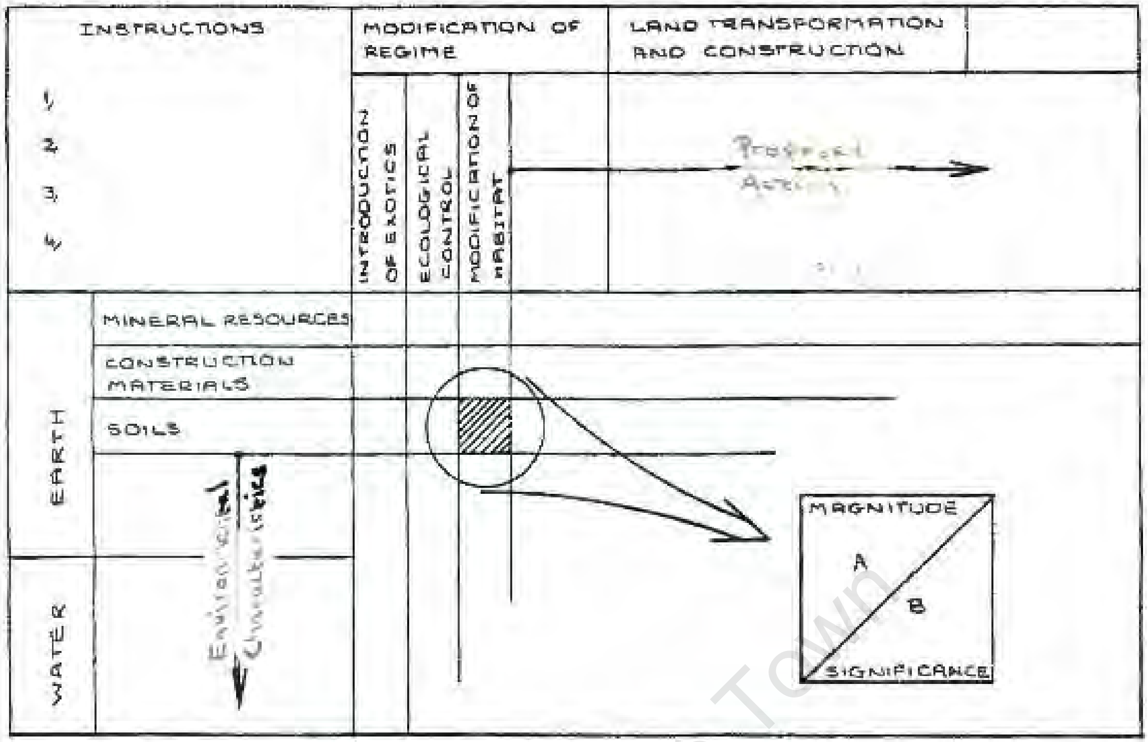


FIGURE 3. Illustrative diagramme of a Leopold impact matrix.

Many adaptations to the original Leopold matrix have been made (Clark et al, 1978b,p112). Chase (1976,p142) has listed five varieties of matrices based on the method of representing impacts:

1) Descriptive. Short written descriptions are used. The land use analysis matrix of Manning and Moncrief (1979) is an example where two or three words are inserted in the appropriate matrix cell to indicate the comparative degree of potential impact.
2) Symbolized. The draft environmental impact statement contained in the Boston Transportation Planning Review (of 1972, cited by Chase, 1976,p145) uses square and round symbols in a matrix to distinguish between direct and indirect impacts. The shapes are shaded to indicate the severity of the impact.

3) Characterized. Chase (1976,p143) cites the environmental matrix used by the Delaware River Basin Commission in which characters are used to rank impacts. This is an appropriate representation for impacts scored on an ordinal scale, as it avoids any quantitative significance.

4) Numeric. Ordinal and interval scaled evaluations are given by numerical scores. Leopold (1971) uses a scale of 1 to 10 to score two impact attributes; significance and importance. Fischer and Davies (1973,p219) use one score on a scale of -5 to +5 to indicate both positive and negative degrees of impact. An interval scaled impact matrix has been attempted by Ross (1976; discussed under mathematical matrices).

5) Combinative. Impacts are represented by both numeric and non-numeric indicators. The method proposed by Fuggle (1979,p23) uses each matrix cell to assess potential impacts in terms of importance, probability, time of occurrence, duration, benefit, effect of remedial measures, and risk.

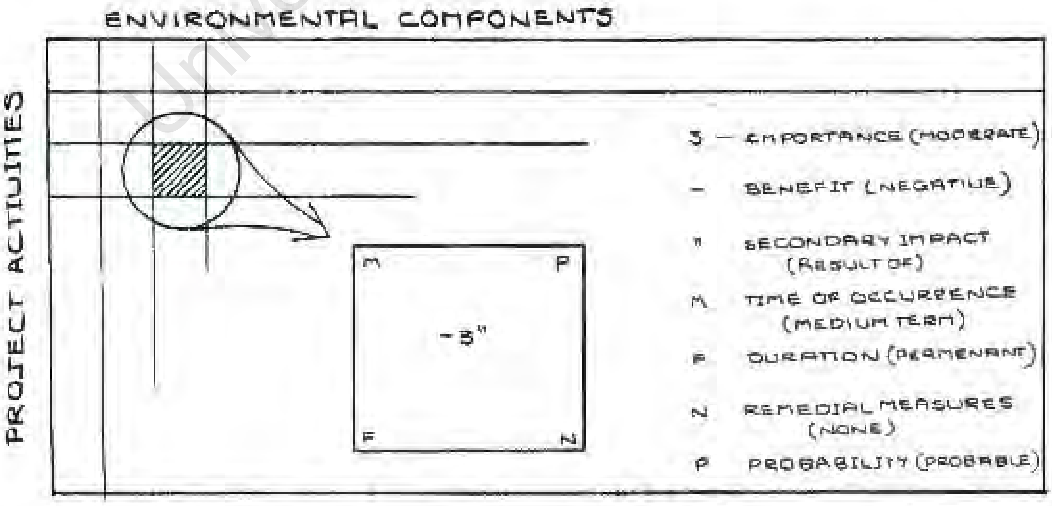


FIGURE 4. Illustrative diagramme of the impact matrix proposed by Fuggle.

Impact importance is subjectively evaluated on a scale of 1 to 5, while all other attributes are represented by character descriptors (see figure 4). An indicator of secondary impacts is optional to the method.

Other changes to the Leopold matrix have involved the compilation of the axes' checklists. Although Leopold's matrix stipulates lists of eighty-eight environmental characteristics and a hundred project actions, later methods (Fischer and Davies, 1973; Fuggle, 1979) introduce a greater degree of flexibility by designating the choice of project actions and environmental characteristics to a multidisciplinary assessment team. Fuggle's matrix is proposed as a preliminary method for EI analysis. Typical of other extended matrix techniques, it has the following advantages :

* it presents an easily understood summary of a large number of primary impacts.
* a generalized but well defined approach forcing a comprehensive consideration of environmental components and primary impacts.
* an easily performed process which can specify the overall character of a project early in the design phase.
* in an extended form, the method can include information about many impact attributes, and clarify the assumptions supporting the assessment.
* matrices have low resource requirements.

Problems relating to the use of matrices are the following :

* unless weight-scaled impact scores are used, the comparison of many project alternatives is difficult. Scaling the multitude of scores contained in a matrix is not a tractable proposition.
* the replicability of the method is undermined by a dependence on highly subjective judgements.

* matrices give inadequate consideration to secondary impacts.

(Warner and Preston, 1974,p8; Chase, 1976,p134 & 141; and Fuggle, 1979,p13.)

Most criticisms are extenuated if matrices are seen as a preliminary approach to EI analysis. Impact evaluation and comparison are not crucial issues. However, the identification of all major impact areas is important and the last point is of particular concern.

Although the term "interaction matrix" may imply a coverage of all interdependencies in a system, a two dimensional display of interactions between two parameter lists can only identify first order interactions. Welch and Lewis (1976,p201) have used a stylized three dimensional diagram to display some of the higher order interactions pertaining to environmental management.

2.2.4.2. Mathematical Matrices

A mathematical matrix is a rectangular array of quantities upon which algebraic operations can be legitimately performed. Mathematical matrices have not played an important role in EI analysis because of the difficulty in quantifying all types of impacts. There are, however, some matrix applications which could be used as supportive assessment techniques, and others which are worth noting for the way in which impact scores can be mathematically manipulated.

The Peterson Matrix

The analysis procedure described by Peterson et al (1974) relies directly on the multiplicative properties of matrices. One matrix is used to score the impacts of project actions on the physical environment (essentially the primary impacts). A second evaluates the effects that the impacted physical components may cause on the human environment (essentially the secondary impacts of the project). A team of assessors evaluate all impacts on an ordinal scale from -3 to +3. The matrices are multiplied to find the effect of the causal elements of the project on the human environment. The product matrix is operated on by a vector of human impact weights. The weighted human impacts are aggregated to produce a single overall score for the project. Figure 5 summarizes the process.

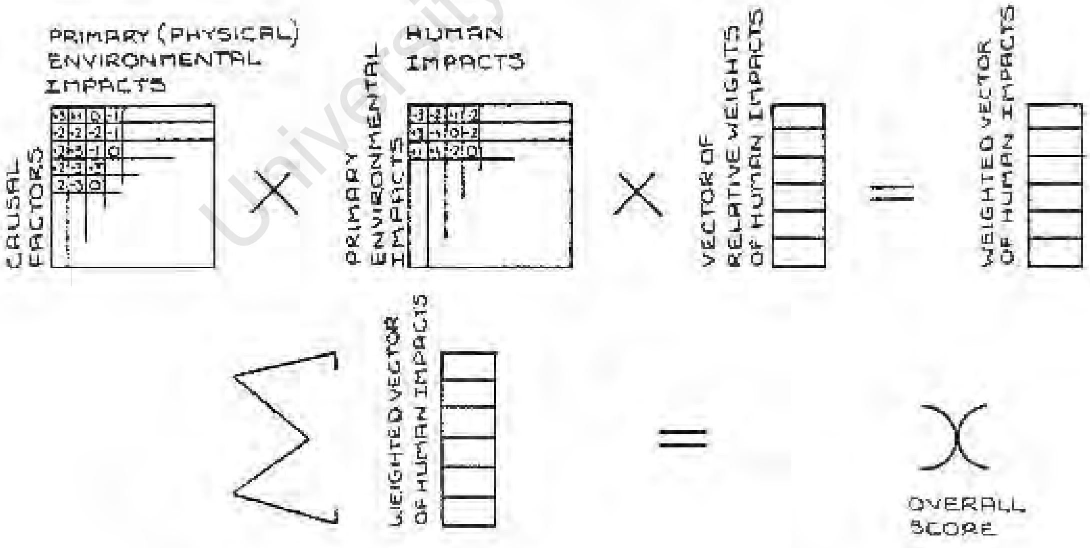


FIGURE 5. Diagrammatic representation of the Peterson matrix method (adapted from Skutsch and Flowerdew, 1976).

The subjective judgements of an assessment team are the only inputs to the method. A low dependence on resources allows for an iterative procedure of assessment and design modification during the formative stages of a project. Changes in design are stimulated by the early recognition of possible impacts, and revision continues until an acceptable project proposal is produced (Clark et al, 1980,p38).

The method achieves some success in considering the primary (physical) and secondary (human) impacts of a project. Coverage of secondary impacts, however, is not detailed or structured. Much information is lost in the aggregation process, which in itself, is invalidated by the multiplication and addition of ordinal scores (Skutsch and Flowerdew, 1976,p215).

The Component Interaction Matrix

The component interaction matrix (CIM) developed by Ross (1974) was first used in an environmental impact assessment of five alternative sites for the trans-shipment of lumber on Nanaimo estuary, British Columbia (Lands Directorate, 1974). The uniqueness of the area under consideration prompted an investigation of secondary impacts in an attempt to present the full implications of the project proposals.

In a CIM, the environment is represented by a list of environmental components arranged along both horizontal and vertical axes, and direct dependencies between the components are identified and marked as ones in the appropriate cells. Interdependencies up to the "n"th order (i.e. all higher order dependencies) can be determined by the use of a matrix powering procedure adapted from network analysis.

The Environment Canada CIM used twenty-one environmental components, and 120 first order dependencies were identified (see figure 6). Matrix manipulation (powering) was performed until

	CURRENTS	WIND	WATER TEMPERATURE	LIGHT	INTERTIDAL VEGETATION	UPLAND VEGETATION	BACTERIA	INSECTS	LARVAE	SHELLFISH	CRABS	OTHER CRUSTACEANS	PELAGIC FISH	BOTTOM FISH	WATER BIRDS	BIRDS OF PREY	SONG BIRDS	MARSH AND SHORE BIRDS	UPLAND AND GAME BIRDS	AQUATIC MAMMALS	UPLAND MAMMALS
CURRENTS		1																			
WIND						1															
WATER TEMPERATURE	1	1		1																	
LIGHT																					
INTERTIDAL VEGETATION	1		1	1																	
UPLAND VEGETATION				1			1	1									1	1	1		1
BACTERIA			1		1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1
INSECTS	1	1	1		1	1	1														
LARVAE	1		1		1		1			1	1	1	1	1							
SHELLFISH	1		1		1		1														
CRABS			1		1				1	1		1	1	1							
OTHER CRUSTACEANS			1		1		1	1	1	1	1		1	1							
PELAGIC FISH			1					1	1												
BOTTOM FISH			1		1			1	1	1	1	1	1								
WATER BIRDS					1			1	1	1	1	1	1	1							
BIRDS OF PREY						1		1			1	1	1	1	1		1	1	1		1
SONG BIRDS						1		1		1		1	1	1							
MARSH AND SHORE BIRDS						1		1		1	1	1	1	1							
UPLAND AND GAME BIRDS						1		1													
AQUATIC MAMMALS										1	1	1	1	1							
UPLAND MAMMALS						1				1	1	1					1		1		

FIGURE 6. The Nanaimo Port component interaction matrix of the Lands Directorate, Environment Canada.

fifth order dependencies had been discovered. From the information revealed by the powering process, a minimum link matrix was derived. All cells of the original CIM are used to contain integer values denoting the length (in terms of intervening nodes) of the shortest linkages connecting the two components (see figure 7). A disruption matrix was also formulated in which the impacts of each project alternative on all primary dependencies are scored on an ordinal scale from 0 to 3.

From figure 7 it can be seen that attention was restricted to the bio-physical environment. Although the components are a mix of

CURRENTS	WIND	WATER TEMPERATURE	LIGHT	INTERTIDAL VEGETATION	UPLAND VEGETATION	BACTERIA	INSECTS	LARVAE	SHELLFISH	CRABS	OTHER CRUSTACEANS	PELAGIC FISH	BOTTOM FISH	WATER BIRDS	BIRDS OF PREY	SONG BIRDS	MARSH AND SHORE BIRDS	UPLAND AND GAME BIRDS	AQUATIC MAMMALS	UPLAND MAMMALS	
4	1	4	3	4	2	3	3	4	4	4	4	4	4	4	4	3	3	3	4	3	CURRENTS
3	3	3	2	3	1	2	2	3	3	3	3	3	3	3	3	2	2	2	3	2	WIND
1	1	4	1	4	2	3	3	4	4	4	4	4	4	4	4	3	3	3	4	3	WATER TEMPERATURE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LIGHT
1	2	1	1	5	3	4	4	5	5	5	5	5	5	5	5	4	4	4	5	4	INTERTIDAL VEGETATION
2	2	2	1	2	2	1	1	2	2	2	2	2	2	2	2	1	1	1	2	1	UPLAND VEGETATION
2	2	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	BACTERIA
1	1	1	2	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	INSECTS
1	2	1	2	1	2	1	2	2	1	1	1	1	1	2	2	2	2	2	2	2	LARVAE
1	2	1	2	1	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	SHELLFISH
2	2	1	2	1	3	2	2	1	1	2	1	1	1	3	3	3	3	3	3	3	CRABS
2	2	1	2	1	2	1	1	1	1	1	2	1	1	2	2	2	2	2	2	2	OTHER CRUSTACEANS
2	2	1	2	2	2	2	1	1	2	2	2	2	2	3	3	3	3	3	3	3	PELAGIC FISH
2	2	1	2	1	2	2	1	1	1	1	1	1	2	3	3	3	3	3	3	3	BOTTOM FISH
2	2	2	2	1	2	2	1	1	1	1	1	1	1	3	3	3	3	3	3	3	WATER BIRDS
2	2	2	2	2	1	2	1	2	2	1	1	1	1	1	3	1	1	1	3	1	BIRDS OF PREY
2	2	2	2	2	1	2	1	2	1	2	1	1	1	3	3	2	2	2	3	2	SONG BIRDS
2	2	2	2	2	1	2	1	2	1	1	1	1	1	3	3	2	2	2	3	2	MARSH AND SHORE BIRDS
2	2	2	2	2	1	2	1	3	3	3	3	3	3	3	3	2	2	2	3	2	UPLAND AND GAME BIRDS
2	3	2	3	2	3	2	2	2	1	1	1	1	1	3	3	3	3	3	3	3	AQUATIC MAMMALS
2	3	2	2	2	1	2	2	2	1	1	1	2	2	3	3	1	2	1	3	2	UPLAND MAMMALS

FIGURE 7. The Nanaimo Port minimum link matrix of the Lands Directorate, Environment Canada.

specific and gross categories (e.g. wind (speed) and intertidal vegetation), the binary system for scoring dependencies does not provide an indication of the magnitude and importance of the interactions.

Provided that the initial identification of dependencies is explicit, the values (derived by mathematical procedures) in the minimum link matrix are substantive. The processes of matrix multiplication are not complicated, but they are tedious for large matrices, and would normally require the use of a computer. It is unfortunate that, while the minimum link matrix can indicate the existence and length of a linkage between any two components, the structure of these linkages is not exposed.

The results of the component interaction analysis were not readily incorporated into the overall assessment of the trans-shipment project. In fact the ad hoc assessment report of the five Nanaimo Port site proposals made little use of the results displayed in the component interaction, minimum link, and disruption matrices.

The CIM has been reviewed by Clark et al (1978a,p38; 1978b,p113; and 1980,p35) and Bisset (1980,p30), but has not received much positive comment. The limitations discussed above seriously undermine its applicability as an EI analysis method. It is, however, afforded attention as one of few approaches which is able to consider secondary impacts. The minimum link matrix is useful as a means of communicating the complex structure of the environmental systems likely to be affected by a project. As such the CIM could be viewed as a functional EI technique, but it would need to be extended before secondary impacts could be identified and evaluated.

The Nanaimo Port alternatives were re-analysed by Ross (1976) after he had extended the CIM approach. In the later method a team of judges weight each of the direct dependencies in the CIM, and these scores are normalized using psychometrical techniques of paired comparisons and non-metric multidimensional scaling. Interval values for the impacts on the dependencies are obtained in the same way (i.e. the disruption matrix is normalized). The dependency weights and impact scores are combined and aggregated over all first order dependencies to give an interval index of disruption for each project alternative.

The approach is, in effect, a weight-scaled matrix where the impacts on dependencies (rather than environmental components) are evaluated. Ross estimated that each judge was required to make well over 2000 paired comparisons, which were then submitted to computer analysis for multidimensional scaling. The ranking of alternative sites achieved by this method agreed with that of the original ad hoc study. Although the CIM matrix is the basis of the technique, the minimum link matrix is not consulted to give any consideration to secondary impacts.

Input-Output Matrices

An input-output study analyses the level of output of each sector of a given economic system in terms of its relationships to the productivity in all the other sectors (Leontief, 1970,p262). Central to the analysis is a form of component interaction matrix. The components are the producing and consuming sectors of the economic environment. Dependencies are quantified in terms of the per unit cash flow of goods between sectors. The component dependencies can be reduced to an input-output matrix of coefficients which reflect the inter-relationships between all sectors. The input-output matrix is generally used to investigate the effects throughout the economic system of a change in productivity in any one or more individual sectors. The analysis involves the mathematical procedures of matrix multiplication and inversion.

Traditionally, economic and environmental assessment studies have been performed separately. However, the hyphenations between the socio-economic and bio-physical environments, and the fact that the economic environment is part of the broader human environment, have stimulated a rationale for including economic analyses in EI assessments (Jain et al, 1977,p128).

Much of the environmental impact caused by economic activity is related to the use of valuable natural resources which are not included in the final price of the goods produced (Hjalte et al, 1977,p7). For instance, the direct consumption of clean water and the disposal of particulate waste-products into the air by manufacturing sectors of an economy deplete the natural resources of clean air and water. The costs of these commodities are external to the economy, and tend to be paid for in terms of reduced environmental quality (i.e. increased levels of air and water pollution).

Leontief (1970,p262) has investigated the polluting potential of economic activities by relating sectorial productivity to the production of waste products. The direct and indirect environmental impacts of economic activities have been investigated in greater detail by Hite and Laurent (1971,p1070). They propose an analytical approach in which an environmental linkages matrix is formulated to quantify the inflows to, and the outflows from, the economy to the bio-physical environment. The product of this, and the input-output matrix, shows the effect on the non-economic environment of one unit final output from each sector (see figure 8). Investigation of the product matrix can

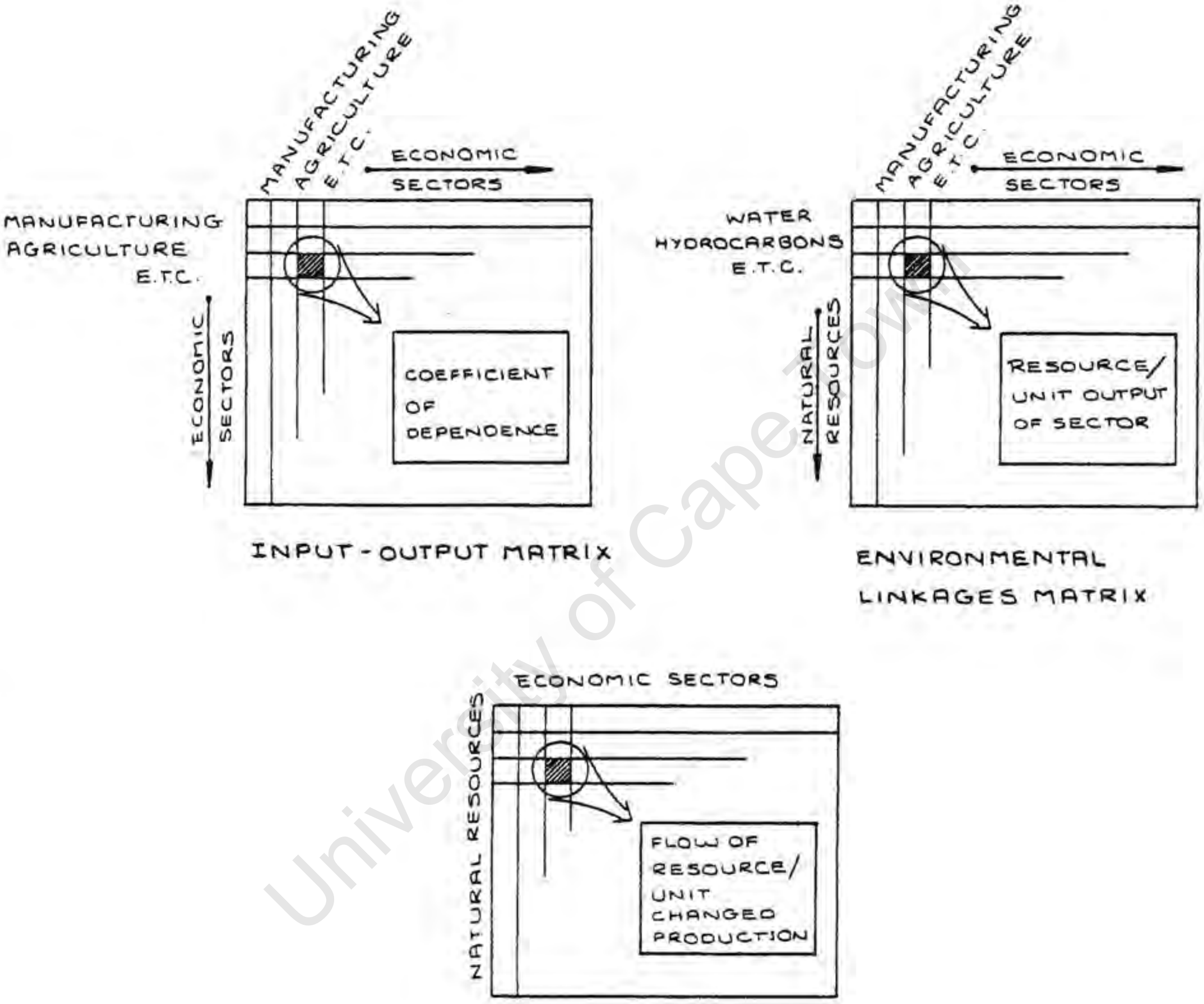


FIGURE 8. Diagrammatic representation of the input-output approach of Hite and Laurent.

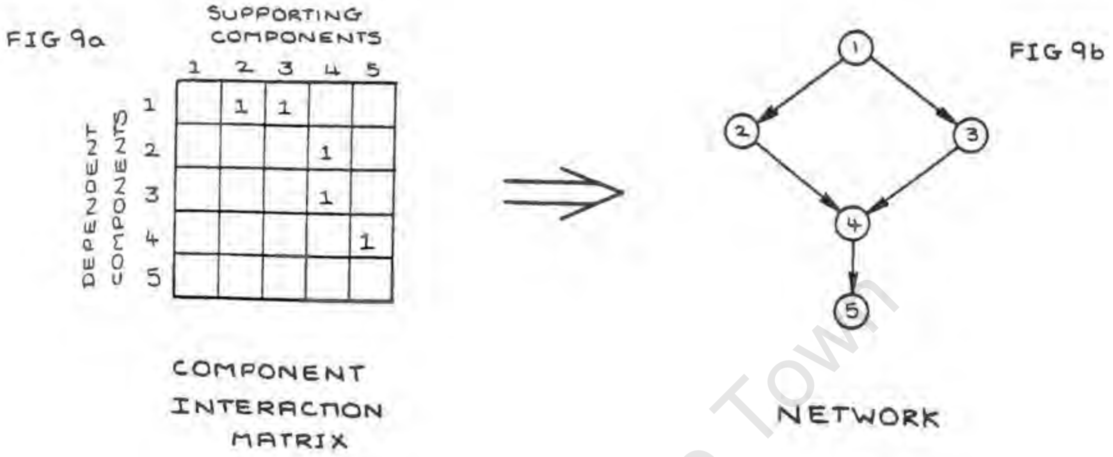
illustrate how an economic sector which is not directly linked in inputs or outputs to the bio-physical environment (e.g. the health service sector) can be responsible for a substantial amount of environmental impact because of its indirect dependence on other sectors (e.g. the manufacturing and food sectors).

Isard et al (1972,p94) and Lee and Fenwick (1973,p25) have also incorporated elements of input-output analysis into approaches addressing hyphenations between the economic and bio-physical environments.

Input-output analyses are not much used in EI assessment (Fuggle, 1979,p18). Very high resource needs are involved in constructing an input-output matrix (in terms of data and analytical effort). Results displayed in a conventional or extended matrix still need to be interpreted before resource flows can be presented as environmental impacts. It is notable, though, that the cash and material flows in the economic and ambient environments can support a quantified investigation of higher order dependencies.

2.2.5. Networks

It has already been stated that presentational matrices can only clearly show the primary interactions within any particular framework. It is possible to investigate higher order linkages in two dimensions by using directional diagrams called networks. In the component interaction matrix in figure 9a, only direct dependencies are marked. However, the network representation of the same system in Figure 9b traces linkages to their full extent. As in presentational matrices, there are two distinct types of networks; those which trace the progression of causes and effects of various project actions, and those which trace the higher order dependencies among the components of a defined system. "Networks" (Warner and Preston, 1974,p4), "stepped matrices" (Canter, 1977,p196) and "flow diagrams" (Munn, 1975,p43) are terms used for the first type of network, while "system diagrams" (Bisset, 1980,p32) describes the second.



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

FIGURE 9. The relationship between component interaction matrices (adjacency matrices) and networks.

The Sorensen Network

The network technique developed by Sorensen (1971) is probably the best known approach for investigating higher order impacts (Clark et al, 1980,p23). Figure 10 shows a portion of a network devised by Sorensen to display the possible consequences of various forms of land use, for a section of Californian coastline. Three options for residential development are related to four primary impacts, and cause-effect linkages are developed for each identified primary impact. The diagram also takes cognisance of feasible mitigatory measures. The objective of the network

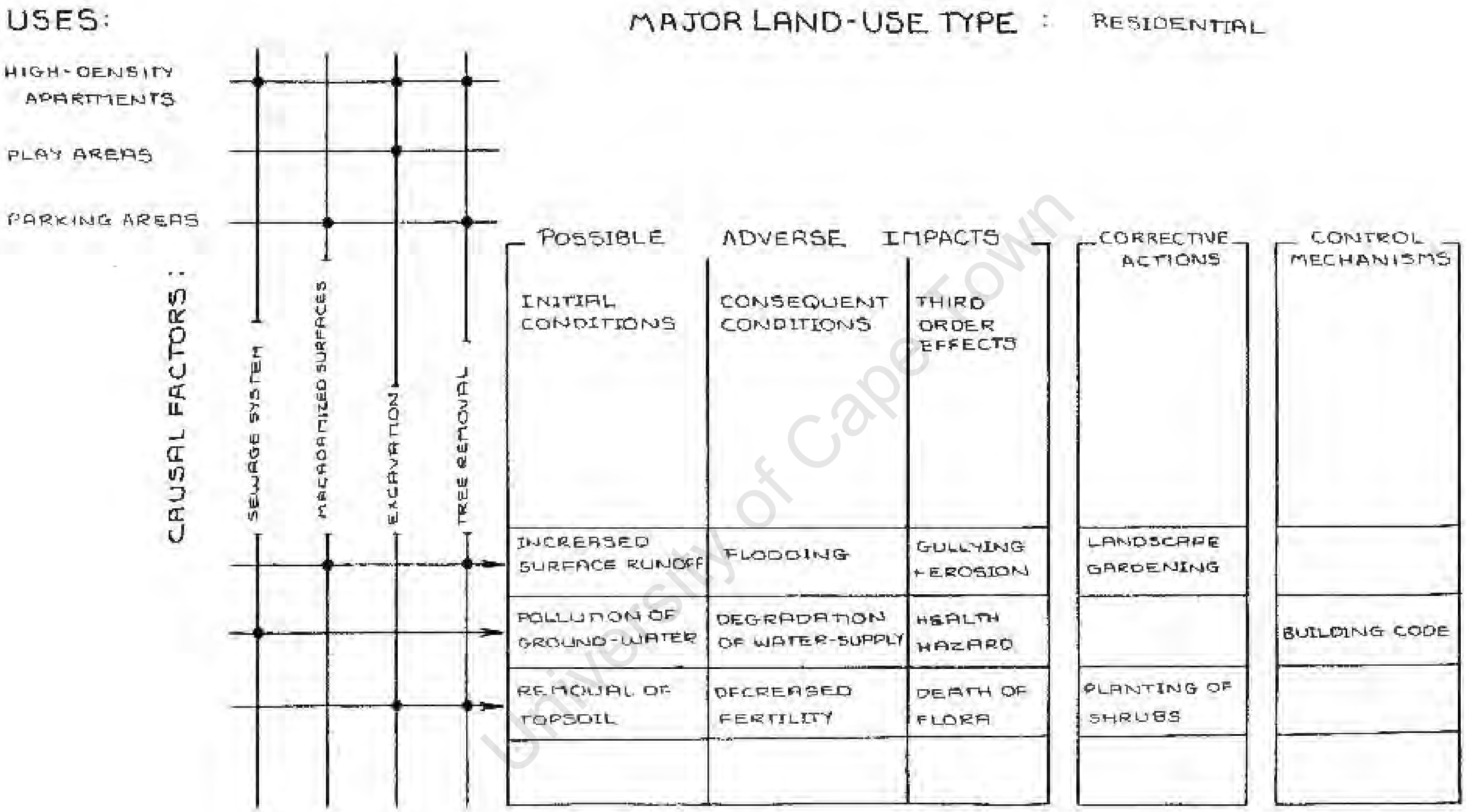


FIGURE 10. Illustrative diagramme of a Sorensen network.

approach is to display, in an easily understood format, the intermediary links between a project and its ultimate impacts. The network is correctly described as "a framework" in the title of Sorensen's paper (1971). The form and content of the diagram has to be predetermined for a particular EI analysis. Application is, therefore, limited to environments and development alternatives for which adequate data is available, and for which reference networks exist.

The reference networks and supportive data for the Californian study and a later study by Sorensen and Moss (of 1973, cited by Munn, 1975,p43), were stored and manipulated by a computer. However, an extended data base needs an expanded display, which can in turn, make the network unwieldy and complex (Munn, 1975,p43; Clark et al, 1980,p33). Complexity increases as higher order impacts are considered, and as a result, the Sorensen network is restricted to third and lower order impacts. Network impacts are not scored in any quantitative way, and therefore the comparison of project alternatives is not readily achieved (Munn, 1975,p43).

The CIM and network approaches to secondary impacts have been compared by Clark et al (1978a,p41), and neither technique was considered to be superior. Although networks display more information than CIMs, the latter are concise and more manageable.

System Diagrams

The technique developed by Gilliland and Risser (1977) is based on the energy network diagrams pioneered by Odum (1971,p37). A free format network portrays the important components of a system, and

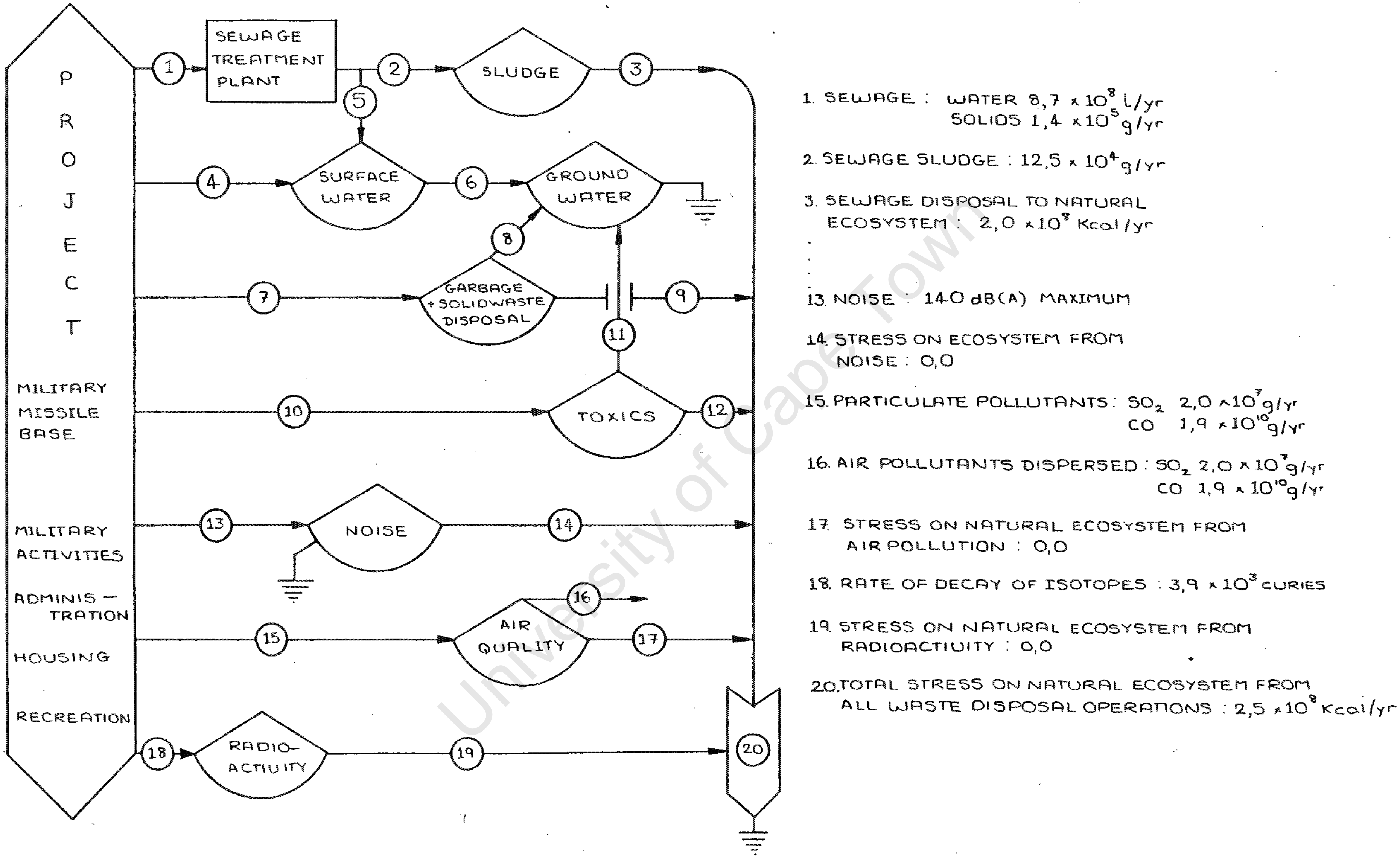


FIGURE 11. A portion of the system diagramme of Gilliland and Risser (1977).

directional dependencies are quantified as energy transfers in appropriate units (e.g. kilocalories, dB(A), or currie). A section of the system diagram used to analyse the impacts of a missile range is shown in figure 11. Primary impacts are first identified by some conventional EI analysis method, and then a measure of the total higher order impact of an action is extracted by an interpretation of the degree of impact on the energy pathways.

Primarily, the technique is intended as an aid to organizing and communicating the investigations of an EI analysis (Gilliland and Risser, 1977,p206). It has the same disadvantages as the networks discussed above (Bisset, 1980,p33). In addition, the quantification of dependencies restricts application to ecological systems, and makes the approach heavily reliant upon data resources (Clark et al, 1978b,p114). However, the diagrams summarize much baseline data, and can be developed into more detailed models if an EI analysis warrants further research (Barile, 1978,p203).

2.2.6. Overlays

The concept of overlays was developed by McHarg (1971; originally 1968) to incorporate broad environmental considerations into the selection of highway routes. The approach has been reviewed as a valuable EI analysis method for projects of regional or linear proportions (Warner and Preston, 1974,p16).

The McHarg overlay is based on a set of transparent maps, each of which represents the spatial variation of an environmental parameter (e.g. susceptibility to erosion, or recreational value).

The maps are shaded to show three degrees of parameter compatibility with the proposed project. A composite picture of the overall social cost of impacting any particular area is approximated by superimposing all the transparent maps. Any number of project alternatives can be located on the final map to investigate the degree of associated impacts. The validity of the analysis is related to the type and number of parameters chosen. For a readable composite map, the number of parameters in a transparency overlay is limited to about ten (Munn, 1975,p58). Clark et al (1980, p116) report that the efficiency of overlays has been extended by computerized techniques for mapping and combining many characteristics. For a computerized analysis, parameters are quantitatively scored onto a reference grid. In the aggregation process, parameter scores can be weighted for significance to give a weight-scaled representation of impact potential in the final diagram.

Although computerized overlays are demanding on data and analytical resources, they have a general applicability to environmental management and can form part of a broad data base for preliminary EI analyses (Warner and Preston, 1974,p14). Parameter maps present data in a summarized and easily interpreted form (Fuggle, 1979,p16), but are unable to reflect the possibility of secondary impacts (Cook, 1977,p18).

2.2.7. Modelling Procedures

Models in EI analysis are simplified representations of the real, complex systems which may be affected by a project (Munn, 1975, p68). Some of the above methods and techniques are in fact models. The component interaction matrix, for example, gives a simple, structural representation of the components and dependencies within a specified environment. However, functional processes, rather than structural components, are responsible for defining relationships within a system. Therefore the explicit identification and evaluation of impacts, and secondary impacts in particular, requires a study of the dynamic mechanisms that control the internal state of a system, (Bisset, 1980,p38). For this reason, although it is possible to construct models at many levels (e.g. conceptual, descriptive, physical or mathematical), dynamic models based on mathematical representation are best suited for extending the scope of an EI study (Reichle, 1975, p263). Mathematical relationships can be formulated to give the time-base necessary for a meaningful study of processes.

Modelling is a resource intensive procedure, and EI analyses have been more likely to draw on existing models than to create new ones (Clark et al, 1978b,p119). Reichle (1975,p263) has suggested that ecosystem models are an effective aid to the assessment of secondary impacts. Predictive models for estimating the higher order impacts of major industrial and urbanization projects have been developed by Guldberg et al (1977) and members of the International Biological Program (Jameson, 1976,p2/1). However, Kane et al (1973,p65) and Holling (1978,p69) have recognized that the output of such models is not always directly applicable to particular EI analyses. They place an emphasis on computerized procedures for interactive simulation modelling within the framework of an EI analysis. The procedure of Kane et al (1973) is aimed at giving non-experts in mathematical modelling the ability to model the processes which are specific to one or more areas of impact. The interactive aid for computer modelling created by Furniss (1977) is a similar approach, but it gives greater control over the form and output of the model.

The latter type of modelling procedures are tractable techniques for extending the depth of study of an EI analysis. Further, the modelling of system processes is possibly the most efficient means of investigating the origins of secondary impacts, and almost certainly the only way of quantifying them.

2.2.8. Evaluation Techniques

EI assessment originates as a decision, reflecting the social values of a community, to consider the environmental cost of development projects. Economic and environmental costs are both implicitly related to the value preferences of society, yet they differ in their ability to be quantified. The cost of economic goods is related to the utility attached to them by society and can be quantified in monetary terms. However, environmental quality is less easily quantified, as there are no common units for the utility of environmental fitness (Hjalte et al, 1977,p9). Consequently, the study of environmental impacts is partially dependent upon subjective judgements (Hart and Cullen, 1976, p233). The extent to which EI analysis is dependent on subjective judgements has been discussed in detail by Matthews (1975). Decisions as to which environmental components and project actions are to be considered, and as to what constitutes an impact, cannot be made without reference to some framework of social values. Once the bounds of an EI analysis have been defined, the interpretation of data is normally an objective process. The quantified evaluation of impacts is, however, an entirely subjective activity (Matthews, 1975,p126).

Various techniques have been developed to ensure that the identification and evaluation of impacts is based upon acceptable value judgements. The validity of the subjective content of a study can be extended by employing a team of multidisciplinary experts to perform the assessment of impacts (Hart and Cullen, 1976,p231), and also by incorporating public opinion into EI analyses (Runyan, 1977, p125). Both tactics require some means of structuring the interaction among the group of assessors to arrive at a broadly representative conclusion on subjective issues. Runyan (1977) has identified four suitable techniques, of which the delphi (developed by Helmer in 1963) is the most versatile and widely applied example (Coates, 1976,p112). The delphi technique is an integral part of the weight-scaled checklist method of Dee et al (1973) where it is used to derive commensurate impact scores to compare project alternatives. Although alternatives are ranked in order of total impact scores, the intermediary judgements supporting the final order are available for inspection (i.e. the impact weights and scaling functions are explicit).

The meaning and precision of subjective judgements can be increased by the processes of scaling and normalization. Golden et al (1979,p28) have listed the mathematical procedures of pairwise ranking, multidimensional analysis, multivariate data analysis, and others, which can be used to extract useful information from unstructured or subjective assessment scores. The matrix based procedure, proposed by Kiely-Brocato et al

(1980), for scaling attitude scores about environmental management options is particularly suited for gauging and structuring public opinion for EI analyses.

Although there is a wide selection of techniques for processing the subjective facets of EI analysis, the major concerns that remain are the feasibility of using the sometimes complicated and computer dependent procedures (e.g. multidimensional scaling) and the necessity for keeping the supportive value judgements available for scrutiny (Coates, 1976; Matthews, 1975,p130).

2.2.9. Adaptive Methods

Up to this point no consideration has been given to the compatibility between the provisions for EI assessment and the form of EI analysis methods. EI assessment is not an economically profitable activity over the short term. Therefore the social preference for a consideration of environmental cost is normally reinforced by legal provisions. The type of analysis methods which will be acceptable to a particular situation is related to the level of commitment embodied in the relevant legal provisions (Hollick, 1981a,p69).

In the U.S.A., where many EI analysis methods originate, the conditions of NEPA guarantee that federal impact statements are incorporated into the decision-making process (Council on Environmental Quality, 1978,p400). There is also a certain amount of autonomy afforded to the assessors of environmental impact, and consequently there is a predisposition towards evaluative methods (Bisset, 1978,p46). The provisions of NEPA and the bureaucratic structure of Federal agencies have made EI analysis a separate process from the planning and development of a project (Jain, 1977,p19).

In contrast, Catlow and Thirlwall (1976,p67) have concluded that descriptive rather than evaluative methods would be suitable for the flexible administrative and planning structures in the United Kingdom. They are of the opinion that it would be in the interests of efficient environmental management to have EI analysis and project design as interacting activities. This would avoid the high costs and delays which have been part of EI assessment in the U.S.A. (Catlow and Thirlwall, 1976,p7).

Clark et al (1976) have devised an adaptive and comprehensive approach to impact assessment for the United Kingdom (the Project Appraisal For Development Control (PADC) system) based on the recommendations of Catlow and Thirlwall (1976). The PADC manual provides a checklist of activities for conducting an EI analysis compatible with the existing planning structure. Central to the approach is an interaction matrix and ten technical advice notes or checklists for predicting impacts. Firm guidelines are provided on the communication of impact information.

Holling (1978) has formalized the opinion that a selection of methods and techniques can be drawn together in a structured framework to meet the constraints of, and conditions for, EI assessment in differing circumstances. The central thesis maintained by Holling is that EI assessment should be part of plan and policy formulation, rather than of post-design project appraisal. Nevertheless, an approach is described where matrices and other traditional methods are accommodated as preliminary analysis procedures, while simulation modelling is recommended as the most adaptable impact prediction tool. Interaction between the researchers and decision- or policy-makers is facilitated by workshops throughout the formative stages of a development proposal.

Adaptive methods have also been developed in an effort to extend the capabilities of EI analysis. A method proposed by Sondheim (1978) has the ability to consider many project alternatives using interval scaled impact scores. The co-ordinators of an analysis define lists of project alternatives and environmental aspects (components). Two assessment panels are chosen : the rating panel is free to use any method (e.g. matrices, networks, models, etc.) to evaluate the impact of each project alternative on each environmental aspect; and the weighting panel must weight the aspects in order of importance. The normalized and scaled rating and weighting scores are arranged in matrices which are multiplied and aggregated to obtain a preferential ordering of the projects (the procedure has similarities to the methods of Peterson et al (1974) and Ross (1976)).

Although the method stresses an evaluation of impacts, the subjective processes are separable from the essentially objective activity of investigating impacts. Further, the subjective base can be extended by incorporating public representatives in the weighting panel. Sondheim (1978,p41) has recognized that the procedure, although comprehensive in its ability to compare many alternatives, is entirely dependent upon supportive methods for a consideration of synergistic or higher order effects.

Jain et al (1977,p93) and Jain and Webster (1977) have described a computer-aided approach to EI analysis which has been developed for the U.S. Army Corps of Engineers by the Construction Engineering Research Laboratory (Urban et al, 1975). Thirty person years of research into the environmental problems related to nine army activities (e.g. construction, training, and procurement) have been structured into an extensive checklist which can be selectively accessed through a computer-based retrieval system (Jain and Webster, 1977,p268). The user provides information regarding the type of project, the environmental setting of the project site, and the level of analytical detail required. Output from the system is partially in the form of interaction matrices which summarize the range of possible primary impacts. Further information is retrievable on secondary impacts, mitigatory measures, and pertinent legal provisions. The form of the output is structured to meet the procedural and analytical requirements of NEPA (Urban et al, 1975).

The method is atypical in that it provides access to an already compiled data base, while most other methods are concerned with the collection of such data. A reasonable coverage of secondary impacts is possible because of the specificity of the area of concern (i.e. nine army activities). The method consolidates existing data and expertise to provide a comprehensive, economically efficient, and easily used approach to EI analysis. Although the approach is sophisticated and resource dependent, it is appropriate where there is a formal and defined commitment to EI assessment.

2.3. Summary and Discussion

In South Africa, as there is no formalized commitment to consider the final evaluations of an impact analysis, a need exists for descriptive rather than evaluative methods. There is also the necessity for procedures which can identify important environmental concerns during the formative stages of a project design. These conditions should be met if EI analysis is to be accepted as a workable extention to the conventional methods of project appraisal.

Having investigated four adaptive approaches to EI analysis, it seems feasible that various existing methods and techniques could be combined to meet the prerequisites stated above. Further, if methods are combined in a sequential manner, an analysis can progress from a simple investigation of major impacts to a more detailed study of the areas of concern.

There are methods (e.g. matrices, overlays, and simple checklists) which can be used in the preliminary stages of a project to guide development in a direction of minimized environmental impact. This is achieved by communicating the more obvious environmental implications of a project. Once developed, more detailed methods (e.g. system diagrams, weight-scaled checklists, and dynamic models) can be used to assess the more complex implications of the project proposal, and to evaluate possible mitigatory measures, and environmental management strategies.

Advanced and preliminary approaches to EI analysis should not be seen as separable activities. It should be the task of preliminary methods to focus attention on important aspects and to create a data base from which advanced methods can develop. Output from both stages can be incorporated in an EI statement for the final evaluation of a project.

Preliminary methods are subject to resource, time, and simplicity constraints, which makes it difficult to perform even a broad analysis of impacts. Nevertheless, there is a wide range of approaches for investigating primary impacts at different levels of detail. However, of the four main approaches which are able to consider secondary impacts - the minimum link matrix; networks; system diagrams; and dynamic models - none is able to provide an explicit procedure for identifying the range of secondary impacts which may be associated with a project. The first (minimum link matrix) gives a very broad indication as to the possibility of secondary impacts, while the latter three are only useful once areas of secondary impact have been identified. Although the utility of all these approaches is recognized, it would appear that there is no method or technique available for investigating secondary impacts at a level appropriate to preliminary EI analysis.

In chapter three a description is given of how the minimum link matrix can be extended to serve as a preliminary technique for investigating secondary impacts.

3. CHAPTER 3 - A TECHNIQUE FOR ADDRESSING SECONDARY IMPACTS

3.1. Introduction

In chapter one, the origin of secondary impacts was associated with the inter-relationships between the components of a system. Such direct dependencies can make components dependent on others with which they are not directly linked. Consequently, actions affecting any component can have repercussions on those which are indirectly dependent on it.

For an exhaustive study of secondary impacts, a full understanding of the structural features and functional processes which define dependencies within an environmental system would be desirable. This level of detail may be possible with advanced analytical procedures. For an initial investigation of secondary impacts, it is sufficient to know about all the possible direct and higher order dependencies in a system.

A satisfactory technique for a preliminary investigation of secondary impacts should embody the following attributes :

* The technique should be simple, descriptive, easily applied to a diverse range of impact studies, and adaptable to a wide selection of EI analysis methods.

* It should be able to accommodate a broad definition of the environment; one which can include both bio-physical and socio-economic components.

* There should be a minimal reliance on subjective judgements. Where subjective judgements are employed, they should be

adequately substantiated.

* Besides exposing the configuration of all higher order dependencies, the technique should identify components which are most likely to originate, or which are most likely to be affected by, secondary impacts.

The approach developed in this chapter is based on the component interaction and minimum link matrices (Lands Directorate, 1974) which are described in the "Mathematical Matrices" section of chapter 2. The elements of a component interaction matrix record (normally on a binary scale) the occurrence of direct dependencies between similar but perpendicular lists of environmental components. A minimum link matrix is of the same form, but the elements record the length of the linkages (in terms of intervening components) between all components. From a simple representation of direct dependencies in a component interaction matrix (CIM), a minimum link matrix (MLM) can be derived. This displays the extent to which each component is dependent on others in a system. The mathematical operations for deriving the MLM process information about higher order dependencies which is lost to the final matrix representation. However, the information can be extracted to extend the capabilities of the technique. The choice of components, and the recognition of direct dependencies represent the full extent of the subjective judgements associated with the MLM. The opinions of a multidisciplinary team could be structured by a delphi or similar technique to attempt a comprehensive listing of components and component dependencies.

For these reasons, it is felt that the CIM is superior to other methods and techniques as a basis for the preliminary study of secondary impacts.

As the component interaction and minimum link matrices are both constituent parts of the proposed approach, confusion will be avoided by referring to the procedure as the component interaction technique (CI technique). In this chapter the theory supporting the CI technique, and its implementation as a computer programme are described and discussed.

3.2. A Brief Introduction to Graph Theory.

Berge (1964) has described how graph theory can be used to study the characteristics of any situation which can be represented as points joined by lines or arrows. Once a system is represented as a graph, certain concepts and theorems of graph theory become available for analysing its connectivity. As the CI technique is based on aspects of graph theory, some relevant terms are defined below (attributed to Wilson and Beineke, 1979,p1 and Nystuen and Dacey, 1961,p33).

Graphs are defined as sets of systematically organized points (termed "vertices") and lines (termed "edges"). A graph is "non-planar" if, when represented in two dimensions, an intersection of two edges does not necessarily define a vertex. A vertex is "adjacent" to another if there is an edge connecting them. If all edges are of equal value for all adjacencies, the graph is a "binary graph". Edges may be assigned some value or weight in a "weighted graph". If the edges of a graph are orientated it is a "directed graph", and the orientation of edges is displayed by an arrow-head on the lines. If a vertex, A, is not adjacent to a vertex, B, but there exists some sequence of vertices and edges which connect A and B, then A and B are "linked" and the sequence of vertices and edges is termed a "path" or "linkage". A directed path must have all edges appropriately orientated. The "valency" of a vertex is the number of edges incident to it. If a graph is directed, then each vertex has an "out-valency" and an "in-valency" depending on the orientation of the edges incident to it.

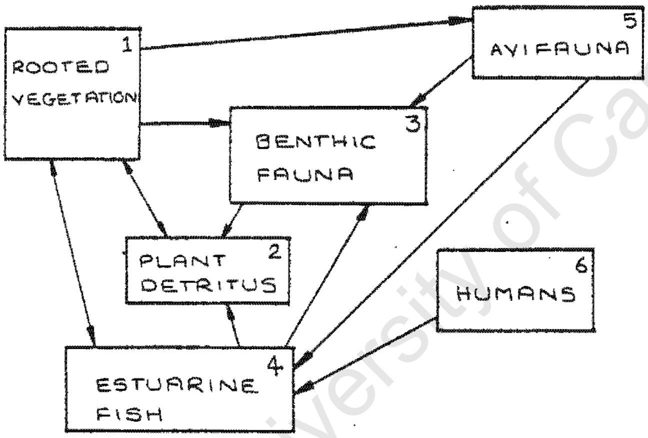
For the purposes of mathematical manipulation, every graph can be associated with an adjacency matrix which completely describes the graph. Every point in the graph is represented by a row and a column in the matrix. Lines of the graph are represented by elements of the matrix. Using standard matrix notation, an adjacency between point i and point j in a binary graph will have element A(i,j) of the adjacency matrix A, set to one. If these points are not adjacent, then element A(i,j) is set to zero. If the line between i and j has some weighting, A(i,j) is assigned an appropriate value.

Any system described in terms of components and dependencies can be reduced to a point-line abstraction which conforms to the definition of a graph. In such a case, the vertices of the graph represent components, and the orientated edges represent directional dependencies. Figure 12 illustrates the relationship between a simple descriptive model of an estuarine system and its associated (directed, binary, non-planar) graph. The environmental model has been compiled as an illustrative aid, rather than an accurate representation of an estuarine system. A more accurate estuarine CIM is discussed in chapter four. The adjacency matrix is also shown and can be recognized as the component interaction matrix of the system. Note that a dependency of component i on component j is given as i -> j in the graph; and as A(i,j) = 1 in the adjacency matrix A. The row (i) components are dependent on the column (j) components. Further,

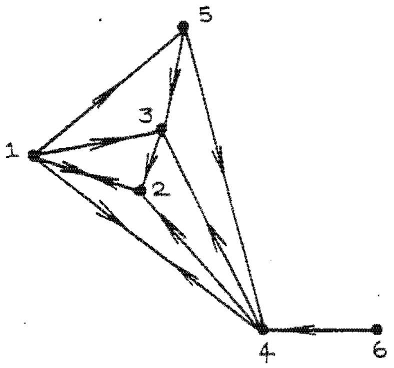
DESCRIPTIVE SYSTEM'S MODEL

COMPONENTS	DEPENDENT COMPONENTS	TYPE OF DEPENDENCY
ROOTED VEGETATION	PLANT DETRITUS	SOURCE OF NUTRIENTS
	BENTHIC FAUNA	NUTRIENTS
	ESTUARINE FISH	NUTRIENTS
	AVIFAUNA	NUTRIENTS
PLANT DETRITUS	ROOTED VEGETATION	SOURCE
BENTHIC FAUNA	PLANT DETRITUS	FOOD
ESTUARINE FISH	ROOTED VEGETATION	FOOD / SHELTER
	PLANT DETRITUS	FOOD
	BENTHIC FAUNA	FOOD
AVIFAUNA [MARINE +UPLAND BIRDS]	BENTHIC FAUNA	FOOD
	ESTUARINE FISH	FOOD
HUMANS	ESTUARINE FISH	FOOD/RECREATION

NETWORK



GRAPH



ADJACENCY OR COMPONENT INTERACTION MATRIX

SUPPORTING COMPONENTS

DEPENDENT COMPONENTS	1	2	3	4	5	6	OUT - VALENCY	ADJACENCY RATIO [IN/OUT]
ROOTED VEGETATION	0	1	1	1	1	0	4	0,5
PLANT DETRITUS	1	0	0	0	0	0	1	3,0
BENTHIC FAUNA	0	1	0	0	0	0	1	3,0
ESTUARINE FISH	1	1	1	0	0	0	3	1,0
AVIFAUNA	0	0	1	1	0	0	2	0,5
HUMANS	0	0	0	1	0	0	1	0
IN - VALENCY	2	3	3	3	1	0		

GAMMA INDEX = 12/(6x5)

= 0,4

FIGURE 12. The descriptive model, network, graph, and adjacency matrix of an illustrative estuarine system.

the dependency i -> j is an out-dependency for i and an in-dependency for j.

3.3. Information from the Component Interaction Matrix

3.3.1. Component Valency

If the elements of a CIM are summed across each row, the out-valency of each component is derived. Similarly, summing down the columns gives the in-valency. These measures give some immediate indication as to the status of each component in the system. In figure 12, the in-valency of "benthic fauna" is given as 3, and the out-valency as 1. This indicates that benthic fauna is only directly dependent on one other component, while three components are dependent upon it. This information is summarized in the valency ratio (in/out valency).

3.3.2. Source and Sink Components

A source vertex has an out-valency of zero, and a sink vertex has an in-valency of zero (Wilson and Beineke, 1979,p4). "Humans", in figure 12, are a sink component and can have no effect on the rest of the system. A source component has a potentially large influence on the components to which it is linked.

3.3.3. Measures of Connectivity

Kansky (1963,p14) has described a number of graph-theoretic measures which reflect the degree of connectivity of a network in various ways. The measures are based on the values of "e" (the number of edges to the graph), and "v" (the number of vertices). For example, the gamma index is given as e/(v(v-1)) for directed non-planar graphs. The index is bound by 0 on the lower limit (indicating a graph of vertices with no connecting edges), and 1 on the upper limit (for graphs where every vertex is adjacent to all others).

While graph indices, derived from the adjacency matrix information, can be useful for comparing the structure of different systems, they do not explicitly reveal information on system configuration. Therefore, they have a limited usefulness in EI analysis; possibly as indicators of the relative susceptibility of different systems to secondary impacts. However, as will be shown, the MLM itself offers a more effective means of investigating the internal structure of a system.

3.4. Graph Theory and the Minimum Link Matrix

Shimbel (1951,p171) has given a detailed exposition of a theorem in graph theory which supports the following statements :

If C = A x A where A is an adjacency matrix and C is the product by matrix multiplication, then C(i,j) is the number of paths between component i and j of length 2 (i.e. there are two dependencies involved in the linkage). Stated in general terms, if P[n] = A(powered to n), where A is an adjacency matrix, then P[n](i,j) is the number of paths consisting of n dependencies between components i and j.

The results of the powering procedure have been used to derive the shortest path or minimum link matrix (Haggett et al, 1977,p319). The following algorithm applies :

- Initialize two binary matrices; A, the adjacency matrix, and L[n], the minimum link matrix, each to describe the graph in question. The value of "n" denotes the level of powering of the matrix A. At this stage n = 1, and L[1] is a matrix of all paths of length one.

- By conventional matrix multiplication compute P[2] = A x A (n = 2). Every element of P[2] is inspected in conjunction with the corresponding element in L[1]. A non-zero value for the element P[2](i,j) indicates one or more two-step paths between components i and j. If L[1](i,j) = 0 then there exists a two-step path which is the minimum link path between i and j, (because there is no shorter path). Element L[1](i,j) is set to 2. Once every element of L[1] has been appropriately adjusted, the matrix shows all the dependency chains of length 1 and 2, and is redefined as L[2].

- The procedure is repeated for P[3] = P[2] x A (n = 3), and if P[3](i,j) is non-zero and L[2](i,j) = 0 then L[2](i,j) is set to 3. On completion, matrix L[2] becomes L[3]. In general terms the procedure is :

```
P[n] = P[n-1] x A
for all i and j
  if P[n](i,j) > 0  and  L[n-1](i,j) = 0
    then L[n-1](i,j) = n
after all i and j have been considered
  L[n] = L[n-1]
```

- The procedure is iterated until one of two conditions are met :

1) There are no more non-zero elements in L[n], which indicates that there is a minimum path between all components in the system. The system graph is then said to be "strongly connected" (Wilson and Beineke, 1979,p8).
2) There is no change in L[n] after powering the adjacency matrix to the level [n+1], although there are still some elements of L[n] at zero. This indicates that there are components in the system which are not directly or indirectly connected to all the others. If the graph is not in fact "disconnected" (i.e. where some components have no links at all to the system), then the graph is "unilaterally connected" (Wilson and Beineke, 1979,p8).

In both these cases the final form of L[n] is the minimum link matrix and will be referred to as L[N]. "N" is taken as the final value of n.

3.5. Information from the Minimum Link Matrix

3.5.1. Graph Diameter

If a system is strongly connected, then the value of N, that is the maximum value of an element in L[N], is called the diameter (Berge, 1964,p126), or solution time (Haggett and Chorley, 1969,p41) of the graph. The diameter of a strongly connected system is information that every component is dependent on all others by paths with N or fewer links. For a system which is unilaterally connected the value of N gives the maximum number of links in the minimum paths that do exist. Note that a system will have a diameter if there are no sink or source components (i.e. every component has an in-valency and an out-valency greater than zero).

3.5.2. Graph Distance

There are various measures which can be derived from the MLM to give an indication of graph structure. The dispersion measure (Shimbel, 1953,p501) is the sum of all the elements of L[N], and offers some indication as to the "compactness" of a graph as a whole. It follows that if all components are closely linked to all others, then the minimum paths will be short, and therefore the sum of all the minimum path lengths (the elements of L[N]) will be small. The measure of dispersion may have application to a graph of components and dependencies if it is accepted that the influence of any one (non-sink) component is likely to be greater in a compact, rather than a dispersed system.

The accessibility measure (also attributed to Shimbel, 1953), has two forms for directed graphs. The out-accessibility measure is the sum of the minimum path lengths across a row of a MLM, and indicates how closely the row component is linked to the system as a whole. Summing elements down a column gives the in-accessibility of a component, which is a measure of how closely linked the whole system is to that component. Carter (1969,p53) has used the measures in this sense to study the accessibility of urban places, but they could also provide information on the structural relationship between a given component, and the environmental system of which it is part.

Both dispersion and accessibility measures are non-metric. A value for one graph may be equal to that of another, without implying any similarity of structure or connectivity. Specifically, the values of the measures are dependent on the number of vertices in the graph. A further problem with the measures is that a value of zero for the element L[N](i,j) implies the shortest possible path between components i and j, when in fact there is no connection at all. It would be more appropriate if the elements were set at infinity. The incorrect "sense" of zero values in a MLM contributes toward values which can give a spurious indication of a well connected system. Therefore, the measures are only useful for strongly connected graphs which, by definition, have no zero values in L[N].

Reed (1970) has overcome the inconsistencies arising from the presence of zeroes by transforming the accessibility measures to measures of "average graph distance". The revised measure is calculated by averaging the accessibility measure over the actual number of minimum paths which connect a component and the system (i.e. the accessibility measure is divided by the number of non-zero elements in the summation).

3.5.3. Disruptive Measures

The average graph distance has been used (Reed, 1970) to rank the nodes of a communication network in order of their disruptive potential to the whole network. The procedure considers both the direct and indirect connections by which nodes are linked in a system, and could therefore, be applicable to a study of the secondary impact potential of components in an environmental system.

Any component in a system has the potential to disrupt the activities of all other components which are directly or indirectly dependent upon it. The disruptive potential of a component can be taken as a function of the number of dependent components, and the length of the shortest paths by which they are dependent. Although both these variables are incorporated in the average graph distance of a component, the disruptive potential is not immediately apparent.

If, however, a component is "removed" from the system (i.e. by removing all its incident edges) and the average graph distances are recomputed, the absolute difference between these and the original measures provides an indication of the component's importance to the system. Should the remaining components only be able to exercise their dependencies through longer paths, the average graph distances will increase. Conversely, if fewer components are now connected, the average graph distances will decrease. The degree of fluctuation in the average graph distance of the remaining components is related to both these conditions, and it is the absolute difference in the average graph distances of the original and reduced systems which is an indication of the component's disruptive potential. In practice, the disruptive measure of a component is computed as the sum over all components, excluding the component in question, of the absolute difference between the average graph distance in the complete and reduced systems (Reed, 1970,p774). The mathematical representation of this formula is given in appendix C. By computing the disruptive measure of all the components in an environmental system, it is possible to rank the components in order of their disruptive potential to the whole system. This can be viewed as a ranked order of the components' ability to give rise to secondary impacts.

Disruptive measures, however, like the dispersion and accessibility measures, are not commensurate for systems with differing numbers of vertices. It is proposed therefore, that the measures be normalized by expressing them as a percentage of some assumed maximum value for a graph of the same number of vertices. In appendix C, a formula for the upper bound of the disruptive measure of a component has been derived. If components in different systems are to be compared, the disruptive measures can be normalized by expressing each as a percentage of the maximum disruptive value.

A procedure is given below for determining the rank order by disruptive measure of the components in any system :

- The minimum link matrix L[n] is derived from the complete adjacency matrix A. The average out-graph distance, G(i) for component i, is computed for each component in the system, and these values are stored in some manner.

- A component x is deleted from the complete adjacency matrix A by setting the elements of row x and column x to zero. The minimum link matrix, L[Nx] is recomputed from the reduced adjacency matrix Ax, and the average graph distances, Gx(i), are calculated for all the non-deleted components.

- The absolute value of G(i) - Gx(i) is computed and summed over all components except x (i.e. for all i except i = x). The sum gives the value of the disruptive measure for component x.

- The previous step is repeated for all x from 1 through to m, where m is the number of components to the system.

- All the component disruptive values can be normalized to give the percentage of total disruption, and the components may then be ranked in order of disruptive potential.

The disruptive measures above refer to in-disruptive measures. Out-disruptive measures can be computed in a similar way (in-graph distances are computed), and the normalizing procedure is unchanged. An out-disruptive measure is an indication of the disruptive potential of a whole system on one of its components, and could be interpreted as the susceptibility of a component to secondary impacts. For clarity, the terms "component disruptive potential" and "component susceptibility to disruption" are used for the in- and out-disruptive measures respectively.

3.6. Extracting Information From the Powering Process

3.6.1. Minimum and Non-minimum Paths

The elements of a matrix P[n] used in the powering procedure give the number of paths of length n connecting any two vertices of a graph. It is possible to construct, from all the levels of P[n] which are used to derive the MLM, a total path matrix T (Taaffe and Gauthier, 1973,p125) where :

$$T = P[1] + P[2] + P[3] + \ldots\ldots + P[N-1] + P[N]$$

The elements of matrix T specify the total number of direct and indirect connections between all the vertices of a system. Summing the values of T across any row gives the "gross vertex connectivity measure" (Carter, 1969,p45) of a vertex, which is similar to the out-accessibility derived from the MLM. The connectivity measures of communication networks have been used by Carter (1969,p45) and Pitts (1965,p18) to investigate the relative importance of network nodes. Taaffe and Gauthier (1973,p131) have recognized that the total path matrix has two major disadvantages, particularly for the study of communication networks :

1) Many redundant paths are included in the matrix values.
2) The indirect influence of a vertex is a function of the length of the connecting path, but summing over all P[n] gives an equal weighting to all paths, irrespective of their length.

Therefore, there is a tendency to favour the MLM (Taaffe and Gauthier, 1973,p132; Carter, 1969,p45) as it circumvents both problems by considering only the shortest paths. However, while redundant paths in communication networks are generally all paths longer than the minimum path, this need not be so for environmental systems. For example, in the system shown in figure 12, there are four paths by which "rooted vegetation" is dependent on "benthic fauna" :

1) rooted vegetation -> benthic fauna
2) rooted vegetation -> estuarine fish -> benthic fauna
3) rooted vegetation -> avifauna -> benthic fauna
4) rooted vegetation -> avifauna -> estuarine fish -> benthic fauna

If the greatest proportion of nutrients is not derived directly from "benthic fauna", then the shortest path may not, in fact, be the most important path. The cycling of nutrients may be faster, and therefore more important, via a longer chain of dependencies. Therefore the complex functional processes which define dependencies can give non-minimum paths a greater weighting than expected. For this reason there may be some value in considering the total path matrix for environmental systems. Unfortunately, the degree of redundancy in the total path matrix is not readily apparent, and it is considered preferable to restrict consideration to the MLM for the CI technique.

Weighting scalars (Katz, 1953,p41) have been used to give a more realistic interpretation of the indirect influence of longer than minimum paths in the total path matrix. Further, weighted graphs can be usefully employed to select minimum paths of maximum influence (Taaffe and Gauthier, 1973,p138). Both these approaches extend the subjective base of the approach, and will only be discussed in chapter 5 as possible extentions to the CI technique.

3.6.2. The Minimum Path Accessibility Measures

As the CI technique will be restricted to information about the minimum paths in a system, the total minimum path matrix, T(min), substitutes for the total path matrix, T. Elements of T(min) give the total number of minimum paths which connect any two vertices. For environmental systems, summing elements across a row of the T(min) gives the total number of minimum paths which make a component dependent on the system as a whole. Column summations indicate the total number of minimum paths which make the whole system dependent on a single component. These are respectively, the out- and in- minimum path accessibility measures, and the following procedure can be applied for their derivation :

* The procedures for determining L[N] and T(min) are performed simultaneously.

* Considering the powering process P[n] = P[n-1] x A, the following adjustments are made to the conventional mode of

matrix multiplication :

normally :

$$P[n](i,j) = P[n-1](i,1) \times A(1,j) + P[n-1](i,2) \times A(2,j) + \ldots$$
$$\ldots + P[n-1](i,m) \times A(m,j)$$

where m is the number of vertices in A.

however :

P[n](i,j) is only set to the sum of products if

$P[n-1](i,j) = 0$

* Once the minimum link matrix has been derived, P[N] = T(min), and the rows and columns can be summed for the out- and in-minimum path accessibility measures.

The method recognizes that it is the presence of non-zero elements (rather than the value of the elements) in P[n-1] which is necessary to derive all the minimum paths. Therefore, only the first non-zero value is carried over from P[n-1](i,j) to P[n](i,j), which is the total number of minimum paths between i and j.

3.6.3. Configuration of the Minimum Paths - ("tracks")

One of the major tasks of the CI technique is to expose the configuration of dependencies which make any component dependent on any other component. To achieve this, detailed information must be extracted during the MLM powering process, and stored in a retrievable manner. Once the question "how is component i dependent on component j" is asked, the stored data is accessed to define the relevant minimum path or paths; that is, a "track" is performed to expose the minimum paths which link i to j. The procedure is not straight forward, and is explained in a number of stages.

Matrix Multiplication and Higher Order Dependencies

P[n](i,j) is derived from the matrix multiplication of P[n-1] and A and the following formula applies :

$$P[n](i,j) = P[n-1](i,1) \times A(1,j) + \ldots + P[n-1](i,k) \times A(k,j) + \ldots$$
$$\ldots + P[n-1](i,m) \times A(m,j)$$

It follows that if P[n](i,j) > 0, then one or more of the products is greater than zero, which is possible, if, for one or more values of k, P[n-1](i,k) and A(k,j) are both greater than zero. Non-zero elements of a matrix signal the existence of a dependency, and for the above case :

if P[n-1](i,k) > 0 then i is dependent on k (i -> k)
and if A(k,j) > 0 then k is dependent on j (k -> j)

Therefore the product identifies the logical sequence of dependencies, i -> k -> j. Now, because the dependency of k on j is derived from the adjacency matrix, it must be a direct dependency. However, as the dependency of i on k is indicated by an element in P[n-1], it must be a dependency to the order n-1 (i.e. there are n-1 dependencies linking i and k). It is therefore necessary to inspect the products of P[n-1](i,k) and all other matrices down to P[1] to give a full description of the dependence of i on j.

A Practical Example

In figure 13, the powering procedure for deriving the MLM of the system described in figure 12 is shown. It may be noticed from the MLM that L[N](6,5) = 3, i.e. that component 6 (humans) has a

- During the powering procedure, whenever a minimum path is found (i.e. when P[n](i,j) > 0 while P[n-1](i,j) = 0), then the values of k which give non-zero products in the equation for P[n](i,j) are stored in S(i,j).

Once the MLM has been derived, the information contained in S is simply a comprehensive equivalent of that which was extracted in the example above.

Information Retrieval

For the general case of "how is i (dependent component) dependent on j (supporting component)", the following procedure extracts information from S to give all the minimum paths of dependence between i and j :

- The value, or values, in S(i,j) are considered. If, however, S(i,j) contains no values, then no path of dependency exists between i and j. If S(i,j) is equal to j, then there is a direct dependency between i and j. If any other value is stored in S(i,j), there is a higher order dependency between i and j, and the following steps of the procedure apply.

- Let the value in S(i,j) equal k; the dependency is i -> k -> ? .... ? -> j. Further information on the dependency between i and j is contained in S(k,j).

- The linkage of the components listed in S(k,j) on j are developed in a similar way, until for some k, S(k,j) is equal to j. This indicates that the last (direct) dependency on to j has been found.

- If more than one value is stored in any of the elements in S, then there is more than one minimum path. The above procedure is applied to each branch of the linkage until all branches converge at j.

In figure 13 a portion of S is shown for the system described in figure 12. All the dependencies which can be tracked using this portion of S are given. They show how all components are directly or indirectly dependent upon "avifauna" (component 5).

3.6.4. Critical Components

From T(min) in figure 13 it can be seen that two pairs of components are dependent by two minimum paths (rooted vegetation -> rooted vegetation; avifauna -> plant detritus). However, figure 14 shows how it is possible to have many minimum paths between any two vertices. If any of the vertices A through to N were removed from the system, 1 and 4 would still be linked by other minimum paths. This is not the case for vertices 2 and 3.

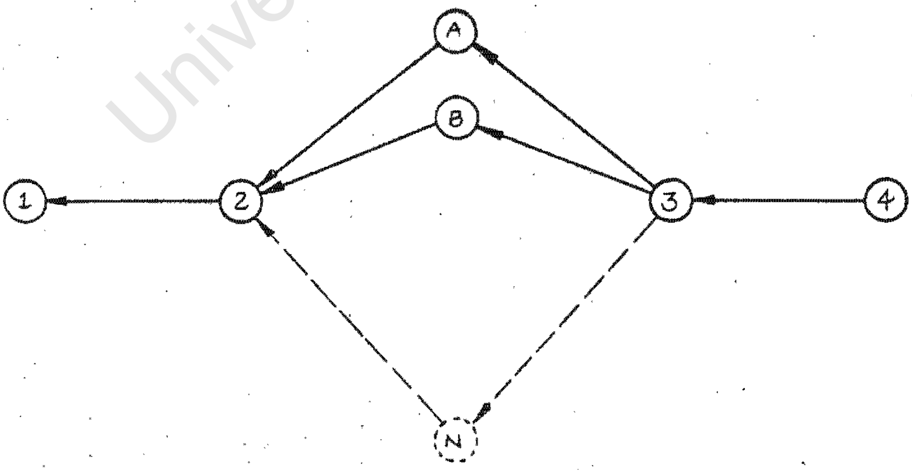


FIGURE 14. Diagramme of a graph to show critical- and cut-components.

Removing either of these would disconnect vertices 1 and 4. A vertex whose removal disconnects the shortest path between a pair of vertices is critical to that linkage. If there is only one minimum path between a pair of vertices, every vertex in the path is critical. Critical links apply to shortest paths only, and it is always possible that longer paths could provide alternatives to the critical paths in question.

For an investigation of secondary impact potential of components in an environmental system, it would be useful to know which components are critical to which dependencies. An iterative procedure, based on the tracking process described in the last section, can provide information on the critical components of a system :

* For all i and j, if there is a higher order dependency between i and j (i.e. L[N](i,j) > 1), a track is performed to give all the minimum paths of dependence.

* If some component k is critical to the path between i and j, the identities of i, j, and k are recorded as part of a summary of all critical components. A critical component is recognized as any component, k, in the path of dependence, for which there is only one value in S(k,j).

* While the process is repeated for every i and j, a total is kept for each component of the number of times it is critical to any path of dependency. The maximum number of paths to which a component can be critical is a function of the number of vertices in the system. A formula defining this function is developed in appendix C. Therefore the critical path totals for components in any system can be normalized by expressing them as a percentage of the maximum. This allows components of

differing systems (i.e. with different values of m) to be compared in terms of "critical importance".

3.6.5. Cut-components

If figure 14 represented a complete graph, vertices 2 and 3 would be cut-vertices. A cut-vertex is one which is critical to both minimum and non-minimum paths, and whose removal renders some part of a strongly or unilaterally connected system completely disjoint from the remaining system (Wilson and Beineke, 1979,p8).

The significance of minimum and non-minimum paths in environmental systems has already been discussed. Consideration has been restricted to minimum paths because the significance of non-minimum paths is not easily assessed. However, as the importance of cut-components is obvious and unambiguous, attention is given to their identification.

The following procedure, which can be performed simultaneously with the disruptive measure procedure, identifies all the cut-components in a system and the dependencies which they disrupt:

- The procedure to determine the disruptive measure of every component in a system has been described. L[N] is the minimum link matrix of the complete system, and L[Nx] is the MLM of the system with component x deleted. The following check is iterated over all i and j for every x :

  if L[N](i,j) is non-zero
  and L[Nx](i,j) is zero

  then it follows that component x has the ability to totally disrupt the dependence of i on j.

- Whenever component x is a cut-component in the path of dependence of i on j, the identities of i, j, and x are noted. This gives a summary of all the components which are cut-components, and the indirect dependencies which they have the potential to destroy.

- As with critical links, the total number of times a component can act as a cut-component may be recorded. The totals are normalized in a similar way to the critical path totals, allowing components of different systems to be compared in terms of "absolute critical importance".

3.7. Implementing the Component Interaction Technique

The CI technique consists of all procedures which are able to extract information pertinent to the identification of secondary impacts from a component interaction matrix. It is considered that the interpretation of the CIM, MLM, disruptive measures, tracks, and other ranked measures mentioned above, is sufficient for conducting a preliminary investigation of possible secondary impacts in any defined environmental system.

As the procedures which generate the information are intricate and tedious to perform, the technique is unavoidably computer bound. This has obvious disadvantages for what is intended as an elementary analytical procedure, and detracts from the general applicability of the technique. However, the subject of system inter-relationships is complex, and it is unlikely that an adequate analysis could be performed by simpler (non-computerized) approaches. Further, it is intended that the reliance on computers be compensated for by structuring and presenting the computer dependent parts of the technique with the following points considered :

- The unfamiliar format of computing procedures often results in resistance to computer usage (Munn, 1975,p87). Interactive programmes which offer a simple interface between the user and computer have been successfully employed to overcome this problem (Furniss, 1977,p1; Jain and Webster, 1977,p258).
- Although the presentation of output greatly affects the ease with which it is interpreted, an understanding of the modus operandi of the programme is also important. Consequently, a computer programme which is adequately described and documented gives the user a basis for interpreting the meaning and validity of all output.
- Sound programme documentation and the use of a universally accepted programming language can facilitate the transfer of a programme to differing computer systems. However, the implementation of an approach is probably better served by an exposition of the algorithms used in the programme.

An interactive ASCII FORTRAN programme has been written to test and display the capabilities of the CI technique. The technical details of the programme are discussed in the programme-manual given in the first section of appendix A, and a full listing of the programme follows in section two. The results of a complete analysis of the environmental system given in figure 12, and a reproduction of the programme/user commands which created the output, are given in the first section of appendix B.

The output of the programme has been structured to facilitate an effective analysis and interpretation of results. Information generated by the subprogramme which performs the dependency tracks is utilized in a plotting programme to provide a diagrammatic display of the minimum paths of dependence (see appendix B).

The literature on graph theory algorithms contains many examples of efficient procedures (in terms of computer time and storage space) for storing and manipulating graph data (Heap, 1972,p50), computing minimum paths (Dreyfus, 1969), and identifying cut-vertices (Read, 1979,p387; Tarjan, 1974). However, while the procedures used in the CI programme are not necessarily efficient in themselves, they are better suited to the multipurpose nature of the programme. The algorithms described in this chapter form the basis of the appropriate CI subprogrammes.

3.8. Summary

A set of procedures which constitute the CI technique has been described. The technique is able to extract information about the structure and higher order dependencies of an environmental system. A simple model of the system, in the form of a component interaction matrix, is the sole input of data to the technique. Consequently, the accurate construction of the model is crucial to the analysis (Matthews, 1975,p123). Further, the procedure merely extracts and structures information without providing a direct evaluation of secondary impacts for an EI analysis. Only after the CI data is incorporated into some EI analysis method, can an assessment of secondary impacts be performed.

These two important aspects related to the application of the CI technique, (i.e. the formulation of a CIM, and the incorporation of the output from the technique into various EI analysis methods) are discussed in the following chapter.

4. CHAPTER FOUR - APPLYING THE COMPONENT INTERACTION TECHNIQUE

4.1. Introduction

Having dealt with the theoretical aspects of the CI technique, points related to its practical application are now discussed. Guidelines are given for constructing a CIM, and for incorporating information from the technique into the various EI analysis methods covered in chapter two.

The CI results of three environmental systems are presented in appendix B. These are referred to in this chapter. Section one of appendix B is a full analysis of the illustrative estuarine system used in chapter three, and is employed to demonstrate the capabilities of the CI technique. Section two presents an abbreviated listing of results associated with a more detailed estuarine system. The detailed estuarine system has a "positive" and "negative" CIM. Unless otherwise stated, all references are to the analysis of the positive system, which is used to demonstrate the practical capabilities of the technique. For comparative purposes, section three contains excerpts from a CI analysis of the Nanaimo Port CIM (Lands Directorate, 1974).

4.2. Deriving the Component Interaction Matrix

The main problems of the CI technique are associated with the representation of complex systems through a list of components and an array of dependencies scored as zeros or ones. The following three sections address some of the problems associated with, and suggest certain guidelines for, the compilation of component on the subjective content of an EI analysis. Before an environmental system can be modelled as a CIM, a relatively detailed understanding of the system is required. In most cases this will ensure that a multidisciplinary team of assessors is engaged to conduct the study. The value of a team approach to EI analysis has already been affirmed. Further, atypical of other methods and techniques, the interdependent nature of the CIM would require each member of the team to have some insight into the structure and processes of the environmental system as a whole. It is felt therefore, that the formulation of a CIM can ensure an assessment of impacts which is based on a broad, but sound understanding of the system in question.

Regardless of the method to be used, the construction of a CIM could be usefully incorporated into the preliminary stages of any impact analysis, as it facilitates a careful consideration of the environment. The following guidelines relate to a listing of components which would be compatible with the CI technique, and the variety of methods with which it may be associated :

* The component list should be compiled by an interdisciplinary team. The training or experience of each member should be appropriate to some aspect of the environment, or project proposal, under consideration.

* The choice of components should represent the combined judgements of the team of assessors. Interaction among team members can be structured by a suitable evaluation technique to give a consensus of opinion on the final list.

* The component list must describe the environment for which the project is intended, and which the project may affect. It is essential therefore, that the assessors have a clear

understanding of the nature and scale of the proposed project. Predetermined checklists (e.g. those of the Leopold or Batelle methods) may be used as a guide to the choice of components. However, the list should be tailored to the conditions particular to the project and associated environment.

* Components should be included which allow an analysis of impacts in both the bio-physical and socio-economic environments. The choice of components to represent the socio-economic environment will be the most susceptible to bias (intentional and unintentional). For urban environments in particular, the subjective base of the component decisions could be extended by giving some form of representation to public opinion.

* By definition components should represent environmental features, rather than environmental processes. This condition can be relaxed for steady-state processes which are themselves susceptible to impacts. For example, the components "freshwater (/flow)" and "nutrients (/cycling)" in the detailed estuarine system, represent not only the physical features of freshwater and nutrients, but also the fundamental processes with which they are associated. If a process does not normally assume a steady-state, and is itself a result of certain impacts (e.g. sand-bar destruction, or river flooding), then it should not be included in, or as, a component for a CI analysis.

* Each component should be qualified by an explicatory description. It is necessary to have the sense and bounds of components defined so that the user can identify direct dependencies and interpret the results of the CI technique. For example, the "W/S biomass" component in the detailed estuarine system is representative of the biomass supported by the

estuarine catchment or watershed (W/S) area. The distinction between this and the biomass supported by the estuarine zone is significant in this CI analysis. Further, a concise descriptive summary keeps the subjective choice and definition of components open to external review. As none of the systems presented in appendix B are related to a specific environment or project, it is intended that the component labels will act as broad component definitions.

* If components are to be ranked using measures derived from the CI technique, it is necessary to have all components of comparable "weight" or "specificity". For example, a component representing all terrestrial vertebrates may have a biased dominance in a system which includes components representing distinct species. Difficulties may be experienced in judging the compatibility of bio-physical with socio-economic components, but a procedure will be developed later in this chapter for recognizing unduly "gross" components.

* The number of components used to model a system is normally a function of the detail required. However, large CIMs are difficult to analyse and interpret. The programme used in this thesis is capable of processing a CIM of up to fifty components. A CIM of fifty components has 2500 elements, which is a formidable body of data. It is possibly more efficient to model large or detailed systems first as a small number of aggregated components, and then to expand the components of interest into detailed subsystems. That is, the CI analysis is performed at two levels of detail.

4.2.2. Determination of Direct Dependencies

While the choice of components is an activity included in most EI analysis methods, the determination of all direct dependencies is particular to the CI technique.

The possible number of direct dependencies which must be considered in a CIM is the square of the number of components. The task is, therefore, more demanding than choosing components. Further, the validity of a CI analysis is absolutely dependent on the exactness in the identification of direct dependencies. Dependencies can generally be recognized by a process of objective reasoning. However, the "all or nothing" choice (1 or 0) for registering a dependency necessitates a value-based decision as to what degree of dependency should constitute the allocation of a 1.

The following guidelines are offered for the determination of direct dependencies :

* It should be the task of the assessment team to decide upon dependencies.

* Some practical method of considering and recording all possible dependencies should be employed. An effective approach is to construct a CIM with the components listed along both (x and y) axes, and to systematically "ask" a standard question for each element of the matrix. For example; "is the existence of component i (row component) in any way directly dependent on the existence of component j (column component)". A "yes" answer will result in the allocation of a one to element A(i,j) of the CIM. The consideration of all dependencies can be a tediously repetative task, especially for large matrices. The CI programme used in this thesis allows a user to enter data

directly as interactive responses to an exhaustive set of programme prompts. In this way a complete and structured consideration of dependencies is ensured (see appendix B, section 1).

- The direction of dependencies must be consistent. In this thesis the convention is to have row components as the dependents and column components as supporting components.

- The descriptive summary of components should be consulted to ensure that the correct sense of a component is used in the decisions on dependencies. A summary should also be constructed of the type and extent of all direct dependencies. The descriptive model given in figure 12 of chapter three could act as a brief dependency summary, and it is important to have some such record for the accurate interpretation of CI analysis results. Dependency summaries are not provided for the systems presented in appendix B.

- Only direct dependencies should be marked in a CIM. Components may be "short-circuited" if higher order dependencies are inadvertently recorded. For example, in the illustrative system, it could be considered that "benthic fauna" is directly dependent on "rooted vegetation" as a source of nutrients. However, recording such a dependency between "benthic fauna" and "rooted vegetation" would short-circuit the "plant detritus" component.

- It is possible to have a wide variety of dependencies in any one CIM (e.g. dependencies of food, shelter, economics, aesthetics). Further, it is also possible to have "positive" and "negative" dependencies. If component i is dependent on component j because j has a beneficial effect on i, then the i -> j dependency is positive. However, if j has an inverse control on i (i.e. an "increase" in j "decreases" i), then the i -> j dependency is negative. Both positive and negative dependencies are scored as ones in a CIM. Two reasons can be given in support of a separate consideration of positive and negative dependencies :

1) A track on the linkage between a pair of components may contain both positive and negative links. This can complicate the interpretation of the dependency.

2) A negative dependency may short-circuit a linkage of positive dependencies.

The detailed estuarine system given in appendix B, section 2, is modelled as two CIMs. The first gives only positive dependencies, which are recorded for all affirmative responses to the question; "is component i related to component j, such that an increase in j gives an increase in i". The second matrix contains only negative dependencies, which are taken to exist when an increase in j causes a decrease in component i, or viceversa. For example, the dependency, "phytoplankton" -> "industrial effluents", is a negative dependency.

Positive and negative dependencies may be represented within a single CIM, provided that the dual nature of dependencies is recognized during the interpretation of results.

* There may be some value in allocating a measure of degree to each identified direct dependency. The CI programme used in this thesis gives the user an option of recording dependencies on a binary (0 or 1) or weighted scale (0 - no dependency; 1 minimal; 2 - appreciable; 3 - complete dependency). All

weighted non-zero values are reduced to ones for the mathematical operations of the technique. The advantage of the weighting option is that the CIM and valency summaries (only) give some (subjective) indication of the importance of components. A weighted version of the illustrative estuarine system is given in appendix B, section 1.

4.2.3. Suitability of Component Definitions

Once the CIM has been constructed, an initial interpretation of some of the CI analysis results should be performed to assess the compatibility among the component definitions (in terms of specificity).

Certain components may dominate a CI analysis simply because they represent a significantly broader section of the environment than the other components. The specificity of components can be assessed from the valency and disruptive measure rankings. A distinct discontinuity in the ranked values of either measures could indicate an excessively gross component or components. However, it is possible that the importance of a component, rather than its broad definition, is responsible for high valency or disruptive values. An important component is one which is associated with some discontinuity, but which cannot be divided into suitable mutually exclusive subcomponents. For example, the "bacteria" component of the Nanaimo Port CIM has an out-valency of seventeen, which is significantly larger than the next ranked component ("birds of prey" at eleven). This probably explains why it has the highest value in the out-disruptive or susceptibility to disruption ranking. The "bacteria" component is not readily subdivided, and so represents an important aspect of the environmental system. The valency measures of the "nutrients (/cycling)" component of the detailed estuarine system do not show a marked discontinuity, but in both the in- and out-disruptive measure rankings, it is a decidedly dominant component. This indicates that the indirect influence of, and on, "nutrients (/cycling)" is substantial. Re-defining the component as organic and inorganic subcomponents could de-emphasize its indirect influence.

Conversely, it may be possible, and necessary, to aggregate certain components. For example, in the initial detailed estuarine system, it was found that "reptiles", "mammals", and "avifauna" depended upon, and were depended upon by, similar components. As none had excessive valency or disruptive values, they were combined to form the "terrestrial vertebrate" component.

Once the components have been adjusted if necessary, the validity of a CIM is largely dependent upon the accuracy in the allocation of direct dependencies. There is no explicit procedure for checking the validity of the direct dependencies.

4.3. The Component Interaction Technique and Impact Analysis

There are three different levels at which a CI analysis can be useful to an investigation of environmental impacts. They are the procedural, communicative, and interpretative levels.

4.3.1. Procedural Level

It was mentioned in the last section that, besides the purely analytical functions of the CI technique, the activities associated with the construction of a CIM can have a positive influence on the procedure for conducting an EI analysis.

4.3.2. Communicative Level

Many of the objectives of this thesis relate to an efficient communication of secondary impact potential of an environmental system to the evaluators of environmental impact. An MLM, as a constituent part of the CI technique, can communicate the extent of the interdependencies in an environmental system. Provided that there are not too many source and sink components (there are 3 source, and 1 sink, components in the detailed estuarine system), most elements in an MLM will be non-zero. An impact on a component j can, theoretically, influence every component i, for which L[n](i,j) is non-zero. Therefore, the arrangement of non-zero values in an MLM gives a concise indication of the secondary impact potential within the system.

While the MLM can display the existence of direct and indirect dependencies, the tracking facility can be used to show how any component is dependent on any other, and consequently, how an impact on a component can indirectly affect another to which it is not directly linked. Tracks are included in each of the three sections of appendix B.

4.3.3. Interpretative Level

Many of the results of the CI technique have little value unless they are interpreted in conjunction with the component definition and dependency summaries.

There are over 2000 non-zero values in the detailed estuarine system MLM, but not all of these need indicate important dependencies. The significance of any linkage is related to the degree of the weakest dependency. As an example, the MLM of the illustrative system shows that "humans" are dependent upon "avifauna" by a path of three links. The significance of the linkage is related to the degree of the weakest dependency. From the track given below (figure 15), it can be argued that the

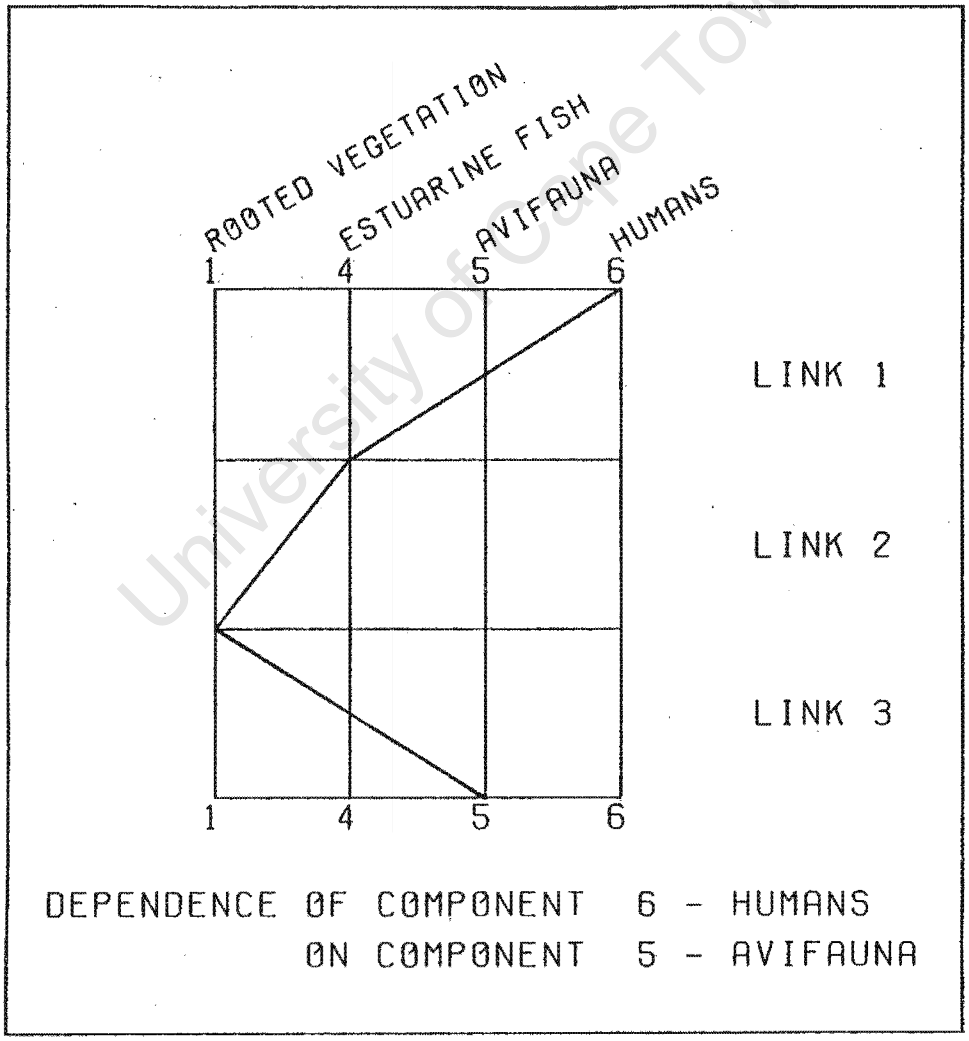


FIGURE 15. A "track" on the dependence of "humans" on "avifauna" in the illustrative estuarine system.

"humans" -> "avifauna" dependency is not very significant because the dependence of "rooted vegetation" on "avifauna" is not important (i.e. the biomass of rooted vegetation supported by nutrients supplied to the system as bird faeces and detritus is not great). The track diagram (figure 15) is output from the CI programme used in this thesis.

The significance of critical- and cut-components can be assessed from tracks of the dependencies listed in the critical- and cut-component summaries.

Once the primary impacts of a project have been associated with the relevant environmental components by an EI analysis method, tracks can be performed to gauge the severity of the associated secondary impacts. The MLM column for the "rooted vegetation" component in the illustrative estuarine system shows that a primary impact (e.g. destruction of rooted vegetation biomass by dredging activities) can affect all the components in the system. The following tracks summarize how every component is dependent on "rooted vegetation":

plant detritus -> rooted vegetation
benthic fauna -> plant detritus -> rooted vegetation
estuarine fish -> rooted vegetation
avifauna -> estuarine fish -> rooted vegetation
humans -> estuarine fish -> rooted vegetation

It may be contended from an inspection of the tracks, that the secondary impacts on the "plant detritus" and "benthic fauna" components are important in view of the total biomass which may be indirectly affected.

If a system is modelled as a large number of components, it may not be feasible to perform and interpret all the tracks relevant to secondary impacts (a direct impact on all but one component (industrial effluents), in the detailed estuarine system could affect 44 other components). For larger CIMs an interpretation of secondary impact potential is more easily obtained from a study of the various summaries and rankings provided by the CI technique.

The terms rankings and measures have been loosely used to describe output from the CI technique related to the valency; average graph distance; minimum paths; critical- and cut-positions; and disruptive potential of components. Each component in a system has a value for each of these measures. Therefore, for each measure, the components can be ranked in order of their associated values. All the measures are on an ordinal scale as they derive from a continuum dichotomized as two intervals; that is, the binary scaled representation of direct dependencies (Siegel, 1956,p25). Consequently emphasis must be placed upon the ranked position of components rather than on actual values.

In chapter three, the graph-theoretic meaning of each of the six measures was discussed. The valency, average graph distance, and disruptive measures have two forms : the "out" measure relates to the dependence of a single component on the rest of the system; and the "in" measure refers to the dependence of the system upon a single component. Where the particular form of these measures is not explicitly stated, the "in" version is implied as it is of greater interest to a generalized study of secondary impacts. The first five measures listed above have an indirect relevance to secondary impact analysis.

The valency measure is a total of the direct dependencies supported (in-valency), or originated (out-valency) by a component. Direct (primary) and indirect dependencies are similar in that both give rise only to secondary impacts. By definition, an impacted component can transmit only secondary impacts to other components, irrespective of the length of the paths of dependency. Nevertheless, direct dependencies are the basic part of all indirect dependencies and therefore, of all secondary impacts. For this reason, valency is a useful indication of the direct influence of, or on, a component.

In the detailed estuarine system, 46 out of 47 components (there is one sink component) all support direct and indirect dependencies on the same set of 44 components (there are three source components). However, the severity of the secondary impacts which may be associated with these 46 components could be seen as a function of their average graph distance. A low average graph distance indicates that the number of links in the paths of dependence is, on average, small. Short paths of dependence are more likely to be significant than longer paths. While this is most probably true for the bio-physical environment, it is uncertain whether shorter paths will always be more significant than longer paths in the socio-economic environment.

The in-minimum path accessibility measure (in-minimum path measure) for a component is the number of minimum paths connecting a component to all others which are dependent on it. This measure has little value for the investigation of secondary impacts because the influence of a component is not affected by the number of alternative paths which support a dependency upon it. The measure does, however, offer an interesting insight to the system's structure. Only one component in the detailed estuarine system has an in-minimum path measure of less than 47 (the number of components in the system). In fact few measures are less than

94, which suggests that on average, each dependency operates through two or more minimum paths.

The out-minimum path accessibility measure (out-minimum path measure) does have some relation to the susceptibility of a component to secondary impacts. A large number of minimum paths linking a component to others on which it is dependent could reduce the possibility of secondary impacts on the component. Alternatives to an impacted minimum path are more likely for components with a high out-minimum path measure.

The critical- and cut-component summaries and rankings have already received attention. The two measures are not mutually exclusive, as all cut-components must, by definition, be critical components. The cut-components of the detailed estuarine system are the "river mouth" and "beaches and dunes" components. They hold cut-positions in every dependency of "tides" and "currents and waves" respectively, on all other non-sink components. However, the dependence of "tides" and "currents and waves" on the estuarine system is probably not significant, and in this case the cut-component summary is not important. The critical-component summary for the detailed estuarine system has been abbreviated in appendix B as there are many critical positions in the system. However, it may often be possible to perform one track to show how a component is critical to many dependencies. If component A is critical to the dependence of B on C, and the B -> A -> C linkage is part of other longer dependencies, then a track on the longest linkage is likely to display a number of the dependencies to which A is a critical component. The number of critical- or cut-positions held by a component has a direct bearing on the component's potential to originate secondary impacts, but gives little indication of its susceptibility to secondary impacts.

It is proposed, with some reservations, that the disruptive measure is the most useful of the CI measures for a direct indication of a component's importance with regard to secondary impact potential, and susceptibility to secondary impacts (respectively the "in" and "out" forms of the measure). However, the disruptive measure does not qualify as an environmental index of secondary impact potential (or susceptibility). An environmental index has the form :

index = measure / standard

and is at least interval scaled (Inhaber, 1976,p5). The disruptive measures do not relate to any standard value, and are ordinal scaled. Further, no standard is immediately obvious as a suitable denominator for a secondary impact index. It is suggested in chapter three that the maximum bound of the disruptive measure could be used to normalize the disruptive scores. The component disruptive potential of "nutrients (/cycling)" in the detailed estuarine system expressed as a percentage of the upper bound (1104.0) is approximately 1%. A similar measure for "rooted vegetation" in the illustrative estuarine system is 25% . Both components are ranked first in their respective disruptive rankings. The large difference in values is related to the fact that a component in a small system is more likely to have a greater influence than one in a large system. This does not mean however, that a component in the large system is any less important. For this reason, it is not possible to specify some normalized value above which a component could be considered "important" in terms of secondary impact potential or susceptibility.

A standard value for the denominator of a disruptive index which could be of some value to secondary impact studies would have to be some function of the number of components in the system. No such function is proposed for the CI technique. It is suggested rather, that the rankings of the other measures be used to interpret the significance of the disruptive measures.

The table below (table 1) displays the Spearman rank correlation coefficients (Siegel, 1956,p202) for the association between component rank positions given by the six "in" measures. Data was taken from the detailed estuarine system, and the analysis performed by computer using a BMDP statistical package (Health Sciences Computing Facility, 1977). All coefficients above 0.350 have a level of significance greater than 0.01 for a one tailed test - i.e. coefficients > 0.350 indicate a significant association of ranks (Siegel, 1956,p210). Most of the coefficients related to average graph distance are negative

	VALENCY MEASURE	GRAPH DIST MEASURE	MINIMUM PATHS	CRITICAL POSITIONS	CUT POSITIONS	DISRUPTIVE MEASURE
VALENCY MEASURE	1.000					
GRAPH DIST MEASURE	-0.713	1.000				
MINIMUM PATHS	-0.105	0.333	1.000			
CRITICAL PATHS	0.414	-0.416	-0.216	1.000		
CUT POSITIONS	-0.055	-0.047	0.097	0.258	1.000	
DISRUPTIVE MEASURE	0.408	-0.352	-0.005	0.761	0.287	1.000

TABLE 1. Spearman Rank Correlation Coefficients

because there is an inverse relationship between the measure and disruptive potential. The bottom row of coefficients indicates the extent to which the disruptive potential ranking correlates with the rankings of the other five measures. For the detailed estuarine system, the disruptive potential of a component is closely associated with the number of critical positions held (coef. = 0.761), and to a lesser extent, the valency of the components (coef. = 0.408). There is also a fair negative correlation between the average graph distance and disruptive potential (coef. = -0.352). However, the coefficient for average graph distance and valency (coef. = -0.713) shows that these two measures are not fully independent. As expected, the significance of the minimum path measure to the disruptive potential of a component is minimal (coef. = 0.005).

It is interesting to note that the valency measure is not strongly correlated to the disruptive measure. A coefficient of 0.408 implies some association, but also shows that the disruptive measure is not fully accounted for by valency. This is evidence that there are many factors, not all of them obvious, which contribute towards a component's importance with regard to secondary impacts. This in turn gives some support for the necessity of performing a CI analysis.

Although the disruptive measure cannot be used as an index, or to compare the components of different systems (an unlikely requirement in any event), it can "flag" those components within a system which are of secondary impact importance. This is a particularly useful facility for a preliminary analysis of impacts.

There is no rigorous procedure for extracting information from the results of a CI analysis to give a clear indication as to which components are important, and for what reasons. This is seen as an advantage of the technique. A thorough understanding of the environment and careful thought, rather than strict compliance with a prescribed procedure, is required by the technique for an identification of secondary impacts.

The usefulness of a CI analysis is reliant upon a loosely structured interpretation of results. The interpretative dimension of the CI technique can be expanded if the CIM data is easily modified. The interactive programme used in this thesis allows the user to delete and add components, and to adjust component definitions at will. These facilities can be used to investigate the effects of different components, and component definitions upon the analysis of the modelled system.

4.4. The Analysis of Secondary Impacts

This section deals specifically with the compatibility of the CI technique with the EI analysis methods discussed in chapter two. The technique has some value to all the methods. While an effort has been made to minimize the subjective base of the CI technique, there is a significant reliance upon the interpretation of data for an analysis of secondary impacts. Rather than reduce the flexibility of the technique by stipulating set procedures for interpreting data, a general set of guidelines is proposed. The guidelines describe the possible contributions that the CI technique could make to an analysis of secondary impacts within the framework of each EI analysis method.

4.4.1. Ad Hoc Approaches, Simple and Descriptive Checklists

Ad hoc methods have not been widely accepted as a general approach to EI analysis, but are sometimes developed specifically for commonly occurring projects (e.g. the ad hoc approach to waste water treatment facilities, of FitzPatrick et al, 1978). Ad hoc approaches offer no guarantee that all potential impacts will be considered. This is an area of some concern (Cook, 1977,p17), and is one in which the procedural attributes of the CI technique could be gainfully employed. Construction of a CIM offers a structured procedure for determining the broad areas of concern related to a project. As ad hoc methods are intended for elementary analyses of environmental impact, a CIM may be appropriate, but not a full CI analysis. However, the communicative facilities of the technique (i.e. MLM and tracks) may be useful for exposing potential secondary impacts related to more significant projects.

Most of the remarks above apply equally to simple and descriptive checklists because of their similarity to ad hoc methods.

4.4.2. Scaled Checklists and Presentational Matrices

Both these broad categories of analysis methods employ a list of environmental attributes against which impacts are evaluated. Normally, only primary impacts are considered. The attribute lists can be compiled within a CI analysis procedure, with advantages which have already been discussed. The secondary impact potential and susceptibility of components can be evaluated from the various CI measures.

Scaled (as opposed to weight-scaled) checklists do not offer an obvious means for registering secondary effects. Impact scores may be arbitrarily scaled to reflect the degree of associated secondary impacts. Alternatively, information pertaining to secondary impacts within the system (an MLM, relevant tracks and rankings) can be included in the EI statement (particularly when an aggregated impact score is derived).

Weight-scaled checklists, and specifically the environmental evaluation system (Dee et al, 1973), are well suited for distinguishing between components of differing secondary impact potentials. In the environmental evaluation system, seventy-eight parameters (components) are weighted by a multidisciplinary team according to their relative importance. Besides using a delphi approach to determine the relative weights, there is little that can substantiate these scores, or insure that the primary and secondary importance of components is considered. The versatility of the environmental evaluation system would be increased if the list of environmental parameters could be adapted to particular environments and projects. Employing the CI technique in the preliminary stages of an analysis would provide a structured approach for determining the component list. Further, the activity of deriving a CIM, and the information which can be obtained from a CI analysis, could substantiate and consolidate the basis from which the value functions and parameter weights are derived.

If the checklist were to be used in its original form, the seventy-eight parameters could be used as the component list for a CIM. An interpretation of CI results could determine whether the original parameter weights do in fact reflect the secondary impact potential, and susceptibility to secondary impacts of the environmental parameters. Should a CI analysis show one or more components as being exceptionally important, the "red-flag" system could be employed to highlight this fact in the analysis report. Where necessary, the communicative facilities of the CI technique can be used to support and explain impact scores which are high because of associated secondary impacts.

Presentational matrices offer various alternatives for representing secondary impacts once a CI analysis has been performed :

* Each component can be assigned some indicator to represent the probability or severity of possible secondary impacts. Some manner of shading the matrix columns assigned to each component is favoured. Symbols or characters could be used if the secondary impact assessments satisfy ordinal scale conditions. Numeric indicators should be avoided as interval scaled scores are not possible.

* Fuggle's combinative matrix (Fuggle, 1979) offers some means of representing secondary impact potential, and susceptibility to secondary impacts within each cell of a matrix. This has added advantages to the former method as the secondary impacts associated with a component are normally related to the type of impact. For example, a reduction in the "avifauna" component of the illustrative estuarine system, caused say, by increased noise levels (traffic, boating, recreationalists etc.) would give rise to different secondary impacts than if the avifauna populations were reduced by increased industrial effluent discharges. The former impact may decrease the diversity of the bird populations, but the biomass may remain constant, or even increase. The latter impact may decrease diversity and biomass. Suitable tracks would have to be performed, and the various

measures interpreted, to give an indication of the secondary impacts. While it may appear advantageous to investigate the secondary impacts of only highly probable, or important primary impacts, the possibility of synergistic effects from minor impacts should not be ignored.

* One of the major advantages of matrices is their ability to summarize and display primary impacts. This facility can be extended within the matrix framework. Figure 16 displays how a

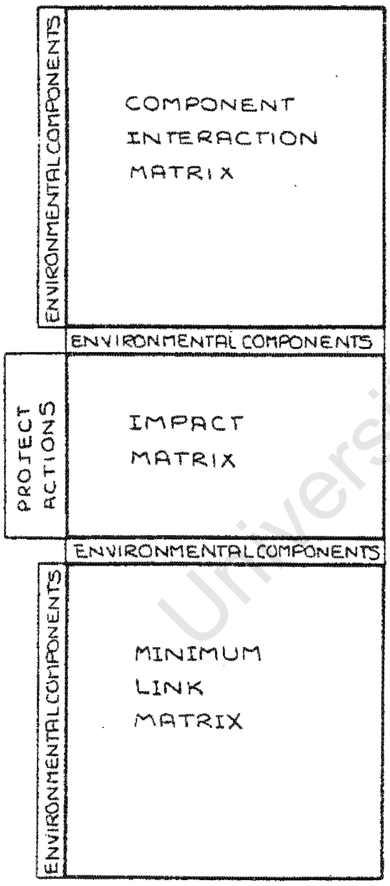


FIGURE 16.

Combining the component interaction, minimum link, and impact matrices.

CIM and MLM can be adjoined to an impact matrix to give some visual indication of the origins of, and potential for, secondary impacts.

4.4.3. Overlays

Only a superficial consideration of secondary impacts can be achieved within the overlay approach. If a CIM is constructed for the environmental system in question, components can be aggregated until a suitable number of broad parameters have been derived. Choosing the mapping parameters in this fashion ensures that most aspects of the environment are given some representation in the individual overlays, and composite map. Further, an indication of the secondary impact potential, and susceptibility to secondary impacts, of each parameter can be assessed from a general overview of results from a CI analysis of the system. A CI analysis would be most useful to a weighted overlay approach (possible when computerized techniques are used to combine the parameter maps). Information from the CI technique can provide the insight required for weighting the parameters to reflect their secondary impact importance.

4.4.4. Networks, System Diagrams, and Models

The CI technique, networks, system diagrams and models could be seen as as a complementing hierarchy of approaches for the assessment of secondary impacts. The CI technique is a simple means of modelling an environmental system, and for exposing all the possible indirect dependencies. The various measures of the technique, and in particular, the tracking facilitiy, can be used to screen every indirect dependence for those which may be important secondary impact pathways. Important pathways can be incorporated into a framework for a network analysis. Employed in this manner, the CI technique provides a substantive procedure for constructing networks in which all significant dependencies are represented. A network can in turn be used as the basis for a system diagram. The direct dependencies between components are quantified in terms of energy transfers, and system diagrams therefore give some representation to the processes which define direct dependencies. An investigation of total energy transfers, critcal- and cut-position measures, and disruptive measures may bring to light certain subsystems of components and processes whose importance and complexity would support a more detailed analysis. For example, from the critical and disruptive measures in the detailed estuarine system analysis, the "nutrient (/cycling)" process, and the components with which it is associated, would appear to be particularly important. Simulation or similar models could be constructed to provide a greater depth of analytical detail for subsystems of obvious importance. Further, modelling the processes of a system is possibly the only approach by which secondary impacts may be objectively quantified.

4.5. Summary

In the context of an adaptive approach to EI analysis, the CI technique offers a much needed means of considering secondary impacts at the preliminary stage of an EI analysis. The technique can be used in conjunction with all the major EI analysis methods to identify and communicate information about secondary impacts. The usefulness of the technique is largely dependent upon the adequacy of the environmental model (the CIM) from which the CI data is extracted, and the interpretative adroitness of the EI analysts.

The following chapter presents some ways in which the capabilities of the technique itself can be extended. In most cases, the extentions expand the subjective base of the technique. Attention is also given to other approaches to secondary impact analysis which achieve a greater level of analytical detail.

The interactive nature of the computer programme used in this thesis to perform the CI analysis was found to have a beneficial effect on the analytical power of the CI technique. Possible extentions and improvements to this programme are given in the programme manual in section 1 of appendix A.

5. CHAPTER FIVE - EXTENTIONS AND ALTERNATIVES

5.1. Introduction

Most of the aims and objectives for this thesis, which are outlined in chapter one, are fulfilled by the CI technique. The reliance on computing facilities remains a minor concern, but appears to be an unavoidable consequence of any detailed consideration of secondary impacts.

Two of the objectives pertaining to the development of a suitable secondary impact technique require a minimal reliance upon subjective judgements, and on analytical resources. If these restrictions are relaxed, two extentions to the CI technique may be feasible, and certain alternative approaches could be favoured.

None of the extentions or alternatives have been tried or tested within this study. Discussion is therefore brief, and intended as a broad guide to possible further investigations or developments.

5.2. Considering Non-Minimum Paths

The distinction between minimum and non-minimum paths was discussed in chapter three, where it was decided that the CI technique would consider only minimum link paths. The cut-component measure is an exception to this rule. There are two reasons why it may be advantageous to extend the CI technique capabilities to consider non-minimum paths in some way :

1) Although it is probable that minimum paths define realistic

dependencies in the bio-physical environment, this may not always be true for socio-economic systems.

2) In terms of impact studies, longer than minimum paths have some significance. If a critical component is impacted, there may exist some longer than minimum path or paths which offer feasible alternative linkages to the disrupted dependency.

The total path matrix was defined in chapter three as one in which each element, T(i,j), represents the total number of alternative paths (both minimum and longer) which link i to j. Using the notations defined in chapter three, an element of the total path matrix T, is calculated as :

$$T(i,j) = P[1](i,j) + \ldots + P[n](i,j) + \ldots + P[N](i,j)$$

where N is taken as the diameter or solution time of the system. (The graph-theoretic terms used here are defined in chapter three.)

Rows and columns of T can be summed to give respectively, the out- and in-gross vertex connectivity measures (Carter, 1969,p45), which are similar to the accessibility measures derived from the MLM. The gross vertex connectivity measures can be converted to "average connectivity measures" by dividing them by the number of non-zero elements to the summation. The average connectivity and average graph distance measures are similar, and the former can be used to calculate a disruptive measure (as described in chapter three) which gives consideration to all the possible paths of dependence between all components of a system.

The accessibility and connectivity measures have been compared in various communication network studies (Taaffe and Gauthier, 1973,p125; Carter, 1969; Pitts, 1965), and found to give noticeably different ranks to network vertices. However, it would appear that the total path disruptive measures have not been previously applied.

Two broad inconsistencies, mentioned in chapter three, detract from the usefulness of the total path measures :

1) Many redundant paths may be included in the total path matrix values.
2) It is possible that there is an inverse relationship between the length (in links) and importance of a dependency, but linkages of all lengths are given an equal weight in the total path matrix.

Further, there are different ways by which a path can be redundant, each of which has a different significance to secondary impacts. Consider the following linkages between components A and X of some system :

1) A -> B -> X
2) A -> B -> C -> X
3) A -> B -> C -> D -> X
4) A -> D -> E -> X
5) A -> B -> E -> B -> X

The first is the minimum path linkage between A and X. All the other paths are redundant in the sense that a shorter path exists. The redundancy of the third path is "magnified" by the fact that it would not offer an alternative to the second linkage if components B or C were impacted. This is not the case for the fourth linkage, as it offers an alternative path which does not include components B or C. The fifth linkage is redundant in the strictest sense, as the path operates through the same component twice (i.e. component B).

Garrison (1960, cited by Taaffe and Gauthier, 1973,p129)has used a weight-scaled total path matrix (first described by Katz, 1953) to study the connectivity of communication networks. The introduction of a scalar weight 's' to the powering process has the effect of reducing the influence of indirect and redundant paths. The weight-scaled total path matrix, Tw, is derived as follows:

$$Tw = s^1 P[1] + s^2 P[2] + \ldots + s^n P[n] + \ldots + s^N P[N]$$

$$\text{where } 0 < s < 1$$

There are no rules or guidelines for choosing the value of s (Haggett and Chorley, 1969,p42). It is clear however, that a low value of s will give less emphasis to the higher powered matrices (i.e. to the longer paths). The degree of subjectivity in the choice of a value for s can be controlled if there is some understanding of the probabilities related to the various forms of redundant paths. For environmental systems, the occurrence of redundant paths depends upon the structure of a system, and there is no obvious means of assessing their predominance.

If a value for s can be chosen with some confidence, the weight-scaled total path matrix can be used to derive disruptive measures which are not only based on a consideration of minimum paths, but on all paths of dependence. The advantage of extending the approach in this manner should be weighed against the disadvantages of increasing the subjective content of the analysis.

5.3. Weighted Dependencies

The possibility of weighting direct dependencies was superficially considered in chapter four, mainly for the purpose of displaying the importance of components in the CIM and valency summaries. The "all or nothing" decision for registering a dependency is easier if some range of alternative scores is offered. The weighted values in such a system are not used in the mathematical procedures of the CI technique (i.e. all weighted values are reduced to ones).

The problem of minimum and non-minimum paths could potentially be resolved if weighted dependency scores were used in the mathematical procedures of a CI analysis. Presently, the CI technique considers the length of a path as a function of the number of constituent links, and all links are given an equal status. It is the task of the analyst to interpret the relative importance of links. The interpretation of link importance occurs after a CI analysis has been performed, normally when specific tracks are considered. None of the CI measures can be interpreted in the light of the variable importance of direct dependencies. Should it be possible to weight the direct dependencies before an analysis, the relative importance weights can be used as information by the CI procedures to ensure that only the most important shortest paths are considered in the analysis.

There are three levels at which weighted dependencies can be allocated and assessed, related to whether the weights are scored on an ordinal, interval, or ratio scale.

Ordinal Scale - "importance weights"

In the illustrative estuarine system (see figures 12 and 13 of chapter 3), there are two minimum paths which make "avifauna" dependent upon "plant detritus". They are :

1) $\text{avifauna} \xrightarrow{2} \text{benthic fauna} \xrightarrow{3} \text{plant detritus}$

2) $\text{avifauna} \xrightarrow{2} \text{estuarine fish} \xrightarrow{1} \text{plant detritus}$

Subjective weights have been assigned to all the direct dependencies (above each dependency arrow in the tracks) according to the following conventions: 1 - minimal; 2 - appreciable; 3 complete dependence. Assuming that the weights satisfy ordinal scale conditions, the fist linkage can be recognized as the more important shortest path, simply because a "3" dependency is more important than a "1" dependency. This is a rather simple example when compared with the following hypothetical tracks :

1) $A \xrightarrow{2} b \xrightarrow{2} c \xrightarrow{2} X$

2) $A \xrightarrow{1} e \xrightarrow{3} X$

In this situation ordinal scaled weights are of little use, because they may not be added and the totals compared (Siegel, 1956,p24). Therefore there is no explicit means of judging the relative importance of the linkages.

Interval Scale - "distance" weights

If however, the weights could be scored on an interval scale, a measure of "distance" is assigned to each dependency and the weights can be added to find the "shortest" linkage. For this approach, the weights must be an inverse of those used above (i.e.

1 - complete; 2 - appreciable; 3 - minimal dependence). This gives the linkages as :

1) $A \xrightarrow{2} b \xrightarrow{2} c \xrightarrow{2} X$ sum = 2+2+2 = 6

2) $A \xrightarrow{3} d \xrightarrow{1} X$ sum = 3+1 = 4

Adding the weights of each linkage shows the second path to be more important. As the weights have a connotation of distance, many communication network algorithms are applicable for finding the most important "shortest" paths (Taaffe and Gauthier, 1973,p139; Dreyfus, 1969).

Ratio Scale - "flow" weights

The concept of distance is not, however, well suited to environmental systems. Consider the following tracks :

1) $A \xrightarrow{1} b \xrightarrow{3} c \xrightarrow{1} X$ sum = 1+3+1 = 5

2) $A \xrightarrow{1} d \xrightarrow{2} e \xrightarrow{2} X$ sum = 1+2+2 = 5

Both these linkages are of equal importance if the "distance" analogue is used. However, it is possibly more correct for the study of environmental systems to consider dependencies in terms of "flow". In the first linkage above, the dependencies of A on b, and c on X are complete, but the minimal dependence of b on c is the regulating " weakest link" in the dependence of A on X. If the weights were rather expressed in terms of "dependence flow" the linkages become :

1) $A \xrightarrow{3/3} b \xrightarrow{1/3} c \xrightarrow{3/3} X$ product = 3/3 x 1/3 x 3/3 = 0.33

2) $A \xrightarrow{3/3} d \xrightarrow{2/3} e \xrightarrow{2/3} X$ product = 3/3 x 2/3 x 2/3 = 0.44

and the second linkage is the more important shortest path, because the product of its weights indicates a greater "flow" of dependence. It is only possible to multiply weights if they are scored on a ratio scale (Siegel, 1956,p29).

The three examples above use a 1 - 3 scale for weighting. This is in keeping with the CI programme's weighting option, but there is no reason why different scales should not be used.

From the discussion above it may be concluded that the advantages of a weighted CIM can only be realized if the weights are scored on an interval or ratio scale. Interval scaled CIMs are possible, and have been employed by Ross (1976) in an attempt to extend the Nanaimo Port trans-shipment EI analysis (see chapter two). The resource intensive techniques of pairwise-ranking and multidimensional scaling were applied to derive the scaled weights. No record of a ratio scaled CIM has been found.

The mathematical procedures for processing a weighted CIM are not excessively complicated or lengthy. The various algorithms presented in chapter three would remain largely unaltered. It would be necessary however, to replace the conventional matrix multiplication procedure by some boolean equivalent, related to how the weights are to be interpreted (Taaffe and Gauthier, 1973,p141).

Although it is possible to process weighted CIMs, the effort involved in deriving interval or ratio scaled weights may detract from such an approach. Weight scaling would be particularly difficult if bio-physical and socio-economic dependencies must be considered within a single weighted CIM.

Should a weighted analysis be favoured, it may be simpler and less contrived if the subjective judgements of a team of assessors were structured by some evaluation technique to provide the direct dependency weights. The choice of analogue to be used in the powering procedure (i.e. "importance", "distance", or "flow"), would depend upon the degree of confidence in the allocation of weights.

A decision as to whether a weighted analysis should be employed in preference to a conventional CI approach would have to consider if the greater degree of subjectivity and analytical effort is worth the additional detail. Further, entirely different approaches may in fact be more useful for similar commitments of analytical resources.

5.4. Input-Output Models

The CI technique, in either its binary or weighted form, cannot give a direct indication of secondary impacts. Only once the primary impacts on a system have been identified, can the CI technique measures and tracks be interpreted to assess the extent of the associated secondary impacts. An input-output model (see "Mathematical Matrices", chapter 2), is a sophisticated form of a CIM in which the component lists are the producing and consuming sectors of a system. The values of the matrix elements reflect the degree to which a row sector supports a column sector (Leontief, 1970). Conventionally, the sectors are economic sectors, and the element values are coefficients of input to the supported sector per unit output of the producing sector. Changes in the system can be modelled by altering the total output or input values of the appropriate sectors. Thereafter, mathematical operations (matrix inversion) can be performed to produce a matrix which reflects how the production and consumption of all other sectors is affected by these changes. If it were possible to score direct dependencies at a level comparable to a dependence coefficient, the facilities of an input-output matrix could be used to give a direct representation of secondary impacts within a framework similar to that of a CI analysis. Such an analysis could be, and has been, performed on socio-economic systems (Lee and Fenwick, 1973; Hite and Laurent, 1971). However, it is unlikely that dependence coefficients could be derived for a combined analysis of bio-physical and socio-economic systems, without extending the subjective content of an analysis beyond reasonable limits. A heavy reliance on value judgements would be necessary as there are no common units by which the "production" and "consumption" of bio-physical and socio-economic "commodities" can be evaluated. Nevertheless, the input-output matrix is potentially the most powerful approach to secondary impacts within the field of mathematical matrices.

5.5. A Brief Note on Induced Secondary Impacts

In chapter one, a distinction was made between secondary and induced secondary impacts. No attention has been afforded to induced secondary impacts because of their widely varying form of occurrence. However, if the sense of a CIM where changed to that of an "activity interaction matrix" (AIM), a superficial consideration of induced secondary impacts is possible. For an AIM, the component lists are replaced by lists which reflect the full range of probable activities related to the project, and activities which the project may induce (e.g. clearance of site, access roads, waste removal, service facilities, commercial growth). The interaction between all activities is investigated, and if a row activity could be directly "stimulated" by a column activity, a one is assigned to the appropriate matrix element. The tracking facility of the CI technique could then be used to display how any activity may induce some other listed activity. That is, the shortest sequence of events linking any two occurrences can be exposed.

5.6. Summary and Conclusions

Various useful extentions to the CI technique are possible if the restrictions on the subjectivity of an analysis are relaxed. The extentions relate to weighting the CIM information in some manner. The technique can also be applied to an investigation of induced secondary impacts.

The strength of the unweighted CI technique is its ability to structure a preliminary investigation of secondary impacts. As such it is uncomplicated, and compatible with a wide range of EI approaches. Extending the technique may detract from its broad applicability. Further, the advantages of the adaptations may be more easily achieved if alternative techniques or methods are employed.

In chapter four, a complement of three alternative methods was discussed (networks, system diagrams, modelling procedures). Network methods give a concise summary of secondary impact paths and are an efficient means of communicating secondary impact information. A certain degree of quantification can be achieved by transforming networks to system diagrams. However, it is recognized that modelling procedures, and simulation models in particular, are a favourable alternative for a detailed analysis of secondary impacts.

The fact that many approaches to secondary impact analysis are computer-bound has been viewed as something of a disadvantage. However, the advent of micro-computers allows analysts of environmental impacts cheap and easy access to computing facilities. Therefore, it could be expected that the acceptability of computerized approaches will increase for all levels (preliminary to detailed) of EI analysis.

6. CHAPTER SIX - SUMMARY AND CONCLUSIONS

6.1. Résumé

This thesis is concerned with the identification of secondary (or higher order) impacts for environmental impact analysis. A literature review has exposed the need for a preliminary environmental analysis approach able to consider secondary impacts. Therefore, a computerized technique has been developed which is able to structure a preliminary investigation of secondary impacts.

6.2. The Literature Review

The following issues, central to environmental impact analysis, were revealed from a review of the literature :

* Secondary impacts can form a large part of the ultimate environmental impact of a project.

* Of the range of environmental impact analysis methods that exists, none is able to give adequate consideration to secondary impacts, particularly during the formative stages of a project's design.

* Methods offering a quantification of impacts tend to be favoured in countries where environmental impact assessment is enforced. In countries where there is little or no legislation covering environmental impact assessment, the most favoured approaches are likely to be those which offer an effective means of

identifying, but not necessarily quantifying, environmental impacts.

- Environmental interests are best served when impacts can be identified during, rather than after, the design of a project. Therefore, in addition to the more detailed methods used for a final assessment of impacts, a need exists for comprehensive, but simple preliminary impact analysis approaches.

- As no methods are able to meet all the requirements for an adequate environmental impact analysis, an adaptive approach would be of value. For an adaptive approach, methods and techniques are selectively combined to meet the requirements and conditions of particular impact assessments. In this way it is possible to balance the strengths of some approaches against the weaknesses of others.

Consequently, this thesis has as its main objective the development of a preliminary environmental impact technique able to identify the secondary impacts of a project within the framework of an adaptive approach to impact analysis. The component interaction technique was developed to this end.

6.3. The Proposed Technique

The basis of the approach is a component interaction matrix. The matrix models the environment as a list of environmental components (arranged along both axes of the matrix) and an array (the actual matrix) of ones and zeros. The presence (1) or absence (0) of a direct dependency between all the listed components is recorded in the matrix. The matrix represents the sum total of input data to the technique, and it is suggested that the model be constructed by a multidisciplinary team of impact assessors.

Computerized matrix powering procedures are able to structure the data to facilitate an investigation of the secondary impact potential in the system. The technique provides two main forms of information :

* All higher order dependencies in the system can be recognized and traced out (i.e. all the shortest paths of dependence can be exposed). Once the primary impacts of a project have been identified (using some conventional impact analysis method), the range of possible secondary impacts can be assessed from a study of the paths which link the impacted components to others which are dependent on them.

* Certain measures are derived which can be used to "flag" those components of the system which have a potential for initiating, or which are particularly susceptible to, secondary impacts.

All information is explicit and can be substantiated by referring back to the component interaction matrix data. An emphasis is placed upon the interpretation of data, and no quantification of secondary impacts is attempted.

Although the component interaction technique is computerized, it was considered that the increasing availability of resource efficient computing facilities, and the advantages of using interactive programming approaches, should minimize the restrictions on implementing the technique.

It was recognized that problems related to value judgements are unavoidable in any approach to environmental impact analysis. The technique, however, has a minimal reliance on subjective judgements. Further, the procedures for constructing the interaction matrix can be used to substantiate and control the subjective content of an impact study.

While the results of a component interaction analysis offer a means for considering secondary impacts, the activity of contructing a matrix model does much to ensure that a preliminary analysis of impacts is based on a comprehensive and structured consideration of the environment. These two attributes of the technique support its application in conjunction with other methods of impact analysis.

Various extentions to the approach have been considered in this thesis, particularly in the form of weighting the direct dependencies in some manner. However, it was concluded that the technique is best served by a binary representation of direct dependencies. Extending the technique would increase either the subjective content of an analysis, or the dependence on analytical resources. Nevertheless, the technique is able to support, or form the basis of, more detailed analyses of impacts by advanced analytical methods.

6.4. Conclusions

The major objective of this thesis is realized in the component interaction technique in so far as it isolates those features related to secondary impacts needing more investigation. However, it is recognized that the detailed analysis and evaluation (quantification) of secondary impacts is an area still requiring attention.

In conclusion, it is held that the proposed technique is a useful and needed addition to the range of existing approaches for analysing environmental impacts. The component interaction technique constitutes an effective tool for conducting a preliminary investigation of secondary impacts.

7. REFERENCES

Anderson FR, 1973. NEPA in the Courts : A Legal Analysis of the National Environmental Policy Act. Johns Hopkins University Press, Baltimore, ISBN 0-8018-1510-X, xv+324p.

Barile DD, 1978. Energetics and Environment Assessment; in Environmental Impact Analysis : Emerging Issues in Planning, Hutchings BL and Jain RK (eds). University of Illinois Press, Urbana, (Proceedings of Conference on Environmental Impact Assessment, Monticello, Illinois, 1977), ISBN 0-252-00696-8, pp. 193-203.

Berge C, 1964. The Theory of Graphs and its Applications. Methuen & Co. Ltd., London, translated by Doig A, x+247p.

Bisset R, 1978. Quantification, Decision-making and Environmental Impact Assessment in the United Kingdom. Journal of Environmental Management, Vol. 7, 1978, pp. 43-58.

Bisset R, 1980. Methods for Environmental Impact Analysis : Recent Trends and Future Prospects. Journal of Environmental Management, Vol. 11, 1980, pp. 27-43.

Blisset M, 1976. Environmental Assessment in Perspective; in Environmental Impact Assessment, Blisset M (ed.). Engineering Foundation, Austin, Texas, pp. 263-273.

Canter LW, 1977. Environmental Impact Assessment. McGraw-Hill Book Company, New York, McGraw-Hill Series in Water Resources and Environmental Engineering, ISBN 0-07-009764-X, xv+331p.

Carter FW, 1969. An Analysis of Medieval Serbian Oecumene : A Theoretical Approach. Geografiska Annaler, Vol. 51, Ser. B, 1969, pp. 39-55.

Catlow J and Thirlwall CG, 1975. Environmental Impact Analysis; in Symposium on Environmental Evaluation, Planning and Transport Advisory Council (ed.). Department of the Environment, London, pp. 128-134.

Catlow J and Thirlwall CG, 1976. Environmental Impact Analysis. Department of the Environment, London, Research Report 11, vii+116p.

Chase GH, 1976. Matrix Techniques in the Evaluation of Environmental Impacts; in Environmental Impact Assessment, Blisset M (ed.). Engineering Foundation, Austin, Texas, pp. 131-152.

Clark BD, Bisset R, and Wathern P, 1980. Environmental Impact Assessment : A Bibliography with Abstracts. Monsell Publishing, London, ISBN 0-7201-0899-3, iv+516p.

Clark BD, Chapman K, Bisset R, and Wathern P, 1976. Assessment of Major Industrial Applications : A Manual. Department of the Environment, London, Department of Energy research report No. 13.

Clark BD, Chapman K, Bisset R, and Wathern P, 1978a. Environmental Impact Assessment in the U.S.A. : A Critical Review. Department of the Environment, London, Research Report 26, xii+74p.

Clark BD, Chapman K, Bisset R, and Wathern P, 1978b. Methods of Environmental Impact Analysis. Built Environment, Vol. 4, No. 2, 1978, pp. 111-121.

Coates JF, 1976. Some Methods and Techniques for Comprehensive Impact Assessment; in Environmental Impact Assessment, Blisset M (ed.). Engineering Foundation, Austin, Texas, pp. 103-127.

Cook PL, 1977. Environmental Impact Analysis in the United States. U.S. Environmental Protection Agency, Washington, Paper presented in Mexico, October, 1977, v+24p.

Council on Environmental Quality, 1978. Environmental Quality : The Ninth Annual Report of the Council on Environmental Quality. United States Government Printing Office, Washington, xli+599p.

Dee N, Baker J, Drobny N, Duke K, Whitman I, and Fahringer D, 1973. An Environmental Evaluation System for Water Resource Planning. Water Resources Research, Vol. 9, No. 3, June 1973, pp. 523-535.

Department of Water Affairs, Forestry and Environmental Conservation, 1980. White Paper on a National Policy Regarding Environmental Conservation. Government Printer, Pretoria, W.P. O-1980, ISBN 0-621-06265-0, 13p.

Dreyfus SE, 1969. An Appraisal of Some Shortest-path Algorithms. Operations Research, Vol. 17, No. 3, May 1969, pp. 395-412.

Erickson PA, 1979. Environmental Impact Assessment : Principles and Applications. Academic Press, New York, ISBN 0-12-241550-7, xviii+395p.

Fischer DW and Davies GS, 1973. An Approach to Assessing Environmental Impacts. Journal of Environmental Management, Vol. 1, No. 3, 1973, pp. 207-227.

FitzPatrick M, Willson J, Ericson D, Fax G, and Wood D, 1978. Manual for Evaluating Secondary Impacts of Wastewater Treatment Facilities. U.S. Environmental Protection Agency, Washington D.C., Special Report EPA-600/5-78-003, vii+175p.

Fuggle RF, 1979. Methods for Preliminary Analysis of Environmental Impact in South AFrica. School of Environmental Studies, University of Cape Town, Cape Town, ISBN 0-7992-0323-8, xii+45p.

Fuggle RF, 1980. Regulating Environmental Change in South Africa. The Civil Engineer in South Africa (Special Issue - Shaping our Environment, Papers presented at the Symposium held in August, 1979), April 1980, pp. 73-83.

Furniss PR, 1977. Description and Manual for the Use of DRIVER : an Interactive Modelling Aid. Council for Scientific and Industrial Research, Pretoria, South African National Scientific Programmes Report No. 17, ISBN 0-7988-1197-8, vi+23p.

Gilliland MW and Risser PG, 1977. The Use of Systems Diagrams for Environmental Impact Assessment : Procedures and an Application. Ecological Modelling, Vol. 3, 1977, pp. 183-209.

Gold RL, 1978. Linking Social with Other Impact Assessments; in Environmental Impact Analysis : Emerging Issues in Planning, Hutchings BL and Jain RK (eds). University of Illinois Press, Urbana, (Proceedings of Conference on Environmental Impact Assessment, Monticello, Illinois, 1977), ISBN 0-252-00696-8, pp. 105-116.

Golden J, Ouellette RP, Saari S, and Cheremisinoff PN, 1979. Environmental Impact Data Book. Ann Arbor Science, Michigan, ISBN 0-250-40212-2, xxi+864p.

Guldberg PH, Benesh FH, and McCurdy T, 1977. Secondary Impacts of Major Land Use Projects. Journal of the American Institute of Planners, Vol. 43, No. 3, July 1977, pp. 260-270.

Haggett P and Chorley RJ, 1969. Network Analysis in Geography. Edward Arnold, London, ISBN 0-7131-5459-4, xii+348p.

Haggett P, Cliff AD, and Frey A, 1977. Locational Analysis in Human Geography. Edward Arnold, London, ISBN 0-7131-5899-9, xiv+605p.

Hart HT and Cullen P, 1976. Principles of Environmental Impact Assessment. Search, Vol. 7, No. 6, June 1976, pp. 231-235.

Health Sciences Computing Facility, 1977. BMDP-77 : Biomedical Computer Programs : P-Series. University of California Press, Berkeley, ISBN 520-03569-0, xiii+880p, (supported by NIH Special Resources Grant RR-3).

Heap BR, 1972. The Production of Graphs by Computer; in Graph Theory and Computing, Read RC (ed.). Academic Press, New York, ISBN 0-12-583850-6, pp. 47-62.

Hite JC and Laurent EA, 1971. Empirical Study of Economic Ecologic Linkages in a Coastal Area. Water Resources Research, Vol. 7, No. 5, October 1971, pp. 1070-1078.

Hjalte K, Lidgren K, and Stahl I, 1977. Environmental Policy and Welfare Economics. Cambridge University Press, London, vii+111p.

Hollick M, 1981a. The Role of Quantitative Decision-making Methods in Environmental Impact Assessment. Journal of Environmental Management, Vol. 12, 1981, pp. 65-78.

Hollick M, 1981b. Environmental Impact Assessment as a Planning Tool. Journal of Environmental Management, Vol. 12, 1981, pp. 79-90.

Holling CS (ed.), 1978. Adaptive Environmental Assessment and Management. John Wiley & Sons, Chichester. Wiley IIASA international series on applied systems, ISBN 0-471-99632-7, xix+377p.

Inhaber H, 1976. Environmental Indices. John Wiley & Sons, New York, ISBN 0-471-42796-9, xiv+178p.

Isard W, Choguill CL, Kissin J, Seyfarth RH, and Tatlock R, 1972. Ecologic-Economic Analysis for Regional Development. The Free Press, New York, xvii+270p.

Jain RK and Urban LV, 1975. A Review and Analysis of Environmental Impact Assessment Methodologies. U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois, Research Report E-69, 22p.

Jain RK, Urban LV, and Stacey GS, 1977. Environmental Impact Analysis : A New Dimension in Decision Making. Van Nostrand Reinhold Co., New York, ISBN 0-442-28807-7, xii+330p.

Jain RK and Webster RD, 1977. Computer-aided Environmental Impact Analysis. Journal of Water Resources - Planning and Management Division, November, 1977, pp. 257-271.

Jameson DL (ed.), 1976. Ecosystem Impacts of Urbanization : Assessment Methodology. U.S. Environmental Protection Agency, Corvallis, Oregon, Ecological research series EPA-600/3-76-072, xvii+228p.

Jones RL, 1975. Environmental Impact Reports : Problems and Opportunities. Civil Engineering, American Society of Civil Engineers, Vol. 45, No. 5, May 1975, pp. 71-73.

Kane J, Vertinsky I, and Thomson W, 1973. KSIM : A Methodology for Interactive Resource Policy Simulation. Water Resources Research, Vol. 9, No. 1, February 1973, pp. 65-79.

Kansky KJ, 1963. Structure of Transportation Networks : Relationships Between Network Geometry and Regional Characteristics. Department of Geography, The University of Chicago, Illinois, Department of Geography Research Paper No. 84, x+155p.

Katz L, 1953. A New Status Index Derived from Sociometric Analysis. Psychometrika, Vol. 18, No. 1, March 1953, pp. 39-43.

Kiely-Brocato KA, Buhyoff GJ, and Leuscher WA, 1980. An Attitude Scaling System with Relevance for Resource Management. Journal of Environmental Management, Vol. 10, 1980, pp. 71-81.

Lands Directorate, Environment Canada, 1974. An Environmental Assessment of Nanaimo Port Alternatives - Summary Report. Lands Directorate, Environment Canada, xv+63p.

Lee TR and Fenwick PD, 1973. The Environmental Matrix : Input-output Techniques Applied to Pollution Problems in Ontario. Water Resources Bulletin, American Water Resources

Association, Vol. 9, No. 1, February 1973, pp. 25-33.

Leontief W, 1970. Environmental Repercussions and the Economic Structure : An Input-output Approach. The Review of Economics and Statistics, Vol. 52, August 1970, pp. 262-271.

Leopold LB, Clarke FE, Hanshaw BB, and Balsley JR, 1971. A Procedure for Evaluating Environmental Impact. United States Geological Survey, Washington D.C., Geological Survey Circular No. 645.

Manning RE and Moncrief LW, 1979. Land Use Analysis Through Matrix Modeling : Theory and Application. Journal of Environmental Management, Vol. 9, 1979, pp. 33-40.

Matthews WH, 1975. Objective and Subjective Judgements in Environmental Impact Analysis. Environmental Conservation, Vol. 2, No. 2, 1975, pp. 121-131.

McHarg I, 1971. Design with Nature. Natural History Press, New York, x+198p.

Munn RE (ed.), 1975. Environmental Impact Assessment : Principles and Procedures. The Scientific Committee on Problems of the Environment, Toronto, Scope Report No. 5, 160p.

Nysteun JD and Dacey MF, 1961. A Graph Theory Interpretation of Nodal Regions. Papers and Proceedings of the Regional Science Association, Vol. 7, 1961, pp. 29-42.

Odum HT, 1971. Environment, Power, and Society. Wiley-Interscience, New York, ISBN 0-471-65275-X, ix+331p.

Peterson GL, Gemmell RS, and Schofer JL, 1974. Assessment of Environmental Impacts : Multidisciplinary Judgements of Large-scale Projects. Ekistics, Vol. 218, January 1974, pp. 23-30.

Pitts FR, 1965. A Graph Theoretic Approach to Historical Geography. The Professional Geographer, Vol. XVII, No. 5, September 1965, pp. 15-20.

Read RC, 1979. Algorithms in Graph Theory; in Applications of Graph Theory, Wilson RJ and Beineke LW (eds). Academic Press, London, ISBN 0-12-757840-4, pp. 381-417.

Reed WE, 1970. Indirect Connectivity and Hierarchies of Urban Dominance. Annals of the Association of American Geographers, Vol. 60, 1970, pp. 770-785.

Reichle DE, 1975. Advances in Ecosystem Analysis. BioScience, Vol. 25, No. 4, April 1975, pp. 257-264.

Ross JH, 1974. Quantification Aids to Environmental Impact Assessment. Lands Directorate, Environment Canada, Ottawa, Occasional Paper No. 3.

Ross JH, 1976. The Numeric Weighting of Environmental Interactions. Lands Directorate, Environment Canada, Occasional Paper No. 10, viii+39p.

Runyan D, 1977. Tools for Community-managed Impact Assessment. Journal of the American Institute of Planners, Vol. 43, No. 3, April 1977, pp. 125-135.

Shimbel A, 1951. Applications of Matrix Algebra to Communication Nets. Bulletin of Mathematical Biophysics, Vol. 13, 1951, pp. 165-178.

Shimbel A, 1953. Structural Parameters of Communication Networks. Bulletin of Mathematical Biophysics, Vol. 15, 1953, pp. 501-507.

Siegel S, 1956. Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill Kogakusha, Ltd., Tokyo, International Student Edition, xvii+312p.

Skutsch M and Flowerdew R, 1976. Measurement Techniques in Environmental Impact Assessment. Environmental Conservation, Vol. 3, No. 3, Autumn 1976, pp. 209-217.

Sondheim MW, 1978. A Comprehensive Methodology for Assessing Environmental Impact. Journal of Environmental Management, Vol. 6, 1978, pp. 27-42.

Sorensen J, 1971. A Framework for Identification and Control of Resource Degradation and Conflict in the Multiple Use of the Coastal Zone. Department of Architecture, University of California, Berkeley.

Taaffe EJ and Gauthier HL, 1973. Geography of Transportation. Prentice-Hall Incorporated, London, ISBN 0-13-351395-5, xiv+226p.

Tarjan RE, 1974. A Note on Finding the Bridges of a Graph. Information Processing Letters, Vol. 2, No. 6, April 1974, pp. 160-161.

U.S.A. Congress, 1970. The National Environmental Policy Act of 1969. PL 91-190, 91st Congress, s.1075, 1st January, 1970.

Urban LV, Balbach HE, Jain RK, Novak EW, and Riggins RE, 1975. Computer-aided Environmental Impact Analysis for Construction Activities : User Manual. Construction Engineering Research Laboratory, Champaign, Illinois, Technical Report E-50, March 1975, 71p.

Ward DV, 1978. Biological Environmental Impact Studies - Theory and Methods. Academic Press, New York, viii+157p.

Warner ML and Preston EH, 1974. A Review of Environmental Impact Assessment Methodologies. U.S. Environmental Protection Agency, Washington D.C., EPA 600/5-74-002, vi+25p.

Welch HW and Lewis GD, 1976. Assessing Environmental Impacts of Multiple Use Land Management. Journal of Environmental Management, Vol. 4, No. 3, 1976, pp. 197-209.

Wilson RJ and Beineke LW (eds), 1979. Applications of Graph Theory. Academic Press, London, ISBN 0-12-757840-4, xii+423p.

8. APPENDIX A - COMPONENT INTERACTION PROGRAMME / MANUAL

8.1. PROGRAMME MANUAL

8.1.1. Introduction

The component interaction programme has been written to perform component interaction analyses of environmental systems modelled as a list of components and an array of zeros and ones. The programme performs all the procedures mentioned in chapter three of this thesis. Chapter three should be read as a preface to this manual as many of the terms and algorithms used are described there.

The programme is written in ASCII FORTRAN (Sperry Univac 1100 Series, level 9R1) for the University of Cape Town Univac 1100/81 time-sharing computer. It is an interactive programme intended for VDU terminals (a cathode ray tube screen and keyboard) or teletype terminals. Data files are created and accessed within the programme (these functions are transparent to the user) by interfacing with the EXEC8 control language.

In its most efficiently segmented form, the absolute programme element executes by occupying under 25K of core space. A full analysis of the largest possible component interaction matrix was found to take two minutes of central processing time on average.

Section two of this appendix lists the programme and related runstreams. Some of the interactive commands from the programme are presented in appendix B. From these it may be seen that if a user understands the terms related to the component interaction technique, few additional instructions are required to perform the computer analysis.

8.1.2. Programme Structure

The programme has four main parts related for four specific functions:

- the data input routines
- data editing routines
- component interaction analysis routines
- data output routines

While all input/output routines are controlled by the programme only, other functions are controlled by user responses to programme prompts. User replies are checked for format and content errors (as far as is possible) to avoid runstream termination. A recognized error solicits a standard error message, and corrective options are offered in most cases.

Data Input

Data is entered as a set of structured interactive responses, or by accessing a specified data file from within the programme. All file manipulations are performed by programme calls to the system function "FASCF2" (which avoids runstream termination if errors are detected in the specified file names). A component interaction programme data file stores information on the size of the component matrix, whether it is weighted or unweighted, the component labels (names), and the array of matrix elements (which records the presence or absence of direct dependencies between the listed components). The programme is capable of processing component interaction matrices which are dimensioned 50 x 50 or less.

Data Editing

The edit mode can be selected at any stage after the data has been entered to the programme. Once in edit mode (i.e. control is with the "editor" subroutine), various editing options are offered. Matrix elements and component labels can be altered, or whole components (i.e a row and a column of the interaction matrix) can be added or deleted. Once the data has been edited, and if a data file is being used to store the matrix information, the new version of the matrix can be stored in the old data file, a new file, or given no permanent storage. All edited versions of the component interaction matrix are printed to the hardcopy printfile.

Component Interaction Analysis

At any stage after the data has been entered, any of the component interaction procedures can be requested by the user. There is one restriction enforced by the programme. The minimum link matrix must first be derived before other measures or tracks are performed. This is because the powering process in the minimum link subroutine ("minlnk") provides much of the information used in the other procedures.

The following information can be obtained:

- the initial matrix
- edited matrices
- minimum link matrices
- valency rankings
- average graph distance rankings
- minimum path accessibility measure rankings
- critical-position summary and rankings
- cut-position summary and ranking
- disruptive measure rankings
- dependency tracks

The matrix data can be edited and analysed repetitively.

Data Output

None of the analysis results are printed to the active terminal. All output is written to an alternative printfile (controlled entirely by the programme), which is "sym'ed" to the system printer at the end of a run. The user is given instructions on how to retrieve the hardcopy, and the programme allows a printout heading to be specified for each version of a matrix. Dependency tracks performed by the programme are plotted by the system plotter. Plotting information is written to two files (unit 2 /plotinfo/; and unit 3 /plotdata/). At the end of a run, if tracks have been performed, a call to "FASCF2" starts a plotting programme which uses the data in the plotfiles to create the tracks. The user is instructed on how to retrieve the tracks.

8.1.3. Technical Features

Most of the comments which follow are pertinent to the programme listing.

The programme has been fully documented, and in addition, the programme commands do much to make the programme "readable". The subroutines, arrays, and variables have been named to give some compliance with the terms introduced in chapter three, or to give an indication of their function. All two dimensional arrays begin with "mx-", and character variables and arrays end with "$".

Files

The files used in the programme are the following:

- * unit 1 - data file
- * unit 2 - alternate printfile
- * unit 3 - plotting information file (headings etc.)
- * unit 4 - plotting data file (actual track information)
- * unit 13 - direct access file for track information (S in chapter three)

A call to the system function "ADATE$" is used to obtain the date and time of the run, and this information is used to specify a unique name for the alternate printfile. The date and time also appear on all printouts.

Runstreams

Runstreams associated with the programme are listed after the programme listings. These are for accessing and executing the programme, and for initiating the plotting of tracks. Note that the plotting is initiated by a runstream which is "started" within the run, but performed by a separate plotting programme ("abstrack").

Plotting

The plotting programme is written in the "CALCOMP" plotting language, and the "Graphics Display Package" of the University of Cape Town is used to interface with that system's plotting hardware. The plotting is completely transparent to the user.

8.1.4. Improvements and Extentions

Experience with the programme has shown that the following improvements would be beneficial to the component interaction programme:

* The input of data is a long process for large component interaction matrices. The programme could offer a means of suspending and resuming the process to allow the user to terminate the programme without losing the data already entered.

* The input and editing of data is efficient as an interactive process, but computing the component interaction measures is time consuming in 'demand' mode. A run 'started' from the programme to perform the analysis of data in 'batch' mode would be preferable.

* The programme can access data files, but not the direct access

files on which the track information is stored (unit 13). If both these files could be accessed, it would not be necessary to perform the minimum link procedure every time a component interaction matrix is analysed in different programme runs.

* Many of the algorithms used to extract the component interaction analysis data are taken directly from chapter three. No effort has been made to optimize the procedures used for powering matrices, and for performing the other mathematical functions included in the technique. Further, the method used for writing and reading to files can be improved upon by more advanced input/output routines available within ASCII FORTRAN.

8.1.5. Reference Manuals

The following computer language manuals were used:

Exec8 Hardware/Software Summary. Sperry Univac 1100 Series, UP-7824.

FORTRAN (ASCII) Level 9R1 - Programmer Reference. Sperry Univac 1100 Series, UP-8244.1-A, June 1979.

Programming Calcomp Pen Plotters (Exec II and Exec VIII Offline). California Computer Products Inc., September 1969.

Graphics Display Package (GDP). University of Cape Town, October 1978.

8.1. COMPONENT INTERACTION PROGRAMME

The following pages list of the component interaction and associated plotting programmes developed and used for this report. The runstreams associated with each programme are also given. The programmes are stored on a file at the University of Cape Town Computer Centre.

COMPONENT INTERACTION PROGRAMME

```
***********************************************************************
*   THE MAIN SEGMENT OF THE COMPONENT INTERACTION PROGRAMME
*
*
*   THIS PROGRAMME PERFORMS THE PROCEDURES OF THE COMPONENT INTERACTION
*   TECHNIQUE
*
*
*   ARRAYS
*   THE PROGRAMME IS BASED ON A MATRIX MODEL OF AN ENVIRONMENTAL SYSTEM
*   ARRAYS USED IN THIS PROGRAMME TO CONTAIN AND MANIPULATE THE COMPONENT
*   INTERACTION DATA (MATRIX MODEL) ARE ALL BOUNDED AT 50 X 50.
*
*   MXCMPT - INTEGER ARRAY TO HOLD THE COMPONENT INTERACTION MATRIX DATA
*   MXLINK - INTEGER ARRAY FOR THE MINIMUM LINK MATRIX DATA
*   MXMINA AND MXMIND - INTEGER ARRAYS FOR THE MINIMUM PATH ACCESSIBILITY
*                       MEASURES DERIVED IN THE ROUTINE "MINLNK" AND PROCESSED
*                       IN "MNPATH".
*   FTYPE - INTEGER INDICATOR OF THE MODE OF DATA INPUT CHOSEN BY THE USER
*           (1 - INITIALIZE DATA FILE; 2 - ACCESS DATA FILE; 3 - NO DATA FILE)
*   MSGN - NUMBER OF MESSAGE LINES IN 'MSG$' TO BE PRINTED BY 'MSGPRT'.
*   SKIP - INTEGER (0 OR 1) INDICATOR TO ROUTINE 'MSGPRT' TO SIGNAL FOR
*           THE USER PROMPT
*   REPLY - INTEGER VARIABLE FOR USER REPLIES TO PROGRAMME OPTIONS
*   STATUS - INTEGER VARIABLE SET BY THE FUNCTION FACSF2 TO REFLECT THE
*            STATUS RETURNED BY THE FUNCTION
*   TTLN - AN INTEGER INDICATOR OF WHICH FORM OF MATRIX IS BEING PRINTED
*           (1=INITIAL; 2=EDITED; 3=MINIMUM LINK; 4=LAST COPY)
*   HDGEND - INTEGER COUNTER OF THE NUMBER OF CHARACTERS IN HDG$
*   FNEND - INTEGER VARIABLE OF THE NUMBER OF CHARACTERS IN THE USER.
*           SUPPLIED DATA FILE NAME.  NEEDED FOR A CALL TO 'FACSF2'
*   FACSF2 - UNIVAC SYSTEM INTEGER FUNCTION TO EXECUTE EXECB COMMANDS
*            FROM WITHIN THE PROGRAMME (SEE SPERRY UNIVAC PROGRAMMER
*            REFERENCE MANUAL UP-8244.1, ASCII FORTRAN LEVEL 9R1, P7-31)
*   FASCF - UNIVAC SYSTEM INTEGER FUNCTION SIMILAR TO FASCF2 EXCEPT THAT
*           ERROR CONDITIONS ARE REGISTERED IN THE PROGRAMME
*   NTRACK - INTEGER COUNTER OF THE NUMBER OF TRACKS REQUESTED BY
*            THE USER THROUGH THE ROUTINE 'TRACKS'
*   NFILER - INTEGER INDICATOR TO CONTROL ACCESS TO RECORDS IN THE DIRECT
*            ACCESS FILE 13
*   ASGNED - INTEGER VARIABLE USED TO CHECK THE STATUS RETURNED BY FACSF2
*            FOR AN EXEC REQUEST TO ASSIGN A FILE.  SET TO THE EQUIVALENT
*            OCTAL VALUE INDICATING FILE ALREADY ASSIGNED TO PROGRAMME
*   IMIN - THE MINIMUM VALUE (INTEGER) OF A COMPONENT MATRIX ELEMENT
*   IMAX - THE MAXIMUM VALUE (INTEGER) OF A COMPONENT MATRIX ELEMENT
*   MSG$ - CHARACTER ARRAY OF THE PROGRAMME MESSAGES FOR PRINTING
*          TO THE TERMINAL SCREEN BY THE ROUTINE 'MSGPRT'
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   REPLY$ - CHARACTER VARIABLE FOR USER REPLIES TO PROGRAMME QUESTIONS
*   FNAME$ - CHARACTER VARIABLE OF THE USER SUPPLIED DATA FILE (UNIT 1) NAME
*   HDG$ - CHARACTER VARIABLE OF A USER SUPPLIED HEADING FOR THE
*          HARDCOPY PRINTOUT OF THE PROGRAMME OUTPUT
*   DATE$ - CHARACTER VARIABLE FOR THE DATE OBTAINED BY A CALL TO SYSTEM
*           FUNCTION 'ADATE$' USED FOR HEADING IN THE PLOTTING PROGRAMME
*   TIME$ - CHARACTER VARIABLE OF THE UNIVAC SUPPLIED TIME (CALLED
*           BY 'ADATE$' IN MAIN PROGRAMME), USED TO SPECIFY A UNIQUE
*           FILE FOR THE ALTERNATE PRINTFILE - UNIT 2
*   HEAD$ - CHARACTER VARIABLE OF A PROGRAMME SUPPLIED HEADING FOR
*           HARDCOPY PRINTOUT FROM 'PRTPRT'
*   ERMSG$ - CHARACTER VARIABLE WITH STANDARD ERROR MESSAGE
*
*   FILES
*   UNIT 1 - PERMANENT DATA FILE CONTAINING THE COMPONENT MATRIX DATA
*   UNIT 2 - ALTERNATE PRINTFILE
*   UNIT 3 - DATA FILE USED FOR THE PLOTTING INFORMATION CREATED
*            BY 'TRACKS'
*   UNIT 4 - SECOND DATA FILE FOR PLOTTING INFORMATION CREATED BY
*            'TRACKS'
*
*   UNIT 13 - DIRECT ACCESS FILE CREATED IN THE MAIN PROGRAMME TO STORE
*             INFORMATION USED IN THE 'TRACK' SUBROUTINE.
*
*

      INTEGER MXCMPT(50,50), MXLINK(50,50), MXMINA(50), MXMIND(50),
     +        FTYPE, MSGN, SKIP, REPLY, STATUS, TTLN, HDGEND, FNEND,
     +        FACSF2, FACSF, NTRACK, NFILER, ASGNED/0100000000000/,
     +        IMIN,IMAX
      CHARACTER MSG$*67(6), MTLBL$*20(50),
     +          REPLY$*3, FNAME$*12, HDG$*55, DATE$*8, TIME$*8,
     +          HEAD$*85, ERMSG$*50
*
*   THE SUBROUTINES IN THE PROGRAMME,IN THE ORDER IN WHICH THEY APPEAR
*   IN THE PROGRAMME LISTING
*
      EXTERNAL MSGPRT,
     +         PRTPRT,
     +         DIMINP,
     +         DATINP,
     +         MXPART,
     +         MXFRQ,
     +         EDITOR,
     +         EDELMT,
     +         EDLBL$,
     +         EDSDTP,
     +         EDADTN,
     +         MINLNK,
     +         TRACKS,
     +         VALNCY,
     +         DSTANS,
     +         MNPATH,
     +         CRITCL,
     +         CUTTER,
     +         DISRPT,
     +         RUNFIN
*   THE PROGRAMME CAN BE SEGMENTED TO SAVE CORE SPACE ON EXECUTION.
*   THE FIRST FIVE SUBROUTINES FORM THE MAIN PROGRAMME.  THE EDITING
*   ROUTINES FORM THE SECOND SEGMENT.  THE "MINLNK" ROUTINE IS THE
*   THIRD SEGEMENT.  ALL THE OTHER ROUTINES FORM THE FOURTH SEGMENT.
*   THE PROGRAMME IS BEST SEGMENTED.  THE FISRT SIX SUBROUTINES FORM
*   THE MAIN SEGMENT, THE EDITING ROUTINES ANOTHER, "MINLNK" ON ITS
*   OWN, AND THE COMPONENT INTERACTION PROCEDURES AND "RUNFIN" THE LAST
*


*
*   INITIALIZING THE STANDARD ERROR MESSAGE AND DEFINING FILE 13 AS THE
*   DIRECT ACCESS FILE FOR TRACK INFORMATION
*
      ERMSG$ = '**** RE-ENTER THE LAST RESPONSE.  IT WAS FAULTY. '
      DEFINE FILE 13(2500,31,U,NFILER)
*
*   THE DATE AND TIME ARE STORED IN DATE$, AND TIME$ FOR HEADINGS AND
*   SPECIFYING A UNIQUE ALTERNATE PRINTFILE (UNIT 2)
*   LABEL 90 IS THE FIRST OPERABLE STATEMENT IN THE PROGRAMME

   90 CALL ADATE(DATE$,TIME$)
      NTRACK = 0
*
*   A CHECK IS MADE TO ENSURE THAT NO PREVIOUS PLOT FILES STILL NEED
*   TO BE PROCESSED.  TRACKS CANNOT BE PERFORMED UNTIL THE PLOT DATA
*   THE FILE "CHECKER" IS A DUMMY FILE WHICH ONLY GETS DELETED WHEN THE
*   TRACK PROGRAMME IS COMPLETE
*
      STATUS = FACSF2('@ASG,AX CHECKER. . ')
      STATUS = FACSF2('@FREE CHECKER. . ')
      IF(STATUS.EQ.0) THEN
        MSG$(1)= 'YOUR LAST SET OF TRACK PLOTS IS STILL BEING *'
        MSG$(2)= 'PROCESSED.  YOU MAY NOT BE ABLE TO PERFORM *'
        MSG$(3)= 'FURTHER TRACKS DURING THIS RUN.            *'
        MSGN = 3
        SKIP = 1
        CALL MSGPRT(MSG$,MSGN,SKIP)
      END IF
      TTLN = 1
*
*   ASSIGNING THE ALTERNATE PRINTFILE (UNIT 2)
*
      STATUS = FACSF2('@ASG,UP PF'//TIME$//'. . ')
      STATUS = FACSF2('@USE 2,PF'//TIME$//'. . ')
      DEFINE FILE 2 (APRNTA,,132)


      MSG$(1)= 'ALL DATA AND COMPUTED INFORMATION ASSOCIATED *'
      MSG$(2)= 'WITH THIS RUN WILL BE AVAILABLE TO YOU AS A  *'
      MSG$(3)= 'HARDCOPY PRINT-OUT LISTED UNDER YOUR USERID. *'
      MSG$(4)= 'ENTER A DESCRIPTIVE HEADING OF UP TO FIFTY   *'
      MSG$(5)= 'CHARACTERS FOR THIS PRINT-OUT.               *'
      MSGN = 5
  100 CALL MSGPRT(MSG$,MSGN,SKIP)
```

```
      READ(*, '(A55)', ERR = 120) HDG$
      HDGEND = 55
*
*  THE FOLLOWING PROCEDURE IS USED THROUGHOUT THE PROGRAMME TO CHECK
*  THE LENGTH OF USER REPLIES TO HEADINGS AND FILE NAMES.
*
  110 IF((HDGEND.GT.0).AND.(HDG$(HDGEND:HDGEND).EQ.' ')) THEN
        HDGEND = HDGEND - 1
        GO TO 110
      END IF
      IF(HDGEND.GT.50) THEN
  120   PRINT*,ERMSG7
        GO TO 100
      END IF


  130 MSG$(1)= 'IN ORDER TO ENTER THE COMPONENT MATRIX DATA *'
      MSG$(2)= 'TYPE "1"  TO ASSIGN A PERMANENT DATA FILE    *'
      MSG$(3)= 'TYPE "2"  TO ACCESS A PERMANENT DATA FILE    *'
      MSG$(4)= 'TYPE "3"  FOR NO PERMANENT DATA RECORD       *'
      MSGN = 4
  140 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ (*, *, ERR = 270) FTYPE

*
*  INITITALIZING A DATA FILE FOR THE COMPONENT INTERACTION MATRIX DATA
*
      IF(FTYPE.EQ.1)THEN
  150   MSG$(1)= 'ENTER A NAME (AS ANY COMDINATION OF 1 TO 12 *'
        MSG$(2)= 'ALPHANUMERIC CHARACTERS) FOR THE DATA FILE. *'
        MSGN = 2
  160   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, '(A12)', ERR = 160) FNAME$
        FNEND = 12
  170   IF((FNEND.GT.0).AND.(FNAME$(FNEND:FNEND).EQ.' ')) THEN
          FNEND = FNEND - 1
          GO TO 170
        END IF
        IF(FNEND.EQ.0) GO TO 160
*
*  ASSIGNING THE FILE SPECIFIED BY THE USER.  IF STATUS REFLECTS AN
*  AN ERROR CONDITION, THE USER IS GIVEN CORRECTIVE OPTIONS
*
        STATUS = FACSF2('@ASG,UP '//FNAME$(1:FNEND)//'.,F2 . ')
        REPLY = 0
        IF(STATUS.NE.0) THEN
          MSG$(1)= 'A FILE OF THIS NAME ALREADY EXISTS UNDER      *'
          MSG$(2)= 'YOUR PROJECT-ID.                              *'
          MSG$(3)= 'TYPE "1" TO ENTER ANDTHER NAME FOR THE FILE *'
          MSG$(4)= 'TYPE "2" TO OVER-WRITE THIS EXISTING FILE     *'
          MSG$(5)= 'TYPE "3" TO USE ANOTHER WAY TO ENTER DATA     *'
          MSG$(6)= 'TYPE "4" TO TERMINATE THE PROGRAMME           *'
          MSGN = 6
  180     CALL MSGPRT(MSG$,MSGN,SKIP)
          READ(*, *, ERR = 190) REPLY
          IF(REPLY.EQ.1) THEN
            GO TO 150
          ELSE IF(REPLY.EQ.2) THEN
            IF(STATUS.NE.ASGNED) THEN
              STATUS = FACSF2('@ASG,A '//FNAME$(1:FNEND)//'. . ')
            END IF
            GO TO 200
          ELSE IF(REPLY.EQ.3) THEN
            GO TO 130
          ELSE IF(REPLY.EQ.4) THEN
*
*  IF THE USER ABORTS THE PROGRAMME THIS CONDITION IS RECORDED ON THE
*  ALTERNATE PRINTFILE WHICH WILL BE RELEASED AND SYM'ED BY "RUNFIN"
*
            HEAD$ = ' * RUN ABORTED - ATTEMPTED TO ASSIGN FILE '//FNAME$
     *      (1:FNEND)//' WHICH ALREADY EXISTS *'
            WRITE(2, '(1H1,//,2X,A,////////,15X,A)') HDG$,HEAD$
            STATUS = ('@ASG,T 1. . ')
*
*  RUNFIN IS CALLED TO TERMINATE THE PROGRAMME
*
            CALL RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
     *      NTRACK,ERMSG$,HDG$,TIME$)
            GO TO 350
          ELSE
  190       PRINT*,ERMSG1
            GO TO 180
          END IF
        END IF
*
*  ONCE THE FILE IS ASSIGNED IT IS DEFINED AS FILE 1.  DATA IS ENTERED
*  BY CALLS TO "DIMINP" AND "DATINP".  IF THE DATA IS ENTERED CORRECTLY
*  THE COMPONENT INTERACTION DATA IS WRITTEN TO THE FILE 1 BY A CALL TO
*  "MXFWRT"
*
  200   STATUS = FACSF2('@USE 1,'//FNAME$(1:FNEND)//'. . ')
        CALL DIMINP(MSG$,MTDMSN,ERMSG$)
        IF(MSG$(6)(1:5).EQ.'ABORT') THEN
          HEAD$ = ' * RUN ABORTED - ATTEMPTED TO DEFINE A MATRIX '//
     *    'GREATER THAN 50 X 50 * '
          WRITE(2, '(1H1,//,2X,A,////////,15X,A)') HDG$,HEAD$
          CALL RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
     *    NTRACK,ERMSG$,HDG$,TIME$)
          GO TO 350
        END IF
        CALL DATINP(MXCMPT,MTLBL$,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
        CALL MXFWRT(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
        TTLN = 1
        IF(REPLY.EQ.2) THEN
          HEAD$ = '      DATE : '//DATE$//'      TIME : '//TIME$//
     *    '       DATA FILE (OVER-WRITTEN) : '//FNAME$
        ELSE
          HEAD$ = '      DATE : '//DATE$//'      TIME : '//TIME$//
     *    '       DATA FILE (ASSIGNED) : '//FNAME$
        END IF
*
*  THE MATRIX DATA IS WRITTEN TO THE ALTERNATE PRINTFILE BY A CALL TO
*  "PRTWRT" WITH TTLN = 1
*
        CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
        MSG$(1)= 'A COPY OF THE COMPONENT INTERACTION MATRIX *'
        MSG$(2)= 'HAS BEEN RECORDED ON YOUR PRINTFILE.       *'
        SKIP = 1
        MSGN = 2
        CALL MSGPRT(MSG$,MSGN,SKIP)

*
*  ACCESSING AN ALREADY EXISTING DATA FILE
*
      ELSE IF(FTYPE.EQ.2) THEN
  220   MSG$(1)= 'ENTER THE NAME OF THE DATA FILE *'
        MSGN = 1
  230   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, '(A12)', ERR = 230) FNAME$
        FNEND = 12
  240   IF((FNEND.GT.0).AND.(FNAME$(FNEND:FNEND).EQ.' ')) THEN
          FNEND = FNEND - 1
          GO TO 240
        END IF
        IF(FNEND.EQ.0) GO TO 230
        STATUS = FACSF('@ASG,A '//FNAME$(1:FNEND)//' . ')
        IF((STATUS.NE.0).AND.(STATUS.NE.ASGNED)) THEN
*
*  IF THE FILE CANNOT BE ASSIGNED THE USER IS OFFERED CORRECTIVE OPTIONS
*
          MSG$(1)= 'THE ABOVE ERROR MESSAGE INDICATES WHY YOUR *'
          MSG$(2)= 'FILE CANNOT BE ASSIGNED TO THIS RUN.       *'
          MSG$(3)= ' TYPE "1" TO RE-ENTER A DATA FILE NAME     *'
          MSG$(4)= ' TYPE "2" TO USE ANOTHER WAY TO ENTER DATA *'
          MSG$(5)= ' TYPE "3" TO TERMINATE THE PROGRAMME       *'
          MSGN = 5
  250     CALL MSGPRT(MSG$,MSGN,SKIP)
          READ(*, *, ERR = 260) REPLY
          IF(REPLY.EQ.1) THEN
            GO TO 220
          ELSE IF(REPLY.EQ.2) THEN
            GO TO 130
          ELSE IF(REPLY.EQ.3) THEN
*
*  IF THE PROGRAMME IS ABORTED THIS CONDITION IS RECORDED ON THE PRINTFILE
*  AND THE TERMINATING FUNCTIONS OF "RUNFIN" ARE IMPLEMENTED
*
            HEAD$ = '* RUN ABORTED - DATA FILE '//FNAME$(1:FNEND)//
     *      ' COULD NOT BE ASSIGNED *'
            WRITE(2, '(1H1,//,2X,A,////////,15X,A)') HDG$,HEAD$
```

```
                STATUS = FACSF2('@ASG,T 1. . ')
                CALL RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
     *          NTRACK,ERMSG$,HDG$,TIME$)
                GO TO 350
             ELSE
  260           PRINT*,ERMSG$
                GO TO 250
             END IF
          END IF
*
*   IF THE FILE IS ACCESSED SUCCESSFULLY IT BECOMES UNIT 1, AND THE
*   COMPONENT MATRIX DATA IS READ INTO THE PROGRAMME BY A CALL TO "MXERD"
*
          STATUS = FACSF2('@USE 1,'//FNAMES$//' . ')
          READ (1, '(I2)') MTDMSN
          CALL MXERD(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
          TTLN = 1
          HEAD$ = '      DATE : '//DATES$//'      TIME : '//TIMES$//
     *    '      DATA FILE (ACCESSED) : '//FNAMES$
*
*   A CALL TO "PRTWRT" WRITES THE MATRIX DATA TO THE PRINTFILE
*
          CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
          MSG$(1)= 'THE COMPONENT MATRIX DATA HAS BEEN ENTERED FROM *'
          MSG$(2)= 'THE FILE INTO THE PROGRAMME. A COPY OF THE MAT- *'
          MSG$(3)= 'RIX HAS BEEN RECORDED ON YOUR PRINTFILE.        *'
          IF(IMAX.EQ.1) THEN
            MSG$(4)= 'THE INTERACTIONS ARE ENTERED AS ZEROS OR ONES.  *'
          ELSE IF(IMAX.EQ.3) THEN
            MSG$(4) ='THE INTERACTIONS ARE SCALED FROM ZERO TO THREE. *'
          END IF
          WRITE(REPLY$, '(I2)') MTDMSN
          MSG$(5)= 'THE COMPONENT MATRIX IS DIMENSIONED AT '//
     *    REPLY$(1:2)//' X '//REPLY$(1:2)//'. *'
          MSGN = 5
          SKIP = 1
          CALL MSGPRT(MSG$,MSGN,SKIP)

*
*   NO PERMANENT COPY OF THE MATRIX DATA REQUIRED.  THE DATA IS READ
*   INTO THE PROGRAMME BY CALLS TO DATINP AND DIMINP, AND THE DATA IS
*   IS WRITTEN TO THE PRINTFILE BY CALLING "PRTWRT"
*
        ELSE IF(FTYPE.EQ.3) THEN
          CALL DIMINP(MSG$,MTDMSN,ERMSG$)
          IF(MSG$(6)(1:5).EQ.'ABORT') THEN
            HEAD$ = ' * RUN ABORTED - ATTEMPTED TO DEFINE A MATRIX '//
     *      'GREATER THAN 50 X 50 * '
            WRITE(2, '(1H1,//,2X,A,////////,15X,A)') HDG$,HEAD$
            CALL RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
     *      NTRACK,ERMSG$,HDG$,TIME$)
            GO TO 350
          END IF
          CALL DATINP(MXCMPT,MTLBL$,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
          TTLN = 1
          HEAD$ = '      DATE : '//DATES$//'      TIME : '//TIMES$//
     *    '      * DATA ENTERED INTERACTIVELY *'
          CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
          MSG$(1)= 'A COPY OF THE COMPONENT INTERACTION MATRIX *'
          MSG$(2)= 'HAS BEEN RECORDED ON YOUR PRINTFILE.       *'
          SKIP = 1
          MSGN = 2
          CALL MSGPRT(MSG$,MSGN,SKIP)

        ELSE
  270     PRINT*,ERMSG$
          GO TO 140
        END IF

*
*   THE PROGRAMME OFFERS THE USER THE CHOICE OF CALLING THE VARIOUS
*   PROCEDURES.  "MINLNK" MUST BE CALLED BEFORE ANY OTHER PROCEDURE IS
*   CALLED.
*
        IF(MTDMSN.GE.20) THEN
          MSG$(1)= 'AS YOUR MATRIX IS REASONABLY LARGE, SOME OF    *'
          MSG$(2)= 'THE FOLLOWING PROCEDURES MAY TAKE SOME TIME.   *'
          MSGN = 2
          SKIP = 1
          CALL MSGPRT(MSG$,MSGN,SKIP)
        END IF
  280   MSG$(1)= 'TYPE "1" TO EDIT OR CORRECT THE MATRIX DATA     *'
        MSG$(2)= 'TYPE "2" TO COMPUTE THE MINIMUM LINK MATRIX     *'
        MSG$(3)= 'TYPE "3" TO DO SELECTED INTERACTION TRACES      *'
        MSG$(4)= 'TYPE "4" FOR MEASURES OF COMPONENT IMPORTANCE *'
        MSG$(5)= 'TYPE "5" TO TERMINATE THE PROGRAMME            *'
        MSGN = 5
  290   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, *, ERR = 340) REPLY
        IF((REPLY.GT.2).AND.(REPLY.LT.5).AND.(TTLN.NE.3)) THEN
          MSG$(1)= 'THE MINIMUM LINK MATRIX MUST BE COMPUTED BEFORE *'
          MSG$(2)= 'FURTHER INFORMATION CAN BE EXTRACTED FROM AN    *'
          MSG$(3)= 'INITIAL OR EDITED COMPONENT INTERACTION MATRIX. *'
          SKIP = 1
          MSGN = 3
          CALL MSGPRT(MSG$,MSGN,SKIP)
          GO TO 290
        END IF
        IF(REPLY.EQ.1) THEN
          CALL EDITOR(MXCMPT,MTLBL$,MSG$,MTDMSN,FTYPE,FNEND,IMIN,IMAX,
     *    TTLN,ERMSG$,HEAD$,HDG$,FNAMES$,DATES$,TIMES$)
*
*   IF THE PROGRAMME HAS TERMINATED IN THE EDIT MODE THE "RUNFIN"
*   ROUTINE IS CALLED TO TERMINATE THE PROGRAMME
*
          IF(HEAD$.EQ.'ABORT') THEN
            CALL RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
     *      NTRACK,ERMSG$,HDG$,TIMES$)
            GO TO 350
          ELSE
            GO TO 280
          END IF
        ELSE IF(REPLY.EQ.2) THEN
          CALL MINLNK(MXCMPT,MXLINK,MXMINA,MXMIND,MTDMSN,NFILER)
          TTLN = 3
          CALL PRTWRT(MXLINK,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
          MSG$(1)= 'A COPY OF THE MINIMUM LINK MATRIX HAS BEEN WRITTEN *'
          MSG$(2)= 'TO YOUR PRINTFILE.                                  *'
          MSGN = 2
          SKIP = 1
          CALL MSGPRT(MSG$,MSGN,SKIP)
          GO TO 280
        ELSE IF(REPLY.EQ.3) THEN
*
*   IF TRACKS ARE TO BE PERFORMED, TWO DATA FILES ARE CREATED (ONLY IF
*   THE THERE IS NO SET OF PLOTS WAITING TO BE PROCESSED) TO HOLD THE
*   PLOTTING INFORMATION FOR THE PLOT PROGRAMME WHICH IS INITIATED IN
*   "RUNFIN".
*   THE FILES ARE ONLY CREATED IF THIS IS THE FIRST CALL TO "TRACKS"
*   (I.E. NTRACK = 0)
*
          IF(NTRACK.EQ.0) THEN
*
*   THE DUMMY FILE 'CHECKER' IS CREATED BY THE PROGRAMME, AND IS USED TO
*   ENSURE THAT PREVIOUS TRACKS HAVE BEEN PROCESSED BEFORE NEW DATA IS
*   WRITTEN TO THE FILES 3 AND 4.  THE DUMMY FILE IS DELETED BY THE
*   'STARTED' RUNSTREAM WHICH CONTROLS THE PLOTTING AFTER THE PLOTTING
*   PROGRAMME HAS TERMINATED.
*
            STATUS = FACSF2('@ASG,UP CHECKER.,F2 . ')
            IF(STATUS.NE.0) THEN
              MSG$(1) ='NEW TRACKS CANNOT BE PERFORMED YET AS YOUR LAST *'
              MSG$(2) ='BATCH IS STILL BEING PROCESSED. TRY AGAIN LATER.*'
              MSGN = 2
              SKIP = 1
              CALL MSGPRT(MSG$,MSGN,SKIP)
              GO TO 280
            END IF
            STATUS = FACSF2('@ASG,UP PLOTINFO.,F2 . ')
            STATUS = FACSF2('@ASG,UP PLOTDATA.,F2 . ')
            STATUS = FACSF2('@USE 3,PLOTINFO. . ')
            STATUS = FACSF2('@USE 4,PLOTDATA. . ')
          END IF
          CALL TRACKS(MXLINK,MTLBL$,MTDMSN,NTRACK,NFILER,HDGEND,
     *    HDG$,DATES$,ERMSG$)
          GO TO 280
        ELSE IF(REPLY.EQ.4) THEN
          MSG$(1)= 'ALL RANKINGS WILL BE WRITTEN TO YOUR PRINTFILE.  *'
          MSGN = 1
          SKIP = 1
```

```
      CALL MSGPRT(MSG$,MSGN,SKIP)
  300 MSG$(1)= 'TYPE "1" FOR THE VALENCY RANKINGS OF COMPONENTS  *'
      MSG$(2)= 'TYPE "2" TO COMPUTE THE AVERAGE PATH RANKINGS    *'
      MSG$(3)= 'TYPE "3" TO COMPUTE THE MINIMUM LINK RANKINGS    *'
      MSG$(4)= 'TYPE "4" TO COMPUTE THE CRITICAL PATH RANKINGS   *'
      MSG$(5)= 'TYPE "5" TO COMPUTE THE CUT-COMPONENT RANKINGS   *'
      MSG$(6)= 'TYPE "6" FOR THE DISRUPTIVE MEASURE RANKINGS     *'
      MSGN = 6
  310 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, *, ERR = 320) REPLY
      IF(REPLY.EQ.1) THEN
        CALL VALNCY(MXCMPT,MTLBL$,MTDMSN,IMAX)
      ELSE IF(REPLY.EQ.2) THEN
        CALL DSTANS(MXLINK,MTLBL$,MTDMSN)
      ELSE IF(REPLY.EQ.3) THEN
        CALL MNPATH(MXMINA,MXMIND,MTLBL$,MTDMSN)
      ELSE IF(REPLY.EQ.4) THEN
        CALL CRITCL(MXLINK,MTLBL$,MTDMSN,NFILER)
      ELSE IF(REPLY.EQ.5) THEN
        CALL CUTTER(MXCMPT,MXLINK,MTLBL$,MTDMSN)
      ELSE IF(REPLY.EQ.6) THEN
        CALL DISRPT(MXCMPT,MTLBL$,MTDMSN)
      ELSE
  320   PRINT*,ERMSG$
        GO TO 310
      END IF
      MSG$(1)= 'FURTHER IMPORTANCE RANKINGS (YES/NO) ? *'
      MSGN = 1
  330 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A3)') REPLY$
      IF(REPLY$.EQ.'YES') THEN
        GO TO 300
      ELSE IF(REPLY$.NE.'NO') THEN
        PRINT*,ERMSG$
        GO TO 330
      END IF
      GO TO 280
    ELSE IF(REPLY.EQ.5) THEN
      CALL RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
   *  NTRACK,ERMSG$,HDG$,TIME$)
      GO TO 350
    ELSE
  340 PRINT*,ERMSG$
      GO TO 290
    END IF
    GO TO 280
  350 MSG$(1)= 'DO YOU WISH TO INITIATE A NEW RUN ?  (YES/NO) *'
    MSGN = 1
    CALL MSGPRT(MSG$,MSGN,SKIP)
    READ(*, '(A3)', ERR = 360) REPLY$
    IF(REPLY$.EQ.'YES') THEN
      GO TO 90
    ELSE IF(REPLY$.NE.'NO') THEN
  360 PRINT*,ERMSG$
      GO TO 350
    END IF
    CALL CLOSE(13,0)
    STATUS = FACSF2('@FREE 13 . ')
  999 STOP
    END


*****************************************************************
      SUBROUTINE MSGPRT(MSG$,MSGN,SKIP)

* THIS ROUTINE PRINTS THE PROGRAMME MESSAGES TO THE TERMINAL SCREEN
*
*  ARGUMENTS
*  MSG$ - ARRAY USED TO PASS A MAX OF SIX MESSAGE LINES TO THE ROUTINE
*  MSGN - THE NUMBER OF LINES TO THE MESSAGE
*  SKIP - AN INDICATOR TO INCLUDE (0) OR SKIP (1) THE REPLY PROMPT
*
*  OTHER ARRAYS AND VARIABLES
*  FMT$ - CHARACTER STRING FOR THE VARIABLE PRINTING FORMAT
*  TRANS - CHARACTER REPRESENTATION OF MSGN FOR VARIABLE FORMAT STRING
*
      INTEGER SKIP, MSGN
      CHARACTER MSG$*60(6),
   *            FMT$*24/'(/,N(5X,''* '',A,/))'/, TRANS*1
*
*  SETTING THE NUMBER OF MESSAGE LINES IN THE VARIABLE FORMAT
*
      WRITE(TRANS,'(I1)') MSGN
      FMT$(4:4) = TRANS
*
*  WRITING THE MESSAGE TO THE TERMINAL SCREEN
*
      WRITE(*, FMT$) (MSG$(N), N = 1,MSGN)
*
*  CHECKING IF THE PROMPT MUST BE PRINTED
*
      IF(SKIP.EQ.1) GO TO 105
      PRINT*,' ????....'
*
*  SETTING THE PROMPT INDICATOR TO THE AFFIRMATIVE
*
  105 SKIP = 0
      RETURN
      END


*****************************************************************
      SUBROUTINE PRTWRT(MXTRIX,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
*
*  THIS ROUTINE PRINTS VARIOUS FORMS OF THE COMPONENT MATRIX TO THE
*  ALTERNATIVE PRINTFILE (FILE 2) TO PROVIDE HARDCOPY TO THE USER
*
*  ARGUMENTS
*  MXTRIX - THE MATRIX WHICH IS TO BE PRINTED
*  MTLBLS - THE CHARACTER ARRAY OF THE COMPONENT LABELS
*  MTDMSN - THE DIMENSION OF THE COMPONENT MATRIX
*  TTLN - AN INTEGER INDICATOR OF WHICH FORM OF MATRIX IS BEING PRINTED
*         (1=INITIAL: 2=EDITED: 3=MINIMUM LINK: 4=LAST COPY)
*  HDG$ - A CHARACTER VARIABLE WITH A USER SUPPLIED DESCRIPTIVE HEADING
*  HEAD$ - A CHARACTER VARIABLE, PROGRAMME SUPPLIED, GIVING THE DATE,
*          TIME, AND TYPE OF DATA FILE USED
*
*  OTHER ARRAYS AND VARIABLES
*  FMT$ - A CHARACTER VARIABLE FOR THE VARIABLE PRINTING FORMAT
*  TRAN$ - A CHARACTER VARIABLE FOR INSERTING THE MATRIX DIMENSION
*          INTO THE VARIABLE FORMAT
*  ASTRK$ - A CHARACTER VARIABLE OF ASTERISKS FOR PRINTING THE MATRIX
*  UNLN$ - A CHARACTER VARIABLE OF DASHES TO UNDERLINE HEADINGS
*
*
      INTEGER MXTRIX(50,50),
   +          TTLN
      CHARACTER MTLBL$*20(50),
   +            HDG$*50, HEAD$*85, FMT$*32, TRAN$*2, ASTRK$*126,
   +            UNLN$*126
*
*  INITIALIZING ASTRK$ AND UNLN$ IF THIS IS THE FIRST CALL TO PRTWRT
*  I.E. IF THIS IS TO PRINT AN INITIAL COMPONENT MATRIX (TTLN = 1)
*
      IF(TTLN.EQ.1) THEN
        DO 105 N = 1, 126
          ASTRK$(N:N) = '*'
          UNLN$(N:N) = '-'
  105   CONTINUE
      END IF
*
*  THE COMPONENT LABELS ARE PRINTED IF THE MATRIX IS NOT THE MINIMUM
*  LINK MATRIX (TTLN NOT EQUAL TO 3)
*
      IF(TTLN.NE.3) THEN
        WRITE(2, '(1H1,A,//,6X,A,/,6X,A)') HEAD$,HDG$,ASTRK$
        WRITE(2, '(/,6X,''THE COMPONENTS ARE AS FOLLOWS:-'',/)')
        WRITE(2, '(50(6X,I2,2X,A,/:))') (N,MTLBL$(N), N = 1,MTDMSN)
*
*  THE SIZE AND FORM OF THE MATRIX IS CHECKED FROM THE VARIABLES
*  'TTLN' AND 'MTDMSN' AND APPROPRIATE HEADINGS ARE PRINTED
*
        IF(MTDMSN.GT.31) THEN
          WRITE(2, '(//,6X,''*** YOUR MATRIX IS TOO LARGE TO FIT ON'',
   +      '' A SINGLE PAGE   ***'',/,6X,''*** THE FOLLOWING PAGES MUST'',
   +      '' BE JOINED WHERE NECESSARY ***'')')
        END IF
      END IF
      IF(TTLN.EQ.1) THEN
        WRITE(2,'(1H1,/,6X,''THE INITIAL COMPONENT INTERACTION MATRIX'',
```

```
     +   /,6X,A)') UNLN$(1:40)
        ELSE IF(TTLN.EQ.2) THEN
          WRITE(2,'(1H1,/,6X,''THE EDITED COMPONENT INTERACTION MATRIX'',
     +   /,6X,A)') UNLN$(1:39)
        ELSE IF(TTLN.EQ.3) THEN
          WRITE(2,'(1H1,/,6X,''THE MINIMUM LINK MATRIX'',/,6X,A)')
     +   UNLN$(1:23)
        ELSE
          WRITE(2, '(1H1)')
        END IF
*
*  THE MATRIX IS PRINTED ACCORDING TO THE VARIABLE FORMAT IN FMT$
*  THE 13TH AND 14TH CHARACTERS OF FMT$ ARE SET TO THE MATRIX
*  DIMENSION (MTDMSN) TO GIVE THE CORRECT FORM OF THE MATRIX
*  IF THE MATRIX IS DIMENSIONED LESS THAN 31 IT IS PRINTED ON ONE PAGE
*  ELSE IT IS PRINTED ON TWO PAGES WITH THE APPROPRIATE MESSAGES
*
        IF(MTDMSN.LE.31) THEN
          FMT$ = '(6X,I2,2H *,50I3,4H   * ,I2,2X,A)'
          WRITE(TRAN$,'(I2)') MTDMSN
          FMT$(13:14) = TRAN$
          WRITE(2, '(/,10X,50I3)') (N, N = 1,MTDMSN)
          WRITE(2, '(9X,A)') ASTRK$(1:((MTDMSN*3)+4))
          DO 115 I = 1, MTDMSN
            WRITE(2, FMT$) I, (MXTRIX(I,J), J = 1,MTDMSN), I, MTLBL$(I)
  115     CONTINUE
          WRITE(2, '(9X,A)') ASTRK$(1:((MTDMSN*3)+4))
          WRITE(2, '(10X,50I3)') (N, N = 1,MTDMSN)
        ELSE
          FMT$ = '(2X,30I3,''  * '',I2,2X,A)'
          WRITE(TRAN$,'(I2)') (MTDMSN-29)
          FMT$(5:6) = TRAN$
          WRITE(2, '(/,10X,30I3)') (N, N = 1,30)
          WRITE(2, '(9X,A)') ASTRK$(1:91)
          DO 125 I = 1, MTDMSN
            WRITE(2, '(6X,I2,'' *'',30I3)') I, (MXTRIX(I,J), J = 1,30)
  125     CONTINUE
          WRITE(2, '(9X,A)') ASTRK$(1:91)
          WRITE(2, '(10X,30I3)') (N, N = 1,30)
          WRITE(2, '(1H1,/,5X,41H(ALIGN SUITABLY WITH THE LEFT SIDE OF THE,
     +   17H PRECEEDING PAGE),/,6X,A)') UNLN$(1:58)
          WRITE(2, '(/,2X,30I3)') (N, N = 30,MTDMSN)
          WRITE(2, '(2X,A)') ASTRK$(1:((MTDMSN-28)*3))
          DO 135 I = 1,MTDMSN
            WRITE(2, FMT$) (MXTRIX(I,J), J = 30,MTDMSN), I, MTLBL$(I)
  135     CONTINUE
          WRITE(2, '(2X,A)') ASTRK$(1:((MTDMSN-28)*3))
          WRITE(2, '(2X,30I3)') (N, N = 30,MTDMSN)
        END IF
*
* CONTROL IS RETUNED TO 'EDITOR' OR THE MAIN PROGRAMME
*
        RETURN
        END


*****************************************************************
        SUBROUTINE DIMINP(MSG$,MTDMSN,ERMSG$)
*
*  THIS ROUTINE ALLOWS THE USER TO ENTER THE COMPONENT MATRIX DIMENSION
*
*  ARGUMENTS
*  MSG$ - CHARACTER ARRAY OF PROGRAM MESSAGES TO THE USER FOR PRINTING
*  MTDMSN - THE COMPONENT MATRIX DIMENSION VALUE (0<MTDMSN<51)
*           BY THE ROUTINE 'MSGPRT'
*  ERMSG$ - CHARACTER VARIABLE CONTAINING THE STANDARD ERROR MESSAGE
*
*  OTHER ARRAYS AND VARIABLES
*  MSGN - NUMBER OF MESSAGE LINES IN 'MSG$' TO BE PRINTED BY 'MSGPRT'
*  SKIP - INTEGER (0 OR 1) INDICATOR TO ROUTINE 'MSSPRT' TO SIGNAL FOR
*         THE USER PROMPT
*  REPLY$ - CHARACTER VARIABLE TO RECEIVE USER REPLY (YES/NO) TO PROG-
*           RAMME QUESTIONS
*
        INTEGER MSGN, SKIP/0/
        CHARACTER MSG$*60(6),
     +           REPLY$*3, ERMSG$*50
*
*  ENTERING THE MATRIX DIMENSION (MTDMSN)
*
```

```
  105 MSG$(1)= 'WHAT IS THE DIMENSION OF THE COMPONENT MATRIX ? *'
      MSG$(2)= '(ENTER AN INTEGER VALUE BETWEEN ONE AND FIFTY)   *'
      MSGN = 2
  115 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ (*, *, ERR = 125) MTDMSN
*
*  CHECKING IF 'MTDMSN' IS WITHIN THE PROGRAMME LIMITS (0<MTDMSN<51)
*
        IF(MTDMSN.GT.50) GO TO 135
        IF((MTDMSN.LE.50).AND.(MTDMSN.GT.1)) GO TO 155
  125 PRINT*,ERMSG$
      GO TO 115
*
*  ALLOWING THE USER TO REDEFINE MTDMSN IF OUT OF LIMITS
*
  135 MSG$(1)= 'THE LARGEST MATRIX POSSIBLE IS 50 X 50.  WANT    *'
      MSG$(2)= 'TO RE-DIMENSION THE COMPONENT MATRIX ? (YES/NO) *'
      MSGN = 2
  145 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ (*, '(A3)', ERR = 135) REPLY$
      IF(REPLY$.EQ.'YES') THEN
        GO TO 135
      ELSE IF(REPLY$.NE.'NO') THEN
        PRINT*,ERMSG$
        GO TO 145
      END IF
*
*  IF THE USER DOES NOT WISH TO REDEFINE AN ERRONEOUS 'MTDMSN', MSG$(6)
*  IS USED TO INDICATE TO THE MAIN PROGRAM THAT ROUTINE 'RUNFIN' MUST
*  BE CALLED AND THE PROGRAMME IS TERMINATED
*
      MSG$(6)(1:5) = 'ABORT'
*
* CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
  155 RETURN
      END


*****************************************************************
      SUBROUTINE DATINP(MXCMPT,MTLBL$,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
*
*  THIS ROUTINE ALLOWS THE USER TO ENTER THE COMPONENT MATRIX DATA
*
*  ARGUMENTS
*  MXCMPT - THE COMPONENT MATRIX INTEGER ARRAY 50X50 MAXIMUM
*  MTLBL$ - THE CHARACTER ARRAY (50 MAX) OF COMPONENT LABELS
*  MSG$ - CHARACTER ARRAY FOR THE PROGRAMME MESSAGES TO BE PRINTED BY
*  MTDMSN - THE COMPONENT MATRIX DIMENSION VALUE
*  IMIN - THE MINIMUM VALUE OF A COMPONENT MATRIX ELEMENT
*  IMAX - THE MAXIMUM VALUE OF A COMPONENT MATRIX ELEMENT
*         (THIS PROGRAMME HAS IMIN = 0; AND IMAX = 1 OR 3 ONLY)
*  ERMSG$ - CHARACTER VARIABLE OF THE STANDARD ERROR MESSAGE
*         ROUTINE 'MSGPRT'
*
*  OTHER ARRAYS AND VARIABLES
*  SKIP - INTEGER INDICATOR (0 OR 1) TO SIGNAL THE PRINTING OF THE
*         USER PROMPT BY ROUTINE 'MSGPRT'
*  REPLY - INTEGER VARIABLE FOR USER REPLY TO QUESTIONS (1,2.... )
*  CHECK$ - CHARACTER VARIABLE FOR TESTING FORMAT OF USER INPUT
*
      INTEGER MXCMPT(50,50),
     +         IMIN, IMAX, SKIP, REPLY
      CHARACTER MTLBL$*20(50), MSG$*60(6),
     +          CHECK$*55, ERMSG$*50
*
*  ENTERING THE COMPONENT LABELS
*  EACH LABEL IS CHECKED FOR LENGTH USING CHECK$ AND THEN ENTERED TO
*  MTLBL$ ONCE CORRECT
*
      MSG$(1)= 'NOTE ANY MISTAKES YOU MAY MAKE WHEN ENTERING   *'
      MSG$(2)= 'DATA.  YOU WILL BE ABLE TO CORRECT THEM LATER *'
      MSGN = 2
      SKIP = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
      MSG$(1)= 'ENTER THE COMPONENT LABELS (ANY COMBINATION   *'
      MSG$(2)= 'OF BETWEEN 1 AND 20 ALPHANUMERIC CHARACTERS) *'
      MSGN = 2
      SKIP = 1
      CALL.MSGPRT(MSG$,MSGN,SKIP)
```

```
         DO 115 LBLN = 1,MTDMSN
   105     PRINT*,' '
           WRITE(*, '(''    COMPONENT '',I2,'' ?'')') LBLN
           READ(*, '(A25)', ERR = 105) CHECK$
           IF(CHECK$(21:25).NE.'     ') THEN
             PRINT*,'**** RE-ENTER THE LAST LABEL.  IT WAS TOO LONG.'
             PRINT*,'**** (LABEL TO BE 20 CHARACTERS OR LESS) '
             GO TO 105
           ELSE
             MTLBL$(LBLN) = CHECK$(1:20)
           END IF
   115 CONTINUE
*
*   THE USER SPECIFIES HOW THE MATRIX ELEMENTS WILL BE ENTERED.
*   FOR WHATEVER METHOD IS USED, THE INPUT IS CHECKED FOR MAGNITUDE
*   AND FORM: IF ERRORS ARE DETECTED THE USER RE-ENTERS THE ELEMENT.
*   IF THE ELEMENTS ARE ENTERED ROW BY ROW, THE NUMBER OF ELEMENTS IS
*   IS CHECKED BY FIRST READING THE INPUT AS A CHARACTER VARIABLE ('CHECK$')
*   AND TESTING THE PRESENCE OF BLANKS AND CHARACTERS.  IF ALL CORRECT THE
*   INPUT IS REREAD INTO 'MXCMPT'
*
         MSG$(1)='TYPE "1" TO ENTER THE MATRIX DATA AS ZEROES OR AS ONES *'
         MSG$(2)='TYPE "2" TO ENTER THE DEPENDENCIES ON A WEIGHTED SCALE *'
         MSGN = 2
   125 CALL MSGPRT(MSG$,MSGN,SKIP)
         READ(*, *, ERR = 125) REPLY
         IF(REPLY.EQ.1) THEN
           IMIN = 0
           IMAX = 1
         ELSE IF(REPLY.EQ.2) THEN
           IMIN = 0
           IMAX = 3
         ELSE
           PRINT*,ERMSG$
           GO TO 125
         END IF
         MSG$(1)= 'TYPE "1" TO ENTER THE COMPONENT INTERACTION *'
         MSG$(2)= '        MATRIX ELEMENT BY ELEMENT.          *'
         MSG$(3)= 'TYPE "2" TO ENTER THE COMPONENT INTERACTION *'
         MSG$(4)= '        MATRIX ROW BY ROW.                  *'
         MSGN = 4
   135 CALL MSGPRT(MSG$,MSGN,SKIP)
         READ(*, *, ERR = 265) REPLY
*
*   ENTERING THE ELEMENTS ONE AT A TIME
*
         IF(REPLY.EQ.1) THEN
           MSG$(1)= 'TO ENTER THE ELEMENTS OF THE COMPONENT MATRIX  *'
           IF(IMAX.EQ.1) THEN
             MSG$(2)= 'TYPE "1" FOR A DEPENDENCY BETWEEN COMPONENTS   *'
             MSG$(3)= 'TYPE "0" FOR NO INTERACTION BETWEEN COMPONENTS *'
             MSGN = 3
           ELSE IF(IMAX.EQ.3) THEN
             MSG$(2)= 'TYPE "0" FOR NO INTERACTION BETWEEN COMPONENTS *'
             MSG$(3)= '           TYPE "1" FOR A LIMITED DEPENDENCE     *'
             MSG$(4)= '           TYPE "2" FOR A APPRECIABLE DEPENDENCE *'
             MSG$(5)= '           TYPE "3" FOR A COMPLETE DEPENDENCE    *'
             MSGN = 5
           END IF
           MSGN = 3
           SKIP = 1
           CALL MSGPRT(MSG$,MSGN,SKIP)
           DO 175 I = 1,MTDMSN
             WRITE(*,'('' **** INTERACTIONS WITH '',A)') MTLBL$(I)
             DO 165 J = 1,MTDMSN
   145         PRINT*,' '
               WRITE(*, '(''   THE DEPENDENCY OF '',A)') MTLBL$(I)
               WRITE(*, '(''                     ON '',A)') MTLBL$(J)
               PRINT*,'   ???... '
               READ(*, *, ERR = 155) MXCMPT(I,J)
*
*   CHECKING THE INPUT
*
               IF((MXCMPT(I,J).LT.IMIN).OR.(MXCMPT(I,J).GT.IMAX)) THEN
   155           PRINT*,ERMSG$
                 GO TO 145
               END IF
   165       CONTINUE
   175     CONTINUE
*
*   ENTERING THE ELEMENTS ROW BY ROW
*
         ELSE IF(REPLY.EQ.2) THEN
           MSG$(1)= 'TO ENTER THE ELEMENTS OF THE COMPONENT MATRIX  *'
           IF(IMAX.EQ.1) THEN
             MSG$(2)= 'TYPE "1" FOR A DEPENDENCY BETWEEN COMPONENTS   *'
             MSG$(3)= 'TYPE "0" FOR NO INTERACTION BETWEEN COMPONENTS *'
             MSGN = 3
           ELSE IF(IMAX.EQ.3) THEN
             MSG$(2)= 'TYPE "0" FOR NO INTERACTION BETWEEN COMPONENTS *'
             MSG$(3)= '           TYPE "1" FOR A LIMITED DEPENDENCE     *'
             MSG$(4)= '           TYPE "2" FOR A APPRECIABLE DEPENDENCE *'
             MSG$(5)= '           TYPE "3" FOR A COMPLETE DEPENDENCE    *'
             MSGN = 5
           END IF
           SKIP = 1
           CALL MSGPRT(MSG$,MSGN,SKIP)
           MSG$(1)= 'TO ENTER THE MATRIX, TYPE IN ONE ROW AT A TIME *'
           MSG$(2)= 'WITH NO COMMAS OR SPACES BETWEEN THE ELEMENTS. *'
           MSGN = 2
           SKIP = 1
           CALL MSGPRT(MSG$,MSGN,SKIP)
           DO 255 I = 1,MTDMSN
   185       PRINT*,' '
             WRITE(*,'('' **** INTERACTIONS WITH '',A)') MTLBL$(I)
             PRINT*,' '
             PRINT*,'????....'
*
*   CHECKING IF THE CORRECT NUMBER OF ELEMENTS ARE PRESENT
*
             READ(*, '(A55)', ERR = 195) CHECK$
             IF((CHECK$(MTDMSN:MTDMSN).EQ.' ').OR.(CHECK$(MTDMSN+1:
     +         MTDMSN+5).NE.'     ')) THEN
   195         PRINT*,'**** RE-ENTER THE LAST ROW.  IT WAS FAULTY.'
               GO TO 185
             ELSE
               READ(0, '(50(I1))', ERR = 195) (MXCMPT(I,J), J = 1, MTDMSN)
             END IF
*
*   CHECKING EACH ELEMENT
*
             DO 245 J = 1,MTDMSN
   205         IF((MXCMPT(I,J).LT.IMIN).OR.(MXCMPT(I,J).GT.IMAX)) THEN
                 WRITE(*, '(//,''**** THE '',I2,'' ELEMENT IN THE LAST '',
     +           ''ROW IS FAULTY.'')') J
                 WRITE(*, '(//,3X,A,/,2X,50I2)') MTLBL$(I),
     +           (MXCMPT(I,N),N = 1,MTDMSN)
                 MSG$(1)= 'TYPE "1" TO RE-ENTER THE ELEMENT   *'
                 MSG$(2)= 'TYPE "2" TO RE-ENTER THE WHOLE ROW *'
                 MSGN = 2
   215           CALL MSGPRT(MSG$,MSGN,SKIP)
                 READ(*, *, ERR = 235) REPLY
                 IF(REPLY.EQ.1) THEN
                   MSG$(1)= 'ENTER THE CORRECTED VALUE  *'
                   MSGN = 1
   225             CALL MSGPRT(MSG$,MSGN,SKIP)
                   READ(*, *, ERR = 225) MXCMPT(I,J)
                   GO TO 205
                 ELSE IF(REPLY.EQ.2) THEN
                   GO TO 185
                 ELSE
   235             PRINT*,ERMSG$
                   GO TO 215
                 END IF
               END IF
   245       CONTINUE
   255     CONTINUE
         ELSE
   265     PRINT*,ERMSG$
           GO TO 135
         END IF
*
* CONTROL RETURNED TO MAIN PROGRAMME
*
         RETURN
         END

**********************************************************************
         SUBROUTINE MXFSRT(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
```

```
*
*   THIS ROUTINE WRITES THE COMPONENT MATRIX DATA TO A PERMANENT DATA FILE
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY (50X50) OF THE COMPONENT MATRIX DATA
*   MTLBL$ - CHARACTER ARRAY (50) OF THE COMPONENT LABELS
*   MTDMSN - THE DIMENSION OF THE COMPONENT MATRIX
*   IMIN - THE MINIMUM (INTEGER) VALUE OF A MATRIX ELEMENT
*   IMAX - THE MAXIMUM (INTEGER) VALUE OF A MATRIX ELEMENT (EITHER 1 OR 3)
*
*   FILES
*   UNIT 1 - THE PERMANENT DATA FILE CREATED BY THE PROGRAMME AND NAMED
*                 BY THE USER
*
*   THE DATA IS WRITTEN TO THE FILE IN THE FOLLOWING ORDER:-
*   RECORD 1 - THE MATRIX DIMENSION 'MTDMSN'
*   RECORD 2 - THE MINIMUM AND MAXIMUM ELEMENT VALUES, 'IMIN' AND 'IMAX'
*   RECORD 3 - LABEL ONE
*   RECORD 4 - ROW ONE OF THE COMPONENT MATRIX STORED IN ARRAY 'MXCMPT'
*   RECORD 5 - LABEL TWO.
*   RECORD 6 - ROW TWO
*   RECORD 7 - ETC.
*   ALL THE RECORDS ARE FORMATTED AND ARE READ WITH THE SAME FORMAT
*   BY THE ROUTINE 'MXFRD' WHEN INPUTTED TO THE PROGRAMME
*
      INTEGER MXCMPT(50,50),
     +        IMIN, IMAX
      CHARACTER MTLBL$*20(50)
      WRITE(1, '(I2)') MTDMSN
      WRITE(1, '(2(I2))') IMIN, IMAX
      DO 105 I = 1 ,MTDMSN
        WRITE(1, '(A)') MTLBL$(I)
        WRITE(1, '(50(I2))')(MXCMPT(I,J), J = 1,MTDMSN)
  105 CONTINUE
      ENDFILE 1
      REWIND 1
* CONTROL RETURNED TO THE MAIN PROGRAMME
*
      RETURN
      END


*****************************************************************
      SUBROUTINE MXFRD(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
*
*   THIS ROUTINE READS THE COMPONENT MATRIX DATA INTO THE PROGRAMME FROM
*   A PERMANENT DATA FILE REFERENCED BY THE USER, ASSIGNED BY THE PROGRAMME
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY (50,50) TO HOLD THE COMPONENT MATRIX ELEMENTS
*   MTLBL$ - CHARACTER ARRAY (50) OF THE COMPONENT LABELS
*   MTDMSN - THE COMPONENT MATRIX DIMENSION
*   IMIN - THE MINIMUM (INTEGER) VALUE OF A MATRIX ELEMENT (SET TO 0)
*   IMAX - THE MAXIMUM VALUE OF A MATRIX ELEMENT (1 OR 3)
*
*   FILES
*   UNIT 1 - THE PERMANENT DATA FILE SPECIFIED BY THE USER AND ASSIGNED
*                 BY THE PROGRAMME
*   THE DATA IS READ IN THE ORDER AND FORMAT SPECIFIED IN ROUTINE 'MXFWRT'
*
      INTEGER MXCMPT(50,50), IMIN, IMAX
      CHARACTER MTLBL$*20(50)
      READ(1, '(2(I2))') IMIN, IMAX
      DO 105 I = 1,MTDMSN
        READ(1,'(A20)') MTLBL$(I)
        READ(1,'(50(I2))') (MXCMPT(I,J), J = 1,MTDMSN)
  105 CONTINUE
      REWIND 1
*
* CONTROL RETURNED TO MAIN PROGRAMME
*
      RETURN
      END


*****************************************************************
      SUBROUTINE EDITOR(MXCMPT,MTLBL$,MSG$,MTDMSN,FTYPE,FNEND,IMIN,IMAX,
     +TTLN,ERMSG$,HEAD$,HDG$,FNAME$,DATE$,TIME$)
*
*   THIS ROUTINE OFFERS THE USER VARIOUS MEANS OF EDITING THE COMPONENT
*   MATRIX DATA
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY OF THE COMPONENT MATRIX DATA
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MSG$ - CHARACTER ARRAY OF UP TO SIX MESSAGE LINES TO BE PRINTED BY
*          TO THE TERMINAL SCREEN BY 'MSGPRT'
*   MTDMSN - INTEGER VARIABLE OF THE COMPONENT MATRIX DIMENSION
*   FTYPE - INTEGER INDICATOR OF THE MODE OF DATA ENTRY (1=EXISTING DATA
*           FILE, 2=NEWLY ASSIGNED FILE, 3=NO PERMANENT COPY)
*   FNEND - INTEGER INDICATOR OF THE NUMBER OF CHARACTERS TO THE USER
*           SUPPLIED DATA FILE NAME
*   IMIN - INTEGER INDICATOR OF THE MINIMUM VALUE IN 'MXCMPT' (ZERO)
*   IMAX - INTEGER INDICATOR OF THE MAXIMUM VALUE IN 'MXCMPT' (1 OR 3)
*   TTLN - INTEGER INDICATOR OF THE FORM OF A MATRIX TO BE PRINTED BY
*          'PRTWRT' (1=INITIAL, 2=EDITED, 3=MIN LINK, 4=LAST VERSION)
*   ERMSG$ - CHARACTER VARIABLE WITH STANDARD ERROR MESSAGE
*   HEAD$ - CHARACTER VARIABLE OF A PROGRAMME SUPPLIED HEADING FOR
*           HARDCOPY PRINTOUT FROM 'PRTWRT'
*   HDG$ - CHARACTER VARIABLE OF A USER SUPPLIED HEADING FOR THE HARDCOPY
*          PRINTOUT CREATED BY 'PRTWRT'
*   FNAME$ - USER SUPPLIED NAME FOR THE DATA FILE (UNIT 1)
*   DATE$ - CHARACTER VARIABLE FOR THE DATE SUPPLIED BY A CALL TO THE
*           SYSTEM FUNCTION 'ADATE$'.  USED IN 'HEAD$'
*   TIME$ - CHARACTER VARIABLE FOR THE TIME SUPPLIED BY A CALL TO THE
*           SYSTEM FUNCTION 'ADATE$'.  USED IN 'HEAD$' AND THE PRINTFILE
*           NAME (I.E. "PF + 'TIME$'")
*
*   OTHER ARRAYS AND VARIABLES
*   MSGN - THE NUMBER OF MESSAGE LINES IN MSG$ FOR PRINTING BY 'MSGPRT'
*   SKIP - INTEGER INDICATOR TO 'MSGPRT' TO PRINT THE USER PROMPT
*   REPLY - INTEGER VARIABLE FOR USER REPLIES TO PROGRAMME OPTIONS
*   STATUS - INTEGER VARIABLE OF THE STATUS RETURNED BY A CALL TO
*            FASCF2
*   HDGEND - INTEGER COUNTER OF THE CHARACTERS IN THE USER SUPPLIED
*            'HDG$'
*   FACSF2 - UNIVAC SYSTEM INTEGER FUNCTION TO EXECUTE EXEC8 COMMANDS
*            FROM WITHIN A PROGRAMME (SEE SPERRY UNIVAC PROGRAMMER
*            REFERENCE MANUAL UP-8244.1, ASCII FORTRAN LEVEL 9RI, P7-31)
*   ASGNED - INTEGER VARIABLE SET AT EQUIVALENT OF AN OCTAL VALUE TO
*            TEST THE STATUS RETURNED BY A CALL TO FASCF2 TO ASSIGN A
*            FILE. IF 'STATUS' = 'ASGNED' THEN THE FILE IS ALREADY ASSIGNED
*   REPLY$ - CHARACTER VARIABLE FOR USER REPLIES TO PROGRAMME QUESTIONS
*
      INTEGER MXCMPT(50,50),
     +        MTDMSN, FTYPE, FNEND, IMIN, IMAX, TTLN,
     +        MSGN, SKIP, REPLY, STATUS, HDGEND, FACSF2,
     +        ASGNED/010000000000/
      CHARACTER MTLBL$*20(50), MSG$*60(6),
     +          ERMSG$*50, HEAD$*85, HDG$*55, FNAME$*12,
     +          DATE$*8, TIME$*8,
     +          REPLY$*3
*
*   THE EDITING OPTION IS CHOSEN FROM THE LIST BELOW.  FOR OPTIONS
*   3 AND 4 THE STATUS OF 'MTDMSN' IS CHECKED.  IN ALL OPTIONS
*   A SUBROUTINE IS CALLED TO PERFORM THE EDITING
*
  105 MSG$(1)= 'TYPE "1" TO CHANGE AN INDIVIDUAL ELEMENT   *'
      MSG$(2)= 'TYPE "2" TO CHANGE A COMPONENT LABEL       *'
      MSG$(3)= 'TYPE "3" TO DELETE A COMPONENT(COL & ROW) *'
      MSG$(4)= 'TYPE "4" TO ADD A COMPONENT TO THE MATRIX *'
      MSG$(5)= 'TYPE "5" TO TERMINATE THE PROGRAMME        *'
      MSGN = 5
  115 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, *, ERR = 175) REPLY
      IF(REPLY.EQ.1) THEN
        CALL EDELMT(MXCMPT,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
      ELSE IF(REPLY.EQ.2) THEN
        CALL EDLBL$(MTLBL$,MSG$,MTDMSN,ERMSG$)
      ELSE IF(REPLY.EQ.3) THEN
*
*  MSG$(6) ACTS AS AN INDICATOR TO 'EDSRTR' AS TO WHETHER IT IS CALLED
*  AS A REPEAT OPTION TO DELETE MORE THAN ONE COMPONENT AT A TIME
*
        MSG$(6) = 'INITIAL'
  125   IF(MTDMSN.EQ.2) THEN
          MSG$(1)= 'THE MATRIX IS PRESENTLY DIMENSIONED AT     *'
          MSG$(2)= '(2 X 2).  DELETING A COMPONENT WILL GIVE *'
          MSG$(3)= 'A TRIVIAL FORM OF THE INTERACTION MATRIX *'
          MSGN = 3
```

```
          SKIP = 1
          CALL MSGPRT(MSG$,MSGN,SKIP)
          GO TO 185
        END IF
        CALL EDSPTR(MXCMPT,MTLBL$,MSG$,MTDMSN,ERMSG$)
        MTDMSN = MTDMSN-1
        MSG$(1)= 'FURTHER COMPONENT DELETIONS ? (YES/NO) *'
        MSGN = 1
  135   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, '(A3)', ERR = 135) REPLY$
        IF(REPLY$.EQ.'YES') THEN
          GO TO 125
        ELSE IF(REPLY$.NE.'NO') THEN
          PRINT*,ERMSG$
          GO TO 135
        END IF
      ELSE IF(REPLY.EQ.4) THEN
  145   IF((MTDMSN+1).GT.50) THEN
          MSGT(1)= 'MAXIMUM DIMENSIONS (50 X 50) FOR THE COMPONENT *'
          MSG$(2)= 'MATRIX WILL BE EXCEEDED IF A FURTHER COMPONENT *'
          MSG$(3)= 'IS ADDED.     DO YOU WISH TO CONTINUE WITH THE *'
          MSG$(4)= 'PRESENT MATRIX ?                       (YES/NO) *'
          MSGN = 4
  155     CALL MSGPRT(MSG$,MSGN,SKIP)
          READ(*, '(A3)', ERR = 155) REPLY$
          IF(REPLY$.EQ.'YES') THEN
            GO TO 185
          ELSE IF(REPLY$.NE.'NO') THEN
            PRINT*,ERMSG$
            GO TO 155
          ELSE
*
*  IF THE USER WISHES TO ABORT THE RUN, HEAD$ IS SET TO 'ABORT' AND
*  THIS WILL INITIATE A CALL TO 'RUNFIN' ONCE CONTROL IS RETURNED TO
*  THE MAIN PROGRAMME
*
            HEAD$ = ' * RUN ABORTED - MATRIX AT MAXIMUM DIMENSIONS *'
            WRITE(2, '(1H1,//////////,15X,A)') HEAD$
*
*  IF A DATAFILE IS BEING USED (TTLN NOT EQUAL TO 3) THEN THE LAST STORED
*  VERSION OF THE MATRIX IS PRINTED TO THE ALTERNATE PRINTFILE BY A
*  CALL TO 'PRTWRT'
*
            IF(FTYPE.NE.3) THEN
              HEAD$ = ' * THE FOLLOWING LISTS THE LAST VERSION OF THE '
     +        //'MATRIX DATA IN FILE - '//FNAME$(1:FNEND)//' *'
              TTLN = 4
              CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
            END IF
          HEAD$ = 'ABORT'
          RETURN
          END IF
        END IF
        CALL EDADTN(MXCMPT,MTLBL$,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
        MTDMSN = MTDMSN+1
        MSG$(1)= 'FURTHER COMPONENT ADDITIONS ? (YES/NO) *'
        MSGN = 1
  165   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, '(A3)', ERR = 165) REPLY$
        IF(REPLY$.EQ.'YES') THEN
          GO TO 145
        ELSE IF(REPLY$.NE.'NO') THEN
          PRINT*,ERMSG$
          GO TO 165
        END IF
      ELSE IF(REPLY.EQ.5) THEN
*
*  THE SAME ROUTINE FOR TERMINATING THE PROGRAMME IS USED AS ABOVE
*
        HEAD$ = '                  *     RUN TERMINATED WHILE EDITING    *'
        WRITE(2, '(1H1,//,2X,A,////////,15X,A)') HDG$,HEAD$
        IF(FTYPE.NE.3) THEN
          HEAD$ = ' * THE FOLLOWING LISTS THE LAST VERSION OF THE '//
     +    'MATRIX DATA IN FILE - '//FNAME$(1:FNEND)//' *'
          TTLN = 4
          CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
        END IF
        HEAD$ = 'ABORT'
        RETURN
      ELSE
```

```
  175   PRINT*,ERMSG$
        GO TO 115
      END IF
*
*  AN OPTION IS GIVEN TO PERFORM FURTHER EDITING.  IF EDITING
*  IS COMPLETE, THE USER IS OFFERED THE OPTION OF CHANGING THE
*  HEADING IN 'HDG$.  IF THE PROGRAMME DATA IS ON FILE (FTYPE = 2),
*  THE USER IS OFFERED VARIOUS UPDATE OPTIONS
*
  185 MSG$(1)= 'FURTHER EDITING OF ANY KIND ?  (YES/NO) *'
      MSGN = 1
  195 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A3)', ERR = 195) REPLY$
      IF(REPLY$.EQ.'YES') THEN
        GO TO 105
      ELSE IF(REPLY$.NE.'NO') THEN
        PRINT*,ERMSG$
        GO TO 195
      END IF


      MSG$(1)= 'DO YOU WISH TO REVISE THE PRINT-OUT HEADING ?  *'
      MSG$(2)= '(YES/NO).                                      *'
      MSGN = 2
  205 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A3)', ERR = 205) REPLY$
      IF(REPLY$.EQ.'YES') THEN
        MSG$(1)= 'ENTER THE NEW HEADING ( < 50 CHARS ). *'
        MSGN = 1
  215   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, '(A55)', ERR = 235) HDG$
        HDGEND = 55
  225   IF((HDGEND.GT.0).AND.(HDG$(HDGEND:HDGEND).EQ.' ')) THEN
          HDGEND = HDGEND - 1
          GO TO 225
        END IF
        IF(HDGEND.GT.50) THEN
          PRINT*,ERMSG$
          GO TO 215
        END IF
      ELSE IF(REPLY$.NE.'NO') THEN
  235   PRINT*,ERMSG$
        GO TO 205
      END IF
  245 IF(FTYPE.NE.3) THEN
        HEAD$ = '       DATE : '//DATE$//'       TIME : '//TIME$//
     +  '        DATA FILE (ASSIGNED) : '//FNAME$
        MSG$(1)= 'TYPE "1" TO OVERWRITE YOUR DATA FILE WITH THE *'
        MSG$(2)= '          REVISED COMPONENT INTERACTION MATRIX. *'
        MSG$(3)= 'TYPE "2" TO WRITE THE REVISED COMPONENT INTER *'
        MSG$(4)= '             -ACTION MATRIX TO A NEW FILE.    *'
        MSG$(5)= 'TYPE "3" FOR NO PERMANENT COPY OF THE REVISED *'
        MSG$(6)= '           COMPONENT INTERACTION MATRIX.      *'
        MSGN = 6
  255   CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, *, ERR = 315) REPLY
        IF(REPLY.EQ.1) THEN
*
*  THE FILE IS OVERWRITTEN BY A CALL TO 'MXFWRT'
*
          CALL MXFWRT(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
          HEAD$ = '       DATE : '//DATE$//'       TIME : '//TIME$//
     +    '        DATA FILE (OVER-WRITTEN) : '//FNAME$
        ELSE IF(REPLY.EQ.2) THEN
*
*  THE USER SUPPLIES A NAME FOR THE DATA FILE.  THIS IS CHECKED FOR
*  UNIQUENESS.  ONCE THE FILE IS ASSIGNED, THE DATA IS WRITTEN TO IT
*  BY A CALL TO 'MXFWRT'.  THE OLD DATAFILE IS FREED FROM THE RUN.
*  ONCE THE NEW FILE IS ASSIGNED, THE HEADING IN 'HEAD$' IS REVISED.
*
  265     STATUS = FACSF2('@FREE '//FNAME$(1:FNEND)//' . ')
          MSG$(1)= 'ENTER A NAME (AS ANY COMBINATION OF 1 TO 12 *'
          MSG$(2)= 'ALPHANUMERIC CHARACTERS) FOR THE DATA FILE. *'
          MSGN = 2
  275     CALL MSGPRT(MSG$,MSGN,SKIP)
          READ(*, '(A12)', ERR = 275) FNAME$
*
* THE LENGTH OF THE FILE NAME IS CHECKED TO BE LESS THAN 12
*
          FNEND = 12
```

```
  285      IF((FNEND.GT.0).AND.(FNAME$(FNEND:FNEND).EQ.' ')) THEN
             FNEND = FNEND - 1
             GO TO 285
           END IF
           IF(FNEND.EQ.0) GO TO 275
           STATUS = FACSF2('@ASG,UP '//FNAME$(1:FNEND)//'.,F2 . ')
           IF(STATUS.NE.0) THEN
*
*  IF THE STATUS OF THE CALL TO FACSF2 IS NOT IN ORDER, CORRECTIVE
*  OPTIONS ARE OFFERED
*
              MSG$(1)= 'A FILE OF THIS NAME ALREADY EXISTS UNDER     *'.
              MSG$(2)= 'YOUR PROJECT-ID.                             *'
              MSG$(3)= 'TYPE "1" TO ENTER ANOTHER NAME FOR THE FILE *'
              MSG$(4)= 'TYPE "2" TO OVER-WRITE THIS EXISTING FILE   *'
              MSG$(5)= 'TYPE "3" TO TERMINATE THE PROGRAMME         *'
              MSGN = 5
  295         CALL MSGPRT(MSG$,MSGN,SKIP)
              READ(*, *, ERR = 305) REPLY
              IF(REPLY.EQ.1) THEN
                GO TO 265
              ELSE IF(REPLY.EQ.2) THEN
                IF(STATUS.NE.ASGNED) THEN
                  STATUS = FACSF2('@ASG,A '//FNAME$(1:FNEND)//' . ')
                END IF
                HEAD$ = '      DATE : '//DATE$//'      TIME : '//TIMES//
     +          '        DATA FILE (OVER-WRITTEN) : '//FNAMES
              ELSE IF(REPLY.EQ.3) THEN
*
*  IF THE RUN IS TO BE TERMINATED, THE LAST VERSION OF THE MATRIX IS
*  IS PRINTED BY A CALL TO 'PRTWRT' WITH TTLN = 4.  A TEMPORARY FILE
*  FOR UNIT 1 IS ASSIGNED SO THAT THE FILE EXISTS FOR FREEING IN 'RUNFIN'
*  'HEAD$' IS SET TO 'ABORT' TO INITIATE A CALL TO 'RUNFIN' WHEN CONTROL
*  IS RETURNED TO THE MAIN PROGRAMME
*
                STATUS = FACSF2('@ASG,T 1. . ')
                HEAD$ = ' * RUN ABORTED - FILE '//FNAME$(1:FNEND)//
     +          ' COULD NOT BE ASSIGNED *'
                WRITE(2, '(1H1,//////////,15X,A)') HEAD$
                HEAD$ = ' * THE FOLLOWING LISTS THE LAST VERSION OF THE '
     +          //'MATRIX DATA IN FILE - '//FNAME$(1:FNEND)//' *'
                TTLN = 4
                CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
                HEAD$ = 'ABORT'
                RETURN
              ELSE
  305           PRINT*,ERMSG$
                GO TO 295
              END IF
            ELSE
              HEAD$ = '      DATE : '//DATE$//'      TIME : '//TIMES//
     +        '        DATA FILE (ASSIGNED) : '//FNAMES
            END IF
            STATUS = FACSF2('@USE 1,'//FNAME$(1:FNEND)//' . ')
            CALL MXFWRT(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
          ELSE IF(REPLY.EQ.3) THEN
            HEAD$ = '      DATE : '//DATE$//'      TIME : '//TIMES//
     +      '        * NO PERMANENT COPY OF EDITED MATRIX *'
          ELSE
  315       PRINT*,ERMSG$
            GO TO 255
          END IF
        END IF
*
*  ONCE THE MATRIX IS EDITED AND APPPOPRIATE UPDATE OPTIONS HAVE BEEN
*  HANDLED, THE REVISED MATRIX IS WRITTEN TO THE ALTERNATE PRINTFILE (2)
*  BY A CALL TO 'PRTWRT' WITH TTLN SET TO 2
*
        TTLN = 2
        CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
        MSG$(1)= 'A COPY OF THE EDITED COMPONENT INTERACTION *'
        MSG$(2)= 'MATRIX HAS BEEN RECORDED ON YOUR PRINTFILE *'
        SKIP = 1
        MSGN = 2
        CALL MSGPRT(MSG$,MSGN,SKIP)
*
* CONTROL RETURNED TO THE MAIN PROGRAMME
*
        RETURN
        END
```

```
******************************************************************
      SUBROUTINE EDELMT(MXCMPT,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
*
*  THIS ROUTINE CORRECTS INDIVIDUAL ELEMENTS OF THE COMPONENT MATRIX
*
*  ARGUMENTS
*  MXCMPT - INTEGER ARRAY OF THE COMPONENT MATRIX ELEMENTS
*  MSG$ - CHARACTER ARRAY FOR UP TO 6 MESSAGES FOR PRINTING TO THE
*         TO THE TERMINAL SCREEN BY 'MSGPRT'
*  MTDMSN - INTEGER VARIABLE OF THE COMPONENT MATRIX DIMENSION
*  IMIN - INTEGER INDICATOR OF THE MINIMUM VALUE IN 'MXCMPT' (0)
*  IMAX - INTEGER INDICATOR OF THE MAXIMUM VALUE IN 'MXCMPT' (1 OR 3)
*  ERMSG$ - CHARACTER VARIABLE FOR THE STANDARD ERROR MESSAGE
*
*  OTHER ARRAYS AND VARIABLES
*  FIX - INTEGER VARIABLE TO RECEIVE AND CHECK USER INPUT
*  MSGN - INTEGER COUNTER OF THE NUMBER OF MESSAGES IN MSG$ TO BE PRINTED
*         BY 'MSGPRT'
*  SKIP - INTEGER INDICATOR TO 'MSGPRT' TO PRINT THE USER PROMPT
*  REPLY$ - CHARACTER VARIABLE FOR USER REPLIES TO PROGRAMME QUESTIONS
*
      INTEGER MXCMPT(50,50),
     +        MTDMSN, IMIN, IMAX,
     +        FIX, MSGN, SKIP
      CHARACTER MSG$*60(6),
     +        ERMSG$*50,
     +        REPLY$*3
*
*  SPECIFYING AND CORRECTING THE ELEMENT
*
  105 MSG$(1)= 'ENTER THE ROW NUMBER, COLUMN NUMBER, AND CORRECT *'
      MSG$(2)= 'VALUE OF THE MATRIX ELEMENT TO BE CORRECTED.     *'
      IF(IMAX.EQ.1) THEN
        MSG$(3)= '                       (ROW,COLUMN, "0" OR "1")  *'
      ELSE
        MSG$(3)= '      (ROW,COLUMN,INTEGER BETWEEN ZERO AND THREE) *'
      END IF
      MSGN = 3
  115 CALL MSGPRT(MSG$,MSGN,SKIP)
  125 READ(*, *, ERR = 145) I, J, FIX
*
*  THE INPUT IS CHECKED TO SEE IF THE ELEMENT EXISTS, AND WHETHER THE
*  NEW ELEMENT VALUE IS WITHIN THE LIMITS OF IMIN AND IMAX
*
      IF(( I .LE. 0 ).OR.( I .GT.MTDMSN)) GO TO 135
      IF(( J .LE. 0 ).OR.( J .GT.MTDMSN)) GO TO 135
      IF((FIX.LT.IMIN).OR.(FIX.GT.IMAX)) THEN
  135   PRINT*,ERMSG$
        GO TO 115
      END IF
*
*  ECHOING THE CORRECTION AND ENTERING THE CORRECTED VALUE TO MXCMPT
*
      WRITE(*, '(A,I2,A,I2,A,I2,A,I2)')
     +' **** THE DEPENDENCY BETWEEN ',I,' AND ',J,
     +' HAS BEEN CHANGED FROM ',MXCMPT(I,J),' TO ',FIX
      MXCMPT(I,J) = FIX
*
*  OPTION FOR FURTHER CORRECTIONS OR TO RETURN TO 'EDITOR'
*
      IF(IMAX.EQ.1) THEN
        MSG$(1)= 'ENTER ROW, COLUMN, "0" OR "1" (OR "EXIT") *'
      ELSE
        MSG$(1)='ENTER ROW, COLUMN, INTEGER BETWEEN 0 & 3 (OR "EXIT") *'
      END IF
      MSGN = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
      GO TO 125
  145 RETURN
      END


******************************************************************
      SUBROUTINE EDLBL$(MTLBL$,MSG$,MTDMSN,ERMSG$)
*
*  THIS ROUTINE EDITS THE COMPONENT LABELS
*
*  ARGUMENTS
```

```
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MSG$ - CHARACTER ARRAY OF UP TO SIX MESSAGE LINES FOR PRINTING
*           TO THE TERMINAL SCREEN BY 'MSGPRT'
*   MTDMSN - THE DIMENSION OF THE COMPONENT MATRIX
*   ERMSG$ - CHARACTER VARIABLE FOR THE STANDARD ERROR MESSAGE
*
*   OTHER ARRAYS AND VARIABLES
*   LBLN - THE NUMBER OF THE COMPONENT LABEL TO BE CORRECTED
*   REPLY$ - CHARACTER VARIABLE FOR USER REPLIES TO PROGRAMME QUESTIONS
*   CHECK$ - CHARACTER VARIABLE USED TO CHECK USER INPUT BEFORE ADJUSTING
*            THE PROGRAMME DATA
*
      INTEGER MTDMSN,
     *        LBLN
      CHARACTER MTLBL$*20(50), MSG$*60(6),
     *          ERMSG$*50,
     *          REPLY$*3, CHECK$*25
*
*   ENTERING THE LABEL NUMBER AND THE NEW LABEL
*
  105 MSG$(1)= 'ENTER THE NUMBER OF THE LABEL TO BE CHANGED *'
      MSGN = 1
  115 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, *, ERR = 125) LBLN
*
*   CHECKING IF THE LABEL TO BE CORRECTED ACTUALLY EXISTS
*
      IF((LBLN.LE.0).OR.(LPLN.GT.MTDMSN)) THEN
  125   PRINT*,ERMSG$
        GO TO 115
      END IF
      PRINT*,' '
      WRITE(*, '('' **** LABEL '',I2,'' IS PRESENTLY - '',A)')
     *LBLN, MTLBL$(LBLN)
      MSG$(1)= 'ENTER THE NEW LABEL ( 1-20 CHARACTERS ) *'
      MSGN = 1
  135 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A25)', ERR = 135) CHECK$
*
*   CHECKING THE LENGTH OF THE NEW LABEL
*
      IF(CHECK$(21:25).NE.'     ') THEN
        PRINT*,'**** RE-ENTER THE LABEL.  IT WAS TOO LONG. '
        PRINT*,'**** (LABEL TO BE 20 CHARACTERS OR LESS)   '
        GO TO 135
      ELSE
        MTLBL$(LBLN) = CHECK$(1:20)
      END IF
*
*   PROCEDURE IS REPEATED OR CONTROL IS RETURNED TO "EDITOR"
*
      MSG$(1)= 'FURTHER LABELS TO CHANGE ? (YES/NO) *'
      MSGN = 1
  145 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A3)', ERR = 145) REPLY$
      IF(REPLY$.EQ.'YES') THEN
        GO TO 105
      ELSE IF(REPLY$.NE.'NO') THEN
        PRINT*,ERMSG$
        GO TO 145
      END IF
      RETURN
      END


*********************************************************************
      SUBROUTINE EDSFTR(MXCMPT,MTLBL$,MSG$,MTDMSN,ERMSG$)
*
*   THIS ROUTINE DELETES A COMPONENT FROM THE COMPONENT MATRIX
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY OF THE COMPONENT MATRIX ELEMENTS
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MSG$ - CHARACTER ARRAY OF UP TO SIX MESSAGE LINES FOR PRINTING TO
*   THE TERMINAL SCREEN BY 'MSGPRT'
*   MTDMSN - THE COMPONENT MATRIX DIMENSION
*   ERMSG$ - CHARACTER VARIABLE OF THE STANDARD ERROR MESSAGE
*
*   OTHER ARRAYS AND VARIABLES
*   CMPTN - THE INTEGER VARIABLE OF THE NUMBER OF THE COMPONENT TO BE
*           DELETED
*   MSGN - INTEGER COUNTER OF THE NUMBER OF MESSAGES IN MSG$ TO BE
*          PRINTED BY 'MSGPRT'
*   SKIP - INTEGER INDICATOR TO 'MSGPRT' TO PRINT THE USER PROMPT
*   REPLY$ - CHARACTER VARIABLE FOR USER ANSWERS TO PROGRAMME QUESTIONS
*
      INTEGER MXCMPT(50,50),
     *        MTDMSN,
     *        CMPTN, MSGN, SKIP
      CHARACTER MTLBL$*20(50), MSG$*60(6),
     *          ERMSG$*50,
     *          REPLY$*3
*
*   IF THIS IS THE FIRST ENTRY TO THE ROUTINE THE COMPONENT LABELS ARE
*   PRINTED (I.E. IF MSG$(6) = 'INITIAL' AS SET IN 'EDITOR')
*
      IF(MSG$(6).EQ.'REPEAT') GO TO 105
      MSG$(1)= 'THE COMPONENTS ARE NUMBERED AS FOLLOWS *'
      MSGN = 1
      SKIP = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
      WRITE(*, '(25(2(6X,I2,2X,A20:)/))') (N,MTLBL$(N), N = 1,MTDMSN)
*
*   ENTERING THE NUMBER OF THE COMPONENT TO BE DELETED AND CHECKING
*   THAT SUCH A COMPONENT EXISTS
*
  105 MSG$(1)= 'ENTER THE NUMBER OF THE COMPONENT TO BE DELETED *'
      MSGN = 1
  115 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, *, ERR = 125) CMPTN
      IF((CMPTN.LT.1).OR.(CMPTN.GT.MTDMSN)) THEN
  125   PRINT*,ERMSG$
        GO TO 115
      END IF
*
*   THE PROCEDURE IS TO MOVE THE ELEMENTS IN THE ROW AND THEN THE COLUMN
*   OF THE COMPONENT TO BE DELETED TO THE OUTERMOST ROW OR COLUMN IN THE
*   ARRAY 'MXCMPT', AND THEN TO REDIMENSION THE MATRIX TO EXCLUDE THE
*   LAST ROW AND COLUMN
*
*   IF THE COMPONENT TO BE DELETED IS IN FACT AT THE EXTREME POSITION,
*   (I.E. CMPTN = MTDMSN) THEN CONTROL IS IMMEDIATELY RETURNED TO
*   'EDITIOR' AND THE MATRIX IS SIMPLE REDIMENSIONED
*
      IF(CMPT.EQ.MTDMSN) GO TO 175
*
*   MOVING THE COMPONENT ROW TO THE EXTREME POSITION
*
      DO 145 N = CMPTN, (MTDMSN-1)
        MTLBL$(N) = MTLBL$(N+1)
        DO 135 K = 1,MTDMSN
          MXCMPT(N,K) = MXCMPT((N+1),K)
  135   CONTINUE
  145 CONTINUE
*
*   MOVING THE ROW TO THE EXTREME POSITION
*
      DO 165 N = CMPTN, (MTDMSN-1)
        DO 155 K = 1, MTDMSN
          MXCMPT(K,N) = MXCMPT(K,(N+1))
  155   CONTINUE
  165 CONTINUE
*
*   SHOWING THE RE-ARRANGED ORDER OF THE COMPONENTS
*
  175 MSG$(1)= 'THE COMPONENTS ARE NOW NUMBERED AS FOLLOWS *'
      MSGN = 1
      SKIP = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
      WRITE(*,'(25(2(6X,I2,2X,A20:)/))') (N,MTLBL$(N), N = 1,(MTDMSN-1))
      MSG$(6) = 'REPEAT'
*
*   CONTROL RETURNED TO "EDITOR"
*
      RETURN
      END


*********************************************************************
      SUBROUTINE EDADTN(MXCMPT,MTLBL$,MSG$,MTDMSN,IMIN,IMAX,ERMSG$)
```

```
*
*   THIS ROUTINE ADDS A COMPONENT TO THE COMPONENT MATRIX
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY OF THE COMPONENT MATRIX ELEMENTS
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MSG$ - CHARACTER ARRAY OF UP TO SIX MESSAGE LINES FOR PRINTING TO
*            THE TERMINAL SCREEN BY 'MSGPRT'
*   MTDMSN - DIMENSION OF THE COMPONENT MATRIX
*   IMIN - INTEGER INDICATOR OF THE MINIMUM VALUE OF 'MXCMPT' (0)
*   IMAX - INTEGER INDICATOR OF THE MAXIMUM VALUE OF 'MXCMPT' (1 OR 3)
*   ERMSG$ - CHARACTER VARIABLE OF THE STANDARD ERROR MESSAGE
*
*   OTHER ARRAYS AND VARIABLES
*   MSGN - INTEGER COUNTER OF THE NUMBER OF MESSAGES IN 'MSG$' TO BE
*            PRINTED BY 'MSGPRT'
*   SKIP - INTEGER INDICATOR TO 'MSGPRT' TO PRINT THE USER PROMPT
*   REPLY$ - CHARACTER VARIABLE OF USER ANSWERS TO PROGRAMME QUESTIONS
*   CHECK$ - CHARACTER VARIABLE TO USER USER INPUT OF COMPONENT LABELS
*             TO ENSURE THAT THEY ARE NOT TOO LONG
*
      INTEGER MXCMPT(50,50),
     +         IMIN, IMAX,
     +        MSGN, SKIP
      CHARACTER MTLPL$*20(50), MSG$*60(6),
     +            ERMSG$*50,
     +            REPLY$*3, CHECK$*25
*
*   THE NEW COMPONENT IS TAGGED ON TO 'MXCMPT' AS AN EXTRA ROW AND
*   COLUMN (INDICATED AS 'NEXT')
*
      NEXT = MTDMSN+1
*
*   ENTERING THE COMPONENT LABEL AN INFORMING THE USER OF THE STATUS
*   OF THE NEW COMPONENT, AND COMPONENT MATRIX
*
      MSG$(1)= 'ENTER THE LABEL FOR THE NEW COMPONENT *'
      MSGN = 1
  105 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A20)', ERR = 105) MTLBL$(NEXT)
      PRINT*,' '
      WRITE(*, '(4X,'' * THE MATRIX HAS BEEN RE-DIMENSIONED TO '',I2,
     + '' X '',I2,''. *'',/,4X,'' * THE NEW COMPONENT IS NUMBER '',I2,
     + 17X,''*'')') NEXT ,NEXT, NEXT
*
*   ENTERING THE ELEMENTS OF THE NEW COMPONENT
*
      MSG$(1)= 'TO ENTER THE ELEMENTS OF THE COMPONENT MATRIX   *'
      IF(IMAX.EQ.1) THEN
        MSG$(2)= 'TYPE "1" FOR A DEPENDENCY BETWEEN COMPONENTS    *'
        MSG$(3)= 'TYPE "0" FOR NO INTERACTION BETWEEN COMPONENTS *'
        MSGN = 3
      ELSE
        MSG$(2)= 'TYPE IN AN INTEGER VALUE BETWEEN "0" AND "3" *'
        MSGN = 2
      END IF
      SKIP = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
*
*   ENTERING THE NEW COMPONENT ROW ELEMENTS
*
      DO 135 I = 1,NEXT
  115   PRINT*,' '
        WRITE(*, '(''    THE DEPENDENCY OF '',A20)') MTLBL$(I)
        WRITE(*, '(''                   ON '',A20)') MTLBL$(NEXT)
        PRINT*,'   ???... '
        READ(*, *, ERR = 125) MXCMPT(I,NEXT)
*
*   CHECKING THAT THE ELEMENT IS WITHIN BOUNDS
*
        IF((MXCMPT(I,NEXT).LT.IMIN).OR.(MXCMPT(I,NEXT).GT.IMAX)) THEN
  125     PRINT*,ERMSG$
          GO TO 115
        END IF
  135 CONTINUE
*
*   ENTERING THE NEW COMPONENT COLUMN
*
      DO 165 J = 1,MTDMSN
  145   PRINT*,' '
        WRITE(*, '(''    THE DEPENDENCY OF '',A20)') MTLBL$(NEXT)
        WRITE(*, '(''                   ON '',A20)') MTLBL$(J)
        PRINT*,'   ???... '
        READ(*, *, ERR = 155) MXCMPT(NEXT,J)
*
*   CHECKING IF THE ELEMENT IS WITHIN BOUNDS
*
        IF((MXCMPT(NEXT,J).LT.IMIN).OR.(MXCMPT(NEXT,J).GT.IMAX)) THEN
  155     PRINT*,ERMSG$
          GO TO 145
        END IF
  165 CONTINUE
*
*  CONTROL RETURNED TO "EDITOR"
*
      RETURN
      END

*****************************************************************
      SUBROUTINE MINLNK(MXCMPT,MXLINK,MXMINA,MXMIND,MTDMSN,NFILER)
*
*   THIS SUBROUTINE DERIVES    A) THE MINIMUM LINK MATRIX
*                              B) MINIMUM PATHS ARRAYS
*                              C) TRACKING INFORMATION
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY OF THE COMPONENT INTERACTION MATRIX DATA
*   MXLINK - INTEGER ARRAY FOR THE MINIMUM LINK MATRIX
*   MXMINA AND MXMIND - INTEGER ARRAYS FOR THE OUT- AND IN-MINIMUM
*                     PATHS TOTALS FOR EACH COMPONENT
*   MXMULT - INTEGER ARRAY FOR STORAGE DURING THE MATRIX MULTIPLICATION
*            PROCESS (STORES THE PRODUCT VALUES)
*   MXUNIT - INTEGER ARRAY FOR STORAGE DURING THE MATRIX MULTIPLICATION
*            PROCESS (STORES THE MATRIX TO BE MULTIPLIED WITH MXCMPT)
*   MXTRAK - INTEGER ARRAY FOR TEMPORARY STORAGE OF VALUES TO BE WRITTEN
*            TO THE DIRECT ACCESS FILE 13.  ONLY 31 RECORDS CAN BE WRITTEN
*            TO ANY ONE RECORD.
*   INDEX - INTEGER COUNTER OF THE NUMBER OF VALUES IN MXTRAK TO BE
*           WRITTEN TO FILE 13
*   PROD - INTEGER STORAGE FOR MATRIX MULTIPLICATION PRODUCTS
*   NFILER - INTEGER INDICATOR TO CONTROL ACCESS TO RECORDS IN THE DIRECT
*            ACCESS FILE 13
*   KILLER - INTEGER INDICATOR TO TERMINATE THE MATRIX POWERING PROCESS
*            WHEN THE MINIMUM LINK MATRIX HAS BEEN DERIVED
*   FILLER - INTEGER VARIABLE SET TO ZERO TO RECORD THE NO INFORMATION
*            SITUATION IN THE DIRECT ACCESS FILE 13
*   INDCTR - INTEGER INDICATOR TO REGISTER ONE OF TWO TYPES OF IRREGULAR
*            RECORDS IN THE DIRECT ACCESS FILE 13. INDCTR = -1 SHOWS THAT
*            NO TRACK EXISTS FOR THE COMPONENTS: AND INDCTR = -2 SHOWS
*            THE MAXIMUM RECORD LENGTH IN FILE 13 IS EXCEEDED FOR THAT
*            PARTICULAR RECORD.  THIS INFORMATION IS USED IN 'TRACKS'
*
*   FILES
*   UNIT 13 - DIRECT ACCESS FILE CREATED IN THE MAIN PROGRAMME TO STORE
*             INFORMATION USED IN THE 'TRACK' SUBROUTINE.
*             EACH RECORD IN FILE 13 RECORDS INFORMATION ABOUT THE
*             LINKAGE BETWEEN COMPONENT I AND J, WHERE THE RECORD NUMBER
*             IS GIVEN AS (MTDMSN(I-1) + J).  THE FIRST VALUE OF THE
*             EITHER INDICATES AN IRREGULAR RECORD, OR THE NUMBER OF
*             VALID VALUES STORED IN THE RECORD.
*
      INTEGER MXCMPT(50,50), MXLINK(50,50), MXMINA(50), MXMIND(50),
     +         MXMULT(50), MXUNIT(50,50), MXTRAK(31),
     +         INDEX, PROD, NFILER, KILLER, FILLER/0/, INDCTR
*
*   INITIALIZING THE STORAGE ARRAYS AND THE ARRAYS USED IN THE MATRIX
*   MULTIPLICATION PROCESS
*
      DO 105 I = 1,MTDMSN
        MXMINA(I) = 0
        MXMIND(I) = 0
        DO 105 J = 1,MTDMSN
          IF(MXCMPT(I,J).GT.0) THEN
*
*   MXUNIT, MXLINK ARE SET TO MXCMPT, AND ALL DIRECT DEPENDENCIES (I.E.
*   NON-ZERO VALUES IN MXCMPT ARE RECORDED IN THE APPROPRIATE RECORDS OF
*   FILE 13.  NOTE THAT FILE 13 RECORDS A MTDMSN X MTDMSN ARRAY IN A
*   COLUMN MAJOR FASHION.  TO ACCESS THE RECORD CORRESPONDING TO
*   ELEMENT(I,J) THE EQUATION (MTDMSN*(I-1) +J) IS USED.
```

```
*
            MXUNIT(I,J) = 1
            MXLINK(I,J) = 1
            INDEX = 1
            NFILER = ((MTDMSN*(I-1)) + J)
            WRITE (13'NFILER) INDEX, I
          ELSE
            MXUNIT(I,J) = 0
            MXLINK(I,J) = 0
          END IF
  105 CONTINUE
*
*  'LINK' INDICATES THE LEVEL OF POWERING OF THE COMPONENT INTERACTION
*  MATRIX, MXCMPT.
*  KILLER IS SET TO ZERO BEFORE EVERY NEW POWERING LEVEL.  IF KILLER
*  DOES NOT CHANGE VALUE TO 1, THE MINIMUM LINK MATRIX HAS BEEN DERIVED
*  I AND J CONTROL THE MATRIX MULTIPLICATION PROCEDURE
*  THE MULTIPLICATION PROCEDURE FOR ELEMENT (I,J) ONLY OCCURS IF
*  THE MINIMUM LINK ELEMENT(I,J) HAS NOT YET BEEN ALTERED FROM ZERO
*
      DO 165 LINK = 2,MTDMSN
        KILLER = 0
        DO 145 I = 1,MTDMSN
          DO 125 J = 1,MTDMSN
            IF(MXLINK(I,J).GT.0) GO TO 125
            MXMULT(J) = 0
            INDEX = 0
            DO 115 K = 1,MTDMSN
*
*  A NON-ZERO PRODUCT IN THE MULTIPLICATION PROCESS INDICATES A NEW
*  MINIMUM LINK PATH.  THE INFORMATION ABOUT THE PATH IS RECORDED IN
*  MXTRAK.
*
              IF((MXCMPT(K,J).GE.1).AND.(MXUNIT(I,K).GE.1)) THEN
                MXMULT(J) = MXMULT(J) + MXUNIT(I,K)
                INDEX = INDEX + 1
*
*  IF MORE THAN THIRTY MINIMUM PATHS HAVE BEEN FOUND, THE MAXIMUM
*  STORAGE IN FILE 13 IS EXCEEDED AND THE INDCTR VARIABLE IS USED
*  TO RECORD THIS CONDITION IN MXTRAK.
*
                IF(INDEX.GT.31) THEN
                  INDCTR = -2
                  NFILER = ((MTDMSN*(I-1)) + J)
                  WRITE(13'NFILER) INDCTR, FILLER, FILLER
                  GO TO 125
                END IF
                MXTRAK(INDEX) = K
              END IF
  115       CONTINUE
*
*  IF A NON-ZERO VALUE HAS BEEN DERIVED IN THE POWERING PROCESS FOR
*  MXUNIT(I,J) THEN ALL THE INFORMATION ABOUT THE MINIMUM PATH/S
*  BETWEEN COMPONENT I AND J IS WRITTEN FROM MXTRAK TO FILE 13, AND
*  THE LENGTH OF THE MINIMUM PATH BETWEEN COMPONENT I AND J (LINK) IS
*  RECORDED IN THE MINIMUM LINK MATRIX, MXLINK.
*
            IF(MXMULT(J).GT.0) THEN
              NFILER = ((MTDMSN*(I-1)) + J)
              WRITE(13'NFILER) INDEX, (MXTRAK(N), N = 1, INDEX)
              MXLINK(I,J) = LINK
*
*  KILLER IS SET TO 1 TO INDICATE THAT SOME CHANGE TO MXLINK HAS OCCURRED
*  DURING THE POWERING PROCESS
*
              KILLER = 1
            END IF
  125     CONTINUE
*
*  MXUNIT IS INITIALIZED FOR THE NEXT LEVEL OF POWERING TO MXCMPT
*  NOTE THAT THE VALUE IN MXUNIT IS ONLY CHANGED ONCE; FROM ZERO TO
*  THE FIRST NON-ZERO VALUE THAT OCCURS.  THEREFORE THE FINAL VALUES
*  IN MXUNIT REPRESENT THE MINIMUM PATH INFORMATION FOR MXMINA AND MXMIND
*
          DO 135 J = 1, MTDMSN
            IF(MXUNIT(I,J).EQ.0) THEN
              MXUNIT(I,J) = MXMULT(J)
            END IF
  135     CONTINUE
  145   CONTINUE
*
*  IF THE MINIMUM LINK HAS BEEN COMPLETELY DERIVED, THE VALUES FOR
*  THE TOTAL MINIMUM PATHS ASSOCIATED WITH EACH COMPONENT
*  ARE WRITTEN TO MXMINA AND MXIND FROM MXUNIT, AND THE FILE 13
*  RECORDS WHICH HAVE NOT BEEN WRITTEN TO ARE MARKED ACCORDINGLY (USING
*  INDCTR = -1).
*
        IF((KILLER.EQ.0).OR.(LINK.EQ.MTDMSN)) THEN
          INDCTR = -1
          DO 155 I = 1, MTDMSN
            DO 155 J = 1, MTDMSN
              IF(MXLINK(I,J).EQ.0) THEN
                NFILER = ((MTDMSN*(I-1)) + J)
                WRITE(13'NFILER) INDCTR, FILLER
              END IF
              MXMINA(I) = MXMINA(I) + MXUNIT(I,J)
              MXMIND(I) = MXMIND(I) + MXUNIT(J,I)
  155     CONTINUE
          GO TO 175
        END IF
  165 CONTINUE
*
*  CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
  175 RETURN
      END


*******************************************************************
      SUBROUTINE TRACKS(MXLINK,MTLBL$,MTDMSN,NTRACK,NFILER,HDGEND,
     +HDG$,DATE$,ERMSG$)
*
*  THIS ROUTINE EXTRACTS INFORMATION FROM FILE 13 TO PERFORM TRACKS OF
*  DEPENDENCE BETWEEN A PAIR COMPONENTS SPECIFIED BY THE USER
*
*  ARGUMENTS
*  MXLINK - INTEGER ARRAY REPRESENTING THE MINIMUM LINK MATRIX
*  MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*  MTDMSN - THE DIMENSION OF THE COMPONENT INTERACTION MATRIX AND OTHER
*           RELATED ARRAYS
*  NTRACK - INTEGER COUNTER OF THE NUMBER OF TRACKS PERFORMED
*  NFILER - INTEGER POINTER ASSOCIATED WITH THE DIRECT ACCESS FILE 13
*  HDGEND - INTEGER COUNTER OF THE NUMBER OF CHARACTERS IN HDG$
*  HDG$ - CHARACTER VARAIBLE OF A USER SUPPLIED HEADING FOR THE HARDCOPY
*         PRINTOUT OF THE PROGRAMME OUTPUT
*  DATE$ - CHARACTER VARIABLE FOR THE DATE OBTAINED BY A CALL TO SYSTEM
*          FUNCTION 'ADATE$' USED FOR HEADING IN THE PLOTTING PROGRAMME
*  ERMSG$ - CHARACTER VARIABLE OF THE STANDARD ERROR MESSAGE
*
*  OTHER ARRAYS AND VARIABLES
*  MXUPPR - INTEGER ARRAY : A "1" IN ANY ELEMENT OF THE ARRAY INDICATES
*           THAT THE COMPONENT WITH THE SAME NUMBER AS THE ELEMENT NUMBER
*           SUPPORTS A LINK IN THE TRACK FOR THE STEP BEING PROCESSED
*  MXLOWR - INTEGER ARRAY : A "1" IN ANY ELEMENT OF THE ARRAY INDICATES
*           THAT THE COMPONENT WITH THE SAME NUMBER AS THE ELEMENT NUMBER
*           TERMINATES A LINK ORIGINATED BY SOME COMPONENT IN MXUPPR
*  MXTNFM - INTEGER ARRAY OF THE COMPONENTS (BY COMPONENT NUMBER) WHICH
*           APPEAR IN THE TRACK (USED AS INFO FOR THE PLOTTING PROGRAMME)
*  MXTRAK - INTEGER ARRAY TO STORE THE DATA READ FROM FILE 13
*  STEP - INTEGER COUNTER OF THE NO OF LINKS TO BE PROCESSED IN THE TRACK
*  TSTEP - INTEGER INDICATOR OF THE MAXIMUM NUMBER OF LINKS IN THE TRACK
*          (READ FROM MXLINK(I,J))
*  TNODE - THE NUMBER OF COMPONENTS WHICH APPEAR IN THE TRACK I.E. THE
*          NUMBER OF VALUES IN MXTNFM
*  PA - INTEGER COUNTER FOR FINDING THE ORIGINATING COMPONENTS OF A LINK
*  NB - THE NUMBER OF COMPONENTS WHICH TERMINATE ANY LINK ORIGINATED BY
*       A COMPONENT IN MXUPPR (READ AS FIRST VALUE IN RECORD OF FILE 13)
*  Q - INTEGER COUNTER OF THE NUMBER OF COMPONENT NUMBERS TO BE READ FROM
*      A RECORD IN FILE 13 (I.E. THE COMPONENTS WHICH TERMINATE LINKS
*      SPECIFIED BY MXUPPR)
*  PB - INTEGER INDICATOR FOR PROCESSING EACH COMPONENT SPECIFIED IN MXTRAK
*  MSGN - THE NUMBER OF PROGRAMME SUPPLIED MESSAGES IN MSG$ TO BE PRINTED
*         TO THE TERMINAL SCREEN BY "MSGPRT"
*  SKIP - INTEGER INDICATOR TO "MSGPRT" TO PRINT THE USER PROMPT
*  REPLY$ - CHARACTER VARIABLE FOR USER REPLIES TO PROGRAMME QUESTIONS
*
*  FILES
*  UNIT 3 - A DATAFILE CREATED BY THE PROGRAMME (CALLED PLOTINFO) USED TO
*           SUPPLY THE PLOTTING PROGRAMME "ABSTRACK", WHICH IS STARTED BY
*           "RUNFIN", WITH PLOTTING INFORMATION (HEADINGS, DATE, ETC).
```

```
*   UNIT 4 - A DATAFILE CREATED BY THE PROGRAMME (CALLED PLOTDATA) USED TO
*            SUPPLY THE PLOTTING PROGRAMME WITH THE DATA REQUIRED TO PLOT
*            A REPRESENTATION OF ALL THE VALID TRACKS PERFORMED BY THE USER.
*   UNIT 13 - DIRECT ACCESS FILE CREATED IN THE MAIN PROGRAMME TO STORE
*            INFORMATION USED IN THE 'TRACK' SUBROUTINE.
*
      INTEGER MXLINK(50,50),
     +        MXUPPR(50), MXLOWR(50), MXTNFM(50), MXTRAK(31),
     +        MTDMSN, NTRACK, NFILER, HDGEND,
     +        STEP, TSTEP, TNODE, PA, NB, B, PB, MSGN, SKIP
      CHARACTER MTLBL$*20(50), MSG$*60(6),
     +          HDG$*55, DATE$*8, ERMSG$*50,
     +          REPLY$*3
*
*   INFORMATION ON EACH TRACK IS WRITTEN TO THE FILES 3 AND 4, AND THEN
*   THE COMPLETE SET IS PLOTTED BY THE STARTED PLOTTING PROGRAMME "ABSTRAK".
*   (STARTED IN "RUNFIN").
*
      MSG$(1)= 'ALL TRACKS ARE DRAWN AS DIAGRAMS ON THE PLOTTING   *'
      MSG$(2)= 'MACHINE AND ARE RETURNED TO YOU UNDER YOUR RUN-ID. *'
      MSGN = 2
      SKIP = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
*
*   ENTERING THE TRACK INFORMATION
*
  105 MSG$(1)= 'TYPE THE NUMBER OF THE DEPENDENT COMPONENT *'
      MSGN = 1
      CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, *, ERR = 115) I
      IF((I.GT.MTDMSN).OR.(I.LE.0)) THEN
  115   PRINT*,ERMSG$
        GO TO 105
      END IF
      MSG$(1)= 'TYPE THE NUMBER OF THE SUPPORTING COMPONENT *'
      MSGN = 1
  125 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, *, ERR = 135) J
      IF((J.GT.MTDMSN).OR.(J.LE.0)) THEN
  135   PRINT*,ERMSG$
        GO TO 125
      END IF
*
*   CHECKING IF A TRACK CAN BE PERFORMED
*
      IF(MXLINK(I,J).EQ.0) THEN
        WRITE(*, '('' **** THERE IS NO INTERACTION BETWEEN '',
     +  ''COMPONENT '',I2,'' AND COMPONENT '',I2,''.'',/,
     +  '' **** THEREFORE A TRACK CANNOT BE PERFORMED.'')') I, J
        GO TO 215
      END IF
*
*   INITIALIZING THE INTERNAL ARRAYS FOR A NEW TRACK
*
      DO 145 K = 1,MTDMSN
        MXUPPR(K) = 0
        MXLOWR(K) = 0
        MXTNFM(K) = 0
  145 CONTINUE
*
*   NUMBER OF LINKS IN THE TRACK IS READ FROM MXLINK INTO TSTEP
*   THE TRACK IS TRACED FROM THE SUPPORTING TO THE DEPENDENT COMPONENT
*
      TSTEP = MXLINK(I,J)
      MXUPPR(J) = 1
      MXTNFM(J) = 1
*
*   THE LINK COUNTER
*
      DO 195 STEP = 1,TSTEP
*
*   FINDING ALL COMPONENTS WHICH ORIGINATE A LINK FOR THE CURRENT STEP
*
        DO 175 PA = 1,MTDMSN
          IF(MXUPPR(PA).EQ.1) THEN
*
*   THE TERMINATING COMPONENTS RELATED TO ANY ORIGINATING COMPONENT ARE
*   READ FROM FILE 13 INTO MXTRAK
*
            NFILER = ((MTDMSN*(I-1)) + PA)
            READ(13'NFILER, ERR = 155) NB, (MXTRAK(N), N = 1, NB, +1)
*
*   THE PROGRAMME CAN ONLY HANDLE THIRTY TERMINATING COMPONENTS FOR A
*   ANY ORIGINATING COMPONENT (LIMITATION IS WITHIN FILE 13 AND
*   AND IS RECORDED AS A "-2" IN THE APPROPRIATE RECORD
*
  155       IF(NB.EQ.-2) THEN
              WRITE(*, '('' **** THERE ARE MORE THAN THIRTY SHORTEST  '',
     +        ''PATHS BETWEEN COMPONENT '',I2,'' AND COMPONENT '',I2,
     +        ''.'',/,'' **** THEREFORE NO TRACK WILL BE PERFORMED.'')')
     +        I, J
              GO TO 215
            END IF
*
*   PROCESSING EACH LINK BETWEEN THE ORIGINATING AND TERMINATING COMPONENTS
*   OF THE CURRENT STEP.  INFORMATION IS WRITTEN TO THE PLOTFILE 4.
*
            DO 165 B = 1, NB
              PB = MXTRAK(B)
              MXLOWR(PB) = 1
              MXTNFM(PB) = 1
              WRITE(4, '(2X,I2,2X,I2)') PA, STEP
              WRITE(4, '(2X,I2,2X,I2)') PB, (STEP+1)
  165       CONTINUE
          END IF
  175   CONTINUE
*
*   ALL TERMINATING COMPONENTS ARE SET TO ORIGINATING COMPONENTS
*   FOR THE NEXT STEP
*
        DO 185 K = 1,MTDMSN
          MXUPPR(K) = MXLOWR(K)
          MXLOWR(K) = 0
  185   CONTINUE
  195 CONTINUE
*
*   WRITING PLOT INFORMATION TO FILE 3
*
      DO 205 K = 1,MTDMSN
*
*   ONLY THE COMPONENTS WHICH APPEAR IN THE TRACK ARE RECORDED ON FILE 3
*
        IF(MXTNFM(K).EQ.1) THEN
          WRITE(3, '(2X,I2,2X,A20)') K, MTLBL$(K)
        END IF
  205 CONTINUE
      NTRACK = NTRACK + 1
      WRITE(3, '(2X,A24)') '99   (RANKING STOP CARD)'
      WRITE(3, '(2X,I2)') TSTEP
      WRITE(3, '(2X,A8,2X,A50,2X,I2)') DATE$, HDG$, HDGEND
      WRITE(3, '(2X,I2,2X,A20,2X,I2,2X,A20)')
     +I, MTLBL$(I), J, MTLBL$(J)
      WRITE(3, '(2X,A13,I2)') 'TRACK NUMBER ', NTRACK
      WRITE(4, '(2X,A13,I2)') 'END OF TRACK ', NTRACK
*
*   OFFERING THE REPEAT OR RETURN TO MAIN PROGRAMME OPTION
*
  215 MSG$(1)= 'DO YOU WISH TO PERFORM FURTHER TRACKS ?  (YES/NO) *'
      MSGN = 1
  225 CALL MSGPRT(MSG$,MSGN,SKIP)
      READ(*, '(A3)') REPLY$
      IF(REPLY$.EQ.'YES') THEN
        GO TO 105
      ELSE IF(REPLY$.NE.'NO') THEN
        PRINT*,ERMSG$
        GO TO 225
      END IF
      RETURN
*
*   CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
      END

******************************************************************
      SUBROUTINE VALNCY(MXCMPT,MTLBL$,MTDMSN,IMAX)
*
*   THIS ROUTINE COMPUTES AND PRINTS THE IN- AND OUT- VALENCIES
*   OF ALL THE COMPONENTS IN RANKED ORDER
*
```

```
*  ARGUMENTS
*  MXCMPT - INTEGER ARRAY OF THE COMPONENT INTERACTION MATRIX DATA
*  MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*  MTDMSN - INTEGER DIMENSION VALUE OF THE COMPONENT MATRIX AND RELATED
*           ARRAYS
*  IMAX - INTEGER INDICATOR OF THE MAXIMUM VALUE IN MXCMPT, USED IN
*         THIS ROUTINE TO INDICATE WHETHER A WEIGHTED COMPONENT MATRIX
*         IS BEING USED
*
*  OTHER ARRAYS AND VARIABLES
*  MXORDA - INTEGER ARRAY TO SORT THE OUT-VALENCY VALUES INTO RANKED ORDER
*  MXORDD - INTEGER ARRAY TO SORT THE IN-VALENCY VALUES INTO RANKED ORDER
*  MXA1,2,3 - INTEGER ARRAYS TO STORE THE OUT-VALENCY VALUES. MXA1 USED
*             FOR UNWEIGHTED MATRICES, ALL USED FOR WEIGHTED MATRICES.
*  MXD1,2,3 - AS ABOVE FOR THE IN-VALENCIES
*  SWOP - INTEGER STORAGE USED IN THE SORTING PROCEDURE
*  UNLN$ - CHARACTER VARIABLE USED TO UNDERLINE THE PRINT HEADINGS
*
*  FILES
*  UNIT 2 - ALTERNATE PRINTFILE FOR HARDCOPY OUTPUT TO USER
*
      INTEGER MXCMPT(50,50),
     +          MXORDA(50), MXORDD(50),
     +          MXA1(50), MXA2(50), MXA3(50),
     +          MXD1(50), MXD2(50), MXD3(50),
     +          MTDMSN, IMAX,
     +          SWOP
      CHARACTER MTLBL$*20(50),
     +             UNLN$*5/'-----'/
*
*  INITIALIZING ALL THE STORAGE AND SORT ARRAYS
*
      DO 105 K = 1, MTDMSN
        MXORDA(K) = K
        MXORDD(K) = K
        MXA1(K) = 0
        MXD1(K) = 0
        MXA2(K) = 0
        MXD2(K) = 0
        MXA3(K) = 0
        MXD3(K) = 0
  105 CONTINUE
*
*  COMPUTING THE IN- AND OUT-VALENCIES
*
      DO 125 I = 1, MTDMSN
        DO 115 J = 1, MTDMSN
          IF(MXCMPT(I,J).EQ.1) THEN
            MXA1(I) = MXA1(I) + 1
            MXD1(J) = MXD1(J) + 1
          ELSE IF(MXCMPT(I,J).EQ.2) THEN
            MXA2(I) = MXA2(I) + 1
            MXD2(J) = MXD2(J) + 1
          ELSE IF(MXCMPT(I,J).EQ.3) THEN
            MXA3(I) = MXA3(I) + 1
            MXD3(J) = MXD3(J) + 1
          END IF
  115   CONTINUE
  125 CONTINUE
*
*  IF AN UNWEIGHTED COMPONENT MATRIX IS BEING USED, MXA1 IS SET TO
*  MXA3 TO CONFORM WITH THE SORTING AND PRINTING PROCEDURES
*
      IF(IMAX.EQ.1) THEN
        DO 135 K = 1, MTDMSN
          MXA3(K) = MXA1(K)
          MXD3(K) = MXD1(K)
  135   CONTINUE
      END IF
*
*  THE SORTING ROUTINE.  IN- AND OUT-VALENCY TOTALS ARE SORTED INTO
*  RANK ORDER AND THE COMPONENT NUMBERS ARE ADJUSTED (IN MXORD ARRAYS)
*  SO THAT THE THE COMPONENT ORDER IS RECORDED.  THE MXA3 ARRAY IS USED
*  FOR THE SORT.  FOR WEIGHTED MATRICES, THE OTHER ARRAYS (MXA/D 2&1)
*  ARE NOT CONSIDERED.
*
      DO 155 I = 1, (MTDMSN-1)
        DO 145 J = (I+1), MTDMSN
          IF(MXA3(I).LT.MXA3(J)) THEN
            SWOP = MXORDA(I)
            MXORDA(I) = MXORDA(J)
            MXORDA(J) = SWOP
            SWOP = MXA3(I)
            MXA3(I) = MXA3(J)
            MXA3(J) = SWOP
          END IF
          IF(MXD3(I).LT.MXD3(J)) THEN
            SWOP = MXORDD(I)
            MXORDD(I) = MXORDD(J)
            MXORDD(J) = SWOP
            SWOP = MXD3(I)
            MXD3(I) = MXD3(J)
            MXD3(J) = SWOP
          END IF
  145   CONTINUE
  155 CONTINUE
*
*  THE PRINTING ROUTINE.  WEIGHTED AND UNWEIGHTED COMPONENT MATRICES
*  ARE PRINTED IN DIFFERENT FORMATS.  IMAX IS USED TO CHECK WHICH FORM
*  IS TO BE USED.
*
      IF(IMAX.EQ.1) THEN
*
*  UNWEIGHTED COMPONENT INTERACTION MATRICES
*
        WRITE(2, '(1H1,/,6X,A,5X,A,/,6X,10(A5),A2,5X,4(A5),A1)')
     +  'THE NUMBER OF COMPONENTS EACH COMPONENT DEPENDS UPON',
     +  '(OUT-VALENCY RANKING)',
     +  (UNLN$, N = 1,10), '--', (UNLN$, N=1,4), '-'
        WRITE(2, '(//,6X,A4,5X,A10,2X,A9,2X,A9)')
     +  'RANK', 'COMPONENTS', 'COMPONENT', 'COMPONENT'
        WRITE(2, '(12X,A13,2X,A9,6X,A5)')
     +  'DEPENDED UPON', 'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A13,2X,A9,2X,A9)')
     +  '----', '-------------', '---------', '---------'
        DO 165 K = 1, MTDMSN
          WRITE(2, '(7X,I2,8X,I2,11X,I2,6X,A20)')
     +    K, MXA3(K), MXORDA(K), MTLBL$(MXORDA(K))
  165   CONTINUE
        WRITE(2, '(//,7X,A10,I2)') 'MAXIMUM = ', MTDMSN
        WRITE(2, '(1H1,/,6X,A,5X,A,/,6X,10(A5),A4,5X,4(A5))')
     +  'THE NUMBER OF DEPENDENCIES SUPPORTED BY EACH COMPONENT',
     +  '(IN-VALENCY RANKING)',
     +  (UNLN$, N=1,10), '----', (UNLN$, N=1,4)
        WRITE(2, '(//,6X,A4,2X,A12,2X,A9,2X,A9)')
     +  'RANK', 'DEPENDENCIES', 'COMPONENT', 'COMPONENT'
        WRITE(2, '(14X,A9,6X,A6,5X,A5)')
     +  'SUPPORTED', 'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A12,2X,A9,2X,A9)')
     +  '----', '------------', '---------', '---------'
        DO 175 K = 1, MTDMSN
          WRITE(2, '(7X,I2,8X,I2,10X,I2,6X,A20)')
     +    K, MXD3(K), MXORDD(K), MTLBL$(MXORDD(K))
  175   CONTINUE
        WRITE(2, '(//,7X,A10,I2)') 'MAXIMUM = ', MTDMSN
      ELSE
*
* WEIGHTED COMPONENT INTERACTION MATRICES
*
        WRITE(2, '(1H1,/,6X,A,5X,A,/,6X,10(A5),A2,5X,4(A5),A1)')
     +  'THE NUMBER OF COMPONENTS EACH COMPONENT DEPENDS UPON',
     +  '(OUT-VALENCY RANKING)',
     +  (UNLN$, N = 1,10), '--', (UNLN$, N=1,4), '-'
        WRITE(2, '(//,6X,A4,3X,A11,2X,A15,2X,A12,2X,A9,2X,A9)')
     +  'RANK', 'LIMITED (1)', 'APPRECIABLE (2)', 'COMPLETE (3)',
     +  'COMPONENT', 'COMPONENT'
        WRITE(2, '(12X,A12,5X,A12,2X,A12,5X,A6,6X,A5)')
     +  'DEPENDENCIES', 'DEPENDENCIES', 'DEPENDENCIES',
     +  'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A12,2X,A15,2X,A12,2X,A9,2X,A9)')
     +  '----', '------------', '---------------', '------------',
     +  '---------', '---------'
        DO 185 K = 1, MTDMSN
          WRITE(2, '(7X,I2,9X,I2,13X,I2,14X,I2,11X,I2,5X,A20)')
     +    K, MXD1(MXORDD(K)), MXD2(MXORDD(K)), MXD3(K),
     +    MXORDD(K), MTLBL$(MXORDD(K))
  185   CONTINUE
        WRITE(2, '(/,7X,A10,I2,13X,I2,14X,I2)') 'MAXIMUM = ',
     +  MTDMSN, MTDMSN, MTDMSN
        WRITE(2, '(/,6X,A)') ' **** THE COMPONENTS ARE RANKED IN '//
```

```
     +  'ORDER OF THE NUMBER OF COMPLETE (3) DEPENDENCIES'
        WRITE(2, '(1H1,/,6X,A,5X,A,/,6X,10(A5),A4,5X,4(A5))')
     +  'THE NUMBER OF DEPENDENCIES SUPPORTED BY EACH COMPONENT',
     +  '(IN-VALENCY RANKING)',
     +  (UNLN$, N=1,10), '----', (UNLN$, N=1,4)
        WRITE(2, '(//,6X,A4,3X,A11,2X,A15,2X,A12,2X,A9,2X,A9)')
     +  'RANK', 'LIMITED (1)', 'APPRECIABLE (2)', 'COMPLETE (3)',
     +  'COMPONENT', 'COMPONENT'
        WRITE(2, '(12X,A12,5X,A12,2X,A12,5X,A6,6X,A5)')
     +  'DEPENDENCIES', 'DEPENDENCIES', 'DEPENDENCIES',
     +  'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A12,2X,A15,2X,A12,2X,A9,2X,A9)')
     +  '----', '------------', '---------------', '------------',
     +  '---------', '---------'
        DO 195 K = 1, MTDMSN
          WRITE(2, '(7X,I2,9X,I2,13X,I2,14X,I2,11X,I2,5X,A20)')
     +    K, MXA1(MXORDA(K)), MXA2(MXORDA(K)), MXA3(K),
     +    MXORDA(K), MTLBL$(MXORDA(K))
  195   CONTINUE
        WRITE(2, '(/,7X,A10,I2,13X,I2,14X,I2)') 'MAXIMUM = ',
     +  MTDMSN, MTDMSN, MTDMSN
        WRITE(2, '(//,6X,A)') ' **** THE COMPONENTS ARE RANKED IN '//
     +  'ORDER OF THE NUMBER OF COMPLETE (3) DEPENDENCIES'
      END IF
*
*   CONTROL RETURNED TO MAIN PROGRAMME
*
      RETURN
      END


**********************************************************************
      SUBROUTINE DSTANS(MXLINK,MTLBL$,MTDMSN)
*
*   THIS ROUTINE COMPUTES THE OUT- AND IN-AVERAGE GRAPH DISTANCES
*   OF THE SYSTEM REPRESENTED IN THE COMPONENT INTERACTION MATRIX
*   FROM THE VALUES IN THE MINIMUM LINK MATRIX
*
*   ARGUMENTS
*   MXLINK - INTEGER ARRAY OF THE MINIMUM LINK MATRIX, DERIVED IN THE
*            'MINLNK' SUBROUTINE
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MTDMSN - INTEGER VALUE OF THE MINIMUM LINK MATRIX DIMENSION AND
*            THE DIMENSION OF OTHER RELATED ARRAYS
*
*   OTHER ARRAYS AND VARIABLES
*   MXA AND MXD - REAL ARRAYS USED TO STORE THE OUT- AND IN-AVERAGE GRAPH
*                 DISTANCES RESPECTIVELY
*   NA AND ND - REAL COUNTERS OF THE TOTAL NUMBER OF VALUES OVER WHICH
*               THE AVERAGES ARE COMPUTED (I.E. THE NUMBER OF NON-ZERO
*               VALUES TO THE RESPECTIVE SUMMATIONS)
*   SWOP - REAL STORAGE FOR THE SORTING PROCEDURE
*   MXORDA AND MXORDD - INTEGER ARRAYS FOR RECORDING THE RANKED ORDER OF
*                       OF COMPONENTS WHEN SORTED BY OUT- AND IN-AVERAGE
*                       GRAPH DISTANCES RESPECTIVELY
*   UNLN$ - CHARACTER VARIABLE TO UNDERLINE HEADINGS IN THE ALTERNATE
*           PRINTFILE (UNIT 2)
*
*   FILES
*   UNIT 2 - ALTERNATE PRINFILE FOR HARDCOPY OUTPUT TO THE USER
*
      REAL MXA(50), MXD(50),
     +     NA, ND, SWOP
      INTEGER MXLINK(50,50),
     +        MXORDA(50), MXORDD(50),
     +        MTDMSN
      CHARACTER MTLBL$*20(50),
     +          UNLN$*5/'-----'/
*
*   INITIALIZING THE SORT AND STORAGE ARRAYS
*
      DO 105 N = 1, MTDMSN
        MXA(N) = 0.0
        MXD(N) = 0.0
        MXORDA(N) = N
        MXORDD(N) = N
  105 CONTINUE
*
*   COMPUTING THE AVERAGE GRAPH DISTANCES
*
      DO 125 I = 1, MTDMSN
        NA = 0.0
        ND = 0.0
        DO 115 J = 1, MTDMSN
          IF(MXLINK(I,J).GT.0) THEN
            NA = NA + 1.0
            MXA(I) = MXA(I) + FLOAT(MXLINK(I,J))
          END IF
          IF(MXLINK(J,I).GT.0) THEN
            ND = ND + 1.0
            MXD(I) = MXD(I) + FLOAT(MXLINK(J,I))
          END IF
  115   CONTINUE
*
*   ENSURING THERE IS NO DIVIDE OVERFLOW IF NA OR ND IS ZERO
*
        IF(NA.GT.0.0) MXA(I) = (MXA(I)/NA)
        IF(ND.GT.0.0) MXD(I) = (MXD(I)/ND)
  125 CONTINUE
*
*   THE SORTING PROCEDURE
*
      DO 145 I = 1, (MTDMSN-1)
        DO 135 J = (I+1), MTDMSN
          IF(MXA(I).LT.MXA(J)) THEN
            SWOP = MXA(I)
            MXA(I) = MXA(J)
            MXA(J) = SWOP
            SWOP = MXORDA(I)
            MXORDA(I) = MXORDA(J)
            MXORDA(J) = SWOP
          END IF
          IF(MXD(I).LT.MXD(J)) THEN
            SWOP = MXD(I)
            MXD(I) = MXD(J)
            MXD(J) = SWOP
            SWOP = MXORDD(I)
            MXORDD(I) = MXORDD(J)
            MXORDD(J) = SWOP
          END IF
  135   CONTINUE
  145 CONTINUE
*
*   THE PRINTING PROCEDURE (TO UNIT 2)
*
      WRITE(2, '(1H1,/,6X,A,/,6X,A,6X,A,/,6X,10(A5),5X,7(A5),A1)')
     +'THE AVERAGE NUMBER OF LINKS PER CONNECTION BETWEEN',
     +'A COMPONENT AND OTHERS UPON WHICH IT IS DEPENDENT',
     +'(AVERAGE OUT-GRAPH DISTANCE RANKING)',
     +(UNLN$, N = 1, 10), (UNLN$, N=1,7), '-'
      WRITE(2, '(//,6X,A4,2X,A8,2X,A9,2X,A9)')
     +'RANK', 'AVE. NO.', 'COMPONENT', 'COMPONENT'
      WRITE(2, '(12X,A8,5X,A6,6X,A5)')
     +'OF LINKS', 'NUMBER', 'LABEL'
      WRITE(2, '(6X,A4,2X,A8,2X,A9,2X,A9)')
     +'----', '--------', '---------', '---------'
      DO 155 K = 1, MTDMSN
        WRITE(2, '(7X,I2,4X,F4.1,9X,I2,5X,A20)')
     +  K, MXA(K), MXORDA(K), MTLBL$(MXORDA(K))
  155 CONTINUE
      NA = 0.5*(FLOAT(MTDMSN) + 1.0)
      WRITE(2, '(//,3X,A10,F4.1)') 'MAXIMUM = ', NA
      WRITE(2, '(1H1,/,6X,A,/,6X,A,7X,A,/,6X,10(A5),5X,7(A5))')
     +'THE AVERAGE NUMBER OF LINKS PER CONNECTION BETWEEN',
     +'A COMPONENT AND OTHERS WHICH ARE DEPENDENT ON IT',
     +'(AVERAGE IN-GRAPH DISTANCE RANKING)',
     +(UNLN$, N = 1, 10), (UNLN$, N=1,7)
      WRITE(2, '(//,6X,A4,2X,A8,2X,A9,2X,A9)')
     +'RANK', 'AVE. NO.', 'COMPONENT', 'COMPONENT'
      WRITE(2, '(12X,A8,5X,A6,6X,A5)')
     +'OF LINKS', 'NUMBER', 'LABEL'
      WRITE(2, '(6X,A4,2X,A8,2X,A9,2X,A9)')
     +'----', '--------', '---------', '---------'
      DO 165 K = 1, MTDMSN
        WRITE(2, '(7X,I2,4X,F4.1,9X,I2,5X,A20)')
     +  K, MXD(K), MXORDD(K), MTLBL$(MXORDD(K))
  165 CONTINUE
      NA = 0.5*(FLOAT(MTDMSN) + 1.0)
      WRITE(2, '(//,3X,A10,F4.1)') 'MAXIMUM = ', NA
*
```

```
*   CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
        RETURN
        END


*****************************************************************
        SUBROUTINE MNPATH(MXMINA,MXMIND,MTLBL$,MTDMSN)
*
*   THIS ROUTINE RANKS AND PRINTS THE NUMBER OF MINIMUM PATHS ASSOCIATED
*   WITH EACH COMPONENT
*
*   ARGUMENTS
*   MXMINA - INTEGER ARRAY OF THE NUMBES OF MINIMUM PATHS CONNECTING
*            A COMPONENT TO OTHERS ON WHICH IT IS DEPENDENT.  ARRAY
*            VALULS SET BY 'MINLNK'.
*   MXMIND - SIMILAR TO MXMINA.  THE NUMBER OF MINIMUM PATHS LINKING A
*            COMPONENT TO OTHERS WHICH ARE DEPENDENT ON IT.
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS.
*   MTDMSN - INTEGER VALUE OF THE COMPONENT INTERACTION MATRIX DIMENSION
*            AND THE DIMENSION OF ASSOCIATED ARRAYS
*
*   OTHER ARRAYS AND VARIABLES
*   MXOPDA - INTEGER ARRAY INTO WHICH MXMINA IS RANKED
*   MXOPDD - INTEGER ARRAY INTO WHICH MXMIND IS RANKED
*   SWOP - INTEGER VARIABLE FOR TEMPORARY STORAGE OF DATA IN SORTING
*   UNLN$ - CHARACTER VARIABLE FOR UNDER-LINING HEADINGS ON PRINTOUT
*
*   FILES
*   UNIT 2 - ALTERNATE PRINTFILE FOR PROGRAMME OUTPUT TO USER
*
        INTEGER MXMINA(50), MXMIND(50),
     +          MXORDA(50), MXORDD(50),
     +          SWOP, MTDMSN
        CHARACTER MTLBL$*20(50),
     +            UNLN$*5/'-----'/
*
*   INITIALIZING THE RANKING ARRAYS
*
        DO 105 N = 1, MTDMSN
          MXORDA(N) = N
          MXORDD(N) = N
  105   CONTINUE
*
*   THE RANKING PROCEDURE
*   BOTH MXMINA AND MXMIND ARE RANKED IN DESCENDING ORDER, AND THE
*   CORRESPONDING ARRAYS, MXORDA AND MXORDD ARE ALSO REARRANGED SO THAT
*   THE CORRECT NUMBERS AND LABELS CAN BE ASSOCIATED WITH THE RANKED
*   VALUES IN THE PRINTOUT.
*
        DO 115 I = 1, (MTDMSN-1)
          DO 115 J = (I+1), MTDMSN
            IF(MXMINA(I).LT.MXMINA(J)) THEN
              SWOP = MXMINA(I)
              MXMINA(I) = MXMINA(J)
              MXMINA(J) = SWOP
              SWOP = MXORDA(I)
              MXORDA(I) = MXORDA(J)
              MXORDA(J) = SWOP
            END IF
            IF(MXMIND(I).LT.MXMIND(J)) THEN
              SWOP = MXMIND(I)
              MXMIND(I) = MXMIND(J)
              MXMIND(J) = SWOP
              SWOP = MXORDD(I)
              MXORDD(I) = MXORDD(J)
              MXORDD(J) = SWOP
            END IF
  115   CONTINUE
*
*   THE RANKED ARRAYS ARE WRITTEN TO THE ALTERNATE PRINTFILE UNIT 2
*   WHICH IS THE ONLY OUTPUT OF THE RANKING TO THE USER.  THE FILE
*   IS SYMTED TO THE PRINTER BY 'RUNFIN'.
*
        WRITE(2, '(1H1,/,6X,A,/,6X,A,5X,A,/,6X,8(A5),A4,5X,8(A5))')
       +'THE NUMBER OF MINIMUM PATHS LINKING A COMP-',
       +'ONENT TO ALL OTHERS ON WHICH IT IS DEPENDENT',
       +'(OUT-MINIMUM PATH ACCESSIBILITY RANKING)',
       +(UNLN$, N = 1, 8), '----', (UNLN$, N=1,8)
        WRITE(2, '(//,6X,A4,2X,A7,2X,A9,2X,A9)')
       +'RANK', 'MINIMUM', 'COMPONENT', 'COMPONENT'
        WRITE(2, '(14X,A5,5X,A6,6X,A5)')
       +'PATHS', 'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A7,2X,A9,2X,A9)')
       +'----', '-------', '---------', '---------'
        DO 125 K = 1, MTDMSN
          WRITE(2, '(7X,I2,5X,I3,8X,I2,5X,A20)')
     +    K, MXMINA(K), MXORDA(K), MTLBL$(MXORDA(K))
  125   CONTINUE
        WRITE(2, '(1H1,/,6X,A,/,6X,A,5X,A,/,6X,9(A5),5X,7(A5),A4)')
       +'THE NUMBER OF MINIMUM PATHS LINKING A COMP-',
       +'ONENT TO ALL OTHERS WHICH ARE DEPENDENT ON IT',
       +'(IN-MINIMUM PATH ACCESSIBILITY RANKING)',
       +(UNLN$, N = 1,9), (UNLN$, N=1,7), '----'
        WRITE(2, '(//,6X,A4,2X,A7,2X,A9,2X,A9)')
       +'RANK', 'MINIMUM', 'COMPONENT', 'COMPONENT'
        WRITE(2, '(14X,A5,5X,A6,6X,A5)')
       +'PATHS', 'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A7,2X,A9,2X,A9)')
       +'----', '-------', '---------', '---------'
        DO 135 K = 1, MTDMSN
          WRITE(2, '(7X,I2,5X,I3,8X,I2,5X,A20)')
     +    K, MXMIND(K), MXORDD(K), MTLBL$(MXORDD(K))
  135   CONTINUE
*
*   CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
        RETURN
        END


*****************************************************************
        SUBROUTINE CRITCL(MXLINK,MTLBL$,MTDMSN,NFILER)
*
*   THIS ROUTINE DETERMINES ALL THE CRITICAL COMPONENTS IN THE SYSTEM
*   A SUMMARY OF THE CRITICAL POSITIONS, AND A RANKED ORDERING OF THE
*   CRITICAL POSITION IMPORTANCE OF THE COMPONENTS IS DERIVED
*
*   ARGUMENTS
*   MXLINK - INTEGER ARRAY REPRESENTING THE MINIMUM LINK MATRIX
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MTDMSN - THE DIMENSION OF THE COMPONENT INTERACTION MATRIX AND OTHER
*            RELATED ARRAYS
*   NFILER - INTEGER POINTER ASSOCIATED WITH THE DIRECT ACCESS FILE 13
*
*   OTHER ARRAYS AND VARIABLES
*   MXTRAK - INTEGER ARRAY TO STORE THE DATA READ FROM FILE 13
*   MXORDR - INTEGER ARRAY TO SORT THE ORDER OF COMPONENTS ACCORDING TO
*            THEIR CRITICAL POSITION TOTALS
*   MXCRIT - INTEGER ARRAY OF THE CRITICAL POSITION TOTALS FOR EACH
*            COMPONENT
*   MXUPPR - INTEGER ARRAY : A "1" IN ANY ELEMENT OF THE ARRAY INDICATES
*            THAT THE COMPONENT WITH THE SAME NUMBER AS THE ELEMENT NUMBER
*            SUPPORTS A LINK IN THE TRACK FOR THE STEP BEING PROCESSED
*   MXLOWR - INTEGER ARRAY : A "1" IN ANY ELEMENT OF THE ARRAY INDICATES
*            THAT THE COMPONENT WITH THE SAME NUMBER AS THE ELEMENT NUMBER
*            TERMINATES A LINK ORIGINATED BY SOME COMPONENT IN MXUPPR
*   NB - INTEGER ARRAY OF THE NUMBER OF VALUES TO AN ASSOCIATED RECORD IN
*        FILE 13.
*   FA - INTEGER COUNTER FOR FINDING THE ORIGINATING COMPONENTS OF A LINK
*   NCHECK - INTEGER INDICATOR THAT A CRITICAL COMPONENT EXISTS IN A
*            TRACK UNDER CONSIDERATION
*   CTLCMT - INTEGER STORAGE FOR THE NUMBER OF AN IDENTIFIED CRITICAL
*            COMPONENT
*   B - INTEGER COUNTER OF THE NUMBER OF COMPONENT NUMBERS TO BE READ FROM
*       A RECORD IN FILE 13 (I.E. THE COMPONENTS WHICH TERMINATE LINKS
*       SPECIFIED BY MXUPPR)
*   SWOP - INTEGER VARIABLE FOR TEMPORARY STORAGE OF DATA IN SORTING
*   LINES - INTEGER COUNTER OF THE NUMBER OF LINES PRINTED TO THE
*           ALTERNATE PRINTFILE (UNIT 2).  A NEW PAGE IS CALLED WHEN
*           IS AT 53.
*   UNLN$ - CHARACTER VARIABLE TO UNDERLINE PRINTFILE HEADINGS
*
*   FILES
*   UNIT 2 - ALTERNATE PRINTFILE FOR PROGRAMME OUTPUT TO USER
*   UNIT 13 - DIRECT ACCESS FILE CREATED IN THE MAIN PROGRAMME TO STORE
*             INFORMATION USED IN THE 'TRACK' SUBROUTINE.
        INTEGER MXLINK(50,50),
     +          MXTRAK(50,30),
     +          MXORDR(50), MXCRIT(50), MXUPPR(50), MXLOWR(50), NB(50),
```

```
     +         MTDMSN,
     +         NFILER, PA, NCHECK, CTLCMT, B, SWOP, LINES
      CHARACTER MTLBL$*20(50),
     +         UNLN$*5/'-----'/
*
*  INITIALIZING VARIABLES AND ARRAYS
*
      LINES = 0
      DO 105 K = 1, MTDMSN
         MXORDR(K) = 0
         MXCRIT(K) = 0
         MXLOWR(K) = 0
         MXUPPR(K) = 0
         NB(K) = 0
  105 CONTINUE
*
*  WRITING THE SUMMARY HEADING TO THE ALTERNATE PRINTFILE, UNIT 2
*
      WRITE(2, '(1H1,/,6X,A,20X,A,/,6X,6(A5),A1,20X,5(A5),A3)')
     +'THE MINIMUM PATH CRITICAL LINKS', '(CRITICAL COMPONENT SUMMARY)',
     +(UNLN$, N=1,6), '-', (UNLN$, N=1,5), '---'
      WRITE(2, '(//,6X,A9,17X,A10,17X,A8)')
     +'DEPENDENT', 'SUPPORTING', 'CRITICAL'
      WRITE(2, '(6X,A9,18X,A9,16X,A9)')
     +'COMPONENT', 'COMPONENT', 'COMPONENT'
      WRITE(2, '(6X,A9,17X,A10,16X,A9)')
     +'---------', '----------', '---------'
*
*  A TRACK IS PERFORMED (AS IN THE ROUTINE "TRACKS") FOR THE
*  LINKAGE BETWEEN EVERY COMPONENT I AND EVERY OTHER COMPONENT J.
*  THE TRACK IS ONLY PERFORMED IF THE LINKAGE IS LONGER THAN 2,
*  AND IF I NOT EQUAL TO J.  THE TRACK PROCEDURE IS SIMILAR TO THAT
*  USED IN "TRACKS".  CRITICAL COMPONENTS ARE RECOGNIZED WHEN THERE IS
*  ONLY ONE VALUE IN MXLOWR (I.E NCHECK = 1).  THE IDENTITIES OF I, J
*  AND ANY CRITICAL COMPONENT FOUND ARE PRINTED TO THE ALTERNATE PRINT
*  FILE.
*
      NFILER = 1
      DO 195 I = 1, MTDMSN
         WRITE(2, '(A)') ' '
         LINES = LINES + 1
         MXORDR(I) = I
         DO 125 J = 1, MTDMSN
            READ(13'NFILER, ERR = 115) NB(J),
     +      (MXTRAK(J,N), N = 1, NB(J), +1)
  115       IF(NB(J).EQ.-2) THEN
               WRITE(*, '(//,4X,A,/,4X,A,/,4X,A,/,4X,A)')
     +         '**** THERE ARE MORE THAN THIRTY MINIMUM PATHS LINKING *',
     +         '**** AT LEAST ONE PAIR OF COMPONENTS. THIS EXCEEDS THE *',
     +         '**** CAPABILITIES OF THE PROGRAMME. CRITICAL LINK ANA- *',
     +         '**** LYSIS CANNOT BE PERFORMED                         *'
               GO TO 235
            END IF
  125    CONTINUE
         DO 185 J = 1, MTDMSN
            IF((MXLINK(I,J).LE.1).OR.(I.EQ.J)) GO TO 185
            DO 175 NSTEP = 1, (MXLINK(I,J) - 1)
               IF(NSTEP.EQ.1) THEN
                  DO 135 K = 1, MTDMSN
                     MXUPPR(K) = 0
                     MXLOWR(K) = 0
  135             CONTINUE
                  MXUPPR(J) = 1
               ELSE
                  DO 145 K = 1, MTDMSN
                     MXUPPR(K) = MXLOWR(K)
                     MXLOWR(K) = 0
  145             CONTINUE
               END IF
               NCHECK = 0
               DO 165 PA = 1, MTDMSN
                  IF(MXUPPR(PA).EQ.1) THEN
                     DO 155 B = 1, NB(PA)
                        IF(MXLOWR(MXTRAK(PA,B)).EQ.0) THEN
                           MXLOWR(MXTRAK(PA,B)) = 1
                           NCHECK = NCHECK + 1
                           CTLCMT = MXTRAK(PA,B)
                        END IF
  155                CONTINUE
                  END IF
  165          CONTINUE
               IF(NCHECK.EQ.1) THEN
                  MXCRIT(CTLCMT) = MXCRIT(CTLCMT) + 1
                  IF(LINES.GT.53) THEN
                     WRITE(2, '(1H1,5X,A9,17X,A10,17X,A8)')
     +               'DEPENDENT', 'SUPPORTING', 'CRITICAL'
                     WRITE(2, '(5X,A9,18X,A9,16X,A9)')
     +               'COMPONENT', 'COMPONENT', 'COMPONENT'
                     WRITE(2, '(6X,A9,17X,A10,16X,A9,/)')
     +               '---------', '----------', '---------'
                     LINES = 0
                  END IF
                  WRITE(2, '(3X,3(3X,I2,1X,A20))')
     +            I, MTLBL$(I), J, MTLBL$(J), CTLCMT, MTLBL$(CTLCMT)
                  LINES = LINES +1
               END IF
  175       CONTINUE
  185    CONTINUE
  195 CONTINUE
*
*  SORTING THE COMPONENTS BY THE TOTAL NUMBER OF TIMES THEY HOLD
*  CRITICAL POSITIONS.  THESE VALUES ARE STORED IN THE ARRAY "MXCRIT".
*
      DO 215 I = 1, (MTDMSN-1)
         DO 205 J = (I+1), MTDMSN
            IF(MXCRIT(I).LT.MXCRIT(J)) THEN
               SWOP = MXCRIT(I)
               MXCRIT(I) = MXCRIT(J)
               MXCRIT(J) = SWOP
               SWOP = MXORDR(I)
               MXORDR(I) = MXORDR(J)
               MXORDR(J) = SWOP
            END IF
  205    CONTINUE
  215 CONTINUE
*
*  PRINTING THE RANKED ORDER AND ASSOCIATED VALUES OF THE SORTED
*  COMPONENTS TO THE ALTERNATE PRINTFILE.
*
      WRITE(2, '(1H1,/,6X,A,5X,A,/,6X,10(A5),A,5X,5(A5),A4)')
     +'COMPONENTS RANKED IN ORDER OF CRITICAL PATHS SUPPORTED',
     +'(CRITICAL IMPORTANCE RANKING)',
     +(UNLN$, N=1,10), '----', (UNLN$, N=1,5), '----'
      WRITE(2, '(//,6X,A4,2X,A8,2X,A9,2X,A9)')
     +'RANK', 'CRITICAL', 'COMPONENT', 'COMPONENT'
      WRITE(2, '(15X,A5,5X,A6,6X,A5)')
     +'PATHS', 'NUMBER', 'LABEL'
      WRITE(2, '(6X,A4,2X,A8,2X,A9,2X,A9)')
     +'----', '--------', '---------', '---------'
      DO 225 K = 1, MTDMSN
         WRITE(2, '(7X,I2,4X,I4,9X,I2,5X,A20)')
     +   K, MXCRIT(K), MXORDR(K), MTLBL$(MXORDR(K))
  225 CONTINUE
      MAX = (MTDMSN-1)*(MTDMSN-1)
      WRITE(2, '(//,3X,A10,I4)') 'MAXIMUM = ', MAX
*
*  CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
  235 RETURN
      END

*************************************************************************
      SUBROUTINE CUTTER(MXCMPT,MXLINK,MTLBL$,MTOMSN)
*
*  THIS ROUTINE DETERMINES ALL THE CUT-COMPONENTS TO THE SYSTEM
*  REPRESENTED IN THE COMPONENT INTERACTION MATRIX, AND RANKS
*  THE COMPONENTS IN THE ORDER OF TOTAL NUMBER OF CUT-POSITIONS
*  HELD
*
*  ARGUMENTS
*  MXCMPT - INTEGER ARRAY OF THE COMPONENT MATRIX ELEMENTS
*  MXLINK - INTEGER ARRAY REPRESENTING THE MINIMUM LINK MATRIX
*  MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*  MTDMSN - THE DIMENSION OF THE COMPONENT INTERACTION MATRIX AND OTHER
*           RELATED ARRAYS
*
*  OTHER ARRAYS AND VARIABLES
*  MXPATH - THE STORAGE ARRAY FOR A CURRENT MINIMUM LINK MATRIX
*  MXUNIT - INTEGER STORAGE ARRAY FOR THE POWERING PROCESS.  STORES THE
```

```
*             MATRIX TO BE OPERATED UPON FOR THE NEXT POWER LEVEL
*   MXMULT - INTEGER STORAGE ARRAY FOR THE POWERING PROCESS.  STORES THE
*             PRODUCT INFORMATION
*   MXCUT - INTEGER ARRAY TO RECORD THE NUMBER OF TIMES A COMPONENT IS A
*            CUT-COMPONENT.
*   MXORDR - INTEGER ARRAY TO RECORD THE RANKED ORDER OF THE COMPONENT
*             NUMBERS IN THE SORTING PROCESS ON "MXCUT"
*   CMPT - INTEGER VARIABLE OF THE COMPONENT BEING CONSIDERED FOR CUT-
*           POSITIONS
*   KILLER - INTEGER INDICATOR TO TERMINATE THE POWERING PROCESS ONCE THE
*             MINIMUM LINK MATRIX HAS BEEN DERIVED
*   LINK - INTEGER COUNTER OF THE POWERING LEVEL IN THE DERIVATION OF
*           THE MINIMUM LINK MATRIX
*   SWOP - INTEGER STORAGE USED IN THE SORTING PROCEDURE
*   LINES - INTEGER COUNTER OF THE NUMBER OF LINES ON THE CURRENT PRINT
*            PAGE IN THE ALTERNATE PRINTFILE.
*   UNLN$ - CHARACTER VARIABLE FOR UNDERLINING THE HEADINGS IN THE
*            ALTERNATE PRINTFILE
*
*   FILES
*   UNIT 2 - THE ALTERNATE PRINTFILE FOR OUTPUT TO THE USER
*   UNIT 13 - DIRECT ACCESS FILE CREATED IN THE MAIN PROGRAMME TO STORE
*              INFORMATION USED IN THE 'TRACK' SUBROUTINE.
*
      INTEGER MXCMPT(50,50), MXLINK(50,50),
     +        MXPATH(50,50), MXUNIT(50,50), MXMULT(50),
     +        MXCUT(50), MXORDR(50),
     +        MTDMSN,
     +        CMPT, KILLER, LINK, SWOP, LINES
      CHARACTER MTLBL$*20(50),
     +           UNLN$*5/'-----'/
*
*   INITIALIZATION OF VARIABLES AND ARRAYS
*
      LINES = 0
      DO 105 N = 1, MTDMSN
        MXORDR(N) = N
        MXCUT(N) = 0
  105 CONTINUE
*
*   PRINTING THE CUT-COMPONENT SUMMARY HEADINGS TO THE ALTERNATE PRINTFILE
*
      WRITE(2, '(1H1,/,6X,A,8X,A,/,6X,8(A5),A3,8X,4(A5),A3)')
     +'MINIMUM AND NON-MINIMUM PATH CRITICAL LINKS',
     +'(CUT-COMPONENT SUMMARY)',
     +(UNLN$, N=1,8), '---', (UNLN$, N=1,4), '---'
      WRITE(2, '(//,6X,A9,17X,A10,21X,A4)')
     +'DEPENDENT', 'SUPPORTING', 'CUT-'
      WRITE(2, '(6X,A9,18X,A9,16X,A9)')
     +'COMPONENT', 'COMPONENT', 'COMPONENT'
      WRITE(2, '(6X,A9,17X,A10,16X,A9)')
     +'---------', '----------', '---------'
*
*   THE MINIMUM LINK MATRIX IS DERIVED IN MXPATH FOR THE COMPONENT
*   INTERACTION MATRIX (CMPT = 0), AND THEN FOR THE MATRIX WITH ONE
*   COMPONENT 'DELETED' (CMPT = 1,MTDMSN).  THE PROCEDURE DOCUMENTED
*   IN "MINLNK" IS USED.
*
      DO 195  CMPT = 0, MTDMSN
        DO 115 I = 1, MTDMSN
          DO 115 J = 1, MTDMSN
            IF((MXCMPT(I,J).GT.0).AND.(I.NE.CMPT).AND.(J.NE.CMPT)) THEN
              MXUNIT(I,J) = 1
              MXPATH(I,J) = 1
            ELSE
              MXUNIT(I,J) = 0
              MXPATH(I,J) = 0
            END IF
  115   CONTINUE
        DO 185 LINK = 2, MTDMSN
          KILLER = 0
          DO 155 I = 1, MTDMSN
            IF(I.EQ.CMPT) GO TO 155
            DO 135 J = 1, MTDMSN
              IF((J.EQ.CMPT).OR.(MXPATH(I,J).GT.0)) GO TO 135
              MXMULT(J) = 0
              DO 125 K = 1, MTDMSN
                IF(K.EQ.CMPT) GO TO 125
                IF((MXCMPT(K,J).GE.1).AND.(MXUNIT(I,K).GE.1)) THEN
                  MXMULT(J) = MXMULT(J) + MXUNIT(I,K)
                END IF
  125         CONTINUE
              IF(MXMULT(J).GT.0) THEN
                KILLER = 1
                MXPATH(I,J) = LINK
              END IF
  135       CONTINUE
            DO 145 J = 1, MTDMSN
              IF((MXUNIT(I,J).EQ.0).AND.(MXMULT(J).GT.0).AND.
     +        (J.NE.CMPT)) THEN
                MXUNIT(I,J) = MXMULT(J)
              END IF
  145       CONTINUE
  155     CONTINUE
*
*   ONCE THE CURRENT MINIMUM LINK MATRIX HAS BEEN DERIVED, EVERY ELEMENT
*   IN MXPATH IS COMPARED WITH THE CORRESPONDING ELEMENT IN MXLINK.  CUT-
*   POSITIONS ARE RECOGNIZED WHEN THERE IS A NON-ZERO ELEMENT IN MXLINK,
*   AND A ZERO ELEMENT IN MXPATH.  IF A CUT-POSITION IN FOUND IN THIS
*   WAY, THE CUT-COMPONENT (CMPT), I, AND J, ARE RECORDED IN THE CUT-
*   COMPONENT SUMMARY.  TOTALS FOR THE NUMBER OF TIMES A COMPONENT HOLDS
*   A CUT-POSITION ARE KEPT IN MXCUT.
*
          IF((KILLER.EQ.0).OR.(LINK.EQ.(MTDMSN-1))) THEN
            WRITE(2, '(A)') ' '
            LINES = LINES + 1
            DO 175 I = 1, MTDMSN
              IF(I.EQ.CMPT) GO TO 175
              DO 165 J = 1, MTDMSN
                IF((J.EQ.CMPT).OR.(J.EQ.I)) GO TO 165
                IF((MXPATH(I,J).EQ.0).AND.(MXLINK(I,J).GT.0)) THEN
                  MXCUT(CMPT) = MXCUT(CMPT) + 1
                  IF(LINES.GT.53) THEN
                    WRITE(2, '(1H1,5X,A9,17X,A10,21X,A4)')
     +              'DEPENDENT', 'SUPPORTING', 'CUT-'
                    WRITE(2, '(6X,A9,18X,A9,16X,A9)')
     +              'COMPONENT', 'COMPONENT', 'COMPONENT'
                    WRITE(2, '(6X,A9,17X,A10,16X,A9)')
     +              '---------', '----------', '---------'
                    LINES = 0
                  END IF
                  WRITE(2, '(3X,3(3X,I2,1X,A20))')
     +            I, MTLBL$(I), J, MTLBL$(J), CMPT, MTLBL$(CMPT)
                  LINES = LINES + 1
                END IF
  165         CONTINUE
  175       CONTINUE
            GO TO 195
          END IF
  185   CONTINUE
  195 CONTINUE
*
*   THE VALUES IN MXCUT AND THE COMPONENT ORDER IN MXORDR ARE SORTED
*   INTO RANK ORDER FOR THE CUT-COMPONENT RANKING
*
      DO 215 I = 1, (MTDMSN-1)
        DO 205 J = (I+1), MTDMSN
          IF(MXCUT(I).LT.MXCUT(J)) THEN
            SWOP = MXCUT(I)
            MXCUT(I) = MXCUT(J)
            MXCUT(J) = SWOP
            SWOP = MXORDR(I)
            MXORDR(I) = MXORDR(J)
            MXORDR(J) = SWOP
          END IF
  205   CONTINUE
  215 CONTINUE
*
*   THE RANKED VALUES IN MXCUT, AND THE RANKED COMPONENT POSITIONS IN
*   MXORDR ARE PRINTED TO THE ALTERNATE PRINTFILE
*
      WRITE(2, '(1H1,/,6X,A,5X,A,/,6X,9(A5),A3,5X,4(A5),A3)')
     +'COMPONENTS RANKED IN ORDER OF CUT POSITIONS HELD',
     +'(CUT-COMPONENT RANKING)',
     +(UNLN$, N=1,9), '---', (UNLN$, N=1,4), '---'
      WRITE(2, '(//,6X,A4,7X,A4,2X,A9,2X,A9)')
     +'RANK', 'CUT-', 'COMPONENT', 'COMPONENT'
      WRITE(2, '(12X,A9,5X,A6,6X,A5)')
     +'POSITIONS', 'NUMBER', 'LABEL'
      WRITE(2, '(6X,A4,2X,A9,2X,A9,2X,A9)')
```

```
     +'-----', '----------', '----------', '----------'
      DO 225 K = 1, MTDMSN
        WRITE(2, '(7X,I2,9X,I4,9X,I2,6X,A20)')
     +  K, MXCUT(K), MXORDR(K), MTLBL$(MXORDR(K))
  225 CONTINUE
      MAX = (MTDMSN-1)*(MTDMSN-1)
      WRITE(2, '(//,3X,A10,I4)') 'MAXIMUM = ', MAX
*
*   CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
      RETURN
      END


***********************************************************************
      SUBROUTINE DISRPT(MXCMPT,MTLBL$,MTDMSN)
*
*   THIS ROUTINE DERIVES THE DISRUPTIVE MEASURES ASSOCIATED WITH EACH
*   COMPONENT AND RANKS THE COMPONENTS ACCORDINGLY
*
*   ARGUMENTS
*   MXCMPT - INTEGER ARRAY OF THE COMPONENT MATRIX ELEMENTS
*   MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*   MTDMSN - THE DIMENSION OF THE COMPONENT INTERACTION MATRIX AND OTHER
*
*   OTHER ARRAYS AND VARIABLES
*   MXRNKA AND MXRNKD - REAL ARRAYS FOR THE OUT- AND IN-DISRUPTIVE
*                       MEASURES RESPECTIVELY FOR EACH COMPONENT
*   MXCMTA AND MXCMTD - THE OUT- AND IN-AVERAGE GRAPH DISTANCES
*                       RESPECTIVELY OF ALL COMPONENTS IN THE FULL
*                       INTERACTION MATRIX (REAL STORAGE ARRAYS)
*   NA AND ND - REAL COUNTERS OF THE NUMBER OF NON-ZERO VALUES TO
*               OUT- AND IN-GRAPH DISTANCE SUMMATIONS RESPECTIVELY
*   STORA AND STORD - REAL STORAGE FOR THE ADDITION OF VALUES FOR THE
*                     AVERAGE GRAPH DISTANCES (OUT- AND IN- RESPECTIVELY)
*   SUMA AND SUMD - REAL STORAGE FOR THE OUT- AND IN- DISRUPTIVE MEASURE
*                   TOTALS RESPECTIVELY
*   RSWOP - REAL STORAGE USED IN THE SORTING PROCEDURE
*   MXLINK - THE MINIMUM LINK MATRIX OF THE CURRENT FORM OF THE INTERACTION
*            MATRIX
*   MXUNIT AND MXMULT - INTEGER STORAGE ARRAYS FOR THE POWERING PROCESS
*   MXORDA AND MXORDD - INTEGER ARRAYS TO RECORD THE RANKED ORDER OF THE
*                       COMPONENTS SORTED FOR THE OUT- AND IN-MEASURES
*   CMPT - THE INTEGER VALUE OF THE COMPONENT NUMBER UNDER CONSIDERATION
*          FOR THE DISRUPTIVE MEASURES
*   KILLER - INTEGER INDICATOR TO INDICATE WHEN THE MINIMUM LINK MATRIX
*            HAS BEEN DERIVED
*   LINK - INTEGER INDICATOR OF THE LEVEL OF POWERING IN THE MINIMUM LINK
*          DERIVATION PROCESS
*   ISWOP - INTEGER STORAGE FOR THE SORTING PROCEDURE
*   UNLN$ - CHARACTER VARIABLE FOR THE UNDERLINING OF PRINTED HEADINGS
*
*   FILES
*   UNIT 2 - THE ALTERNATE PRINTFILE ONTO WHICH THE RANKINGS ARE PRINTED
*
      REAL MXRNKA(50), MXRNKD(50),
     +     MXCMTA(50), MXCMTD(50),
     +     NA, ND, STORA, STORD, SUMA, SUMD, RSWOP
      INTEGER MXCMPT(50,50),
     +        MXLINK(50,50), MXUNIT(50,50),
     +        MXMULT(50), MXORDA(50), MXORDD(50),
     +        MTDMSN,
     +        CMPT, KILLER, LINK, ISWOP
      CHARACTER MTLBL$*20(50),
     +          UNLN$*5/'-----'/
      EQUIVALENCE (MXORDA(1), MXUNIT(1,1)), (MXORDD(1), MXMULT(1))
*
*   INITIALIZING THE VARIABLES AND ARRAYS
*
      DO 105 K = 1, MTDMSN
        MXRNKA(K) = 0.0
        MXRNKD(K) = 0.0
        MXCMTA(K) = 0.0
        MXCMTD(K) = 0.0
  105 CONTINUE
*
*   CMPT IS USED TO CONTROL THE REPETITATIVE PROCESS OF 'REMOVING' ONE
*   COMPONENT FROM THE SYSTEM AT A TIME.  THE MINIMUM LINK MATRIX
*   IS THEN DERIVED FOR EACH REDUCED SYSTEM.  THE AVERAGE GRAPH DISTANCES
*   ARE COMPUTED FOR EACH SYSTEM, AND USED TO COMPUTE THE DISRUPTIVE
*   MEASURES FOR THE 'REMOVED' COMPONENT.  THE DISRUPTIVE MEASURES OF
*   EACH COMPONENT ARE STORED IN MXRNKA AND MXRNKD.  THE AVERAGE GRAPH
*   DISTANCES FOR THE FULL SYSTEM WHICH ARE USED FOR EACH CALCULATION ARE
*   STORED IN MXCMTA AND MXCMTD.  THE VALUES OF MXCMTA/D ARE COMPUTED FOR
*   THE FIRST ITERATION OF CMPT (I.E. CMPT = 0).   THE PROCEDURES FOR
*   DERIVING THE MINIMUM LINK MATRIX IS SIMILAR TO THAT IN "MINLNK", AND
*   THOSE FOR COMPUTING THE AVERAGE GRAPH DISTANCES ARE DOCUMENTED IN
*   "DSTANS".
*
      DO 215  CMPT = 0, MTDMSN
        DO 115 I = 1, MTDMSN
          DO 115 J = 1, MTDMSN
*
*   INITIALIZING THE POWERING MATRICES FOR A NEW MINIMUM LINK PROCESS
*
            IF((MXCMPT(I,J).GT.0).AND.(I.NE.CMPT).AND.(J.NE.CMPT)) THEN
              MXUNIT(I,J) = 1
              MXLINK(I,J) = 1
            ELSE
              MXUNIT(I,J) = 0
              MXLINK(I,J) = 0
            END IF
  115   CONTINUE
*
*   DERIVING THE MINIMUM LINK MATRIX
*
        DO 205 LINK = 2, MTDMSN
          KILLER = 0
          DO 155 I = 1, MTDMSN
            IF(I.EQ.CMPT) GO TO 155
            DO 135 J = 1, MTDMSN
              IF((J.EQ.CMPT).OR.(MXLINK(I,J).GT.0)) GO TO 135
              MXMULT(J) = 0
              DO 125 K = 1, MTDMSN
                IF(K.EQ.CMPT) GO TO 125
                IF((MXCMPT(K,J).GE.1).AND.(MXUNIT(I,K).GE.1)) THEN
                  MXMULT(J) = MXMULT(J) + MXUNIT(I,K)
                END IF
  125         CONTINUE
              IF(MXMULT(J).GT.0) THEN
                KILLER = 1
                MXLINK(I,J) = LINK
              END IF
  135       CONTINUE
            DO 145 J = 1, MTDMSN
              IF((MXUNIT(I,J).EQ.0).AND.(MXMULT(J).GT.0).AND.
     +        (J.NE.CMPT)) THEN
                MXUNIT(I,J) = MXMULT(J)
              END IF
  145       CONTINUE
  155     CONTINUE
*
*   COMPUTING THE AVERAGE GRAPH DISTANCES FOR EACH COMPONENT
*
            IF((KILLER.EQ.0).OR.(LINK.EQ.MTDMSN)) THEN
*
*   THE FULL SYSTEM AVERAGE GRAPH DISTANCES
*
              IF(CMPT.EQ.0) THEN
                DO 175 I = 1, MTDMSN
                  NA = 0.0
                  ND = 0.0
                  DO 165 J = 1, MTDMSN
                    IF(MXLINK(I,J).GT.0) THEN
                      MXCMTA(I) = MXCMTA(I) + FLOAT(MXLINK(I,J))
                      NA = NA + 1.0
                    END IF
                    IF(MXLINK(J,I).GT.0) THEN
                      MXCMTD(I) = MXCMTD(I) + FLOAT(MXLINK(J,I))
                      ND = ND + 1.0
                    END IF
  165             CONTINUE
                  IF(NA.GT.0.0) MXCMTA(I) = (MXCMTA(I)/NA)
                  IF(ND.GT.0.0) MXCMTD(I) = (MXCMTD(I)/ND)
  175           CONTINUE
*
*   THE AVERAGE GRAPH DISTANCES FOR THE REDUCED SYSTEMS
*
              ELSE
                SUMA = 0.0
```

```
                  SUMD = 0.0
                  DO 195 I = 1, MTDMSN
                    IF(I.EQ.CMPT) GO TO 195
                    STORA = 0.0
                    STORD = 0.0
                    NA = 0.0
                    ND = 0.0
                    DO 185 J = 1, MTDMSN
                      IF(J.EQ.CMPT) GO TO 185
                      IF(MXLINK(I,J).GT.0) THEN
                        STORA = STORA + FLOAT(MXLINK(I,J))
                        NA = NA + 1.0
                      END IF
                      IF(MXLINK(J,I).GT.0) THEN
                        STORD = STORD + FLOAT(MXLINK(J,I))
                        ND = ND + 1.0
                      END IF
  185               CONTINUE
*
*  COMPUTING THE DISRUPTIVE MEASURES FROM THE DIFFERENCES BETWEEN THE
*  FULL SYSTEM AND REDUCED SYSTEM AVERAGE GRAPH DISTANCES OVER ALL THE
*  REMAINING COMPONENTS IN THE SYSTEM
*
                    IF(NA.GT.0.0) THEN
                      SUMA = SUMA + ABS(MXCMTA(I) - (STORA/NA))
                    ELSE
                      SUMA = SUMA + MXCMTA(I)
                    END IF
                    IF(ND.GT.0.0) THEN
                      SUMD = SUMD + ABS(MXCMTD(I) - (STORD/ND))
                    ELSE
                      SUMD = SUMD + MXCMTD(I)
                    END IF
  195             CONTINUE
                  MXRNKA(CMPT) = SUMA
                  MXRNKD(CMPT) = SUMD
                END IF
                GO TO 215
              END IF
  205       CONTINUE
  215 CONTINUE
*
*  SORTING THE COMPONENTS INTO RANKED ORDER OF THEIR DISRUPTIVE MEASURES
*
        DO 225 N = 1, MTDMSN
          MXORDA(N) = N
          MXORDD(N) = N
  225 CONTINUE
        DO 245 I = 1, (MTDMSN-1)
          DO 235 J = (I+1), MTDMSN
            IF(MXRNKA(I).LT.MXRNKA(J)) THEN
              RSWOP = MXRNKA(I)
              MXRNKA(I) = MXRNKA(J)
              MXRNKA(J) = RSWOP
              ISWOP = MXORDA(I)
              MXORDA(I) = MXORDA(J)
              MXORDA(J) = ISWOP
            END IF
            IF(MXRNKD(I).LT.MXRNKD(J)) THEN
              RSWOP = MXRNKD(I)
              MXRNKD(I) = MXRNKD(J)
              MXRNKD(J) = RSWOP
              ISWOP = MXORDD(I)
              MXORDD(I) = MXORDD(J)
              MXORDD(J) = ISWOP
            END IF
  235     CONTINUE
  245 CONTINUE
*
* PRINTING THE RANKED ORDER OF COMPONENTS AND DISRUPTIVE MEASURES TO THE
* ALTERNATIVE PRINTFILE
*
        WRITE(2, '(1H1,/,6X,A,/,6X,A,5X,A,/,6X,11(A5),A1,5X,9(A5),A3)')
       +'COMPONENTS RANKED IN ORDER OF THE DISRUPTIVE POTENTIAL',
       +'ON THEM FROM THE OTHER COMPONENTS UPON WHICH THEY DEPEND',
       +'(COMPONENT SUSCEPTIBILITY TO DISRUPTION RANKING)',
       +(UNLN$, N = 1, 11), '-', (UNLN$, N=1,9), '---'
        WRITE(2, '(//,6X,A4,2X,A10,2X,A9,2X,A9)')
       +'RANK', 'DISRUPTIVE', 'COMPONENT', 'COMPONENT'
        WRITE(2, '(15X,A7,5X,A6,6X,A5)')
       +'MEASURE', 'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A10,2X,A9,2X,A9)')
       +'----', '----------', '---------', '---------'
        DO 255 K = 1, MTDMSN
          WRITE(2, '(7X,I2,3X,F7.1,9X,I2,5X,A20)')
       +  K, MXRNKD(K), MXORDD(K), MTLBL$(MXORDD(K))
  255 CONTINUE
        NA = (FLOAT(MTDMSN-1)*FLOAT(MTDMSN+1))/2.0
        WRITE(2, '(//,2X,A10,F7.1)') 'MAXIMUM = ', NA
        WRITE(2, '(1H1,/,6X,A,/,6X,A,5X,A,/,6X,9(A5),A2,5X,8(A5))')
       +'COMPONENTS RANKED IN ORDER OF THEIR DISRUPTIVE',
       +'POTENTIAL ON OTHER COMPONENTS DEPENDENT ON THEM',
       +'(COMPONENT DISRUPTIVE POTENTIAL RANKING)',
       +(UNLN$, N = 1, 9), '--', (UNLN$, N=1,8)
        WRITE(2, '(//,6X,A4,2X,A10,2X,A9,2X,A9)')
       +'RANK', 'DISRUPTIVE', 'COMPONENT', 'COMPONENT'
        WRITE(2, '(15X,A7,5X,A6,6X,A5)')
       +'MEASURE', 'NUMBER', 'LABEL'
        WRITE(2, '(6X,A4,2X,A10,2X,A9,2X,A9)')
       +'----', '----------', '---------', '---------'
        DO 265 K = 1, MTDMSN
          WRITE(2, '(7X,I2,3X,F7.1,9X,I2,5X,A20)')
       +  K, MXRNKA(K), MXORDA(K), MTLBL$(MXORDA(K))
  265 CONTINUE
        WRITE(2, '(//,2X,A10,F7.1)') 'MAXIMUM = ', NA
*
* CONTROL IS RETURNED TO THE MAIN PROGRAMME
*
        RETURN
        END


*****************************************************************************
        SUBROUTINE RUNFIN(MXCMPT,MTLBL$,MSG$,MTDMSN,TTLN,IMIN,IMAX,FTYPE,
       +NTRACK,ERMSG$,HDG$,TIME$)
*
*  THIS ROUTINE PERFORMS THE TERMINATING FUNCTIONS OF THE PROGRAMME
*
*  ARGUMENTS
*  MXCMPT - INTEGER ARRAY OF THE COMPONENT INTERACTION MATRIX DATA
*  MTLBL$ - CHARACTER ARRAY OF THE COMPONENT LABELS
*  MSG$ - CHARACTER ARRAY OF THE PROGRAMME MESSAGES FOR PRINTING
*           TO THE TERMINAL SCREEN BY THE ROUTINE 'MSGPRT'
*  MTDMSN - INTEGER VARIABLE OF THE COMPONENT MATRIX DIMENSION
*  TTLN - INTEGER INDICATOR OF THE FORM OF THE MATRIX TO BE PRINTED TO
*           THE ALTERNATE PRINTFILE (UNIT 2) BY THE ROUTINE 'PRTMRT'
*  IMIN - THE MINIMUM VALUE (INTEGER) OF A COMPONENT MATRIX ELEMENT
*  IMAX - THE MAXIMUM VALUE (INTEGER) OF A COMPONENT MATRIX ELEMENT
*  FTYPE - INTEGER INDICATOR OF THE MODE OF DATA ENTRY USED BY THE
*           USER TO ENTER DATA TO THE PROGRAMME
*  NTRACK - INTEGER COUNTER OF THE NUMBER OF TRACKS REQUESTED BY
*           THE USER THROUGH THE ROUTINE 'TRACKS'
*  ERMSG$ - CHARACTER VARIABLE OF THE STANDARD ERROR MESSAGE
*  HDG$ - CHARACTER VARIABLE OF A USER SUPPLIED HEADING FOR THE
*           HARDCOPY PRINTOUT OF THE PROGRAMME OUTPUT
*  TIME$ - CHARACTER VARIABLE OF THE UNIVAC SUPPLIED TIME (CALLED
*           BY 'ADATE$' IN MAIN PROGRAMME), USED TO SPECIFY A UNIQUE
*           FILE FOR THE ALTERNATE PRINTFILE - UNIT 2
*
*  OTHER ARRAYS AND VARIABLES
*  MSGN - INTEGER VARIABLE OF THE NUMBER OF MESSAGES IN MSG$ TO BE
*           PRINTED BY 'MSGPRT'
*  SKIP - INTEGER INDICATOR TO 'MSGPRT' TO PRINT THE USER PROMPT
*  FNEND - INTEGER VARIABLE OF THE NUMBER OF CHARACTERS IN THE USER
*           SUPPLIED DATA FILE NAME.  NEEDED FOR A CALL TO 'FACSF2'
*  REPLY - INTEGER VARAIBLE FOR USER REPLIES TO PROGRAMME OPTIONS
*  FACSF2 - UNIVAC SYSTEM INTEGER FUNCTION TO EXECUTE EXEC8 COMMANDS
*           FROM WITHIN THE PROGRAMME (SEE SPERRY UNIVAC PROGRAMMER
*           REFERENCE MANUAL UP-8244.1, ASCII FORTRAN LEVEL 9R1, P7-31)
*  STATUS - INTEGER VARIABLE SET BY THE FUNCTION FACSF2 TO REFLECT THE
*           STATUS RETURNED BY THE FUNCTION
*  ASGNED - INTEGER VARIABLE USED TO CHECK THE STATUS RETURNED BY FACSF2
*           FOR AN EXEC REQUEST TO ASSIGN A FILE.  SET TO THE EQUIVALENT
*           OCTAL VALUE INDICATING FILE ALREADY ASSIGNED TO PROGRAMME
*  REPLY$ - CHARACTER VARIABLE FOR USER REPLIES TO PROGRAMME QUESTIONS
*  FNAME$ - CHARACTER VARIABLE OF THE USER SUPPLIED DATA FILE (UNIT 1) NAME
*
*  FILES
*  UNIT 1 - PERMANENT DATA FILE CONTAINING THE COMPONENT MATRIX DATA
*  UNIT 2 - ALTERNATE PRINTFILE
```

```
*   UNIT 3 - DATA FILE USED FOR THE PLOTTING INFORMATION CREATED
*            BY 'TRACKS'
*   UNIT 4 - SECOND DATA FILE FOR PLOTTING INFORMATION CREATED BY
*            'TRACKS'
*
      INTEGER MXCMPT(50,50),
     +        MTDMSN, TTLN, IMIN, IMAX, FTYPE, NTRACK,
     +        MSGN, SKIP, FNEND, REPLY, FACSF2, STATUS,
     +        ASGNED/010000000000/
      CHARACTER MTLBL$*20(50), MSG$*60(6),
     +        ERMSG$*50, HDG$*55, TIMES$*8,
     +        HEAD$*80, REPLY$*3, FNAME$*12
*
*   OFFERING THE USER THE OPPORTUNITY OF WRITING IMPERMANENT DATA
*   (FTYPE = 3) DATA TO A PERMANENT FILE.  THE PROCEDURE FOR ASSIGNING
*   A FILE AND WRITING THE MATRIX DATA TO THE FILE (UNIT 1) FOLLOWS
*
      IF(FTYPE.EQ.3) THEN
        MSG$(1)= 'WOULD YOU LIKE TO CHANGE YOUR MIND AND WRITE *'
        MSG$(2)= 'THE COMPONENT INTERACTION MATRIX DATA TO A   *'
        MSG$(3)= 'PERMANENT DATA FILE?  (YES/NO)               *'
        MSGN = 3
 105      CALL MSGPRT(MSG$,MSGN,SKIP)
        READ(*, '(A3)', ERR = 105) REPLY$
        IF(REPLY$.EQ.'YES') THEN
 115      MSG$(1)= 'ENTER A NAME (AS ANY COMBINATION OF 1 TO 12 *'
          MSG$(2)= 'ALPHANUMERIC CHARACTERS) FOR THE DATA FILE. *'
          MSGN = 2
 125      CALL MSGPRT(MSG$,MSGN,SKIP)
          READ(*, '(A12)', ERR = 125) FNAME$
          FNEND = 12
 135      IF((FNEND.GT.0).AND.(FNAME$(FNEND:FNEND).EQ.' ')) THEN
            FNEND = FNEND - 1
          GO TO 135
          END IF
          IF(FNEND.EQ.0) GO TO 125
          STATUS = FACSF2('@ASG,UP '//FNAME$(1:FNEND)//'.,F2 . ')
          IF(STATUS.NE.0) THEN
            MSG$(1)= 'A FILE OF THIS NAME ALREADY EXISTS UNDER  *'
            MSG$(2)= 'YOUR PROJECT-ID.                           *'
            MSG$(3)= 'TYPE "1" TO ENTER ANOTHER FILE NAME        *'
            MSG$(4)= 'TYPE "2" TO OVER-WRITE THIS EXISTING FILE *'
            MSG$(5)= 'TYPE "3" TO TERMINATE THE PROGRAMME        *'
            MSGN = 5
 145          CALL MSGPRT(MSG$,MSGN,SKIP)
            READ(*, *, ERR = 155) REPLY
            IF(REPLY.EQ.1) THEN
              GO TO 115
            ELSE IF(REPLY.EQ.2) THEN
              IF(STATUS.NE.ASGNED) THEN
                STATUS = FACSF2('@ASG,A '//FNAME$(1:FNEND)//'. . ')
              END IF
            ELSE IF(REPLY.EQ.3) THEN
               HEAD$ = '* RUN ABORTED - NO PERMANENT COPY OF DATA *'
               WRITE(2, '(1H1,//////////,15X,A)') HEAD$
               GO TO 165
            ELSE
 155            PRINT*,ERMSG$
              GO TO 145
            END IF
          END IF
          STATUS = FACSF2('@USE 1,'//FNAME$(1:FNEND)//'. . ')
          CALL MXFWRT(MXCMPT,MTLBL$,MTDMSN,IMIN,IMAX)
          HEAD$ = ' THE MATRIX DATA (PRINTED BELOW) HAS BEEN '//
     +      'STORED ON PERMANENT FILE - '//FNAME$
          TTLN = 4
          CALL PRTWRT(MXCMPT,MTLBL$,MTDMSN,TTLN,HDG$,HEAD$)
        ELSE IF(REPLY$.NE.'NO') THEN
          PRINT*,ERMSG$
          GO TO 105
        END IF
      END IF
*
*   THE ALTERNATE PRINT FILE (UNIT 2) IS BREAKPOINTED AND QUEUED FOR
*   PRINTING.  THE DATA FILE (UNIT 1) IS ALSO FREED.  IF TRACKS HAVE
*   BEEN PERFORMED (I.E. NTRACK GREATER THAN ZERO), UNITS 3 AND 4 ARE
*   RELEASED AND THE PLOTTING PROGRAMME (PLOT.START) IS STARTED BY A
*   CALL TO FACSF2.
*
 165  STATUS = FACSF2('@FREE 1 . ')
      STATUS = FACSF2('@BRKPT 2 . ')
      IF(NTRACK.GT.0) THEN
        ENDFILE 3
        ENDFILE 4
        STATUS = FACSF2('@FREE 3 . ')
        STATUS = FACSF2('@FREE 4 . ')
        STATUS = FACSF2('@START PLOT.START . ')
        MSG$(1)= 'THE TRACK/S THAT YOU INITIATED WILL BE PROCESSED *'
        MSG$(2)= 'BY THE PLOTTER.   THE PLOTTING PRINT-OUT WILL BE *'
        MSG$(3)= 'RETURNED TO YOU UNDER YOUR RUN-ID.               *'
        MSGN = 3
        SKIP = 1
        CALL MSGPRT(MSG$,MSGN,SKIP)
      END IF
*
*   CONTROL RETURNED TO MAIN PROGRAMME WHERE THE RUN IS TERMINATED
*
      RETURN
      END
```

PLOTTING PROGRAMME

```
*******************************************************************
*
*   THIS PROGRAMME PERFORMS THE PLOTTING OF DEPENDENCY TRACKS FOR THE
*   COMPONENT INTERACTION PROGRAMME.   AN ABSOLUTE ELEMENT OF THIS
*   PROGRAMME IS EXECUTED BY A CONTROLLING RUNSTREAM WHICH IS STARTED BY
*   THE COMPONENT INTERACTION PROGRAMME.
*
*   THE DATAFILES USED TO SUPPLY THE PLOTTING DATA TO THE PROGRAMME ARE
*   CREATED IN THE MAIN PROGRAMME, AND WRITTEN TO IN THE "TRACKS" ROUTINE
*   OF THE COMPONENT INTERACTION PROGRAMME
*
*   VARIABLES AND ARRAYS
*   ORDER - INTEGER ARRAY TO STORE THE ORDER OF THE COMPONENTS IN ANY ONE
*           TRACK.  ONLY THE COMPONENTS USED IN A TRACK FORM PART OF THE
*           TRACK PLOT.  THE COMPONENT NUMBERS ARE RANKED IN THE ORDER IN
*           WHICH THEY APPEAR IN THE TRACK GRID.
*   RANK$ - CHARACTER ARRAY OF THE COMPONENT NUMBERS OF THE COMPONENTS
*           USED IN ANY ONE TRACK
*   LABEL$ - THE COMPONENT LABELS OF THE COMPONENTS USED IN ANY ONE TRACK
*   HDG$ - CHARACTER VARIABLE OF THE HEADING FOR THE TRACKS READ FROM FILE
*          3, AND USER SUPPLIED IN THE MAIN COMPONENT INTERACTION PROGRAMME
*   DATE$ - CHARACTER STORAGE FOR THE DATE READ FROM FILE 3
*   AN$ & BN$ - CHARACTER VARIABLE FOR THE NUMBER OF THE DEPENDENT
*               AND ORIGINATING COMPONENTS OF THE TRACK
*   ALBL$ AND BLBL$ - CHARACTER VARIABLES FOR THE COMPONENT LABELS OF THE
*                     DEPENDENT AND ORIGINATING COMPONENTS OF A TRACK
*   TRKN$ - CHARACTER VARIABLE OF THE TRACK NUMBER (EVERY TRACK HAS A
*           SEQUENTIAL NUMBER)
*   CHAR$ - CHARACTER VARIABLE FOR CHARACTER INFORMATION TO BE PLOTTED
*           BY A CALL TO SYMBOL
*
*   FILES
*   UNIT 3 - PLOT INFORMATION FILE, ASSIGNED TO THE RUN BE THE STARTING
*            RUNSTREAM.
*   UNIT 4 - PLOT DATA FILE, ASSIGNED TO THE RUN BY THE STARTING RUNSTREAM
*
*   PROCEDURE
*   INFORMATION ABOUT EACH PLOT IS READ FIRST FROM FILE 3. THIS IS USED
*   TO DETERMINE THE SIZE OF THE PLOT, TO PLOT THE HEADINGS, AND TO PLOT
*   THE REFERENCE GRID FOR THE TRACKS.
*   THE DATA IN FILE 4 IS THEN READ AND USED TO PLOT THE TRACKS ONTO THE
*   REFERENCE GRID.  END TRACK CONDITIONS ARE RECOGNIZED IN THE VARIOUS
*   READ STATEMENTS, AND NEW TRACKS ARE INITIATED BY BEGINNING AT STATEMENT
*   10.  ONCE THE FILES HAVE BEEN READ THE END, THE PROGRAMME
*   TERMINATES, AND THE CONTROLLING RUNSTREAM DELETES THE PLOTDATA FILES.
*
*   THE PROGRAMME IS WRITTEN IN THE CALCOMP PLOTTING LANGUAGE, WITH SOME
*   NON-CONFORMING STATEMENTS WHICH ALLOW THE PROGRAMME TO INTERFACE WITH
*   UNIVERSITY OF CAPE TOWN 'GDP' LANGUAGE WHICH IS COMPATIBLE WITH THE
*   SYSTEM PLOTTER AND OTHER HARDWARE.
*
*******************************************************************
      INTEGER ORDER(50),
     +        TSTEP, TNODE, HDGEND, X, Y, INTER
      CHARACTER HDG$*50, DATE$*8, AN$*2, BN$*2, ALBL$*20, BLBL$*20,
     +          TRKN$*15, CHAR$*55,
     +          RANK$*2(50), LABEL$*20(50)


*
*   THE VARIABLE CHAR$ IS SET SO THAT THE PLOTTING IS ONLY
*   INITIALIZED ONCE
*
      CHAR$ = 'START'


*
*   READING THE PLOT INFORMATION FROM UNIT 3
*
   10 CALL NEWPEN(1)
      TNODE = 1
*
*   AN END OF FILE CONDITION RECOGNIZED BY THE READ STATEMENT CAUSES THE
*   PLOTTING TO BE TERMINATED BY STATEMENT NUMBER 80.
*
   20 READ(3, '(2X,A2,2X,A20)', END = 80) RANK$(TNODE), LABEL$(TNODE)
         IF(RANK$(TNODE).EQ.'99') GO TO 30
*
*   THE COMPONENT NUMBERS OF THE COMPONENTS USED IN THE TRACK ARE
*   CORRELATED WITH THEIR POSITIONS IN THE TRACK GRID
*
         READ(RANK$(TNODE), '(I2)', ERR = 30) INTER
         ORDER(INTER) = TNODE
         TNODE = TNODE + 1
      GO TO 20
   30 TNODE = TNODE - 1
      READ(3, '(2X,I2)') TSTEP
      READ(3, '(2X,A8,2X,A50,2X,I2)') DATE$, HDG$, HDGEND
      READ(3, '(2X,A2,2X,A20,2X,A2,2X,A20)') AN$, ALBL$, BN$, BLBL$
      READ(3, '(2X,A15)') TRKN$


*
*   THE TRACK IS INITIALIZED.  THE SIZE OF THE PAGE IS DETERMINED FROM
*   THE NUMBER OF LINKS IN THE TRACK (TSTEP), AND THE NUMBER OF NODES
*   (TNODE).  IF THE NUMBER OF NODES IS LESS THAN 7, THE DEFAULT SIZE
*   (IN THE X DIRECTION) IS USED TO ENSURE THAT HEADINGS WILL FIT.
*
      IF(CHAR$.NE.'START') CALL NEWPAG
      CALL PAGNAM(TRKN$)
      IF(TNODE.GT.7) THEN
        XPAGE = ((TNODE*2.0) + 6.5)
        YPAGE = ((TSTEP*2.5) + 10.8)
        CALL PAGSIZ(XPAGE,YPAGE)
        CALL PLOT(0.5,0.5,-3)


*
*   PLOTTING THE BORDER
*
        XPAGE = (XPAGE - 1.0)
        YPAGE = (YPAGE - 1.0)
        CALL PLOT(XPAGE,0.0,2)
        CALL PLOT(XPAGE,YPAGE,2)
        CALL PLOT(0.0,YPAGE,2)
        CALL PLOT(0.0,0.0,2)


*
*   PLOTTING THE HEADINGS
*
        CHAR$ = '  DEPENDENCE OF COMPONENT '//AN$//' - '//ALBL$
        XPAGE = (TNODE - 6.20)
        CALL SYMBOL(XPAGE,1.75,0.35,CHAR$,0,51)
        CHAR$ = ' ON COMPONENT '//BN$//' - '//BLBL$
        XPAGE = (XPAGE + 4.2)
        CALL SYMBOL(XPAGE,1.0,0.35,CHAR$,0,39)
        YPAGE = ((TSTEP*2.5) + 8.9)
        CHAR$ = 'DATE - '//DATE$
        CALL SYMBOL(1.1,YPAGE,0.35,CHAR$,0,13)
        XPAGE = ((TNODE*2.0) - 0.75)
        CALL SYMBOL(XPAGE,YPAGE,0.35,TRKN$,0,15)
        YPAGE = ((TSTEP*2.5) + 8.00)
        XPAGE = ((TNODE + 2.75) - (HDGEND*0.175))
        CALL SYMBOL(XPAGE,YPAGE,0.35,HDG$,0,HOGEND)


*
*   REDEFINING THE ORIGIN FOR THE TRACK GRID
*
        XPAGE = 1.1
        YPAGE = 3.5
        CALL PLOT(XPAGE,YPAGE,-3)
      ELSE
*
*   IF THE DEFAULT SIZEIS USED THE FOLLOWING PLOTTING PERFORMS SIMILAR
*   PLOTTING TO THAT ABOVE
*
        XPAGE = 20.25
        YPAGE = ((TSTEP*2.5) + 10.8)
        CALL PAGSIZ(XPAGE,YPAGE)
        CALL PLOT(0.5,0.5,-3)


        YPAGE = (YPAGE - 1.0)
        XPAGE = (XPAGE - 1.0)
        CALL PLOT(XPAGE,0.0,2)
        CALL PLOT(XPAGE,YPAGE,2)
        CALL PLOT(0.0,YPAGE,2)
```

```
        CALL PLOT(0.0,0.0,2)


        CHAR$ = '  DEPENDENCE OF COMPONENT '//AN$//' - '//ALBL$
        CALL SYMBOL(0.7,1.75,0.35,CHAR$,0,51)
        CHAR$ = ' ON COMPONENT '//BN$//' - '//BLBL$
        CALL SYMBOL(4.9,1.0,0.35,CHAR$,0,39)
        YPAGE = ((TSTEP*2.5) + 8.8)
        CHAR$ = 'DATE - '//DATE$
        CALL SYMBOL(0.7,YPAGE,0.35,CHAR$,0,13)
        CALL SYMBOL(13.05,YPAGE,0.35,TRKN$,0,15)
        YPAGE = ((TSTEP*2.5) + 8.0)
        XPAGE = ((55 - HDGEND)*0.35/2.0)
        CALL SYMBOL(XPAGE,YPAGE,0.35,HDG$,0,HDGEND)


        XPAGE = (7.875 - TNODE)
        YPAGE = 3.5
        CALL PLOT(XPAGE,YPAGE,-3)
      END IF

*
*  THE TRACK GRID AND LABELS ARE PLOTTED
*
      CHAR$ = 'LINK '
      XPAGE = (TNODE*2.0)
      DO 40 K = 1, TSTEP
        YPAGE = (1.075 + ((K - 1)*2.5))
        CALL SYMBOL(XPAGE,YPAGE,0.35,CHAR$,0,5)
        YPAGE = FLOAT(TSTEP - (K-1))
        CALL NUMBER(999.,999.,0.35,YPAGE,0,-1)
   40 CONTINUE

*
*
*  THE TRACK DIAGRAMME IS SCALED TO 2.5 THE PRESENT PLOT SCALE
*
      CALL FACTOR(2.0)
*
*  PLOTTING THE COMPONENT NUMBERS AND LABELS ON THE GRID
*
      DO 50 K = 0,(TNODE-1)
        CHAR$ = RANK$(K+1)
        XPAGE = (K - 0.26)
        YPAGE = -0.225
        CALL SYMBOL(XPAGE,YPAGE,0.175,CHAR$,0,2)
        XPAGE = FLOAT(K)
        CALL PLOT(XPAGE,0.0,3)
        YPAGE = (TSTEP*1.25)
        CALL PLOT(XPAGE,YPAGE,2)
        XPAGE = (K - 0.26)
        YPAGE = ((TSTEP*1.25) + 0.05)
        CALL SYMBOL(XPAGE,YPAGE,0.175,CHAR$,0,2)
        XPAGE = FLOAT(K)
        YPAGE = ((TSTEP*1.25) + 0.275)
        CALL SYMBOL(XPAGE,YPAGE,0.150,LABEL$(K+1),30.0,20)
   50 CONTINUE


      XPAGE = (TNODE - 1.0)
      DO 60 K = 0, TSTEP
        YPAGE = (K*1.25)
        CALL PLOT(0,YPAGE,3)
        CALL PLOT(XPAGE,YPAGE,2)
   60 CONTINUE

*
*  READING THE PLOTING DATA FROM FILE 4, AND PLOTTING THE TRACKS ONTO
*  THE REFERENCE GRID.  AN END CONDITION IS RECOGNIZED WHEN A NON-
*  INTEGER VARIABLE IS READ FROM THE FILE ('END OF TRACK').
*
      CALL NEWPEN(2)
   70 READ(4, '(2X,I2,2X,I2)', ERR = 10) X, Y
        XPAGE = (ORDER(X) - 1)
        YPAGE = ((Y - 1)*1.25)
        CALL PLOT(XPAGE,YPAGE,3)
        READ(4, '(2X,I2,2X,I2)', ERR = 10) X, Y
        XPAGE = (ORDER(X) - 1)
        YPAGE = ((Y - 1)*1.25)
        CALL PLOT(XPAGE,YPAGE,2)
        GO TO 70
   80 CALL ENDPLT
      STOP
      END
```

```
****************************************
THIS RUNSTREAM CONTROLS THE PLOTTING OF
THE DEPENDENCY TRACKS
****************************************
*  the system run statement
  @run runid,account,matrix,10,100
*  assigning the plot data and information files
  @asg,axk plotinfo.
  @use 3,plotinfo.
  @asg,axk plotdata.
  @use 4,plotdata.
*  The GDP programme at the University of Cape Town
*  interfaces with the system plotting hardware
  @gdp*abs.input
*  the plotting programme is executed
  @xqt plot.abstrack
*  the plot data files are released and deleted
  @free 3.
  @free 4.
*  the dummy file "checker" is deleted to indicate to any
*  subsequent runs of the component interaction programme
*  that the last plot run is complete
  @delete checker.
  @fin
```

```
****************************************
THIS RUNSTREAM ACCESSES AND EXECUTES THE
COMPONENT INTERACTION PROGRAMME
****************************************


 @run statement
 security statement
*
*  assigning the programme file
*
 @asg,a component*matrix.
*
*  executing the programme
*
 @xqt matrix.analysis
*
*  the programme commands and user replies
*
 @fin
```

9. APPENDIX B - COMPONENT INTERACTION ANALYSIS EXAMPLES

9.1. Illustrative Estuarine System

9.1.1. Unweighted Illustrative Estuarine System

- component interaction matrix
- minimum link matrix
- valency rankings
- average graph distance rankings
- minimum path accessibility measure rankings
- critical-component summary and ranking
- cut-component summary and ranking
- disruptive measure rankings
- dependency track examples

```
ILLUSTRATIVE ESTUARINE SYSTEM
*********************************

THE COMPONENTS ARE AS FOLLOWS:-

  1  ROOTED VEGETATION
  2  PLANT DETRITUS
  3  BENTHIC FAUNA
  4  ESTUARINE FISH
  5  AVIFAUNA
  6  HUMANS
```

```
THE INITIAL COMPONENT INTERACTION MATRIX
----------------------------------------

       1   2   3   4   5   6
    *************************
  1 *  0   1   1   1   1   0   *  1  ROOTED VEGETATION
  2 *  1   0   0   0   0   0   *  2  PLANT DETRITUS
  3 *  0   1   0   0   0   0   *  3  BENTHIC FAUNA
  4 *  1   1   1   0   0   0   *  4  ESTUARINE FISH
  5 *  0   0   1   1   0   0   *  5  AVIFAUNA
  6 *  0   0   0   1   0   0   *  6  HUMANS
    *************************
       1   2   3   4   5   6
```

```
THE MINIMUM LINK MATRIX
-----------------------

       1   2   3   4   5   6
    *************************
  1 *  2   1   1   1   1   0   *  1  ROOTED VEGETATION
  2 *  1   2   2   2   2   0   *  2  PLANT DETRITUS
  3 *  2   1   3   3   3   0   *  3  BENTHIC FAUNA
  4 *  1   1   1   2   2   0   *  4  ESTUARINE FISH
  5 *  2   2   1   1   3   0   *  5  AVIFAUNA
  6 *  2   2   2   1   3   0   *  6  HUMANS
    *************************
       1   2   3   4   5   6
```

```
(OUT-VALENCY RANKING)

THE NUMBER OF COMPONENTS EACH COMPONENT DEPENDS UPON
----------------------------------------------------

RANK     COMPONENTS     COMPONENT    COMPONENT
       DEPENDED UPON      NUMBER        LABEL
----   -------------   ---------    ---------
 1           4              1       ROOTED VEGETATION
 2           3              4       ESTUARINE FISH
 3           2              5       AVIFAUNA
 4           1              2       PLANT DETRITUS
 5           1              3       BENTHIC FAUNA
 6           1              6       HUMANS

  MAXIMUM =  6
```

```
THE NUMBER OF DEPENDENCIES SUPPORTED BY EACH COMPONENT
------------------------------------------------------

RANK  DEPENDENCIES   COMPONENT    COMPONENT
        SUPPORTED      NUMBER       LABEL
----  ------------   ---------    ---------
 1         3             2        PLANT DETRITUS
 2         3             3        BENTHIC FAUNA
 3         3             4        ESTUARINE FISH
 4         2             1        ROOTED VEGETATION
 5         1             5        AVIFAUNA
 6         0             6        HUMANS

MAXIMUM =  6

(IN-VALENCY RANKING)
```

```
(AVERAGE OUT-GRAPH DISTANCE RANKING)

THE AVERAGE NUMBER OF LINKS PER CONNECTION BETWEEN
A COMPONENT AND OTHERS UPON WHICH IT IS DEPENDENT
--------------------------------------------------

RANK   AVE. NO.   COMPONENT   COMPONENT
       OF LINKS     NUMBER       LABEL
----   --------   ---------   ---------
 1       2.4          3       BENTHIC FAUNA
 2       2.0          6       HUMANS
 3       1.8          5       AVIFAUNA
 4       1.8          2       PLANT DETRITUS
 5       1.4          4       ESTUARINE FISH
 6       1.2          1       ROOTED VEGETATION

MAXIMUM =  3.5
```

```
THE AVERAGE NUMBER OF LINKS PER CONNECTION BETWEEN
A COMPONENT AND OTHERS WHICH ARE DEPENDENT ON IT
--------------------------------------------------

RANK   AVE. NO.   COMPONENT   COMPONENT
       OF LINKS     NUMBER       LABEL
----   --------   ---------   ---------
 1       2.3          5       AVIFAUNA
 2       1.7          3       BENTHIC FAUNA
 3       1.7          4       ESTUARINE FISH
 4       1.7          1       ROOTED VEGETATION
 5       1.5          2       PLANT DETRITUS
 6        .0          6       HUMANS

MAXIMUM =  3.5

(AVERAGE IN-GRAPH DISTANCE RANKING)
```

(OUT-MINIMUM PATH ACCESSIBILITY RANKING)

```
THE NUMBER OF MINIMUM PATHS LINKING A COMP-
ONENT TO ALL OTHERS ON WHICH IT IS DEPENDENT
--------------------------------------------

RANK   MINIMUM   COMPONENT   COMPONENT
        PATHS      NUMBER      LABEL
----   -------   ---------   ---------
 1        6          1       ROOTED VEGETATION
 2        6          5       AVIFAUNA
 3        5          3       BENTHIC FAUNA
 4        5          4       ESTUARINE FISH
 5        5          2       PLANT DETRITUS
 6        5          6       HUMANS
```

(IN-MINIMUM PATH ACCESSIBILITY RANKING)

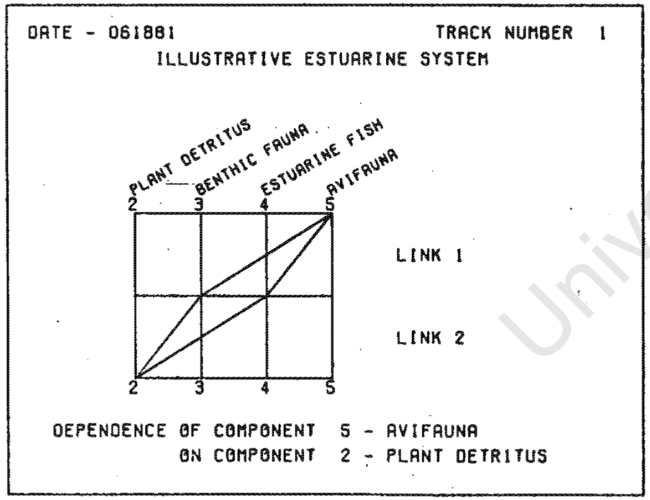
```
THE NUMBER OF MINIMUM PATHS LINKING A COMP-
ONENT TO ALL OTHERS WHICH ARE DEPENDENT ON IT
---------------------------------------------

RANK   MINIMUM   COMPONENT   COMPONENT
        PATHS      NUMBER      LABEL
----   -------   ---------   ---------
 1        7          1       ROOTED VEGETATION
 2        7          2       PLANT DETRITUS
 3        6          3       BENTHIC FAUNA
 4        6          4       ESTUARINE FISH
 5        6          5       AVIFAUNA
 6        0          6       HUMANS
```

```
THE MINIMUM PATH CRITICAL LINKS                     (CRITICAL COMPONENT SUMMARY)
-------------------------------                     ----------------------------

DEPENDENT                SUPPORTING                  CRITICAL
COMPONENT                 COMPONENT                  COMPONENT
---------                ----------                  ---------

 2 PLANT DETRITUS         3 BENTHIC FAUNA             1 ROOTED VEGETATION
 2 PLANT DETRITUS         4 ESTUARINE FISH            1 ROOTED VEGETATION
 2 PLANT DETRITUS         5 AVIFAUNA                  1 ROOTED VEGETATION

 3 BENTHIC FAUNA          1 ROOTED VEGETATION         2 PLANT DETRITUS
 3 BENTHIC FAUNA          4 ESTUARINE FISH            1 ROOTED VEGETATION
 3 BENTHIC FAUNA          4 ESTUARINE FISH            2 PLANT DETRITUS
 3 BENTHIC FAUNA          5 AVIFAUNA                  1 ROOTED VEGETATION
 3 BENTHIC FAUNA          5 AVIFAUNA                  2 PLANT DETRITUS

 4 ESTUARINE FISH         5 AVIFAUNA                  1 ROOTED VEGETATION

 5 AVIFAUNA               1 ROOTED VEGETATION         4 ESTUARINE FISH

 6 HUMANS                 1 ROOTED VEGETATION         4 ESTUARINE FISH
 6 HUMANS                 2 PLANT DETRITUS            4 ESTUARINE FISH
 6 HUMANS                 3 BENTHIC FAUNA             4 ESTUARINE FISH
 6 HUMANS                 5 AVIFAUNA                  1 ROOTED VEGETATION
 6 HUMANS                 5 AVIFAUNA                  4 ESTUARINE FISH
```

(CRITICAL IMPORTANCE RANKING)

```
  COMPONENTS RANKED IN ORDER OF CRITICAL PATHS SUPPORTED
  ------------------------------------------------------

  RANK   CRITICAL   COMPONENT   COMPONENT
            PATHS      NUMBER      LABEL
  ----   --------   ---------   ---------
   1        7           1       ROOTED VEGETATION
   2        5           4       ESTUARINE FISH
   3        3           2       PLANT DETRITUS
   4        0           3       BENTHIC FAUNA
   5        0           5       AVIFAUNA
   6        0           6       HUMANS

MAXIMUM =   25
```

```
MINIMUM AND NON-MINIMUM PATH CRITICAL LINKS          (CUT-COMPONENT SUMMARY)
-------------------------------------------          -----------------------

DEPENDENT                   SUPPORTING                  CUT-
COMPONENT                   COMPONENT                COMPONENT
---------                   ---------                ---------

 2 PLANT DETRITUS           3 BENTHIC FAUNA          1 ROOTED VEGETATION
 2 PLANT DETRITUS           4 ESTUARINE FISH         1 ROOTED VEGETATION
 2 PLANT DETRITUS           5 AVIFAUNA               1 ROOTED VEGETATION
 3 BENTHIC FAUNA            4 ESTUARINE FISH         1 ROOTED VEGETATION
 3 BENTHIC FAUNA            5 AVIFAUNA               1 ROOTED VEGETATION
 4 ESTUARINE FISH           5 AVIFAUNA               1 ROOTED VEGETATION
 6 HUMANS                   5 AVIFAUNA               1 ROOTED VEGETATION

 3 BENTHIC FAUNA            1 ROOTED VEGETATION      2 PLANT DETRITUS
 3 BENTHIC FAUNA            4 ESTUARINE FISH         2 PLANT DETRITUS
 3 BENTHIC FAUNA            5 AVIFAUNA               2 PLANT DETRITUS

 6 HUMANS                   1 ROOTED VEGETATION      4 ESTUARINE FISH
 6 HUMANS                   2 PLANT DETRITUS         4 ESTUARINE FISH
 6 HUMANS                   3 BENTHIC FAUNA          4 ESTUARINE FISH
 6 HUMANS                   5 AVIFAUNA               4 ESTUARINE FISH
```



DATE - 061881 TRACK NUMBER 1

ILLUSTRATIVE ESTUARINE SYSTEM

DEPENDENCE OF COMPONENT 5 - AVIFAUNA
ON COMPONENT 2 - PLANT DETRITUS

```
   COMPONENTS RANKED IN ORDER OF CUT POSITIONS HELD    (CUT-COMPONENT RANKING)
   ------------------------------------------------

   RANK       CUT-   COMPONENT   COMPONENT
         POSITIONS      NUMBER      LABEL
   ----  ---------   ---------   ---------
    1        7           1       ROOTED VEGETATION
    2        4           4       ESTUARINE FISH
    3        3           2       PLANT DETRITUS
    4        0           3       BENTHIC FAUNA
    5        0           5       AVIFAUNA
    6        0           6       HUMANS

MAXIMUM =   25
```

```
     COMPONENTS RANKED IN ORDER OF THEIR DISRUPTIVE
     POTENTIAL ON OTHER COMPONENTS DEPENDENT ON THEM
     -----------------------------------------------

     RANK  DISRUPTIVE   COMPONENT   COMPONENT
              MEASURE      NUMBER      LABEL
     ----  ----------   ---------   ---------
      1       4.4           1       ROOTED VEGETATION
      2       2.9           4       ESTUARINE FISH
      3       2.6           2       PLANT DETRITUS
      4        .6           5       AVIFAUNA
      5        .4           3       BENTHIC FAUNA
      6        .0           6       HUMANS


       (COMPONENT DISRUPTIVE POTENTIAL RANKING)
       ----------------------------------------
```

```
COMPONENTS RANKED IN ORDER OF THE DISRUPTIVE POTENTIAL
ON THEM FROM THE OTHER COMPONENTS UPON WHICH THEY DEPEND
--------------------------------------------------------

RANK  DISRUPTIVE   COMPONENT   COMPONENT
         MEASURE      NUMBER      LABEL
----  ----------   ---------   ---------
 1       3.3           1       ROOTED VEGETATION
 2       1.0           2       PLANT DETRITUS
 3        .6           4       ESTUARINE FISH
 4        .6           3       BENTHIC FAUNA
 5        .5           6       HUMANS
 6        .4           5       AVIFAUNA


     (COMPONENT SUSCEPTIBILITY TO DISRUPTION RANKING)
     ------------------------------------------------
```

9.1.2. Weighted illustrative estuarine system

- component interaction matrix
- valency rankings

```
                "WEIGHTED" ILLUSTRATIVE ESTUARINE SYSTEM
                ****************************************

                THE COMPONENTS ARE AS FOLLOWS:-

                 1  ROOTED VEGETATION
                 2  PLANT DETRITUS
                 3  BENTHIC FAUNA
                 4  ESTUARINE FISH
                 5  AVIFAUNA
                 6  HUMANS


            THE INITIAL COMPONENT INTERACTION MATRIX
            ----------------------------------------

                  1  2  3  4  5  6
               **********************
             1 *  0  1  1  1  1  0  *  1  ROOTED VEGETATION
             2 *  3  0  0  0  0  0  *  2  PLANT DETRITUS
             3 *  0  3  0  0  0  0  *  3  BENTHIC FAUNA
             4 *  1  2  3  0  0  0  *  4  ESTUARINE FISH
             5 *  0  0  2  2  0  0  *  5  AVIFAUNA
             6 *  0  0  0  1  0  0  *  6  HUMANS
               **********************
                  1  2  3  4  5  6

THE NUMBER OF COMPONENTS EACH COMPONENT DEPENDS UPON     (OUT-VALENCY RANKING)
----------------------------------------------------     ---------------------

RANK   LIMITED (1)   APPRECIABLE (2)  COMPLETE (3)   COMPONENT  COMPONENT
       DEPENDENCIES     DEPENDENCIES  DEPENDENCIES     NUMBER      LABEL
----   ------------  ---------------  ------------   ---------  ---------
  1         1               0              1             1      ROOTED VEGETATION
  2         1               1              1             2      PLANT DETRITUS
  3         1               1              1             3      BENTHIC FAUNA
  4         2               1              0             4      ESTUARINE FISH
  5         1               0              0             5      AVIFAUNA
  6         0               0              0             6      HUMANS

 MAXIMUM =  6               6              6

 **** THE COMPONENTS ARE RANKED IN ORDER OF THE NUMBER OF COMPLETE (3) DEPENDENCIES


THE NUMBER OF DEPENDENCIES SUPPORTED BY EACH COMPONENT     (IN-VALENCY RANKING)
------------------------------------------------------     --------------------

RANK   LIMITED (1)   APPRECIABLE (2)  COMPLETE (3)   COMPONENT  COMPONENT
       DEPENDENCIES     DEPENDENCIES  DEPENDENCIES     NUMBER      LABEL
----   ------------  ---------------  ------------   ---------  ---------
  1         0               0              1             2      PLANT DETRITUS
  2         0               0              1             3      BENTHIC FAUNA
  3         1               1              1             4      ESTUARINE FISH
  4         4               0              0             1      ROOTED VEGETATION
  5         0               2              0             5      AVIFAUNA
  6         1               0              0             6      HUMANS

 MAXIMUM =  6               6              6

 **** THE COMPONENTS ARE RANKED IN ORDER OF THE NUMBER OF COMPLETE (3) DEPENDENCIES
```

9.1.3. Programme / user commands

```
     * All data and computed information associated *
     * with this run will be available to you as a  *
     * hardcopy print-out listed under your userid. *
     * Enter a descriptive heading of up to fifty   *
     * characters for this print-out.               *

  ????....
 >

     * In order to enter the component matrix data *
     * type "1"  to assign a permanent data file   *
     * type "2"  to access a permanent data file   *
     * type "3"  for no permanent data record      *

  ????....
 >2

     * enter the name of the data file *

  ????....
 >nonexz
 **ERTRAN ERROR **  BAD STATUS:400010000000  FOR CALL FACSF
 @asg,a nonexz .

     * The above error message indicates why your *
     * file cannot be assigned to this run.       *
     *  type "1" to re-enter a data file name     *
     *  type "2" to use another way to enter data *
     *  type "3" to terminate the programme       *

     * In order to enter the component matrix data *
     * type "1"  to assign a permanent data file   *
     * type "2"  to access a permanent data file   *
     * type ."3"  for no permanent data record      *

  ????....
 >3

     * What is the dimension of the component matrix ? *
     * (enter an integer value between one and fifty)  *

  ????....
 >6

     * Note any mistakes you may make when entering  *
     * data.  You will be able to correct them later *


     * Enter the component labels (any combination  *
     * of between 1 and 20 alphanumeric characters) *


  component   1 ?
 >rooted vegetation

  component   2 ?
 >plant detritus
```

```
    * type "1" to enter the matrix data as zeroes or as ones *
    * type "2" to enter the dependencies on a weighted scale *

 ????....
>1

    * type "1" to enter the component interaction *
    *       matrix element by element.            *
    * type "2" to enter the component interaction *
    *       matrix row by row.                    *

 ????....
>1

    * To enter the elements of the component matrix  *
    * type "1" for a dependency between components   *
    * type "0" for no interaction between components *

**** interactions with rooted vegetation

  The dependency of rooted vegetation
                 on rooted vegetation
   ???...
>0

  The dependency of rooted vegetation
                 on plant detritus
   ???...
>1

  The dependency of rooted vegetation
                 on benthic fauna
   ???...
>1

  The dependency of rooted vegetation
                 on estuarine fish
   ???...
>1
**** interactions with plant detritus

  The dependency of plant detritus
                 on rooted vegetation
   ???...
>1

  The dependency of plant detritus
                 on plant detritus
   ???...
>0

  The dependency of plant detritus
                 on benthic fauna
   ???...
>99
**** Re-enter the last response.  It was faulty.

  The dependency of plant detritus
                 on benthic fauna
   ???...
>0
```

```
    * A copy of the component interaction matrix *
    * has been recorded on your printfile.       *


    * type "1" to edit or correct the matrix data   *
    * type "2" to compute the minimum link matrix   *
    * type "3" to do selected interaction traces    *
    * type "4" for measures of component importance *
    * type "5" to terminate the programme           *

 ????....
>1

    * type "1" to change an individual element   *
    * type "2" to change a component label       *
    * type "3" to delete a component(col & row) *
    * type "4" to add a component to the matrix *
    * type "5" to terminate the programme        *

 ????....
>1

    * Enter the row number, column number, and correct *
    * value of the matrix element to be corrected.     *
    *      (row,column,integer between zero and three) *

 ????....
>6,4,1
**** The dependency between  6 and  4 has been changed from  2 to  1

    * Enter row, column, integer between 0 & 3 (or "exit") *

 ????....
>exit

    * Further editing of any kind ?  (yes/no) *

 ????....
>no

    * Do you wish to revise the print-out heading ?  *
    * (yes/no).                                      *

 ????....
>no

    * type "1" to overwrite your data file with the *
    *         revised component interaction matrix. *
    * type "2" to write the revised component inter *
    *         -action matrix to a new file.         *
    * type "3" for no permanent copy of the revised *
    *         component interaction matrix.         *

 ????....
>1
```

```
    * type "1" to edit or correct the matrix data   *
    * type "2" to compute the minimum link matrix   *
    * type "3" to do selected interaction traces    *
    * type "4" for measures of component importance *
    * type "5" to terminate the programme           *

 ????....
>2

    * A copy of the minimum link matrix has been written *
    * to your printfile.                                 *


    * type "1" to edit or correct the matrix data   *
    * type "2" to compute the minimum link matrix   *
    * type "3" to do selected interaction traces    *
    * type "4" for measures of component importance *
    * type "5" to terminate the programme           *

 ????....
>3

    * All tracks are drawn as diagrams on the plotting   *
    * machine and are returned to you under your run-id. *


    * type the number of the dependent component *

 ????....
>6

    * type the number of the supporting component *

 ????....
>5

    * Do you wish to perform further tracks ?  (yes/no) *

 ????....
>no

    * type "1" to edit or correct the matrix data   *
    * type "2" to compute the minimum link matrix   *
    * type "3" to do selected interaction traces    *
    * type "4" for measures of component importance *
    * type "5" to terminate the programme           *

 ????....
>4

    * All rankings will be written to your printfile.  *


    * type "1" for the valency rankings of components *
    * type "2" to compute the average path rankings   *
    * type "3" to compute the minimum link rankings   *
    * type "4" to compute the critical path rankings  *
    * type "5" to compute the cut-component rankings  *
    * type "6" for the disruptive measure rankings    *

 ????....
>1
```

```
    * Do you wish to initiate a new run ?  (yes/no) *

 ????....
>yes

    * All data and computed information associated *
    * with this run will be available to you as a  *
    * hardcopy print-out listed under your userid. *
    * Enter a descriptive heading of up to fifty   *
    * characters for this print-out.               *

 ????....
>appendix b - weighted illustrative estuarine system
**** Re-enter the last response.  It was faulty.

    * All data and computed information associated *
    * with this run will be available to you as a  *
    * hardcopy print-out listed under your userid. *
    * Enter a descriptive heading of up to fifty   *
    * characters for this print-out.               *

 ????....
>appendix b - weighted CIM

    * In order to enter the component matrix data *
    * type "1"  to assign a permanent data file   *
    * type "2"  to access a permanent data file   *
    * type "3"  for no permanent data record      *

 ????....
>1

    * Enter a name (as any combination of 1 to 12 *
    * alphanumeric characters) for the data file. *

 ????....
>estuarine

    * A file of this name already exists under    *
    * your project-id.                            *
    * type "1" to enter another name for the file *
    * type "2" to over-write this existing file   *
    * type "3" to use another way to enter data   *
    * type "4" to terminate the programme         *

 ????....
>1

    * Enter a name (as any combination of 1 to 12 *
    * alphanumeric characters) for the data file. *

 ????....
>weighted
```

E.T.C.

9.2. Detailed Estuarine System

9.2.1. The Positive Component Interaction Matrix

* component interaction/minimum link matrix
* in-valency ranking
* in-average graph distance ranking
* in-minimum path accessibility measure ranking
* critical-component summary (abbreviated) and ranking
* cut-component summary and ranking
* in-disruptive measure ranking
* dependency track examples

```
"POSITIVE" ESTUARINE SYSTEM
********************************

THE COMPONENTS ARE AS FOLLOWS:-

 1   FRESHWATER (/FLOW)
 2   NUTRIENTS (/CYCLING)
 3   ESTUARINE SEDIMENTS
 4   RIVER BED
 5   RIVER BANKS
 6   RIVER MOUTH
 7   SEAWATER
 8   TIDES
 9   MARINE SEDIMENTS
10   CURRENTS & WAVES
11   BEACHES & DUNES
12   PHYTOPLANKTON
13   AQUATIC PLANTS
14   MARGINAL VEGETATION
15   TERRESTRIAL VEGATN
16   MICRO-ORGANISMS
17   ZOOPLANKTON
18   BENTHIC FAUNA
19   NEKTONIC FAUNA
20   TERRSTRL VERTEBRATES
21   MUDFLATS
22   MARSH/REEDBED/SWAMPS
23   DETRITUS
24   FLOODPLAIN/ISLANDS
25   TOTAL ESTUARY ZONE
26   W/S PRECIPITATION
27   W/S RUNOFF
28   W/S GROUNDWATER
29   W/S GEOLOGY/TOPOGPHY
30   W/S BIOMASS
31   W/S SOLAR RADIATION
32   W/S WIND
33   RESERVES & SITES
34   DAMS
35   PLANT COMMUNITIES
36   MINES
37   FARMS
38   AGRICULTURAL RUNOFF
39   INDUSTRIES
40   INDUSTRIAL EFFLUENTS
41   HOUSING
42   HOUSEHOLD SEWAGE
43   BUSINESSES
44   RECREATIONALISTS
45   SHIPS & BOATS
46   ROAD/AIRPORT/BRIDGES
47   CANALS & MARINAS
```

```
THE MINIMUM LINK MATRIX
-----------------------

        1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
    ***************************************************************************************************************************************************
 1  *   2  3  4  2  3  4  4  4  5  4  4  4  4  4  3  4  4  4  4  4  4  4  4  3  4  1  1  2  2  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  *  1 FRESHWATER (/FLOW)
 2  *   2  2  1  2  2  2  1  1  2  2  2  1  1  1  1  1  1  1  1  1  1  1  1  2  1  2  2  2  1  4  4  3  0  1  0  2  4  0  2  1  3  3  0  2  3  2  3  *  2 NUTRIENTS (/CYCLING)
 3  *   1  1  2  2  3  2  1  2  2  2  1  2  1  2  2  2  2  1  2  2  2  2  1  3  1  2  1  2  1  3  3  1  0  1  0  4  3  0  2  2  4  0  0  0  0  1  0  *  3 ESTUARINE SEDIMENTS
 4  *   1  3  3  2  2  2  2  3  4  3  3  4  4  4  3  4  4  3  3  3  4  3  4  2  3  2  1  2  2  4  4  2  0  2  0  5  4  0  2  3  5  0  0  0  0  2  0  *  4 RIVER BED
 5  *   2  3  3  2  2  2  2  3  4  3  3  4  4  4  3  4  4  3  3  3  4  3  4  2  3  3  3  3  3  2  2  2  0  2  0  3  2  0  2  3  3  0  0  0  1  3  0  *  5 RIVER BANKS
 6  *   1  3  4  3  3  2  4  1  5  4  4  4  4  4  3  4  4  4  4  4  4  4  4  3  4  2  1  2  2  3  3  2  0  2  0  4  3  0  3  3  4  0  0  0  0  1  0  *  6 RIVER MOUTH
 7  *   2  4  3  4  4  1  4  1  3  1  2  5  4  5  4  5  5  4  5  5  5  5  4  4  4  3  2  3  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  *  7 SEAWATER
 8  *   2  4  5  4  4  1  5  2  6  5  5  5  5  5  4  5  5  5  5  5  5  5  5  4  5  3  2  3  3  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  *  8 TIDES
 9  *   2  4  3  4  4  1  1  1  3  1  2  5  4  5  4  5  5  4  5  5  5  5  4  4  4  3  2  3  3  3  3  3  0  0  0  4  3  0  4  5  4  0  0  0  0  1  0  *  9 MARINE SEDIMENTS
10  *   3  3  2  4  5  3  3  2  2  2  1  4  3  4  4  4  4  3  4  4  4  4  3  5  3  4  3  4  3  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 10 CURRENTS & WAVES
11  *   2  2  1  3  4  2  2  1  1  1  2  3  2  3  3  3  3  2  3  3  3  3  2  4  2  3  2  3  2  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 11 BEACHES & DUNES
12  *   1  1  2  3  3  2  1  1  3  2  3  2  2  2  2  2  2  2  2  2  2  2  2  3  2  2  2  3  2  3  3  2  0  0  0  4  3  0  4  1  4  0  0  0  0  1  0  * 12 PHYTOPLANKTON
13  *   1  1  2  1  3  2  1  2  3  2  3  2  2  2  2  2  2  2  2  2  2  2  2  3  1  2  2  3  2  3  3  2  0  2  0  4  3  0  3  1  4  0  0  0  0  1  0  * 13 AQUATIC PLANTS
14  *   1  1  2  3  1  2  1  2  3  2  3  2  2  2  2  2  2  2  2  2  2  2  2  3  2  2  2  3  2  3  3  3  0  0  0  4  3  0  4  1  4  0  0  0  0  1  0  * 14 MARGINAL VEGETATION
15  *   1  1  2  2  2  3  2  2  3  3  3  2  2  2  2  2  2  2  2  2  2  2  2  1  2  2  2  3  2  3  3  3  0  0  0  1  3  0  4  1  4  0  0  0  0  1  0  * 15 TERRESTRIAL VEGATN
16  *   1  1  1  3  2  2  1  2  3  2  2  2  2  2  2  2  2  2  2  2  1  2  1  2  2  2  2  3  2  3  3  3  0  0  0  4  3  0  4  1  4  0  0  0  0  1  0  * 16 MICRO-ORGANISMS
17  *   1  1  2  2  3  2  1  1  3  2  3  1  2  2  2  1  2  2  2  2  2  1  1  3  1  2  2  3  2  3  3  2  0  0  0  4  3  0  4  1  4  1  0  0  0  1  0  * 17 ZOOPLANKTON
18  *   1  1  1  1  2  2  1  1  1  2  2  1  2  1  2  1  1  1  2  2  1  1  1  2  1  2  2  3  2  3  3  2  0  2  0  1  3  0  3  1  4  1  0  0  0  1  0  * 18 BENTHIC FAUNA
19  *   1  1  2  2  2  1  1  2  2  2  3  1  1  1  2  1  1  1  1  2  2  2  1  3  1  2  2  3  2  3  3  3  0  0  0  4  3  0  4  1  4  1  0  0  0  1  0  * 19 NEKTONIC FAUNA
20  *   1  1  1  2  1  2  2  2  2  2  1  2  1  1  1  2  1  1  1  2  1  1  2  2  1  2  2  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 20 TERRSTRL VERTEBRATES
21  *   1  2  1  2  2  2  2  1  2  3  2  2  2  2  2  1  2  1  2  2  2  2  1  2  1  2  2  3  2  3  3  2  0  2  0  4  3  0  2  4  2  2  0  1  2  1  2  * 21 MUDFLATS
22  *   1  1  1  2  1  2  1  1  3  2  2  2  2  2  2  1  2  2  2  2  2  2  2  1  1  2  2  3  2  4  4  2  0  2  0  5  4  0  1  3  5  0  0  0  0  2  0  * 22 MARSH/REEDBED/SWAMPS
23  *   2  2  2  2  2  2  2  2  2  2  2  1  1  1  1  2  1  1  1  1  2  1  2  2  2  2  2  2  2  4  4  4  0  0  0  5  4  0  5  2  5  0  0  0  0  2  0  * 23 DETRITUS
24  *   2  2  1  1  1  3  2  3  3  3  2  3  2  3  3  3  3  2  3  3  3  3  2  3  2  3  2  3  2  3  2  3  0  0  0  4  3  0  1  5  4  0  0  0  0  1  0  * 24 FLOODPLAIN/ISLANDS
25  *   1  2  1  1  2  1  1  1  2  2  2  2  2  2  2  2  2  1  1  2  1  1  2  2  2  2  2  3  2  3  3  2  0  2  0  1  3  1  1  1  2  1  0  1  2  1  1  * 25 TOTAL ESTUARY ZONE
26  *   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 26 W/S PRECIPITATION
27  *   1  2  3  3  2  3  3  3  4  3  3  3  3  3  2  3  3  3  3  3  3  3  3  2  3  1  2  1  1  1  1  3  0  1  0  2  1  0  2  2  2  0  0  0  0  2  0  * 27 W/S RUNOFF
28  *   1  3  4  3  3  4  4  4  5  4  4  4  4  4  3  4  4  4  4  4  4  4  4  3  4  1  1  2  1  1  3  3  0  0  0  1  2  0  2  4  2  3  0  0  0  1  0  * 28 W/S GROUNDWATER
29  *   2  3  4  4  3  4  4  4  5  4  4  4  4  4  3  4  4  4  4  4  4  4  4  3  4  1  1  1  2  1  1  1  0  0  0  2  1  0  2  3  2  0  0  0  0  2  0  * 29 W/S GEOLOGY/TOPOGPHY
30  *   1  1  2  3  3  3  2  2  3  3  3  2  2  2  2  2  2  2  2  2  2  2  2  2  2  1  2  2  1  3  3  3  0  0  0  1  1  0  1  3  1  0  0  0  0  1  0  * 30 W/S BIOMASS
31  *   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 31 W/S SOLAR RADIATION
32  *   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 32 W/S WIND
33  *   2  2  2  2  2  2  2  2  2  2  2  2  2  2  1  2  2  1  1  1  2  2  2  2  1  2  2  2  2  3  3  2  0  2  0  1  1  0  1  3  1  1  1  1  2  1  2  * 33 RESERVES & SITES
34  *   1  2  3  1  3  3  3  3  4  4  4  3  3  3  2  3  3  3  3  3  3  3  3  2  3  1  2  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 34 DAMS
35  *   1  1  2  3  3  3  2  2  3  3  3  2  2  2  1  2  2  2  2  2  2  2  2  2  2  1  1  1  1  3  3  1  0  0  0  1  1  0  1  3  1  0  0  0  0  2  0  * 35 PLANT COMMUNITIES
36  *   1  3  3  2  2  2  2  3  4  3  3  4  4  4  3  4  4  3  3  3  4  3  4  2  3  2  2  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 36 MINES
37  *   1  1  2  2  2  2  2  2  3  3  3  2  2  2  1  2  2  2  2  2  2  2  2  1  2  1  1  1  2  2  2  2  0  0  0  2  2  0  3  2  3  0  0  0  0  2  0  * 37 FARMS
38  *   1  2  3  3  3  4  3  3  4  4  4  3  3  3  3  3  3  3  3  3  3  3  3  3  3  1  1  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 38 AGRICULTURAL RUNOFF
39  *   1  3  3  2  2  2  2  3  4  3  3  4  4  4  3  4  4  3  3  3  4  3  4  2  3  2  2  3  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 39 INDUSTRIES
40  *   1  3  3  1  3  3  3  3  3  1  2  4  4  4  3  4  4  4  4  4  4  4  4  3  4  2  1  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 40 INDUSTRIAL EFFLUENTS
41  *   1  3  3  2  2  3  3  3  4  3  3  4  4  4  3  4  4  3  3  3  4  3  4  2  3  2  2  1  1  2  2  2  0  0  0  3  2  0  3  4  3  0  0  0  0  1  0  * 41 HOUSING
42  *   1  4  3  3  4  4  4  3  3  1  2  5  4  5  4  5  5  4  5  5  5  5  4  4  4  2  2  3  3  2  2  2  0  0  0  3  2  0  3  2  3  0  0  0  0  2  0  * 42 HOUSEHOLD SEWAGE
43  *   1  2  3  2  2  2  2  2  3  2  2  3  3  3  2  3  3  2  2  2  3  2  3  2  2  2  2  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  * 43 BUSINESSES
44  *   1  2  2  2  1  1  2  1  2  1  1  2  2  2  1  2  2  1  1  1  2  1  2  1  1  2  2  2  2  3  3  1  0  1  0  4  3  0  1  3  1  1  0  0  1  1  1  * 44 RECREATIONALISTS
45  *   1  3  3  1  2  1  1  2  3  2  2  3  3  3  2  3  3  2  2  2  3  2  3  2  2  2  2  3  2  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  * 45 SHIPS & BOATS
46  *   2  2  2  2  1  2  2  2  3  2  2  3  3  3  2  3  3  2  2  2  3  2  3  1  2  2  2  2  2  2  2  2  0  0  0  3  2  0  3  4  3  0  0  0  0  3  0  * 46 ROAD/AIRPORT/BRIDGES
47  *   1  2  3  1  1  1  1  2  3  2  2  3  3  3  2  3  3  2  2  2  3  2  3  2  2  2  2  2  3  3  3  3  0  0  0  4  3  0  4  3  4  1  0  0  0  3  0  * 47 CANALS & MARINAS
    ***************************************************************************************************************************************************
        1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
```

```
THE NUMBER OF DEPENDENCIES SUPPORTED BY EACH COMPONENT
------------------------------------------------------

RANK  DEPENDENCIES   COMPONENT   COMPONENT
        SUPPORTED     NUMBER       LABEL
----  ------------  ---------   ---------
  1        31            1      FRESHWATER (/FLOW)
  2        14            2      NUTRIENTS (/CYCLING)
  3        14            7      SEAWATER
  4        13           31      W/S SOLAR RADIATION
  5        12            8      TIDES
  6        11           25      TOTAL ESTUARY ZONE
  7        11           46      ROAD/AIRPORT/BRIDGES
  8        10           29      W/S GEOLOGY/TOPOGPHY
  9        10           18      BENTHIC FAUNA
 10        10           43      BUSINESSES
 11        10           27      W/S RUNOFF
 12         9           26      W/S PRECIPITATION
 13         9            3      ESTUARINE SEDIMENTS
 14         8            4      RIVER BED
 15         8            6      RIVER MOUTH
 16         7           19      NEKTONIC FAUNA
 17         7           22      MARSH/REEDBED/SWAMPS
 18         7           23      DETRITUS
 19         7           37      FARMS
 20         7           15      TERRESTRIAL VEGATN
 21         7            5      RIVER BANKS
 22         7           47      CANALS & MARINAS
 23         6           16      MICRO-ORGANISMS
 24         6           39      INDUSTRIES
 25         6           10      CURRENTS & WAVES
 26         6           44      RECREATIONALISTS
 27         6           32      W/S WIND
 28         6           34      DAMS
 29         5           24      FLOODPLAIN/ISLANDS
 30         5           13      AQUATIC PLANTS
 31         5           35      PLANT COMMUNITIES
 32         5           20      TERRSTRL VERTEBRATES
 33         5           21      MUDFLATS
 34         5           41      HOUSING
 35         5           28      W/S GROUNDWATER
 36         5           17      ZOOPLANKTON
 37         5           45      SHIPS & BOATS
 38         5           30      W/S BIOMASS
 39         5           12      PHYTOPLANKTON
 40         4           11      BEACHES & DUNES
 41         4           14      MARGINAL VEGETATION
 42         3           42      HOUSEHOLD SEWAGE
 43         3           38      AGRICULTURAL RUNOFF
 44         2           33      RESERVES & SITES
 45         2           36      MINES
 46         2            9      MARINE SEDIMENTS
 47         0           40      INDUSTRIAL EFFLUENTS

MAXIMUM = 47
```

(IN-VALENCY RANKING)

```
THE AVERAGE NUMBER OF LINKS PER CONNECTION BETWEEN
A COMPONENT AND OTHERS WHICH ARE DEPENDENT ON IT
--------------------------------------------------

RANK  AVE. NO.   COMPONENT   COMPONENT
      OF LINKS    NUMBER       LABEL
----  --------   ---------   ---------
  1     3.2         33       RESERVES & SITES
  2     3.1          9       MARINE SEDIMENTS
  3     3.0         42       HOUSEHOLD SEWAGE
  4     2.9         14       MARGINAL VEGETATION
  5     2.9         21       MUDFLATS
  6     2.9         30       W/S BIOMASS
  7     2.9         12       PHYTOPLANKTON
  8     2.9         17       ZOOPLANKTON
  9     2.9         45       SHIPS & BOATS
 10     2.8         16       MICRO-ORGANISMS
 11     2.8         38       AGRICULTURAL RUNOFF
 12     2.7         35       PLANT COMMUNITIES
 13     2.7         13       AQUATIC PLANTS
 14     2.7         23       DETRITUS
 15     2.7         20       TERRSTRL VERTEBRATES
 16     2.6         22       MARSH/REEDBED/SWAMPS
 17     2.6         19       NEKTONIC FAUNA
 18     2.6         11       BEACHES & DUNES
 19     2.6         36       MINES
 20     2.6         41       HOUSING
 21     2.5         39       INDUSTRIES
 22     2.5         24       FLOODPLAIN/ISLANDS
 23     2.5         10       CURRENTS & WAVES
 24     2.5         44       RECREATIONALISTS
 25     2.5         43       BUSINESSES
 26     2.5         47       CANALS & MARINAS
 27     2.4          5       RIVER BANKS
 28     2.4         18       BENTHIC FAUNA
 29     2.4          3       ESTUARINE SEDIMENTS
 30     2.4         25       TOTAL ESTUARY ZONE
 31     2.4         28       W/S GROUNDWATER
 32     2.4         32       W/S WIND
 33     2.4         15       TERRESTRIAL VEGATN
 34     2.3          4       RIVER BED
 35     2.2          6       RIVER MOUTH
 36     2.2          7       SEAWATER
 37     2.2         37       FARMS
 38     2.1          2       NUTRIENTS (/CYCLING)
 39     2.1          8       TIDES
 40     2.1         34       DAMS
 41     2.0         46       ROAD/AIRPORT/BRIDGES
 42     2.0         26       W/S PRECIPITATION
 43     2.0         31       W/S SOLAR RADIATION
 44     2.0         29       W/S GEOLOGY/TOPOGPHY
 45     1.8         27       W/S RUNOFF
 46     1.3          1       FRESHWATER (/FLOW)
 47      .0         40       INDUSTRIAL EFFLUENTS

MAXIMUM = 24.0
```

(AVERAGE IN-GRAPH DISTANCE RANKING)

THE NUMBER OF MINIMUM PATHS LINKING A COMPONENT TO ALL OTHERS WHICH ARE DEPENDENT ON IT

(IN-MINIMUM PATH ACCESSIBILITY RANKING)

RANK	MINIMUM PATHS	COMPONENT NUMBER	COMPONENT LABEL
1	146	45	SHIPS & BOATS
2	136	47	CANALS & MARINAS
3	132	21	MUDFLATS
4	126	16	MICRO-ORGANISMS
5	126	24	FLOODPLAIN/ISLANDS
6	125	17	ZOOPLANKTON
7	122	5	RIVER BANKS
8	122	41	HOUSING
9	120	31	W/S SOLAR RADIATION
10	119	43	BUSINESSES
11	116	28	W/S GROUNDWATER
12	115	37	FARMS
13	113	42	HOUSEHOLD SEWAGE
14	112	4	RIVER BED
15	111	6	RIVER MOUTH
16	109	12	PHYTOPLANKTON
17	107	22	MARSH/REEDBED/SWAMPS
18	106	38	AGRICULTURAL RUNOFF
19	104	23	DETRITUS
20	101	8	TIDES
21	101	11	BEACHES & DUNES
22	100	7	SEAWATER
23	100	3	ESTUARINE SEDIMENTS
24	100	13	AQUATIC PLANTS
25	99	14	MARGINAL VEGETATION
26	99	2	NUTRIENTS (/CYCLING)
27	98	32	W/S WIND
28	98	9	MARINE SEDIMENTS
29	98	30	W/S BIOMASS
30	98	10	CURRENTS & WAVES
31	96	35	PLANT COMMUNITIES
32	96	44	RECREATIONALISTS
33	94	39	INDUSTRIES
34	93	33	RESERVES & SITES
35	92	19	NEKTONIC FAUNA
36	92	46	ROAD/AIRPORT/BRIDGES
37	92	34	DAMS
38	87	1	FRESHWATER (/FLOW)
39	86	26	W/S PRECIPITATION
40	83	27	W/S RUNOFF
41	83	15	TERRESTRIAL VEGATN
42	81	20	TERRSTRL VERTEBRATES
43	81	29	W/S GEOLOGY/TOPOGPHY
44	77	25	TOTAL ESTUARY ZONE
45	74	36	MINES
46	72	18	BENTHIC FAUNA
47	0	40	INDUSTRIAL EFFLUENTS

COMPONENTS RANKED IN ORDER OF CRITICAL PATHS SUPPORTED

(CRITICAL IMPORTANCE RANKING)

RANK	CRITICAL PATHS	COMPONENT NUMBER	COMPONENT LABEL
1	262	2	NUTRIENTS (/CYCLING)
2	172	27	W/S RUNOFF
3	141	3	ESTUARINE SEDIMENTS
4	139	37	FARMS
5	138	44	RECREATIONALISTS
6	89	46	ROAD/AIRPORT/BRIDGES
7	78	6	RIVER MOUTH
8	72	11	BEACHES & DUNES
9	69	1	FRESHWATER (/FLOW)
10	35	10	CURRENTS & WAVES
11	22	34	DAMS
12	19	30	W/S BIOMASS
13	16	18	BENTHIC FAUNA
14	15	25	TOTAL ESTUARY ZONE
15	14	47	CANALS & MARINAS
16	13	20	TERRSTRL VERTEBRATES
17	10	36	MINES
18	9	45	SHIPS & BOATS
19	9	24	FLOODPLAIN/ISLANDS
20	9	8	TIDES
21	8	23	DETRITUS
22	8	7	SEAWATER
23	7	5	RIVER BANKS
24	7	4	RIVER BED
25	5	32	W/S WIND
26	5	22	MARSH/REEDBED/SWAMPS
27	5	35	PLANT COMMUNITIES
28	4	39	INDUSTRIES
29	3	26	W/S PRECIPITATION
30	3	41	HOUSING
31	2	19	NEKTONIC FAUNA
32	2	21	MUDFLATS
33	2	28	W/S GROUNDWATER
34	2	29	W/S GEOLOGY/TOPOGPHY
35	1	14	MARGINAL VEGETATION
36	1	15	TERRESTRIAL VEGATN
37	1	33	RESERVES & SITES
38	1	42	HOUSEHOLD SEWAGE
39	0	16	MICRO-ORGANISMS
40	0	40	INDUSTRIAL EFFLUENTS
41	0	17	ZOOPLANKTON
42	0	13	AQUATIC PLANTS
43	0	43	BUSINESSES
44	0	9	MARINE SEDIMENTS
45	0	12	PHYTOPLANKTON
46	0	31	W/S SOLAR RADIATION
47	0	38	AGRICULTURAL RUNOFF

MAXIMUM = 2116

```
THE MINIMUM PATH CRITICAL LINKS                          (CRITICAL COMPONENT SUMMARY)
-------------------------------                          ----------------------------

DEPENDENT                 SUPPORTING                 CRITICAL
COMPONENT                  COMPONENT                 COMPONENT
---------                 ----------                 ---------

 1 FRESHWATER (/FLOW)       2 NUTRIENTS (/CYCLING)   37 FARMS
 1 FRESHWATER (/FLOW)       4 RIVER BED              34 DAMS
 1 FRESHWATER (/FLOW)       5 RIVER BANKS            46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)       5 RIVER BANKS            27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      10 CURRENTS & WAVES       44 RECREATIONALISTS
 1 FRESHWATER (/FLOW)      10 CURRENTS & WAVES       46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)      10 CURRENTS & WAVES       27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      11 BEACHES & DUNES        44 RECREATIONALISTS
 1 FRESHWATER (/FLOW)      11 BEACHES & DUNES        46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)      11 BEACHES & DUNES        27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      12 PHYTOPLANKTON           2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      12 PHYTOPLANKTON          37 FARMS
 1 FRESHWATER (/FLOW)      13 AQUATIC PLANTS          2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      13 AQUATIC PLANTS         37 FARMS
 1 FRESHWATER (/FLOW)      14 MARGINAL VEGETATION     2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      14 MARGINAL VEGETATION    37 FARMS
 1 FRESHWATER (/FLOW)      15 TERRESTRIAL VEGATN     37 FARMS
 1 FRESHWATER (/FLOW)      16 MICRO-ORGANISMS         2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      16 MICRO-ORGANISMS        37 FARMS
 1 FRESHWATER (/FLOW)      17 ZOOPLANKTON             2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      17 ZOOPLANKTON            37 FARMS
 1 FRESHWATER (/FLOW)      21 MUDFLATS                2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      21 MUDFLATS               37 FARMS
 1 FRESHWATER (/FLOW)      23 DETRITUS                2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      23 DETRITUS               37 FARMS
 1 FRESHWATER (/FLOW)      28 W/S GROUNDWATER        27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      30 W/S BIOMASS             2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      30 W/S BIOMASS            37 FARMS
 1 FRESHWATER (/FLOW)      31 W/S SOLAR RADIATION    26 W/S PRECIPITATION
 1 FRESHWATER (/FLOW)      33 RESERVES & SITES       44 RECREATIONALISTS
 1 FRESHWATER (/FLOW)      33 RESERVES & SITES       46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)      33 RESERVES & SITES       27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      35 PLANT COMMUNITIES      44 RECREATIONALISTS
 1 FRESHWATER (/FLOW)      35 PLANT COMMUNITIES      46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)      35 PLANT COMMUNITIES      27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      36 MINES                  46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)      36 MINES                  27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      38 AGRICULTURAL RUNOFF     2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      38 AGRICULTURAL RUNOFF    37 FARMS
 1 FRESHWATER (/FLOW)      39 INDUSTRIES             34 DAMS
 1 FRESHWATER (/FLOW)      41 HOUSING                34 DAMS
 1 FRESHWATER (/FLOW)      42 HOUSEHOLD SEWAGE        2 NUTRIENTS (/CYCLING)
 1 FRESHWATER (/FLOW)      42 HOUSEHOLD SEWAGE       37 FARMS
 1 FRESHWATER (/FLOW)      44 RECREATIONALISTS       46 ROAD/AIRPORT/BRIDGES
 1 FRESHWATER (/FLOW)      44 RECREATIONALISTS       27 W/S RUNOFF
 1 FRESHWATER (/FLOW)      45 SHIPS & BOATS          39 INDUSTRIES
 1 FRESHWATER (/FLOW)      45 SHIPS & BOATS          34 DAMS
 1 FRESHWATER (/FLOW)      46 ROAD/AIRPORT/BRIDGES   27 W/S RUNOFF

 2 NUTRIENTS (/CYCLING)     9 MARINE SEDIMENTS       18 BENTHIC FAUNA
 2 NUTRIENTS (/CYCLING)    28 W/S GROUNDWATER        29 W/S GEOLOGY/TOPOGPHY
 2 NUTRIENTS (/CYCLING)    32 W/S WIND                8 TIDES
 2 NUTRIENTS (/CYCLING)    33 RESERVES & SITES       20 TERRSTRL VERTEBRATES
```

```
MINIMUM AND NON-MINIMUM PATH CRITICAL LINKS      (CUT-COMPONENT SUMMARY)
-------------------------------------------      ----------------------

DEPENDENT                   SUPPORTING                  CUT-
COMPONENT                    COMPONENT               COMPONENT
---------                   ----------               ---------

 8 TIDES                     1 FRESHWATER (/FLOW)     6 RIVER MOUTH
 8 TIDES                     2 NUTRIENTS (/CYCLING)   6 RIVER MOUTH
 8 TIDES                     3 ESTUARINE SEDIMENTS    6 RIVER MOUTH
 8 TIDES                     4 RIVER BED              6 RIVER MOUTH
 8 TIDES                     5 RIVER BANKS            6 RIVER MOUTH
 8 TIDES                     7 SEAWATER               6 RIVER MOUTH
 8 TIDES                     9 MARINE SEDIMENTS       6 RIVER MOUTH
 8 TIDES                    10 CURRENTS & WAVES       6 RIVER MOUTH
 8 TIDES                    11 BEACHES & DUNES        6 RIVER MOUTH
 8 TIDES                    12 PHYTOPLANKTON          6 RIVER MOUTH
 8 TIDES                    13 AQUATIC PLANTS         6 RIVER MOUTH
 8 TIDES                    14 MARGINAL VEGETATION    6 RIVER MOUTH
 8 TIDES                    15 TERRESTRIAL VEGATN     6 RIVER MOUTH
 8 TIDES                    16 MICRO-ORGANISMS        6 RIVER MOUTH
 8 TIDES                    17 ZOOPLANKTON            6 RIVER MOUTH
 8 TIDES                    18 BENTHIC FAUNA          6 RIVER MOUTH
 8 TIDES                    19 NEKTONIC FAUNA         6 RIVER MOUTH
 8 TIDES                    20 TERRSTRL VERTEBRATES   6 RIVER MOUTH
 8 TIDES                    21 MUDFLATS               6 RIVER MOUTH
 8 TIDES                    22 MARSH/REEDBED/SWAMPS   6 RIVER MOUTH
 8 TIDES                    23 DETRITUS               6 RIVER MOUTH
 8 TIDES                    24 FLOODPLAIN/ISLANDS     6 RIVER MOUTH
 8 TIDES                    25 TOTAL ESTUARY ZONE-    6 RIVER MOUTH
 8 TIDES                    26 W/S PRECIPITATION      6 RIVER MOUTH
 8 TIDES                    27 W/S RUNOFF             6 RIVER MOUTH
 8 TIDES                    28 W/S GROUNDWATER        6 RIVER MOUTH
 8 TIDES                    29 W/S GEOLOGY/TOPOGPHY   6 RIVER MOUTH
 8 TIDES                    30 W/S BIOMASS            6 RIVER MOUTH
 8 TIDES                    33 RESERVES & SITES       6 RIVER MOUTH
 8 TIDES                    34 DAMS                   6 RIVER MOUTH
 8 TIDES                    35 PLANT COMMUNITIES      6 RIVER MOUTH
 8 TIDES                    36 MINES                  6 RIVER MOUTH
 8 TIDES                    37 FARMS                  6 RIVER MOUTH
 8 TIDES                    38 AGRICULTURAL RUNOFF    6 RIVER MOUTH
 8 TIDES                    39 INDUSTRIES             6 RIVER MOUTH
 8 TIDES                    41 HOUSING                6 RIVER MOUTH
 8 TIDES                    42 HOUSEHOLD SEWAGE       6 RIVER MOUTH
 8 TIDES                    43 BUSINESSES             6 RIVER MOUTH
 8 TIDES                    44 RECREATIONALISTS       6 RIVER MOUTH
 8 TIDES                    45 SHIPS & BOATS          6 RIVER MOUTH
 8 TIDES                    46 ROAD/AIRPORT/BRIDGES   6 RIVER MOUTH
 8 TIDES                    47 CANALS & MARINAS       6 RIVER MOUTH
```

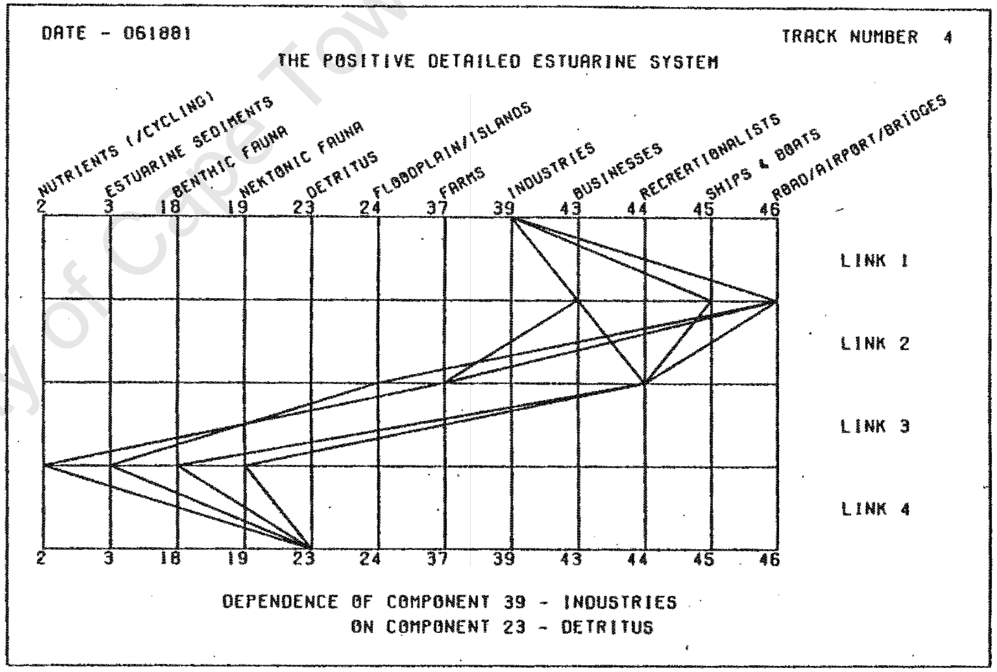
```
DEPENDENT                  SUPPORTING                  CUT-
COMPONENT                   COMPONENT               COMPONENT
---------                  ----------               ---------
10 CURRENTS & WAVES         1 FRESHWATER (/FLOW)    11 BEACHES & DUNES
10 CURRENTS & WAVES         2 NUTRIENTS (/CYCLING)  11 BEACHES & DUNES
10 CURRENTS & WAVES         3 ESTUARINE SEDIMENTS   11 BEACHES & DUNES
10 CURRENTS & WAVES         4 RIVER BED             11 BEACHES & DUNES
10 CURRENTS & WAVES         5 RIVER BANKS           11 BEACHES & DUNES
10 CURRENTS & WAVES         6 RIVER MOUTH           11 BEACHES & DUNES
10 CURRENTS & WAVES         7 SEAWATER              11 BEACHES & DUNES
10 CURRENTS & WAVES         8 TIDES                 11 BEACHES & DUNES
10 CURRENTS & WAVES         9 MARINE SEDIMENTS      11 BEACHES & DUNES
10 CURRENTS & WAVES        12 PHYTOPLANKTON         11 BEACHES & DUNES
10 CURRENTS & WAVES        13 AQUATIC PLANTS        11 BEACHES & DUNES
10 CURRENTS & WAVES        14 MARGINAL VEGETATION   11 BEACHES & DUNES
10 CURRENTS & WAVES        15 TERRESTRIAL VEGATN    11 BEACHES & DUNES
10 CURRENTS & WAVES        16 MICRO-ORGANISMS       11 BEACHES & DUNES
10 CURRENTS & WAVES        17 ZOOPLANKTON           11 BEACHES & DUNES
10 CURRENTS & WAVES        18 BENTHIC FAUNA         11 BEACHES & DUNES
10 CURRENTS & WAVES        19 NEKTONIC FAUNA        11 BEACHES & DUNES
10 CURRENTS & WAVES        20 TERRSTRL VERTEBRATES  11 BEACHES & DUNES
10 CURRENTS & WAVES        21 MUDFLATS              11 BEACHES & DUNES
10 CURRENTS & WAVES        22 MARSH/REEDBED/SWAMPS  11 BEACHES & DUNES
10 CURRENTS & WAVES        23 DETRITUS              11 BEACHES & DUNES
10 CURRENTS & WAVES        24 FLOODPLAIN/ISLANDS    11 BEACHES & DUNES
10 CURRENTS & WAVES        25 TOTAL ESTUARY ZONE    11 BEACHES & DUNES
10 CURRENTS & WAVES        26 W/S PRECIPITATION     11 BEACHES & DUNES
10 CURRENTS & WAVES        27 W/S RUNOFF            11 BEACHES & DUNES
10 CURRENTS & WAVES        28 W/S GROUNDWATER       11 BEACHES & DUNES
10 CURRENTS & WAVES        29 W/S GEOLOGY/TOPOGPHY  11 BEACHES & DUNES
10 CURRENTS & WAVES        30 W/S BIOMASS           11 BEACHES & DUNES
10 CURRENTS & WAVES        33 RESERVES & SITES      11 BEACHES & DUNES
10 CURRENTS & WAVES        34 DAMS                  11 BEACHES & DUNES
10 CURRENTS & WAVES        35 PLANT COMMUNITIES     11 BEACHES & DUNES
10 CURRENTS & WAVES        36 MINES                 11 BEACHES & DUNES
10 CURRENTS & WAVES        37 FARMS                 11 BEACHES & DUNES
10 CURRENTS & WAVES        38 AGRICULTURAL RUNOFF   11 BEACHES & DUNES
10 CURRENTS & WAVES        39 INDUSTRIES            11 BEACHES & DUNES
10 CURRENTS & WAVES        41 HOUSING               11 BEACHES & DUNES
10 CURRENTS & WAVES        42 HOUSEHOLD SEWAGE      11 BEACHES & DUNES
10 CURRENTS & WAVES        43 BUSINESSES            11 BEACHES & DUNES
10 CURRENTS & WAVES        44 RECREATIONALISTS      11 BEACHES & DUNES
10 CURRENTS & WAVES        45 SHIPS & BOATS         11 BEACHES & DUNES
10 CURRENTS & WAVES        46 ROAD/AIRPORT/BRIDGES  11 BEACHES & DUNES
10 CURRENTS & WAVES        47 CANALS & MARINAS      11 BEACHES & DUNES
```

COMPONENTS RANKED IN ORDER OF CUT POSITIONS HELD

(CUT-COMPONENT RANKING)

RANK	CUT-POSITIONS	COMPONENT NUMBER	COMPONENT LABEL
1	42	6	RIVER MOUTH
2	42	11	BEACHES & DUNES
3	0	3	ESTUARINE SEDIMENTS
4	0	4	RIVER BED
5	0	5	RIVER BANKS
6	0	1	FRESHWATER (/FLOW)
7	0	7	SEAWATER
8	0	8	TIDES
9	0	9	MARINE SEDIMENTS
10	0	10	CURRENTS & WAVES
11	0	2	NUTRIENTS (/CYCLING)
12	0	12	PHYTOPLANKTON
13	0	13	AQUATIC PLANTS
14	0	14	MARGINAL VEGETATION
15	0	15	TERRESTRIAL VEGATN
16	0	16	MICRO-ORGANISMS
17	0	17	ZOOPLANKTON
18	0	18	BENTHIC FAUNA
19	0	19	NEKTONIC FAUNA
20	0	20	TERRSTRL VERTEBRATES
21	0	21	MUDFLATS
22	0	22	MARSH/REEDBED/SWAMPS
23	0	23	DETRITUS
24	0	24	FLOODPLAIN/ISLANDS
25	0	25	TOTAL ESTUARY ZONE
26	0	26	W/S PRECIPITATION
27	0	27	W/S RUNOFF
28	0	28	W/S GROUNDAATER
29	0	29	W/S GEOLOGY/TOPOGPHY
30	0	30	W/S BIOMASS
31	0	31	W/S SOLAR RADIATION
32	0	32	W/S WIND
33	0	33	RESERVES & SITES
34	0	34	DAMS
35	0	35	PLANT COMMUNITIES
36	0	36	MINES
37	0	37	FARMS
38	0	38	AGRICULTURAL RUNOFF
39	0	39	INDUSTRIES
40	0	40	INDUSTRIAL EFFLUENTS
41	0	41	HOUSING
42	0	42	HOUSEHOLD SEWAGE
43	0	43	BUSINESSES
44	0	44	RECREATIONALISTS
45	0	45	SHIPS & BOATS
46	0	46	ROAD/AIRPORT/BRIDGES
47	0	47	CANALS & MARINAS

MAXIMUM = 2116



DEPENDENCE OF COMPONENT 39 - INDUSTRIES
ON COMPONENT 23 - DETRITUS

COMPONENTS RANKED IN ORDER OF THE DISRUPTIVE POTENTIAL ON THEM FROM THE OTHER COMPONENTS UPON WHICH THEY DEPEND

(COMPONENT SUSCEPTIBILITY TO DISRUPTION RANKING)

RANK	DISRUPTIVE MEASURE	COMPONENT NUMBER	COMPONENT LABEL
1	11.0	2	NUTRIENTS (/CYCLING)
2	6.8	3	ESTUARINE SEDIMENTS
3	4.9	27	W/S RUNOFF
4	4.7	44	RECREATIONALISTS
5	4.6	37	FARMS
6	3.1	46	ROAD/AIRPORT/BRIDGES
7	2.9	6	RIVER MOUTH
8	2.9	1	FRESHWATER (/FLOW)
9	2.8	11	BEACHES & DUNES
10	2.5	10	CURRENTS & WAVES
11	2.1	8	TIDES
12	1.7	7	SEAWATER
13	1.5	9	MARINE SEDIMENTS
14	1.4	42	HOUSEHOLD SEWAGE
15	1.2	20	TERRSTRL VERTEBRATES
16	1.2	18	BENTHIC FAUNA
17	1.2	34	DAMS
18	1.2	29	W/S GEOLOGY/TOPOGPHY
19	1.2	25	TOTAL ESTUARY ZONE
20	1.1	28	W/S GROUNDWATER
21	1.0	30	W/S BIOMASS
22	1.0	23	DETRITUS
23	1.0	36	MINES
24	1.0	5	RIVER BANKS
25	1.0	47	CANALS & MARINAS
26	.9	41	HOUSING
27	.9	4	RIVER BED
28	.8	40	INDUSTRIAL EFFLUENTS
29	.8	45	SHIPS & BOATS
30	.8	39	INDUSTRIES
31	.8	19	NEKTONIC FAUNA
32	.8	22	MARSH/REEDBED/SWAMPS
33	.8	35	PLANT COMMUNITIES
34	.8	24	FLOODPLAIN/ISLANDS
35	.7	33	RESERVES & SITES
36	.7	21	MUDFLATS
37	.7	12	PHYTOPLANKTON
38	.7	15	TERRESTRIAL VEGATN
39	.6	16	MICRO-ORGANISMS
40	.6	38	AGRICULTURAL RUNOFF
41	.6	17	ZOOPLANKTON
42	.6	14	MARGINAL VEGETATION
43	.6	13	AQUATIC PLANTS
44	.5	43	BUSINESSES
45	.2	32	W/S WIND
46	.1	26	W/S PRECIPITATION
47	.0	31	W/S SOLAR RADIATION

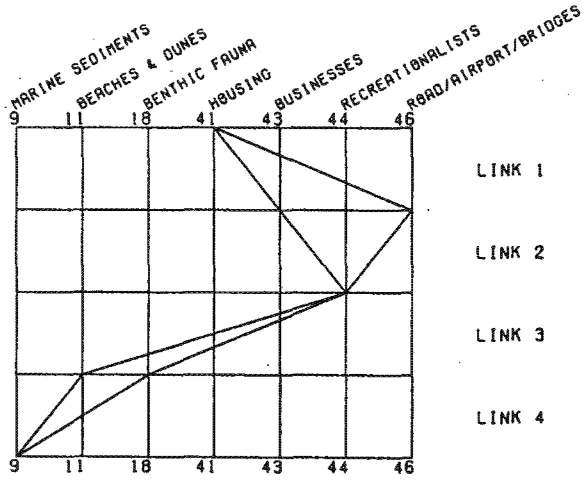
COMPONENTS RANKED IN ORDER OF THEIR DISRUPTIVE POTENTIAL ON OTHER COMPONENTS DEPENDENT ON THEM

(COMPONENT DISRUPTIVE POTENTIAL RANKING)

RANK	DISRUPTIVE MEASURE	COMPONENT NUMBER	COMPONENT LABEL
1	10.0	2	NUTRIENTS (/CYCLING)
2	6.4	3	ESTUARINE SEDIMENTS
3	5.4	27	W/S RUNOFF
4	4.4	6	RIVER MOUTH
5	4.4	37	FARMS
6	4.0	44	RECREATIONALISTS
7	3.5	11	BEACHES & DUNES
8	3.1	46	ROAD/AIRPORT/BRIDGES
9	3.1	1	FRESHWATER (/FLOW)
10	2.6	31	W/S SOLAR RADIATION
11	1.9	10	CURRENTS & WAVES
12	1.1	30	W/S BIOMASS
13	1.1	34	DAMS
14	1.0	47	CANALS & MARINAS
15	1.0	45	SHIPS & BOATS
16	.9	32	W/S WIND
17	.9	9	MARINE SEDIMENTS
18	.9	25	TOTAL ESTUARY ZONE
19	.8	8	TIDES
20	.8	7	SEAWATER
21	.8	39	INDUSTRIES
22	.8	23	DETRITUS
23	.8	41	HOUSING
24	.8	18	BENTHIC FAUNA
25	.8	33	RESERVES & SITES
26	.7	28	W/S GROUNDWATER
27	.7	20	TERRSTRL VERTEBRATES
28	.7	4	RIVER BED
29	.7	26	W/S PRECIPITATION
30	.7	43	BUSINESSES
31	.7	36	MINES
32	.7	5	RIVER BANKS
33	.7	24	FLOODPLAIN/ISLANDS
34	.7	42	HOUSEHOLD SEWAGE
35	.7	21	MUDFLATS
36	.7	16	MICRO-ORGANISMS
37	.7	29	W/S GEOLOGY/TOPOGPHY
38	.6	12	PHYTOPLANKTON
39	.6	14	MARGINAL VEGETATION
40	.6	35	PLANT COMMUNITIES
41	.6	17	ZOOPLANKTON
42	.6	38	AGRICULTURAL RUNOFF
43	.6	22	MARSH/REEDBED/SWAMPS
44	.6	13	AQUATIC PLANTS
45	.5	19	NEKTONIC FAUNA
46	.3	15	TERRESTRIAL VEGATN
47	.0	40	INDUSTRIAL EFFLUENTS

DATE - 061881 TRACK NUMBER 2

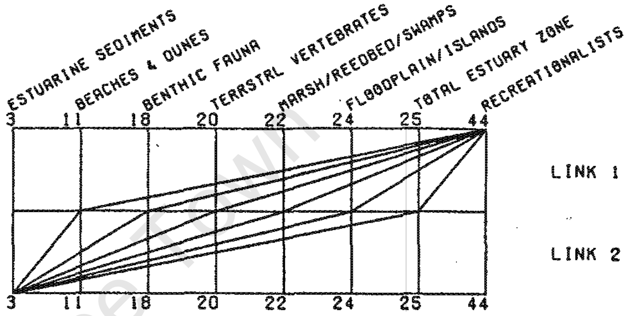
THE POSITIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 41 - HOUSING
ON COMPONENT 9 - MARINE SEDIMENTS

DATE - 061881 TRACK NUMBER 3

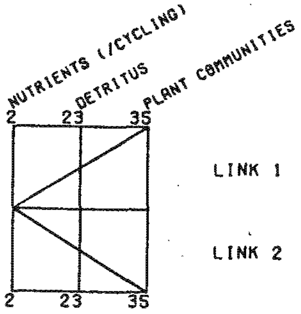
THE POSITIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 44 - RECREATIONALISTS
ON COMPONENT 3 - ESTUARINE SEDIMENTS

DATE - 061881 TRACK NUMBER 1

THE POSITIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 35 - PLANT COMMUNITIES
ON COMPONENT 23 - DETRITUS

9.2.2. The Negative Component Interaction Matrix

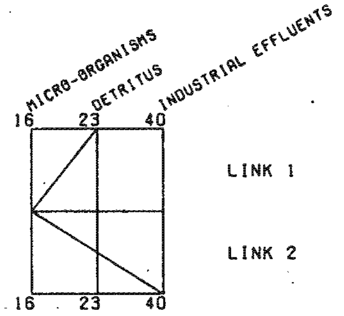
- component interaction/minimum link matrix
- dependency track examples

```
"POSITIVE" ESTUARINE SYSTEM
*******************************

THE COMPONENTS ARE AS FOLLOWS:-

 1   FRESHWATER (/FLOW)
 2   NUTRIENTS (/CYCLING)
 3   ESTUARINE SEDIMENTS
 4   RIVER BED
 5   RIVER BANKS
 6   RIVER MOUTH
 7   SEAWATER
 8   TIDES
 9   MARINE SEDIMENTS
10   CURRENTS & WAVES
11   BEACHES & DUNES
12   PHYTOPLANKTON
13   AQUATIC PLANTS
14   MARGINAL VEGETATION
15   TERRESTRIAL VEGATN
16   MICRO-ORGANISMS
17   ZOOPLANKTON
18   BENTHIC FAUNA
19   NEKTONIC FAUNA
20   TERRSTRL VERTEBRATES
21   MUDFLATS
22   MARSH/REEDBED/SWAMPS
23   DETRITUS
24   FLOODPLAIN/ISLANDS
25   TOTAL ESTUARY ZONE
26   W/S PRECIPITATION
27   W/S RUNOFF
28   W/S GROUNDWATER
29   W/S GEOLOGY/TOPOGPHY
30   W/S BIOMASS
31   W/S SOLAR RADIATION
32   W/S WIND
33   RESERVES & SITES
34   DAMS
35   PLANT COMMUNITIES
36   MINES
37   FARMS
38   AGRICULTURAL RUNOFF
39   INDUSTRIES
40   INDUSTRIAL EFFLUENTS
41   HOUSING
42   HOUSEHOLD SEWAGE
43   BUSINESSES
44   RECREATIONALISTS
45   SHIPS & BOATS
46   ROAD/AIRPORT/BRIDGES
47   CANALS & MARINAS
```

```
THE MINIMUM LINK MATRIX
-----------------------

      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
   ****************************************************************************************************************************************************
 1 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  2  4  4  1  4  3  2  4  2  0  2  4  3  3  3  2  3  *   1 FRESHWATER (/FLOW)
 2 *  0  0  2  0  0  3  0  3  1  3  4  0  0  3  1  4  0  0  0  0  1  1  0  0  0  0  0  0  3  1  2  2  2  3  2  2  2  1  3  0  3  1  3  2  3  2  3  *   2 NUTRIENTS (/CYCLING)
 3 *  0  0  2  0  0  1  0  1  1  1  2  0  0  1  4  0  0  0  0  0  0  1  0  0  0  0  0  0  2  2  2  2  3  2  1  1  2  1  2  0  3  2  2  2  2  2  3  *   3 ESTUARINE SEDIMENTS
 4 *  0  0  1  0  0  2  0  1  2  2  3  0  0  2  5  0  0  0  0  0  0  1  0  0  0  0  0  0  3  4  3  3  3  2  3  2  2  4  2  0  2  4  2  2  2  1  1  *   4 RIVER BED
 5 *  1  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1  0  3  4  3  3  3  3  3  2  2  4  2  0  2  4  2  2  2  1  1  *   5 RIVER BANKS
 6 *  0  0  1  0  0  2  0  1  1  1  1  0  0  2  4  0  0  0  0  0  0  2  0  0  0  0  0  0  2  4  3  2  4  2  4  3  2  4  3  0  3  4  3  3  4  2  3  *   6 RIVER MOUTH
 7 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  3  2  5  3  4  4  3  4  4  0  4  5  4  4  5  3  4  *   7 SEAWATER
 8 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  2  1  5  3  5  4  3  5  4  0  4  5  4  4  5  3  4  *   8 TIDES
 9 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  2  5  3  2  5  3  4  4  3  4  4  0  4  5  4  4  5  3  4  *   9 MARINE SEDIMENTS
10 *  0  0  0  0  0  0  0  0  0  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  2  1  5  4  3  3  4  3  4  0  5  4  4  4  4  4  5  *  10 CURRENTS & WAVES
11 *  0  0  0  0  0  0  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  2  1  4  3  2  2  3  2  3  0  4  3  3  3  3  3  4  *  11 BEACHES & DUNES
12 *  0  0  0  0  0  0  0  1  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2  1  2  3  2  3  3  3  2  3  0  3  2  4  3  4  3  4  *  12 PHYTOPLANKTON
13 *  0  0  1  0  0  2  0  2  2  2  3  0  0  2  4  0  0  0  0  0  0  2  0  0  0  0  0  0  2  2  1  3  3  2  3  3  3  2  3  0  3  2  3  2  3  2  2  *  13 AQUATIC PLANTS
14 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2  1  3  3  2  3  3  3  2  3  0  3  2  3  3  3  2  2  *  14 MARGINAL VEGETATION
15 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2  1  3  3  2  3  3  3  2  3  0  3  2  4  3  4  3  3  *  15 TERRESTRIAL VEGATN
16 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2  2  3  3  2  2  2  2  2  3  0  3  1  3  3  3  2  3  *  16 MICRO-ORGANISMS
17 *  0  0  0  0  0  0  0  1  0  0  0  0  0  0  4  2  0  0  0  0  0  0  0  0  0  0  0  0  2  2  2  2  3  2  3  3  2  2  3  0  3  2  3  2  3  3  3  *  17 ZOOPLANKTON
18 *  0  0  1  0  0  2  0  2  1  2  3  0  0  2  4  2  0  0  0  0  0  2  0  0  0  0  0  0  2  2  2  2  3  2  2  2  2  2  3  0  3  2  3  2  3  2  2  *  18 BENTHIC FAUNA
19 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  2  0  0  0  0  0  0  0  0  0  0  0  0  2  2  2  3  3  2  3  3  2  2  3  0  3  2  3  2  3  3  3  *  19 NEKTONIC FAUNA
20 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  2  1  2  1  2  2  2  3  0  3  2  2  2  3  2  2  *  20 TERRSTRL VERTEBRATES
21 *  0  0  0  0  0  0  0  2  0  2  3  0  0  0  4  3  0  0  0  0  0  0  0  0  0  0  0  0  2  2  2  2  3  2  2  2  2  2  2  0  2  2  2  2  3  1  2  *  21 MUDFLATS
22 *  0  0  1  0  0  2  0  2  2  2  3  0  0  2  5  0  0  0  0  0  0  2  0  0  0  0  0  0  3  2  1  2  3  2  2  2  3  2  3  0  3  2  3  2  3  2  2  *  22 MARSH/REEDBED/SWAMPS
23 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  1  0  0  0  0  0  0  0  0  0  0  0  0  3  1  2  3  2  2  2  3  1  1  3  0  3  1  2  3  3  2  2  *  23 DETRITUS
24 *  1  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  3  3  4  3  2  2  3  2  3  0  3  3  3  3  3  2  2  *  24 FLOODPLAIN/ISLANDS
25 *  0  0  1  0  0  2  0  2  2  2  3  0  0  2  4  2  0  0  0  0  0  2  0  0  0  0  0  0  2  3  1  2  2  2  2  2  3  2  3  0  2  3  2  1  2  2  2  *  25 TOTAL ESTUARY ZONE
26 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  *  26 W/S PRECIPITATION
27 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  2  3  2  3  3  2  3  2  1  3  2  0  2  3  2  2  3  1  2  *  27 W/S RUNOFF
28 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  2  4  2  4  4  2  4  3  2  4  2  0  1  4  2  3  3  2  3  *  28 W/S GROUNDWATER
29 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  2  4  2  4  4  3  4  3  2  4  3  0  2  4  3  3  4  2  3  *  29 W/S GEOLOGY/TOPOGPHY
30 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  2  1  1  3  3  2  3  3  1  2  3  0  3  2  2  3  3  2  2  *  30 W/S BIOMASS
31 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  *  31 W/S SOLAR RADIATION
32 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  *  32 W/S WIND
33 *  0  0  0  0  0  0  0  2  0  2  3  0  0  0  2  2  0  0  0  0  0  0  0  0  0  0  0  0  2  2  2  2  2  2  1  2  2  3  2  0  2  3  1  1  2  1  2  *  33 RESERVES & SITES
34 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  2  3  4  2  4  3  1  3  1  0  1  3  2  3  2  2  2  *  34 DAMS
35 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  2  2  1  1  3  2  1  3  2  2  3  0  2  2  3  3  4  2  3  *  35 PLANT COMMUNITIES
36 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  3  2  3  2  3  2  2  4  1  0  2  4  1  2  1  1  2  *  36 MINES
37 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  1  3  3  1  3  2  2  2  2  0  2  2  1  2  2  1  1  *  37 FARMS
38 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  4  4  2  4  3  2  3  3  0  3  3  3  3  4  2  3  *  38 AGRICULTURAL RUNOFF
39 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  3  2  3  2  3  2  2  4  2  0  2  4  1  2  1  1  2  *  39 INDUSTRIES
40 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  3  2  4  2  4  3  2  4  3  0  3  4  3  3  3  2  2  *  40 INDUSTRIAL EFFLUENTS
41 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  1  4  3  3  3  1  3  2  2  4  1  0  2  4  1  2  2  1  2  *  41 HOUSING
42 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  0  0  1  5  3  2  5  2  4  4  3  4  3  0  3  5  4  4  4  3  4  *  42 HOUSEHOLD SEWAGE
43 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  2  2  2  1  2  2  1  3  1  0  1  3  1  1  1  1  1  *  43 BUSINESSES
44 *  0  0  0  0  0  0  0  1  0  1  2  0  0  0  4  2  0  0  0  0  0  0  0  0  0  0  0  0  2  2  1  1  1  1  1  2  2  3  2  0  1  3  1  2  1  1  1  *  44 RECREATIONALISTS
45 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  2  1  2  1  2  3  2  4  1  0  2  4  1  1  2  2  1  *  45 SHIPS & BOATS
46 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  1  3  2  2  2  2  2  1  1  3  1  0  1  3  1  1  2  2  1  *  46 ROAD/AIRPORT/BRIDGES
47 *  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  2  0  0  0  0  0  0  0  0  0  0  0  0  2  3  2  2  2  2  2  3  1  3  2  0  2  3  1  1  1  2  2  *  47 CANALS & MARINAS
   ****************************************************************************************************************************************************
      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
```

DATE - 061881 TRACK NUMBER 3

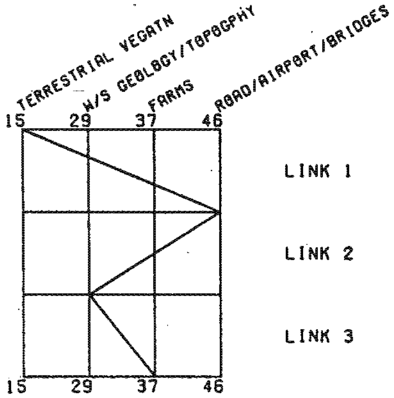
NEGATIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 23 - DETRITUS
ON COMPONENT 40 - INDUSTRIAL EFFLUENTS

DATE - 061881 TRACK NUMBER 4

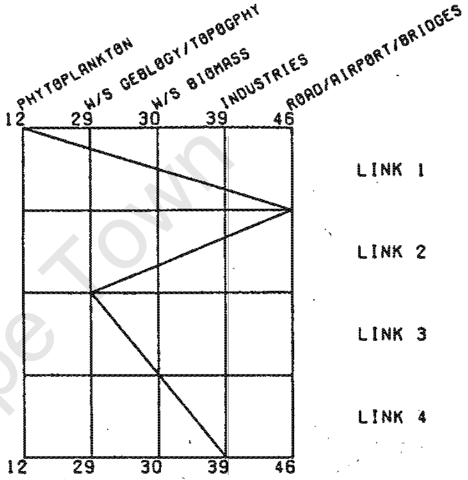
NEGATIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 15 - TERRESTRIAL VEGATN
ON COMPONENT 37 - FARMS

DATE - 061881 TRACK NUMBER 1

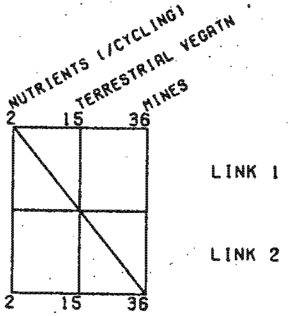
NEGATIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 12 - PHYTOPLANKTON
ON COMPONENT 39 - INDUSTRIES

DATE - 061881 TRACK NUMBER 2

NEGATIVE DETAILED ESTUARINE SYSTEM



DEPENDENCE OF COMPONENT 2 - NUTRIENTS (/CYCLING)
ON COMPONENT 36 - MINES

9.3. Nanaimo Port Component Interaction Matrix Analysis

* component interaction/minimum link matrix
* critical-component summary and ranking
* cut-component ranking
* disruptive measure rankings
* dependency track examples

```
NANAIMO PORT COMPONENT INTERACTION MATRIX
******************************************

THE COMPONENTS ARE AS FOLLOWS:-

 1  CURRENTS
 2  WIND
 3  WATER TEMPERATURE
 4  LIGHT
 5  INTERTIDAL VEGETATN
 6  UPLAND VEGETATION
 7  BACTERIA
 8  INSECTS
 9  LARVAE
10  SHELLFISH
11  CRABS
12  OTHER CRUSTACEANS
13  PELAGIC FISH
14  BOTTOM FISH
15  WATER BIRDS
16  BIRDS OF PREY
17  SONG BIRDS
18  MARSH & SHORE BIRDS
19  UPLAND & GAME BIRDS
20  AQUATIC MAMMALS
21  UPLAND MAMMALS
```

```
THE MINIMUM LINK MATRIX
-----------------------

        1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
    *****************************************************************
 1  *   4  1  4  3  4  2  3  3  4  4  4  4  4  4  4  4  3  3  3  4  3   *  1  CURRENTS
 2  *   3  3  3  2  3  1  2  2  3  3  3  3  3  3  3  3  2  2  2  3  2   *  2  WIND
 3  *   1  1  4  1  4  2  3  3  4  4  4  4  4  4  4  4  3  3  3  4  3   *  3  WATER TEMPERATURE
 4  *   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   *  4  LIGHT
 5  *   1  2  1  1  5  3  4  4  5  5  5  5  5  5  5  5  4  4  4  5  4   *  5  INTERTIDAL VEGETATN
 6  *   2  2  2  1  2  2  1  1  2  2  2  2  2  2  2  2  1  1  1  2  1   *  6  UPLAND VEGETATION
 7  *   2  2  1  2  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1   *  7  BACTERIA
 8  *   1  1  1  2  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2   *  8  INSECTS
 9  *   1  2  1  2  1  2  1  2  2  1  1  1  1  1  2  2  2  2  2  2  2   *  9  LARVAE
10  *   1  2  1  2  1  2  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2   * 10  SHELLFISH
11  *   2  2  1  2  1  3  2  2  1  1  2  1  1  1  3  3  3  3  3  3  3   * 11  CRABS
12  *   2  2  1  2  1  2  1  1  1  1  1  2  1  1  2  2  2  2  2  2  2   * 12  OTHER CRUSTACEANS
13  *   2  2  1  2  2  2  2  1  1  2  2  2  2  2  3  3  3  3  3  3  3   * 13  PELAGIC FISH
14  *   2  2  1  2  1  2  2  1  1  1  1  1  1  2  3  3  3  3  3  3  3   * 14  BOTTOM FISH
15  *   2  2  2  2  1  2  2  1  1  1  1  1  1  1  3  3  3  3  3  3  3   * 15  WATER BIRDS
16  *   2  2  2  2  2  1  2  1  2  2  1  1  1  1  1  3  1  1  1  3  1   * 16  BIRDS OF PREY
17  *   2  2  2  2  2  1  2  1  2  1  2  1  1  1  3  3  2  2  2  3  2   * 17  SONG BIRDS
18  *   2  2  2  2  2  1  2  1  2  1  1  1  1  1  3  3  2  2  2  3  2   * 18  MARSH & SHORE BIRDS
19  *   2  2  2  2  2  1  2  1  3  3  3  3  3  3  3  3  2  2  2  3  2   * 19  UPLAND & GAME BIRDS
20  *   2  3  2  3  2  3  2  2  2  1  1  1  1  1  3  3  3  3  3  3  3   * 20  AQUATIC MAMMALS
21  *   2  3  2  2  2  1  2  2  2  1  1  1  2  2  3  3  1  2  1  3  2   * 21  UPLAND MAMMALS
    *****************************************************************
        1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
```

(OUT-VALENCY RANKING)

```
THE NUMBER OF COMPONENTS EACH COMPONENT DEPENDS UPON
----------------------------------------------------

RANK     COMPONENTS    COMPONENT   COMPONENT
      DEPENDED UPON     NUMBER        LABEL
----  -------------   ----------  ---------
  1         17             7      BACTERIA
  2         11            16      BIRDS OF PREY
  3          9            12      OTHER CRUSTACEANS
  4          9             9      LARVAE
  5          8            15      WATER BIRDS
  6          8            14      BOTTOM FISH
  7          7             6      UPLAND VEGETATION
  8          7            11      CRABS
  9          7            18      MARSH & SHORE BIRDS
 10          6            17      SONG BIRDS
 11          6             8      INSECTS
 12          6            21      UPLAND MAMMALS
 13          5            20      AQUATIC MAMMALS
 14          4            10      SHELLFISH
 15          3             3      WATER TEMPERATURE
 16          3            13      PELAGIC FISH
 17          3             5      INTERTIDAL VEGETATN
 18          2            19      UPLAND & GAME BIRDS
 19          1             1      CURRENTS
 20          1             2      WIND
 21          0             4      LIGHT

 MAXIMUM = 21
```

(IN-VALENCY RANKING)

```
THE NUMBER OF DEPENDENCIES SUPPORTED BY EACH COMPONENT
------------------------------------------------------

RANK   DEPENDENCIES    COMPONENT   COMPONENT
        SUPPORTED       NUMBER       LABEL
----   ------------   ----------  ---------
  1         10             8      INSECTS
  2         10            10      SHELLFISH
  3         10            12      OTHER CRUSTACEANS
  4         10            13      PELAGIC FISH
  5          9             3      WATER TEMPERATURE
  6          9            11      CRABS
  7          9            14      BOTTOM FISH
  8          8             5      INTERTIDAL VEGETATN
  9          8             6      UPLAND VEGETATION
 10          6             9      LARVAE
 11          5             7      BACTERIA
 12          5             1      CURRENTS
 13          4            17      SONG BIRDS
 14          4            19      UPLAND & GAME BIRDS
 15          3             2      WIND
 16          3            18      MARSH & SHORE BIRDS
 17          3             4      LIGHT
 18          3            21      UPLAND MAMMALS
 19          2            15      WATER BIRDS
 20          1            20      AQUATIC MAMMALS
 21          1            16      BIRDS OF PREY

 MAXIMUM = 21
```

(CRITICAL IMPORTANCE RANKING)

```
COMPONENTS RANKED IN ORDER OF CRITICAL PATHS SUPPORTED
------------------------------------------------------

RANK   CRITICAL   COMPONENT   COMPONENT
         PATHS      NUMBER      LABEL
----   --------   ---------   ---------
  1       93           7      BACTERIA
  2       83           6      UPLAND VEGETATION
  3       52           2      WIND
  4       15           8      INSECTS
  5        4           9      LARVAE
  6        2           3      WATER TEMPERATURE
  7        2          10      SHELLFISH
  8        1           5      INTERTIDAL VEGETATN
  9        0           1      CURRENTS
 10        0           4      LIGHT
 11        0          11      CRABS
 12        0          12      OTHER CRUSTACEANS
 13        0          13      PELAGIC FISH
 14        0          14      BOTTOM FISH
 15        0          15      WATER BIRDS
 16        0          16      BIRDS OF PREY
 17        0          17      SONG BIRDS
 18        0          18      MARSH & SHORE BIRDS
 19        0          19      UPLAND & GAME BIRDS
 20        0          20      AQUATIC MAMMALS
 21        0          21      UPLAND MAMMALS

MAXIMUM =  400
```

(CUT-COMPONENT RANKING)

```
COMPONENTS RANKED IN ORDER OF CUT POSITIONS HELD
------------------------------------------------

RANK        CUT-   COMPONENT   COMPONENT
       POSITIONS     NUMBER      LABEL
----   ---------   ---------   ---------
  1       68            6      UPLAND VEGETATION
  2       53            7      BACTERIA
  3       52            2      WIND
  4        0            4      LIGHT
  5        0            5      INTERTIDAL VEGETATN
  6        0            1      CURRENTS
  7        0            3      WATER TEMPERATURE
  8        0            8      INSECTS
  9        0            9      LARVAE
 10        0           10      SHELLFISH
 11        0           11      CRABS
 12        0           12      OTHER CRUSTACEANS
 13        0           13      PELAGIC FISH
 14        0           14      BOTTOM FISH
 15        0           15      WATER BIRDS
 16        0           16      BIRDS OF PREY
 17        0           17      SONG BIRDS
 18        0           18      MARSH & SHORE BIRDS
 19        0           19      UPLAND & GAME BIRDS
 20        0           20      AQUATIC MAMMALS
 21        0           21      UPLAND MAMMALS

MAXIMUM =  400
```

(COMPONENT SUSCEPTIBILITY TO DISRUPTION RANKING)

COMPONENTS RANKED IN ORDER OF THE DISRUPTIVE POTENTIAL ON THEM FROM THE OTHER COMPONENTS UPON WHICH THEY DEPEND

RANK	DISRUPTIVE MEASURE	COMPONENT NUMBER	COMPONENT LABEL
1	11.6	7	BACTERIA
2	6.4	6	UPLAND VEGETATION
3	6.3	2	WIND
4	2.1	5	INTERTIDAL VEGETATN
5	1.5	8	INSECTS
6	1.5	3	WATER TEMPERATURE
7	1.4	1	CURRENTS
8	.9	9	LARVAE
9	.7	16	BIRDS OF PREY
10	.7	12	OTHER CRUSTACEANS
11	.6	20	AQUATIC MAMMALS
12	.6	11	CRABS
13	.6	15	WATER BIRDS
14	.6	14	BOTTOM FISH
15	.5	10	SHELLFISH
16	.5	19	UPLAND & GAME BIRDS
17	.5	21	UPLAND MAMMALS
18	.4	18	MARSH & SHORE BIRDS
19	.4	17	SONG BIRDS
20	.4	13	PELAGIC FISH
21	.0	4	LIGHT

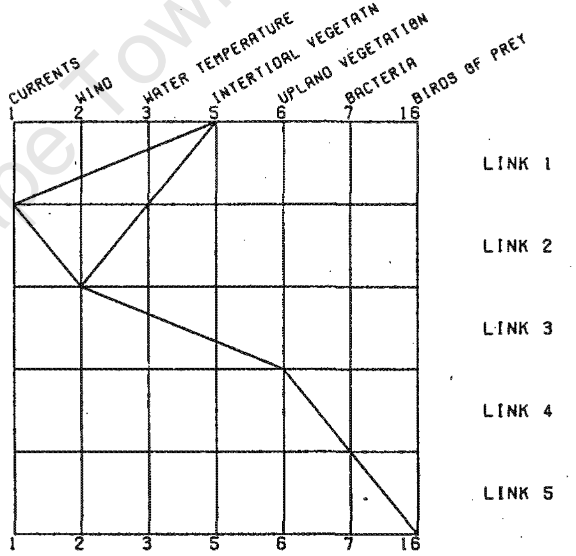
(COMPONENT DISRUPTIVE POTENTIAL RANKING)

COMPONENTS RANKED IN ORDER OF THEIR DISRUPTIVE POTENTIAL ON OTHER COMPONENTS DEPENDENT ON THEM

RANK	DISRUPTIVE MEASURE	COMPONENT NUMBER	COMPONENT LABEL
1	11.2	6	UPLAND VEGETATION
2	8.8	2	WIND
3	4.7	7	BACTERIA
4	1.5	8	INSECTS
5	.8	9	LARVAE
6	.7	10	SHELLFISH
7	.7	20	AQUATIC MAMMALS
8	.7	3	WATER TEMPERATURE
9	.7	16	BIRDS OF PREY
10	.6	15	WATER BIRDS
11	.6	12	OTHER CRUSTACEANS
12	.6	13	PELAGIC FISH
13	.6	11	CRABS
14	.6	14	BOTTOM FISH
15	.6	4	LIGHT
16	.5	1	CURRENTS
17	.5	5	INTERTIDAL VEGETATN
18	.5	17	SONG BIRDS
19	.5	19	UPLAND & GAME BIRDS
20	.5	21	UPLAND MAMMALS
21	.5	18	MARSH & SHORE BIRDS.

DATE - 061881 TRACK NUMBER 1

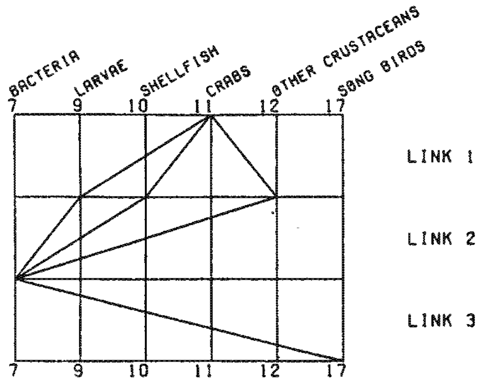
NANAIMO PORT COMPONENT INTERACTION MATRIX



DEPENDENCE OF COMPONENT 5 - INTERTIDAL VEGETATN ON COMPONENT 16 - BIRDS OF PREY

DATE - 061881 TRACK NUMBER 2

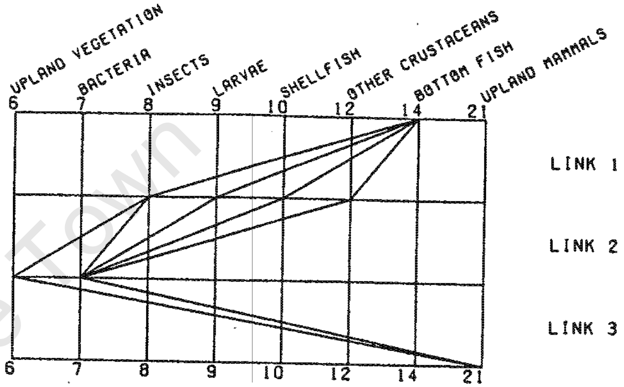
NANAIMO PORT COMPONENT INTERACTION MATRIX



DEPENDENCE OF COMPONENT 11 - CRABS
ON COMPONENT 17 - SONG BIRDS

DATE - 061881 TRACK NUMBER 3

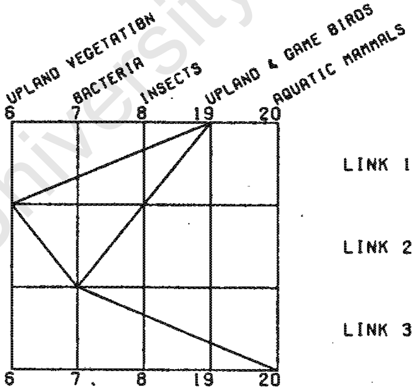
NANAIMO PORT COMPONENT INTERACTION MATRIX



DEPENDENCE OF COMPONENT 14 - BOTTOM FISH
ON COMPONENT 21 - UPLAND MAMMALS

DATE - 061881 TRACK NUMBER 4

NANAIMO PORT COMPONENT INTERACTION MATRIX



DEPENDENCE OF COMPONENT 19 - UPLAND & GAME BIRDS
ON COMPONENT 20 - AQUATIC MAMMALS

10. APPENDIX C - GRAPH THEORY DEFINITIONS AND FORMULAE

10.1. Symbols and Formulae

A - the adjacency matrix of any non-planar directed graph

Ax - the adjacency matrix of 'A' with vertex 'x' removed (i.e row and column x are set to zeros)

x - any single vertex in 'A'

m - the number of vertices in 'A'

L[N] - the minimum link matrix of 'A'

L[Nx] - the minimum link matrix of 'Ax'

the in-valency of any vertex j in A = $\sum_{i=1}^{m} A(i,j)$

the out-valency of any vertex i in A = $\sum_{j=1}^{m} A(i,j)$

the average in-graph distance of any vertex j in A = $\sum_{i=1}^{m} L[N](i,j) / n$

where n = the sum of the non-zero elements in column j of L[N]

G(i) - the average out-graph distance of any vertex i in A = $\sum_{j=1}^{m} L[N](i,j) / n$

where n = sum of non-zero elements in row i of L[N]

Gx(i) - the average out-graph distance of any vertex i in Ax = $\sum_{j=1}^{m} L[Nx](i,j) / n$

where n = sum of non-zero elements in row i of L[Nx]

D(x) - the (in-) disruptive measure of vertex x in graph A where :

$$D(x) = \sum_{i=1}^{x-1} \left| G(i) - Gx(i) \right| + \sum_{i=x+1}^{m} \left| G(i) - Gx(i) \right|$$

The alternative forms of the last three measures (i.e. in- or out-forms) can be computed by summing down columns or across rows as is appropriate.

10.2. The Maximum Values of Some Graph Measures

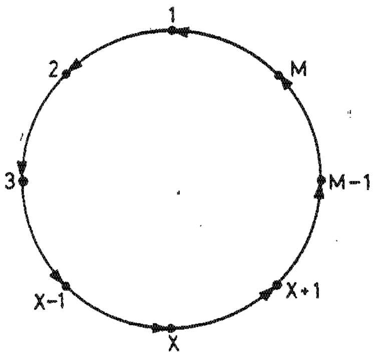
Maximum Valencies

Many applications of graph theory (e.g. for the study of communication networks) do not consider the possibility of a vertex linked to itself. However, for environmental systems, an edge A(i,j), where i=j, does have a useful meaning. The maximum valency for any vertex is therefore equal to the total number of vertices in the graph (i.e. the value of m).

Maximum Average Graph Distance Measure

The maximum value of G(i) occurs when each vertex is connected to one and only one other vertex because :

- * A graph, A, with each vertex connected to one and only one vertex has the form :



and any row i in L[N] will contain a sequence of values from 1 to m. Row 1 for the above graph would be :

j =	1	2		x-1	x	x+1		m-1	m
$i=1;\ \overset{m}{\underset{j=1}{L[N](i,j)}} =$	1	2	...	x-1	x	x+1	...	m-1	m

Therefore $\sum_{i=1}^{m} L[N](i,j) = m(m+1)/2$ (which is the sum of the arithmetic progression)

and $G(i) = (m+1)/2$ as $n = m$ (there are no zero values in row i)

* It is not possible to have an element in L[N]>m as this would not be a minimum linkage (i.e. a path with more than m linkages must pass through at least one vertex more than once). For any longest minimum path in row i there must be a sequence of shorter paths. That is, if i and j are connected by a minimum path of m links, there must be some vertex k, such that i -> k by m-1 links, and k -> j by one link; and so on.

* Adding an edge to A (to give A') either produces more connections, or leaves the linkage the same. If more connections are created in the graph, one or more elements of row i in L'[N] must decrease in value (i.e. some minimum paths are shortened by the addition of a non-duplicating edge). Therefore G'(i) is equal to or less than G(i).

* Removing an edge from A (to give A") reduces one or more elements in row i of L"[N] to zero. However, the sequential increase in the row elements must still occur, and for a row with one zero value there will be values from 1 to m-1. Therefore :

$$\sum_{j=1}^{m} L''[N](i,j) = m(m-1)/2$$

and $G''(i) = m/2$ as $n = m-1$ in this case (there is one zero value in row i)

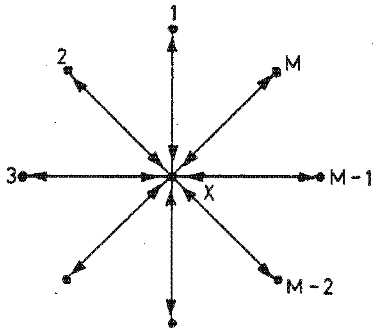
Therefore G"(i) is always less than G(i).

The maximum value of G(i) is (m+1)/2.

$$G(i)\max = (m+1)/2$$

Maximum Number of Minimum Paths to which a Vertex can be Critical

A critical vertex is one whose removal from a system disconnects one or more minimum paths.



From the figure above it may be seen that a configuration is always possible such that the removal of a vertex (X) disrupts a minimum linkage between every other vertex. As each vertex is linked to m-1 other vertices, the maximum number of minimum paths to which a vertex may be critical is (m-1)(m-1).

Maximum Number of Paths to which a Vertex may be a Cut-vertex

A cut-vertex is one which is critical to minimum and non-minimum paths. As X is also a cut-vertex to all paths in the case described above, the maximum number of paths to which a vertex may be a cut-vertex is also (m-1)(m-1).

Maximum Value of the Disruption Measure - D(x)

Given :

$$D(x) = \sum_{i \neq x}^{m} \left| G(i) - Gx(i) \right|$$

D(x)max occurs for G(i) a maximum, and Gx(i) a minimum, or viceversa. In general the minimum value of G(i)>0 for any graph with a non-zero number of edges. Therefore :

$$\begin{aligned} D(x)\max &< \sum_{i \neq x}^{m} \left| G(i)\max - 0 \right| \\ &< (m-1)\left| (m+1)/2 - 0 \right| \\ &< (m+1)(m-1)/2 \end{aligned}$$

In fact it can be shown, considering particular cases, that $D(x)\max = (m-3m+4)(m-1)/2m$ for $m>7$; $m(m+1)/4$ for $4<m<8$; and $(2m-1)(m-1)/m$ for $1<m<5$. However, for this thesis, and the CI programme listed in Appendix A, D(x)max is taken as the proven upper bound, which is :

$$D(x)\max = (m+1)(m-1)/2$$