SEL REPORT NO. 128

METAPHOR (VERSION 1)
PROGRAMMER'S GUIDE

BY
D.G. FURCHTGOTT

UNDER THE SUPERVISION OF
J.F. MEYER

PREPARED FOR
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA 23365
G.E. MIGNEAULT, TECHNICAL OFFICER
NASA GRANT NSG 1306

JANUARY 1979

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
SYSTEMS ENGINEERING LABORATORY
THE UNIVERSITY OF MICHIGAN, ANN ARBOR

014524-6-T

METAPHOR (Version 1): Programmer's Guide

By
D. G. Furchtgott

Under the Supervision of
J. F. Meyer

Prepared For
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365
G. E. Migneault, NASA Technical Officer
NASA Grant NSG 1306

January 1979

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

This report describes the internal structure of the first version of METAPHOR*, an interactive software package to facilitate performability modeling and evaluation. A companion "User's Guide" for METAPHOR (Version 1) is currently in the process of being documented. As the capability of METAPHOR is extended via incorporation of additional evaluation programs, revised or supplemented guides will be prepared in order to maintain an up-to-date documentation of the system. It is assumed that the reader is familiar with the context of METAPHOR, that is, the performability modeling and evaluation methods developed under the subject grant and described in a number of previous reports and publications [1]-[9].

As we currently envision METAPHOR, it is the prototype of a software package that, ultimately, will contain programmed tools to facilitate each step of performability model construction and model solution. In certain steps, such facilitation will take the form of complete automation; in other cases, particularly steps involving model construction, an interactive mode will be necessary wherein the programmed tool acts strictly as an aid. More specifically, the major steps to be facilitated are:

1) Construction of the base model,

2) Elaboration of the base model into a model hierarchy,

---

*Michigan Evaluation Aid for Perphormability

3) Formulation of the capability function in terms of the interlevel translations between adjacent models of the hierarchy,

4) For each accomplishment level a, computation of the base model trajectory set Ua that corresponds to a,

5) For each trajectory set Ua, computation of its probability (the performability value for accomplishment level a).

In addition to facilitating specific steps of the modeling and evaluation process, METAPHOR is intended to serve as a performability evaluation tutor for a person who is learning to use its programs.

In developing Version 1 of METAPHOR, emphasis was placed on obtaining a general structure that can accommodate the various types of evaluation programs that are planned for the system. In addition, Version 1 contains specific programs which facilitate steps 1) and 5) outlined above. Finally, the tutorial aspect of METAPHOR is fairly well developed in Version 1 with an extensive repertoire of HELP requests, along with a preprogrammed series of questions relating to specific topics.

METAPHOR is written in APL [10]-[11] , chosen because of its notational compactness and array handling abilities. However, the eventual translation of the prototype package into a faster and more portable language such as FORTRAN may be desired, and this report should also provide valuable documentation for such a process. A working knowledge of APL is necessary for a complete understanding at the lowest level. However, the descriptions contained herein are hopefully general enough so that, even lacking familiarity with APL, the essential mechanisms of the system should be apparent.

The appendix provides a complete listing of the METAPHOR package to date, with the functions organized into topical groupings. Various conventions utilized in the package are discussed in Section 2, and some techniques used for control and intrapackage information exchange are studied in Section 3. In Section 4, the various algorithms employed by METAPHOR for performability evaluation are treated. Descriptions of the global variables and some of the important local variables are then presented in Section 5, while Section 6 gives the interdependence of the METAPHOR functions. Finally, Section 7 presents an overview of each METAPHOR function, giving its calling sequence, purpose, global variables used, functions calling it, functions it calls, general comments, and listing.

In those sections containing entries in alphabetical order, standard APL lexicographical ordering is employed: A, B, C, ..., X, Y, Z, 0, 1, 2, ..., 7, 8, 9, _A_, _B_, _C_, ..., _X_, _Y_, _Z_, _0_, _1_, _2_, ..., _7_, _8_, _9_.

2. Conventions

Several programming conventions dealing with various aspects of METAPHOR have been adopted. These are stated here to ease the process of absorbing the constructional details and to allow a more uniform product when modifications are made. In particular, conventions regarding the names of functions, variables, and labels have been established, and are examined in Sections 2.1-3. Finally, some other general conventions are discussed in Section 2.4.

2.1 Function Names

The name of each function is suggestive of its purpose, with the following guidelines.

1) Each function controlling the execution of a command begins with "COMMAND" and ends with the name of the command it administers, e.g., COMMANDALTER.

2) Each function that somehow obtains a value (either by asking the user or by generating it) begins with "GET," and ends with the name (or long abbreviation) of the item it fetches, e.g., GETACCLEVPROB, GETALTERVECTOR.

3) Each function that prints information in response to a HELP command ends with "INFO" and begins with the first letters of major syllables of the function for which it is the information routine. These letters are also underlined. For example, the help response to the GETALTERVECTOR function is contained in GAVINFO. (See Section 3.1.)

4) Each function that inputs information begins with "IN," viz., INPUT and INYES.

5) Each function that outputs information begins with "PRINT," viz., PRINT, PRINTPERFORMABILITY, and PRINTQUAD.

6) Each function that checks the validity of input information begins with "CHECK," e.g., CHECKPROB.

7) Each function that is utilized to input commands with arguments (namely BRIEF and ECHO, see Section 3.1) must be the name of the command.

The functions in the current METAPHOR package are listed in the appendix and in Section 7; the latter section also discusses each function, giving its purpose and commenting on its structure and behavior.

2.2 Variable Names

As with function names, variable names are suggestive of their purpose. The patterns below have been maintained.

1) Variables used to enter commands are the same names as the commands themselves, e.g., ALTER='ALTER '. (See Section 3.1.)

2) Variables used to identify other requests, e.g., ON and GIVEN, are the same names as the requests themselves.

3) Variables used as calling function indicators for the HELP command, e.g., GAV and GBV, are the same abbreviations used to prefix the name of the information routines associated with the variables, i.e., GAVINFO and GBVINFO for the examples above. (See Section 3.2.1.)

4) Variables used as flags to indicate whether certain other variables have been defined (e.g., DEFBASICVARIABLES and DEFF) begin with "DEF" and end with the variable name for which the variable is a flag. (See Section 3.2.2.)

5) Commonly used abbreviations in labels are as follows:

NUM	NUMBER
ACCLEV	ACCOMPLISHMENT LEVEL
PROB	PROBABILITY
TRI	TRINARY
TRAJ	TRAJECTORY
BIN	BINARY
INFO	INFORMATION
EVAL	EVALUATE
COM	COMMENT
CALC	CALCULATE.

2.3 Label Names

Standard APL defines all labels to be local, that is, a label name used in function A will not be recognized (known) by some other function B. However, the version of APL implemented at The University of Michigan Computing Center defines labels globally, that is, each function is aware of every label in the package. If two different statements were given the same label, incorrect (and unpredictable) results may occur. Thus, in this version of METAPHOR, every label has been uniquely named. In addition, several other conventions regarding labels have been observed in the construction of METAPHOR and are stated below.

1) Every label begins with the first letters of major syllables or key phrase segments of the function for which it is a label. These letters are also underlined. Below is a list of all such abbreviations.

A	COMMANDALTER
CC	COMMANDCOM
CCALC	COMMANDCALC
CHECK	CHECKPOSI
CTP	CALCTRAJPROB
D	COMMANDDATA
E	ENCODE
GAP	GETACCLEVPROB
GAV	GETALTERVECTOR
GBV	GETBASICVARIABLES
GDV	GETDATAVECTOR
GENHM	GENERATEHMATRIX
GENPM	GENERATEPMATRIX
GFV	GETFVECTOR
GG	GGIVEN
GGM	GETGMATRICES
GIV	GETIVECTOR
GN	GNFAIL
GNA	GETNUMACCLEV
GNBV	GETNUMBASICVARIABLES
GNP	GETNUMPHASES
GNTS	GETNUMTRAJSETS
GS	GETSTATES
GVV	GETVVALUES
H	COMMANDHELP

IN	INPUT
IY	INYES
MET	METAPHOR
PQ	PRINTQUAD.

2) Every label to a section involving some form of input has the postfix "IN," e.g., GNPIN.

3) Every label denoting the start of a loop has the postfix "LOOP", e.g., ELOOP.

4) Every label referencing a section dealing with the response to a particular input has the name of that input appended to the label. For example, the statement indicated by the label GENPMGIVEN is branched to (in the function GENERATEPMATRIX) after the reply "GIVEN" is input in response to the question "What type of P matrix?"

5) All other labels reflect the purpose of the section referred to by the label.

2.4 Other Conventions

Other conformities and guidelines that have been followed are listed below.

1) Names of commands do not exceed six characters in length. Also the set of letters comprising one command cannot be a subset (proper or otherwise) of the set of letters comprising another command. Thus, since EVAL is a command, no command can be named VALUE because E, V, A, and L all appear in VALUE; similarly, because of ALTER, LATER would be illegal. (For further explanation, see Section 3.1.)

2) Names of matrix generator types, e.g., IDENTITY and NFAIL, cannot exceed eight characters in length. As with commands, the set of letters comprising one such type can not be a subset of the set of letters comprising another type.

3) The parameter "Z" is used as the result argument in the definition of all functions which return a value, e.g., the definition of CHECKBIN is "Z ← CHECKBIN CHECKNO."

4) The variable IN is used to handle most input data. See Section 3.1 for details.

5) Most program statements are written to be as clear as possible. Further, ample internal documentation (comments, etc.) has been incorporated throughout the package.

3. Control Techniques

Within the METAPHOR package, various methods of control and information exchange among the various functions are employed. The more important of these techniques are explained in the following sections. Section 3.1 examines the mechanics associated with the INPUT function, while Section 3.2 discusses the HELP and ALTER commands. Most other sections of METAPHOR employ straightforward control and information exchange techniques.

3.1 INPUT

Because of METAPHORS's versatile input capabilities--namely being able to accept as input numerical data or commands with zero, one, or two modifiers--the input handling routine INPUT is necessarily somewhat complex. The techniques exercised in INPUT are outlined in this section. Figure 1 gives a flowchart of the function. Standard flowchart symbols are employed; in particular, a rectangle denotes a process, a diamond denotes a decision, and a rhomboid indicates some form of input or output.

When the quad command (the APL "square") is placed to the right of a specification symbol (left arrow, ←-), input data from the user is required; APL prints a prompt symbol and interrupts program

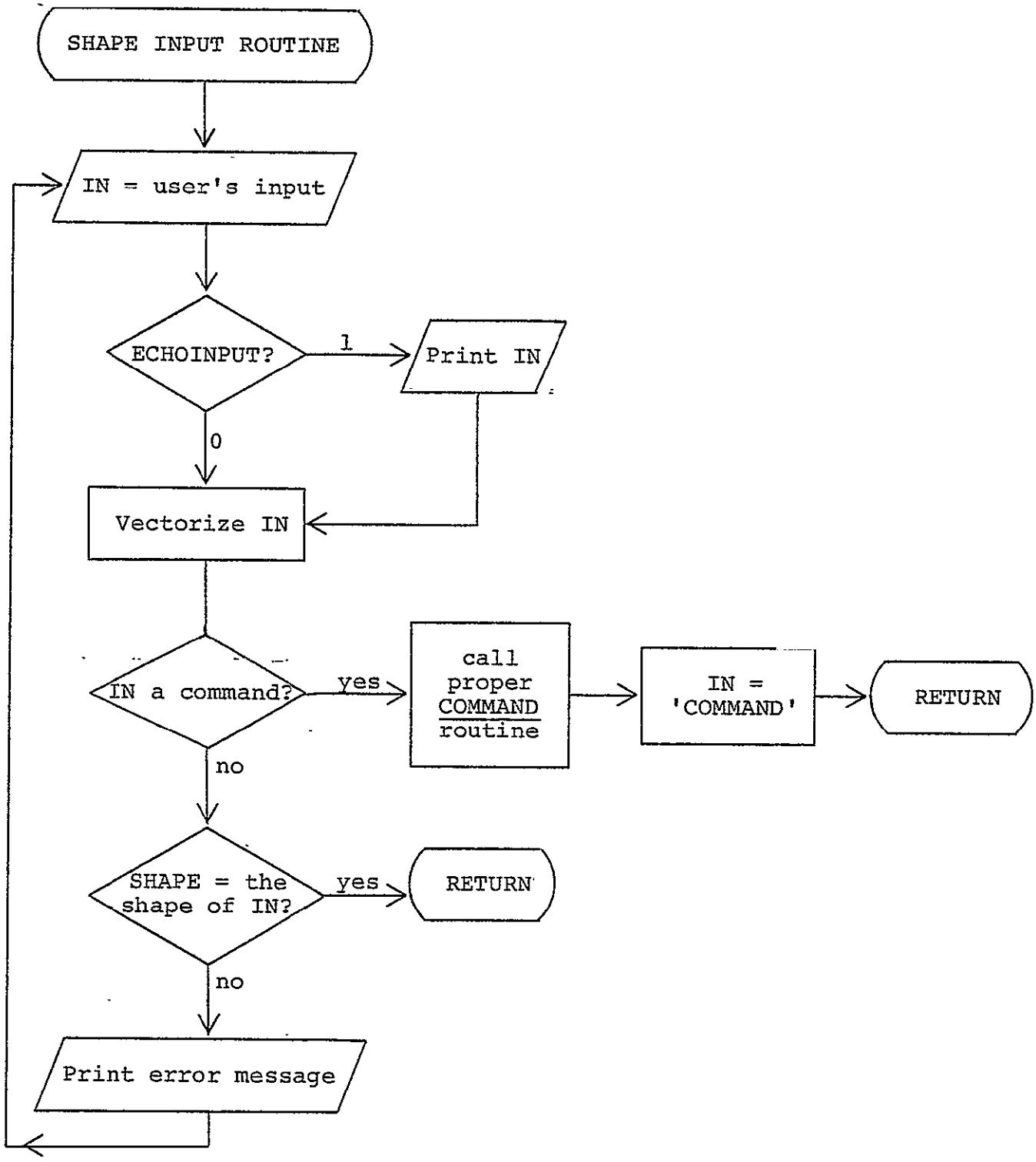


Figure 1. Flowchart for the function INPUT

execution until an expression is entered to replace the quad. This expression can evaluate to either a numeric quantity or a character quantity. In METAPHOR, commands and other character information are input by entering the name of a variable containing a string corresponding to the desired input. Thus, for example, when a user types HELP, he is not providing the program with the string 'HELP', but rather a variable named HELP which contains a six character string 'HELP '. The variable HELP could of course contain any value. The decision to use the actual string as the internal value (instead of some arbitrary internal code, as say 17) was made so that if echoing were set on, the command input (rather than some internal code) would be echoed.

The function INPUT must determine if the item input is a command or some form of numeric data. This is accomplished by converting the input to a vector and then comparing the input to a list of all valid commands in the array COMMANDLIST. Presently this array contains

```
HELP~~
EXIT~~
DATA~~
ALTER~
CALC~~
ECHO~~
BRIEF~
COM~~~
EVAL~~
```

Here'~' denotes a blank.

If the input is a command, the result of these computations will yield a 9x6 binary array with exactly one row consisting of all

1's. By ANDing each row, a 9 element vector will result with a 1 in the column corresponding to the row of COMMANDLIST containing the command. As an illustration, suppose EVAL is input. The comparison of EVAL with COMMANDLIST produces

```
011011
100011
010111
100101
011011
100011
000101
000111
111111
```

--and the ANDing of each row produces 000000001. Hence a command in indicated and it is the ninth in the command list, viz., EVAL. The vector above is called COMMANDVECTOR in the function INPUT. Note that if 'VALUE' were entered, the same result would follow. For this reason the restriction noted in Section 2.4 has been established: no command can consist of a set of letters which is a subset of the set of letters comprising another command. For instance, one could not define a command called CHOOSE because the letters in the command ECHO all appear in CHOOSE. Presently four commands consist of two words. These are BRIEF ON, BRIEF OFF, ECHO ON, and ECHO OFF. These commands may be conceptually considered one word commands (e.g., BRIEF) modified with a parameter (e.g., ON).

The problem of entering commands with parameters, as for example "BRIEF ON", is solved by defining a function which returns the proper values for the command. Thus, if the user inputs "BRIEF ON", APL executes the function BRIEF taking ON is an argument.

BRIEF then returns a twelve character vector to the quad (input) symbol; the first six characters of the vector are 'BRIEF ' and so INPUT can thus determine the nature of the command, while the second six characters is a representation of the argument 'ON ' and so COMMANDBRIEF can determine the proper response to the command. Up to two such parameters could be defined for a command, more if the parameters were of similar nature. For instance, a command TEST could be defined with format

A TEST B

where A is a scalar and B is a one dimensional vector. Then a typical user input might be

ALL TEST 1,5 7 9

where ALL would be the name of some variable and 1, 5, 7, 9 would be the vector to be used by TEST. At present, only the single parameter commands BRIEF and ECHO are implemented in METAPHOR.

If the input is determined to be a command, then using the variable COMMANDVECTOR, the program flow branches to a section of the function INPUT where the proper COMMAND routine is called. Finally, the variable IN is replaced with the code value 'COMMAND', and INPUT returns. Thus, when the function that requested input checks the variable IN, it will find that a command was executed, but that no data was entered. The function will then repeat its original query and again ask for input.

If the input is determined to be specifically a HELP command, the function INPUT calls COMMANDHELP with the parameter that is given as the second argument of the INPUT call, i.e., the value in

the parameter ROUTINE. This parameter contains a code identifying the function requesting INPUT.

If, however, the user's entry to INPUT is determined not to be a command, it is assumed to be data. To insure that the data is the proper shape (dimension), it is compared to the expected size given as the first parameter to INPUT, namely the parameter SHAPE. Since only scalar or vector (i.e., no multidimensional array) entries are expected, the input is vectorized using the ravel (,) function. If the dimension is not as expected, an error message is printed and the user is again asked for input. As an example, suppose the user tells METAPHOR (in the function GETNUMPHASES) that his model contains 4 phases. METAPHOR then asks for the number of states in each phase (via the function GETSTATES), expecting 4 values-- one for each phase. If the user gives 5 values, an error message would be printed and the request for the states per phase would be repeated.

All input acquired through the function INPUT is placed in the global variable IN. IN thus makes available to all functions the most recent item given METAPHOR.

3.2 User Assistance Commands

Several of the user assistance commands, namely HELP, ALTER, and DATA, have somewhat involved control mechanisms. These techniques are treated within this section.

3.2.1 HELP

As mentioned in Section 3.1, the HELP command causes immediate execution of the function COMMANDHELP. Associated with COMMANDHELP is a parameter (called ROUTINE) informing COMMANDHELP of the function for which the user requires help. The sequence occurs as follows. Suppose some function, call it F, requires input from the user. Function F then calls the function INPUT and passes to INPUT an alphanumeric code indicating F is the function calling INPUT. Each function requiring input has been assigned a unique code and so INPUT is unambiguously aware of the calling function. Now suppose the user types "HELP". INPUT recognizes this and immediately calls the function COMMANDHELP, passing to COMMANDHELP the code for F. This code is given to COMMANDHELP so that the appropriate INFO routine can be chosen for the question METAPHOR is asking the user. The codes are as given below:

FUNCTION	CODE NAME	CODE VALUE
GETNUMPHASES	GNP	1
GETSTATES	GS	2
GENERATEPMATRIX	GPM	3
GGIVEN	GG	4
GNFAIL	GN	5
GDEDFAIL	GD	6
GENERATEHMATRIX	GHM	7
GETNUMBASICVARIABLES	GNBV	8
GETBASICVARIABLES	GBV	9
GETNUMACCLEV	GNA	10
GETIVECTOR	GIV	11
GETGMATRICES	GGM	12
GETFVECTOR	GFV	13
GETVVECTOR	GVV	14
GETNUMTRAJSETS	GNTS	15
GETALTERVECTOR	GAV	16
GETDATAVECTOR	GDV	17
METAPHOR	MET	18

For example, suppose METAPHOR is executing the function GENERATEHMATRIX and so asks the user "WHAT TYPE OF H MATRIX?" The function GENERATEHMATRIX then requests the function INPUT with the call "8 INPUT GHM", indicating to INPUT that it is expecting an entry with 8 (character) values and that the function requesting the input is GENERATEHMATRIX. If the user has forgotten the legal types of H matrices, he can respond with "HELP," whereupon INPUT, by the method discussed in Section 3.1, will recognize the HELP request and will call the function COMMANDHELP 7. By using the parameter 7, COMMANDHELP, in turn, will call the function GHMINFO to print the desired aid. INPUT then places the string 'COMMAND' into IN; next GENERATEHMATRIX examines IN, determines a command was entered, again asks "WHAT TYPE OF H MATRIX?", and once more requests data through the INPUT function.

Figure 2 shows a flowchart section of a typical data input

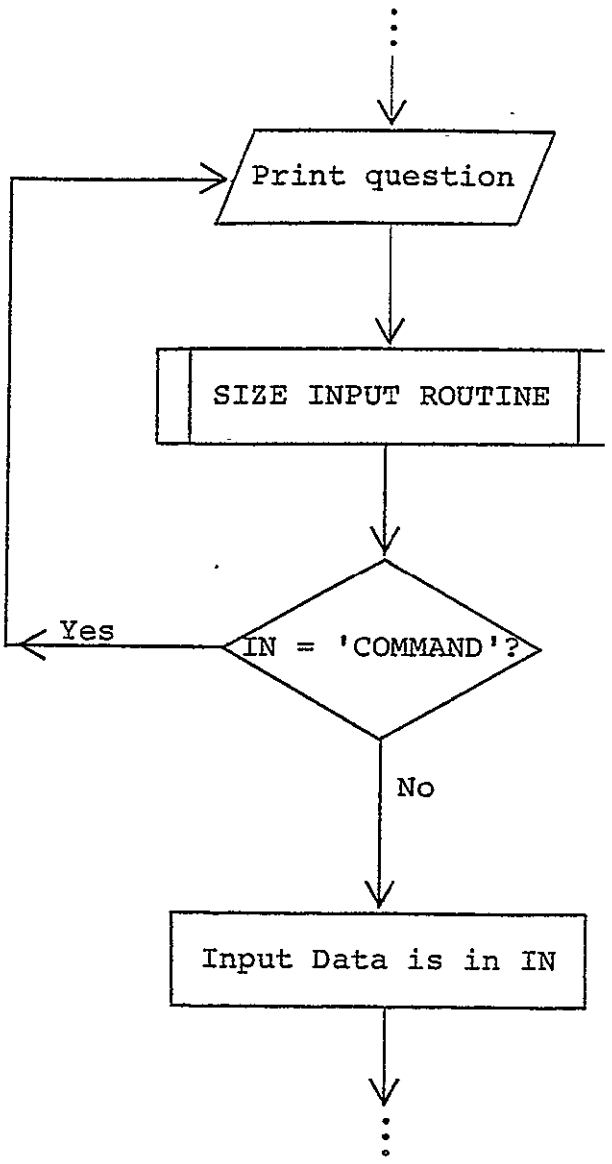


Figure 2. Typical flowchart section involving a call to the function INPUT

sequence.

3.2.2 ALTER and DATA

The structure of the ALTER and DATA commands are similar and will be discussed concurrently. When the user enters "ALTER", the function INPUT intercepts the command and calls the function COMMANDALTER to handle the directive. In turn, COMMANDALTER first calls the function GETALTERVECTOR to obtain from the user the variables to be changed. This information is used to form a vector (named ALTERVECTOR) which encodes those variables the user desires to change. Each index of ALTERVECTOR corresponds to an alterable variable and contains a 1 if the variables is to be changed and a 0 if not. The index code is as follows:

Index 1--Alter the P matrices.

Index 2--Alter the H matrices.

Index 3--Alter the time-invariant* basic variables.

Index 4--Alter the values of all accomplishment levels.

Index 5--Alter the values of just the present accomplishment level.

Index 6--Alter the initial vector.

Index 7--Alter the characteristic matrices.

Index 8--Alter the characteristic vector.

Index 9--Alter the characterization of the time-invariant basic variables.

Index 10--Alter the number of trajectory sets.

---

*By "time-invariant" variables in this context we mean an element (sample) of a time invariant process.

When the user enters "DATA", a similar sequence occurs. INPUT calls the function COMMANDDATA, which in turn calls the function GETDATAVECTOR. GETDATAVECTOR returns a variable called DATAVECTOR having the code below:

Index 1--Display the number of phases.

Index 2--Display the number of states in each phase.

Index 3--Display the P matrices.

Index 4--Display the H matrices.

Index 5--Display the number of time-invariant basic variables.

Index 6--Display the probabilities of each of the time-invariant basic variables.

Index 7--Display the number of accomplishment levels.

Index 8--Display the number of trajectory sets.

Index 9--Display the initial vector.

Index 10--Display the characteristic matrices.

Index 11--Display the characteristic vector.

Index 12--Display the characterization of the time-invariant basic variables.

Index 13--Display the performability.

Both COMMANDALTER and COMMANDDATA then loop through their control vectors (namely ALTERVECTOR or DATAVECTOR), altering or displaying the proper variables until all requests have been serviced. To insure that METAPHOR does not attempt to display or alter a value which has not been defined, a definition flag is associated with each alterable or displayable variable. The name of each such flag begins with "DEF". (See Section 2.2.) These are set

to 1 if the variable is defined and to 0 if the variable in question is undefined. Should the user request either the display of an undefined variable or the change of an undefined variable, an error message is printed and the next request is processed. The definition flags, along with the variables with which they are associated, are given in the following list:

FLAG	VARIABLE
DEFACCLEVEL	[Variables associated with accomplishment level definition]
DEFBASICVARIABLES	BASICVARIABLES
DEFF	F
DEFG	G
DEFI	I
DEFNUMACCLEV	NUMACCLEV
DEFNUMBASICVARIABLES	NUMBASICVARIABLES
DEFNUMPHASES	NUMPHASES
DEFNUMSTATES	NUMSTATES
DEFTRAJSETS	NUMTRAJSETS
DEFP	P
DEFPERFORMABILITY	PERFORMABILITY
DEFV	V

Alteration of variables is achieved by calling the proper functions associated with obtaining the variables. For instance, if the user desires to change the I vector, then the function GETIVECTOR is called. Presently, alteration of the F vector is not possible (because after the F vector is input, it is immediately involved in calculations, see Section 4.2), however it is included in the ALTER routine for completeness. Display of variables is achieved by simply printing the variables. Before any alteration or display, a heading is printed stating both that an alteration or display is to be executed and the variable that is involved. Figures 3-4 give the flowcharts of COMMANDALTER and COMMANDDATA.

The functions GETALTERVECTOR and GETDATAVECTOR are fairly

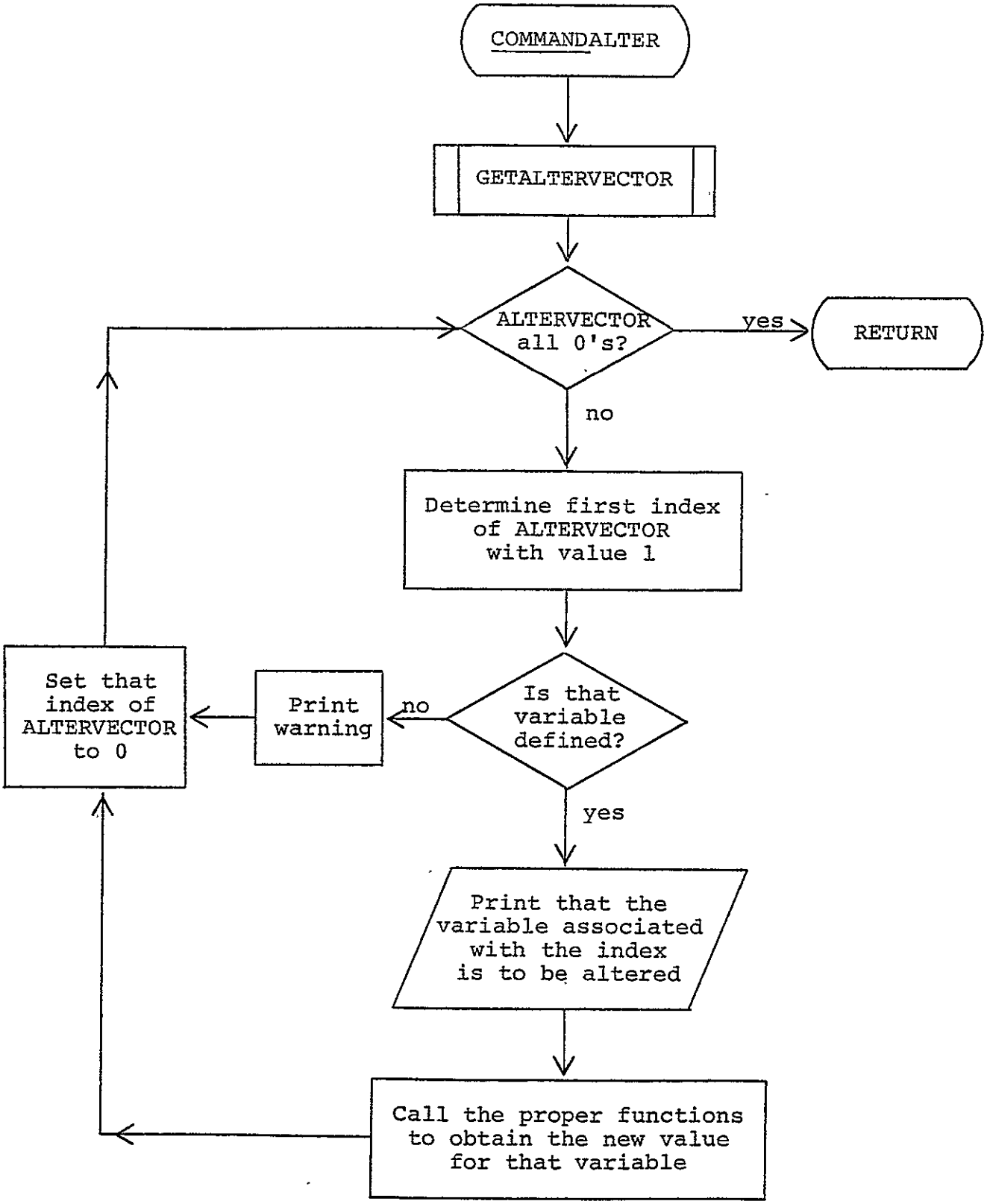


Figure 3. Flowchart for the function COMMANDALTER

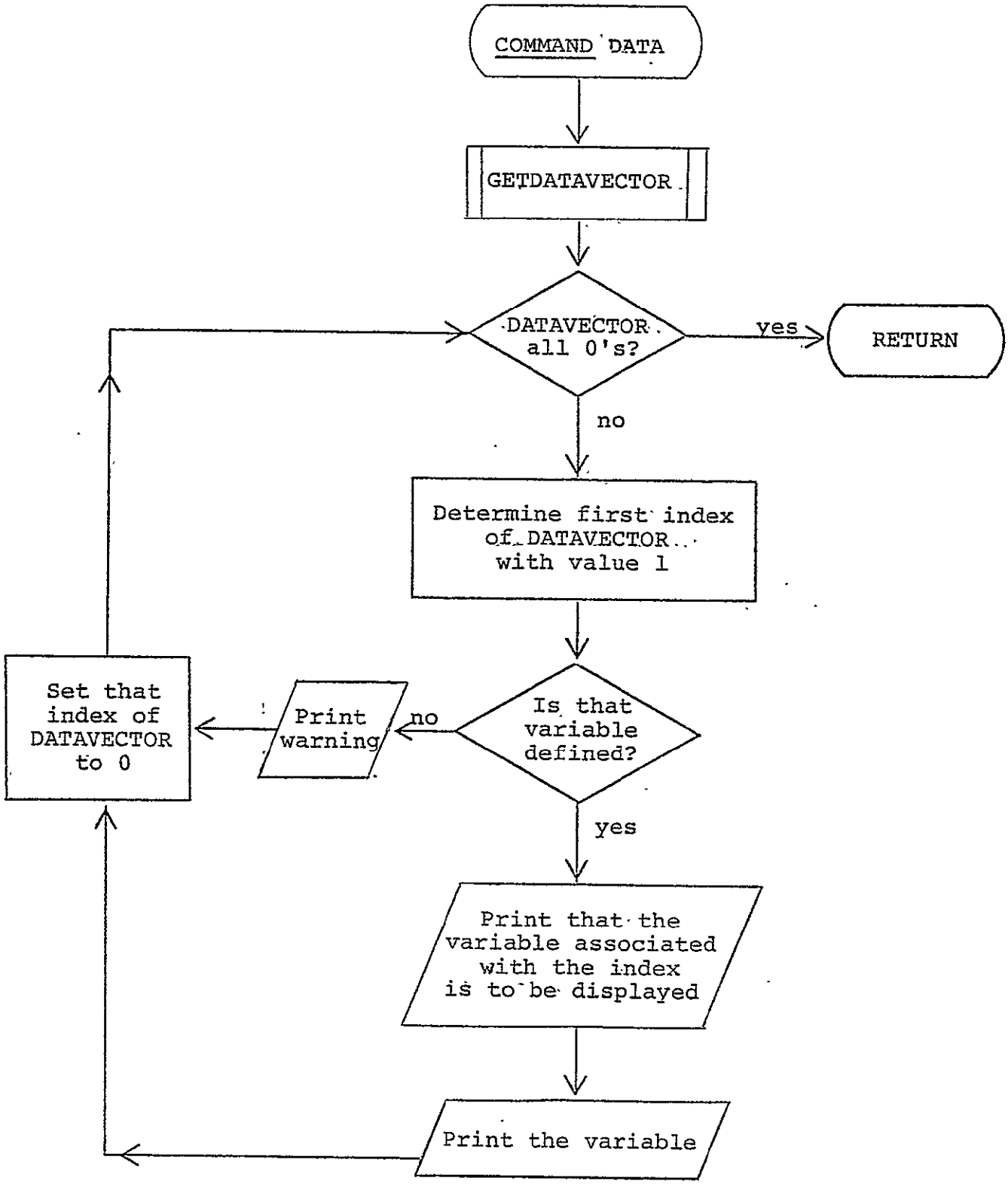


Figure 4. Flowchart for the function COMMANDDATA

3.2.2 ALTER and DATA

straightforward. They involve two of the few instances in METAPHOR where input is not handled by the INPUT function. A line is printed containing abbreviations of the variables which can be affected. The user must then place an X below each variable upon which he wishes to operate. The functions then check to see if the input was 'HELP' and if so, call COMMANDHELP to print the information; otherwise, they check the positions of the X's in the input string to determine which variables the user is interested in. The result of the input is then returned as either ALTERVECTOR or DATAVECTOR. Flowcharts of these functions are given in Figures 5-6.

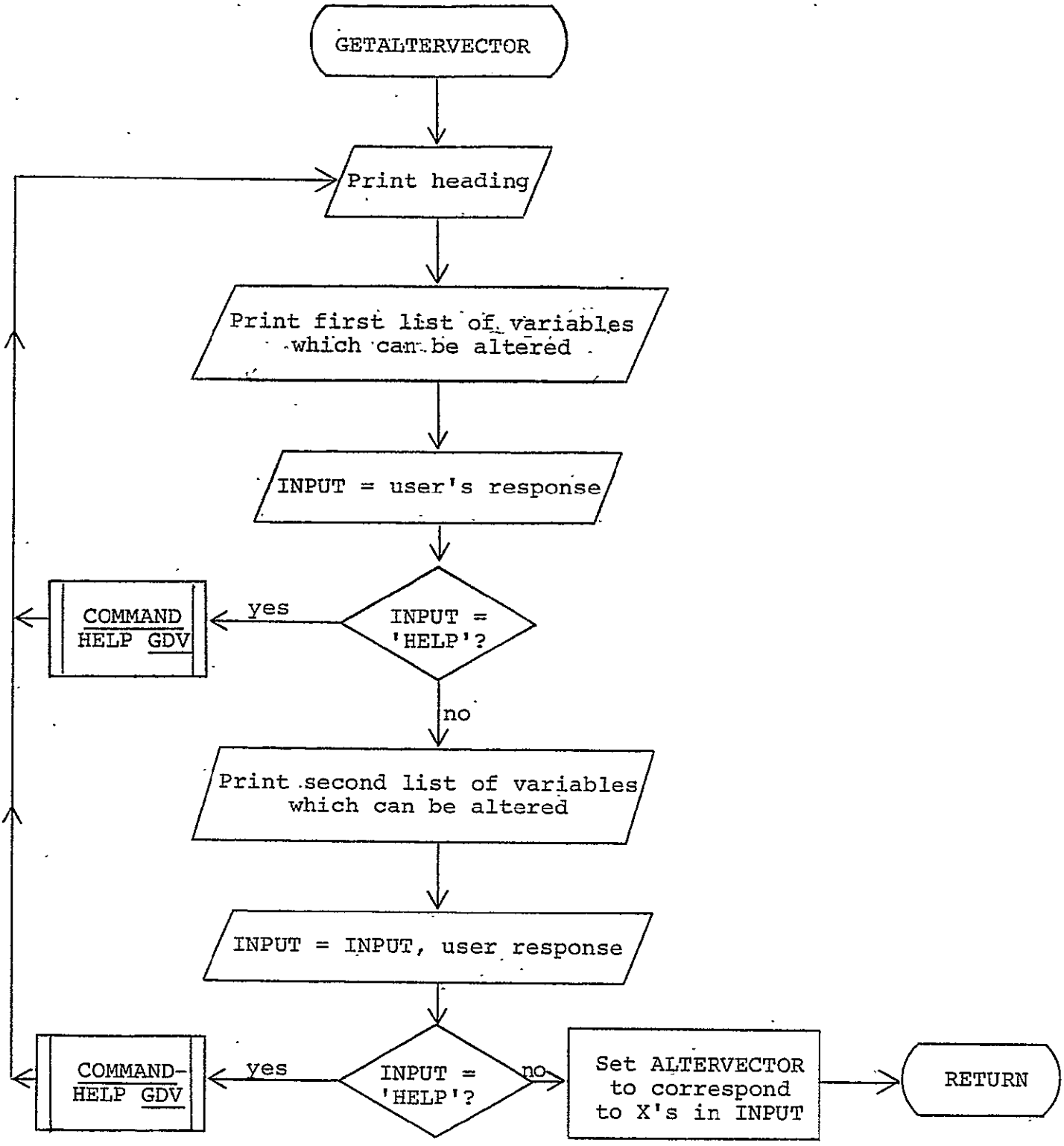


Figure 5. Flowchart for the function GETALTERVECTOR

3.2.2 ALTER and DATA

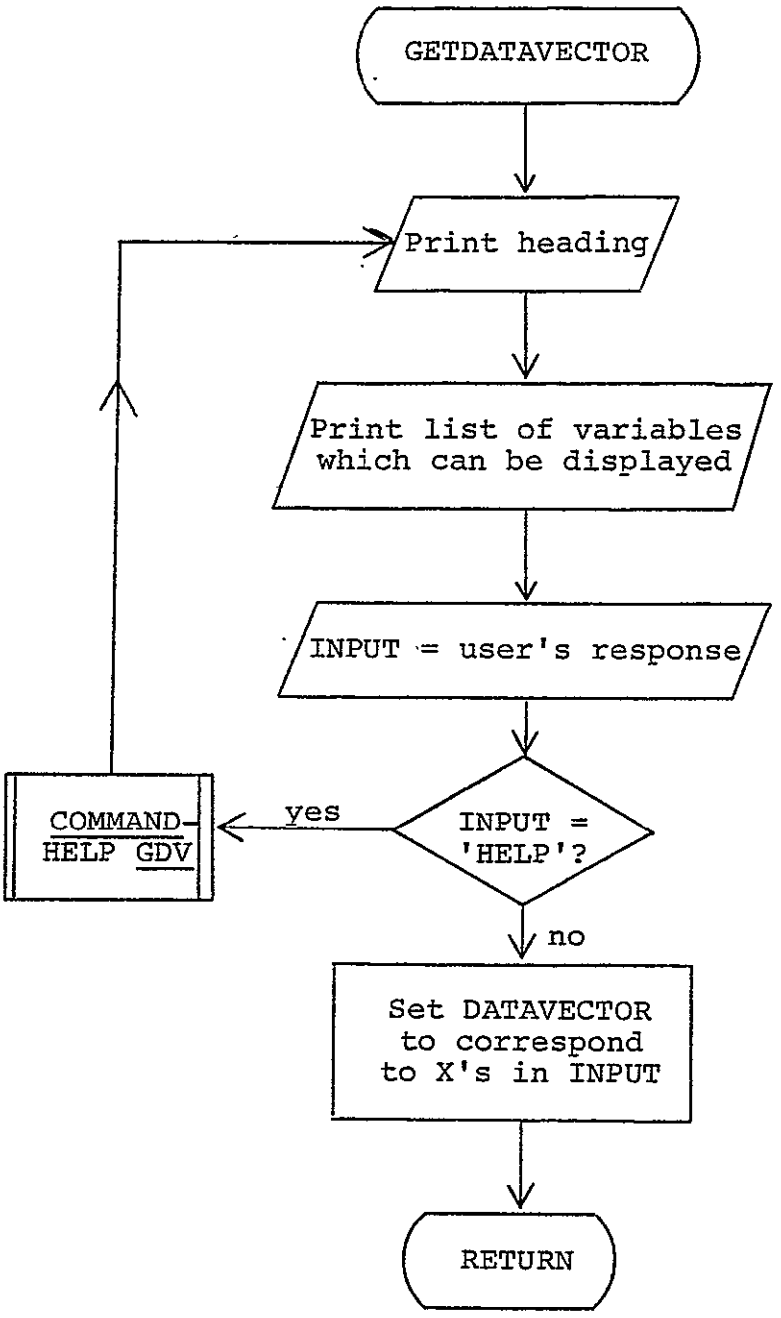


Figure 6. Flowchart for the function GETDATAVECTOR

4. Algorithms Employed in EVAL

Two important classes of algorithms are utilized in connection with the EVAL command to effect performability evaluation. The first of these classes deals with the automatic and semi-automatic generation of H and P matrices for the model, while the second class is concerned with the actual computation of the performability. Both categories are discussed below, the matrix generation algorithms in Section 4.1 and the computation algorithm in Section 4.2.

4.1 Matrix Generation

Currently, several matrix generation algorithms are implemented in METAPHOR. These are called DEDFAIL, NFAIL, and IDENTITY. The classes of matrices so generated have been found to be useful as commonplace classes of H and P matrices for degradable systems. (See the third Semi-Annual Status Report [3], Section 3.5.8.1, and the fourth Semi-Annual Status Report [4], Section 3.2.2.) the IDENTITY algorithm, which generates an identity matrix is straightforward and will not be discussed here. The NFAIL (Section 4.1.1) and DEDFAIL (Section 4.1.2) algorithms compute transition matrices for special types of systems. Each assumes that the structure of the system is described in terms of "components" where the state of each component is either "operational" or "failed." Both DEDFAIL and NFAIL assume that all components are alike and fail independently with the same constant failure rate. Finally, components are assumed to fail permanently, i.e., once a component has failed, it remains failed for the duration of the phase. The

difference between the two resides in how the states of the system are defined in terms of component states. DEDFAIL keeps track of each component in the system, i.e., whether a given component is operational or failed can be deduced from the state of the system. In METAPHOR, the most important use of DEDFAIL is in modeling a system wherein each component (e.g., processor) is dedicated to a different task (hence the name DEDFAIL). In such situations, the processing capability generally depends on the state of each component and hence the system state must convey the state of each component.

NFAIL, on the other hand, assumes that the components of the system are lumped into groups. NFAIL then keeps track only of the number of components which are operational within each of these groups. For instance, if two tasks and four processors are configured such that two processors are executing each task, then failure of either processor assigned to a given task will have the same effect on system performance. Accordingly, processors sharing the same task can be lumped, resulting in 2 groups with 2 processors per group. NFAIL is equivalent to DEDFAIL when NFAIL has n groups of 1 element each.

4.1.1 DEDFAIL

DEDFAIL generates a state transition matrix under the conditions stated above. If the system has N components, then the number of states declared for the phase must be a power of two. The resulting transition matrix is NxN where the (i,j)th entry denotes the probability that the system is in state j at the end of the

phase given it was in state i at the beginning of the phase. The i-th row or column of the matrix represents the state determined as follows: Assign each component a unique integer between 1 and N. Take the binary representation of (2**N)-1-i. Then the i-th digit of the binary representation (read from left to right) represents the state of the corresponding component in the system, 0 if failed, 1 if not failed.

For example, consider a system with two components having the transition in Figure 7. With a failure rate of 0.001 and a phase length of 10, the following transition matrix results:

	(1,1)	(1,0)	(0,1)	(0,0)
(1,1)	9.98E-1	9.99E-4	9.99E-4	9.99E-7
(1,0)	0.00E0	9.99E-1	0.00E0	1.00E-3
(1,0)	0.00E0	0.00E0	9.99E-1	1.00E-3
(0,0)	0.00E0	0.00E0	0.00E0	1.00E0

Here (1,1) means that both components are not failed, (1,0) that the first component is not failed but the second one is, (0,1) that the first component is failed but the second is not, and (0,0) that both components are failed.

DEDFAIL is equivalent to NFAIL when NFAIL has N groups of 1 component each.

DEDFAIL computes the transition matrix based on knowledge of the length of the phase, the failure rate of the components, and the number of states. This is accomplished by separately considering each entry (i,j) in the matrix. First, the number of transitions required for the entry is determined by taking the binary representation of the i-th row and j-th column as described above,

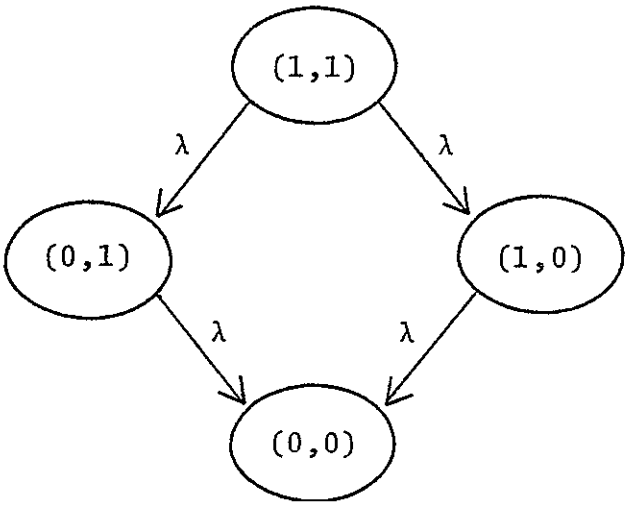


Figure 7. Transition diagram for the example of Section 4.1.1

subtracting j from i component by component, and counting the 1's--if there is a 1 in j which does not appear in the corresponding position of i, then the transition is impossible and so the probability is 0. Finally, the probability of making the transition from state i to state j is calculated using standard combinatorial methods. A flowchart for DEDFAIL is given in Figure 8. In METAPHOR, DEDFAIL is implemented in GDEDFAIL.

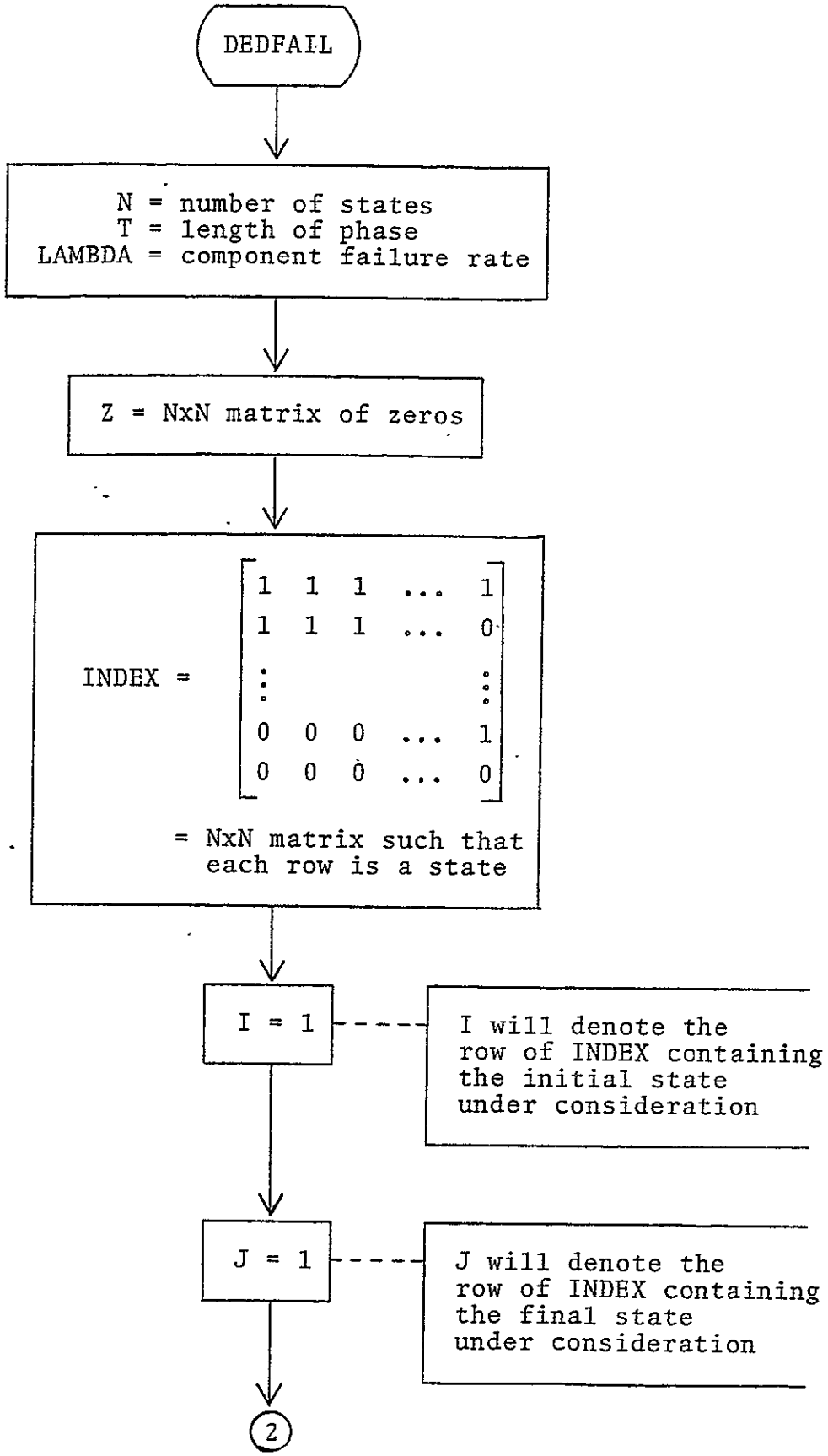


Figure 8. Flowchart for the algorithm DEDFAIL

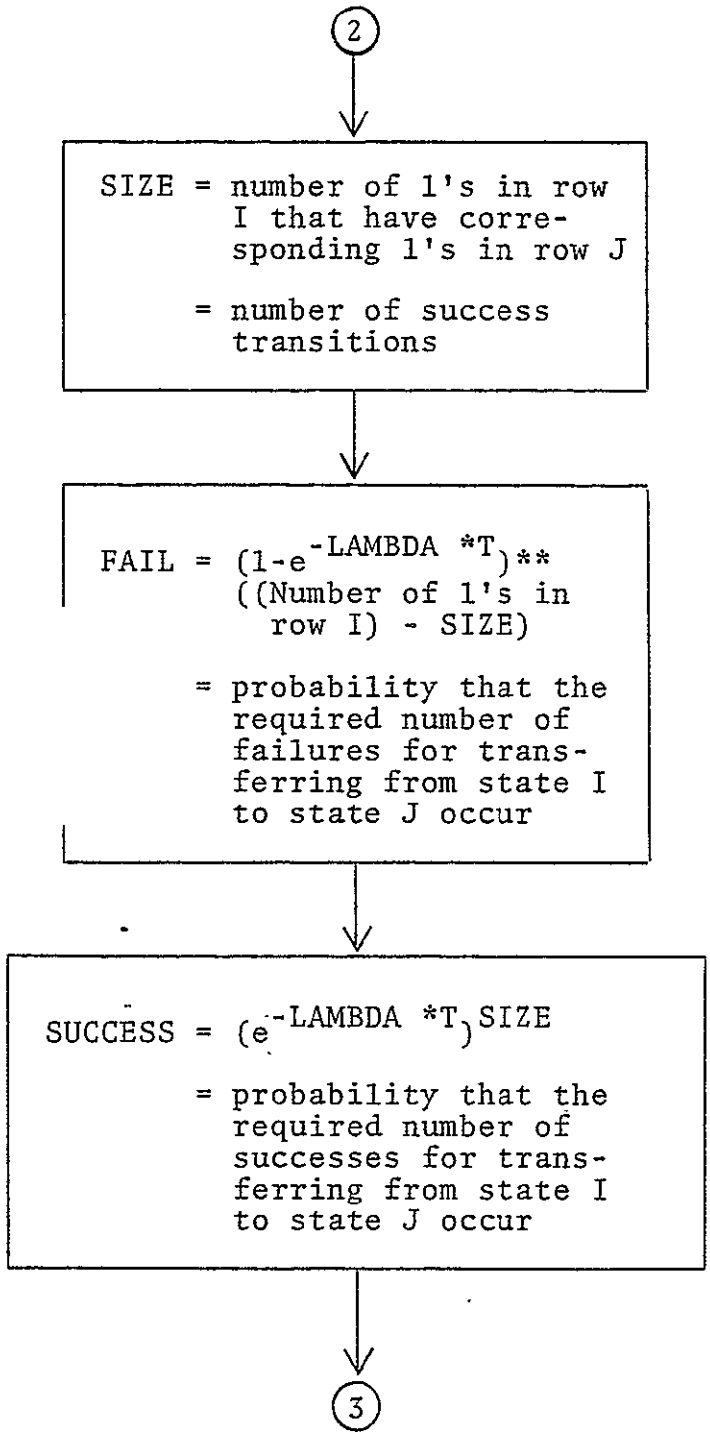


Figure 8. Flowchart for the algorithm DEDFAIL, continued

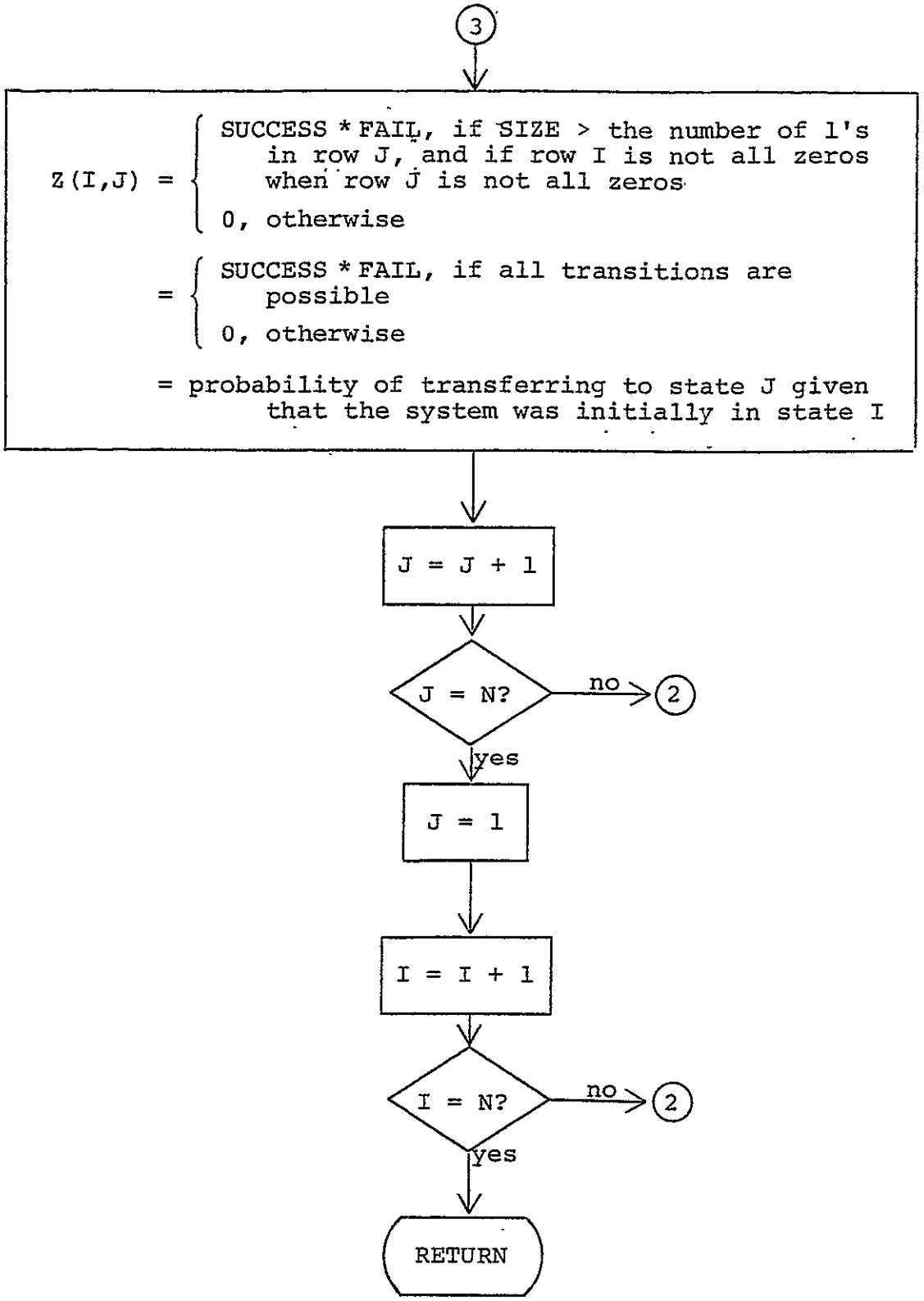


Figure 8. Flowchart for the algorithm DEDFAIL, continued

4.1.2 NFAIL

NFAIL generates a state transition matrix assuming the conditions given in Section 4.1. The state of the system represents the number of active (non-failed) components in each group. The number of states declared for the phase is determined as follows: to the number of components in each group add 1 and take the product of the resulting terms. For example, consider a system having 3 groups containing respectively 5, 2, and 7 components. The phase then has (5+1)*(2+1)*(7+1)=144 states. In general, if the number of states for the phase is in N, then the resulting matrix will be NxN in shape, such that the (i,j)th entry will be the probability that the system is in state j at the end of the phase given it was in state i at the beginning of the phase.

Consider a system having M groups and suppose the number of components in the m-th group is denoted by the function K(m). Then the i-th row or column of the resulting transition matrix represents the state determined as follows. First, take the mixed radix number system such that the j-th place (counted from the right) of a number has weight (multiplier):

$$\begin{cases} 1, & \text{if } j=1 \\ (K(1)+1)(K(2)+1)\ \ldots\ (K(j-1)+1), & \text{if } j>1 \end{cases}.$$

The values ("digits") that the j-th place can take on are 0, 1, ..., K(j). (The mechanism used to generate these radices is the APL ENCODE [or REPRESENTATION or "T"] function. Thus, values for each place are returned as entries in a vector, and so the values are <u>not</u> limited to 0, 1, ..., 9.) For instance, with the 5,2,7 system above, we would employ a number system having 3 places. The first place

has weight 1 and can range form 0 to 7, the second place has weight 7+1=8 and can vary from 0 to 2, and the third place has weight (7+1)*(2+1)=24 and can range from 0 to 2. The base ten number 55 would then be written '207' since 55 = 2*24 + 0*8 + 7*1, while the base ten number 17 would be expressed '021' because 17 = 0*24 + 2*8 + 1*1. A number represented in the above system has the following interpretation: the value of each digit of a number denotes the state of the corresponding group, i.e., the number of active (non-failed) components in the group. Hence '021' means groups 1, 2, and 3 have 1, 2, and 0 active components respectively. For example, consider a system with 2 groups, the first having 1 component and the second 2 components, having the transition diagram in Figure 9. With a transition rate of 0.001 and a period of 10, the following transition matrix results:

	(2,1)	(2,0)	(1,1)	(1,0)	(0,1)	(0,0)
(2,1)	9.704E-1	9.753E-3	1.951E-2	1.960E-4	9.802E-5	9.851E-7
(2,0)	0.000E0	9.802E-1	0.000E0	1.970E-2	0.000E0	9.901E-5
(1,1)	0.000E0	0.000E0	9.802E-1	9.851E-3	9.851E-3	9.901E-5
(1,0)	0.000E0	0.000E0	0.000E0	9.900E-1	0.000E0	9.950E-3
(0,1)	0.000E0	0.000E0	0.000E0	0.000E0	9.900E-1	9.950E-3
(0,0)	0.000E0	0.000E0	0.000E0	0.000E0	0.000E0	1.000E0

where (2,1) means that group 1 has one active components and group 2 has one active component, (2,0) means that group 1 has no active components and group 2 two active components, and so forth. NFAIL is equivalent to DEDFAIL when N groups of one component each are specified.

NFAIL computes the transition matrix based on knowledge of the length of the phase, the failure rate of the components, the number

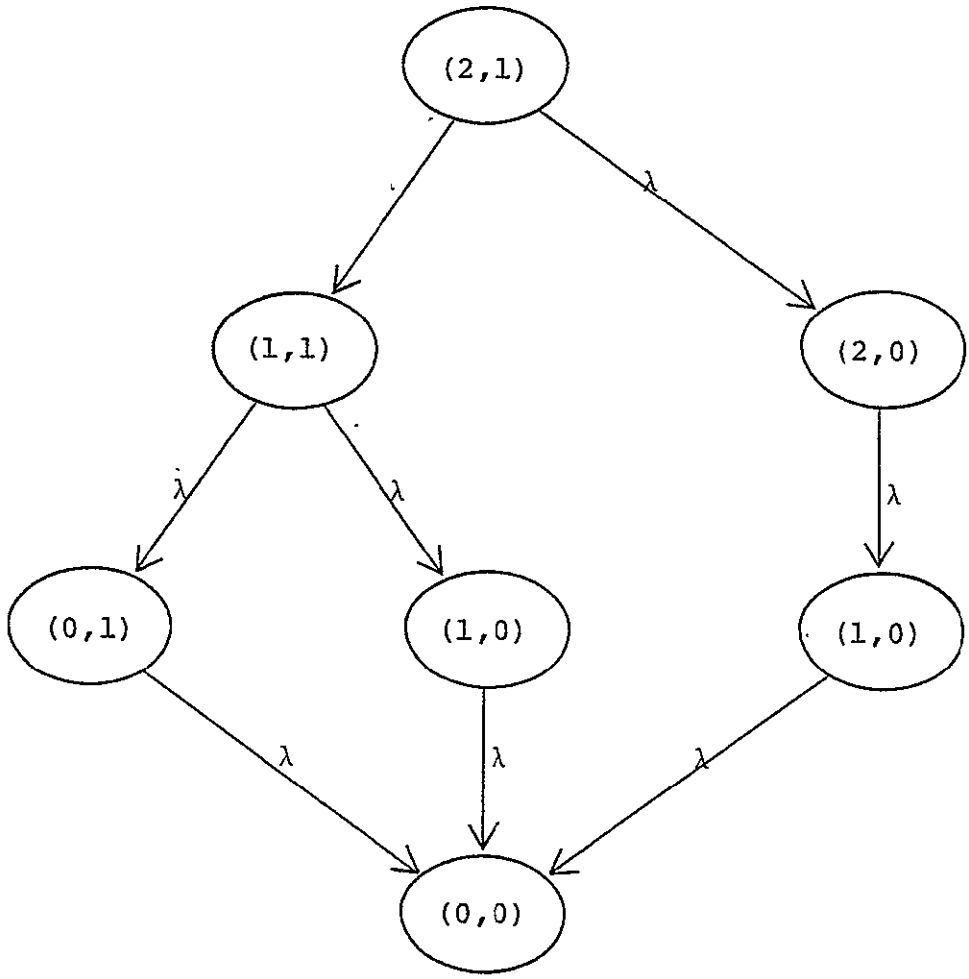


Figure 9. Transition diagram for the example of Section 4.1.2

4.1.2 NFAIL

of groups, and the number of components in each group. NFAIL computes the transition matrix by examining each entry in the matrix, determining the number of transitions required for the entry, and determining the probability of making those transitions. The number of transition required is computed as follows: Take the representation of the (i,j)th entry as described above, subtract j from i component by component, and sum the differences--if there is a value in some position of j which is greater than the value of the corresponding position of i, then the transition is impossible and so the probability is 0.

A flow chart for the algorithm NFAIL is given in Figure 10. In METAPHOR, NFAIL is implemented in GNFAIL.

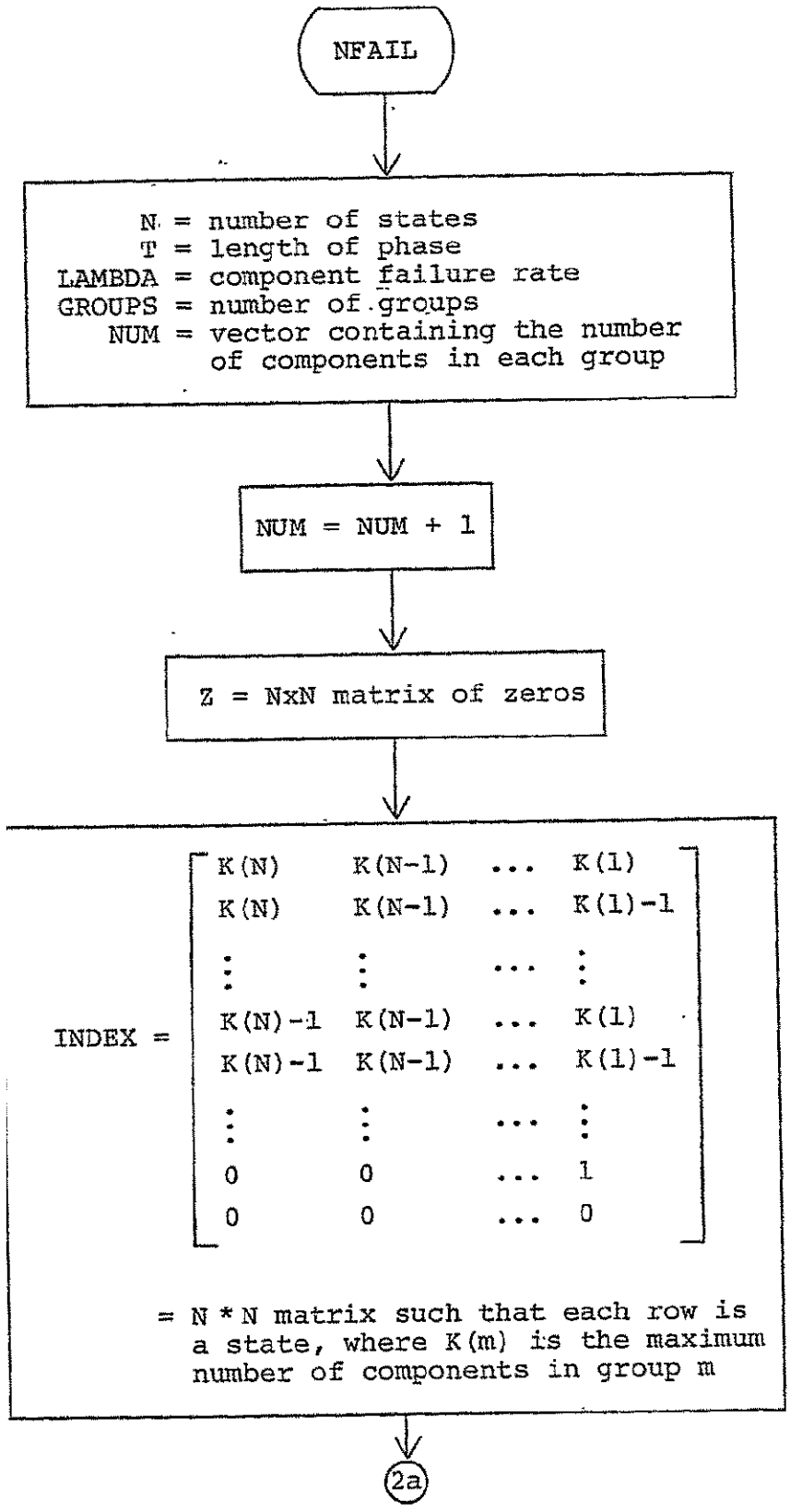


Figure 10. Flowchart for the algorithm NFAIL

4.1.2 NFAIL

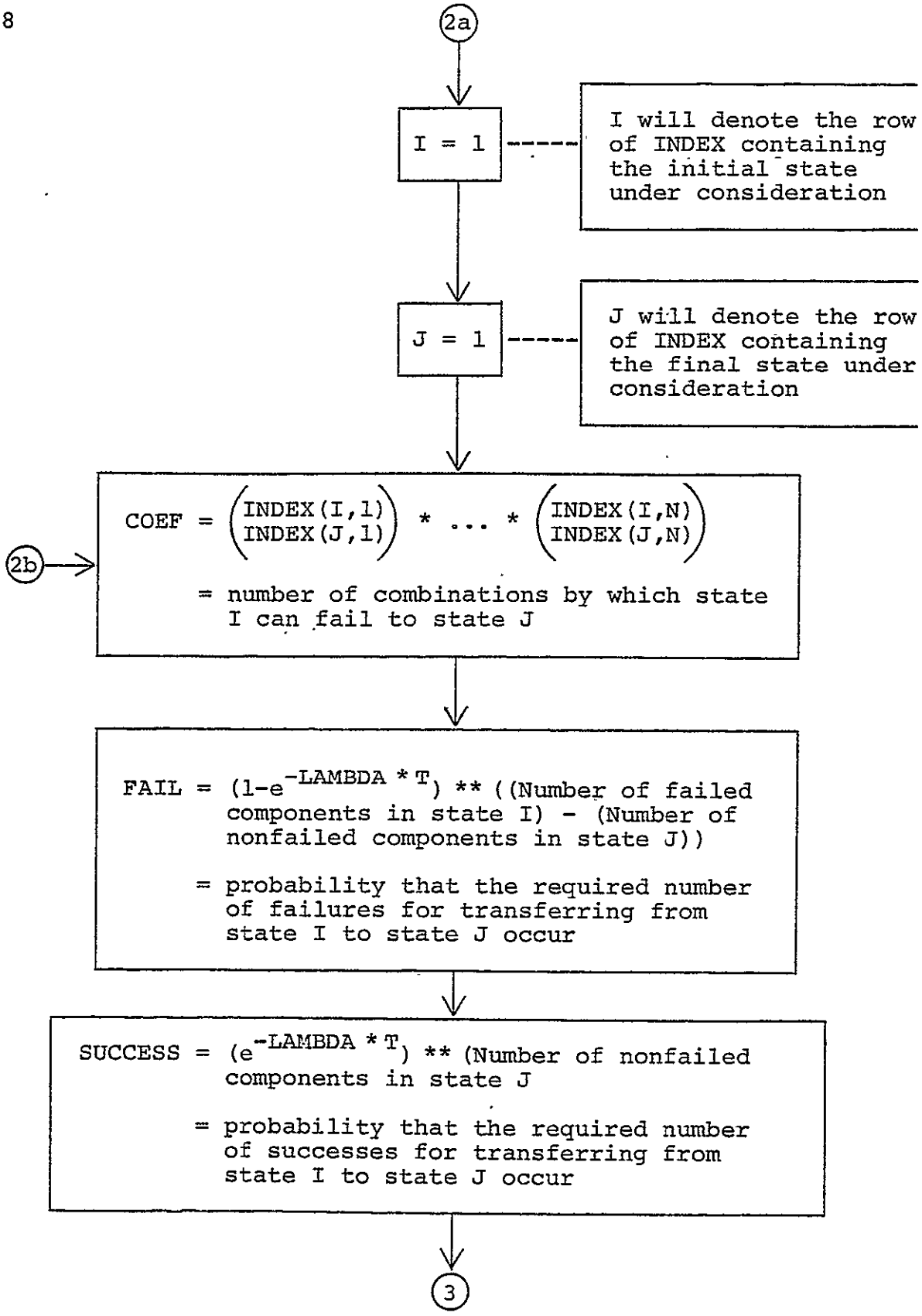


Figure 10. Flowchart for the algorithm NFAIL, continued

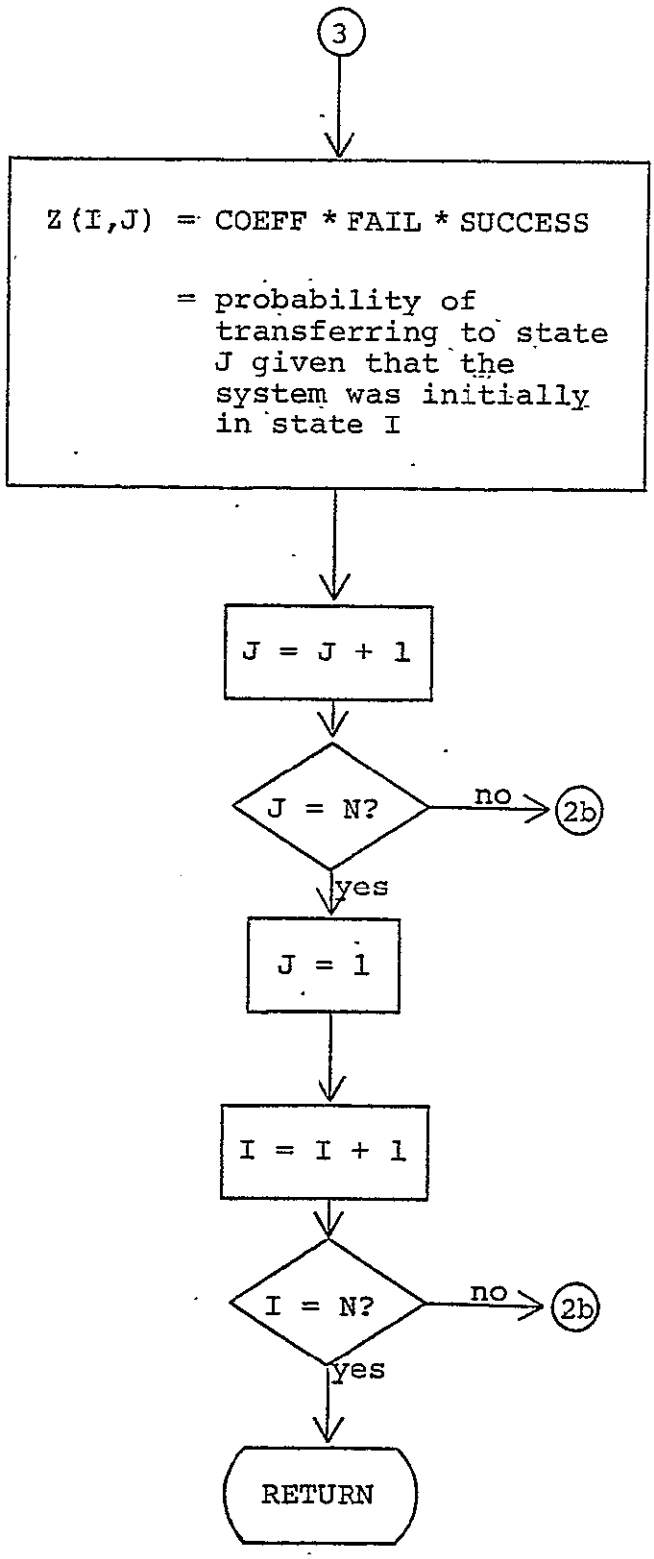


Figure 10. Flowchart for the algorithm NFAIL, continued

4.2 EVALUATION

Another algorithm presently implemented in METAPHOR automates the computation of trajectory set probabilities, i.e., step 5) of the major steps outlined in the introduction. This algorithm is based on the theory described in this third Semi-Annual Status Report ([3], Section 3.4). A flowchart representation of the algorithm is given in Figure 11.

The fetching of the number of phases, the number of states per phase, the intraphase (P) and interphase (H) matrices, the number of time-invariant basic variables, the probabilities of those time-invariant basic variables, and the number of accomplishment levels is straightforward. Next METAPHOR computes the probability of each accomplishment level separately. First, the number of trajectory sets in the accomplishment level is obtained, and then the probability of each trajectory set is determined. This is done by procuring the initial state (I) vector, the characteristic (G) matrices, and the characteristic (F) vector (See Section 3.4 of [3]; also, a characterization of the time-invariant basic variables is obtained. This latter characterization is called the "V vector" in METAPHOR and is similar to the I vector. That is, each entry in V is associated with an entry in V (say V[k]), and the probability that the event corresponding to that basic variable occurs during the mission is placed in V[k]. From these characterizations, the trajectory set probability is calculated according to

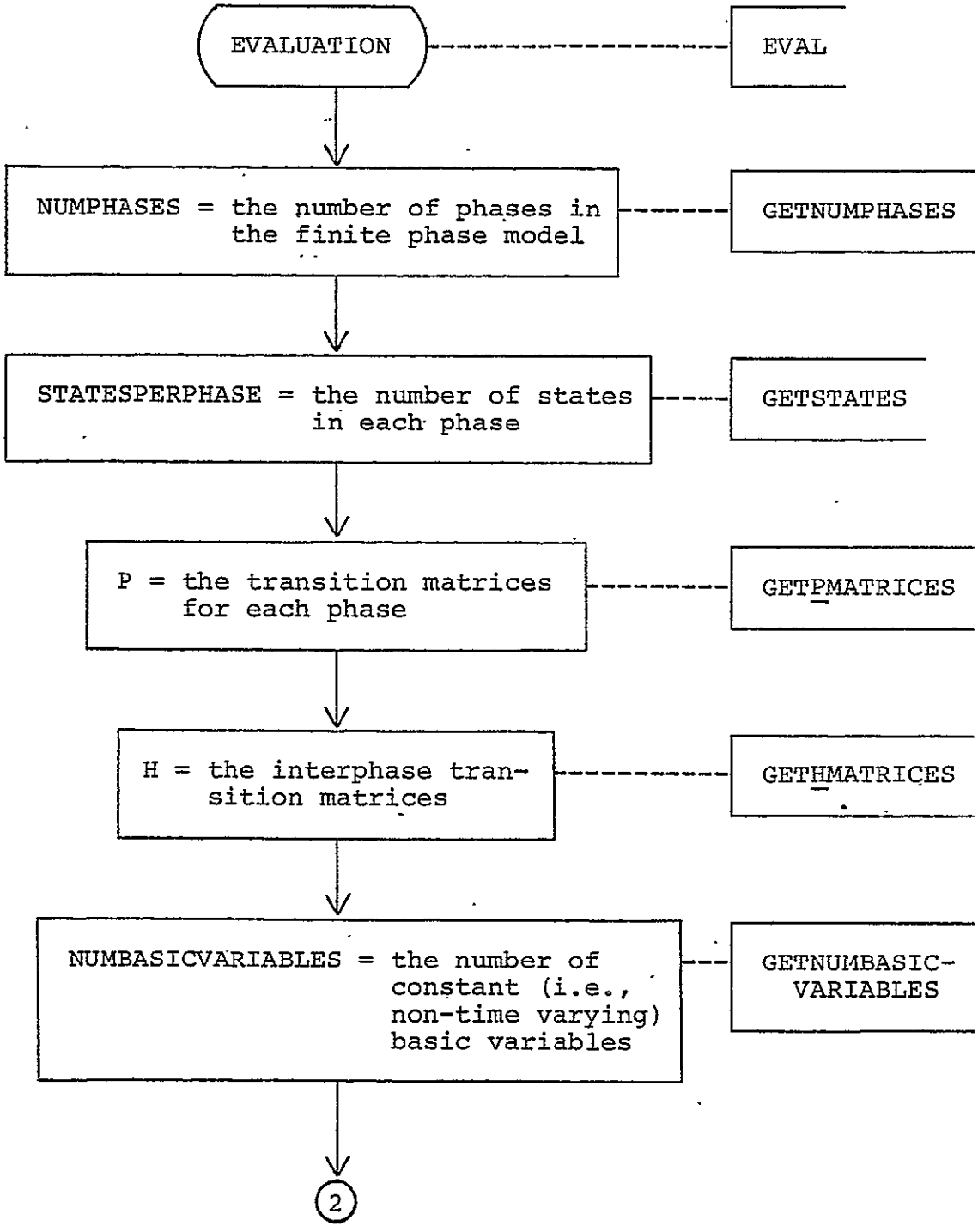


Figure 11. Flowchart for the algorithm EVALUATION

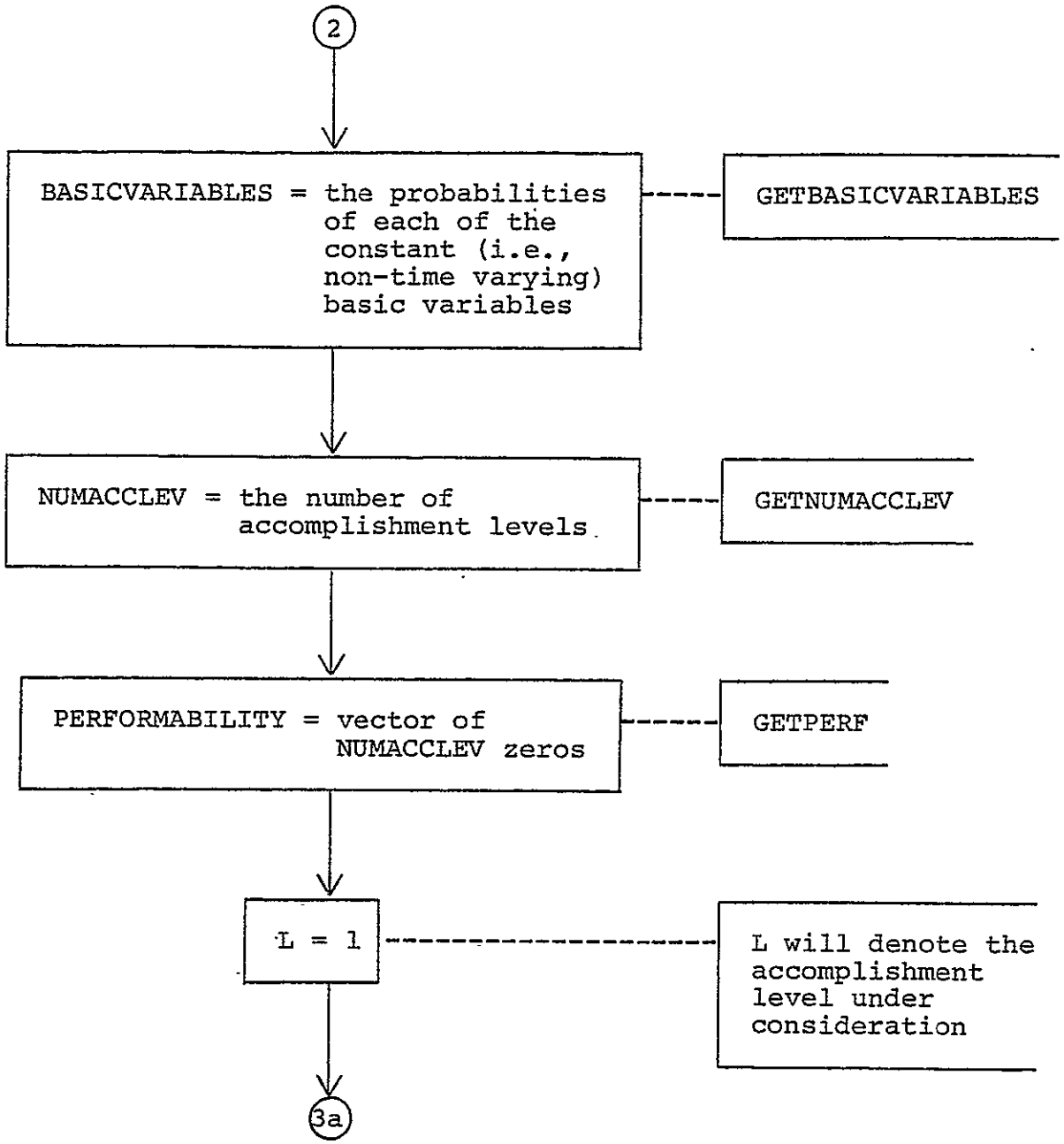


Figure 11. Flowchart for the algorithm EVALUATION, continued

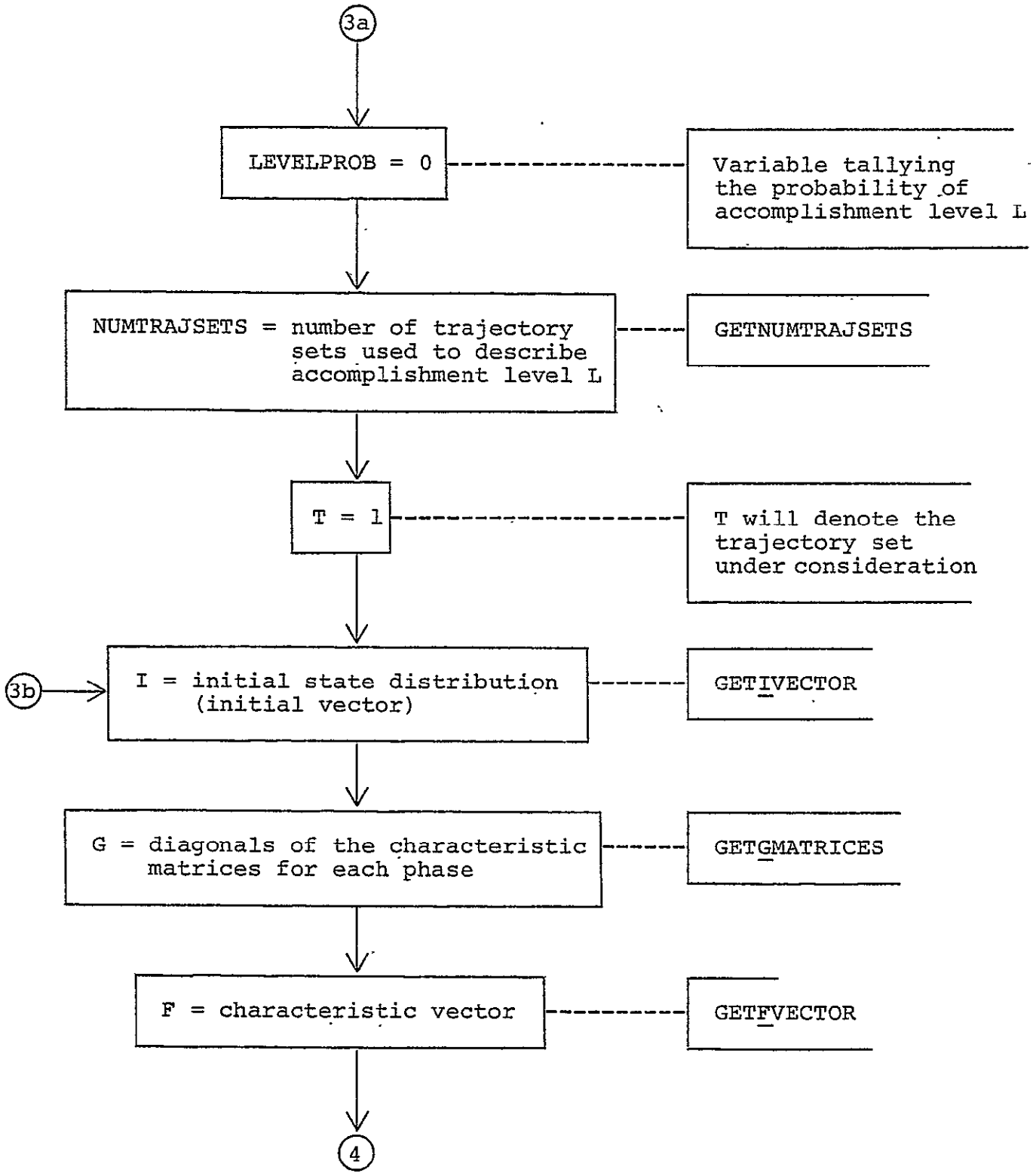


Figure 11. Flowchart for the algorithm EVALUATION, continued

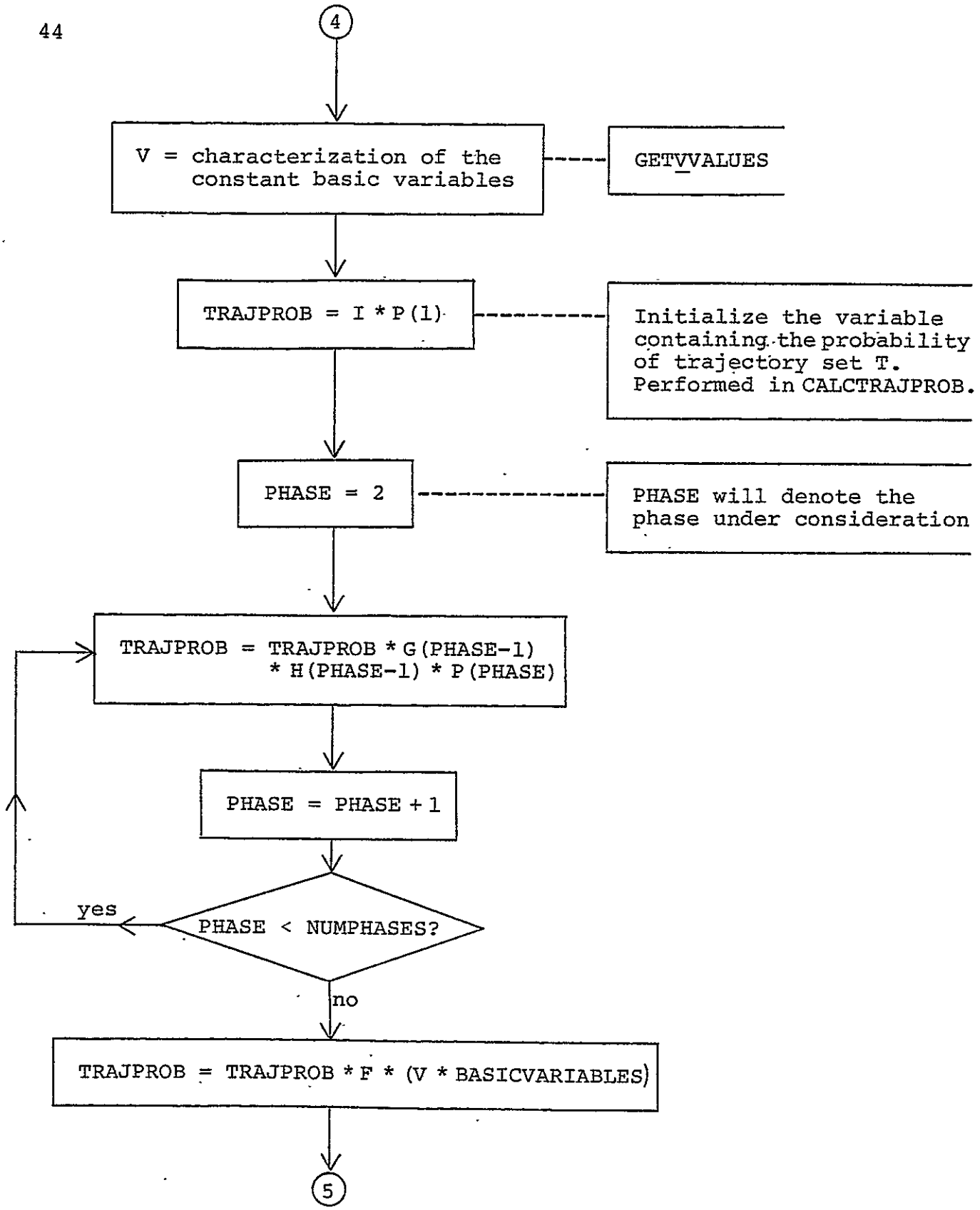


Figure 11. Flowchart for the algorithm EVALUATION, continued

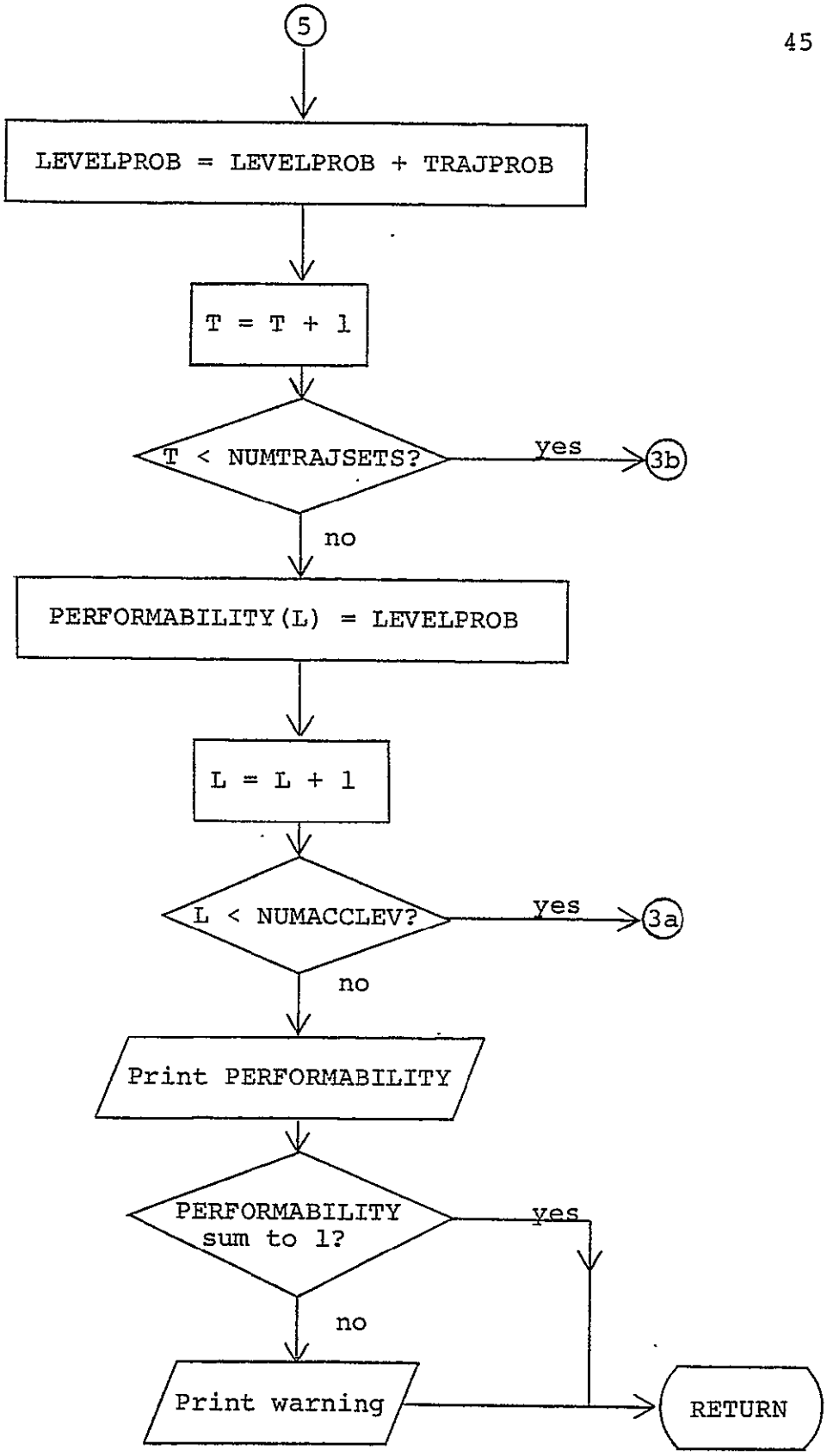


Figure 11. Flowchart for the algorithm EVALUATION, continued

$$\text{Probability} = I * P(1) * G(1) * H(1) * P(2) * G(2) * H(2) * \ldots * H(n) * F * V * \text{BASICVARIABLES}$$

where n is the number of phases, and * denotes matrix multiplication. In METAPHOR, the algorithm that computes these probabilities according to the above formulas is implemented in the function COMMANDEVAL and the various functions associated with COMMANDEVAL.

5. Description of METAPHOR Variables

This section presents descriptions of each METAPHOR global variable as well as the more important local variables. The latter are primarily local variables utilized in more than one function. In addition to the descriptions, the functions making reference to each variables are listed in parenthesis after the variable name.

ALTER (DECLAREMETAPHOR) Constant containing the string 'ALTER '. Used for inputting the ALTER command.

BASICVARIABLES (CALCTRAJPROB, GETBASICVARIABLES, DATA) Vector of length 2*NUMBASICVARIABLES. The first half of BASICVARIABLES contains the probability of the corresponding time-invariant basic variable, while the second half contains one minus the corresponding entries ir the first half of BASICVARIABLES.

BRIEF (DECLAREMETAPHOR) Constant containing the string 'BRIEF '. Used for inputting the BRIEF command.

BRIEFOUTPUT (DECLAREMETAPHOR, PRINT, PRINTQUAD, COMMANDBRIEF) scalar flag set to 1 if terse output desired, set to 0 if not.

CALC (DECLAREMETAPHOR) Constant containing the string 'CALC Used for inputting the CALC command.

CHECKNO (CHECKBIN, CHECKPOSI, CHECKPROB, CHECKTRI) Local variable containing the value to be checked.

COM (DECLAREMETAPHOR) Constant containing the string 'CALC Used for inputting the CALC command.

COMMANDLIST (DECLAREMETAPHOR, INPUT) Array list of valid commands. Used to determine whether an input is a command, and if so, which command.

COMMANDSIZE (DECLAREMETAPHOR) scalar constant denoting the maximum length of the valid commands.

DATA (DECLAREMETAPHOR) Constant containing the string 'DATA '. Used for inputting the DATA command.

DEDFAIL (DECLAREMETAPHOR) Constant vector containing the string 'DEDFAIL '. Used for inputting the DEDFAIL matrix generator instruction.

DEFACCLEVEL (DECLAREMETAPHOR, GETPERFORMABILITY, COMMANDALTER) Definition flag denoting whether the accomplishment levels have been defined. The value is 1 if so, 0 otherwise.

DEFBASICVARIABLES (DECLAREMETAPHOR, GETBASICVARIABLES, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the time-invariant basic variables have been defined. The value is 1 if so, 0 otherwise.

DEFF (DECLAREMETAPHOR, GETACCLEVPROB, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the final vector has been defined. The value is 1 if so, 0 otherwise.

DEFG (DECLAREMETAPHOR, GENERATEHMATRIX4, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the characterisitc matrices have been defined. The value is 1 if so, 0 otherwise.

DEFH (DECLAREMETAPHOR, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the interphase transition matrices have been defined. The value is 1 if so, 0 otherwise.

DEFI (DECLAREMETAPHOR, GETACCLEVPROB, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the initial vector has been defined. The value is 1 if so, 0 otherwise.

DEFNUMACCLEV (DECLAREMETAPHOR, GETNUMACCLEV, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the number of accomplishment levels has been defined. The value is 1 if so, 0 otherwise.

DEFNUMBASICVARIABLES (DECLAREMETAPHOR, GETNUMBASICVARIABLES, COMMANDDATA) Definition flag denoting whether the number of basic variables has been defined. The value is 1 if so, 0 otherwise.

DEFNUMPHASES (DECLAREMETAPHOR, GETNUMPHASES, COMMANDDATA) Definition flag denoting whether the number of phases has been defined. The value is 1 if so, 0 otherwise.

DEFNUMSTATES (DECLAREMETAPHOR, GETSTATES, COMMANDDATA) Definition flag denoting whether the number of states has been defined. The value is 1 if so, 0 otherwise.

DEFNUMTRAJSETS (DECLAREMETAPHOR, GETACCLEVPROB, GETNUMTRAJSETS, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the number of trajectory sets has been defined. The value is 1 if so, 0 otherwise.

DEFP (DECLAREMETAPHOR, GETPMATRICES, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the intraphase transition matrices have been defined. The value is 1 if so, 0 otherwise.

DEFPERFORMABILITY (DECLAREMETAPHOR, GETPERFORMABILITY, COMMANDDATA) value of the performability of the system have been defined. The value is 1 if so, 0 otherwise.

DEFV (DECLAREMETAPHOR, GETACCLEVPROB, COMMANDALTER, COMMANDDATA) Definition flag denoting whether the vector characterizing the time-invariant basic variables have been defined. The value is 1 if so, 0 otherwise.

ECHOINPUT (DECLAREMETAPHOR, INPUT, INYES, COMMANDECHO) A scalar flag set to 1 if the echoing of input desired, set to 0 if not.

F (CALCTRAJPROB, DATA, GETACCLEVPROB) Vector local to the function GETACCLEVPROB containing the characteristic (F) vector. Its shape is NUMSTATES[NUMPHASES].

GIVEN (DECLAREMETAPHOR) Constant vector containing the string 'GIVEN '. Used for inputting the GIVEN matrix instruction.

H (CALCTRAJPROB, GETHMATRICES, DATA) Array containing the interphase (H) matrices. Its shape is (NUMPHASES-1)x(MAXNUMSTATES)x(MAXNUMSTATES).

G (CALCTRAJPROB, DATA, GETACCLEVPROB) Array local to the function GETACCLEVPROB containing the characteristic (G) matrices. Its shape is (MAXNUMSTATES)x(MAXNUMSTATES).

HELP (DECLAREMETAPHOR) Constant containing the string 'HELP ' Used for inputting the HELP command.

I (CALCTRAJPROB, DATA, GETACCLEVPROB) Vector local to the function GETACCLEVPROB containing the initial (I) vector. Its shape is NUMSTATES[1].

IDENTITY (DECLAREMETAPHOR, GENERATEHMATRIX, GENERATEPMATRIX) constant vector containing the string 'IDENTITY'. Used for inputting the identity matrix generator instruction.

IN (GENERATEHMATRIX, GENERATEPMATRIX, GETALTERVECTOR, GETBASICVARIABLES, GETNUMACCLEV, GETNUMBASICVARIABLES, GETNUMPHASES, GETNUMTRAJSETS, GETSTATES, GETFVECTOR, GETGMATRICES, GETHMATRICES, GETIVECTOR, GETPMATRICES, GETVVALUES, INPUT, GDEDFAIL, GGIVEN, GNFAIL) vector containing the latest data input to METAPHOR.

MAXNUMSTATES (GETSTATES, GETGMATRICES, GETHMATRICES, GETIVECTOR, GETPMATRICES) scalar denoting the maximum number of states in any phase of the finite phase model.

NFAIL (DECLAREMETAPHOR) Constant vector containing the string 'NFAIL '. Used for inputting the NFAIL matrix generator instruction.

NUMACCLEV (GETNUMACCLEV, GETPERFORMABILITY) Scalar denoting the number of accomplishment levels in the model.

NUMBASICVARIABLES (GETNUMACCLEV, GETPERFORMABILITY) Scalar denoting the number of time-invariant basic variables in the model.

NUMPHASES (GETNUMPHASES, GETHMATRICES, GETPMATRICES) Scalar denoting the number of phases in the model.

NUMSTATES (GETSTATES) Vector denoting the number of states in each phase of the model.

NUMTRAJSETS (GETACCLEVPROB, GETPERFORMABILITY) Scalar denoting the number of trajectory in the accomplishment level currently under consideration.

OFF (DECLAREMETAPHOR) Constant containing the string 'OFF Used for inputting the ON command parameter.

ON (DECLAREMETAPHOR) Constant containing the string 'ON Used for inputting the ON command parameter.

P (CALCTRAJPROB, GETPMATRICES, DATA) Array containing the intraphase (P) matrices. Its shape is (NUMPHASES)x(MAXNUMSTATES)x(MAXNUMSTATES).

PERFORMABILITY (GETPERFORMABILITY, PRINTPERFORMABILITY) Scalar containing the calculated value of the performability for the model.

Q (PRINT, PRINTQUAD) Local variable containing the text to be printed.

ROUTINE (INPUT) Local variable containing the code of the routine calling INPUT.

SHAPE (INPUT) Local variable containing the expected dimension of the input.

STATESPERPHASE (GETSTATES, GETFVECTOR, GETGMATRICES, GETHMATRICES, GETIVECTOR, GETPMATRICES) Vector containing the number of states in each phase.

SWITCH (BRIEF, ECHO) Local variable containing the command 'ON' or 'OFF'.

V (CALCTRAJPROB, DATA, GETACCLEVPROB) Vector local to the function GETACCLEVPROB containing the vector characterizing the time-invariant basic variables in the model. (See Section 4.2.). Its shape is NUMBASICVARIABLES.

Z (BRIEF, CALCTRAJPROB, CHECKBIN, CHECKPOSI, CHECKTRI, ECHO, ENCODE, GENERATEHMATRIX, GENERATEPMATRIX, GETACCLEVPROB, GETBASICVARIABLES, GETDATAVECTOR, GETNUMTRAJSETS, GETFVECTOR, GETGMATRICES, GETIVECTOR, GETVVALUES, INYES, PRINT, PRINTQUAD, GDEDFAIL, GGIVEN, GIDENTITY, GNFAIL) Local variable which will be passed the result of the function.

GAV (DECLAREMETAPHOR, GETALTERVECTOR) Constant used to identify the function GETALTERVECTOR. Contains the value 16. Used by COMMANDHELP.

GBV (DECLAREMETAPHOR, GETBASICVARIABLES) Constant used to identify the function GETBASICVARIABLES. Contains the value 9. Used by COMMANDHELP.

GD (DECLAREMETAPHOR, GDEDFAIL) Constant used to identify the function GDEDFAIL. Contains the value 6. Used by COMMANDHELP.

GDV (DECLAREMETAPHOR, GETDATAVECTOR) Constant used to identify the function GETDATAVECTOR. Contains the value 17. Used by COMMANDHELP.

GFV (DECLAREMETAPHOR, GETFVECTOR) Constant used to identify the function GETFVECTOR. Contains the value 13. Used by COMMANDHELP.

GGM (DECLAREMETAPHOR, GETGMATRICES) Constant used to identify the function GETGMATRICES. Contains the value 12. Used by COMMANDHELP.

GG (DECLAREMETAPHOR, GGIVEN) Constant used to identify the function GGIVEN. Contains the value 4. Used by COMMANDHELP.

GHM (DECLAREMETAPHOR, GENERATEHMATRIX) Constant used to identify the function GETHMATRICES. Contains the value 7. Used by COMMANDHELP.

GIV (DECLAREMETAPHOR, GETIVECTOR) Constant used to identify the function GETIVECTOR. Contains the value 11. Used by COMMANDHELP.

GN (DECLAREMETAPHOR, GETACCLEVPROB) Constant used to identify the function GNFAIL. Contains the value 5. Used by COMMANDHELP.

GNA (DECLAREMETAPHOR, GETNUMACCLEV) Constant used to identify the function GETNUMACCLEV. Contains the value 10. Used by COMMANDHELP.

GNBV (DECLAREMETAPHOR, GETNUMBASICVARIABLES) Constant used to identify the function GETNUMBASICVARIABLES. Contains the value 8. Used by COMMANDHELP.

GNP (DECLAREMETAPHOR, GETNUMPHASES) Constant used to identify the function GETNUMPHASES. Contains the value 1. Used by COMMANDHELP.

GNTS (DECLAREMETAPHOR, GETNUMTRAJSETS) Constant used to identify the function GETNUMTRAJSETS. Contains the value 15. Used by COMMANDHELP.

GPM (DECLAREMETAPHOR, GENERATEPMATRIX) Constant used to identify the function GETPMATRICES. Contains the value 3. Used by COMMANDHELP.

GS (DECLAREMETAPHOR, GETSTATES) Constant used to identify the function GETSTATES. Contains the value 2. Used by COMMANDHELP.

GVV (DECLAREMETAPHOR, GETVVALUES) Constant used to identify the function GETVVECTOR. Contains the value 14. Used by COMMANDHELP.

HMATRIXLIST (DECLAREMETAPHOR, GENERATEHMATRIX) Array list of valid H matrix types. Used to determine whether an input is an H matrix type, and if so, which type.

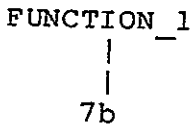
MET (DECLAREMETAPHOR, METAPHOR) Constant used to identify the function METAPHOR. Contains the value 18. Used by COMMANDHELP.

PMATRIXLIST (DECLAREMETAPHOR, GENERATEPMATRIX) Array list of valid P matrix types. Used to determine whether an input is an P matrix type, and if so, which type.

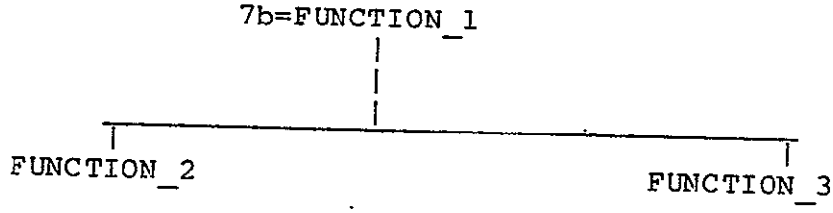
6. Interdependence of METAPHOR Functions

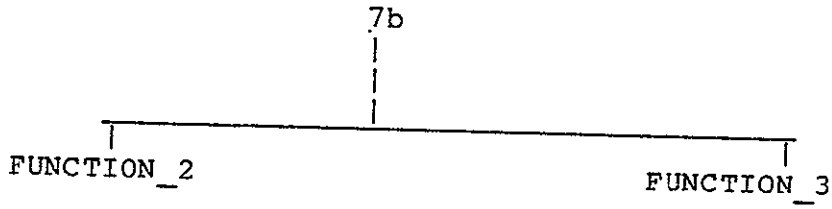
This section discusses the interdependence of METAPHOR functions expressed as a hierarchy denoting which functions call which other functions. Specifically, the graph of Figure 12 is used to portray these relationships.

Each function F in METAPHOR is included at least once in the graph; every function that is called by F is connected below F in the graph, while one function that calls F is connected above. If F does not call other functions, then no functions are listed below F, while if no functions call F (as is the case with METAPHOR, BRIEF, and ECHO), then no function is listed above F. Occasionally, either insufficient room is available on the page to place a segment of the graph, or else a segment has already been positioned somewhere else. In this case, a number (or a number and a letter) is placed below the function to indicate a continuation on some other page of the diagram. The number refers to the page of the diagram with the continuation. The number (or number and letter) is placed somewhere on the continuation either with an equal sign (=) next to the function name for which it denotes a continuation or else with a list of functions below it. For example, if at some location the following entry occurs



then on the seventh page of the diagram, one would find one of the situations below:





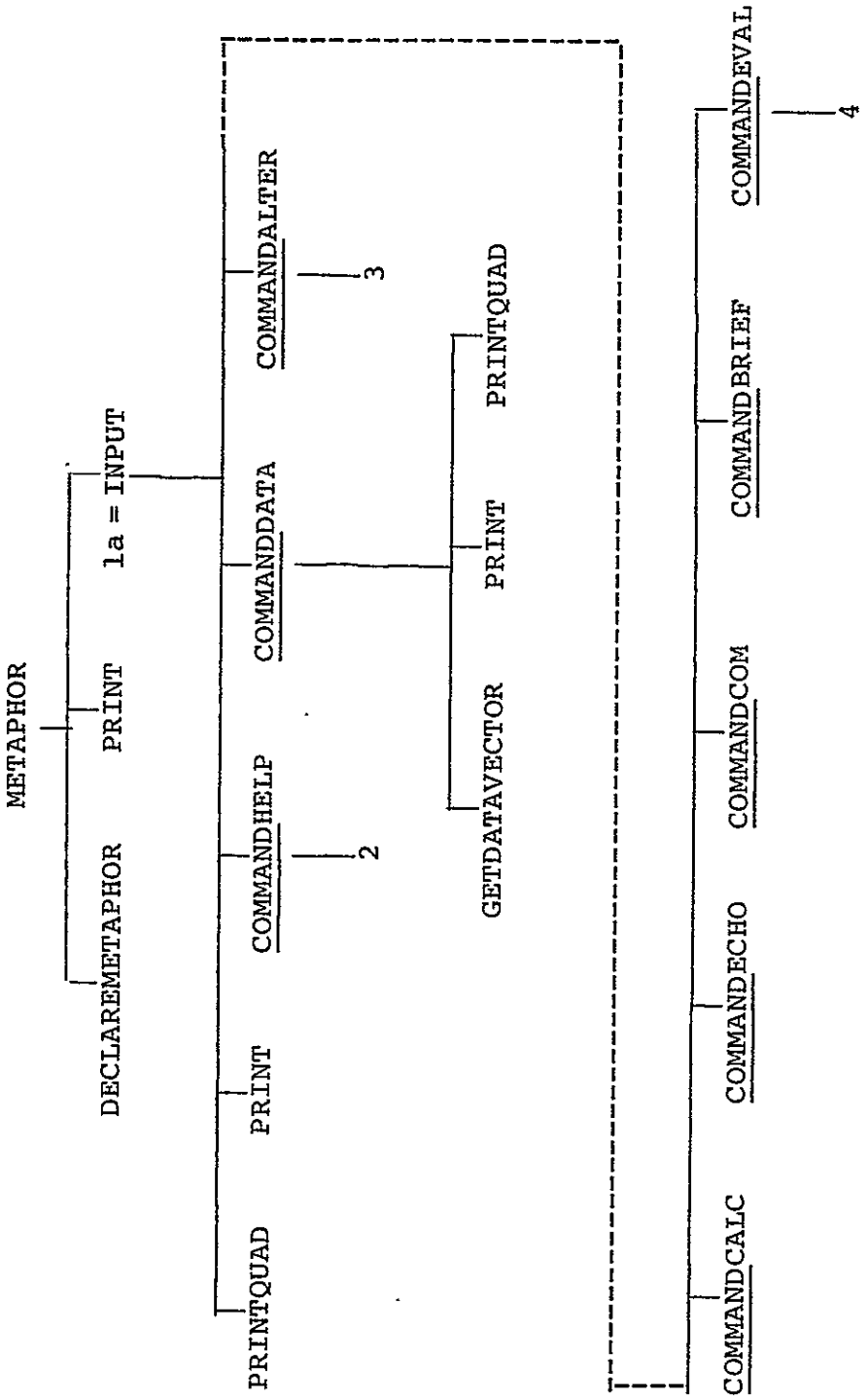


Figure 12. Interdependence of METAPHOR functions

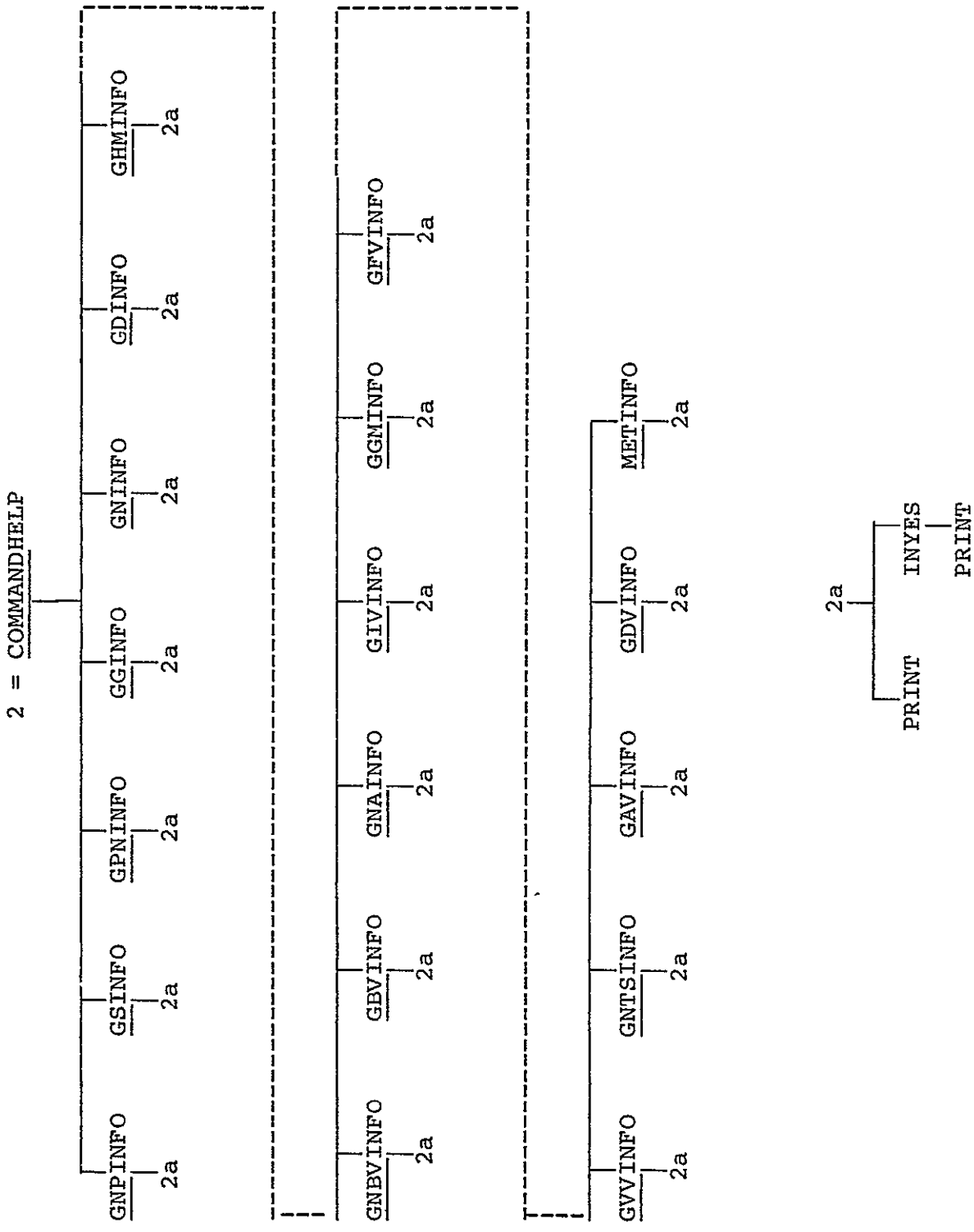


Figure 12. Interdependence of METAPHOR functions, continued

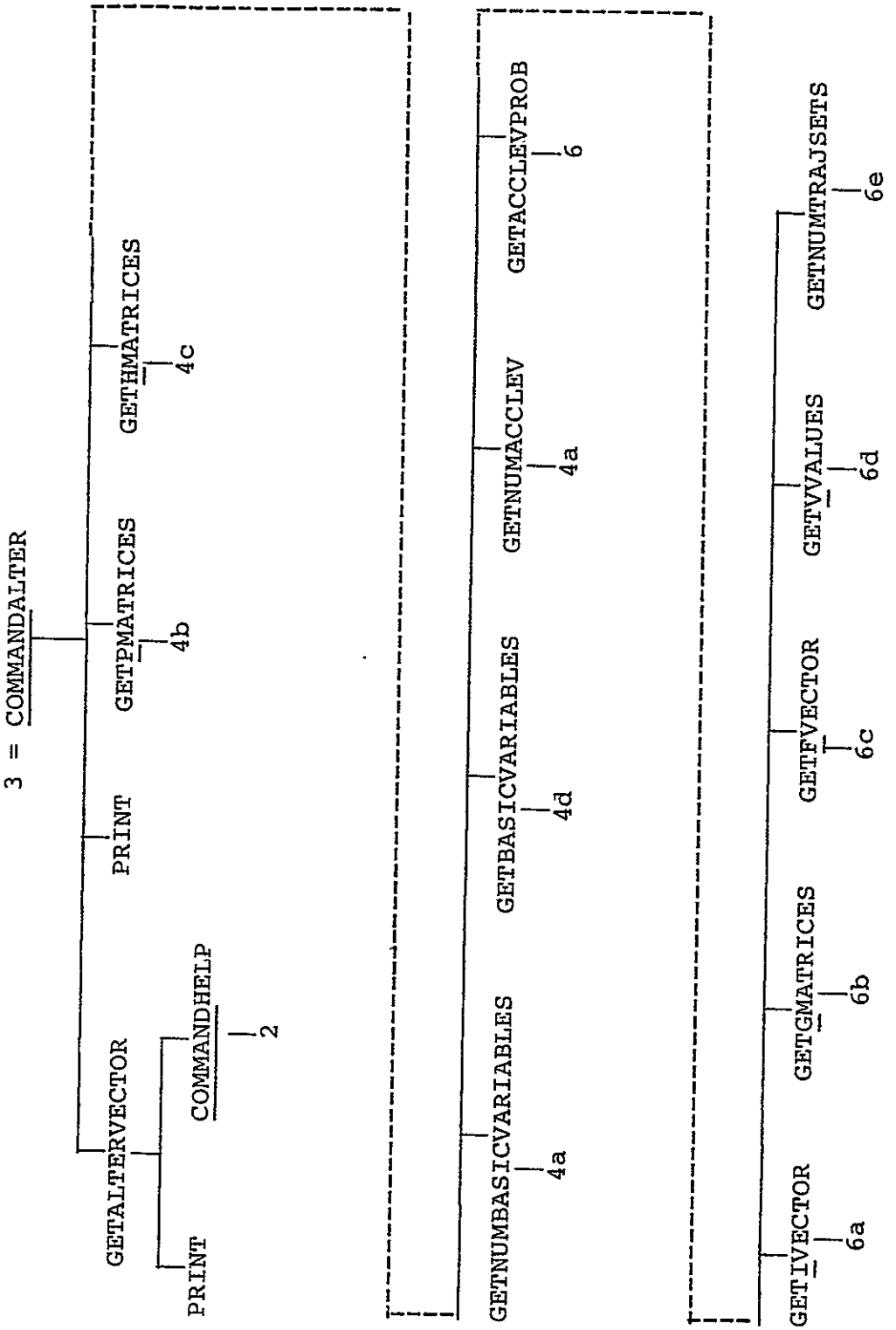


Figure 12. Interdependence of METAPHOR functions, continued

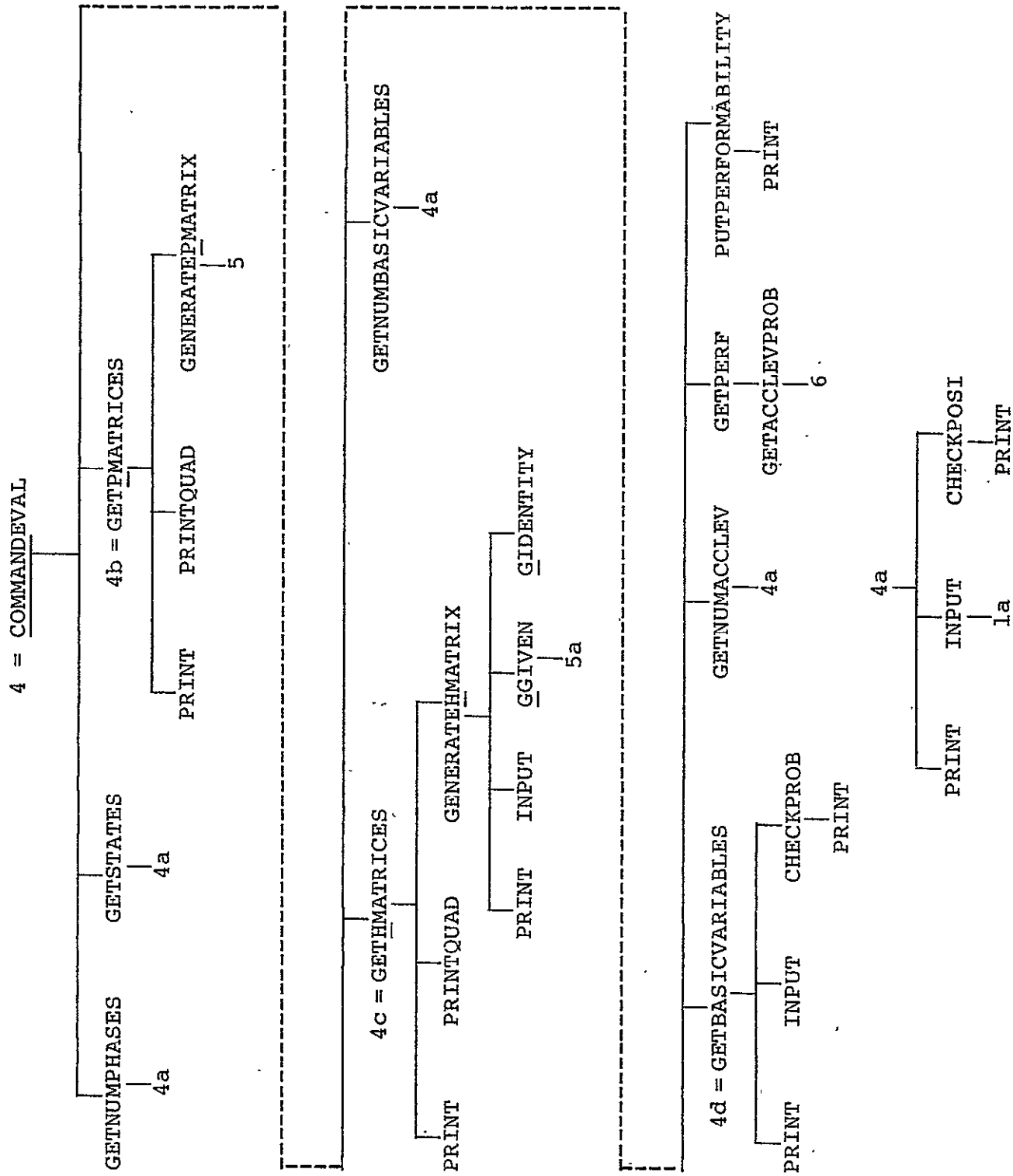


Figure 12. Interdependence of METAPHOR functions, continued

6. Interdependence of METAPHOR Functions

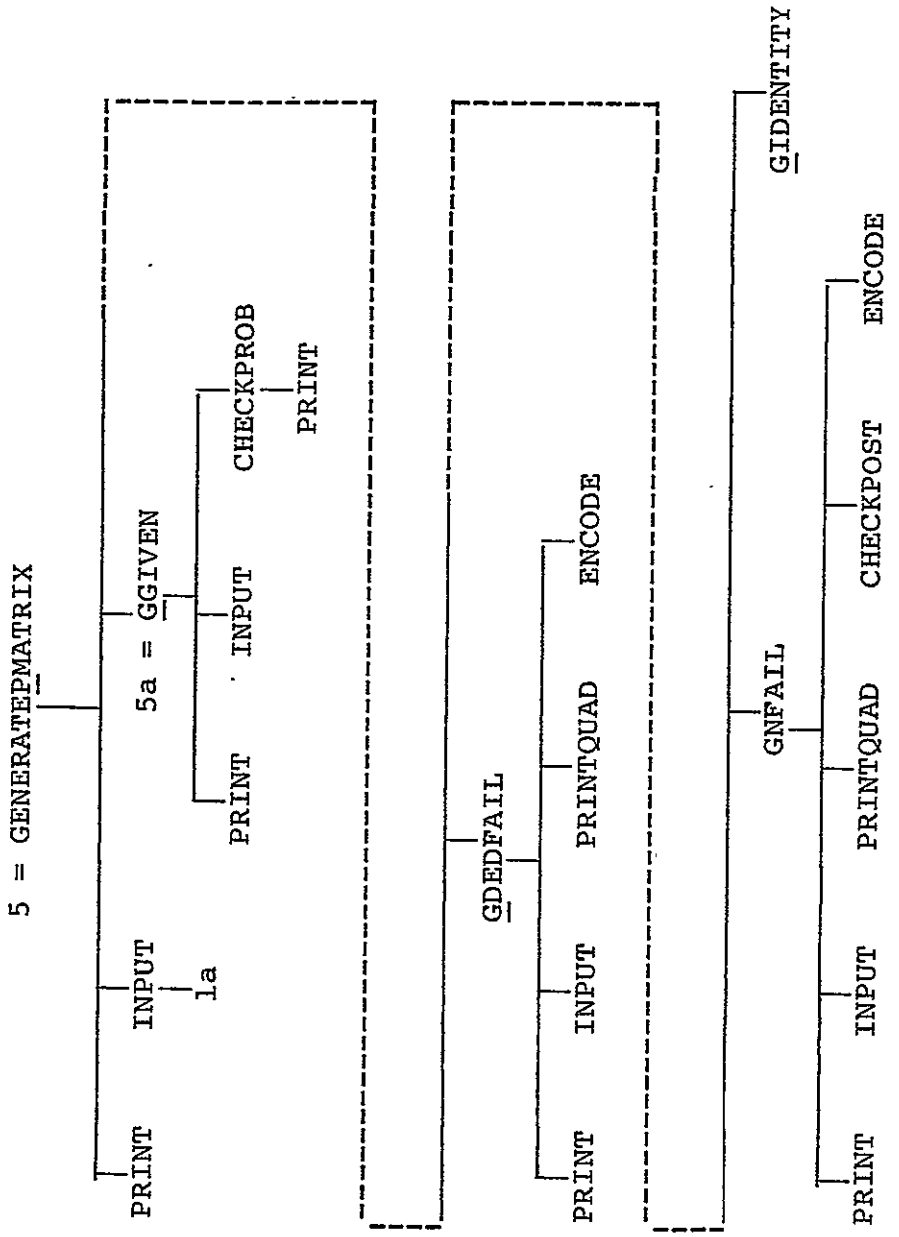


Figure 12. Interdependence of METAPHOR functions, continued

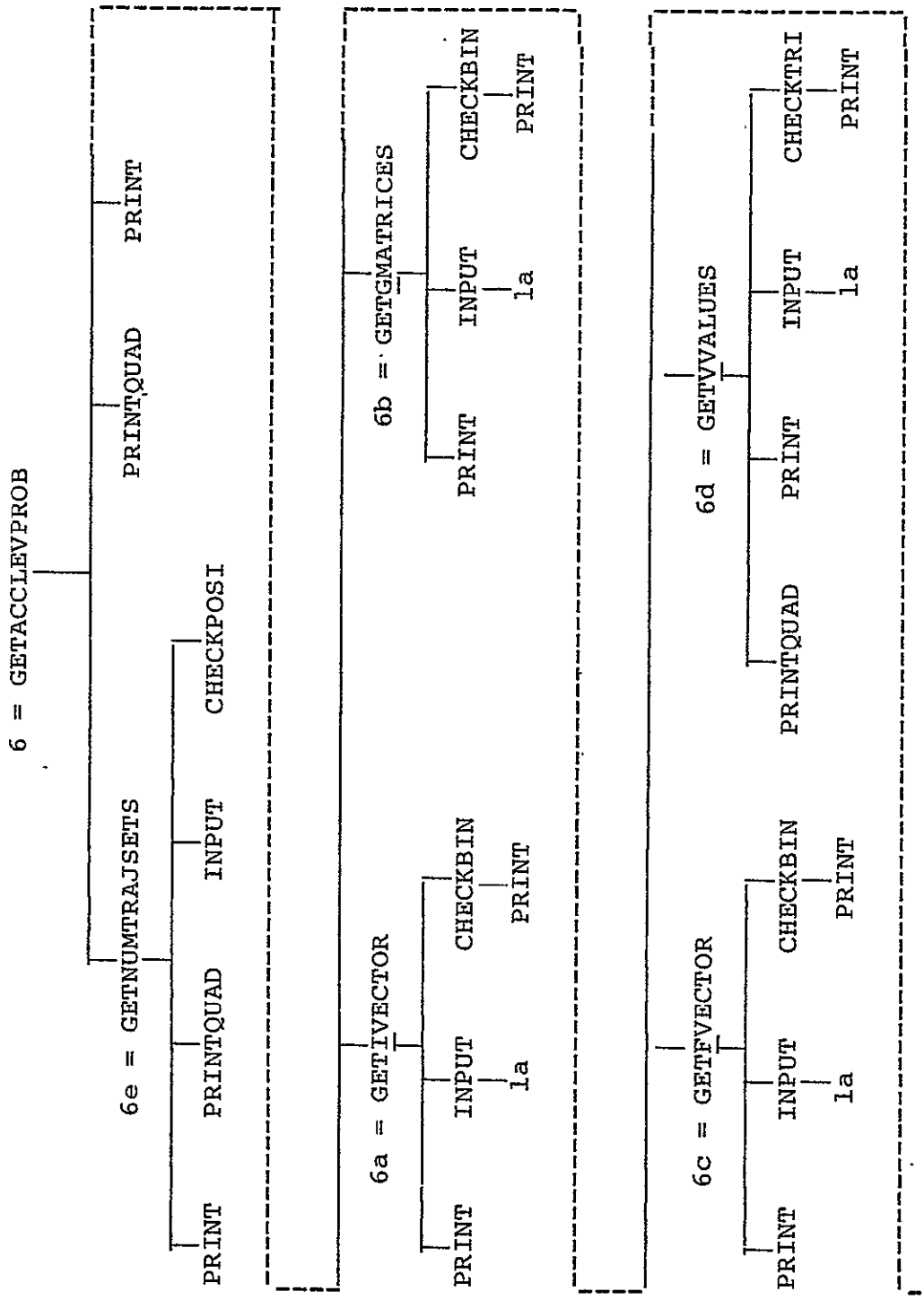


Figure 12. Interdependence of METAPHOR functions, continued

6. Interdependence of METAPHOR Functions

7. METAPHOR Function Descriptions

This section gives an overview of each METAPHOR function, giving its calling sequence, purpose, global variables used, functions calling it, functions it calls, general comments, and listing.

BRIEF

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← BRIEF arg

PURPOSE: To allow the input of the BRIEF user command.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: None.

FUNCTIONS CALLED: None.

COMMENTS: Called when the user types 'BRIEF arg' in response to a (non-yes/no) question. The valid arguments are 'ON' and 'OFF.' BRIEF ON suppresses most of the output from METAPHOR, while BRIEF OFF allows the normal full output. The default is BRIEF OFF. BRIEF returns a twelve character representation of the BRIEF command (for INPUT) and the value of arg in result. For example, after the command BRIEF ON, the vector 'BRIEF ON ' is returned in result.

LISTING:

```
     ∇ Z←BRIEF SWITCH
[1]   ⍝
[2]   ⍝  ROUTINE TO INPUT THE BRIEF COMMAND
[3]   ⍝
[4]   ⍝ RETURN THE VALUE
[5]   Z←'BRIEF ',SWITCH
     ∇
```

CALCTRAJPROB

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← CALCTRAJPROB

PURPOSE: To calculate the probability of a given trajectory set.

GLOBAL VARIABLES: BASICVARIABLES, H, NUMBASICVARIABLES, P.

CALLING FUNCTIONS: GETACCLEVPROB.

FUNCTIONS CALLED: None.

COMMENTS: Uses the values of F, G, I, and V found in GETFVECTOR, GETGMATRICES, GETIVECTOR, and GETVVALUES as well as the global variables to calculate the probability of a trajectory set. The formula used to perform the calculations is:

Probability = I * P(1) * G(1) * H(1) * P(2) * G(2) * H(2) * ... * H(n) * F * V * BASICVARIABLES

where there are n phases, and * denotes matrix multiplication. The probability is returned in result.

LISTING:

```
     ∇ Z←CALCTRAJPROB;J;PHASE
[1]    ⍝
[2]    ⍝   CALCULATES THE PROBABILITY OF THE GIVEN TRAJECTORY
[3]    ⍝
[4]    ⍝ COMPUTE THE INITIAL PHASE PROBABILITY
[5]    Z←I+.×P[1;;]
[6]    ⍝
[7]    ⍝  LOOP THROUGH THE OTHER PHASES
[8]    ⍝ INITIALIZE PHASE COUNTER
[9]    PHASE←2
[10]   ⍝ FIRST CONSIDER THE SUCCESS STATES MATRIX
[11] CTPLOOP:Z←Z×G[PHASE-1;]
[12]   ⍝ THEN THE INTERPHASE MATRIX
[13]   Z←Z+.×H[PHASE-1;;]
[14]   ⍝ THEN THE TRANSITION MATRIX
[15]   Z←Z+.×P[PHASE;;]
```

```
[16]  ⍝
[17]  ⍝ INCREMENT PHASE COUNTER AND BRANCH IF APPROPRIATE
[18]  PHASE←PHASE+1
[19]  →(PHASE≤NUMPHASES)/CTPLOOP
[20]  ⍝
[21]  ⍝ POST-MULTIPLY BY FINAL STATE
[22]  Z←Z+.×F
[23]  ⍝ FINALLY, MULTIPLY BY THE TIME+INVARIANT BASIC VARIABLE
      PROBABILITIES
[24]  ⍝ IF NO TIME+INVARIANT BASIC VARIABLES, EXIT
[25]  →(NUMBASICVARIABLES=0)/0
[26]  Z←Z×+/V×BASICVARIABLES
      ∇
```

CHECKBIN

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← CHECKBIN arg

PURPOSE: To check that arg has only binary (0, 1) values.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GETFVECTOR, GETGMATRICES, GETIVECTOR.

FUNCTIONS CALLED: PRINT.

COMMENTS: Checks the input argument arg for 0's and 1's. The argument can be either a scalar or a vector. If each entry is binary, CHECKBIN returns a 0, else a warning is printed and a 1 is returned in result.

LISTING:

```
     ∇ Z←CHECKBIN CHECKNO
[1]    ⍝
[2]    ⍝  RETURN 0 IF CHECKNO CONTAINS ONLY BINARY ZEROS AND ONES
[3]    ⍝   ELSE PRINT MESSAGE AND RETURN 1
[4]    ⍝
[5]    ⍝  CHECK FOR PROPER BINARY ELEMENTS
[6]    Z←~(∧/CHECKNO∈ 0 1)
[7]    →(~Z)/0
[8]    PRINT 'EACH ENTRY MUST BE EITHER 0 OR 1'
[9]    →0
     ∇
```

CHECKPOSI

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← CHECKPOSI arg

PURPOSE: To check that arg has only positive integer values.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GETNUMACCLEV, GETNUMBASICVARIABLES, GETNUMPHASES, GETNUMTRAJSETS, GETSTATES.

FUNCTIONS CALLED: PRINT.

COMMENTS: Checks the input argument arg for positive integers. The argument can be either a scalar or a vector. If each entry is a positive integer, CHECKPOSI returns a 0, else a warning is printed and a 1 is returned in result.

LISTING:

```
     ∇ Z←CHECKPOSI CHECKNO
[1]   ⍝
[2]   ⍝   RETURN 0 IF CHECKNO IS A POSITIVE INTEGER, ELSE PRINT
      MESSAGE AND RETURN 1
[3]   ⍝
[4]   ⍝  CHECK FOR POSITIVENESS
[5]   Z←~∧/(CHECKNO>0)
[6]   →(~Z)/CHECKINT
[7]   PRINT 'INPUT NOT POSITIVE'
[8]   →0
[9]   ⍝
[10]  ⍝ CHECK FOR INTEGER
[11] CHECKINT:Z←~∧/CHECKNO=⌈CHECKNO
[12]  →(~Z)/0
[13]  PRINT 'INPUT NOT AN INTEGER'
     ∇
```

CHECKPROB

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← CHECKPROB arg

PURPOSE: To check that arg has only valid probability (between 0 and 1) values.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GETBASICVARIABLES, GGIVEN.

FUNCTIONS CALLED: PRINT.

COMMENTS: Checks the input argument arg for numbers between 0 and 1 inclusive. The argument can be either a scalar or a vector. If each entry is a valid probability, CHECKPROB returns a 0, else a warning is printed and a 1 is returned in result.

LISTING:

```
     ∇ Z←CHECKPROB CHECKNO
[1]   ⍝
[2]   ⍝   RETURN 0 IF CHECKNO IS BETWEEN 0 AND 1 INCLUSIVE, ELSE
      PRINT MESSAGE AND RETURN 1
[3]   ⍝
[4]   ⍝  CHECK FOR PROPER RANGE
[5]   Z←~(∧/CHECKNO≥0)∧(∧/CHECKNO≤1)
[6]   →(~Z)/0
[7]   PRINT 'INPUT NOT BETWEEN 0 AND 1'
     ∇
```

CHECKTRI

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← CHECKTRI arg

PURPOSE: To check that arg has only trinary (0, 1, 2) values.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GETVVALUES.

FUNCTIONS CALLED: PRINT.

COMMENTS: Checks the input argument arg for 0's, 1's, and 2's. The argument can be either a scalar or a vector. If each entry is binary, CHECKTRI returns a 0, else a warning is printed and a 1 is returned in result.

LISTING:

```
     ∇ Z←CHECKTRI CHECKNO
[1]   ⍝
[2]   ⍝ RETURN 0 IF CHECKNO CONTAINS ONLY TRINARY ZEROS, ONES, AND
      TWOS
[3]   ⍝   ELSE PRINT MESSAGE AND RETURN 1
[4]   ⍝
[5]   ⍝ CHECK FOR PROPER TRINARY ELEMENTS
[6]   Z←~(∧/CHECKNO∈ 0 1 2)
[7]   →(~Z)/0
[8]   PRINT 'EACH ENTRY MUST BE EITHER 0,1, OR 2'
[9]   →0
     ∇
```

DECLAREMETAPHOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: DECLAREMETAPHOR

PURPOSE: To initialize the global variables and constants employed in the METAPHOR package.

GLOBAL VARIABLES: ALTER, BRIEFOUTPUT, CALC, COM, COMMANDLIST, COMMANDSIZE, DATA, DEDFAIL, DEFACCLEVEL, DEFBASICVARIABLES, DEFF, DEFG, DEFH, DEFI, DEFNUMACCLEV, DEFNUMBASICVARIABLES, DEFNUMPHASES, DEFNUMSTATES, DEFNUMTRAJSETS, DEFP, DEFPERFORMABILITY, DEFV, ECHOINPUT, EVAL, EXIT, GIVEN, HELP, IDENTITY, NFAIL, OFF, ON, GAV, GBV, GD, GDV, GFV, GGM, GG, GHM, GIV, GN, GNA, GNBV, GNP, GNTS, GPM, GS, GVV, HMATRIXDUMMYLIST, HMATRIXLIST, MET, PMATRIXLIST.

FUNCTIONS CALLED: None.

COMMENTS: Initializes the control constants used by METAPHOR to determine its own state. These consist of the following:

1. A series of variables containing the command words METAPHOR recognizes. The variables are ALTER, CALC, COM, DATA, EVAL, EXIT, and HELP. Also declared are two switch settings, ON and OFF.

2. A matrix used to determine which command has been input. It can recognize the commands above, as well as ECHO and BRIEF. The matrix is COMMANDLIST . In addition, the size of the commands is set in COMMANDSIZE.

3. A series of variables containing the types of matrices METAPHOR will generate. Specifically, these are DEDFAIL, GIVEN, IDENTITY, and NFAIL.

4. Two matrices used to determine which

matrix type has been specified. These matrices are HMATRIXLIST and PMATRIXLIST.

5. A series of variables used by METAPHOR to specify the function it is presently executing. These are used to select the proper HELP routine to print in response to a HELP request. These are all underlined and generally begin with a G. For example, GAV and GBV. Note, however, that MET is also such an indicator.

6. The number of HELP routines. This is placed in NUMHELPROUTINES.

7. A series of variables used as flags to determine if a given value has been defined. These are initially set to 'not defined.' these are used in conjunction with the ALTER and DATA commands to determine if a given value can be changed or displayed. All of these flags begin with 'DEF,' for instance, DEFACCLEVEL and DEFBASICVARIABLES.

8. Two switches used to control output and input echoing. These are initialized to off, thus allowing normal output and no input echoing. These switches are BRIEFOUTPUT and ECHOINPUT.

LISTING:

```
     ∇ DECLAREMETAPHOR
[1]    ⍝
[2]    ⍝   DECLARES AND INITIALIZES THE CONSTANTS USED IN THE 'PERF'
       FUNCTION SERIES
[3]    ⍝
[4]    ⍝  SET THE LIST OF COMMANDS
[5]    HELP←'HELP  '
[6]    EXIT←'EXIT  '
[7]    DATA←'DATA  '
[8]    ALTER←'ALTER '
[9]    CALC←'CALC  '
[10]   COM←'COM   '
[11]   EVAL←'EVAL  '
[12]   ON←'ON    '
[13]   OFF←'OFF   '
[14]   COMMANDSIZE←6
[15]   COMMANDLIST←(9,COMMANDSIZE)⍴'HELP  ','EXIT  ','DATA  ','ALTER
       ','CALC  ','ECHO  ','BRIEF ','COM   ','EVAL  '
[16]   ⍝
[17]   ⍝
```

```
[18]  ⍝ SET LIST OF MATRIX GENERATOR TYPES
[19]  GIVEN←'GIVEN    '
[20]  DEDFAIL←'DEDFAIL '
[21]  NFAIL←'NFAIL    '
[22]  IDENTITY←'IDENTITY'
[23]  PMATRIXLIST←(4 8)⍴'GIVEN    ','DEDFAIL ','NFAIL
      ','IDENTITY'
[24]  HMATRIXLIST←(2 8)⍴'GIVEN    ','IDENTITY'
[25]  ⍝
[26]  ⍝
[27]  ⍝
[28]  ⍝ SET LIST OF ROUTINES FOR HELP CALLS
[29]  GNP←1
[30]  GS←2
[31]  GPM←3
[32]  GG←4
[33]  GN←5
[34]  GD←6
[35]  GHM←7
[36]  GNBV←8
[37]  GBV←9
[38]  GNA←10
[39]  GIV←11
[40]  GGM←12
[41]  GFV←13
[42]  GVV←14
[43]  GNTS←15
[44]  GAV←16
[45]  GDV←17
[46]  MET←18
[47]  ⍝
[48]  ⍝ SET NUMBER OF HELP ROUTINES
[49]  NUMHELPROUTINES←18
[50]  ⍝
[51]  ⍝ SET VARIABLE DEFINITION SWITCHES.  1 IF VARIABLE DEFINED, 0
      IF NOT.
[52]  DEFNUMPHASES←0
[53]  DEFNUMSTATES←0
[54]  DEFP←0
[55]  DEFH←0
[56]  DEFNUMBASICVARIABLES←0
[57]  DEFBASICVARIABLES←0
[58]  DEFNUMACCLEV←0
[59]  DEFNUMTRAJSETS←0
[60]  DEFI←0
[61]  DEFG←0
[62]  DEFF←0
[63]  DEFV←0
[64]  DEFPERFORMABILITY←0
[65]  DEFACCLEVEL←0
[66]  ⍝
[67]  ⍝ MAKE DEFAULT SWITCH SETTINGS
[68]  BRIEFOUTPUT←0
```

```
[69]  ECHOINPUT←0
      ∇
```

7. *DECLAREMETAPHOR METAPHOR FUNCTION DESCRIPTION*

ECHO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← ECHO arg

PURPOSE: To allow the input of the ECHO user command.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: None.

FUNCTIONS CALLED: None.

COMMENTS: Called when the user types 'ECHO arg' in response to a (non-yes/no) question. The valid arguments are 'ON' and 'OFF.' ECHO ON causes the input to be repeated back to the user, while ECHO OFF suppresses the repetition. ECHO returns a twelve character representation of the ECHO command (for INPUT) and the value of arg in result. For example, after the command ECHO OFF, the vector 'ECHO OFF ' is returned in result. The default is ECHO OFF.

LISTING:

```
     ∇ Z←ECHO SWITCH
[1]   ⍝
[2]   ⍝  ROUTINE TO INPUT THE ECHO COMMAND
[3]   ⍝
[4]   ⍝ RETURN THE VALUE
[5]   Z←'ECHO  ',SWITCH
     ∇
```

ENCODE

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← arg1 ENCODE arg2

PURPOSE: To implement the encode (T) operator for vectors as the right hand argument (arg2). This function is necessary because the resident APL program available at The University of Michigan will not support such an encode operation. The encoded vector is returned in result.

GLOBAL VARIABLES: None.

FUNCTIONS CALLED: None.

COMMENTS: Given a vector as arg2, ENCODE loops through each value n of the vector, determining (arg1 T n), i.e., the representation of n in base arg1. An initial test is made to determine if arg2 contains a single value, and if so, no looping is done.

LISTING:

```
     ∇ Z←M ENCODE N;ROW
[1]    ⍝
[2]    ⍝
[3]    ⍝  THE ENCODE FUNCTION EMPLOYED ON MTS WILL NOT ACCEPT
       VECTORS AS
[4]    ⍝  ARGUMENTS ON THE RIGHT HAND SIDE.  THIS FUNCTION SIMULATES
       THAT CAPABILITY
[5]    ⍝  THE COMPARABLE APL NOTATION WOULD BE:    M⊤N
[6]    ⍝
[7]    ⍝  TEST FOR A SINGLE INPUT
[8]    →(0=⍴⍴N)/ESINGLE
[9]    ⍝  INITIALIZE THE ARRAY AND LOOP COUNTER
[10]   Z←((⌈/1,⍴M),(⍴N))⍴0
[11]   COL←1
[12]  ELOOP:Z[;COL]←M⊤N[COL]
[13]   COL←COL+1
[14]   →(COL≤⍴N)/ELOOP
[15]   ⍝ EXIT
[16]   →0
[17]   ⍝
[18]   ⍝  IF ONLY ONE ARGUMENT TO BE DECODED
```

```
[19] ESINGLE:Z←M⊤N
     ∇
```

GENERATEHMATRIX

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← arg1 GENERATEHMATRIX arg2

PURPOSE: To supervise the construction of the H matrix for the given phase.

GLOBAL VARIABLES: IN, GHM, HMATRIXLIST.

CALLING FUNCTIONS: GETHMATRICES.

FUNCTIONS CALLED: INPUT, PRINT, GGIVEN, GIDENTITY.

COMMENTS: Asks the user what type of H matrix is desired for the particular phase being considered. Presently, METAPHOR supports two types of H matrix input. For the first of these, the user inputs 'GIVEN' and enters the values of the matrix himself. GENERATEHMATRIX calls GGIVEN to accomplish this. The second choice is to choose an identity H matrix via an 'IDENTITY' input. If the two phases do not have the same number of states, an error message is printed and the user is again prompted for the H matrix type; else GENERATEHMATRIX calls GIDENTITY to generate an identity matrix. If an illegal (i.e., unsupported) type of matrix is specified, an error message is generated and another matrix type is requested.

The resulting H matrix is required to have shape arg1xarg2. Arg1 and arg2 are typically the number of states in the phases associated with the H matrix. The H matrix obtained is returned in result.

LISTING:

```
     ∇ Z←M GENERATEHMATRIX N;TYPE;IN
[1]   ⍝
[2]   ⍝
[3]   ⍝   ROUTINE FOR SUPERVISING THE CONSTRUCTION OF THE H MATRIX
      FOR THE GIVEN PHASE
[4]   ⍝    THE MATRIX IS TO HAVE SHAPE M×N
[5]   ⍝
[6]   ⍝  GET TYPE OF MATRIX .
[7]   PRINT ''
[8]  GENHMIN:PRINT 'WHAT TYPE OF H MATRIX?'
[9]   8 INPUT GHM
[10]  ⍝ CHECK FOR COMMAND
[11]  →(1=∧ ∈'COMMAND')/GENHMIN
[12]  ⍝  CHECK FOR TYPE
[13]  →(∧/HMATRIXLIST∈(IN))/GENHMGIVEN,GENHMIDENTITY
[14]  ⍝  ELSE ILLEGAL TYPE
[15]  PRINT 'ILLEGAL H MATRIX TYPE.  TYPE HELP FOR INFORMATION'
[16]  →GENHMIN
[17]  ⍝
[18]  ⍝
[19]  ⍝  USER WILL GIVE H MATRIX VALUES
[20] GENHMGIVEN:Z←M GGIVEN N
[21]  →0
[22]  ⍝
[23]  ⍝
[24]  ⍝  IDENTITY MATRIX GENERATOR
[25]  ⍝  FIRST CHECK TO MAKE SURE AN IDENTITY MATRIX IS  APPROPRIATE
      HERE
[26] GENHMIDENTITY:→(M=N)/GENHMIDENTITYGET
[27]  PRINT  'THESE  TWO  PHASES  DO  NOT  HAVE  THE  SAME NUMBER OF
      STATES.'
[28]  PRINT 'AN IDENTITY MATRIX IS INAPPROPRIATE FOR THE  INTERPHASE
      TRANSITION MATRIX.'
[29]  →GENHMIN
[30] GENHMIDENTITYGET:Z←GIDENTITY N
[31]  →0
     ∇
```

GENERATEPMATRIX

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GENERATEPMATRIX arg

PURPOSE: To supervise the construction of the P matrix for the given phase.

GLOBAL VARIABLES: IN, GPM, PMATRIXLIST.

CALLING FUNCTIONS: GETPMATRICES.

FUNCTIONS CALLED: INPUT, PRINT, GDEDFAIL, GGIVEN, GIDENTITY, GNFAIL.

COMMENTS: Asks the user what type of P matrix is desired for the particular phase being considered. Presently, METAPHOR supports four types of P matrix input. For the first of these, the user inputs 'GIVEN' and enters the values of the matrix himself. GENERATEPMATRIX calls GGIVEN to accomplish this. The second choice is to choose an identity P matrix via an 'IDENTITY' input. If the two phases do not have the same number of states, an error message is printed and the user is again prompted for the P matrix type; else GENERATEPMATRIX calls GIDENTITY to generate an identity matrix. Also, the user can input 'DEDFAIL' or 'NFAIL' and METAPHOR will construct a P matrix based on certain assumptions concerning the structure of the system being modeled. The functions called to perform the generation are GDEDFAIL and GNFAIL. If an illegal (i.e., unsupported) type of matrix is specified, an error message is generated and another matrix type is requested.

The resulting P matrix is required to have shape argxarg. Arg is typically the number of states in the phase associated with the P matrix.

The P matrix obtained is returned in

result.

LISTING:

```
     ∇ Z←GENERATEPMATRIX N;TYPE;IN
[1]   ⍝
[2]   ⍝
[3]   ⍝   ROUTINE FOR SUPERVISING THE CONSTRUCTION OF THE P MATRIX
      FOR THE GIVEN PHASE
[4]   ⍝    THE MATRIX IS TO HAVE SHAPE N×N
[5]   ⍝
[6]   ⍝  GET TYPE OF MATRIX
[7]   PRINT ''
[8]  GENPMIN:PRINT 'WHAT TYPE OF P MATRIX?'
[9]   8 INPUT GPM
[10]  ⍝ CHECK FOR COMMAND
[11]  →(1=∧ ∈'COMMAND')/GENPMIN
[12]  ⍝  CHECK FOR TYPE
[13]  →(∧/PMATRIXLIST∈(
      IN))/GENPMGIVEN,GENPMDEDFAIL,GENPMNFAIL,GENPMIDENTITY
[14]  ⍝  ELSE ILLEGAL TYPE
[15]  PRINT 'ILLEGAL P MATRIX TYPE.  TYPE HELP FOR INFORMATION'
[16]  →GENPMIN
[17]  ⍝
[18]  ⍝
[19]  ⍝  USER WILL GIVE P MATRIX VALUES
[20] GENPMGIVEN:Z←N GGIVEN N
[21]  →0
[22]  ⍝
[23]  ⍝
[24]  ⍝  DEDICATED COMPONENT SYSTEM
[25] GENPMDEDFAIL:Z←GDEDFAIL N
[26]  ⍝  IF  ROUTINE  FAILED,  GO  BACK  TO  TRY  AGAIN.  ELSE  EXIT
      ROUTINE
[27]  →((Z[1;1]<0),1)/GENPMIN,0
[28]  ⍝
[29]  ⍝
[30]  ⍝   N GROUPS OF COMPONENTS SYSTEM
[31] GENPMNFAIL:Z←GNFAIL N
[32]  →0
[33]  ⍝
[34]  ⍝
[35]  ⍝  IDENTITY MATRIX GENERATOR
[36] GENPMIDENTITY:Z←GIDENTITY N
[37]  →0
     ∇
```

GETACCLEVPROB

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GETACCLEVPROB arg

PURPOSE: To determine the probability of arg, where arg is an accomplishment level.

GLOBAL VARIABLES: DEFF, DEFG, DEFI, DEFNUMTRAJSETS, DEFV, NUMTRAJSETS.

CALLING FUNCTIONS: GETPERFORMABILITY.
|INDEX(;GETPERFORMABILITY, METAPHOR function')

FUNCTIONS CALLED: CALCTRAJPROB, GETNUMTRAJSETS GETFVECTOR, GETGMATRICES, GETIVECTOR, GETVVALUES, PRINT, PRINTQUAD.

COMMENTS: GETACCLEVPROB controls the inputting of the information required to obtain the probability of the hierarchy level given in arg. In particular, GETACCLEVPROB requests the number of trajectory sets in the accomplishment level, and then for each set, fetches the initial vector, the characteristic matrices the characteristic vector, and the characterization of the time-invariant basic variables. The probabilities of each trajectory set is then calculated and added to the probability of the accomplishment set. Finally, the definition flags associated with the trajectory set information are reset before the function is exited; this insures that the ALTER and DISPLAY commands do not try to access these variables outside the GETACCLEVPROB function.

The performability is returned in result.

LISTING:

```
     ∇ Z←GETACCLEVPROB LEVEL;T;TRAJPROB;NUMTRAJSETS;I;G;F;V
[1]    ⍝
[2]    ⍝   ROUTINE FOR DETERMINING THE PROBABILITY OF AN
       ACCOMPLISHMENT LEVEL
[3]    ⍝
[4]    ⍝ INITIALIZE THE COLLECTION VARIABLE
[5]    Z←0
[6]    ⍝
[7]    ⍝  GET THE NUMBER OF TRAJECTORY SETS
[8]    NUMTRAJSETS←GETNUMTRAJSETS LEVEL
[9]    ⍝  LOOP THROUGH FOR EACH SET, GETTING ITS VALUES AND
       CALCULATING ITS PROBABILITY
[10]   ⍝  INITIALIZE COUNTER
[11]   T←1
[12] GAPTLOOP:⍞←((PRINTQUAD 'TRAJECTORY SET ');PRINTQUAD T)
[13]   PRINT ''
[14]   I←GETIVECTOR
[15]   G←GETGMATRICES
[16]   F←GETFVECTOR
[17]   V←GETVVALUES
[18]   TRAJPROB←CALCTRAJPROB
[19]   ⍝
[20]   ⍝  ADD THE PROBABILITY TO THE COLLECTION VARIABLE
[21]   Z←Z+TRAJPROB
[22]   ⍝
[23]   ⍝  INCREMENT TRAJECTORY COUNTER AND BRANCH IF NECESSARY
[24]   T←T+1
[25]   →(T≤NUMTRAJSETS)/GAPTLOOP
[26]   ⍝ RESET DEFINITION FLAGS
[27]   DEFNUMTRAJSETS←DEFI←DEFG←DEFF←DEFV←0
     ∇
```

GETALTERVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GETALTERVECTOR

PURPOSE: To determine which data the user wishes to change.

GLOBAL VARIABLES: GAV.

CALLING FUNCTIONS: COMMANDALTER.

FUNCTIONS CALLED: PRINT, COMMANDHELP.

COMMENTS: The user is presented a line containing abbreviations of the variables which can be changed. The user then prints an X below each variable he wishes to alter. GETALTERVECTOR returns a vector in result with 1's corresponding to those variables to be altered, and 0's corresponding to those not to be altered. The position of the 1's and 0's has the following code:

Index 1--Alter the P matrices.

Index 2--Alter the H matrices.

Index 3--Alter the time-invariant basic variables.

Index 4--Alter the values of all accomplishment levels.

Index 5--Alter the values of just the present accomplishment level.

Index 6--Alter the initial vector.

Index 7--Alter the characteristic matrices.

Index 8--Alter the characteristic vector.

Index 9--Alter the characterization of the time-invariant basic variables.

7. GETALTERVECTOR METAPHOR Function Description

Index 10--Alter the number of trajectory sets.

LISTING:

```
     ∇ Z←GETALTERVECTOR;GAVINPUT
[1]    ⍝
[2]    ⍝   ROUTINE FOR GETTING THE ALTER VECTOR FOR CHANGING DATA
[3]    ⍝
[4]    ⍝  PRINT EXPLANATION
[5]   GAVIN:PRINT 'PUT AN X BELOW EACH ITEM TO BE CHANGED.  HELP
       AVAILABLE.'
[6]    ⍝  GET ALTER REQUESTS, RESHAPING ALONG THE WAY
[7]    PRINT 'P  H  CONST.BAS.VARS  ALL.ACC.LEVELS
       PRESENT.ACC.LEVEL  I  G  F V NUM.TRAJ.SETS'
[8]    GAVINPUT←80⍴(⍞,(80⍴' '))
[9]    ⍝ LOOK FOR HELP REQUEST.  IF PRESENT, CALL FOR HELP ROUTINE.
[10]   →(~∧/'HELP'∈GAVINPUT)/GAVVECTORSET
[11]   COMMANDHELP GAV
[12]   →GAVIN
[13]   ⍝  INITIALIZE ALTER VECTOR
[14]  GAVVECTORSET:Z←10⍴0
[15]   ⍝  DETERMINE CHANGE VECTOR
[16]   Z[1]←'X'∈GAVINPUT[1]
[17]   Z[2]←'X'∈GAVINPUT[4]
[19]   Z[4]←'X'∈GAVINPUT[22+⍳14]
[20]   Z[5]←'X'∈GAVINPUT[38+⍳17]
[21]   Z[6]←'X'∈GAVINPUT[58]
[22]   Z[7]←'X'∈GAVINPUT[61]
[23]   Z[8]←'X'∈GAVINPUT[64]
[24]   Z[9]←'X'∈GAVINPUT[66]
[25]   Z[10]←'X'∈GAVINPUT[67+⍳13]
     ∇
```

GETBASICVARIABLES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETBASICVARIABLES

PURPOSE: To obtain the probabilities of the time-invariant basic variables.

GLOBAL VARIABLES: BASICVARIABLES, DEFBASICVARIABLES, IN, NUMBASICVARIABLES, GBV.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: CHECKPROB, INPUT, PRINT.

COMMENTS: Asks the user for the probabilities of the time-invariant basic variables, inputs the probabilities, checks them, and places then in BASICVARIABLES. If these values are not proper probabilities (i.e., numbers between zero and one), the query is repeated. In addition, the definition flag associated with BASICVARIABLES is set so that the probabilities of the time-invariant basic variables can be displayed or changed.

LISTING:

```
     ∇ GETBASICVARIABLES
[1]   ⍝
[2]   ⍝  ROUTINE FOR FETCHING THE PROBABILITIES OF EACH OF THE
      TIME-INVARIANT BASIC VARIABLES
[3]   ⍝
[4]   ⍝  SEE IF IT IS NECESSARY TO INPUT BASIC VARIABLES
[5]   →(NUMBASICVARIABLES=0)/0
[6]   ⍝
[7]   ⍝INPUT PROBABILITIES AND CHECK VALIDITY
[8]   PRINT ''
[9]  GBVIN:PRINT 'PROBABILITIES OF EACH TIME-INVARIANT BASIC
      VARIABLE?  (SPACE BETWEEN EACH NUMBER)'
[10]  NUMBASICVARIABLES INPUT GBV
[11]  ⍝ CHECK FOR COMMAND
[12]  →(1=∧ ∈'COMMAND')/GBVIN
[13]  ⍝ CHECK FOR PROPER PROBABILITY MAGNITUDE
[14]  →(CHECKPROB IN)/GBVIN
[15]  ⍝ SET BASIC VARIABLE PROBABILITIES AND EXIT
```

```
[16]  BASICVARIABLES←IN,1-IN
[17]  DEFBASICVARIABLES←1
      ∇
```

GETDATAVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result <- GETDATAVECTOR

PURPOSE: To determine which data the user wishes to display.

GLOBAL VARIABLES: GDV.

CALLING FUNCTIONS: COMMANDDATA.

FUNCTIONS CALLED: PRINT, COMMANDHELP.

COMMENTS: The user is presented two lines containing abbreviations of the variables which can be displayed. The user then prints an X below each variable he wishes to have shown. GETDATAVECTOR returns in result a vector with 1's corresponding to those variables to be displayed, and 0's corresponding to those not to be displayed. The position of the 1's and 0's has the following code:

Index 1--Display the number of phases.

Index 2--Display the number of states in each phase.

Index 3--Display the P matrices.

Index 4--Display the H matrices.

Index 5--Display the number of time-invariant basic variables.

Index 6--Display the probabilities of each of the time-invariant basic variables.

Index 7--Display the number of accomplishment levels.

Index 8--Display the number of trajectory sets.

7. GETDATAVECTOR METAPHOR Function Description

Index 9--Display the initial vector.

Index 10--Display the characteristic matrices.

Index 11--Display the characteristic vector.

Index 12--Display the characterization of the time-invariant basic variables.

Index 13--Display the performability.

LISTING:

```
     ∇ Z←GETDATAVECTOR;GDVINPUT
[1]   ⍝
[2]   ⍝   ROUTINE FOR GETTING THE DATA VECTOR FOR DISPLAYING DATA
[3]   ⍝
[4]   ⍝  PRINT EXPLANATION
[5]  GDVIN:PRINT 'PUT AN X BELOW EACH ITEM TO BE DISPLAYED.  HELP
      AVAILABLE.'
[6]   ⍝  GET DATA REQUESTS, RESHAPING ALONG THE WAY
[7]   PRINT 'NUM.PHASES  NUM.STATES  P  H  NUM.CONST.BAS.VARS
      PROB.CONST.BAS.VARS'
[8]   GDVINPUT←69⍴(⍞,(69⍴' '))
[9]   ⍝LOOK FOR HELP REQUEST.  IF PRESENT, CALL FOR HELP ROUTINE
[10]  →(~∧/'HELP'∈GDVINPUT)/GDVINCONT
[11]  COMMANDHELP GDV
[12]  →GDVIN
[13] GDVINCONT:PRINT
      'NUM.ACC.LEVELS  NUM.TRAJ.SETS  I  G  F  V  PERF'
[14]  GDVINPUT←116⍴(GDVINPUT,⍞,(47⍴' '))
[15]  PRINT ''
[16]  ⍝LOOK FOR HELP REQUEST.  IF PRESENT, CALL FOR HELP ROUTINE
[17]  →(~∧/'HELP'∈GDVINPUT)/GDVVECTORSET
[18]  COMMANDHELP GDV
[19]  →GDVIN
[20]  ⍝  INITIALIZE DISPLAY VECTOR
[21] GDVVECTORSET:Z←13⍴0
[22]  ⍝  DETERMINE DISPLAY VECTOR
[23]  Z[1]←'X'∈GDVINPUT[⍳10]
[24]  Z[2]←'X'∈GDVINPUT[12+⍳10]
[25]  Z[3]←'X'∈GDVINPUT[25]
[26]  Z[4]←'X'∈GDVINPUT[28]
[27]  Z[5]←'X'∈GDVINPUT[30+⍳18]
[28]  Z[6]←'X'∈GDVINPUT[50+⍳19]
[29]  Z[7]←'X'∈GDVINPUT[69+⍳14]
[30]  Z[8]←'X'∈GDVINPUT[85+⍳13]
[31]  Z[9]←'X'∈GDVINPUT[101]
[32]  Z[10]←'X'∈GDVINPUT[104]
[33]  Z[11]←'X'∈GDVINPUT[107]
[34]  Z[12]←'X'∈GDVINPUT[110]
```

```
[35]  Z[13]←'X'∈GDVINPUT[112+⍳4]
      ∇
```

7. GETDATAVECTOR METAPHOR FUNCTION DESCRIPTION

GETNUMACCLEV

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETNUMACCLEV

PURPOSE: To obtain the number of accomplishment levels in the model.

GLOBAL VARIABLES: DEFNUMACCLEV, IN, NUMACCLEV, GNA.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: CHECKPOSI, INPUT, PRINT.

COMMENTS: Asks the user for the number of accomplishment levels, inputs the number, checks it, and places it in NUMACCLEV. If the value is not a positive integer, the query is repeated. In addition, the definition flag associated with NUMACCLEV is set so that the number of accomplishment levels can be displayed or changed.

LISTING:

```
     ∇ GETNUMACCLEV
[1]   ⍝
[2]   ⍝
[3]   ⍝   ROUTINE FOR FETCHING THE NUMBER OF MISSION ACCOMPLISHMENT
      LEVELS
[4]   ⍝
[5]   ⍝  INPUT NUMBER OF ACCOMPLISHMENT LEVELS
[6]   PRINT ''
[7]  GNAIN:PRINT 'NUMBER OF ACCOMPLISHMENT LEVELS?'
[8]   1 INPUT GNA
[9]   ⍝ CHECK FOR COMMAND
[10]  →(1=∧ ∈'COMMAND')/GNAIN
[11]  ⍝ CHECK VALIDITY OF INPUT
[12]  →(CHECKPOSI IN)/GNAIN
[13]  ⍝ ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT
[14]  NUMACCLEV←IN
[15]  DEFNUMACCLEV←1
     ∇
```

GETNUMBASICVARIABLES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETNUMBASICVARIABLES

PURPOSE: To obtain the number of time-invariant basic variables.

GLOBAL VARIABLES: DEFNUMBASICVARIABLES, IN, NUMBASICVARIABLES, GNBV.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: CHECKPOSI, INPUT, PRINT.

COMMENTS: Asks the user for the number of time-invariant basic variables, inputs the number, checks it, and places it in NUMBASICVARIABLES. If the value is not a positive integer, the query is repeated. In addition, the definition flag associated with NUMBASICVARIABLES is set so that the number of time-invariant basic variables can be displayed or changed.

LISTING:

```
      GETNUMBASICVARIABLES
[1]   ⍝
[2]   ⍝
[3]   ⍝  ROUTINE FOR FETCHING THE NUMBER OF TIME-INVARIANT) BASIC
      VARIABLES
[4]   ⍝
[5]   ⍝  INPUT NUMBER OF TIME-INVARIANT BASIC VARIABLES
[6]   PRINT ''
[7]  GNBVIN:PRINT 'NUMBER OF TIME-INVARIANT BASIC VARIABLES?'
[8]   1 INPUT GNBV
[9]   ⍝ CHECK FOR COMMAND
[10]  →(1=∧ ∈'COMMAND')/GNBVIN
[11]  ⍝ CHECK VALIDITY OF INPUT
[12]  →(IN=0)/GNBVSET
[13]  →(CHECKPOSI IN)/GNBVIN
[14]  ⍝  ELSE  SET  THE NUMBER OF TIME-INVARIANT BASIC VARIABLES AND
      EXIT
[15] GNBVSET:NUMBASICVARIABLES←IN
[16]  DEFNUMBASICVARIABLES←1
      ∇
```

7. *GETNUMBASICVARIABLES METAPHOR FUNCTION DESCRIPTION*

C-2

GETNUMPHASES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETNUMPHASES

PURPOSE: To obtain the number of phases.

GLOBAL VARIABLES: DEFNUMPHASES, IN, NUMPHASES, GNP.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: CHECKPOSI, INPUT, PRINT.

COMMENTS: Asks the user for the number of phases, inputs the number, checks it, and places it in NUMPHASES. If the value is not a positive integer, the query is repeated. In addition, the definition flag associated with NUMPHASES is set so that the number of phases can be displayed or changed.

LISTING:

```
     ∇ GETNUMPHASES
[1]    ⍝
[2]    ⍝
[3]    ⍝   ROUTINE FOR FETCHING THE NUMBER OF MISSION PHASES
[4]    ⍝
[5]    ⍝  INPUT NUMBER OF PHASES
[6]    PRINT ''
[7]   GNPIN:PRINT 'NUMBER OF PHASES?'
[8]    1 INPUT GNP
[9]    ⍝ CHECK FOR COMMAND
[10]   →(~1=∧ ∈'COMMAND')/GNPIN
[11]   ⍝ CHECK VALIDITY OF INPUT
[12]   →(CHECKPOSI IN)/GNPIN
[13]   ⍝ ELSE SET THE NUMBER OF PHASES AND EXIT
[14]   NUMPHASES←IN
[15]   DEFNUMPHASES←1
     ∇
```

GETNUMTRAJSETS

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETNUMTRAJSETS

PURPOSE: To obtain the number of trajectory sets in the accomplishment level being considered.

GLOBAL VARIABLES: DEFNUMTRAJSETS, IN, NUMTRAJSETS, GNTS.

CALLING FUNCTIONS: GETACCLEVPROB.

FUNCTIONS CALLED: CHECKPOSI, INPUT, PRINT.

COMMENTS: Asks the user for the number of trajectory sets in the accomplishment level being considered, inputs the number, checks it, and places it in NUMTRAJSETS. If the value is not a positive integer, the query is repeated. In addition, the definition flag associated with NUMTRAJSETS is set so that the number of trajectory sets in the accomplishment level being considered can be displayed or changed.

LISTING:

```
     ∇ Z←GETNUMTRAJSETS L
[1]    ⍝
[2]    ⍝
[3]    ⍝   ROUTINE FOR FETCHING THE NUMBER OF ACCOMPLISHMENT LEVEL
       TRAJECTORY SETS FOR LEVEL L
[4]    ⍝
[5]    ⍝  INPUT NUMBER OF TRAJECTORY SETS
[6]    PRINT ''
[7]    ⎕←((PRINTQUAD 'ACCOMPLISHMENT LEVEL ');PRINTQUAD L-1)
[8]    PRINT ''
[9]    GNTSIN:PRINT 'NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT
       LEVEL?'
[10]   1 INPUT GNTS
[11]   ⍝ CHECK FOR COMMAND
[12]   →(1=∧ ∈'COMMAND')/GNTSIN
[13]   ⍝ CHECK VALIDITY OF INPUT
[14]   →(CHECKPOSI IN)/GNTSIN
[15]   ⍝ ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT
[16]   Z←IN
```

```
[17]  DEFNUMTRAJSETS←1
      ∇
```

GETPERFORMABILITY

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETPERFORMABILITY

PURPOSE: To determine the performability for the model given METAPHOR

GLOBAL VARIABLES: DEFACCLEVEL, DEFPERFORMABILITY, NUMACCLEV, PERFORMABILITY.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: GETACCLEVPROB.

COMMENTS: For each accomplishment level, GETPERF calls GETACCLEVPROB to determine the probability of that accomplishment level. That result is stored in the performability vector, PERFORMABILITY. GETPERF also sets the definition flags associated with ACCLEVEL and PERFORMABILITY before calling GETACCLEVPROB so that the performability and the information associated with a particular accomplishment level can be altered and displayed. Before exiting, GETPERF resets the ACCLEVEL definition flag.

LISTING:

```
    ∇ GETPERFORMABILITY;LEVELPROB;L
[1]   ⍝
[2]   ⍝   ROUTINE FOR COMPUTING THE PERFORMABILITY OF THE SYSTEM
[3]   ⍝
[4]   ⍝  INITIALIZE THE PERFORMABILITY VECTOR AND SET DEFINITION
      FLAGS
[5]   PERFORMABILITY←NUMACCLEV⍴0
[6]   DEFACCLEVEL←1
[7]   DEFPERFORMABILITY←1
[8]   ⍝
[9]   ⍝  LOOP THROUGH EACH ACCOMPLISHMENT LEVEL
[10]  ⍝ INITIALIZE LEVEL COUNTER
[11]  L←1
[12]  ⍝  GET THE PROBABILITY OF EACH ACCOMPLISHMENT LEVEL
[13] GPLOOP:LEVELPROB←GETACCLEVPROB L
```

```
[14]  ⍝ INSERT THE PROBABILITY INTO THE PERFORMABILITY VECTOR
[15]  PERFORMABILITY[L]←LEVELPROB
[16]  ⍝ INCREMENT THE LEVEL COUNTER AND BRANCH IF NECESSARY
[17]  L←L+1
[18]  →(L≤NUMACCLEV)/GPLOOP
[19]  ⍝ ELSE EXIT ROUTINE, SETTING DFFINITION FLAG
[20]  DEFACCLEVEL←0
      ∇
```

GETSTATES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETSTATES

PURPOSE: To obtain the number of states in each phase of the model.

GLOBAL VARIABLES: DEFNUMSTATES, IN, NUMSTATES, STATESPERPHASE.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: CHECKPOSI, INPUT, PRINT.

COMMENTS: Asks the user for the number of states in each phase of the model, inputs the probabilities, checks them, and places then in STATESPERPHASE. If these values are not positive integers, the query is repeated. In addition, the definition flag associated with STATESPERPHASE is set so that the number of states in each phase of the model can be displayed or changed.

LISTING:

```
     ∇ GETSTATES
[1]   ⍝
[2]   ⍝   ROUTINE FOR FETCHING THE NUMBER OF STATES IN EACH PHASE
[3]   ⍝
[4]   ⍝INPUT NUMBER OF STATES AND CHECK VALIDITY
[5]   PRINT ''
[6]  GSIN:PRINT 'NUMBER OF STATES PER PHASE?  (SPACE BETWEEN EACH
     NUMBER)'
[7]   NUMPHASES INPUT GS
[8]   ⍝ CHECK FOR COMMAND
[9]   →(IN=¯1)/GSIN
[10]  ⍝ CHECK FOR POSITIVE INTEGER
[11]  →(CHECKPOSI IN)/GSIN
[12]  ⍝ SET STATES AND EXIT
[13]  STATESPERPHASE←IN
[14]  MAXNUMSTATES←⌈/STATESPERPHASE
[15]  DEFNUMSTATES←1
     ∇
```

GETFVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GETFVECTOR

PURPOSE: To input and check the characteristic vector for the trajectory set under consideration.

GLOBAL VARIABLES: IN, STATESPERPHASE, GFV.

CALLING FUNCTIONS: GETACCLEVPROB.

FUNCTIONS CALLED: CHECKBIN, INPUT, PRINT.

COMMENTS: Asks the user for the value of the characteristic vector for the phase under consideration, inputs the vector, checks it for proper binary values, and returns it in result. If these values are not proper binary quantities (i.e., 0 or 1), the query is repeated. In addition, the definition flag associated with F is set so that the F vector can be altered or displayed.

LISTING:

```
     ∇ Z←GETFVECTOR
[1]    ⍝
[2]    ⍝   ROUTINE FOR INPUTTING FINAL STATE VECTOR
[3]    ⍝
[4]    ⍝  INPUT AND CHECK THE F VECTOR
[5]   GFVIN:PRINT 'ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):'
[6]    STATESPERPHASE[NUMPHASES]INPUT GFV
[7]    ⍝ CHECK FOR COMMAND
[8]    →(1=∧/∊'COMMAND')/GFVIN
[9]    ⍝ CHECK VALIDITY OF INPUT.
[10]   →(CHECKBIN IN)/GFVIN
[11]   ⍝  SET F VECTOR AND LEAVE
[12]   Z←(MAXNUMSTATES,1)⍴IN,(MAXNUMSTATES⍴0)
[13]   DEFF←1
     ∇
```

GETGMATRICES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GETGMATRICES

PURPOSE: To input and check the characteristic matrices for the trajectory set under consideration.

GLOBAL VARIABLES: IN, STATESPERPHASE, GGM.

CALLING FUNCTIONS: GETACCLEVPROB.

FUNCTIONS CALLED: CHECKBIN, INPUT, PRINT.

COMMENTS: Asks the user for the values of the characteristic matrices for the phase under consideration, inputs the matrices, checks them for proper binary values, and returns them in result. If these values are not proper binary quantities (i.e., 0 or 1), the query is repeated. In addition, the definition flag associated with G is set so that the G matrices can be altered or displayed.

LISTING:

```
     ∇ Z←GETGMATRICES;PHASE
[1]    ⍝
[2]    ⍝  ROUTINE FOR INPUTTING THE G MATRICES FOR A TRAJECTORY
       SET
[3]    ⍝  EACH G MATRIX WILL BE REPRESENTED AS A VECTOR OF ITS
       DIAGONAL ELEMENTS
[4]    ⍝
[5]    ⍝  INITIALIZE THE G DIAGONAL MATRIX.  ALSO SET DEFINITION
       FLAG
[6]    Z←(MAXNUMSTATES,MAXNUMSTATES)⍴0
[7]    DEFG←1
[8]    ⍝
[9]    ⍝  LOOP THROUGH EACH PHASE EXCEPT THE LAST, GETTING THE G
       MATRICES
[10]   ⍝  INITIALIZE THE PHASE COUNTER
[11]   PHASE←1
[12]   ⍝  INPUT AND CHECK THE G VECTOR
[13] GGMIN:⎕←((PRINTQUAD 'PHASE ');(PRINTQUAD PHASE);PRINTQUAD
       ':')
```

```
[14]  PRINT ''
[15]  PRINT 'ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):'
[16]  STATESPERPHASE[PHASE]INPUT GGM
[17]  ⍝ CHECK FOR COMMAND
[18]  →(1=∧ ∈'COMMAND')/GGMIN
[19]  ⍝ CHECK VALIDITY OF INPUT.
[20]  →(CHECKBIN IN)/GGMIN
[21]  ⍝  PLACE THE INPUT IN THE SET OF G VECTORS
[22]  Z[PHASE;⍳⍴IN]←IN
[23]  ⍝ INCREMENT THE PHASE COUNTER AND BRANCH IF APPROPRIATE
[24]  PHASE←PHASE+1
[25]  →(PHASE<NUMPHASES)/GGMIN
      ∇
```

GETHMATRICES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETHMATRICES

PURPOSE: To supervise the input of the H matrices for the model.

GLOBAL VARIABLES: DEFH, H, MAXNUMSTATES, NUMPHASES, STATESPERPHASE.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: GENERATEHMATRIX, PRINT, PRINTQUAD.

COMMENTS: Fetches the H matrix for each phase transition via the GENERATEHMATRIX function. Each H matrix is then placed in the variable H. A message is printed at the beginning of the function informing the user that he is to enter the interphase transition matrices. Finally, before the function is exited, the definition flag associated with H is set, thus allowing H to be altered and displayed.

LISTING:

```
     ∇ GETHMATRICES;PHASE;NEXTH
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING THE H MATRICES
[3]   ⍝
[4]   ⍝   ONE MATRIX FOR EACH PHASE
[5]   ⍝
[6]   ⍝ IF NO H MATRICES, SET H←1 AND LEAVE
[7]   →(NUMPHASES≥2)/GHMMULTIPHASE
[8]   H← 1 1 ,1⍴1
[9]   ⍝  INITIALIZE THE ARRAY OF H MATRICES
[10] GHMMULTIPHASE:H←((NUMPHASES-1),MAXNUMSTATES, MAXNUMSTATES)⍴0
[11]  ⍝
[12]  ⍝  INPUT AND CHECK THE H MATRICES
[13]  PRINT ''
[14]  PRINT ''
[15]  PRINT 'SPECIFY THE H MATRICES FOR EACH PHASE, 1 PHASE AT A
      TIME'
[16]  ⍝
```

```
[17]  ⍝
[18]  ⍝  INITIALIZE PHASE COUNTER
[19]  PHASE←2
[20] GHMPHASEIN:PRINT ''
[21]  ⎕←((PRINTQUAD 'PHASE ');(PRINTQUAD PHASE-1);(PRINTQUAD
      '-');(PRINTQUAD PHASE);PRINTQUAD ':')
[22]  PRINT ''
[23]  ⍝  GET H MATRIX FOR PHASE
[24]  NEXTH←STATESPERPHASE[PHASE-1]GENERATEHMATRIX
      STATESPERPHASE[PHASE]
[25]  ⍝ INSERT THE MATRIX INTO THE ARRAY OF MATRICES
[26]  H[PHASE-1;⍳STATESPERPHASE[PHASE-1];
      ⍳STATESPERPHASE[PHASE]]←NEXTH
[27]  ⍝
[28]  ⍝  INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[29]  PHASE←PHASE+1
[30]  →(PHASE≤NUMPHASES)/GHMPHASEIN
[31]  ⍝ ELSE SET H DEFINITION FLAG AND LEAVE
[32]  DEFH←1
      ∇
```

GETIVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GETIVECTOR

PURPOSE: To input and check the initial vector for the trajectory set under consideration.

GLOBAL VARIABLES: IN, STATESPERPHASE, GIV.

CALLING FUNCTIONS: GETACCLEVPROB.

FUNCTIONS CALLED: CHECKBIN, INPUT, PRINT.

COMMENTS: Asks the user for the value of the initial vector for the phase under consideration, inputs the vector, checks it for proper binary values, and returns it in result. If these values are not proper binary quantities (i.e., 0 or 1), the query is repeated. In addition, the definition flag associated with I is set so that the I vector can be altered or displayed.

LISTING:

```
     ∇ Z←GETIVECTOR
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING INITIAL STATE VECTOR
[3]   ⍝
[4]   ⍝ INPUT AND CHECK THE I VECTOR
[5]  GIVIN:PRINT 'ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):'
[6]   STATESPERPHASE[1]INPUT GIV
[7]   ⍝ CHECK FOR COMMAND
[8]   →(1=∧ ∈'COMMAND')/GIVIN
[9]   ⍝ CHECK VALIDITY OF INPUT.
[10]  →(CHECKBIN IN)/GIVIN
[11]  ⍝  SET I VECTOR AND LEAVE
[12]  Z←MAXNUMSTATES⍴IN,(MAXNUMSTATES⍴0)
[13]  DEFI←1
     ∇
```

GETPMATRICES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETPMATRICES

PURPOSE: To supervise the input of the P matrices for the model.

GLOBAL VARIABLES: DEFP, P, MAXNUMSTATES, NUMPHASES, STATESPERPHASE.

CALLING FUNCTIONS: COMMANDEVAL.

FUNCTIONS CALLED: GENERATEPMATRIX, PRINT, PRINTQUAD.

COMMENTS: Fetches the P matrix for each phase via the GENERATEPMATRIX function. Each P matrix is then placed in the variable P. A message is printed at the beginning of the function informing the user that he is to enter the transition matrices. Finally, before the function is exited, the definition flag associated with P is set, thus allowing P to be altered and displayed.

LISTING:

```
     ∇ GETPMATRICES;PHASE;NEXTP
[1]    ⍝
[2]    ⍝   ROUTINE FOR INPUTTING THE P MATRICES
[3]    ⍝
[4]    ⍝   ONE MATRIX FOR EACH PHASE
[5]    ⍝
[6]    ⍝  INITIALIZE THE ARRAY OF P MATRICES
[7]    P←(NUMPHASES,MAXNUMSTATES,MAXNUMSTATES)⍴0
[8]    ⍝
[9]    ⍝  INPUT AND CHECK THE P MATRICES
[10]   PRINT ''
[11]   PRINT ''
[12]   PRINT 'SPECIFY THE P MATRICES FOR EACH PHASE, 1 PHASE AT A
       TIME'
[13]   ⍝
[14]   ⍝
[15]   ⍝  INITIALIZE PHASE COUNTER
[16]   PHASE←1
[17] GPMPHASEIN:PRINT ''
```

```
[18]  ⎕←((PRINTQUAD 'PHASE ');(PRINTQUAD PHASE);PRINTQUAD ':')
[19]  PRINT ''
[20]  ⍝  GET P MATRIX FOR PHASE
[21]  NEXTP←GENERATEPMATRIX STATESPERPHASE[PHASE]
[22]  ⍝ INSERT THE MATRIX INTO THE ARRAY OF MATRICES
[23]  P[PHASE;⍳STATESPERPHASE[PHASE];
      ⍳STATESPERPHASE[PHASE]]←NEXTP
[24]  ⍝
[25]  ⍝  INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[26]  PHASE←PHASE+1
[27]  →(PHASE≤NUMPHASES)/GPMPHASEIN
[28]  ⍝ ELSE SET P DEFINITION FLAG AND LEAVE
[29]  DEFP←1
      ∇
```

GETVVALUES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GETVVALUES

PURPOSE: To input and check the characterization of the time-invariant basic variables for the trajectory set under consideration.

GLOBAL VARIABLES: IN, NUMBASICVARIABLES, GVV.

CALLING FUNCTIONS: GETACCLEVPROB.

FUNCTIONS CALLED: CHECKBIN, INPUT, PRINT.

COMMENTS: Asks the user for the value of the characterization of the time-invariant basic variables for the phase under consideration, inputs the vector, checks it for proper binary values, and returns it in result. If these values are not proper binary quantities (i.e., 0 or 1), the query is repeated. In addition, the definition flag associated with V is set so that the V vector can be altered or displayed.

LISTING:

```
     ∇ Z←GETVVALUES;GVVTRUE;GVVFALSE
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING TIME+INVARIANT BASIC VARIABLE
      PROBABILITIES
[3]   ⍝
[4]   ⍝  INPUT AND CHECK THE V VECTOR
[5]   ⍝ IF NO TIME-INVARIANT BASIC VARIABLES, THEN SET V←0 , TURN ON
      THE DEFINITION FLAG, AND EXIT
[6]   Z←0
[7]   DEFV←1
[8]   →(NUMBASICVARIABLES=0)/0
[9]  GVVIN:⎕←((PRINTQUAD 'ENTER THE ');(PRINTQUAD
      NUMBASICVARIABLES);PRINTQUAD ' ELEMENT TIME+INVARIANT BASIC
      VARIABLE VECTOR  (SPACE BETWEEN EACH ENTRY):')
[10]  PRINT ''
[11]  NUMBASICVARIABLES INPUT GVV
[12]  ⍝ CHECK FOR COMMAND
[13]  →(1=∧ ∈'COMMAND')/GVVIN
```

```
[14]  ⍝ CHECK VALIDITY OF INPUT.
[15]  →(CHECKTRI IN)/GVVIN
[16]  ⍝ SET V VECTOR
[17]  ⍝ FIRST, SET THE TRUE VECTOR
[18]  GVVTRUE←IN∈ 0 2
[19]  ⍝ THEN SET THE FALSE VECTOR
[20]  GVVFALSE←IN∈ 1 2
[21]  ⍝ COMBINE FOR THE V VECTOR
[22]  Z←GVVTRUE,GVVFALSE
      ∇
```

INPUT

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: arg1 INPUT arg2

PURPOSE: To accept input from the user, check for the proper length, and coordinate command control.

GLOBAL VARIABLES: COMMANDLIST, ECHOINPUT, IN.

CALLING FUNCTIONS: GENERATEHMATRIX, GETBASICVARIABLES, GETNUMACCLEV, GETNUMBASICVARIABLES, GETNUMPHASES, GETNUMTRAJSETS, GETSTATES, GETFVECTOR, GETGMATRICES, GETIVECTOR, GETVVALUES, METAPHOR.

FUNCTIONS CALLED: PRINTQUAD, COMMANDALTER, COMMANDBRIEF, COMMANDCALC, COMMANDCOM, COMMANDDATA, COMMANDECHO, COMMANDEVAL, COMMANDHELP.

COMMENTS: Input from the user is accepted and placed in the variable IN. If the user has set ECHO ON, the input is repeated. This input is checked to make sure it is of length arg1. If not, an the input is then checked for a command. If a command is present, the proper COMMAND function is CALLED; AFTER THE COMMAND HAS BEEN EXECUTED, INPUT REPLACES IN WITH THE VALUE 'COMMAND' AND RETURNS. The value in arg2 is an indicator of the calling function and is used by the HELP function.

LISTING:

```
    ∇ SHAPE INPUT ROUTINE;COMMANDVECTOR
[1]   ⍝
[2]   ⍝  ROUTINE FOR INPUTTING ANSWERS.  INPUT CAN BE DATA OR
      COMMANDS.
[3]   ⍝     IF NUMERIC, DATA RETURNED IN 'IN', ELSE IF COMMAND,
      'COMMAND' RETURNED
[4]   ⍝
[5]   ⍝  GET INPUT
[6]  IN:IN←⍞
[7]   ⍝ ECHO INPUT IF DESIRED
[8]   →(~ECHOINPUT)/INVECTORIZE
```

```
[9]   ⍞←((PRINTQUAD '⍞:  ');PRINTQUAD IN)
[10]  PRINT ''
[11]  ⍝
[12]  ⍝ CHANGE ALL SCALAR INPUTS TO VECTORS
[13] INVECTORIZE:IN←,IN
[14]  ⍝
[15]  ⍝  CHECK FOR COMMAND AND SET COMMAND VECTOR
[16]  COMMANDVECTOR←∧/COMMANDLIST∈IN
[17]  ⍝
[18]  ⍝
[19]  ⍝ EXECUTE COMMAND IF PRESENT
[20] INCOMMAND:→(COMMANDVECTOR)/INHELP,INEXIT,INDATA,
      INALTER,INCALC,INECHO,INBRIEF,INCOM,INEVAL
[21]  ⍝ ELSE CHECK THE SHAPE OF THE INPUT.  'SHAPE' IS ALWAYS AN
      INTEGER SCALAR.  IF GOOD, LEAVE.
[22]  →(SHAPE=⍴IN)/0
[23]  ⍝ OTHERWISE THE INPUT WAS OF THE WRONG DIMENSION.  PRINT ERROR
      MESSAGE AND TRY AGAIN.
[24]  ⍞←((PRINTQUAD 'ENTER ');(PRINTQUAD SHAPE);PRINTQUAD '
      ITEMS')
[25]  →ININ
[26]  →0
[27]  ⍝
[28]  ⍝
[29]  ⍝ HELP REQUESTED
[30] INHELP:COMMANDHELP ROUTINE
[31]  IN←'COMMAND'
[32]  →0
[33]  ⍝
[34]  ⍝
[35]  ⍝ END THE PROGRAM
[36] INEXIT:→
[37]  ⍝
[38]  ⍝
[39]  ⍝ DISPLAY OF DATA REQUESTED
[40] INDATA:COMMANDDATA
[41]  IN←'COMMAND'
[42]  →0
[43]  ⍝
[44]  ⍝
[45]  ⍝ CHANGE OF DATA REQUESTED
[46] INALTER:COMMANDALTER
[47]  IN←'COMMAND'
[48]  →0
[49]  ⍝
[50]  ⍝
[51]  ⍝  CALCULATION OF PERFORMABILITY REQUESTED
[52] INCALC:COMMANDCALC
[53]  IN←'COMMAND'
[54]  →0
[55]  ⍝
[56]  ⍝ SETTING OF THE ECHO SWITCH DESIRED
[57] INECHO:COMMANDECHO IN
```

```
[58]   IN←'COMMAND'
[59]   →0
[60]   ⍝
[61]   ⍝ COMMENT DESIRED
[62]  INCOM:COMMANDCOM
[63]   IN←'COMMAND'
[64]   →0
[65]   ⍝
[66]   ⍝ SETTING OF BRIEF SWITCH DESIRED
[67]  INBRIEF:COMMANDBRIEF IN
[68]   IN←'COMMAND'
[69]   →0
[70]   ⍝
[71]   ⍝ PERFORMABILITY COMPUTATION DESIRED
[72]  INEVAL:COMMANDEVAL
[73]   →0
       ∇
```

INYES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← INYES

PURPOSE: To determine whether a user has answered 'YES' or 'NO' to a question.

GLOBAL VARIABLES: ECHOINPUT, IN.

CALLING FUNCTIONS: GDEDFAIL, GNFAIL, GAVINFO, GBVINFO, GDINFO, GDVINFO, GFVINFO, GGINFO, GGMINFO, , GHMINFO, GIVINFO, GNINFO, GNAINFO, GNBVINFO, GNPINFO, GNTSINFO, GPMINFO, GSINFO, GVVINFO, METINFO.

FUNCTIONS CALLED: PRINT.

COMMENTS: Prints a prompt mimicking the prompt associated with the APL 'quad' input. If the user has set ECHO ON, the input is repeated. The input is then scanned for yes or no answers. Specifically, this is achieved by looking first for either a 'Y' or a '1' somewhere in the input string. If either is found, the user is assumed to have answered 'YES.' if neither is found, the string is scanned for an 'N' or a '1'; if either is found, the user is then assumed to have answered 'NO.' If still none of these are found, the answer is considered ambiguous and the user is asked to enter another answer. If the user answers 'YES,' INYES returns a 1 in result, else if 'NO,' a 0 is returned.

LISTING:

```
     ∇ Z←INYES;IN
[1]   ⍝
[2]   ⍝  ROUTINE ASKING YES AND NO ANSWERS.  1 RETURNED IF YES, 0 IF
      NO
[3]   ⍝
[4]   ⍝ PRINT PROMPT AND GET ANSWER
[5]  IYIN:PRINT '□:'
[6]   IN←⍞
[7]   ⍝ ECHO THE INPUT IF DESIRED
[8]   →(~ECHOINPUT)/IYSCAN
[9]   PRINT(IN)
[10]  ⍝ ASSUME YES HAS PRIORITY.  LOOK FOR Y OR 1
[11] IYSCAN:Z←∨/'Y1'∈IN
[12]  ⍝ IF N OR 0 INPUT OR IF YES INPUT, EXIT
[13]  →(Z∨∨/'N0'∈IN)/0
[14]  ⍝ ELSE TRY AGAIN
[15]  PRINT 'ENTER YES OR NO'
[16]  →IYIN
     ∇
```

METAPHOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: METAPHOR

PURPOSE: The entry and main function for the METAPHOR software package.

GLOBAL VARIABLES: MET.

CALLING FUNCTIONS: None.

FUNCTIONS CALLED: DECLAREMETAPHOR, INPUT.

COMMENTS: Prints a heading identifying the package, version, and any current comments, additions, deletions, etc., regarding METAPHOR of which the user should be aware. METAPHOR then allows the user to input any commands which METAPHOR can support. METAPHOR continues in this command execution mode until an EXIT command is issued, causing the user to return to the APL command mode.

LISTING:

```
    ∇ METAPHOR
[1]   ⍝
[2]   ⍝
[3]   ⍝
[4]   ⍝  PROGRAM FOR EVALUATION OF PERFORMABILITIES
[5]   ⍝
[6]   ⍝
[7]   ⍝  SET PROGRAM CONSTANTS
[8]   DECLAREMETAPHOR
[9]   ⍝
[10]  ⍝
[11]  ⍝ PRINT HEADING
[12]  PRINT ''
[13]  PRINT ''
[14]  PRINT 'MICHIGAN EVALUATION AID FOR PERPHORMABILITY'
[15]  PRINT ''
[16]  PRINT ''
[17]  PRINT ''
[18]  PRINT ''
[19]  PRINT 'TYPE HELP FOR ASSISTANCE'
```

```
[20]  ⍝  PROGRAM LOOPS UNTIL EXIT TAKEN
[21] MIN:6 INPUT MET
[22]  →MIN
      ∇
```

PRINT

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← PRINT arg

PURPOSE: To print output from METAPHOR if the BRIEF ON command has not been issued

GLOBAL VARIABLES: BRIEFOUTPUT.

CALLING FUNCTIONS: CHECKBIN, CHECKPOSI, CHECKPROB, CHECKTRI, GENERATEHMATRIX, GENERATEPMATRIX, GETACCLEVPROB, GETALTERVECTOR, GETBASICVARIABLES, GETDATAVECTOR, GETNUMACCLEV, GETNUMBASICVARIABLES, GETNUMPHASES, GETNUMTRAJSETS, GETSTATES, GETFVECTOR, GETGMATRICES, GETHMATRICES, GETIVECTOR, GETPMATRICES, GETVVALUES, INYES, METAPHOR, PRINTQUAD, PRINTPERFORMABILITY, COMMANDALTER, COMMANDBRIEF, COMMANDDATA, COMMANDECHO, GDEDFAIL, GGIVEN, GNFAIL, GAVINFO, GBVINFO, GDINFO, GDVINFO, GFVINFO, GGINFO, GGMINFO, GHMINFO, GIVINFO, GNINFO, GNAINFO, GNBVINFO, GNPINFO, GNTSINFO, GPMINFO, GSINFO, GVVINFO, METINFO.

FUNCTIONS CALLED: NONE.

COMMENTS: arg is the string which METAPHOR desires to print, with a carriage return at the end of the line. PRINT returns that string in result if the user has not issued the BRIEF ON command, else, PRINT returns the null string.

LISTING:

```
     ∇ Z←PRINT Q
[1]   ⍝
[2]   ⍝
[3]   ⍝    PRINTING ROUTINE FOR METAPHOR
[4]   ⍝  IF BRIEF←1 NO OUTPUT IS GIVEN
[5]   ⍝
[6]   ⍝ CHECK FOR TERSE INPUT FLAG
[7]   →(BRIEFOUTPUT=1)/0
[8]   ⍝ FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[9]   Z←Q
     ∇
```

PRINTPERFORMABILITY

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: PRINTPERFORMABILITY

PURPOSE: To print the value of the performability once it is computed.

GLOBAL VARIABLES: PERFORMABILITY.

CALLING FUNCTIONS: EVAL

FUNCTIONS CALLED: PRINT.

COMMENTS: Prints the performability of the mission. If the performability does of the performability and the warning cannot be suppressed.

LISTING:

```
∇ PRINTPERFORMABILITY
[1]   ⍝
[2]   ⍝   ROUTINE FOR PRINTING THE PERFORMABILITY ONCE IT IS
      CALCULATED
[3]   ⍝
[4]   PRINT ''
[5]   PRINT ''
[6]   PRINT ''
[7]   'PERFORMABILITY FOR THIS MISSION ← ';PERFORMABILITY
[8]   ⍝
[9]   ⍝  PRINT WARNING IF THE PERFORMABILITY DOES NOT SUM TO
      ONE.  ELSE EXIT
[10]  →(1=+/PERFORMABILITY)/0
[11]  'CAUTION...THE PERFORMABILITY SUMS TO ';(+
      /PERFORMABILITY);', NOT EQUAL TO ONE'
      ∇
```

PRINTQUAD

METAPHOR FUNCTION DESCRIPTION

PURPOSE: To print output from METAPHOR if the BRIEF ON command has not been issued. No carriage return is appended to the output.

CALLING SEQUENCE: result ← PRINTQUAD arg

GLOBAL VARIABLES: BRIEFOUTPUT.

CALLING FUNCTIONS: GETACCLEVPOB, GETNUMTRAJSETS, GETGMATRICES, GETHMATRICES, GETPMATRICES, GETVVALUES, INPUT, COMMANDCALC, COMMANDCOM, COMMANDDATA, GDEDFAIL, GGIVEN, GNFAIL.

FUNCTIONS CALLED: None.

COMMENTS: arg is part of a string METAPHOR desires to print with no carriage return after the printing. PRINTQUAD returns that string in result if the user has not issued the BRIEF ON command; else, METAPHOR returns the null string.

LISTING:

```
     ∇ Z←PRINTQUAD Q
[1]   ⍝
[2]   ⍝
[3]   ⍝    PRINTING ROUTINE (WITH QUOTE QUAD] FOR METAPHOR
[4]   ⍝  IF BRIEF←1 NO OUTPUT IS GIVEN
[5]   ⍝
[6]   ⍝ CHECK FOR TERSE INPUT FLAG
[7]   →(BRIEFOUTPUT=1)/PQBRIEF
[8]   ⍝ FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[9]   Z←Q
[10]  →0
[11]  ⍝ ELSE RETURN THE EMPTY STRING
[12] PQBRIEF:Z←''
     ∇
```

COMMANDALTER

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDALTER

PURPOSE: To execute the ALTER command by altering the data which the user specifies.

GLOBAL VARIABLES: DEFACCLEVEL, DEFBASICVARIABLES, DEFF, DEFG, DEFH, DEFI, DEFNUMACCLEV, DEFNUMTRAJSETS, DEFP, DEFV.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: GETACCLEVPROB, GETALTERVECTOR, GETBASICVARIABLES, GETNUMACCLEV, GETNUMBASICVARIABLES, GETNUMTRAJSETS, GETFVECTOR, GETGMATRICES, GETHMATRICES, GETIVECTOR, GETPMATRICES, GETVVALUES, PRINT.

COMMENTS: Calls GETALTERVECTOR to determine which values should be changed. ALTERVECTOR is a binary vector such that the position of the 1's and 0's has the following code:

Position 1--Alter the P matrices.

Position 2--Alter the H matrices.

Index 3--Alter the time-invariant basic variables.

Index 4--Alter the values of all accomplishment levels.

Index 5--Alter the values of just the present accomplishment level.

Index 6--Alter the initial vector.

Index 7--Alter the characteristic matrices.

Index 8--Alter the characteristic vector.

Index 9--Alter the characterization of the time-invariant basic variables.

Index 10--Alter the number of trajectory sets.

For each alteration specified, COMMANDALTER checks via the definition flags (e.g., DEFBASICVARIABLES) that the corresponding value has been defined. Changed and calls the suitable functions to implement the change. When all requests have been processed, the function returns.

LISTING:

```
∇ COMMANDALTER;ALTERVECTOR
[1]   ⍝
[2]   ⍝   ROUTINE FOR CHANGING CURRENT DATA
[3]   ⍝
[4]   ⍝  GET DATA TO BE ALTERED
[5]   ALTERVECTOR←GETALTERVECTOR
[6]   ⍝
[7]   ⍝ CHANGE THAT DATA.  EXIT WHEN THROUGH.
[8]  ALOOP:→ALTERVECTOR/AP,AH,ABASICVARIABLES,AALLACCLEVELS,
      APRESENTACCLEVEL,AI,AG,AF,AV,ANUMTRAJSETS
[9]   →0
[10]  ⍝
[11]  ⍝
[12]  ⍝  CHANGE THE REQUESTED DATA
[13]  ⍝
[14]  ⍝ CHANGE THE P MATRICES
[15] AP:→(DEFP=1)/APALTER
[16]  ⍝P MATRICES ARE NOT DEFINED AT THIS TIME.  PRINT ERROR
      MESSAGE
[17]  PRINT 'P MATRICES ARE NOT DEFINED AT THIS TIME.'
[18]  →APLOOP
[19] APALTER:PRINT 'ALTERING P'
[20]  GETPMATRICES
[21] APLOOP:ALTERVECTOR[1]←0
[22]  →ALOOP
[23]  ⍝
[24]  ⍝ CHANGE H MATRICES
[25] AH:→(DEFH=1)/AHALTER
[26]  ⍝H MATRICES ARE NOT DEFINED AT THIS TIME.  PRINT ERROR
      MESSAGE
[27]  PRINT 'H MATRICES ARE NOT DEFINED AT THIS TIME.'
[28]  →AHLOOP
[29] AHALTER:PRINT 'ALTERING H'
[30]  GETHMATRICES
[31] AHLOOP:ALTERVECTOR[2]←0
[32]  →ALOOP
```

7. *COMMANDALTER METAPHOR FUNCTION DESCRIPTION*

```
[33]  ⍝
[34]  ⍝ CHANGE THE TIME+INVARIANT BASIC VARIABLES
[35] ABASICVARIABLES:→(
      DEFBASICVARIABLES=1)/ABASICVARIABLESALTER
[36]  ⍝TIME+INVARIANT BASIC VARIABLES ARE NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE
[37]  PRINT 'TIME-INVARIANT BASIC VARIABLES ARE NOT DEFINED AT
      THIS TIME.'
[38]  →ABASICVARIABLESLOOP
[39] ABASICVARIABLESALTER:PRINT 'ALTERING THE NUMBER OF BASIC
      VARIABLES'
[40]  GETNUMBASICVARIABLES
[41]  GETBASICVARIABLES
[42] ABASICVARIABLESLOOP:ALTERVECTOR[3]←0
[43]  →ALOOP
[44]  ⍝
[45]  ⍝ CHANGE ALL ACCOMPLISHMENT LEVELS
[46] AALLACCLEVELS:→(DEFNUMACCLEV=1)/AALLACCLEVELSALTER
[47]  ⍝THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE
[48]  PRINT 'THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS
      TIME.'
[49]  →AALLACCLEVELSLOOP
[50] AALLACCLEVELSALTER:PRINT 'ALTERING THE NUMBER OF
      ACCOMPLISHMENT LEVELS'
[51]  GETNUMACCLEV
[52]  GETACCLEVPROB
[53] AALLACCLEVELSLOOP:ALTERVECTOR[4]←0
[54]  →ALOOP
[55]  ⍝
[56]  ⍝ CHANGE JUST THE PRESENT (LAST) ACCOMPLISHMENT LEVEL
[57] APRESENTACCLEVEL:→(DEFACCLEVEL=1)/APRESENTACCLEVELALTER
[58]  ⍝AN ACCOMPLISHMENT LEVEL IS NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE
[59]  PRINT 'AN ACCOMPLISHMENT LEVEL IS NOT DEFINED AT THIS
      TIME.'
[60]  →APRESENTACCLEVELLOOP
[61] APRESENTACCLEVELALTER:PRINT 'ALTERING THE PRESENT
      ACCOMPLISHMENT LEVEL'
[62]  PERFORMABILITY[L]←GETACCLEVPROB L
[63] APRESENTACCLEVELLOOP:ALTERVECTOR[5]←0
[64]  →ALOOP
[65]  ⍝
[66]  ⍝ CHANGE THE I VECTOR
[67] AI:→(DEFI=1)/AIALTER
[68]  ⍝I VECTOR IS NOT DEFINED AT THIS TIME.  PRINT ERROR
      MESSAGE
[69]  PRINT 'I VECTOR IS NOT DEFINED AT THIS TIME.'
[70]  →AILOOP
[71] AIALTER:PRINT 'ALTERING THE I VECTOR'
[72]  I←GETIVECTOR
[73] AILOOP:ALTERVECTOR[6]←0
[74]  →ALOOP
```

```
[75]  ⍝
[76]  ⍝ CHANGE THE G MATRICES
[77] AG:→(DEFG=1)/AGALTER
[78]  ⍝G MATRICES ARE NOT DEFINED AT THIS TIME.  PRINT ERROR
      MESSAGE
[79]  PRINT 'G MATRICES ARE NOT DEFINED AT THIS TIME.'
[80]  →AGLOOP
[81] AGALTER:PRINT 'ALTERING THE G MATRICES'
[82]  G←GETGMATRICES
[83] AGLOOP:ALTERVECTOR[7]←0
[84]  →ALOOP
[85]  ⍝
[86]  ⍝ CHANGE THE F VECTOR
[87] AF:→(DEFF=1)/AFALTER
[88]  ⍝F VECTOR IS NOT DEFINED AT THIS TIME.  PRINT ERROR
      MESSAGE
[89]  PRINT 'F VECTOR IS NOT DEFINED AT THIS TIME.'
[90]  →AFLOOP
[91] AFALTER:PRINT 'ALTERING THE F VECTOR'
[92]  F←GETFVECTOR
[93] AFLOOP:ALTERVECTOR[8]←0
[94]  →ALOOP
[95]  ⍝
[96]  ⍝ CHANGE THE TIME+INVARIANT BASIC VARIABLE VECTOR
[97] AV:→(DEFV=1)/AVALTER
[98]  ⍝THE TIME+INVARIANT BASIC VARIABLE VECTOR IS NOT DEFINED
      AT THIS TIME.  PRINT ERROR MESSAGE
[99]  PRINT 'THE TIME+INVARIANT BASIC VARIABLE VECTOR IS NOT
      DEFINED AT THIS TIME.'
[100] →AVLOOP
[101]AVALTER:PRINT 'ALTERING THE V VECTOR'
[102] V←GETVVALUES
[103]AVLOOP:ALTERVECTOR[9]←0
[104] →ALOOP
[105] ⍝
[106] ⍝ CHANGE THE NUMBER OF TRAJECTORY SETS BEING
      CONSIDERED
[107]ANUMTRAJSETS:→(DEFNUMTRAJSETS=1)/ANUMTRAJSETSALTER
[108] ⍝THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE.
[109] PRINT 'THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT
      THIS TIME.'
[110]ANUMTRAJSETSALTER:PRINT 'ALTERING THE NUMBER OF
      TRAJECTORY SETS'
[111] NUMTRAJSETS←GETNUMTRAJSETS
[112] ALTERVECTOR[10]←0
[113] →ALOOP
      ∇
```

COMMANDBRIEF

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDBRIEF arg

PURPOSE: To execute the BRIEF command by setting the BRIEF switch to arg.

GLOBAL VARIABLES: BRIEFOUTPUT.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: None.

COMMENTS: The brief switch is set to the value of arg. Usually, this should
arg. Usually, this should be either ON or OFF. A confirmation is also printed.

LISTING:

```
     ∇ COMMANDBRIEF SWITCH
[1]   ⍝
[2]   ⍝ ROUTINE FOR TURNING THE BRIEF SWITCH ON AND OFF. 'ON'
      CAUSES
[3]   ⍝      METAPHOR TO USE A TERSE FORM OF OUTPUT. 'OFF' CAUSE THE
      NORMAL FULL OUTPUT.
[4]   ⍝
[5]   ⍝ TURN SWITCH ON IF REQUESTED, ELSE TURN SWITCH OFF.
[6]   BRIEFOUTPUT←∧/'ON'∈SWITCH
[7]   ⍝ PRINT CONFIRMATION REGARDLESS OF BRIEF SWITCH
[8]   'BRIEF ';SWITCH
     ∇
```

COMMANDCALC

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDCALC

PURPOSE: To execute the CALC command by allowing the user to utilize the APL calculator mode.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: INPUT

FUNCTIONS CALLED: None.

COMMENTS: Prints a prompt sign '?' and inputs the users equation. If the input is

LISTING:

```
     ∇ COMMANDCALC;CCALCINPUT
[1]    ⍝
[2]    ⍝   ROUTINE TO UTILIZE THE APL CALCULATOR MODE FROM THE
       METAPHOR PACKAGE.
[3]    ⍝      HALTS WHEN THE INPUT IS NULL.
[4]    ⍝
[5]  CCALCIN:⎕←(PRINTQUAD '?')
[6]    ⎕←CCALCINPUT←⍞
[7]    ⍝ LEAVE IF EXIT SPECIFIED.  ELSE GET,NEXT CALCULATION.
[8]    →(1=∧/CCALCINPUT=EXIT)/0
[9]    →CCALCIN
     ∇
```

COMMANDCOM

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDCOM

PURPOSE: To allow the user to enter a comment on the output.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: None.

COMMENTS: Prints a prompt sign '***' and allows the user to print a comment on the same line. If the comment is empty, COMMANDCOM exits; else the process is repeated.

LISTING:

```
     ∇ COMMANDCOM;CCINPUT
[1]    ⍝
[2]    ⍝  ROUTINE FOR ALLOWING THE USER TO PRINT A COMMENT
[3]    ⍝     EACH COMMENT IS PRECEDED BY '***' AND THIS
[4]    ⍝     HALTS WHEN THE INPUT IS NULL
[5]    ⍝
[6]    ⍝ PRINT PROMPT SYMBOLS
[7]   CCIN:⎕←(PRINTQUAD '***')
[8]    CCINPUT←⍞
[9]    ⍝ IF COMMENT NOT EMPTY, GET NEXT COMMENT.  ELSE LEAVE.
[10]   →(3≠⍴CCINPUT)/CCIN
     ∇
```

COMMANDDATA

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDDATA

PURPOSE: To execute the DATA command by displaying the data which the user specifies.

GLOBAL VARIABLES: DEFBASICVARIABLES, DEFF, DEFG, DEFH, DEFI, DEFNUMACCLEV, DEFNUMBASICVARIABLES, DEFNUMPHASES, DEFNUMSTATES, DEFNUMTRAJSETS, DEFP, DEFPERFORMABILITY, DEFV.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: GETDATAVECTOR, PRINT, PRINTQUAD.

COMMENTS: Calls GETDATAVECTOR to determine which values should be displayed. DATAVECTOR is a binary vector such that the position of the 1's and 0's has the following code:

Position 1--Display the number of phases.

Position 2--Display the number of states in each phase.

Position 3--Display the P matrices.

Position 4--Display the H matrices.

Position 5--Display the number of time-invariant basic variables.

Position 6--Display the probabilities of each of the time-invariant basic variables.

Position 7--Display the number of accomplishment levels.

Position 8--Display the number of trajectory sets.

Position 9--Display the initial vector.

Position 10--Display the characteristic matrices.

Position 11--Display the characteristic vector.

Position 12--Display the characterization of the time-invariant basic variables.

Position 13--Display the performability.

For each display specified, COMMANDDATA checks via the definition flags (e.g., DEFBASICVARIABLES) that the corresponding value has been defined. Displayed and then displays the item. When all requests have been processed, the function returns.

LISTING:

```
     ∇ COMMANDDATA;DATAVECTOR
[1]    ⍝
[2]    ⍝   ROUTINE FOR DISPLAYING CURRENT DATA
[3]    ⍝
[4]    ⍝  GET DATA TO BE DISPLAYED
[5]    DATAVECTOR←GETDATAVECTOR
[6]    ⍝
[7]    ⍝ DISPLAY THAT DATA.  EXIT WHEN THROUGH.
[8]  DLOOP:→DATAVECTOR/DNUMPHASES,DNUMSTATES,DP,DH,DNUMBASICVAR
       DBASICVARIABLES,DNUMACCLEV,DNUMTRAJSETS,DI,DG,DF,DV,DPERF
[9]    →0
[10]   ⍝
[11]   ⍝
[12]   ⍝  SHOW THE REQUESTED INFORMATION
[13]   ⍝
[14] DNUMPHASES:→(DEFNUMPHASES=1)/DNUMPHASESOUT
[15]   PRINT 'NUMBER OF PHASES HAS NOT BEEN DEFINED'
[16]   →DNUMPHASESLOOP
[17] DNUMPHASESOUT:⍞←((PRINTQUAD 'NUMBER OF PHASES IS ');PRINTQUAD
       NUMPHASES)
[18]   PRINT ''
[19] DNUMPHASESLOOP:DATAVECTOR[1]←0
[20]   →DLOOP
[21]   ⍝
[22] DNUMSTATES:→(DEFNUMSTATES=1)/DNUMSTATESOUT
[23]   PRINT 'NUMBER OF STATES HAS NOT BEEN DEFINED'
[24]   →DNUMSTATESLOOP
[25] DNUMSTATESOUT:⍞←((PRINTQUAD 'NUMBER OF STATES PER PHASE IS
       ');PRINTQUAD STATESPERPHASE)
[26]   PRINT ''
```

```
[27] DNUMSTATESLOOP:DATAVECTOR[2]←0
[28]  →DLOOP
[29]  ⍝
[30] DP:→(DEFP=1)/DPOUT
[31]  PRINT 'P MATRICES HAVE NOT BEEN DEFINED'
[32]  →DPLOOP
[33] DPOUT:PRINT 'THE P MATRICES ARE:
[34]  P
[35] DPLOOP:DATAVECTOR[3]←0
[36]  →DLOOP
[37]  ⍝
[38] DH:→(DEFH=1)/DHOUT
[39]  PRINT 'H MATRICES HAVE NOT BEEN DEFINED'
[40]  →DHLOOP
[41] DHOUT:PRINT 'THE H MATRICES ARE:'
[42]  H
[43] DHLOOP:DATAVECTOR[4]←0
[44]  →DLOOP
[45]  ⍝
[46] DNUMBASICVARIABLES:(
      DEFNUMBASICVARIABLES=1)/DNUMBASICVARIABLESOUT
[47]  PRINT 'THE NUMBER OF BASIC VARIABLES HAS NOT BEEN DEFINED'
[48]  →DNUMBASICVARIABLESLOOP
[49] DNUMBASICVARIABLESOUT:⍞←((PRINTQUAD 'THE NUMBER OF TIME+
      INVARIANT BASIC VARIABLES IS ');PRINTQUAD NUMBASICVARIABLES)
[50]  PRINT ''
[51] DNUMBASICVARIABLESLOOP:DATAVECTOR[5]←0
[52]  →DLOOP
[53]  ⍝
[54] DBASICVARIABLES:→(DEFBASICVARIABLES=1)/DBASICVARIABLESOUT
[55]  PRINT 'THE TIME+INVARIANT BASIC VARIABLES HAVE NOT BEEN
      DEFINED'
[56]  →DBASICVARIABLESLOOP
[57] DBASICVARIABLESOUT:PRINT 'THE TIME+INVARIANT BASE VARIABLES
      HAVE PROBABILITIES:'
[58]  BASICVARIABLES
[59] DBASICVARIABLESLOOP:DATAVECTOR[6]←0
[60]  →DLOOP
[61]  ⍝
[62] DNUMACCLEV:→(DEFNUMACCLEV=1)/DNUMACCLEVOUT
[63]  PRINT 'THE NUMBER OF ACCOMPLISHMENT LEVELS NOT DEFINED'
[64]  →DNUMACCLEVLOOP
[65] DNUMACCLEVOUT:⍞←((PRINTQUAD 'THE NUMBER OF ACCOMPLISHMENT
      LEVELS IS ');PRINTQUAD NUMACCLEV)
[66]  PRINT ''
[67] DNUMACCLEVLOOP:DATAVECTOR[7]←0
[68]  →DLOOP
[69]  ⍝
[70] DNUMTRAJSETS:→(DEFNUMTRAJSETS=1)/DNUMTRAJSETSOUT
[71]  PRINT 'THE NUMBER OF TRAJECTORY SETS NOT DEFINED'
[72]  →DNUMTRAJSETSLOOP
[73] DNUMTRAJSETSOUT:PRINT 'THE NUMBER OF TRAJECTORY SETS IS:'
[74]  NUMTRAJSETS
```

```
[75] DNUMTRAJSETSLOOP:DATAVECTOR[8]←0
[76]  →DLOOP
[77]  ⍝
[78] DI:→(DEFI=1)/DIOUT
[79]  PRINT 'I VECTOR NOT DEFINED'
[80]  →DILOOP
[81] DIOUT:⍞←((PRINTQUAD 'THE INITIAL VECTOR IS ');PRINTQUAD I)
[82]  PRINT ''
[83] DILOOP:DATAVECTOR[9]←0
[84]  →DLOOP
[85]  ⍝
[86] DG:→(DEFG=1)/DGOUT
[87]  PRINT 'G MATRICES NOT DEFINED'
[88]  →DGLOOP
[89] DGOUT:PRINT 'THE G MATRICES ARE:'
[90]  G
[91] DGLOOP:DATAVECTOR[10]←0
[92]  →DLOOP
[93]  ⍝
[94] DF:→(DEFF=1)/DFOUT
[95]  PRINT 'F VECTOR NOT DEFINED'
[96]  →DFLOOP
[97] DFOUT:⍞←((PRINTQUAD 'THE FINAL VECTOR IS ');PRINTQUAD F)
[98]  PRINT ''
[99] DFLOOP:DATAVECTOR[11]←0
[100] →DLOOP
[101] ⍝
[102]DV:→(DEFV=1)/DVOUT
[103] PRINT 'THE TIME+INVARIANT BASIC VARIABLE VECTOR NOT DEFINED'
[104] →DVLOOP
[105]DVOUT:⍞←((PRINTQUAD 'THE TIME+INVARIANT BASIC VARIABLE VECTOR
      IS ');PRINTQUAD V)
[106] PRINT ''
[107]DVLOOP:DATAVECTOR[12]←0
[108] →DLOOP
[109] ⍝
[110]DPERF:→(DEFPERFORMABILITY=1)/DPERFOUT
[111] PRINT 'PERFORMABILITY NOT DEFINED'
[112] →DPERFLOOP
[113]DPERFOUT:⍞←((PRINTQUAD 'THE PERFORMABILITY IS ');PRINTQUAD
      PERFORMABILITY)
[114]DPERFLOOP:DATAVECTOR[13]←0
[115] →DLOOP
      ∇
```

COMMANDECHO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDECHO arg

PURPOSE: To execute the ECHO command by setting the ECHO switch to arg.

GLOBAL VARIABLES: ECHOINPUT.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: None.

COMMENTS: The echo switch is set to the value of arg. Usually, this should be either ON or OFF. A confirmation is also printed.

LISTING:

```
     ∇ COMMANDECHO SWITCH
[1]    ⍝
[2]    ⍝ ROUTINE FOR TURNING THE ECHO SWITCH ON AND OFF. 'ON'
       CAUSES
[3]    ⍝       METAPHOR TO REPEAT EVERY INPUT LINE. 'OFF' SUPPRESSES
       THE REPETITION.
[4]    ⍝
[5]    ⍝ TURN SWITCH ON IF REQUESTED, ELSE TURN SWITCH OFF.
[6]    ECHOINPUT←∧/'ON'∈SWITCH
[7]    'ECHO ';SWITCH
     ∇
```

COMMANDEVAL

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDEVAL

PURPOSE: To execute the 'EVAL' command by calling the proper functions to evaluate the user's performability model.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: GETBASICVARIABLES, GETNUMACCLEV, GETNUMBASICVARIABLES, GETNUMPHASES, GETPERFORMABILITY, GETSTATES, GETHMATRICES, GETPMATRICES, PRINTPERFORMABILITY.

COMMENTS: Gets the number of phases in the finite phase model, the number of states in each of those phases, the intraphase state transition matrices, the interphase state transition matrices, the number of time-invariant basic variables, and the probabilities of those time-invariant basic variables.

LISTING:

```
    ∇ COMMANDEVAL
[1]   ⍝
[2]   ⍝   PERFORMABILITY COMPUTATION PORTION OF METAPHOR
[3]   ⍝
[4]   ⍝  FETCH THE MATRICES REQUIRED FOR PERFORMABILITY
      CALCULATIONS
[5]  MGNUMP:GETNUMPHASES
[6]   GETSTATES
[7]   GETPMATRICES
[8]   GETHMATRICES
[9]   GETNUMBASICVARIABLES
[10]  GETBASICVARIABLES
[11]  GETNUMACCLEV
[12]  ⍝
[13]  ⍝  FOR EACH ACCOMPLISHMENT LEVEL, FETCH THE TRAJECTORY SETS
      AND CALCULATE THEIR PROBABILITIES
[14]  GETPERFORMABILITY
```

```
[15]  ⍝
[16]  ⍝  PRINT THE RESULTING PERFORMABILITY
[17]  PRINTPERFORMABILITY
      ∇
```

COMMANDHELP

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: COMMANDHELP arg with the calling function indicated in arg.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: INPUT.

FUNCTIONS CALLED: GAVINFO, GBVINFO, GDINFO, GDVINFO, GFVINFO, GGINFO, GGMINFO, , GHMINFO, GIVINFO, GNINFO, GNAINFO, GNBVINFO, GNPINFO, GNTSINFO, GPMINFO, GSINFO, GVVINFO, METINFO.

COMMENTS: Each function in METAPHOR that asks a (non-yes/no) question has a number associated with it. WHEN COMMANDHELP' is typed in response to such a question, INPUT passes the number to COMMANDHELP for processing. COMMANDHELP then chooses the proper INFO function and executes it.

LISTING:

```
     ∇ COMMANDHELP ROUTINE
[1]    ⍝
[2]    ⍝  ROUTINE PRINTING HELP INFORMATION
[3]    ⍝
[4]    ⍝ GET THE PROPER INFO ROUTINE
[5]    →(ROUTINE=⍳NUMHELPROUTINES)/HGNP,HGS,HGPM,HGG
       HGN,HGD,HGHM,HGNBV,HGBV,HGNA,HGIV,
       HGGM,HGFV,HGVV,HGNTS,HGAV,HGDV,HMET
[6]    ⍝
[7]    ⍝ 'GET NUMBER OF PHASES' HELP
[8]  HGNP:GNPINFO
[9]    →0
[10]   ⍝
[11]   ⍝ 'GET STATES' HELP
[12] HGS:GSINFO
[13]   →0
[14]   ⍝
[15]   ⍝ 'GENERATE P MATRIX' HELP
[16] HGPM:GPMINFO
[17]   →0
[18]   ⍝
[19]   ⍝ 'GET GIVEN MATRIX' HELP
```

```
[20] HGG:GGINFO
[21]  →0
[22]  ⍵
[23]  ⍵ 'GET NFAIL MATRIX' HELP
[24] HGN:GNINFO
[25]  →0
[26]  ⍵
[27]  ⍵ 'GET DEDFAIL MATRIX' HELP
[28] HGD:GDINFO
[29]  →0
[30]  ⍵
[31]  ⍵ 'GENERATE H MATRIX' HELP
[32] HGHM:GHMINFO
[33]  →0
[34]  ⍵
[35]  ⍵ 'GET NUMBER OF TIME+INVARIANT BASIC VARIABLES' HELP
[36] HGNBV:GNBVINFO
[37]  →0
[38]  ⍵
[39]  ⍵ 'GET TIME-INVARIANT BASIC VARIABLES' HELP
[40] HGBV:GBVINFO
[41]  →0
[42]  ⍵
[43]  ⍵ 'GET NUMBER OF ACCOMPLISHMENT LEVELS' HELP
[44] HGNA:GNAINFO
[45]  →0
[46]  ⍵
[47]  ⍵ 'GET I VECTOR' HELP
[48] HGIV:GIVINFO
[49]  →0
[50]  ⍵
[51]  ⍵ 'GET G MATRIX' HELP
[52] HGGM:GGMINFO
[53]  →0
[54]  ⍵
[55]  ⍵ 'GET F VECTOR' HELP
[56] HGFV:GFVINFO
[57]  →0
[58]  ⍵
[59]  ⍵ 'GET V VECTOR' HELP
[60] HGVV:GVVINFO
[61]  →0
[62]  ⍵
[63]  ⍵ 'GET NUMBER OF TRAJECTORY SETS' HELP
[64] HGNTS:GNTSINFO
[65]  →0
[66]  ⍵
[67]  ⍵ 'GET ALTER VECTOR VECTOR' HELP
[68] HGAV:GAVINFO
[69]  →0
[70]  ⍵
[71]  ⍵ 'GET DATA VECTOR VECTOR' HELP
[72] HGDV:GDVINFO
```

```
[73]  →0
[74]  ⍝
[75]  ⍝ 'METAPHOR' HELP
[76] HMET:METINFO
[77]  →0
      ∇
```

GDEDFAIL

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GDEDFAIL arg

PURPOSE: To generate a DEDFAIL type matrix with shape arg*arg.

GLOBAL VARIABLES: IN, GD.

CALLING FUNCTIONS: GENERATEPMATRIX.

FUNCTIONS CALLED: ENCODE, INPUT, PRINT, PRINTQUAD.

COMMENTS: (See Section 4.1.1.)
GDEDFAIL generates a state transition matrix assuming the matrix represents a system having arg components, each failing independently, each distinguishable, and each having the same failure rate. The failures are also assumed to be Poisson, and once a component has failed, it cannot become good again. The state of the system is the state of each of the components. The number of states declared for the phase must be a power of two. The resulting matrix will be argxarg such that the (i,j)th entry will denote the probability that the system is in state j at the end of the phase given it was in state i at the beginning of the phase.

If the system has N components (and hence 2**N states), then the i-th row or column of the matrix represents the state determined as follows: take the binary representation of (2**N)-1-i. Each digit of the binary representation represents the state of a corresponding component in the system, 0 if failed, 1 if not failed. For example, consider a system with two components with the following transition matrix computed by GDEDFAIL:

	11	10	01	00
11	9.98E-1	9.99E-4	9.99E-4	9.99E-7
10	0.00E0	9.99E-1	0.00E0	1.00E-3
01	0.00E0	0.00E0	9.99E-1	1.00E-3
00	0.00E0	0.00E0	0.00E0	1.00E0

where 11 means that both components are not failed, 10 that the first component is not failed but the second one is, 01 that the first component is failed but the second is not, and 00 that both components are failed.

GDEDFAIL is equivalent to GNFAIL when GNFAIL has arg groups of 1 component each.

GDEDFAIL first obtains the length of the phase from the user and checks it for positiveness. For the failure rate of the components; these are assumed to be equal. If this is not between the the failure rate is asked for again, while if confirmed, GDEDFAIL proceeds to compute the transition matrix. This is accomplished by examining each entry in the matrix, determining the number of transitions required for the entry (take the binary representation of the (i,j)th entry as described above, subtract j from i component by component, and count the 1's--if there is a 1 in j which does not appear in the corresponding position of i, then the transition is impossible and so the probability is 0), and determining the probability of making those transitions.

Once the matrix has been computed, it is returned in result.

LISTING:

```
    ∇ Z←GDEDFAIL N;T;LAMBDA;INDEX;I;J;SIZE;SUCCESS;FAIL
[1]   ⍝
[2]   ⍝   ROUTINE FOR GENERATING THE DEDFAIL TYPE P MATRIX
[3]   ⍝   THE MATRIX IS TO HAVE SHAPE N×N
[4]   ⍝
[5]   ⍝  CHECK TO MAKE SURE THIS ROUTINE IS APPROPRIATE
[6]   →((2⊛N)=⌈(2⊛N))/GDIN
[7]   ⍝  PRINT EXPLANATION, NOTE FAILURE, AND EXIT
[8]   PRINT 'THE NUMBER OF STATES IN DEDFAIL MUST BE A POWER OF
      TWO.'
```

```
[9]   ⎕←((PRINTQUAD 'THIS PHASE HAS ');(PRINTQUAD N);PRINTQUAD
      STATES.')
[10]  PRINT ''
[11]  Z←(N,N)⍴¯1
[12]  →0
[13]  ⍝
[14]  ⍝  GET PHASE LENGTH
[15] GDIN:PRINT 'ENTER PHASE LENGTH'
[16]  1 INPUT GD
[17]  ⍝  CHECK COMMAND
[18]  →(1=∧ ∈'COMMAND')/GDIN
[19]  ⍝  CHECK FOR POSITIVE NUMBER
[20]  →(IN>0)/GDSETT
[21]  PRINT 'LENGTH IN TIME.  MUST BE POSITIVE'
[22]  →GDIN
[23]  ⍝
[24]  ⍝  SET T TO PHASE LENGTH AND GET FAILURE RATE
[25] GDSETT:T←IN
[26] GDINL:PRINT 'ENTER COMPONENT FAILURE RATE'
[27]  1 INPUT GD
[28]  ⍝  CHECK COMMAND
[29]  →(1=∧
      ∈'COMMAND')/GDINL
[30]  ⍝  CHECK FOR POSITIVE NUMBER
[31]  →(IN>0)/GDCHECKSIZE
[32]  PRINT 'RATE IN FAILURES PER UNIT TIME.  MUST BE POSITIVE'
[33]  →GDINL
[34]  ⍝ CHECK REASONABLENESS OF FAILURE RATE
[35] GDCHECKSIZE:→((IN≥1E¯10)∧(IN≤0.1))/GDSETL
[36]  ⍝ PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT
[37]  ⎕←PRINTQUAD IN
[38]  →(IN≥0.1)/GDLBIG
[39]  ⎕←PRINTQUAD 'IS SMALL'
[40]  →GDLYESNOIN
[41] GDLBIG:⎕←PRINTQUAD 'IS LARGE'
[42] GDLYESNOIN:PRINT ' FOR A FAILURE RATE.  DO YOU WANT THIS
      VALUE?'
[43]  →(~INYES)/GDINL
[44]  ⍝
[45]  ⍝  SET LAMBDA TO FAILURE RATE AND PERFORM CALCULATIONS
[46] GDSETL:LAMBDA←IN
[47]  ⍝
[48]  ⍝  INITIALIZE THE P MATRIX
[49]  Z←(N,N)⍴0
[50]  ⍝  DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX
[51]  INDEX←⍉((2⊛N)⍴2)ENCODE(N-⍳N)
[52]  ⍝
[53]  ⍝  LOOP THROUGH INDEX TO CREATE P
[54]  ⍝ INITIALIZE LOOPS
[55]  I←1
[56]  J←1
[57]  ⍝  DETERMINE THE NUMBER OF SUCCESS TRANSITIONS
[58] GDLOOP:SIZE←+/INDEX[I;]/INDEX[J;]
```

```
[59]  ⍝ FIND THE SUCCESS AND FAILURE PROBABILITIES
[60]  FAIL←(1-*-LAMBDA×T)*((+/INDEX[I;])-SIZE)
[61]  SUCCESS←*-LAMBDA×T×SIZE
[62]  Z[I;J]←SUCCESS×FAIL×(SIZE≥+
      /INDEX[J;])×(∨/INDEX[I;]≥INDEX[J;])
[63]  ⍝ INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE
[64]  J←J+1
[65]  →(J≤N)/GDLOOP
[66]  ⍝ RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF
      APPROPRIATE
[67]  J←1
[68]  I←I+1
[69]  →(I≤N)/GDLOOP
      ∇
```

GGIVEN

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← arg1 GGIVEN arg2
PURPOSE: To input a given arg1xarg2 matrix from the user.

GLOBAL VARIABLES: IN, GG.

CALLING FUNCTIONS: GENERATEHMATRIX, GENERATEPMATRIX.

FUNCTIONS CALLED: INPUT, PRINT, PRINTQUAD.

COMMENTS: Prints a prompt to enter the matrix and a prompt for each row. Inputs each row of the matrix, checking that each row has arg2 elements,

LISTING:

```
    ∇ Z←M GGIVEN N;ROW
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING CONSTANT MATRIX OF PROBABILITIES
[3]   ⍝   EACH ROW MUST SUM TO ONE.  THE MATRIX IS TO HAVE SHAPE
      M×N
[4]   ⍝
[5]   ⍝  INITIALIZE THE MATRIX
[6]   Z←(M,N)⍴0
[7]   ⍝
[8]   ⍝  INPUT AND CHECK THE MATRIX
[9]   PRINT 'ENTER THE MATRIX, 1 ROW AT A TIME'
[10]  ⍝
[11]  ⍝ INITIALIZE ROW COUNTER
[12]  ROW←1
[13]  PRINT ''
[14] GGIN:⎕←((PRINTQUAD 'ROW ');(PRINTQUAD ROW);PRINTQUAD ':')
[15]  PRINT ''
[16]  ⍝ GET ROW DATA
[17]  N INPUT GG
[18]  ⍝ CHECK FOR COMMAND
[19]  →(1=∧ ∈'COMMAND')/GGIN
[20]  ⍝ CHECK VALIDITY OF INPUT.  FIRST CHECK FOR PROBABILITIES
[21]  →(CHECKPROB IN)/GGIN
[22]  ⍝ CHECK THAT EACH ROW SUMS TO ONE.  IF NOT, ASK AGAIN.
[23]  →((+/IN)=1)/GGINSERT
[24]  PRINT 'THE SUM OF PROBABILITIES IN EACH ROW MUST BE 1'
```

```
[25]  →GGIN
[26]  ⍝
[27]  ⍝  INSERT THE ROW INTO THE MATRIX
[28] GGINSERT:Z[ROW;]←IN
[29]  ⍝
[30]  ⍝  INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[31]  ROW←ROW+1
[32]  →(ROW≤M)/GGIN
[33]  ⍝ ELSE LEAVE
      ∇
```

GIDENTITY

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result ← GIDENTITY arg

PURPOSE: To generate an argxarg identity matrix.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GENERATEHMATRIX, GENERATEPMATRIX.

FUNCTIONS CALLED: None.

COMMENTS: Returns an argxarg identity matrix in result.

LISTING:

```
     ∇ Z←GIDENTITY N
[1]    ⍝
[2]    ⍝  ROUTINE FOR GENERATING AN N×N IDENTITY MATRIX
[3]    ⍝
[4]    ⍝ RETURN THE MATRIX
[5]    Z←(N,N)⍴(1,(N⍴0))
     ∇
```

GNFAIL

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: result <- GNFAIL arg

PURPOSE: TO GENERATE A NFAIL TYPE MATRIX WITH SHAPE ARGXARG.

GLOBAL VARIABLES: IN, GN.

CALLING FUNCTIONS: GENERATEPMATRIX.

FUNCTIONS CALLED: ENCODE, INPUT, PRINT, PRINTQUAD.

COMMENTS: (See Section 4.1.2.)
GNFAIL generates a state transition matrix assuming the matrix represents a system having M groups of K(m) components each, where K is a function of the group m. The components fail independently with the same failure rate and are assumed to have a Poisson distribution. Also, once a component has failed, it cannot become good again. The state of the system is the number of active (non-failed) components in each group. The number of states declared for the phase must be the product of (the number of components in each group plus one). For example, if the system has 3 groups containing respectively 2, 5, and 7 components, then the phase has (2+1)*(5+1)*(7+1)=144 states. The number of states for the phase is contained in arg. The resulting matrix will be argxarg in shape, such that the (i,j)th entry will give the probability that the system is in state j at the end of the phase given it was in state i at the beginning of the phase.

If the system has M groups with K(m) components each, then the i-th row or column of the matrix represents the state determined as follows: take the number system such that the j-th digit is of base K(j)+2, i.e., has maximum value k(j)+1.

This is similar to the APL ENCODE [REPRESENTATION] (T) function. Each digit of a number from the system denotes the state of the corresponding group, e.e., the number of active (non-failed) components in the group. The i-th row then represents (the maximum value of the system)+1-i. For example, consider a system with 2 groups, the first having 2 components and the second 1 component, with the following transition matrix computed by GNFAIL:

	(2,1)	(2,0)	(1,1)	(1,0)	(0,1)	(0,0)
(2,1)	9.704E-1	9.753E-3	1.951E-2	1.960E-4	9.802E-5	9.851E-7
(2,0)	0.000E0	9.802E-1	0.000E0	1.970E-2	0.000E0	9.901E-5
(1,1)	0.000E0	0.000E0	9.802E-1	9.851E-3	9.851E-3	9.901E-5
(1,0)	0.000E0	0.000E0	0.000E0	9.900E-1	0.000E0	9.950E-3
(0,1)	0.000E0	0.000E0	0.000E0	0.000E0	9.900E-1	9.950E-3
(0,0)	0.000E0	0.000E0	0.000E0	0.000E0	0.000E0	1.000E0

where (2,1) means that group 1 has two active components and group 2 has one active component, (2,0) means that group 1 has two active components and group 2 no active components, and so forth. GNFAIL is equivalent to GDEDFAIL when arg groups of one component each are specified.

GNFAIL first obtains the length of the phase from the user and checks it for positiveness. For the failure rate of the components; these are assumed to be equal. If this is not between the the failure rate is asked for again, while if confirmed, GNFAIL proceeds to obtain the number of groups and the number of states in each group. If the number of states given in arg does not equal the product of and the number of groups and number of states in each group is again adked for. This is accomplished by examining each entry in the matrix, determining the number of transitions required for the entry (take the representation of the (i,j)th entry as described above, subtract j from i component by component, and sum the differences--if there is a a value in some position of j which is greater than

the value of the corresponding position of i, then the transition is impossible and so the probability is 0), and determining the probability of making those transitions.

Once the matrix has been computed, it is returned in result.

LISTING:

```
     ∇ Z←GNFAIL N;T;LAMBDA;GROUPS;NUM;INDEX;I;J;COEFF;FAIL;SUCCESS
[1]   ⍝
[2]   ⍝   ROUTINE FOR GENERATING THE NFAIL TYPE P MATRIX
[3]   ⍝   THE MATRIX IS TO HAVE SHAPE N×N
[4]   ⍝
[5]   ⍝  GET PHASE LENGTH
[6]  GNINT:PRINT 'ENTER PHASE LENGTH'
[7]   1 INPUT GN
[8]   ⍝  CHECK COMMAND
[9]   →(1=∧ ∈'COMMAND')/GNINT
[10]  ⍝  CHECK FOR POSITIVE NUMBER
[11]  →(IN>0)/GNSETT
[12]  PRINT 'LENGTH IN TIME.  MUST BE POSITIVE'
[13]  →GNINT
[14]  ⍝
[15]  ⍝  SET T TO PHASE LENGTH AND GET FAILURE RATE
[16] GNSETT:T←IN
[17] GNINL:PRINT 'ENTER COMPONENT FAILURE RATE'
[18]  1 INPUT GN
[19]  ⍝  CHECK COMMAND
[20]  →(1=∧
      ∈'COMMAND')/GNINL
[21]  ⍝  CHECK FOR POSITIVE NUMBER
[22]  →(IN>0)/GNCHECKSIZE
[23]  PRINT 'RATE IN FAILURES PER UNIT TIME.  MUST BE POSITIVE'
[24]  →GNINL
[25]  ⍝ CHECK REASONABLENESS OF FAILURE RATE
[26] GNCHECKSIZE:→((IN≥1E¯10)∧(IN≤0.1))/GNSETL
[27]  ⍝ PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT
[28]  ⎕←PRINTQUAD IN
[29]  →(IN≥0.1)/GNLBIG
[30]  ⎕←PRINTQUAD 'IS SMALL'
[31]  →GNLYESNOIN
[32] GNLBIG:⎕←PRINTQUAD 'IS LARGE'
[33] GNLYESNOIN:PRINT ' FOR A FAILURE RATE.  DO YOU WANT THIS
      VALUE?'
[34]  →(~INYES)/GNINL
[35]  ⍝
[36]  ⍝  SET LAMBDA TO FAILURE RATE AND GET NUMBER OF GROUPS
[37] GNSETL:LAMBDA←IN
[38] GNING:PRINT 'ENTER NUMBER OF GROUPS'
[39]  1 INPUT GN
```

```
[40]  ⍝  CHECK COMMAND
[41]  →(1=∧ ∈'COMMAND')/GNING
[42]  ⍝  CHECK FOR POSITIVE INTEGER
[43]  →(CHECKPOSI IN)/GNING
[44]  ⍝
[45]  ⍝  SET GROUPS TO NUMBER OF GROUPS AND GET COMPONENTS PER
      GROUP
[46]  GROUPS←IN
[47]  GNINN:PRINT 'ENTER NUMBER OF COMPONENTS PER GROUP (SPACE
      BETWEEN EACH NUMBER):'
[48]  GROUPS INPUT GN
[49]  ⍝  CHECK COMMAND
[50]  →(1=∧
      ∈'COMMAND')/GNINN
[51]  ⍝  CHECK FOR POSITIVE INTEGER
[52]  →(CHECKPOSI IN)/GNINN
[53]  ⍝  CHECK FOR THE PROPER NUMBER OF STATES
[54]  →(N=×/IN+1)/GNCALC
[55]  ⎕←((PRINTQUAD 'THERE ARE ');(PRINTQUAD N);PRINTQUAD ' STATES
      IN THIS PHASE.  THE PRODUCT OF [EACH COMPONENT')
[56]  PRINT ''
[57]  PRINT 'NUMBER PLUS 1] MUST BE THE NUMBER OF STATES.'
[58]  PRINT 'HOW MANY GROUPS'
[59]  →GNING
[60]  ⍝
[61]  ⍝  SET NUM TO THE NUMBER OF POSITIONS FOR THE COMPONENTS IN
      THE GROUPS AND CALCULATE THE P MATRIX
[62] GNCALC:NUM←IN+1
[63]  ⍝
[64]  ⍝  INITIALIZE THE P MATRIX
[65]  Z←(N,N)⍴0
[66]  ⍝  DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX
[67]  INDEX←⍉(NUM)ENCODE(N-⍳N)
[71]  ⍝
[72]  ⍝  LOOP THROUGH INDEX TO CREATE P
[73]  ⍝ INITIALIZE LOOPS
[74] I←1
[75]  J←1
[76]  ⍝  FIRST DETERMINE THE PROPER COEFFICIENT FOR THE TERM UNDER
      CONSIDERATION
[77] GNLOOP:COEFF←×(INDEX[J;]!INDEX[I;])
[78]  ⍝ MULTIPLY THE COEFFICIENT WITH THE PROPER EXPONENTIALS
[79]  FAIL←(1-*-LAMBDA×T)*(+/INDEX[I;])-+/INDEX[J;]
[80]  SUCCESS←*-LAMBDA×T×(+/INDEX[J;])
[81]  Z[I;J]←COEFF×FAIL×SUCCESS
[82]  ⍝ INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE
[83]  J←J+1
[84]  →(J≤N)/GNLOOP
[85]  ⍝ RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF
      APPROPRIATE
[86]  J←1
[87]  I←I+1
```

7. GNFAIL METAPHOR FUNCTION DESCRIPTION

```
[88]  →(I≤N)/GNLOOP
      ∇
```

GAVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GAVINFO

PURPOSE: To generate the GETALTERVECTOR HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETALTERVECTOR function. If the user wishes references, GAVINFO also prints a small selection of references.

LISTING:

```
    ∇ GAVINFO
[1]   ⍝
[2]   ⍝  ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE DISPLAYED
[3]   ⍝
[4]   PRINT 'PLACE AN X BELOW EACH ENTRY IN THE GIVEN LIST WHICH YOU
      WANT TO '
[5]   PRINT 'CHANGE.   THE ABBREVIATIONS ARE AS FOLLOWS:'
[6]   PRINT ''
[7]   PRINT '                P     THE INTRAPHASE STATE TRANSITION
      (P) MATRICES'
[8]   PRINT ''
[9]   PRINT '                H     THE INTERPHASE STATE TRANSITION
      (H) MATRICES'
[10]  PRINT ''
[11]  PRINT '     CONST.BAS.VARS     THE NUMBER OF TIME+INVARIANT
      BASIC VARIABLES AND'
[12]  PRINT '                        THEIR ASSOCIATED
      PROBABILITIES'
[13]  PRINT ''
[14]  PRINT '     ALL.ACC.LEVELS     USING THE PRESENT H AND P
      MATRICES AND THE'
[15]  PRINT '                        PRESENT TIME-INVARIANT BASIC
      VARIABLE INFORMATION,'
[16]  PRINT '                        DETERMINE THE PERFORMABILITY OF
      THE SYSTEM.'
[17]  PRINT '                        METAPHOR WILL ASK FOR THE
```

```
      APPROPRIATE INFORMATION'
[18]  PRINT '                              REGARDING THE ACCOMPLISHMENT
      LEVELS.'
[19]  PRINT ''
[20]  PRINT '  PRESENT.ACC.LEVEL           ALTER ONLY THE ACCOMPLISHMENT
      LEVEL '
[21]  PRINT '                              PRESENTLY UNDER
      CONSIDERATION.'
[22]  PRINT ''
[23]  PRINT '                    I         INITIAL VECTOR'
[24]  PRINT ''
[25]  PRINT '                    G         CHARACTERISTIC MATRICES'
[26]  PRINT ''
[27]  PRINT '                    F         CHARACTERISTIC VECTOR'
[28]  PRINT '                              (AT PRESENT, THIS ALTER
      OPERATION IS NOT EXECUTABLE.)'
[29]  PRINT ''
[30]  PRINT '                    V         VECTOR CHARACTERIZING THE TIME+
      INVARIANT BASIC VARIABLES'
[31]  PRINT ''
[32]  PRINT '       NUM.TRAJ.SETS          ALTER THE NUMBER OF TRAJECTORY
      SETS DESCRIBING'
[33]  PRINT '                              THE ACCOMPLISHMENT LEVEL UNDER
      CONSIDERATION'
[34]  PRINT ''
[35]  PRINT 'IF AN ITEM IS UNDEFINED WHEN AN ALTERATION IS
      REQUESTED, AN ERROR '
[36]  PRINT 'MESSAGE WILL BE PRINTED AND THAT ALTERATION
      SUPPRESED.  MORE THAN'
[37]  PRINT 'ONE ITEM MAY BE CHANGED WITH A SINGLE ALTER
      COMMAND.  '
[38]  PRINT 'EXAMPLE:'
[39]  PRINT ''
[40]  PRINT
      'P  H  CONST.BAS.VARS  ALL.ACC.LEVELS  PRESENT.ACC.LEVEL  I  G
      V NUM.TRAJ.SETS'
[41]  PRINT 'X          X                  X          '
[42]  PRINT ''
[43]  PRINT 'THIS INFORMS METAPHOR THAT THE P AND H MATRICES ARE TO
      BE CHANGED AND THAT THE  '
[44]  PRINT 'PERFORMABILITY IS TO BE CALCULATED.  IF YOU WISH TO
      CHANGE THE NUMBER OF PHASES '
[45]  PRINT 'OR ASSOCIATED STATES, TYPE END AND BEGIN METAPHOR
      AGAIN'
[46]  PRINT ''
[47]  PRINT 'DO YOU WANT REFERENCES?'
[48]  ⍝ LEAVE IF NOT
[49]  →(~INYES)/0
[50]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[51]  PRINT ''
[52]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
```

```
[53]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[54]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[55]  PRINT ''
[56]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[57]  PRINT ''
[58]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[59]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[60]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[61]  PRINT ''
[62]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[63]  PRINT ''
[64]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[65]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[66]  PRINT ''
      ∇
```

GBVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GBVINFO

PURPOSE: To generate the GETBASICVARIABLES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETBASICVARIABLES function. If the user wishes references, GBVINFO also prints a small selection of references.

LISTING:

```
    ∇ GBVINFO
[1]   ⍝
[2]   ⍝ ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF BASIC
      VARIABLES
[3]   ⍝
[4]   PRINT 'ENTER THE PROBABILITIES OF THE BASIC VARIABLES WHOSE
      PROBABILITIES REMAIN CONSTANT'
[5]   PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF
      TIME-INVARIANT BASIC VARIABLES.)'
[6]   PRINT 'THE PROBABILITIES SHOULD BE ENTERED AS A ROW OF
      POSITIVE NUMBERS'
[7]   PRINT 'BETWEEN ZERO AND ONE, INCLUSIVE.  THE NUMBERS SHOULD
      BE'
[8]   PRINT 'SEPARATED BY SPACES AND÷OR COMMAS.  THE ORDER OF THE
      NUMBERS SHOULD'
[9]   PRINT 'CORRESPOND TO THE ORDER OF THE TIME+INVARIANT BASIC
      VARIABLE VECTORS'
[10]  PRINT 'WHICH WILL BE ASKED FOR LATER.'
[11]  PRINT 'EXAMPLE:'
[12]  PRINT '      .2,  0.3  .4,0.1'
[13]  PRINT 'THIS INFORMS METAPHOR THAT THE PROBABILITIES OF THE
      FOUR TIME+INVARIANT'
[14]  PRINT 'BASIC VARIABLES ARE 0.2, 0.3, 0.4, AND 0.1
      RESPECTIVELY.  THE NUMBER'
[15]  PRINT 'OF TIME-INVARIANT VARIABLES DECLARED EARLIER MUST HAVE
      BEEN FOUR OR AN'
```

```
[16]  PRINT 'ERROR MESSAGE WILL RESULT.'
[17]  PRINT ''
[18]  PRINT 'DO YOU WANT REFERENCES?'
[19]  ⍝ LEAVE IF NOT
[20]  →(~INYES)/0
[21]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[22]  PRINT ''
[23]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[24]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[25]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[26]  PRINT ''
[27]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[28]  PRINT ''
[29]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[30]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[31]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[32]  PRINT ''
[33]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[34]  PRINT ''
[35]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[36]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[37]  PRINT ''
      ∇
```

GDINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GDINFO

PURPOSE: To generate the GDEDFAIL HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GDEDFAIL function. If the user wishes references, GDINFO also prints a small selection of references.

LISTING:

```
     ∇ GDINFO
[1]    ⍝
[2]    ⍝  ROUTINE GIVING HELP ON DEDFAIL TYPE P MATRIX
[3]    ⍝
[4]    PRINT 'METAPHOR WILL GENERATE A P MATRIX ASSUMING THE MATRIX
       REPRESENTS '
[5]    PRINT 'A SYSTEM HAVING N COMPONENTS, EACH FAILING
       INDEPENDENTLY AND EACH '
[6]    PRINT 'DISTINGUISHABLE.  THE FAILURES ARE ALSO ASSUMED TO BE
       '
[7]    PRINT 'POISSON, AND ONCE A COMPONENT HAS FAILED, IT CANNOT'
[8]    PRINT 'BECOME GOOD AGAIN.'
[9]    PRINT 'THE STATE OF THE SYSTEM IS THE STATE OF EACH OF'
[10]   PRINT 'THE COMPONENTS.  THE NUMBER OF STATES DFCLARED FOR THE
       PHASE MUST BE '
[11]   PRINT 'A POWER OF TWO.  YOU WILL BE ASKED THE LENGTH OF THE
       PHASE; ENTER A'
[12]   PRINT 'SINGLE POSITIVE INTEGER.  NEXT YOU WILL BE PROMPTED FOR
       THE FAILURE'
[13]   PRINT 'RATE OF THE COMPONENTS.  AGAIN ENTER A SINGLE POSITIVE
       NUMBER.  IF '
[14]   PRINT 'THIS NUMBER IS NOT BETWEEN 1E¯1 AND 1E¯10, YOU WILL BE
       ASKED FOR CONFIRMATION.'
[15]   PRINT ''
[16]   PRINT 'DO YOU WANT REFERENCES?'
[17]   ⍝ LEAVE IF NOT
[18]   →(~INYES)/0
```

```
[19]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[20]  PRINT ''
[21]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[22]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[23]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[24]  PRINT ''
[25]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[26]  PRINT ''
[27]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[28]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[29]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[30]  PRINT ''
[31]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[32]  PRINT ''
[33]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[34]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[35]  PRINT ''
      ∇
```

GDVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GDVINFO

PURPOSE: To generate the GETDATAVECTOR HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETDATAVECTOR function. If the user wishes references, GDVINFO also prints a small selection of references.

LISTING:

```
    ∇ GDVINFO
[1]   ⍝
[2]   ⍝  ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE ALTERED
[3]   ⍝
[4]   PRINT 'ENTER AN X BELOW EACH ITEM IN THE GIVEN LIST WHICH YOU
      WANT TO '
[5]   PRINT 'DISPLAY.  THE ABBREVIATIONS ARE AS FOLLOWS:'
[6]   PRINT ''
[7]   PRINT '          NUM.PHASES      THE NUMBER OF PHASES'
[8]   PRINT ''
[9]   PRINT '          NUM.STATES      THE NUMBER OF STATES'
[10]  PRINT ''
[11]  PRINT '                   P      THE INTRAPHASE TRANSITION (P)
      MATRICES'
[12]  PRINT ''
[13]  PRINT ' NUM.CONST.BAS.VARS       THE NUMBER OF TIME-INVARIANT
      BASIC VARIABLES'
[14]  PRINT ''
[15]  PRINT 'PROB.CONST.BAS.VARS       THE PROBABILITIES OF EACH OF
      THE TIME-INVARIANT'
[16]  PRINT '                          BASIC VARIABLES'
[17]  PRINT ''
[18]  PRINT '      NUM.ACC.LEVELS      THE NUMBER OF ACCOMPLISHMENT
      LEVELS'
[19]  PRINT ''
[20]  PRINT '       NUM.TRAJ.SETS      THE NUMBER OF TRAJECTORY SETS
```

```
      ASSOCIATED WITH'
[21]  PRINT '                          THE ACCOMPLISHMENT LEVEL UNDER
      CONSIDERATION'
[22]  PRINT ''
[23]  PRINT '                    I     THE INITIAL VECTOR FOR THE
      TRAJECTORY SET'
[24]  PRINT '                          UNDER CONSIDERATION'
[25]  PRINT ''
[26]  PRINT '                    G     THE CHARACTERISTIC MATRICES FOR
      THE TRAJECTORY'
[27]  PRINT '                          SET UNDER CONSIDERATION'
[28]  PRINT ''
[29]  PRINT '                    F     THE CHARACTERISTIC VECTOR FOR
      THE TRAJECTORY'
[30]  PRINT '                          SET UNDER CONSIDERATION'
[31]  PRINT ''
[32]  PRINT '                    V     THE VECTOR CHARACTERIZING THE
      TIME+INVARIANT '
[33]  PRINT '                          BASIC VARIABLES FOR THE
      TRAJECTORY SET '
[34]  PRINT '                          UNDER CONSIDERATION'
[35]  PRINT ''
[36]  PRINT '               PERF       THE PERFORMABILITY'
[37]  PRINT ''
[38]  PRINT 'IF AN ITEM IS UNDEFINED WHEN A DISPLAY IS REQUESTED, AN
      ERROR MESSAGE'
[39]  PRINT 'WILL BE PRINTED AND THAT DISPLAY WILL BE
      SUPPRESED.  MORE THAN ONE ITEM '
[40]  PRINT 'MAY BE DISPLAYED WITH A SINGLE DATA COMMAND.'
[41]  PRINT 'EXAMPLE:'
[42]  PRINT ''
[43]  PRINT
      'NUM.PHASES  NUM.STATES  P  H  NUM.CONST.BAS.VARS  PROB.CONST.B
[44]  PRINT 'X
      X                                          X                '
[45]  PRINT 'NUM.ACC.LEVELS  NUM.TRAJ.SETS  I  G  F  V  PERF'
[46]  PRINT
      '  X              X                          X
      '
[47]  PRINT ''
[48]  PRINT 'THIS INFORMS METAPHOR THAT THE NUMBER OF PHASES,
      STATES, AND ACCOMPLISHMENT'
[49]  PRINT 'LEVELS AS WELL AS THE PROBABILITIES OF THE TIME+
      INVARIANT BASIC VARIABLES'
[50]  PRINT 'AND THE PERFORMABILITY ARE TO BE DISPLAYED.'
[51]  PRINT ''
[52]  PRINT 'DO YOU WANT REFERENCES?'
[53]  ⍝ LEAVE IF NOT
[54]  →(~INYES)/0
[55]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[56]  PRINT ''
[57]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
```

```
      THE '
[58]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[59]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[60]  PRINT ''
[61]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[62]  PRINT ''
[63]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[64]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[65]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[66]  PRINT ''
[67]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[68]  PRINT ''
[69]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[70]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[71]  PRINT ''
      ∇
```

GFVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GFVINFO

PURPOSE: To generate the GETFVECTOR HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETFVECTOR function. If the user wishes references, GFVINFO also prints a small selection of references.

LISTING:

```
    ∇ GFVINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING THE F VECTORS
[3]   ⍝
[4]   PRINT 'ENTER THE CHARACTERISTIC (F) VECTOR FOR THE TRAJECTORY
      SET UNDER CONSIDERATION.  '
[5]   PRINT 'EACH ENTRY SHOULD BE EITHER 0 OR 1 AND SHOULD BE
      SEPARATED FROM THE OTHER ENTRIES '
[6]   PRINT 'BY SPACES AND/OR COMMAS. THE NUMBER OF ENTRIES SHOULD
      BE THE SAME AS THE NUMBER'
[7]   PRINT 'OF STATES OF THE FINAL PHASE MODEL.  ALSO THE ORDER OF
      THE ENTRIES SHOULD'
[8]   PRINT 'CORRESPOND TO THE ORDER OF THE STATES AS CONSIDERED
      ELSEWHERE IN THE'
[9]   PRINT 'METAPHOR PACKAGE FOR THE FINAL PHASE.'
[10]  PRINT 'EXAMPLE:'
[11]  PRINT '   1 0, 1 ,0'
[12]  PRINT 'THIS INFORMS METAPHOR THAT THE CHARACTERISTIC VECTOR
      FOR THIS TRAJECTORY '
[13]  PRINT 'SET IS'
[14]  PRINT '  1'
[15]  PRINT '  0'
[16]  PRINT '  1'
[17]  PRINT '  0 .'
[18]  PRINT ''
[19]  PRINT 'DO YOU WANT REFERENCES?'
```

```
[20]  ⍝ LEAVE IF NOT
[21]  →(~INYES)/0
[22]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[23]  PRINT ''
[24]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[25]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[26]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[27]  PRINT ''
[28]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[29]  PRINT ''
[30]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[31]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[32]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[33]  PRINT ''
[34]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[35]  PRINT ''
[36]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[37]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[38]  PRINT ''
      ∇
```

GGINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GGINFO

PURPOSE: To generate the GGIVEN HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GGIVEN function. If the user wishes references, GGINFO also prints a small selection of references.

LISTING:

```
    ∇ GGINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING GIVEN P MATRICES
[3]   ⍝
[4]   PRINT 'ENTER AN M×N ARRAY, ONE ROW AT A TIME.  EACH ENTRY
      MUST'
[5]   PRINT 'BE BETWEEN 0 AND 1 INCLUSIVE AND THE ENTRIES OF EACH
      ROW MUST SUM'
[6]   PRINT 'TO ONE.  ENTER EACH ROW AS A SERIES OF N NUMBERS WITH
      SPACES AND÷OR'
[7]   PRINT 'COMMAS BETWEEN EACH.'
[8]   PRINT 'EXAMPLE:'
[9]   PRINT '     .25 0.5,.1 0.15'
[10]  PRINT 'HERE, THE MATRIX HAS FOUR ENTRIES PER ROW.'
[11]  PRINT ''
[12]  PRINT 'DO YOU WANT REFERENCES?'
[13]  ⍝ LEAVE IF NOT
[14]  →(~INYES)/0
[15]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[16]  PRINT ''
[17]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[18]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[19]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[20]  PRINT ''
```

```
[21]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[22]  PRINT ''
[23]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[24]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[25]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[26]  PRINT ''
[27]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[28]  PRINT ''
[29]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[30]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[31]  PRINT ''
      ∇
```

GGMINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GGMINFO

PURPOSE: To generate the GETGMATRICES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETGMATRICES function. If the user wishes references, GGMINFO also prints a small selection of references.

LISTING:

```
     ∇ GGMINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING G MATRICES
[3]   ⍝
[4]   PRINT 'ENTER THE DIAGONAL OF THE CHARACTERISTIC (G) MATRIX FOR
      THE PHASE UNDER CONSIDERATION.'
[5]   PRINT 'EACH ENTRY SHOULD BE EITHER 0 OR 1 AND SHOULD BE
      SEPARATED FROM THE OTHER '
[6]   PRINT 'ENTRIES BY SPACES AND/OR COMMAS. THE NUMBER OF ENTRIES
      SHOULD BE THE SAME'
[7]   PRINT 'AS THE NUMBER OF STATES FOR THE PHASE. ALSO THE ORDER
      OF THE ENTRIES SHOULD'
[8]   PRINT 'CORRESPOND TO THE ORDER OF THE STATES AS CONSIDERED
      ELSEWHERE WITHIN THE'
[9]   PRINT 'METAPHOR PACKAGE FOR THE PHASE.'
[10]  PRINT 'EXAMPLE:'
[11]  PRINT '    0 0, 1,1  1'
[12]  PRINT 'THIS INFORMS METAPHOR THAT THE CHARACTERISTIC MATRIX
      FOR THIS PHASE IS'
[13]  PRINT '           0 0 0 0 0'
[14]  PRINT '           0 0 0 0 0'
[15]  PRINT '           0 0 1 0 0'
[16]  PRINT '           0 0 0 1 0'
[17]  PRINT '           0 0 0 0 1
[18]  PRINT ''
[19]  PRINT 'DO YOU WANT REFERENCES?'
```

```
[20]  ⍝ LEAVE IF NOT
[21]  →(~INYES)/0
[22]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[23]  PRINT ''
[24]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[25]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[26]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[27]  PRINT ''
[28]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[29]  PRINT ''
[30]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[31]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[32]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[33]  PRINT ''
[34]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[35]  PRINT ''
[36]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[37]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[38]  PRINT ''
      ∇
```

GHMINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GHMINFO

PURPOSE: To generate the GETHMATRICES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETHMATRICES function. If the user wishes references, GHMINFO also prints a small selection of references.

LISTING:

```
    ∇ GHMINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING GIVEN H MATRICES
[3]   ⍝
[4]   PRINT 'TYPE ONE OF:   GIVEN, IDENTITY'
[5]   PRINT 'DO YOU WANT MORE HELP?'
[6]   ⍝LEAVE IF NO MORE HELP WANTED
[7]   →(~INYES)/0
[8]   PRINT 'ENTER ONE OF THE FOLLOWING TYPES FOR THE INTERPHASE
      TRANSITION (H) MATRIX:'
[9]   PRINT ''
[10]  PRINT '     GIVEN      YOU WILL INPUT AN H MATRIX, ONE ROW AT
      A TIME.'
[11]  PRINT ''
[12]  PRINT '     IDENTITY  METAPHOR WILL GENERATE AN H MATRIX
      ASSUMING THE MATRIX'
[13]  PRINT '               REPRESENTS A SYSTEM IN WHICH THERE IS NO
      STATE CHANGE'
[14]  PRINT '               BETWEEN PHASES.  THUS, METAPHOR
      GENERATES AN IDENTITY MATRIX.'
[15]  PRINT ''
[16]  PRINT ''
[17]  PRINT 'DO YOU WANT REFERENCES?'
[18]  ⍝ LEAVE IF NOT
[19]  →(~INYES)/0
[20]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
```

```
      ANALYSIS, SEE'
[21]  PRINT ''
[22]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[23]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[24]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[25]  PRINT ''
[26]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[27]  PRINT ''
[28]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[29]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[30]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[31]  PRINT ''
[32]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[33]  PRINT ''
[34]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[35]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[36]  PRINT ''
      ∇
```

GIVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GIVINFO

PURPOSE: To generate the GETIVECTOR HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETIVECTOR function. If the user wishes references, GIVINFO also prints a small selection of references.

LISTING:

```
    ∇ GIVINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING I VECTORS
[3]   ⍝
[4]   PRINT 'ENTER THE INITIAL PROBABILITY DISTRIBUTION FOR THIS
      TRAJECTORY.'
[5]   PRINT 'TYPE A PROBABILITY BETWEEN ZERO AND ONE INCLUSIVE
      CORRESPONDING TO EACH'
[6]   PRINT 'STATE''S INITIAL PROBABILITY. SEPARATE EACH NUMBER
      WITH SPACES AND÷OR COMMAS.'
[7]   PRINT 'THE ORDER OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER
      OF THE INITIAL STATES.'
[8]   PRINT 'THE NUMBER OF ENTRIES SHOULD BE THE SAME AS THE NUMBER
      OF STATES'
[9]   PRINT 'IN THE FIRST PHASE MODEL.'
[10]  PRINT 'EXAMPLE:'
[11]  PRINT '          .3  0.5,.2'
[12]  PRINT 'THIS INFORMS METAPHOR THAT FOR THE TRAJECTORY SET UNDER
      CONSIDERATION, THE'
[13]  PRINT 'PROBABILITY THE SYSTEM BEGINS IN THE FIRST STATE OF
      PHASE 1 IS 0.2, FOR THE'
[14]  PRINT 'SECOND STATE, THE PROBABILITY IS 0.5, AND FOR THE THIRD
      STATE, THE PROBABILITY'
[15]  PRINT 'IS 0.2.'
[16]  PRINT ''
[17]  PRINT 'DO YOU WANT REFERENCES?'
```

```
[18]  ⍝ LEAVE IF NOT
[19]  →(~INYES)/0
[20]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[21]  PRINT ''
[22]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[23]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[24]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[25]  PRINT ''
[26]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[27]  PRINT ''
[28]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[29]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[30]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[31]  PRINT ''
[32]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[33]  PRINT ''
[34]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[35]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[36]  PRINT ''
      ∇
```

GNINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GNINFO

PURPOSE: To generate the GNFAIL HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GNFAIL function. If the user wishes references, GNINFO also prints a small selection of references.

LISTING:

```
∇ GNINFO
[1]  ⍝
[2]  ⍝  ROUTINE GIVING HELP ON NFAIL TYPE P MATRIX
[3]  ⍝
[4]  PRINT 'METAPHOR WILL GENERATE A P MATRIX ASSUMING THE MATRIX
     REPRESENTS A'
[5]  PRINT 'SYSTEM HAVING M GROUPS OF K(N) COMPONENTS EACH, WHERE K
     IS A '
[6]  PRINT 'FUNCTION OF THE GROUP.  THE COMPONENTS FAIL
     INDEPENDENTLY'
[7]  PRINT 'AND ARE ASSUMED TO HAVE A POISSON DISTRIBUTION.  ALSO,
     ONCE'
[8]  PRINT 'A COMPONENT HAS FAILED, IT CANNOT BECOME GOOD AGAIN.'
[9]  PRINT 'THE STATE OF THE SYSTEM IS THE NUMBER OF ACTIVE
     COMPONENTS'
[10] PRINT 'IN EACH GROUP.  THE NUMBER OF STATES DECLARED'
[11] PRINT 'FOR THE PHASE MUST BE THE PRODUCT OF [THE NUMBER OF
     COMPONENTS '
[12] PRINT 'IN EACH GROUP PLUS ONE].  FOR EXAMPLE, IF THE SYSTEM
     HAS 3 GROUPS'
[13] PRINT 'CONTAINING RESPECTIVELY 2, 5, AND 7 COMPONENTS, THEN
     THE PHASE HAS'
[14] PRINT '(2+1)×(5+1)×(7+1)←144 STATES. '
[15] PRINT ''
[16] PRINT 'YOU WILL BE ASKED THE LENGTH OF THE PHASE; ENTER A
     SINGLE POSITIVE INTEGER.'
[17] PRINT 'NEXT YOU WILL BE PROMPTED FOR THE FAILURE RATE OF THE
```

```
      COMPONENTS.  '
[18]  PRINT 'AGAIN ENTER A SINGLE POSITIVE NUMBER.  IF THIS NUMBER
      IS NOT BETWEEN'
[19]  PRINT '1E¯1 AND 1E¯10, YOU WILL BE ASKED FOR
      CONFIRMATION.  YOU WILL THEN '
[20]  PRINT 'BE ASKED THE NUMBER OF GROUPS; ENTER THIS AS A SINGLE
      POSITIVE INTEGER.'
[21]  PRINT 'FINALLY, METAPHOR WILL REQUEST THE NUMBER OF COMPONENTS
      IN EACH GROUP.'
[22]  PRINT 'THIS SHOULD BE INPUT AS A ROW OF POSITIVE INTEGERS
      SEPARATED BY '
[23]  PRINT 'SPACES OR COMMAS.'
[24]  PRINT ''
[25]  PRINT 'DO YOU WANT REFERENCES?'
[26]  ⍝ LEAVE IF NOT
[27]  →(~INYES)/0
[28]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[29]  PRINT ''
[30]  PRINT '    J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[31]  PRINT '    EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[32]  PRINT '    NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[33]  PRINT ''
[34]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[35]  PRINT ''
[36]  PRINT '    J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[37]  PRINT '    EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[38]  PRINT '    NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[39]  PRINT ''
[40]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[41]  PRINT ''
[42]  PRINT '    S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[43]  PRINT '    ASSOCIATES,INC., CHICAGO, 1972.'
[44]  PRINT ''
      ∇
```

GNAINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GNAINFO

PURPOSE: To generate the GETNUMACCLEV HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETNUMACCLEV function. If the user wishes references, GNAINFO also prints a small selection of references.

LISTING:

```
     ∇ GNAINFO
[1]    ⍝
[2]    ⍝ ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF
       ACCOMPLISHMENT LEVELS
[3]    ⍝
[4]    PRINT 'ENTER THE NUMBER OF ACCOMPLISHMENT LEVELS FOR THIS
       MODEL AS A SINGLE POSITIVE INTEGER.'
[5]    PRINT 'EXAMPLE:'
[6]    PRINT '          5'
[7]    PRINT 'THIS INDICATES TO METAPHOR THAT THE MODEL IT IS
       EVALUATING HAS 5 ACCOMPLISHMENT LEVELS.'
[8]    PRINT ''
[9]    PRINT 'DO YOU WANT REFERENCES?'
[10]   ⍝ LEAVE IF NOT
[11]   →(~INYES)/0
[12]   PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
       ANALYSIS, SEE'
[13]   PRINT ''
[14]   PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
       THE'
[15]   PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
       GRANT'
[16]   PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[17]   PRINT ''
[18]   PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[19]   PRINT ''
```

```
[20]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[21]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[22]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[23]  PRINT ''
[24]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[25]  PRINT ''
[26]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[27]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[28]  PRINT ''
      ∇
```

GNBVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GNBVINFO

PURPOSE: To generate the GETNUMBASICVARIABLES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETNUMBASICVARIABLES function. If the user wishes references, GNBVINFO also prints a small selection of references.

LISTING:

```
    ∇ GNBVINFO
[1]   ⍝
[2]   ⍝  ROUTINE GIVING HELP ON INPUTTING BASIC VARIABLES
[3]   ⍝
[4]   PRINT 'ENTER THE NUMBER OF BASIC VARIABLES WHOSE PROBABILITIES
      REMAIN TIME-INVARIANT'
[5]   PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF
      TIME-INVARIANT BASIC VARIABLES.)'
[6]   PRINT 'THE NUMBER SHOULD BE A SINGLE POSITIVE INTEGER.'
[7]   PRINT 'EXAMPLE:'
[8]   PRINT '      2'
[9]   PRINT 'THIS INFORMS METAPHOR THAT TWO TIME-INVARIANT BASIC
      VARIABLES ARE CONSIDERED IN THE'
[10]  PRINT 'MODEL.'
[11]  PRINT ''
[12]  PRINT 'DO YOU WANT REFERENCES?'
[13]  ⍝ LEAVE IF NOT
[14]  →(~INYES)/0
[15]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[16]  PRINT ''
[17]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '.
[18]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[19]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
```

```
[20]  PRINT ''
[21]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[22]  PRINT ''
[23]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[24]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[25]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[26]  PRINT ''
[27]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[28]  PRINT ''
[29]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[30]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[31]  PRINT ''
      ∇
```

GNPINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GNPINFO

PURPOSE: To generate the GETNUMPHASES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETNUMPHASES function. If the user wishes references, GNPINFO also prints a small selection of references.

LISTING:

```
     ∇ GNPINFO
[1]    ⍝
[2]    ⍝  ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF PHASES
[3]    ⍝
[4]    PRINT 'ENTER THE NUMBER OF PHASES IN THE FINITE PHASE MODEL AS
       A SINGLE'
[5]    PRINT 'POSITIVE INTEGER.'
[6]    PRINT 'EXAMPLE:'
[7]    PRINT '      3'
[8]    PRINT 'THIS INDICATES TO METAPHOR THAT THE MODEL TO BE
       EVALUATED HAS 3 PHASES'
[9]    PRINT 'DO YOU WANT REFERENCES?'
[10]   ⍝ LEAVE IF NOT
[11]   →(~INYES)/0
[12]   PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
       ANALYSIS, SEE'
[13]   PRINT ''
[14]   PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
       THE '
[15]   PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
       GRANT'
[16]   PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[17]   PRINT ''
[18]   PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[19]   PRINT ''
[20]   PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
```

```
      THE '
[21]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[22]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[23]  PRINT ''
[24]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[25]  PRINT ''
[26]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[27]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[28]  PRINT ''
    ∇
```

GNTSINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GNTSINFO

PURPOSE: To generate the GETNUMTRAJSETS HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETNUMTRAJSETS function. If the user wishes references, GNJSINFO also prints a small selection of references.

LISTING:

```
     ∇ GNTSINFO
[1]    ⍝
[2]    ⍝   ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF
       TRAJECTORIES IN A GIVEN ACCOMPLISHMENT LEVEL
[3]    ⍝
[4]    PRINT 'ENTER THE NUMBER OF TRAJECTORY SETS ASSOCIATED WITH
       THIS ACCOMPLISHMENT'
[5]    PRINT 'LEVEL AS A SINGLE POSITIVE INTEGER.'
[6]    PRINT 'EXAMPLE:'
[7]    PRINT '              4'
[8]    PRINT 'THIS INFORMS METAPHOR THAT THE ACCOMPLISHMENT LEVEL HAS
       FOUR TRAJECTORY'
[9]    PRINT 'SETS DESCRIBING IT.'
[10]   PRINT ''
[11]   PRINT 'DO YOU WANT REFERENCES?'
[12]   ⍝ LEAVE IF NOT
[13]   →(~INYES)/0
[14]   PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
       ANALYSIS, SEE'
[15]   PRINT ''
[16]   PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
       THE '
[17]   PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
       GRANT'
[18]   PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[19]   PRINT ''
```

```
[20]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[21]  PRINT ''
[22]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[23]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[24]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[25]  PRINT ''
[26]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[27]  PRINT ''
[28]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[29]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[30]  PRINT ''
      ∇
```

GPMINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GPMINFO

PURPOSE: To generate the GETPMATRICES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETPMATRICES function. If the user wishes references, GPMINFO also prints a small selection of references.

LISTING:

```
     ∇ GPMINFO
[1]    ⍝
[2]    ⍝   ROUTINE GIVING HELP ON INPUTTING P MATRICES
[3]    ⍝
[4]    PRINT 'TYPE ONE OF:   GIVEN, DEDFAIL, NFAIL, IDENTITY'
[5]    PRINT 'DO YOU WANT MORE HELP?'
[6]    ⍝LEAVE IF NO MORE HELP WANTED
[7]    →(~INYES)/0
[8]    PRINT 'ENTER ONE OF THE FOLLOWING TYPES FOR THE STATE
       TRANSITION (P) MATRIX:'
[9]    PRINT ''
[10]   PRINT '     GIVEN      YOU WILL INPUT A P MATRIX, ONE ROW AT A
       TIME.'
[11]   PRINT ''
[12]   PRINT '     DEDFAIL    METAPHOR WILL GENERATE A P MATRIX
       ASSUMING THE MATRIX'
[13]   PRINT '                REPRESENTS A SYSTEM HAVING N COMPONENTS,
       EACH FAILING'
[14]   PRINT '                INDEPENDENTLY AND EACH
       DISTINGUISHABLE.  THE STATE'
[15]   PRINT '                OF THE SYSTEM IS THE STATE OF EACH OF
       THE COMPONENTS.'
[16]   PRINT '                YOU WILL BE ASKED THE LENGTH OF THE
       PHASE AND THE FAILURE'
[17]   PRINT '                RATE OF THE COMPONENTS.'
[18]   PRINT ''
```

```
[19]  PRINT '     NFAIL     METAPHOR WILL GENERATE A P MATRIX
      ASSUMING THE MATRIX'
[20]  PRINT '               REPRESENTS A SYSTEM HAVING M GROUPS OF
      K(M) COMPONENTS EACH.'
[21]  PRINT '               THE COMPONENTS FAIL INDEPENDENTLY AND
      THE STATE OF THE'
[22]  PRINT '               SYSTEM IS THE NUMBER OF ACTIVE (NON+
      FAILED) COMPONENTS'
[23]  PRINT '               IN EACH GROUP.  YOU WILL BE ASKED THE
      NUMBER OF GROUPS,'
[24]  PRINT '               THE NUMBER OF COMPONENTS IN EACH GROUP,
      THE LENGTH'
[25]  PRINT ' .             OF THE PHASE, AND THE FAILURE RATE OF
      THE COMPONENTS.'
[26]  PRINT ''
[27]  PRINT '     IDENTITY  METAPHOR WILL GENERATE A P MATRIX
      ASSUMING THE MATRIX'
[28]  PRINT '               REPRESENTS A SYSTEM IN WHICH THERE IS NO
      FAILURE, I.E.,'
[29]  PRINT '               NO CHANGES IN STATES ARE MADE.  THUS,
      METAPHOR GENERATES'
[30]  PRINT '               AN IDENTITY MATRIX.'
[31]  PRINT ''
[32]  PRINT ''
[33]  PRINT 'DO YOU WANT REFERENCES?'
[34]  ⍝ LEAVE IF NOT
[35]  →(~INYES)/0
[36]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[37]  PRINT ''
[38]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[39]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[40]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[41]  PRINT ''
[42]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[43]  PRINT ''
[44]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[45]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[46]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[47]  PRINT ''
[48]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[49]  PRINT ''
[50]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[51]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[52]  PRINT ''
      ∇
```

GSINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GSINFO

PURPOSE: To generate the GETSTATES HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETSTATES function. If the user wishes references, GSINFO also prints a small selection of references.

LISTING:

```
    ∇ GSINFO
[1]   ⍝
[2]   ⍝  ROUTINE FOR GIVING HELP ON INPUTTING THE NUMBER OF STATES
      IN EACH PHASE
[3]   ⍝
[4]   PRINT 'ENTER THE NUMBER OF STATES FOR EACH PHASE IN THE FINITE
      PHASE MODEL.'
[5]   PRINT 'TYPE A POSITIVE INTEGER FOR EACH PHASE, SEPARATING EACH
      WITH SPACES AND/OR COMMAS.'
[6]   PRINT 'THE NUMBER OF STATES MUST BE A POSITIVE INTEGER.'
[7]   PRINT 'EXAMPLE:'
[8]   PRINT '        4   3,5'
[9]   PRINT 'THIS INDICATES TO METAPHOR THAT THE FIRST PHASE HAS 4
      STATES, THE SECOND'
[10]  PRINT 'PHASE HAS 3 STATES, AND THE THIRD PHASE HAS 5
      STATES.'
[11]  PRINT 'METAPHOR CHECKS TO MAKE SURE THAT THE NUMBER OF GROUPS
      OF STATES'
[12]  PRINT 'MATCHES THE NUMBER OF PHASES INPUT EARLIER.  AN ERROR
      MESSAGE WILL BE'
[13]  PRINT 'PRINTED IF THEY DO NOT MATCH.'
[14]  PRINT 'DO YOU WANT REFERENCES?'
[15]  ⍝ LEAVE IF NOT
[16]  →(~INYES)/0
[17]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
```

```
[18]  PRINT ''
[19]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[20]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[21]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[22]  PRINT ''
[23]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[24]  PRINT ''
[25]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[26]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[27]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[28]  PRINT ''
[29]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[30]  PRINT ''
[31]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[32]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[33]  PRINT ''
      ∇
```

GVVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GVVINFO

PURPOSE: To generate the GETVVECTOR HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the GETVVECTOR function. If the user wishes references, GVVINFO also prints a small selection of references.

LISTING:

```
    ∇ GVVINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING THE TIME-INVARIANT BASIC
      VARIABLE VECTOR
[3]   ⍝
[4]   PRINT 'ENTER A VECTOR OF 0''S, 1''S, AND 2''S TO INDICATE
      WHETHER THE '
[5]   PRINT 'TIME-INVARIANT BASIC VARIABLE''S OCCURRENCES OR NON+
      OCCURRENCES SHOULD BE'
[6]   PRINT 'CONSIDERED IN THE TRAJECTORY SET.  THE CODING IS AS
      FOLLOWS:'
[7]   PRINT ''
[8]   PRINT '       0       THE CORRESPONDING BASIC VARIABLE''S
      OCCURRENCE SHOULD'
[9]   PRINT '               BE CONSIDERED'
[10]  PRINT ''
[11]  PRINT '               THE CORRESPONDING BASIC VARIABLE''S NON-
      OCCURRENCE'
[12]  PRINT '               SHOULD BE CONSIDERED'
[13]  PRINT ''
[14]  PRINT '               EITHER THE OCCURRENCE OR  NON-OCCURRENCE
      OF THE '
[15]  PRINT '               CORRESPONDING BASIC VARIABLE SHOULD BE
      CONSIDERED'
[16]  PRINT '               (I.E., THE BASIC VARIABLE IS A ''DONT''T
      CARE'')'
```

```
[17]  PRINT ''
[18]  PRINT 'ENTER A ROW OF 0''S, 1''S, AND 2''S, SEPARATING EACH
      ENTRY BY SPACES'
[19]  PRINT 'AND/OR COMMAS.  THE NUMBER OF ENTRIES SHOULD BE THE
      SAME AS THE '
[20]  PRINT 'NUMBER OF TIME-INVARIANT BASIC VARIABLES DECLARED
      EARLIER.  ALSO THE ORDER'
[21]  PRINT 'OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER OF THE
      BASIC VARIABLES'
[22]  PRINT 'AS CONSIDERED ELSEWHERE IN THE METAPHOR PACKAGE.'
[23]  PRINT 'EXAMPLE:'
[24]  PRINT '          0,1  1 , 2'
[25]  PRINT 'THIS INDICATES TO METAPHOR THAT FOR THIS TRAJECTORY
      SET, THE OCCURRENCE'
[26]  PRINT 'OF THE FIRST TIME-INVARIANT BASIC VARIABLE IS IMPORTANT
      TO THE TRAJECTORY'
[27]  PRINT 'SET, THE NON-OCCURRENCE OF THE SECOND AND THIRD TIME-
      INVARIANT BASIC VARIABLES'
[28]  PRINT 'IS IMPORTANT, AND THAT THE FOURTH TIME+INVARIANT BASIC
      VARIABLE IS IRRELEVANT.'
[29]  PRINT ''
[30]  PRINT 'DO YOU WANT REFERENCES?'
[31]  ⍝ LEAVE IF NOT
[32]  →(~INYES)/0
[33]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[34]  PRINT ''
[35]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[36]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[37]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[38]  PRINT ''
[39]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[40]  PRINT ''
[41]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[42]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[43]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[44]  PRINT ''
[45]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[46]  PRINT ''
[47]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[48]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[49]  PRINT ''
      ∇
```

METINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: METINFO

PURPOSE: To generate the METAPHOR HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

COMMENTS: Prints the information associated with the METAPHOR function. If the user wishes references, METINFO also prints a small selection of references.

LISTING:

```
    ∇ METINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON METAPHOR
[3]   ⍝
[4]   PRINT 'METAPHOR IS AN INTERACTIVE SOFTWARE PACKAGE AIDING THE
      MODELING'
[5]   PRINT 'AND ANALYSIS OF PERFORMABILITY.  AT PRESENT, METAPHOR
      IS CAPABLE '
[6]   PRINT 'ONLY OF EVALUATING CERTAIN PERFORMABILITY MODELS.'
[7]   PRINT 'THE COMMANDS PRESENTLY AVAILABLE ARE:  EVAL,HELP, DATA,
      ALTER, CALC,'
[8]   PRINT '          COM, BRIEF [ON|OFF], ECHO [ON|OFF], AND
      EXIT.'
[9]   PRINT 'DO YOU WANT MORE HELP?'
[10]  ⍝ IF NOT, LEAVE
[11]  →(~INYES)/0
[12]  PRINT 'THE COMMANDS CAN BE ENTERED AT ANY TIME EXCEPT IN
      RESPONSE TO A '
[13]  PRINT 'YES/NO QUESTION.  THE COMMANDS ARE AS FOLLOWS:'
[14]  PRINT ''
[15]  PRINT '          EVAL      EVALUATE A USER SUPPLIED
      PERFORMABILITY MODEL'
[16]  PRINT ''
[17]  PRINT '          HELP      GIVE MORE INFORMATION ABOUT THE
      QUESTION BEING '
[18]  PRINT '                    ASKED'
[19]  PRINT ''
```

```
[20]  PRINT '          DATA      DISPLAY VARIABLE INFORMATION AND
      MODEL PARAMETERS'
[21]  PRINT ''
[22]  PRINT '         ALTER      CHANGE VARIABLE INFORMATION AND
      MODEL PARAMETERS'
[23]  PRINT ''
[24]  PRINT '          CALC      ENTER THE APL CALCULATOR MODE.  TYPE
      ''EXIT'' TO LEAVE.'
[25]  PRINT ''
[26]  PRINT '           COM      ENTER COMMENTS ON THE OUTPUT'
[27]  PRINT ''
[28]  PRINT 'BRIEF [ON|OFF]      TURN BRIEF OUTPUT ON OR OFF'
[29]  PRINT ''
[30]  PRINT ' ECHO [ON|OFF]      TURN INPUT ECHO ON OR OFF'
[31]  PRINT ''
[32]  PRINT '          EXIT      LEAVE METAPHOR'
[33]  PRINT ''
[34]  PRINT 'DO YOU WANT REFERENCES?'
[35]  A LEAVE IF NOT
[36]  →(~INYES)/0
[37]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[38]  PRINT ''
[39]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[40]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[41]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[42]  PRINT ''
[43]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[44]  PRINT ''
[45]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[46]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[47]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[48]  PRINT ''
[49]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[50]  PRINT ''
[51]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[52]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[53]  PRINT ''
      ∇
```

8. REFERENCES

[1] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," Semi-Annual Status Report Number 1, NASA Grant NSG 1306, November, 1976.

[2] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," Semi-Annual Status Report Number 2, NASA Grant NSG 1306, July, 1977.

[3] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," Semi-Annual Status Report Number 3, NASA Grant NSG 1306, January 1978.

[4] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," Semi-Annual Status Report Number 4, NASA Grant NSG 1306, July 1978.

[5] J. F. Meyer, "A model hierarchy for evaluating the effectiveness of computing systems," Texte des Conferences III-e Congres National de Fiabilite, Tome II, Perros-Guirec, France, pp. 539-555, September 1976.

[6] R. A. Ballance and J. F. Meyer, "Functional dependence and its application to system evaluation," Proc. Of the 1978 Johns Hopkins Conference on Information Sciences and Systems, Baltimore, MD, pp. 280-285, March 1978.

[7] J. F. Meyer, "On evaluating the performability of degradable computing systems," Proc. 1978 Int'l Symp. on Fault-Tolerant Computing, Toulouse, France, pp. 44-49, June, 1978.

[8] D. G. Furchtgott and J. F. Meyer, "Performability evaluation of fault-tolerant multiprocessors", 1978 Government Microcircuit Applications Conference Digest of Papers, Monterey, California, pp. 362-365, November, 1978.

[9] J. F. Meyer, D. G. Furchtgott, L. T. Wu, "Performability evaluation of the SIFT computer," SEL Report No. 127, Systems Engineering Lab, The University of Michigan, Ann Arbor, MI, January 1979.

[10] K. E. Iverson, A Programming Language. New York: Wiley, 1962.

[11] S. Pakin, APL/360 Reference Manual. Chicago, IL: Science Research Associates, 1972.

APPENDIX: Listing of METAPHOR

Below we present a complete listing of METAPHOR, Version 1. The functions are grouped somewhat arbitrarily by purpose. Each information function is below the function which it describes. An alphabetical ordering is presented in the descriptions of Section 6. As a summary of the contents of this section, the outline below gives the function groupings with no listing.

Main Functions

- METAPHOR
- DECLAREMETAPHOR
- METINFO

Command Functions

- COMMANDALTER
- COMMANDBRIEF
- COMMANDCALC
- COMMANDCOM
- COMMANDDATA
- COMMANDECHO
- COMMANDEVAL
- COMMANDHELP

Command Support Functions

- BRIEF
- ECHO
- GETALTERVECTOR
- GAVINFO
- GETDATAVECTOR
- GDVINFO

Command EVAL Implementation Functions

- GETNUMPHASES
- GNPINFO
- GETSTATES
- GSINFO
- GETPMATRICES
- GETHMATRICES
- GETNUMBASICVARIABLES
- GNBVINFO
- GETBASICVARIABLES
- GBVINFO
- GETNUMACCLEV
- GNAINFO
- GETPERF
- PUTPERFORMABILITY

Matrix Generator Functions

- GENERATEHMATRIX
- GHMINFO
- GENERATEPMATRIX
- GPMINFO
- GDEDFAIL
- GDINFO
- GGIVEN
- GGINFO
- GIDENTITY
- GNFAIL
- GNINFO

Performability Computation Functions

- GETACCLEVPROB
- GETNUMTRAJSETS
- GNTSINFO
- GETIVECTOR
- GIVINFO
- GETGMATRICES
- GGMINFO

GETFVECTOR
GFVINFO
GETVVALUES
GVVINFO
CALCTRAJPROB

I/O and Checking Functions

INPUT
INYES
CHECKBIN
CHECKPOSI
CHECKPROB
CHECKTRI
PRINT
PRINTQUAD

APL Support Function

ENCODE.

MAIN FUNCTIONS

```
      ∇ METAPHOR
[1]    ⍝
[2]    ⍝
[3]    ⍝
[4]    ⍝  PROGRAM FOR EVALUATION OF PERFORMABILITIES
[5]    ⍝
[6]    ⍝
[7]    ⍝  SET PROGRAM CONSTANTS
[8]    DECLAREMETAPHOR
[9]    ⍝
[10]   ⍝
[11]   ⍝ PRINT HEADING
[12]   PRINT ''
[13]   PRINT ''
[14]   PRINT 'MICHIGAN EVALUATION AID FOR PERPHORMABILITY'
[15]   PRINT ''
[16]   PRINT ''
[17]   PRINT ''
[18]   PRINT ''
[19]   PRINT 'TYPE HELP FOR ASSISTANCE'
[20]   ⍝  PROGRAM LOOPS UNTIL EXIT TAKEN
[21] MIN:6 INPUT MET
[22]   →MIN
      ∇
```

```
      ∇ DECLAREMETAPHOR
[1]    ⍝
[2]    ⍝   DECLARES AND INITIALIZES THE CONSTANTS USED IN THE 'PERF'
       FUNCTION SERIES
[3]    ⍝
[4]    ⍝  SET THE LIST OF COMMANDS
[5]    HELP←'HELP  '
[6]    EXIT←'EXIT  '
[7]    DATA←'DATA  '
[8]    ALTER←'ALTER '
[9]    CALC←'CALC  '
[10]   COM←'COM   '
[11]   EVAL←'EVAL  '
[12]   ON←'ON    '
[13]   OFF←'OFF   '
[14]   COMMANDSIZE←6
[15]   COMMANDLIST←(9,COMMANDSIZE)⍴'HELP  ','EXIT  ','DATA  ','ALTER
       ','CALC  ','ECHO  ','BRIEF ','COM   ','EVAL  '
[16]   ⍝
[17]   ⍝
[18]   ⍝  SET LIST OF MATRIX GENERATOR TYPES
[19]   GIVEN←'GIVEN   '
[20]   DEDFAIL←'DEDFAIL '
```

```
[21]  NFAIL←'NFAIL    '
[22]  IDENTITY←'IDENTITY'
[23]  PMATRIXLIST←(4 8)ρ'GIVEN   ','DEDFAIL ','NFAIL
      ','IDENTITY'
[24]  HMATRIXLIST←(2 8)ρ'GIVEN   ','IDENTITY'
[25]  ⍝
[26]  ⍝
[27]  ⍝
[28]  ⍝  SET LIST OF ROUTINES FOR HELP CALLS
[29]  GNP←1
[30]  GS←2
[31]  GPM←3
[32]  GG←4
[33]  GN←5
[34]  GD←6
[35]  GHM←7
[36]  GNBV←8
[37]  GBV←9
[38]  GNA←10
[39]  GIV←11
[40]  GGM←12
[41]  GFV←13
[42]  GVV←14
[43]  GNTS←15
[44]  GAV←16
[45]  GDV←17
[46]  MET←18
[47]  ⍝
[48]  ⍝ SET NUMBER OF HELP ROUTINES
[49]  NUMHELPROUTINES←18
[50]  ⍝
[51]  ⍝ SET VARIABLE DEFINITION SWITCHES.  1 IF VARIABLE DEFINED, 0
      IF NOT.
[52]  DEFNUMPHASES←0
[53]  DEFNUMSTATES←0
[54]  DEFP←0
[55]  DEFH←0
[56]  DEFNUMBASICVARIABLES←0
[57]  DEFBASICVARIABLES←0
[58]  DEFNUMACCLEV←0
[59]  DEFNUMTRAJSETS←0
[60]  DEFI←0
[61]  DEFG←0
[62]  DEFF←0
[63]  DEFV←0
[64]  DEFPERFORMABILITY←0
[65]  DEFACCLEVEL←0
[66]  ⍝
[67]  ⍝  MAKE DEFAULT SWITCH SETTINGS
[68]  BRIEFOUTPUT←0
[69]  ECHOINPUT←0
      ∇
```

```
     ∇ METINFO
[1]  ⍝
[2]  ⍝   ROUTINE GIVING HELP ON METAPHOR
[3]  ⍝
[4]  PRINT 'METAPHOR IS AN INTERACTIVE SOFTWARE PACKAGE AIDING THE
     MODELING'
[5]  PRINT 'AND ANALYSIS OF PERFORMABILITY.  AT PRESENT, METAPHOR
     IS CAPABLE '
[6]  PRINT 'ONLY OF EVALUATING CERTAIN PERFORMABILITY MODELS.'
[7]  PRINT 'THE COMMANDS PRESENTLY AVAILABLE ARE:  EVAL,HELP, DATA,
     ALTER, CALC,'
[8]  PRINT '          COM, BRIEF [ON|OFF], ECHO [ON|OFF], AND
     EXIT.'
[9]  PRINT 'DO YOU WANT MORE HELP?'
[10] ⍝ IF NOT, LEAVE
[11] →(~INYES)/0
[12] PRINT 'THE COMMANDS CAN BE ENTERED AT ANY TIME EXCEPT IN
     RESPONSE TO A '
[13] PRINT 'YES/NO QUESTION.  THE COMMANDS ARE AS FOLLOWS:'
[14] PRINT ''
[15] PRINT '          EVAL     EVALUATE A USER SUPPLIED
     PERFORMABILITY MODEL'
[16] PRINT ''
[17] PRINT '          HELP     GIVE MORE INFORMATION ABOUT THE
     QUESTION BEING '
[18] PRINT '                   ASKED'
[19] PRINT ''
[20] PRINT '          DATA     DISPLAY VARIABLE INFORMATION AND
     MODEL PARAMETERS'
[21] PRINT ''
[22] PRINT '         ALTER     CHANGE VARIABLE INFORMATION AND
     MODEL PARAMETERS'
[23] PRINT ''
[24] PRINT '          CALC     ENTER THE APL CALCULATOR MODE.  TYPE
     ''EXIT'' TO LEAVE.'
[25] PRINT ''
[26] PRINT '           COM     ENTER COMMENTS ON THE OUTPUT'
[27] PRINT ''
[28] PRINT 'BRIEF [ON|OFF]     TURN BRIEF OUTPUT ON OR OFF'
[29] PRINT ''
[30] PRINT ' ECHO [ON|OFF]     TURN INPUT ECHO ON OR OFF'
[31] PRINT ''
[32] PRINT '          EXIT     LEAVE METAPHOR'
[33] PRINT ''
[34] PRINT 'DO YOU WANT REFERENCES?'
[35] ⍝ LEAVE IF NOT
[36] →(~INYES)/0
[37] PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
     ANALYSIS, SEE'
[38] PRINT ''
[39] PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
     THE '
```

```
[40]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[41]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[42]  PRINT ''
[43]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[44]  PRINT ''
[45]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[46]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[47]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[48]  PRINT ''
[49]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[50]  PRINT ''
[51]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[52]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[53]  PRINT ''
```

COMMAND FUNCTIONS

```
      ∇ COMMANDALTER;ALTERVECTOR
[1]    ⍝
[2]    ⍝   ROUTINE FOR CHANGING CURRENT DATA
[3]    ⍝
[4]    ⍝  GET DATA TO BE ALTERED
[5]    ALTERVECTOR←GETALTERVECTOR
[6]    ⍝
[7]    ⍝ CHANGE THAT DATA.  EXIT WHEN THROUGH.
[8]   ALOOP:→ALTERVECTOR/AP,AH,ABASICVARIABLES,AALLACCLEVELS,
      APRESENTACCLEVEL,AI,AG,AF,AV,ANUMTRAJSETS
[9]    →0
[10]   ⍝
[11]   ⍝
[12]   ⍝  CHANGE THE REQUESTED DATA
[13]   ⍝
[14]   ⍝ CHANGE THE P MATRICES
[15]  AP:→(DEFP=1)/APALTER
[16]   ⍝P MATRICES ARE NOT DEFINED AT THIS TIME.  PRINT ERROR
       MESSAGE
[17]   PRINT 'P MATRICES ARE NOT DEFINED AT THIS TIME.'
[18]   →APLOOP
[19]  APALTER:PRINT 'ALTERING P'
[20]   GETPMATRICES
[21]  APLOOP:ALTERVECTOR[1]←0
[22]   →ALOOP
[23]   ⍝
[24]   ⍝ CHANGE H MATRICES
[25]  AH:→(DEFH=1)/AHALTER
[26]   ⍝H MATRICES ARE NOT DEFINED AT THIS TIME.  PRINT ERROR
       MESSAGE
```

```
[27]  PRINT 'H MATRICES ARE NOT DEFINED AT THIS TIME.'
[28]  →AHLOOP
[29] AHALTER:PRINT 'ALTERING H'
[30]  GETHMATRICES
[31] AHLOOP:ALTERVECTOR[2]←0
[32]  →ALOOP
[33]  ⍝
[34]  ⍝ CHANGE THE TIME-INVARIANT BASIC VARIABLES
[35] ABASICVARIABLES:→( DEFBASICVARIABLES=1)/ABASICVARIABLESALTER
[36]  ⍝TIME-INVARIANT BASIC VARIABLES ARE NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE
[37]  PRINT 'TIME-INVARIANT BASIC VARIABLES ARE NOT DEFINED AT THIS
      TIME.'
[38]  →ABASICVARIABLESLOOP
[39] ABASICVARIABLESALTER:PRINT 'ALTERING THE NUMBER OF BASIC
      VARIABLES'
[40]  GETNUMBASICVARIABLES
[41]  GETBASICVARIABLES
[42] ABASICVARIABLESLOOP:ALTERVECTOR[3]←0
[43]  →ALOOP
[44]  ⍝
[45]  ⍝ CHANGE ALL ACCOMPLISHMENT LEVELS
[46] AALLACCLEVELS:→(DEFNUMACCLEV=1)/AALLACCLEVELSALTER
[47]  ⍝THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE
[48]  PRINT 'THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS
      TIME.'
[49]  →AALLACCLEVELSLOOP
[50] AALLACCLEVELSALTER:PRINT 'ALTERING THE NUMBER OF ACCOMPLISHMENT
      LEVELS'
[51]  GETNUMACCLEV
[52]  GETACCLEVPROB
[53] AALLACCLEVELSLOOP:ALTERVECTOR[4]←0
[54]  →ALOOP
[55]  ⍝
[56]  ⍝ CHANGE JUST THE PRESENT (LAST) ACCOMPLISHMENT LEVEL
[57] APRESENTACCLEVEL:→(DEFACCLEVEL=1)/APRESENTACCLEVELALTER
[58]  ⍝AN ACCOMPLISHMENT LEVEL IS NOT DEFINED AT THIS TIME.  PRINT
      ERROR MESSAGE
[59]  PRINT 'AN ACCOMPLISHMENT LEVEL IS NOT DEFINED AT THIS TIME.'
[60]  →APRESENTACCLEVELLOOP
[61] APRESENTACCLEVELALTER:PRINT 'ALTERING THE PRESENT
      ACCOMPLISHMENT LEVEL'
[62]  PERFORMABILITY[L]←GETACCLEVPROB L
[63] APRESENTACCLEVELLOOP:ALTERVECTOR[5]←0
[64]  →ALOOP
[65]  ⍝
[66]  ⍝ CHANGE THE I VECTOR
[67] AI:→(DEFI=1)/AIALTER
[68]  ⍝I VECTOR IS NOT DEFINED AT THIS TIME.  PRINT ERROR MESSAGE
[69]  PRINT 'I VECTOR IS NOT DEFINED AT THIS TIME.'
[70]  →AILOOP
```

```
[71] AIALTER:PRINT 'ALTERING THE I VECTOR'
[72]  I←GETIVECTOR
[73] AILOOP:ALTERVECTOR[6]←0
[74]  →ALOOP
[75]  ⍝
[76]  ⍝ CHANGE THE G MATRICES
[77] AG:→(DEFG=1)/AGALTER
[78]  ⍝G MATRICES ARE NOT DEFINED AT THIS TIME.  PRINT ERROR
      MESSAGE
[79]  PRINT 'G MATRICES ARE NOT DEFINED AT THIS TIME.'
[80]  →AGLOOP
[81] AGALTER:PRINT 'ALTERING THE G MATRICES'
[82]  G←GETGMATRICES
[83] AGLOOP:ALTERVECTOR[7]←0
[84]  →ALOOP
[85]  ⍝
[86]  ⍝ CHANGE THE F VECTOR
[87] AF:→(DEFF=1)/AFALTER
[88]  ⍝F VECTOR IS NOT DEFINED AT THIS TIME.  PRINT ERROR MESSAGE
[89]  PRINT 'F VECTOR IS NOT DEFINED AT THIS TIME.'
[90]  →AFLOOP
[91] AFALTER:PRINT 'ALTERING THE F VECTOR'
[92]  F←GETFVECTOR
[93] AFLOOP:ALTERVECTOR[8]←0
[94]  →ALOOP
[95]  ⍝
[96]  ⍝ CHANGE THE TIME-INVARIANT BASIC VARIABLE VECTOR
[97] AV:→(DEFV=1)/AVALTER
[98]  ⍝THE TIME-INVARIANT BASIC VARIABLE VECTOR IS NOT DEFINED AT
      THIS TIME.  PRINT ERROR MESSAGE
[99]  PRINT 'THE TIME-INVARIANT BASIC VARIABLE VECTOR IS NOT DEFINED
      AT THIS TIME.'
[100] →AVLOOP
[101]AVALTER:PRINT 'ALTERING THE V VECTOR'
[102] V←GETVVALUES
[103]AVLOOP:ALTERVECTOR[9]←0
[104] →ALOOP
[105] ⍝
[106] ⍝ CHANGE THE NUMBER OF TRAJECTORY SETS BEING CONSIDERED
[107]ANUMTRAJSETS:→(DEFNUMTRAJSETS=1)/ANUMTRAJSETSALTER
[108] ⍝THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT THIS
      TIME.  PRINT ERROR MESSAGE.
[109] PRINT 'THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT THIS
      TIME.'
[110]ANUMTRAJSETSALTER:PRINT 'ALTERING THE NUMBER OF TRAJECTORY
      SETS'
[111] NUMTRAJSETS←GETNUMTRAJSETS
[112] ALTERVECTOR[10]←0
[113] →ALOOP
      ∇
```

```
          ∇ COMMANDBRIEF SWITCH
[1]     ⍝
[2]     ⍝  ROUTINE FOR TURNING THE BRIEF SWITCH ON AND OFF.  'ON'
        CAUSES
[3]     ⍝      METAPHOR TO USE A TERSE FORM OF OUTPUT.  'OFF' CAUSE THE
        NORMAL FULL OUTPUT.
[4]     ⍝
[5]     ⍝ TURN SWITCH ON IF REQUESTED, ELSE TURN SWITCH OFF.
[6]     BRIEFOUTPUT←∧/'ON'∈SWITCH
[7]     ⍝ PRINT CONFIRMATION REGARDLESS OF BRIEF SWITCH
[8]     'BRIEF ';SWITCH
        ∇

        ∇ COMMANDCALC;CCALCINPUT
[1]     ⍝
[2]     ⍝   ROUTINE TO UTILIZE THE APL CALCULATOR MODE FROM THE
        METAPHOR PACKAGE.
[3]     ⍝      HALTS WHEN THE INPUT IS NULL.
[4]     ⍝
[5]     CCALCIN:⍞←(PRINTQUAD '?')
[6]     ⍬←CCALCINPUT←⍞
[7]     ⍝ LEAVE IF EXIT SPECIFIED.  ELSE GET NEXT CALCULATION.
[8]     →(1=∧/CCALCINPUT=EXIT)/0
[9]     →CCALCIN
        ∇

        ∇ COMMANDCOM;CCINPUT
[1]     ⍝
[2]     ⍝  ROUTINE FOR ALLOWING THE USER TO PRINT A COMMENT
[3]     ⍝      EACH COMMENT IS PRECEDED BY '***' AND THIS
[4]     ⍝      HALTS WHEN THE INPUT IS NULL
[5]     ⍝
[6]     ⍝ PRINT PROMPT SYMBOLS
[7]     CCIN:⍞←(PRINTQUAD '***')
[8]     CCINPUT←⍞
[9]     ⍝ IF COMMENT NOT EMPTY, GET NEXT COMMENT.  ELSE LEAVE.
[10]    →(3≠⍴CCINPUT)/CCIN
        ∇

        ∇ COMMANDDATA;DATAVECTOR
[1]     ⍝
[2]     ⍝   ROUTINE FOR DISPLAYING CURRENT DATA
[3]     ⍝
[4]     ⍝  GET DATA TO BE DISPLAYED
[5]     DATAVECTOR←GETDATAVECTOR
[6]     ⍝
[7]     ⍝ DISPLAY THAT DATA.  EXIT WHEN THROUGH.

[8]     DLOOP:→DATAVECTOR/DNUMPHASES,DNUMSTATES,DP,DH,DNUMBASICVAR
```

```
      DBASICVARIABLES,DNUMACCLEV,DNUMTRAJSETS,DI,DG,DF,DV,DPERF
[9]   →0
[10]  ⍝
[11]  ⍝
[12]  ⍝  SHOW THE REQUESTED INFORMATION
[13]  ⍝
[14] DNUMPHASES:→(DEFNUMPHASES=1)/DNUMPHASESOUT
[15]  PRINT 'NUMBER OF PHASES HAS NOT BEEN DEFINED'
[16]  →DNUMPHASESLOOP
[17] DNUMPHASESOUT:⍞←((PRINTQUAD 'NUMBER OF PHASES IS ');PRINTQUAD
      NUMPHASES)
[18]  PRINT ''
[19] DNUMPHASESLOOP:DATAVECTOR[1]←0
[20]  →DLOOP
[21]  ⍝
[22] DNUMSTATES:→(DEFNUMSTATES=1)/DNUMSTATESOUT
[23]  PRINT 'NUMBER OF STATES HAS NOT BEEN DEFINED'
[24]  →DNUMSTATESLOOP
[25] DNUMSTATESOUT:⍞←((PRINTQUAD 'NUMBER OF STATES PER PHASE IS
      ');PRINTQUAD STATESPERPHASE)
[26]  PRINT ''
[27] DNUMSTATESLOOP:DATAVECTOR[2]←0
[28]  →DLOOP
[29]  ⍝
[30] DP:→(DEFP=1)/DPOUT
[31]  PRINT 'P MATRICES HAVE NOT BEEN DEFINED'
[32]  →DPLOOP
[33] DPOUT:PRINT 'THE P MATRICES ARE:'
[34]  P
[35] DPLOOP:DATAVECTOR[3]←0
[36]  →DLOOP
[37]  ⍝
[38] DH:→(DEFH=1)/DHOUT
[39]  PRINT 'H MATRICES HAVE NOT BEEN DEFINED'
[40]  →DHLOOP
[41] DHOUT:PRINT 'THE H MATRICES ARE:'
[42]  H
[43] DHLOOP:DATAVECTOR[4]←0
[44]  →DLOOP
[45]  ⍝
[46] ∆NUMBASICVARIABLES:(
      DEFNUMBASICVARIABLES=1)/∆NUMBASICVARIABLESOUT
[47]  PRINT 'THE NUMBER OF BASIC VARIABLES HAS NOT BEEN DEFINED'
[48]  →DNUMBASICVARIABLESLOOP
[49] DNUMBASICVARIABLESOUT:⍞←((PRINTQUAD 'THE NUMBER OF TIME-
      INVARIANT BASIC VARIABLES IS ');PRINTQUAD NUMBASICVARIABLES)
[50]  PRINT ''
[51] DNUMBASICVARIABLESLOOP:DATAVECTOR[5]←0
[52]  →DLOOP
[53]  ⍝
[54] DBASICVARIABLES:→(DEFBASICVARIABLES=1)/DBASICVARIABLESOUT
[55]  PRINT 'THE TIME-INVARIANT BASIC VARIABLES HAVE NOT BEEN
```

```
      DEFINED'
[56]   →DBASICVARIABLESLOOP
[57]  DBASICVARIABLESOUT:PRINT 'THE TIME-INVARIANT BASE VARIABLES
      HAVE PROBABILITIES:'
[58]   BASICVARIABLES
[59]  DBASICVARIABLESLOOP:DATAVECTOR[6]←0
[60]   →DLOOP
[61]   ⍝
[62]  DNUMACCLEV:→(DEFNUMACCLEV=1)/DNUMACCLEVOUT
[63]   PRINT 'THE NUMBER OF ACCOMPLISHMENT LEVELS NOT DEFINED'
[64]   →DNUMACCLEVLOOP
[65]  DNUMACCLEVOUT:⍞←((PRINTQUAD 'THE NUMBER OF ACCOMPLISHMENT
      LEVELS IS ');PRINTQUAD NUMACCLEV)
[66]   PRINT ''
[67]  DNUMACCLEVLOOP:DATAVECTOR[7]←0
[68]   →DLOOP
[69]   ⍝
[70]  DNUMTRAJSETS:→(DEFNUMTRAJSETS=1)/DNUMTRAJSETSOUT
[71]   PRINT 'THE NUMBER OF TRAJECTORY SETS NOT DEFINED'
[72]   →DNUMTRAJSETSLOOP
[73]  DNUMTRAJSETSOUT:PRINT 'THE NUMBER OF TRAJECTORY SETS IS:'
[74]   NUMTRAJSETS
[75]  DNUMTRAJSETSLOOP:DATAVECTOR[8]←0
[76]   →DLOOP
[77]   ⍝
[78]  DI:→(DEFI=1)/DIOUT
[79]   PRINT 'I VECTOR NOT DEFINED'
[80]   →DILOOP
[81]  DIOUT:⍞←((PRINTQUAD 'THE INITIAL VECTOR IS ');PRINTQUAD I)
[82]   PRINT ''
[83]  DILOOP:DATAVECTOR[9]←0
[84]   →DLOOP
[85]   ⍝
[86]  DG:→(DEFG=1)/DGOUT
[87]   PRINT 'G MATRICES NOT DEFINED'
[88]   →DGLOOP
[89]  DGOUT:PRINT 'THE G MATRICES ARE:'
[90]   G
[91]  DGLOOP:DATAVECTOR[10]←0
[92]   →DLOOP
[93]   ⍝
[94]  DF:→(DEFF=1)/DFOUT
[95]   PRINT 'F VECTOR NOT DEFINED'
[96]   →DFLOOP
[97]  DFOUT:⍞←((PRINTQUAD 'THE FINAL VECTOR IS ');PRINTQUAD F)
[98]   PRINT ''
[99]  DFLOOP:DATAVECTOR[11]←0
[100]  →DLOOP
[101]  ⍝
[102] DV:→(DEFV=1)/DVOUT
[103]  PRINT 'THE TIME-INVARIANT BASIC VARIABLE VECTOR NOT DEFINED'
[104]  →DVLOOP
```

```
[105]DVOUT:⎕←((PRINTQUAD 'THE TIME-INVARIANT BASIC VARIABLE VECTOR
      IS ');PRINTQUAD V)
[106] PRINT ''
[107]DVLOOP:DATAVECTOR[12]←0
[108] →DLOOP
[109] ⍝
[110]DPERF:→(DEFPERFORMABILITY=1)/DPERFOUT
[111] PRINT 'PERFORMABILITY NOT DEFINED'
[112] →DPERFLOOP
[113]DPERFOUT:⎕←((PRINTQUAD 'THE PERFORMABILITY IS ');PRINTQUAD
      PERFORMABILITY)
[114]DPERFLOOP:DATAVECTOR[13]←0
[115] →DLOOP
     ∇


     ∇ COMMANDECHO SWITCH
[1]   ⍝
[2]   ⍝ ROUTINE FOR TURNING THE ECHO SWITCH ON AND OFF. 'ON'
      CAUSES
[3]   ⍝      METAPHOR TO REPEAT EVERY INPUT LINE. 'OFF' SUPPRESSES
      THE REPETITION.
[4]   ⍝
[5]   ⍝ TURN SWITCH ON IF REQUESTED, ELSE TURN SWITCH OFF.
[6]   ECHOINPUT←∧/'ON'∈SWITCH
[7]   'ECHO ';SWITCH
     ∇


     ∇ COMMANDEVAL
[1]   ⍝
[2]   ⍝   PERFORMABILITY COMPUTATION PORTION OF METAPHOR
[3]   ⍝
[4]   ⍝ FETCH THE MATRICES REQUIRED FOR PERFORMABILITY
      CALCULATIONS
[5]  MGNUMP:GETNUMPHASES
[6]   GETSTATES
[7]   GETPMATRICES
[8]   GETHMATRICES
[9]   GETNUMBASICVARIABLES
[10]  GETBASICVARIABLES
[11]  GETNUMACCLEV
[12]  ⍝
[13]  ⍝ FOR EACH ACCOMPLISHMENT LEVEL, FETCH THE TRAJECTORY SETS
      AND CALCULATE THEIR PROBABILITIES
[14]  GETPERFORMABILITY
[15]  ⍝
[16]  ⍝ PRINT THE RESULTING PERFORMABILITY
[17]  PRINTPERFORMABILITY
     ∇
```

```
          ∇ COMMANDHELP ROUTINE
[1]    ⍝
[2]    ⍝  ROUTINE PRINTING HELP INFORMATION
[3]    ⍝
[4]    ⍝ GET THE PROPER INFO ROUTINE
[5]    →(ROUTINE=⍳NUMHELPROUTINES)/HGNP,HGS,HGPM,HGG,
       HGN,HGD,HGHM,HGNBV,HGBV,HGNA,HGIV,
       HGGM,HGFV,HGVV,HGNTS,HGAV,HGDV,HMET
[6]    ⍝
[7]    ⍝ 'GET NUMBER OF PHASES' HELP
[8]    HGNP:GNPINFO
[9]    →0
[10]   ⍝
[11]   ⍝ 'GET STATES' HELP
[12]   HGS:GSINFO
[13]   →0
[14]   ⍝
[15]   ⍝ 'GENERATE P MATRIX' HELP
[16]   HGPM:GPMINFO
[17]   →0
[18]   ⍝
[19]   ⍝ 'GET GIVEN MATRIX' HELP
[20]   HGG:GGINFO
[21]   →0
[22]   ⍝
[23]   ⍝ 'GET NFAIL MATRIX' HELP
[24]   HGN:GNINFO
[25]   →0
[26]   ⍝
[27]   ⍝ 'GET DEDFAIL MATRIX' HELP
[28]   HGD:GDINFO
[29]   →0
[30]   ⍝
[31]   ⍝ 'GENERATE H MATRIX' HELP
[32]   HGHM:GHMINFO
[33]   →0
[34]   ⍝
[35]   ⍝ 'GET NUMBER OF TIME-INVARIANT BASIC VARIABLES' HELP
[36]   HGNBV:GNBVINFO
[37]   →0
[38]   ⍝
[39]   ⍝ 'GET TIME-INVARIANT BASIC VARIABLES' HELP
[40]   HGBV:GBVINFO
[41]   →0
[42]   ⍝
[43]   ⍝ 'GET NUMBER OF ACCOMPLISHMENT LEVELS' HELP
[44]   HGNA:GNAINFO
[45]   →0
[46]   ⍝
[47]   ⍝ 'GET I VECTOR' HELP
[48]   HGIV:GIVINFO
[49]   →0
```

```
[50]   ⍝
[51]   ⍝ 'GET G MATRIX' HELP
[52]  HGGM:GGMINFO
[53]   →0
[54]   ⍝
[55]   ⍝ 'GET F VECTOR' HELP
[56]  HGFV:GFVINFO
[57]   →0
[58]   ⍝
[59]   ⍝ 'GET V VECTOR' HELP
[60]  HGVV:GVVINFO
[61]   →0
[62]   ⍝
[63]   ⍝ 'GET NUMBER OF TRAJECTORY SETS' HELP
[64]  HGNTS:GNTSINFO
[65]   →0
[66]   ⍝
[67]   ⍝ 'GET ALTER VECTOR VECTOR' HELP
[68]  HGAV:GAVINFO
[69]   →0
[70]   ⍝
[71]   ⍝ 'GET DATA VECTOR VECTOR' HELP
[72]  HGDV:GDVINFO
[73]   →0
[74]   ⍝
[75]   ⍝ 'METAPHOR' HELP
[76]  HMET:METINFO
[77]   →0
      ∇
```

COMMAND SUPPORT FUNCTIONS

```
      ∇ Z←BRIEF SWITCH
[1]    ⍝
[2]    ⍝  ROUTINE TO INPUT THE BRIEF COMMAND
[3]    ⍝
[4]    ⍝ RETURN THE VALUE
[5]    Z←'BRIEF ',SWITCH
      ∇


      ∇ Z←ECHO SWITCH
[1]    ⍝
[2]    ⍝  ROUTINE TO INPUT THE ECHO COMMAND
[3]    ⍝
[4]    ⍝ RETURN THE VALUE
[5]    Z←'ECHO  ',SWITCH
      ∇
```

```
     ∇ Z←GETALTERVECTOR;GAVINPUT
[1]   ⍝
[2]   ⍝   ROUTINE FOR GETTING THE ALTER VECTOR FOR CHANGING DATA
[3]   ⍝
[4]   ⍝  PRINT EXPLANATION
[5]  GAVIN:PRINT 'PUT AN X BELOW EACH ITEM TO BE CHANGED.  HELP
      AVAILABLE.'
[6]   ⍝  GET ALTER REQUESTS, RESHAPING ALONG THE WAY
[7]   PRINT 'P  H  CONST.BAS.VARS  ALL.ACC.LEVELS
      PRESENT.ACC.LEVEL  I  G  F V NUM.TRAJ.SETS'
[8]   GAVINPUT←80⍴(⎕,(80⍴' '))
[9]   ⍝ LOOK FOR HELP REQUEST.  IF PRESENT, CALL FOR HELP ROUTINE.
[10]  →(~∧/'HELP'∈GAVINPUT)/GAVVECTORSET
[11]  COMMANDHELP GAV
[12]  →GAVIN
[13]  ⍝  INITIALIZE ALTER VECTOR
[14] GAVVECTORSET:Z←10⍴0
[15]  ⍝  DETERMINE CHANGE VECTOR
[16]  Z[1]←'X'∈GAVINPUT[1]
[17]  Z[2]←'X'∈GAVINPUT[4]
[19]  Z[4]←'X'∈GAVINPUT[22+⍳14]
[20]  Z[5]←'X'∈GAVINPUT[38+⍳17]
[21]  Z[6]←'X'∈GAVINPUT[58]
[22]  Z[7]←'X'∈GAVINPUT[61]
[23]  Z[8]←'X'∈GAVINPUT[64]
[24]  Z[9]←'X'∈GAVINPUT[66]
[25]  Z[10]←'X'∈GAVINPUT[67+⍳13]
     ∇

     ∇ GAVINFO
[1]   ⍝
[2]   ⍝  ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE DISPLAYED
[3]   ⍝
[4]   PRINT 'PLACE AN X BELOW EACH ENTRY IN THE GIVEN LIST WHICH YOU
      WANT TO '
[5]   PRINT 'CHANGE.   THE ABBREVIATIONS ARE AS FOLLOWS:'
[6]   PRINT ''
[7]   PRINT '                    P      THE INTRAPHASE STATE TRANSITION
      (P) MATRICES'
[8]   PRINT ''
[9]   PRINT '                    H      THE INTERPHASE STATE TRANSITION
      (H) MATRICES'
[10]  PRINT ''
[11]  PRINT '     CONST.BAS.VARS      THE NUMBER OF TIME-INVARIANT
      BASIC VARIABLES AND'
[12]  PRINT '                         THEIR ASSOCIATED
      PROBABILITIES'
[13]  PRINT ''
[14]  PRINT '     ALL.ACC.LEVELS      USING THE PRESENT H AND P
      MATRICES AND THE'
[15]  PRINT '                         PRESENT TIME-INVARIANT BASIC
```

```
      VARIABLE INFORMATION,'
[16]  PRINT '                              DETERMINE THE PERFORMABILITY OF
      THE SYSTEM.'
[17]  PRINT '                              METAPHOR WILL ASK FOR THE
      APPROPRIATE INFORMATION'
[18]  PRINT '                              REGARDING THE ACCOMPLISHMENT
      LEVELS.'
[19]  PRINT ''
[20]  PRINT '  PRESENT.ACC.LEVEL           ALTER ONLY THE ACCOMPLISHMENT
      LEVEL '
[21]  PRINT '                              PRESENTLY UNDER
      CONSIDERATION.'
[22]  PRINT ''
[23]  PRINT '                    I         INITIAL VECTOR'
[24]  PRINT ''
[25]  PRINT '                    G         CHARACTERISTIC MATRICES'
[26]  PRINT ''
[27]  PRINT '                    F         CHARACTERISTIC VECTOR'
[28]  PRINT '                              (AT PRESENT, THIS ALTER
      OPERATION IS NOT EXECUTABLE.)'
[29]  PRINT ''
[30]  PRINT '                    V         VECTOR CHARACTERIZING THE TIME-
      INVARIANT BASIC VARIABLES'
[31]  PRINT ''
[32]  PRINT '      NUM.TRAJ.SETS           ALTER THE NUMBER OF TRAJECTORY
      SETS DESCRIBING'
[33]  PRINT '                              THE ACCOMPLISHMENT LEVEL UNDER
      CONSIDERATION'
[34]  PRINT ''
[35]  PRINT 'IF AN ITEM IS UNDEFINED WHEN AN ALTERATION IS
      REQUESTED, AN ERROR '
[36]  PRINT 'MESSAGE WILL BE PRINTED AND THAT ALTERATION
      SUPPRESED.  MORE THAN'
[37]  PRINT 'ONE ITEM MAY BE CHANGED WITH A SINGLE ALTER
      COMMAND.  '
[38]  PRINT 'EXAMPLE:'
[39]  PRINT ''
[40]  PRINT
      'P  H  CONST.BAS.VARS  ALL.ACC.LEVELS  PRESENT.ACC.LEVEL  I  G
      V NUM.TRAJ.SETS'
[41]  PRINT 'X          X                  X          '
[42]  PRINT ''
[43]  PRINT 'THIS INFORMS METAPHOR THAT THE P AND H MATRICES ARE TO
      BE CHANGED AND THAT THE  '
[44]  PRINT 'PERFORMABILITY IS TO BE CALCULATED.  IF YOU WISH TO
      CHANGE THE NUMBER OF PHASES '
[45]  PRINT 'OR ASSOCIATED STATES, TYPE END AND BEGIN METAPHOR
      AGAIN'
[46]  PRINT ''
[47]  PRINT 'DO YOU WANT REFERENCES?'
[48]  ⍝ LEAVE IF NOT
[49]  →(~INYES)/0
```

```
[50]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[51]  PRINT ''
[52]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[53]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[54]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[55]  PRINT ''
[56]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[57]  PRINT ''
[58]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[59]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[60]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[61]  PRINT ''
[62]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[63]  PRINT ''
[64]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[65]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[66]  PRINT ''
      ∇


      ∇ Z←GETDATAVECTOR;GDVINPUT
[1]   ⍝
[2]   ⍝   ROUTINE FOR GETTING THE DATA VECTOR FOR DISPLAYING DATA
[3]   ⍝
[4]   ⍝  PRINT EXPLANATION
[5]   GDVIN:PRINT 'PUT AN X BELOW EACH ITEM TO BE DISPLAYED.  HELP
      AVAILABLE.'
[6]   ⍝  GET DATA REQUESTS, RESHAPING ALONG THE WAY
[7]   PRINT 'NUM.PHASES  NUM.STATES  P  H  NUM.CONST.BAS.VARS
      PROB.CONST.BAS.VARS'
[8]   GDVINPUT←69⍴(⎕,(69⍴' '))
[9]   ⍝LOOK FOR HELP REQUEST.  IF PRESENT, CALL FOR HELP ROUTINE
[10]  →(~∧/'HELP'∈GDVINPUT)/GDVINCONT
[11]  COMMANDHELP GDV
[12]  →GDVIN
[13]  GDVINCONT:PRINT
      'NUM.ACC.LEVELS  NUM.TRAJ.SETS  I  G  F  V  PERF'
[14]  GDVINPUT←116⍴(GDVINPUT,⎕,(47⍴' '))
[15]  PRINT ''
[16]  ⍝LOOK FOR HELP REQUEST.  IF PRESENT, CALL FOR HELP ROUTINE
[17]  →(~∧/'HELP'∈GDVINPUT)/GDVVECTORSET
[18]  COMMANDHELP GDV
[19]  →GDVIN
[20]  ⍝  INITIALIZE DISPLAY VECTOR
[21]  GDVVECTORSET:Z←13⍴0
[22]  ⍝  DETERMINE DISPLAY VECTOR
```

```
[23]  Z[1]←'X'∈GDVINPUT[⍳10]
[24]  Z[2]←'X'∈GDVINPUT[12+⍳10]
[25]  Z[3]←'X'∈GDVINPUT[25]
[26]  Z[4]←'X'∈GDVINPUT[28]
[27]  Z[5]←'X'∈GDVINPUT[30+⍳18]
[28]  Z[6]←'X'∈GDVINPUT[50+⍳19]
[29]  Z[7]←'X'∈GDVINPUT[69+⍳14]
[30]  Z[8]←'X'∈GDVINPUT[85+⍳13]
[31]  Z[9]←'X'∈GDVINPUT[101]
[32]  Z[10]←'X'∈GDVINPUT[104]
[33]  Z[11]←'X'∈GDVINPUT[107]
[34]  Z[12]←'X'∈GDVINPUT[110]
[35]  Z[13]←'X'∈GDVINPUT[112+⍳4]
      ∇
```

```
      ∇ GDVINFO
[1]    ⍝
[2]    ⍝  ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE ALTERED
[3]    ⍝
[4]    PRINT 'ENTER AN X BELOW EACH ITEM IN THE GIVEN LIST WHICH YOU
       WANT TO '
[5]    PRINT 'DISPLAY.  THE ABBREVIATIONS ARE AS FOLLOWS:'
[6]    PRINT ''
[7]    PRINT '          NUM.PHASES      THE NUMBER OF PHASES'
[8]    PRINT ''
[9]    PRINT '          NUM.STATES      THE NUMBER OF STATES'
[10]   PRINT ''
[11]   PRINT '                      P    THE INTRAPHASE TRANSITION (P)
       MATRICES'
[12]   PRINT ''
[13]   PRINT ' NUM.CONST.BAS.VARS       THE NUMBER OF TIME-INVARIANT
       BASIC VARIABLES'
[14]   PRINT ''
[15]   PRINT 'PROB.CONST.BAS.VARS       THE PROBABILITIES OF EACH OF
       THE TIME-INVARIANT'
[16]   PRINT '                         BASIC VARIABLES'
[17]   PRINT ''
[18]   PRINT '      NUM.ACC.LEVELS      'HE NUMBER OF ACCOMPLISHMENT
       LEVELS'
[19]   PRINT ''
[20]   PRINT '       NUM.TRAJ.SETS      'HE NUMBER OF TRAJECTORY SETS
       ASSOCIATED WITH'
[21]   PRINT '                         'HE ACCOMPLISHMENT LEVEL UNDER
       CONSIDERATION'
[22]   PRINT ''
[23]   PRINT '                      I    THE INITIAL VECTOR FOR THE
       TRAJECTORY SET'
[24]   PRINT '                         UNDER CONSIDERATION'
[25]   PRINT ''
[26]   PRINT '                      G    THE CHARACTERISTIC MATRICES FOR
       THE TRAJECTORY'
```

```
[27]  PRINT '                              SET UNDER CONSIDERATION'
[28]  PRINT ''
[29]  PRINT '                        F     THE CHARACTERISTIC VECTOR FOR
      THE TRAJECTORY'
[30]  PRINT '                              SET UNDER CONSIDERATION'
[31]  PRINT ''
[32]  PRINT '                        V     THE VECTOR CHARACTERIZING THE
      TIME-INVARIANT '
[33]  PRINT '                              BASIC VARIABLES FOR THE
      TRAJECTORY SET '
[34]  PRINT '                              UNDER CONSIDERATION'
[35]  PRINT ''
[36]  PRINT '                  PERF        THE PERFORMABILITY'
[37]  PRINT ''
[38]  PRINT 'IF AN ITEM IS UNDEFINED WHEN A DISPLAY IS REQUESTED, AN
      ERROR MESSAGE'
[39]  PRINT 'WILL BE PRINTED AND THAT DISPLAY WILL BE
      SUPPRESED.  MORE THAN ONE ITEM '
[40]  PRINT 'MAY BE DISPLAYED WITH A SINGLE DATA COMMAND.'
[41]  PRINT 'EXAMPLE:'
[42]  PRINT ''
[43]  PRINT
      'NUM.PHASES  NUM.STATES  P  H  NUM.CONST.BAS.VARS  PROB.CONST.B
[44]  PRINT 'X
      X                                          X                '
[45]  PRINT 'NUM.ACC.LEVELS  NUM.TRAJ.SETS  I  G  F  V  PERF'
[46]  PRINT
      '  X              X                        X
      '
[47]  PRINT ''
[48]  PRINT 'THIS INFORMS METAPHOR THAT THE NUMBER OF PHASES,
      STATES, AND ACCOMPLISHMENT'
[49]  PRINT 'LEVELS AS WELL AS THE PROBABILITIES OF THE TIME-
      INVARIANT BASIC VARIABLES'
[50]  PRINT 'AND THE PERFORMABILITY ARE TO BE DISPLAYED.'
[51]  PRINT ''
[52]  PRINT 'DO YOU WANT REFERENCES?'
[53]  ⍝ LEAVE IF NOT
[54]  →(~INYES)/0
[55]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[56]  PRINT ''
[57]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[58]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[59]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[60]  PRINT ''
[61]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[62]  PRINT ''
[63]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
```

```
[64]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[65]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[66]  PRINT ''
[67]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[68]  PRINT ''
[69]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[70]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[71]  PRINT ''
      ∇
```

COMMAND EVAL IMPLEMENTATION FUNCTIONS

```
      ∇ GETNUMPHASES
[1]    ⍝
[2]    ⍝
[3]    ⍝   ROUTINE FOR FETCHING THE NUMBER OF MISSION PHASES
[4]    ⍝
[5]    ⍝  INPUT NUMBER OF PHASES
[6]    PRINT ''
[7]   GNPIN:PRINT 'NUMBER OF PHASES?'
[8]    1 INPUT GNP
[9]    ⍝ CHECK FOR COMMAND
[10]   →(1=∧ ∈'COMMAND')/GNPIN
[11]   ⍝ CHECK VALIDITY OF INPUT
[12]   →(CHECKPOSI IN)/GNPIN
[13]   ⍝ ELSE SET THE NUMBER OF PHASES AND EXIT
[14]   NUMPHASES←IN
[15]   DEFNUMPHASES←1
       ∇
```

```
      ∇ GNPINFO
[1]    ⍝
[2]    ⍝  ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF PHASES
[3]    ⍝
[4]    PRINT 'ENTER THE NUMBER OF PHASES IN THE FINITE PHASE MODEL AS
       A SINGLE'
[5]    PRINT 'POSITIVE INTEGER.'
[6]    PRINT 'EXAMPLE:'
[7]    PRINT '      3'
[8]    PRINT 'THIS  INDICATES  TO  METAPHOR  THAT  THE  MODEL  TO  BE
       EVALUATED HAS 3 PHASES'
[9]    PRINT 'DO YOU WANT REFERENCES?'
[10]   ⍝ LEAVE IF NOT
[11]   →(~INYES)/0
[12]   PRINT  'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
       ANALYSIS, SEE'
[13]   PRINT ''
[14]   PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
```

```
      THE '
[15]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[16]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[17]  PRINT ''
[18]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[19]  PRINT ''
[20]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[21]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[22]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[23]  PRINT ''
[24]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[25]  PRINT ''
[26]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[27]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[28]  PRINT ''
     ∇
```

```
     ∇ GETSTATES
[1]   ⍝
[2]   ⍝   ROUTINE FOR FETCHING THE NUMBER OF STATES IN EACH PHASE
[3]   ⍝
[4]   ⍝INPUT NUMBER OF STATES AND CHECK VALIDITY
[5]   PRINT ''
[6]  GSIN:PRINT 'NUMBER OF STATES PER PHASE? (SPACE BETWEEN EACH
      NUMBER)'
[7]   NUMPHASES INPUT GS
[8]   ⍝ CHECK FOR COMMAND
[9]   →(IN=¯1)/GSIN
[10]  ⍝ CHECK FOR POSITIVE INTEGER
[11]  →(CHECKPOSI IN)/GSIN
[12]  ⍝ SET STATES AND EXIT
[13]  STATESPERPHASE←IN
[14]  MAXNUMSTATES←⌈/STATESPERPHASE
[15]  DEFNUMSTATES←1
     ∇
```

```
[1]   ⍝
[2]   ⍝   ROUTINE FOR GIVING HELP ON INPUTTING THE NUMBER OF STATES
      IN EACH PHASE
[3]   ⍝
[4]   PRINT 'ENTER THE NUMBER OF STATES FOR EACH PHASE IN THE FINITE
      PHASE MODEL.'
[5]   PRINT 'TYPE A POSITIVE INTEGER FOR EACH PHASE, SEPARATING EACH
      WITH SPACES AND/OR COMMAS.'
[6]   PRINT 'THE NUMBER OF STATES MUST BE A POSITIVE INTEGER.'
[7]   PRINT 'EXAMPLE:'
```

```
[8]   PRINT '          4   3,5'
[9]    PRINT  'THIS INDICATES TO METAPHOR THAT THE FIRST PHASE HAS 4
      STATES, THE SECOND'
[10]  PRINT  'PHASE  HAS  3  STATES,  AND  THE  THIRD  PHASE  HAS  5
      STATES.'
[11]  PRINT  'METAPHOR CHECKS TO MAKE SURE THAT THE NUMBER OF GROUPS
      OF STATES'
[12]  PRINT 'MATCHES THE NUMBER OF PHASES INPUT  EARLIER.  AN  ERROR
      MESSAGE WILL BE'
[13]  PRINT 'PRINTED IF THEY DO NOT MATCH.'
[14]  PRINT 'DO YOU WANT REFERENCES?'
[15]  ⍝ LEAVE IF NOT
[16]  →(~INYES)/0
[17]  PRINT  'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[18]  PRINT ''
[19]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
      THE '
[20]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[21]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[22]  PRINT ''
[23]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[24]  PRINT ''
[25]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
      THE '
[26]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[27]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[28]  PRINT ''
[29]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[30]  PRINT ''
[31]  PRINT '    S. PAKIN,  ''APL\360  REFERENCE  MANUAL,''  SCIENCE
      RESEARCH'
[32]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[33]  PRINT ''
      ∇


      ∇ GETPMATRICES;PHASE;NEXTP
[1]    ⍝
[2]    ⍝   ROUTINE FOR INPUTTING THE P MATRICES
[3]    ⍝
[4]    ⍝   ONE MATRIX FOR EACH PHASE
[5]    ⍝
[6]    ⍝  INITIALIZE THE ARRAY OF P MATRICES
[7]    P←(NUMPHASES,MAXNUMSTATES,MAXNUMSTATES)⍴0
[8]    ⍝
[9]    ⍝  INPUT AND CHECK THE P MATRICES
[10]  PRINT ''
[11]  PRINT ''
[12]  PRINT  'SPECIFY  THE  P  MATRICES FOR EACH PHASE, 1 PHASE AT A
```

```
      TIME'
[13]  ⍝
[14]  ⍝
[15]  ⍝  INITIALIZE PHASE COUNTER
[16]  PHASE←1
[17] GPMPHASEIN:PRINT ''
[18]  ⎕←((PRINTQUAD 'PHASE ');(PRINTQUAD PHASE);PRINTQUAD ':')
[19]  PRINT ''
[20]  ⍝  GET P MATRIX FOR PHASE
[21]  NEXTP←GENERATEPMATRIX STATESPERPHASE[PHASE]
[22]  ⍝ INSERT THE MATRIX INTO THE ARRAY OF MATRICES
[23]  P[PHASE;⍳STATESPERPHASE[PHASE];
      ⍳STATESPERPHASE[PHASE]]←NEXTP
[24]  ⍝
[25]  ⍝  INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[26]  PHASE←PHASE+1
[27]  →(PHASE≤NUMPHASES)/GPMPHASEIN
[28]  ⍝ ELSE SET P DEFINITION FLAG AND LEAVE
[29]  DEFP←1
      ∇


      ∇ GETHMATRICES;PHASE;NEXTH
[1]   ⍝
[2]   ⍝    ROUTINE FOR INPUTTING THE H MATRICES
[3]   ⍝
[4]   ⍝    ONE MATRIX FOR EACH PHASE
[5]   ⍝
[6]   ⍝ IF NO H MATRICES, SET H←1 AND LEAVE
[7]   →(NUMPHASES≥2)/GHMMULTIPHASE
[8]   H← 1 1 ,1⍴1
[9]   ⍝  INITIALIZE THE ARRAY OF H MATRICES
[10] GHMMULTIPHASE:H←((NUMPHASES-1),MAXNUMSTATES, MAXNUMSTATES)⍴0
[11]  ⍝
[12]  ⍝  INPUT AND CHECK THE H MATRICES
[13]  PRINT ''
[14]  PRINT ''
[15]  PRINT 'SPECIFY THE H MATRICES FOR EACH PHASE,  1  PHASE  AT  A
      TIME'
[16]  ⍝
[17]  ⍝
[18]  ⍝  INITIALIZE PHASE COUNTER
[19]  PHASE←2
[20] GHMPHASEIN:PRINT ''
[21]  ⎕←((PRINTQUAD    'PHASE    ');(PRINTQUAD   PHASE-1);(PRINTQUAD
      '-');(PRINTQUAD PHASE);PRINTQUAD ':')
[22]  PRINT ''
[23]  ⍝  GET H MATRIX FOR PHASE
[24]  NEXTH←STATESPERPHASE[PHASE-1]GENERATEHMATRIX
      STATESPERPHASE[PHASE]
[25]  ⍝ INSERT THE MATRIX INTO THE ARRAY OF MATRICES
[26]  H[PHASE-1;⍳STATESPERPHASE[PHASE-1];
```

```
      ιSTATESPERPHASE[PHASE]]←NEXTH
[27]  ⍝
[28]  ⍝  INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[29]  PHASE←PHASE+1
[30]  →(PHASE≤NUMPHASES)/GHMPHASEIN
[31]  ⍝ ELSE SET H DEFINITION FLAG AND LEAVE
[32]  DEFH←1
      ∇


      ∇ GETNUMBASICVARIABLES
[1]   ⍝
[2]   ⍝
[3]   ⍝   ROUTINE FOR FETCHING THE NUMBER OF  TIME-INVARIANT)  BASIC
      VARIABLES
[4]   ⍝
[5]   ⍝  INPUT NUMBER OF TIME-INVARIANT BASIC VARIABLES
[6]   PRINT ''
[7]   GNBVIN:PRINT 'NUMBER OF TIME-INVARIANT BASIC VARIABLES?'
[8]   1 INPUT GNBV
[9]   ⍝ CHECK FOR COMMAND
[10]  →(1=∧
      ∈'COMMAND')/GNBVIN
[11]  ⍝ CHECK VALIDITY OF INPUT
[12]  →(IN=0)/GNBVSET
[13]  →(CHECKPOSI IN)/GNBVIN
[14]  ⍝ ELSE SET THE NUMBER OF TIME-INVARIANT BASIC VARIABLES AND
      EXIT
[15]  GNBVSET:NUMBASICVARIABLES←IN
[16]  DEFNUMBASICVARIABLES←1
      ∇


      ∇ GNBVINFO
[1]   ⍝
[2]   ⍝  ROUTINE GIVING HELP ON INPUTTING BASIC VARIABLES
[3]   ⍝
[4]   PRINT 'ENTER THE NUMBER OF BASIC VARIABLES WHOSE PROBABILITIES
      REMAIN TIME-INVARIANT'
[5]   PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF
      TIME-INVARIANT BASIC VARIABLES.)'
[6]   PRINT 'THE NUMBER SHOULD BE A SINGLE POSITIVE INTEGER.'
[7]   PRINT 'EXAMPLE:'
[8]   PRINT '      2'
[9]   PRINT 'THIS INFORMS METAPHOR THAT TWO TIME-INVARIANT BASIC
      VARIABLES ARE CONSIDERED IN THE'
[10]  PRINT 'MODEL.'
[11]  PRINT ''
[12]  PRINT 'DO YOU WANT REFERENCES?'
[13]  ⍝ LEAVE IF NOT
[14]  →(~INYES)/0
[15]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
```

```
      ANALYSIS, SEE'
[16]  PRINT ''
[17]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[18]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[19]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[20]  PRINT ''
[21]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[22]  PRINT ''
[23]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[24]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[25]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[26]  PRINT ''
[27]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[28]  PRINT ''
[29]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[30]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[31]  PRINT ''
     ∇
```

```
     ∇ GETBASICVARIABLES
[1]   ⍝
[2]   ⍝   ROUTINE FOR FETCHING THE PROBABILITIES OF EACH OF THE
      TIME-INVARIANT BASIC VARIABLES
[3]   ⍝
[4]   ⍝  SEE IF IT IS NECESSARY TO INPUT BASIC VARIABLES
[5]   →(NUMBASICVARIABLES=0)/0
[6]   ⍝
[7]   ⍝INPUT PROBABILITIES AND CHECK VALIDITY
[8]   PRINT ''
[9]  GBVIN:PRINT 'PROBABILITIES OF EACH TIME-INVARIANT BASIC
      VARIABLE?  (SPACE BETWEEN EACH NUMBER)'
[10]  NUMBASICVARIABLES INPUT GBV
[11]  ⍝ CHECK FOR COMMAND
[12]  →(1=∧ ∈'COMMAND')/GBVIN
[13]  ⍝ CHECK FOR PROPER PROBABILITY MAGNITUDE
[14]  →(CHECKPROB IN)/GBVIN
[15]  ⍝ SET BASIC VARIABLE PROBABILITIES AND EXIT
[16]  BASICVARIABLES←IN,1-IN
[17]  DEFBASICVARIABLES←1
     ∇
```

```
     ∇ GBVINFO
[1]  ⍝
[2]   ⍝ ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF BASIC
      VARIABLES
[3]  ⍝
[4]  PRINT 'ENTER THE PROBABILITIES OF THE BASIC VARIABLES WHOSE
      PROBABILITIES REMAIN CONSTANT'
[5]   PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF
      TIME-INVARIANT BASIC VARIABLES.)'
[6]  PRINT 'THE PROBABILITIES SHOULD BE ENTERED AS A ROW OF
      POSITIVE NUMBERS'
[7]   PRINT 'BETWEEN ZERO AND ONE, INCLUSIVE. THE NUMBERS SHOULD
      BE'
[8]  PRINT 'SEPARATED BY SPACES AND/OR COMMAS. THE ORDER OF THE
      NUMBERS SHOULD'
[9]   PRINT 'CORRESPOND TO THE ORDER OF THE TIME-INVARIANT BASIC
      VARIABLE VECTORS'
[10] PRINT 'WHICH WILL BE ASKED FOR LATER.'
[11] PRINT 'EXAMPLE:'
[12] PRINT '      .2,  0.3  .4,0.1'
[13] PRINT 'THIS INFORMS METAPHOR THAT THE PROBABILITIES OF THE
      FOUR TIME-INVARIANT'
[14] PRINT 'BASIC VARIABLES ARE 0.2, 0.3, 0.4, AND 0.1
      RESPECTIVELY. THE NUMBER'
[15] PRINT 'OF TIME-INVARIANT VARIABLES DECLARED EARLIER MUST HAVE
      BEEN FOUR OR AN'
[16] PRINT 'ERROR MESSAGE WILL RESULT.'
[17] PRINT ''
[18] PRINT 'DO YOU WANT REFERENCES?'
[19] ⍝ LEAVE IF NOT
[20] →(~INYES)/0
[21] PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[22] PRINT ''
[23] PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[24] PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[25] PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[26] PRINT ''
[27] PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[28] PRINT ''
[29] PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[30] PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[31] PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[32] PRINT ''
[33] PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[34] PRINT ''
[35] PRINT '    S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
```

```
[36]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[37]  PRINT ''
      ∇

      ∇ GETNUMACCLEV
[1]    ⍝
[2]    ⍝
[3]    ⍝   ROUTINE FOR FETCHING THE NUMBER OF MISSION ACCOMPLISHMENT
       LEVELS
[4]    ⍝
[5]    ⍝  INPUT NUMBER OF ACCOMPLISHMENT LEVELS
[6]    PRINT ''
[7]   GNAIN:PRINT 'NUMBER OF ACCOMPLISHMENT LEVELS?'
[8]    1 INPUT GNA
[9]    ⍝ CHECK FOR COMMAND
[10]   →(1=∧
       ∈'COMMAND')/GNAIN
[11]   ⍝ CHECK VALIDITY OF INPUT
[12]   →(CHECKPOSI IN)/GNAIN
[13]   ⍝ ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT
[14]   NUMACCLEV←IN
[15]   DEFNUMACCLEV←1
       ∇

      ∇ GNAINFO
[1]    ⍝
[2]    ⍝  ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF
       ACCOMPLISHMENT LEVELS
[3]    ⍝
[4]    PRINT 'ENTER THE NUMBER OF ACCOMPLISHMENT LEVELS FOR THIS
       MODEL AS A SINGLE POSITIVE INTEGER.'
[5]    PRINT 'EXAMPLE:'
[6]    PRINT '          5'
[7]    PRINT 'THIS INDICATES TO METAPHOR THAT THE MODEL IT IS
       EVALUATING HAS 5 ACCOMPLISHMENT LEVELS.'
[8]    PRINT ''
[9]    PRINT 'DO YOU WANT REFERENCES?'
[10]   ⍝ LEAVE IF NOT
[11]   →(~INYES)/0
[12]   PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
       ANALYSIS, SEE'
[13]   PRINT ''
[14]   PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
       THE '
[15]   PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
       GRANT'
[16]   PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[17]   PRINT ''
[18]   PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[19]   PRINT ''
```

```
[20]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[21]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[22]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[23]  PRINT ''
[24]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[25]  PRINT ''
[26]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[27]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[28]  PRINT ''
      ∇
```

```
      ∇ GETPERFORMABILITY;LEVELPROB;L
[1]    ⍝
[2]    ⍝   ROUTINE FOR COMPUTING THE PERFORMABILITY OF THE SYSTEM
[3]    ⍝
[4]    ⍝  INITIALIZE THE PERFORMABILITY VECTOR AND SET DEFINITION
       FLAGS
[5]    PERFORMABILITY←NUMACCLEV⍴0
[6]    DEFACCLEVEL←1
[7]    DEFPERFORMABILITY←1
[8]    ⍝
[9]    ⍝  LOOP THROUGH EACH ACCOMPLISHMENT LEVEL
[10]   ⍝ INITIALIZE LEVEL COUNTER
[11]   L←1
[12]   ⍝  GET THE PROBABILITY OF EACH ACCOMPLISHMENT LEVEL
[13]  GPLOOP:LEVELPROB←GETACCLEVPROB L
[14]   ⍝  INSERT THE PROBABILITY INTO THE PERFORMABILITY VECTOR
[15]   PERFORMABILITY[L]←LEVELPROB
[16]   ⍝  INCREMENT THE LEVEL COUNTER AND BRANCH IF NECESSARY
[17]   L←L+1
[18]   →(L≤NUMACCLEV)/GPLOOP
[19]   ⍝ ELSE EXIT ROUTINE, SETTING DEFINITION FLAG
[20]   DEFACCLEVEL←0
```

MATRIX GENERATOR FUNCTIONS

```
     ∇ Z←M GENERATEHMATRIX N;TYPE;IN
[1]   ⍝
[2]   ⍝
[3]   ⍝   ROUTINE FOR SUPERVISING THE CONSTRUCTION OF THE H MATRIX
      FOR THE GIVEN PHASE
[4]   ⍝    THE MATRIX IS TO HAVE SHAPE M×N
[5]   ⍝
[6]   ⍝  GET TYPE OF MATRIX
[7]   PRINT ''
[8]  GENHMIN:PRINT 'WHAT TYPE OF H MATRIX?'
[9]   8 INPUT GHM
[10]  ⍝ CHECK FOR COMMAND
[11]  →(1=∧ ∈'COMMAND')/GENHMIN
[12]  ⍝  CHECK FOR TYPE
[13]  →(∧/HMATRIXLIST∈(IN))/GENHMGIVEN,GENHMIDENTITY
[14]  ⍝  ELSE ILLEGAL TYPE
[15]  PRINT 'ILLEGAL H MATRIX TYPE.  TYPE HELP FOR INFORMATION'
[16]  →GENHMIN
[17]  ⍝
[18]  ⍝
[19]  ⍝  USER WILL GIVE H MATRIX VALUES
[20] GENHMGIVEN:Z←M GGIVEN N
[21]  →0
[22]  ⍝
[23]  ⍝
[24]  ⍝  IDENTITY MATRIX GENERATOR
[25]  ⍝  FIRST CHECK TO MAKE SURE AN IDENTITY MATRIX IS  APPROPRIATE
      HERE
[26] GENHMIDENTITY:→(M=N)/GENHMIDENTITYGET
[27]  PRINT  'THESE  TWO  PHASES  DO  NOT  HAVE  THE  SAME NUMBER OF
      STATES.'
[28]  PRINT 'AN IDENTITY MATRIX IS INAPPROPRIATE FOR THE  INTERPHASE
      TRANSITION MATRIX.'
[29]  →GENHMIN
[30] GENHMIDENTITYGET:Z←GIDENTITY N
[31]  →0
     ∇


     ∇ GHMINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING GIVEN H MATRICES
[3]   ⍝
[4]   PRINT 'TYPE ONE OF:   GIVEN, IDENTITY'
[5]   PRINT 'DO YOU WANT MORE HELP?'
[6]   ⍝LEAVE IF NO MORE HELP WANTED
[7]   →(~INYES)/0
[8]    PRINT  'ENTER  ONE  OF THE FOLLOWING TYPES FOR THE INTERPHASE
      TRANSITION (H) MATRIX:'
```

```
[9]   PRINT ''
[10]  PRINT '     GIVEN      YOU WILL INPUT AN H MATRIX, ONE ROW  AT
      A TIME.'
[11]  PRINT ''
[12]  PRINT '      IDENTITY METAPHOR WILL GENERATE AN H MATRIX
      ASSUMING THE MATRIX'
[13]  PRINT '                  REPRESENTS A SYSTEM IN WHICH THERE IS NO
      STATE CHANGE'
[14]  PRINT  '                      BETWEEN  PHASES.  THUS,   METAPHOR
      GENERATES AN IDENTITY MATRIX.'
[15]  PRINT ''
[16]  PRINT ''
[17]  PRINT 'DO YOU WANT REFERENCES?'
[18]  ⍝ LEAVE IF NOT
[19]  →(~INYES)/0
[20]  PRINT  'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[21]  PRINT ''
[22]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
      THE '
[23]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[24]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[25]  PRINT ''
[26]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[27]  PRINT ''
[28]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
      THE '
[29]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[30]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[31]  PRINT ''
[32]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[33]  PRINT ''
[34]  PRINT '    S. PAKIN,  ''APL\360  REFERENCE  MANUAL,''  SCIENCE
      RESEARCH'
[35]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[36]  PRINT ''
      ∇
```

```
      ∇ Z←GENERATEPMATRIX N;TYPE;IN
[1]    ⍝
[2]    ⍝
[3]     ⍝    ROUTINE FOR SUPERVISING THE CONSTRUCTION OF THE P MATRIX
      FOR THE GIVEN PHASE
[4]    ⍝    THE MATRIX IS TO HAVE SHAPE N×N
[5]    ⍝
[6]    ⍝  GET TYPE OF MATRIX
[7]    PRINT ''
[8]   GENPMIN:PRINT 'WHAT TYPE OF P MATRIX?'
[9]    8 INPUT GPM
```

```
[10]  ⍝ CHECK FOR COMMAND
[11]  →(1=∧
      ∈'COMMAND').GENPMIN
[12]  ⍝  CHECK FOR TYPE
[13]  →(∧/PMATRIXLIST∈(
      IN))/GENPMGIVEN,GENPMDEDFAIL,GENPMNFAIL,GENPMIDENTITY
[14]  ⍝  ELSE ILLEGAL TYPE
[15]  PRINT 'ILLEGAL P MATRIX TYPE.  TYPE HELP FOR INFORMATION'
[16]  →GENPMIN
[17]  ⍝
[18]  ⍝
[19]  ⍝  USER WILL GIVE P MATRIX VALUES
[20] GENPMGIVEN:Z←N GGIVEN N
[21]  →0
[22]  ⍝
[23]  ⍝
[24]  ⍝  DEDICATED COMPONENT SYSTEM
[25] GENPMDEDFAIL:Z←GDEDFAIL N
[26]  ⍝ IF ROUTINE FAILED, GO BACK TO TRY AGAIN.  ELSE EXIT
      ROUTINE
[27]  →((Z[1;1]<0),1)/GENPMIN,0
[28]  ⍝
[29]  ⍝
[30]  ⍝  N GROUPS OF COMPONENTS SYSTEM
[31] GENPMNFAIL:Z←GNFAIL N
[32]  →0
[33]  ⍝
[34]  ⍝
[35]  ⍝  IDENTITY MATRIX GENERATOR
[36] GENPMIDENTITY:Z←GIDENTITY N
[37]  →0
      ∇


      ∇ GPMINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING P MATRICES
[3]   ⍝
[4]   PRINT 'TYPE ONE OF:   GIVEN, DEDFAIL, NFAIL, IDENTITY'
[5]   PRINT 'DO YOU WANT MORE HELP?'
[6]   ⍝LEAVE IF NO MORE HELP WANTED
[7]   →(~INYES)/0
[8]   PRINT 'ENTER ONE OF THE FOLLOWING TYPES FOR THE STATE
      TRANSITION (P) MATRIX:'
[9]   PRINT ''
[10]  PRINT '     GIVEN     YOU WILL INPUT A P MATRIX, ONE ROW AT A
      TIME.'
[11]  PRINT ''
[12]  PRINT '     DEDFAIL   METAPHOR WILL GENERATE A P MATRIX
      ASSUMING THE MATRIX'
[13]  PRINT '               REPRESENTS A SYSTEM HAVING N COMPONENTS,
      EACH FAILING'
```

```
[14]  PRINT '                    INDEPENDENTLY AND EACH
      DISTINGUISHABLE.  THE STATE'
[15]  PRINT '                    OF THE SYSTEM IS THE STATE OF EACH OF
      THE COMPONENTS.'
[16]  PRINT '                    YOU WILL BE ASKED THE LENGTH OF THE
      PHASE AND THE FAILURE'
[17]  PRINT '                    RATE OF THE COMPONENTS.'
[18]  PRINT ''
[19]  PRINT '     NFAIL          METAPHOR WILL GENERATE A P MATRIX
      ASSUMING THE MATRIX'
[20]  PRINT '                    REPRESENTS A SYSTEM HAVING M GROUPS OF
      K(M) COMPONENTS EACH.'
[21]  PRINT '                    THE COMPONENTS FAIL INDEPENDENTLY AND
      THE STATE OF THE'
[22]  PRINT '                    SYSTEM IS THE NUMBER OF ACTIVE (NON-
      FAILED) COMPONENTS'
[23]  PRINT '                    IN EACH GROUP.  YOU WILL BE ASKED THE
      NUMBER OF GROUPS,'
[24]  PRINT '                    THE NUMBER OF COMPONENTS IN EACH GROUP,
      THE LENGTH'
[25]  PRINT '                    OF THE PHASE, AND THE FAILURE RATE OF
      THE COMPONENTS.'
[26]  PRINT ''
[27]  PRINT '     IDENTITY       METAPHOR WILL GENERATE A P MATRIX
      ASSUMING THE MATRIX'
[28]  PRINT '                    REPRESENTS A SYSTEM IN WHICH THERE IS NO
      FAILURE, I.E.,'
[29]  PRINT '                    NO CHANGES IN STATES ARE MADE.  THUS,
      METAPHOR GENERATES'
[30]  PRINT '                    AN IDENTITY MATRIX.'
[31]  PRINT ''
[32]  PRINT ''
[33]  PRINT 'DO YOU WANT REFERENCES?'
[34]  R LEAVE IF NOT
[35]  →(~INYES)/0
[36]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[37]  PRINT ''
[38]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[39]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[40]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[41]  PRINT ''
[42]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[43]  PRINT ''
[44]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[45]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[46]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[47]  PRINT ''
```

```
[48]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[49]  PRINT ''
[50]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[51]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[52]  PRINT ''
      ∇


      ∇ Z←GDEDFAIL N;T;LAMBDA;INDEX;I;J;SIZE;SUCCESS;FAIL
[1]   ⍝
[2]   ⍝   ROUTINE FOR GENERATING THE DEDFAIL TYPE P MATRIX
[3]   ⍝   THE MATRIX IS TO HAVE SHAPE N×N
[4]   ⍝
[5]   ⍝  CHECK TO MAKE SURE THIS ROUTINE IS APPROPRIATE
[6]   →((2⊛N)=⌈(2⊛N))/GDIN
[7]   ⍝  PRINT EXPLANATION, NOTE FAILURE, AND EXIT
[8]   PRINT 'THE NUMBER OF STATES IN DEDFAIL MUST BE A POWER OF
      TWO.'
[9]   ⎕←((PRINTQUAD 'THIS PHASE HAS ');(PRINTQUAD N);PRINTQUAD '
      STATES.')
[10]  PRINT ''
[11]  Z←(N,N)⍴¯1
[12]  →0
[13]  ⍝
[14]  ⍝  GET PHASE LENGTH
[15] GDIN:PRINT 'ENTER PHASE LENGTH'
[16]  1 INPUT GD
[17]  ⍝  CHECK COMMAND
[18]  →(1=∧ ∈'COMMAND')/GDIN
[19]  ⍝  CHECK FOR POSITIVE NUMBER
[20]  →(IN>0)/GDSETT
[21]  PRINT 'LENGTH IN TIME.  MUST BE POSITIVE'
[22]  →GDIN
[23]  ⍝
[24]  ⍝  SET T TO PHASE LENGTH AND GET FAILURE RATE
[25] GDSETT:T←IN
[26] GDINL:PRINT 'ENTER COMPONENT FAILURE RATE'
[27]  1 INPUT GD
[28]  ⍝  CHECK COMMAND
[29]  →(1=∧
      ∈'COMMAND')/GDINL
[30]  ⍝  CHECK FOR POSITIVE NUMBER
[31]  →(IN>0)/GDCHECKSIZE
[32]  PRINT 'RATE IN FAILURES PER UNIT TIME.  MUST BE POSITIVE'
[33]  →GDINL
[34]  ⍝ CHECK REASONABLENESS OF FAILURE RATE
[35] GDCHECKSIZE:→((IN≥1E¯10)∧(IN≤0.1))/GDSETL
[36]  ⍝ PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT
[37]  ⎕←PRINTQUAD IN
[38]  →(IN≥0.1)/GDLBIG
[39]  ⎕←PRINTQUAD 'IS SMALL'
```

```
[40]  →GDLYESNOIN
[41] GDLBIG:⎕←PRINTQUAD 'IS LARGE'
[42] GDLYESNOIN:PRINT ' FOR A FAILURE RATE.  DO YOU WANT THIS
      VALUE?'
[43]  →(~INYES)/GDINL
[44]  ⍝
[45]  ⍝  SET LAMBDA TO FAILURE RATE AND PERFORM CALCULATIONS
[46] GDSETL:LAMBDA←IN
[47]  ⍝
[48]  ⍝  INITIALIZE THE P MATRIX
[49]  Z←(N,N)⍴0
[50]  ⍝  DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX
[51]  INDEX←⍉((2⍴N)⍴2)ENCODE(N-⍳N)
[52]  ⍝
[53]  ⍝  LOOP THROUGH INDEX TO CREATE P
[54]  ⍝ INITIALIZE LOOPS
[55]  I←1
[56]  J←1
[57]  ⍝  DETERMINE THE NUMBER OF SUCCESS TRANSITIONS
[58] GDLOOP:SIZE←+/INDEX[I;]∧INDEX[J;]
[59]  ⍝ FIND THE SUCCESS AND FAILURE PROBABILITIES
[60]  FAIL←(1-*-LAMBDA×T)*((+/INDEX[I;])-SIZE)
[61]  SUCCESS←*-LAMBDA×T×SIZE
[62]  Z[I;J]←SUCCESS×FAIL×(SIZE≥+
      /INDEX[J;])×(∨/INDEX[I;]≥INDEX[J;])
[63]  ⍝ INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE
[64]  J←J+1
[65]  →(J≤N)/GDLOOP
[66]  ⍝ RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF
      APPROPRIATE
[67]  J←1
[68]  I←I+1
[69]  →(I≤N)/GDLOOP
      ∇


      ∇ GDINFO
[1]   ⍝
[2]   ⍝  ROUTINE GIVING HELP ON DEDFAIL TYPE P MATRIX
[3]   ⍝
[4]   PRINT 'METAPHOR WILL GENERATE A P MATRIX ASSUMING THE MATRIX
      REPRESENTS '
[5]   PRINT 'A SYSTEM HAVING N COMPONENTS, EACH FAILING
      INDEPENDENTLY AND EACH '
[6]   PRINT 'DISTINGUISHABLE.  THE FAILURES ARE ALSO ASSUMED TO BE
      '
[7]   PRINT 'POISSON, AND ONCE A COMPONENT HAS FAILED, IT CANNOT'
[8]   PRINT 'BECOME GOOD AGAIN.'
[9]   PRINT 'THE STATE OF THE SYSTEM IS THE STATE OF EACH OF'
[10]  PRINT 'THE COMPONENTS.  THE NUMBER OF STATES DECLARED FOR THE
      PHASE MUST BE '
[11]  PRINT 'A POWER OF TWO.  YOU WILL BE ASKED THE LENGTH OF THE
```

```
      PHASE; ENTER A'
[12]  PRINT 'SINGLE POSITIVE INTEGER.  NEXT YOU WILL BE PROMPTED FOR
      THE FAILURE'
[13]  PRINT 'RATE OF THE COMPONENTS.  AGAIN ENTER A SINGLE POSITIVE
      NUMBER.  IF '
[14]  PRINT 'THIS NUMBER IS NOT BETWEEN 1E¯1 AND 1E¯10, YOU WILL BE
      ASKED FOR CONFIRMATION.'
[15]  PRINT ''
[16]  PRINT 'DO YOU WANT REFERENCES?'
[17]  ⍝ LEAVE IF NOT
[18]  →(~INYES)/0
[19]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[20]  PRINT ''
[21]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[22]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[23]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[24]  PRINT ''
[25]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[26]  PRINT ''
[27]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[28]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[29]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[30]  PRINT ''
[31]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[32]  PRINT ''
[33]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[34]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[35]  PRINT ''
      ∇


      ∇ Z←M GGIVEN N;ROW
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING CONSTANT MATRIX OF PROBABILITIES
[3]   ⍝   EACH ROW MUST SUM TO ONE.  THE MATRIX IS TO HAVE SHAPE
      M×N
[4]   ⍝
[5]   ⍝  INITIALIZE THE MATRIX
[6]   Z←(M,N)⍴0
[7]   ⍝
[8]   ⍝  INPUT AND CHECK THE MATRIX
[9]   PRINT 'ENTER THE MATRIX, 1 ROW AT A TIME'
[10]  ⍝
[11]  ⍝ INITIALIZE ROW COUNTER
[12]  ROW←1
[13]  PRINT ''
```

```
[14] GGIN:⎕←((PRINTQUAD 'ROW ');(PRINTQUAD ROW);PRINTQUAD ':')
[15]  PRINT ''
[16]  ⍝ GET ROW DATA
[17]  N INPUT GG
[18]  ⍝ CHECK FOR COMMAND
[19]  →(1=∧ ∈'COMMAND')/GGIN
[20]  ⍝ CHECK VALIDITY OF INPUT.  FIRST CHECK FOR PROBABILITIES
[21]  →(CHECKPROB IN)/GGIN
[22]  ⍝ CHECK THAT EACH ROW SUMS TO ONE.  IF NOT, ASK AGAIN.
[23]  →((+/IN)=1)/GGINSERT
[24]  PRINT 'THE SUM OF PROBABILITIES IN EACH ROW MUST BE 1'
[25]  →GGIN
[26]  ⍝
[27]  ⍝  INSERT THE ROW INTO THE MATRIX
[28] GGINSERT:Z[ROW;]←IN
[29]  ⍝
[30]  ⍝  INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[31]  ROW←ROW+1
[32]  →(ROW≤M)/GGIN
[33]  ⍝ ELSE LEAVE
      ∇


      ∇ GGINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING GIVEN P MATRICES
[3]   ⍝
[4]   PRINT 'ENTER AN M×N ARRAY, ONE  ROW  AT  A  TIME.  EACH  ENTRY
      MUST'
[5]    PRINT  'BE  BETWEEN 0 AND 1 INCLUSIVE AND THE ENTRIES OF EACH
      ROW MUST SUM'
[6]   PRINT 'TO ONE.  ENTER EACH ROW AS A SERIES OF N  NUMBERS  WITH
      SPACES AND/OR'
[7]   PRINT 'COMMAS BETWEEN EACH.'
[8]   PRINT 'EXAMPLE:'
[9]   PRINT '     .25 0.5,.1 0.15'
[10]  PRINT 'HERE, THE MATRIX HAS FOUR ENTRIES PER ROW.'
[11]  PRINT ''
[12]  PRINT 'DO YOU WANT REFERENCES?'
[13]  ⍝ LEAVE IF NOT
[14]  →(~INYES)/0
[15]  PRINT  'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[16]  PRINT ''
[17]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
      THE '
[18]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[19]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[20]  PRINT ''
[21]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[22]  PRINT ''
```

```
[23]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR  EVALUATING
      THE '
[24]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[25]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[26]  PRINT ''
[27]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[28]  PRINT ''
[29]  PRINT '    S. PAKIN,  ''APL\360  REFERENCE  MANUAL,''  SCIENCE
      RESEARCH'
[30]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[31]  PRINT ''
      ∇


      ∇ Z←GIDENTITY N
[1]    ⍝
[2]    ⍝  ROUTINE FOR GENERATING AN N×N IDENTITY MATRIX
[3]    ⍝
[4]    ⍝ RETURN THE MATRIX
[5]    Z←(N,N)⍴(1,(N⍴0))
      ∇


      ∇ Z←GNFAIL N;T;LAMBDA;GROUPS;NUM;INDEX;I;J;COEFF;FAIL;SUCCESS
[1]    ⍝
[2]    ⍝   ROUTINE FOR GENERATING THE NFAIL TYPE P MATRIX
[3]    ⍝   THE MATRIX IS TO HAVE SHAPE N×N
[4]    ⍝
[5]    ⍝  GET PHASE LENGTH
[6]   GNINT:PRINT 'ENTER PHASE LENGTH'
[7]    1 INPUT GN
[8]    ⍝  CHECK COMMAND
[9]    →(1=∧
       ∈'COMMAND')/GNINT
[10]   ⍝  CHECK FOR POSITIVE NUMBER
[11]   →(IN>0)/GNSETT
[12]   PRINT 'LENGTH IN TIME.  MUST BE POSITIVE'
[13]   →GNINT
[14]   ⍝
[15]   ⍝  SET T TO PHASE LENGTH AND GET FAILURE RATE
[16]  GNSETT:T←IN
[17]  GNINL:PRINT 'ENTER COMPONENT FAILURE RATE'
[18]   1 INPUT GN
[19]   ⍝  CHECK COMMAND
[20]   →(1=∧ ∈'COMMAND')/GNINL
[21]   ⍝  CHECK FOR POSITIVE NUMBER
[22]   →(IN>0)/GNCHECKSIZE
[23]   PRINT 'RATE IN FAILURES PER UNIT TIME.  MUST BE POSITIVE'
[24]   →GNINL
[25]   ⍝ CHECK REASONABLENESS OF FAILURE RATE
[26]  GNCHECKSIZE:→((IN≥1E¯10)∧(IN≤0.1))/GNSETL
```

```
[27]   ⍝ PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT
[28]   ⎕←PRINTQUAD IN
[29]   →(IN≥0.1)/GNLBIG
[30]   ⎕←PRINTQUAD 'IS SMALL'
[31]   →GNLYESNOIN
[32]  GNLBIG:⎕←PRINTQUAD 'IS LARGE'
[33]   GNLYESNOIN:PRINT ' FOR A FAILURE RATE. DO YOU WANT THIS
       VALUE?'
[34]   →(~INYES)/GNINL
[35]   ⍝
[36]   ⍝ SET LAMBDA TO FAILURE RATE AND GET NUMBER OF GROUPS
[37]  GNSETL:LAMBDA←IN
[38]  GNING:PRINT 'ENTER NUMBER OF GROUPS'
[39]   1 INPUT GN
[40]   ⍝ CHECK COMMAND
[41]   →(1=∧
       ∈'COMMAND')/GNING
[42]   ⍝ CHECK FOR POSITIVE INTEGER
[43]   →(CHECKPOSI IN)/GNING
[44]   ⍝
[45]   ⍝ SET GROUPS TO NUMBER OF GROUPS AND GET COMPONENTS.PER
       GROUP
[46]   GROUPS←IN
[47]  GNINN:PRINT 'ENTER NUMBER OF COMPONENTS PER GROUP (SPACE
       BETWEEN EACH NUMBER):'
[48]   GROUPS INPUT GN
[49]   ⍝ CHECK COMMAND
[50]   →(1=∧ ∈'COMMAND')/GNINN
[51]   ⍝ CHECK FOR POSITIVE INTEGER
[52]   →(CHECKPOSI IN)/GNINN
[53]   ⍝ CHECK FOR THE PROPER NUMBER OF STATES
[54]   →(N=×/IN+1)/GNCALC
[55]   ⎕←((PRINTQUAD 'THERE ARE ');(PRINTQUAD N);PRINTQUAD ' STATES
       IN THIS PHASE. THE PRODUCT OF [EACH COMPONENT')
[56]   PRINT ''
[57]   PRINT 'NUMBER PLUS 1] MUST BE THE NUMBER OF STATES.'
[58]   PRINT 'HOW MANY GROUPS'
[59]   →GNING
[60]   ⍝
[61]   ⍝ SET NUM TO THE NUMBER OF POSITIONS FOR THE COMPONENTS IN
       THE GROUPS AND CALCULATE THE P MATRIX
[62]  GNCALC:NUM←IN+1
[63]   ⍝
[64]   ⍝ INITIALIZE THE P MATRIX
[65]   Z←(N,N)⍴0
[66]   ⍝ DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX
[67]   INDEX←⍉(NUM)ENCODE(N-⍳N)
[71]   ⍝
[72]   ⍝ LOOP THROUGH INDEX TO CREATE P
[73]   ⍝ INITIALIZE LOOPS
[74]  I←1
[75]   J←1
```

```
[76]  ⍝ FIRST DETERMINE THE PROPER COEFFICIENT FOR THE TERM UNDER
      CONSIDERATION
[77] GNLOOP:COEFF←×(INDEX[J;]!INDEX[I;])
[78]  ⍝ MULTIPLY THE COEFFICIENT WITH THE PROPER EXPONENTIALS
[79]  FAIL←(1-*-LAMBDA×T)*(+/INDEX[I;])-+/INDEX[J;]
[80]  SUCCESS←*-LAMBDA×T×(+/INDEX[J;])
[81]  Z[I;J]←COEFF×FAIL×SUCCESS
[82]  ⍝ INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE
[83]  J←J+1
[84]  →(J≤N)/GNLOOP
[85]  ⍝ RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF
      APPROPRIATE
[86]  J←1
[87]  I←I+1
[88]  →(I≤N)/GNLOOP
      ∇
```

```
      ∇ GNINFO
[1]   ⍝
[2]   ⍝ ROUTINE GIVING HELP ON NFAIL TYPE P MATRIX
[3]   ⍝
[4]   PRINT 'METAPHOR WILL GENERATE A P MATRIX ASSUMING THE MATRIX
      REPRESENTS A'
[5]   PRINT 'SYSTEM HAVING M GROUPS OF K(N) COMPONENTS EACH, WHERE K
      IS A '
[6]     PRINT 'FUNCTION OF THE GROUP. THE COMPONENTS FAIL
      INDEPENDENTLY'
[7]   PRINT 'AND ARE ASSUMED TO HAVE A POISSON DISTRIBUTION. ALSO,
      ONCE'
[8]   PRINT 'A COMPONENT HAS FAILED, IT CANNOT BECOME GOOD AGAIN.'
[9]    PRINT 'THE STATE OF THE SYSTEM IS THE NUMBER OF ACTIVE
      COMPONENTS'
[10]  PRINT 'IN EACH GROUP. THE NUMBER OF STATES DECLARED'
[11]  PRINT 'FOR THE PHASE MUST BE THE PRODUCT OF [THE NUMBER OF
      COMPONENTS '
[12]  PRINT 'IN EACH GROUP PLUS ONE]. FOR EXAMPLE, IF THE SYSTEM
      HAS 3 GROUPS'
[13]  PRINT 'CONTAINING RESPECTIVELY 2, 5, AND 7 COMPONENTS, THEN
      THE PHASE HAS'
[14]  PRINT '(2+1)×(5+1)×(7+1)←144 STATES. '
[15]  PRINT ''
[16]  PRINT 'YOU WILL BE ASKED THE LENGTH OF THE PHASE; ENTER A
      SINGLE POSITIVE INTEGER.'
[17]  PRINT 'NEXT YOU WILL BE PROMPTED FOR THE FAILURE RATE OF THE
      COMPONENTS. '
[18]  PRINT 'AGAIN ENTER A SINGLE POSITIVE NUMBER. IF THIS NUMBER
      IS NOT BETWEEN'
[19]  PRINT '1E¯1 AND 1E¯10, YOU WILL BE ASKED FOR
      CONFIRMATION. YOU WILL THEN '
[20]  PRINT 'BE ASKED THE NUMBER OF GROUPS; ENTER THIS AS A SINGLE
      POSITIVE INTEGER.'
```

```
[21]  PRINT 'FINALLY, METAPHOR WILL REQUEST THE NUMBER OF COMPONENTS
      IN EACH GROUP.'
[22]  PRINT 'THIS SHOULD BE INPUT AS A ROW OF POSITIVE INTEGERS
      SEPARATED BY '
[23]  PRINT 'SPACES OR COMMAS.'
[24]  PRINT ''
[25]  PRINT 'DO YOU WANT REFERENCES?'
[26]  ⍝ LEAVE IF NOT
[27]  →(~INYES)/0
[28]  PRINT  'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[29]  PRINT ''
[30]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[31]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[32]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[33]  PRINT ''
[34]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[35]  PRINT ''
[36]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[37]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[38]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[39]  PRINT ''
[40]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[41]  PRINT ''
[42]  PRINT '   S. PAKIN,  ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[43]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[44]  PRINT ''
      ∇
```

TRAJECTORY SET EVALUATION FUNCTIONS

```
      ∇ Z←GETACCLEVPROB LEVEL;T;TRAJPROB;NUMTRAJSETS;I;G;F;V
[1]    ⍝
[2]      ⍝     ROUTINE   FOR   DETERMINING   THE   PROBABILITY   OF   AN
      ACCOMPLISHMENT LEVEL
[3]    ⍝
[4]    ⍝ INITIALIZE THE COLLECTION VARIABLE
[5]    Z←0
[6]    ⍝
[7]    ⍝  GET THE NUMBER OF TRAJECTORY SETS
[8]    NUMTRAJSETS←GETNUMTRAJSETS LEVEL
[9]     ⍝  LOOP  THROUGH  FOR  EACH  SET,  GETTING  ITS  VALUES  AND
      CALCULATING ITS PROBABILITY
[10]   ⍝  INITIALIZE COUNTER
[11]   T←1
[12] GAPTLOOP:⍞←((PRINTQUAD 'TRAJECTORY SET ');PRINTQUAD T)
```

```
[13]  PRINT ''
[14]  I←GETIVECTOR
[15]  G←GETGMATRICES
[16]  F←GETFVECTOR
[17]  V←GETVVALUES
[18]  TRAJPROB←CALCTRAJPROB
[19]  ⍝
[20]  ⍝  ADD THE PROBABILITY TO THE COLLECTION VARIABLE
[21]  Z←Z+TRAJPROB
[22]  ⍝
[23]  ⍝  INCREMENT TRAJECTORY COUNTER AND BRANCH IF NECESSARY
[24]  T←T+1
[25]  →(T≤NUMTRAJSETS)/GAPTLOOP
[26]  ⍝ RESET DEFINITION FLAGS
[27]  DEFNUMTRAJSETS←DEFI←DEFG←DEFF←DEFV←0
      ∇

      ∇ Z←GETNUMTRAJSETS L
[1]   ⍝
[2]   ⍝
[3]   ⍝    ROUTINE  FOR FETCHING THE NUMBER OF ACCOMPLISHMENT LEVEL
      TRAJECTORY SETS FOR LEVEL L
[4]   ⍝
[5]   ⍝  INPUT NUMBER OF TRAJECTORY SETS
[6]   PRINT ''
[7]   ⎕←((PRINTQUAD 'ACCOMPLISHMENT LEVEL ');PRINTQUAD L-1)
[8]   PRINT ''
[9]   GNTSIN:PRINT 'NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT
      LEVEL?'
[10]  1 INPUT GNTS
[11]  ⍝ CHECK FOR COMMAND
[12]  →(1=∧
      ∈'COMMAND')/GNTSIN
[13]  ⍝ CHECK VALIDITY OF INPUT
[14]  →(CHECKPOSI IN)/GNTSIN
[15]  ⍝ ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT
[16]  Z←IN
[17]  DEFNUMTRAJSETS←1
      ∇

      ∇ GNTSINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF
      TRAJECTORIES IN A GIVEN ACCOMPLISHMENT LEVEL
[3]   ⍝
[4]   PRINT 'ENTER THE NUMBER OF TRAJECTORY SETS ASSOCIATED WITH
      THIS ACCOMPLISHMENT'
[5]   PRINT 'LEVEL AS A SINGLE POSITIVE INTEGER.'
[6]   PRINT 'EXAMPLE:'
[7]   PRINT '              4'
```

```
[8]   PRINT 'THIS INFORMS METAPHOR THAT THE ACCOMPLISHMENT LEVEL HAS
      FOUR TRAJECTORY'
[9]   PRINT 'SETS DESCRIBING IT.'
[10]  PRINT ''
[11]  PRINT 'DO YOU WANT REFERENCES?'
[12]  ⍝ LEAVE IF NOT
[13]  →(~INYES)/0
[14]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[15]  PRINT ''
[16]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[17]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[18]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[19]  PRINT ''
[20]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[21]  PRINT ''
[22]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[23]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[24]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[25]  PRINT ''
[26]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[27]  PRINT ''
[28]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[29]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[30]  PRINT ''
      ∇


      ∇ Z←GETIVECTOR
[1]    ⍝
[2]    ⍝   ROUTINE FOR INPUTTING INITIAL STATE VECTOR
[3]    ⍝
[5]   GIVIN:PRINT 'ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):'
[6]    STATESPERPHASE[1]INPUT GIV
[7]    ⍝ CHECK FOR COMMAND
[8]    →(1=∧ ∈'COMMAND')/GIVIN
[9]    ⍝ CHECK VALIDITY OF INPUT.
[10]   →(CHECKBIN IN)/GIVIN
[11]   ⍝  SET I VECTOR AND LEAVE
[12]   Z←MAXNUMSTATES⍴IN,(MAXNUMSTATES⍴0)
[13]   DEFI←1
      ∇
```

```
          ∇ GIVINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING I VECTORS
[3]   ⍝
[4]   PRINT 'ENTER THE INITIAL PROBABILITY DISTRIBUTION FOR THIS
      TRAJECTORY.'
[5]    PRINT 'TYPE A PROBABILITY BETWEEN ZERO AND ONE INCLUSIVE
      CORRESPONDING TO EACH'
[6]   PRINT 'STATE''S INITIAL PROBABILITY. SEPARATE EACH NUMBER
      WITH SPACES AND/OR COMMAS.'
[7]   PRINT 'THE ORDER OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER
      OF THE INITIAL STATES.'
[8]    PRINT 'THE NUMBER OF ENTRIES SHOULD BE THE SAME AS THE NUMBER
      OF STATES'
[9]   PRINT 'IN THE FIRST PHASE MODEL.'
[10]  PRINT 'EXAMPLE:'
[11]  PRINT '          .3  0.5,.2'
[12]  PRINT 'THIS INFORMS METAPHOR THAT FOR THE TRAJECTORY SET UNDER
      CONSIDERATION, THE'
[13]  PRINT 'PROBABILITY THE SYSTEM BEGINS IN THE FIRST STATE OF
      PHASE 1 IS 0.2, FOR THE'
[14]  PRINT 'SECOND STATE, THE PROBABILITY IS 0.5, AND FOR THE THIRD
      STATE, THE PROBABILITY'
[15]  PRINT 'IS 0.2.'
[16]  PRINT ''
[17]  PRINT 'DO YOU WANT REFERENCES?'
[18]  ⍝ LEAVE IF NOT
[19]  →(~INYES)/0
[20]  PRINT  'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[21]  PRINT ''
[22]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[23]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[24]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[25]  PRINT ''
[26]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[27]  PRINT ''
[28]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[29]  PRINT  '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[30]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[31]  PRINT ''
[32]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[33]  PRINT ''
[34]  PRINT '    S. PAKIN,  ''APL\360  REFERENCE  MANUAL,''  SCIENCE
      RESEARCH'
[35]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[36]  PRINT ''
     ∇
```

```
     ∇ Z←GETGMATRICES;PHASE
[1]    ⍝
[2]    ⍝    ROUTINE FOR INPUTTING THE G MATRICES FOR A TRAJECTORY
       SET
[3]    ⍝ EACH G MATRIX WILL BE REPRESENTED AS A VECTOR OF ITS
       DIAGONAL ELEMENTS
[4]    ⍝
[5]    ⍝ INITIALIZE THE G DIAGONAL MATRIX. ALSO SET DEFINITION
       FLAG
[6]    Z←(MAXNUMSTATES,MAXNUMSTATES)⍴0
[7]    DEFG←1
[8]    ⍝
[9]    ⍝ LOOP THROUGH EACH PHASE EXCEPT THE LAST, GETTING THE G
       MATRICES
[10]   ⍝ INITIALIZE THE PHASE COUNTER
[11]   PHASE←1
[12]   ⍝ INPUT AND CHECK THE G VECTOR
[13]   GGMIN:⎕←((PRINTQUAD 'PHASE ');(PRINTQUAD PHASE);PRINTQUAD
       ':')
[14]   PRINT ''
[15]   PRINT 'ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):'
[16]   STATESPERPHASE[PHASE]INPUT GGM
[17]   ⍝ CHECK FOR COMMAND
[18]   →(1=∧
       ∈!COMMAND')/GGMIN
[19]   ⍝ CHECK VALIDITY OF INPUT.
[20]   →(CHECKBIN IN)/GGMIN
[21]   ⍝ PLACE THE INPUT IN THE SET OF G VECTORS
[22]   Z[PHASE;⍳⍴IN]←IN
[23]   ⍝ INCREMENT THE PHASE COUNTER AND BRANCH IF APPROPRIATE
[24]   PHASE←PHASE+1
[25]   →(PHASE<NUMPHASES)/GGMIN
     ∇

     ∇ GGMINFO
[1]    ⍝
[2]    ⍝    ROUTINE GIVING HELP ON INPUTTING G MATRICES
[3]    ⍝
[4]    PRINT 'ENTER THE DIAGONAL OF THE CHARACTERISTIC (G) MATRIX FOR
       THE PHASE UNDER CONSIDERATION.'
[5]    PRINT 'EACH ENTRY SHOULD BE EITHER 0 OR 1 AND SHOULD BE
       SEPARATED FROM THE OTHER '
[6]    PRINT 'ENTRIES BY SPACES AND/OR COMMAS. THE NUMBER OF ENTRIES
       SHOULD BE THE SAME'
[7]    PRINT 'AS THE NUMBER OF STATES FOR THE PHASE. ALSO THE ORDER
       OF THE ENTRIES SHOULD'
[8]    PRINT 'CORRESPOND TO THE ORDER OF THE STATES AS CONSIDERED
       ELSEWHERE WITHIN THE'
[9]    PRINT 'METAPHOR PACKAGE FOR THE PHASE.'
[10]   PRINT 'EXAMPLE:'
[11]   PRINT '    0 0, 1,1  1'
```

```
[12]  PRINT 'THIS INFORMS METAPHOR THAT THE CHARACTERISTIC MATRIX
      FOR THIS PHASE IS'
[13]  PRINT '            0 0 0 0 0'
[14]  PRINT '            0 0 0 0 0'
[15]  PRINT '            0 0 1 0 0'
[16]  PRINT '            0 0 0 1 0'
[17]  PRINT '            0 0 0 0 1
[18]  PRINT ''
[19]  PRINT 'DO YOU WANT REFERENCES?'
[20]  ⍝ LEAVE IF NOT
[21]  →(~INYES)/0
[22]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[23]  PRINT ''
[24]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[25]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[26]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[27]  PRINT ''
[28]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[29]  PRINT ''
[30]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[31]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[32]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[33]  PRINT ''
[34]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[35]  PRINT ''
[36]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[37]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[38]  PRINT ''
      ∇
```

```
      ∇ Z←GETFVECTOR
[1]    ⍝
[2]    ⍝   ROUTINE FOR INPUTTING FINAL STATE VECTOR
[3]    ⍝
[4]    ⍝  INPUT AND CHECK THE F VECTOR
[5]   GFVIN:PRINT 'ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):'
[6]    STATESPERPHASE[NUMPHASES]INPUT GFV
[7]    ⍝ CHECK FOR COMMAND
[8]    →(1=∧ ∈'COMMAND')/GFVIN
[9]    ⍝ CHECK VALIDITY OF INPUT.
[10]   →(CHECKBIN IN)/GFVIN
[11]   ⍝  SET F VECTOR AND LEAVE
[12]   Z←(MAXNUMSTATES,1)⍴IN,(MAXNUMSTATES⍴0)
[13]  DEFF←1
      ∇
```

```
        ∇ GFVINFO
[1]     ⍝
[2]     ⍝   ROUTINE GIVING HELP ON INPUTTING THE F VECTORS
[3]     ⍝
[4]     PRINT 'ENTER THE CHARACTERISTIC (F) VECTOR FOR THE  TRAJECTORY
        SET UNDER CONSIDERATION.  '
[5]      PRINT  'EACH  ENTRY  SHOULD  BE  EITHER  0 OR 1 AND SHOULD BE
        SEPARATED FROM THE OTHER ENTRIES '
[6]     PRINT 'BY SPACES AND/OR COMMAS. THE NUMBER OF  ENTRIES  SHOULD
        BE THE SAME AS THE NUMBER'
[7]      PRINT 'OF STATES OF THE FINAL PHASE MODEL.  ALSO THE ORDER OF
        THE ENTRIES SHOULD'
[8]     PRINT 'CORRESPOND TO THE ORDER OF  THE  STATES  AS  CONSIDERED
        ELSEWHERE IN THE'
[9]     PRINT 'METAPHOR PACKAGE FOR THE FINAL PHASE.'
[10]    PRINT 'EXAMPLE:'
[11]    PRINT '    1 0, 1 ,0'
[12]    PRINT  'THIS  INFORMS  METAPHOR THAT THE CHARACTERISTIC VECTOR
        FOR THIS TRAJECTORY '
[13]    PRINT 'SET IS'
[14]    PRINT '  1'
[15]    PRINT '  0'
[16]    PRINT '  1'
[17]    PRINT '  0 .'
[18]    PRINT ''
[19]    PRINT 'DO YOU WANT REFERENCES?'
[20]    ⍝ LEAVE IF NOT
[21]    →(~INYES)/0
[22]    PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING  AND
        ANALYSIS, SEE'
[23]    PRINT ''
[24]    PRINT  '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
        THE '
[25]    PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,''  NASA
        GRANT'
[26]    PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[27]    PRINT ''
[28]    PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[29]    PRINT ''
[30]    PRINT  '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
        THE '
[31]    PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,''  NASA
        GRANT'
[32]    PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[33]    PRINT ''
[34]    PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[35]    PRINT ''
[36]    PRINT  '    S. PAKIN,  ''APL\360  REFERENCE  MANUAL,''  SCIENCE
        RESEARCH'
[37]    PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[38]    PRINT ''
    ∇
```

```
     ∇ Z←GETVVALUES;GVVTRUE;GVVFALSE
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING TIME-INVARIANT BASIC VARIABLE
      PROBABILITIES
[3]   ⍝
[4]   ⍝  INPUT AND CHECK THE V VECTOR
[5]   ⍝ IF NO TIME-INVARIANT BASIC VARIABLES, THEN SET V←0 , TURN ON
      THE DEFINITION FLAG, AND EXIT
[6]   Z←0
[7]   DEFV←1
[8]   →(NUMBASICVARIABLES=0)/0
[9]   GVVIN:⍞←((PRINTQUAD 'ENTER THE ');(PRINTQUAD
      NUMBASICVARIABLES);PRINTQUAD ' ELEMENT TIME-INVARIANT BASIC
      VARIABLE VECTOR (SPACE BETWEEN EACH ENTRY):')
[10]  PRINT ''
[11]  NUMBASICVARIABLES INPUT GVV
[12]  ⍝ CHECK FOR COMMAND
[13]  →(1=∧
      ∈'COMMAND')/GVVIN
[14]  ⍝ CHECK VALIDITY OF INPUT.
[15]  →(CHECKTRI IN)/GVVIN
[16]  ⍝ SET V VECTOR
[17]  ⍝ FIRST, SET THE TRUE VECTOR
[18]  GVVTRUE←IN∈ 0 2
[19]  ⍝ THEN SET THE FALSE VECTOR
[20]  GVVFALSE←IN∈ 1 2
[21]  ⍝ COMBINE FOR THE V VECTOR
[22]  Z←GVVTRUE,GVVFALSE
      ∇


      ∇ GVVINFO
[1]   ⍝
[2]   ⍝   ROUTINE GIVING HELP ON INPUTTING THE TIME-INVARIANT BASIC
      VARIABLE VECTOR
[3]   ⍝
[4]   PRINT 'ENTER A VECTOR OF 0''S, 1''S, AND 2''S TO INDICATE
      WHETHER THE '
[5]   PRINT 'TIME-INVARIANT BASIC VARIABLE''S OCCURRENCES OR NON-
      OCCURRENCES SHOULD BE'
[6]   PRINT 'CONSIDERED IN THE TRAJECTORY SET.  THE CODING IS AS
      FOLLOWS:'
[7]   PRINT ''
[8]   PRINT '       0       THE CORRESPONDING BASIC VARIABLE''S
      OCCURRENCE SHOULD'
[9]   PRINT '               BE CONSIDERED'
[10]  PRINT ''
[11]  PRINT '       1       THE CORRESPONDING BASIC VARIABLE''S NON-
      OCCURRENCE'
[12]  PRINT '               SHOULD BE CONSIDERED'
[13]  PRINT ''
[14]  PRINT '       2       EITHER THE OCCURRENCE OR  NON-OCCURRENCE
```

```
      OF THE '
[15]  PRINT '                CORRESPONDING BASIC VARIABLE SHOULD BE
      CONSIDERED'
[16]  PRINT '                (I.E., THE BASIC VARIABLE IS A ''DONT''T
      CARE'')'
[17]  PRINT ''
[18]  PRINT 'ENTER A ROW OF 0''S, 1''S, AND 2''S, SEPARATING EACH
      ENTRY BY SPACES'
[19]  PRINT 'AND/OR COMMAS.  THE NUMBER OF ENTRIES SHOULD BE THE
      SAME AS THE '
[20]  PRINT 'NUMBER OF TIME-INVARIANT BASIC VARIABLES DECLARED
      EARLIER.  ALSO THE ORDER'
[21]  PRINT 'OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER OF THE
      BASIC VARIABLES'
[22]  PRINT 'AS CONSIDERED ELSEWHERE IN THE METAPHOR PACKAGE.'
[23]  PRINT 'EXAMPLE:'
[24]  PRINT '          0,1  1 , 2'
[25]  PRINT 'THIS INDICATES TO METAPHOR THAT FOR THIS TRAJECTORY
      SET, THE OCCURRENCE'
[26]  PRINT 'OF THE FIRST TIME-INVARIANT BASIC VARIABLE IS IMPORTANT
      TO THE TRAJECTORY'
[27]  PRINT 'SET, THE NON-OCCURRENCE OF THE SECOND AND THIRD TIME-
      INVARIANT BASIC VARIABLES'
[28]  PRINT 'IS IMPORTANT, AND THAT THE FOURTH TIME-INVARIANT BASIC
      VARIABLE IS IRRELEVANT.'
[29]  PRINT ''
[30]  PRINT 'DO YOU WANT REFERENCES?'
[31]  ⍝ LEAVE IF NOT
[32]  →(~INYES)/0
[33]  PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
      ANALYSIS, SEE'
[34]  PRINT ''
[35]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE '
[36]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[37]  PRINT '   NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
[38]  PRINT ''
[39]  PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
[40]  PRINT ''
[41]  PRINT '   J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
      THE ' '
[42]  PRINT '   EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA
      GRANT'
[43]  PRINT '   NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
[44]  PRINT ''
[45]  PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
[46]  PRINT ''
[47]  PRINT '   S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE
      RESEARCH'
[48]  PRINT '   ASSOCIATES,INC., CHICAGO, 1972.'
[49]  PRINT ''
```

```
∇
```

```
     ∇ Z←CALCTRAJPROB;J;PHASE
[1]   ⍝
[2]   ⍝   CALCULATES THE PROBABILITY OF THE GIVEN TRAJECTORY
[3]   ⍝
[4]   ⍝ COMPUTE THE INITIAL PHASE PROBABILITY
[5]   Z←I+.×P[1;;]
[6]   ⍝
[7]   ⍝  LOOP THROUGH THE OTHER PHASES
[8]   ⍝ INITIALIZE PHASE COUNTER
[9]   PHASE←2
[10]  ⍝ FIRST CONSIDER THE SUCCESS STATES MATRIX
[11] CTPLOOP:Z←Z×G[PHASE-1;]
[12]  ⍝ THEN THE INTERPHASE MATRIX
[13]  Z←Z+.×H[PHASE-1;;]
[14]  ⍝ THEN THE TRANSITION MATRIX
[15]  Z←Z+.×P[PHASE;;]
[16]  ⍝
[17]  ⍝ INCREMENT PHASE COUNTER AND BRANCH IF APPROPRIATE
[18]  PHASE←PHASE+1
[19]  →(PHASE≤NUMPHASES)/CTPLOOP
[20]  ⍝
[21]  ⍝ POST-MULTIPLY BY FINAL STATE
[22]  Z←Z+.×F
[23]  ⍝ FINALLY, MULTIPLY BY THE TIME-INVARIANT BASIC VARIABLE
      PROBABILITIES
[24]  ⍝ IF NO TIME-INVARIANT BASIC VARIABLES, EXIT
[25]  →(NUMBASICVARIABLES=0)/0
[26]  Z←Z×+/V×BASICVARIABLES
     ∇
```

I/O AND CHECKING FUNCTIONS

```
     ∇ SHAPE INPUT ROUTINE;COMMANDVECTOR
[1]   ⍝
[2]   ⍝   ROUTINE FOR INPUTTING ANSWERS.  INPUT CAN BE DATA OR
      COMMANDS.
[3]   ⍝     IF NUMERIC, DATA RETURNED IN 'IN', ELSE IF COMMAND,
      'COMMAND' RETURNED
[4]   ⍝
[5]   ⍝  GET INPUT
[6]  ININ:IN←⍞
[7]   ⍝ ECHO INPUT IF DESIRED
[8]   →(~ECHOINPUT)/INVECTORIZE
[9]   ⍞←((PRINTQUAD '⍞:  ');PRINTQUAD IN)
[10]  PRINT ''
[11]  ⍝
[12]  ⍝ CHANGE ALL SCALAR INPUTS TO VECTORS
[13] INVECTORIZE:IN←,IN
```

```
[14]  ⍝
[15]  ⍝  CHECK FOR COMMAND AND SET COMMAND VECTOR
[16]  COMMANDVECTOR←∧/COMMANDLIST∈IN
[17]  ⍝
[18]  ⍝
[19]  ⍝ EXECUTE COMMAND IF PRESENT
[20] INCOMMAND:→(COMMANDVECTOR)/INHELP,INEXIT,INDATA,
      INALTER,INCALC,INECHO,INBRIEF,INCOM,INEVAL
[21]  ⍝ ELSE CHECK THE SHAPE OF THE INPUT.  'SHAPE' IS ALWAYS AN
      INTEGER SCALAR.  IF GOOD, LEAVE.
[22]  →(SHAPE=⍴IN)/0
[23]  ⍝ OTHERWISE THE INPUT WAS OF THE WRONG DIMENSION.  PRINT ERROR
      MESSAGE AND TRY AGAIN.
[24]  ⍞←((PRINTQUAD 'ENTER ');(PRINTQUAD SHAPE);PRINTQUAD '
      ITEMS')
[25]  →ININ
[26]  →0
[27]  ⍝
[28]  ⍝
[29]  ⍝ HELP REQUESTED
[30] INHELP:COMMANDHELP ROUTINE
[31]  IN←'COMMAND'
[32]  →0
[33]  ⍝
[34]  ⍝
[35]  ⍝ END THE PROGRAM
[36] INEXIT:→
[37]  ⍝
[38]  ⍝
[39]  ⍝ DISPLAY OF DATA REQUESTED
[40] INDATA:COMMANDDATA
[41]  IN←'COMMAND'
[42]  →0
[43]  ⍝
[44]  ⍝
[45]  ⍝ CHANGE OF DATA REQUESTED
[46] INALTER:COMMANDALTER
[47]  IN←'COMMAND'
[48]  →0
[49]  ⍝
[50]  ⍝
[51]  ⍝  CALCULATION OF PERFORMABILITY REQUESTED
[52] INCALC:COMMANDCALC
[53]  IN←'COMMAND'
[54]  →0
[55]  ⍝
[56]  ⍝ SETTING OF THE ECHO SWITCH DESIRED
[57] INECHO:COMMANDECHO IN
[58]  IN←'COMMAND'
[59]  →0
[60]  ⍝
[61]  ⍝ COMMENT DESIRED
```

```
[62] INCOM:COMMANDCOM
[63]  IN←'COMMAND'
[64]  →0
[65]  ⍝
[66]  ⍝ SETTING OF BRIEF SWITCH DESIRED
[67] INBRIEF:COMMANDBRIEF IN
[68]  IN←'COMMAND'
[69]  →0
[70]  ⍝
[71]  ⍝ PERFORMABILITY COMPUTATION DESIRED
[72] INEVAL:COMMANDEVAL
[73]  →0
      ∇


      ∇ Z←INYES;IN
[1]   ⍝
[2]   ⍝  ROUTINE ASKING YES AND NO ANSWERS.  1 RETURNED IF YES, 0 IF
      NO
[3]   ⍝
[4]   ⍝ PRINT PROMPT AND GET ANSWER
[5]  IYIN:PRINT '□:'
[6]   IN←⍞
[7]   ⍝ ECHO THE INPUT IF DESIRED
[8]   →(~ECHOINPUT)/IYSCAN
[9]   PRINT(IN)
[10]  ⍝ ASSUME YES HAS PRIORITY.  LOOK FOR Y OR 1
[11] IYSCAN:Z←∨/'Y1'∈IN
[12]  ⍝ IF N OR 0 INPUT OR IF YES INPUT, EXIT
[13]  →(Z∨∨/'N0'∈IN)/0
[14]  ⍝ ELSE TRY AGAIN
[15]  PRINT 'ENTER YES OR NO'
[16]  →IYIN
      ∇


      ∇ Z←CHECKBIN CHECKNO
[1]   ⍝
[2]   ⍝  RETURN 0 IF CHECKNO CONTAINS ONLY BINARY ZEROS AND ONES
[3]   ⍝   ELSE PRINT MESSAGE AND RETURN 1
[4]   ⍝
[5]   ⍝  CHECK FOR PROPER BINARY ELEMENTS
[6]   Z←~(∧/CHECKNO∈ 0 1)
[7]   →(~Z)/0
[8]   PRINT 'EACH ENTRY MUST BE EITHER 0 OR 1'
[9]   →0
      ∇
```

```
          ∇ Z←CHECKPOSI CHECKNO
[1]    ⍝
[2]    ⍝   RETURN 0 IF CHECKNO IS A POSITIVE INTEGER, ELSE PRINT
       MESSAGE AND RETURN 1
[3]    ⍝
[4]    ⍝  CHECK FOR POSITIVENESS
[5]    Z←~∧/(CHECKNO>0)
[6]    →(~Z)/CHECKINT
[7]    PRINT 'INPUT NOT POSITIVE'
[8]    →0
[9]    ⍝
[10]   ⍝ CHECK FOR INTEGER
[11]  CHECKINT:Z←~∧/CHECKNO=⌈CHECKNO
[12]   →(~Z)/0
[13]   PRINT 'INPUT NOT AN INTEGER'
      ∇


      ∇ Z←CHECKPROB CHECKNO
[1]    ⍝
[2]    ⍝   RETURN 0 IF CHECKNO IS BETWEEN 0 AND 1 INCLUSIVE, ELSE
       PRINT MESSAGE AND RETURN 1
[3]    ⍝
[4]    ⍝  CHECK FOR PROPER RANGE
[5]    Z←~(∧/CHECKNO≥0)∧(∧/CHECKNO≤1)
[6]    →(~Z)/0
[7]    PRINT 'INPUT NOT BETWEEN 0 AND 1'
      ∇


      ∇ Z←CHECKTRI CHECKNO
[1]    ⍝
[2]    ⍝  RETURN 0 IF CHECKNO CONTAINS ONLY TRINARY ZEROS, ONES, AND
       TWOS
[3]    ⍝   ELSE PRINT MESSAGE AND RETURN 1
[4]    ⍝
[5]    ⍝  CHECK FOR PROPER TRINARY ELEMENTS
[6]    Z←~(∧/CHECKNO∈ 0 1 2)
[7]    →(~Z)/0
[8]    PRINT 'EACH ENTRY MUST BE EITHER 0,1, OR 2'
[9]    →0
      ∇
```

```
           ∇ Z←PRINT Q
[1]    ⍝
[2]    ⍝
[3]    ⍝    PRINTING ROUTINE FOR METAPHOR
[4]    ⍝  IF BRIEF←1 NO OUTPUT IS GIVEN
[5]    ⍝
[6]    ⍝ CHECK FOR TERSE INPUT FLAG
[7]    →(BRIEFOUTPUT=1)/0
[8]    ⍝ FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[9]    Z←Q
       ∇


       ∇ Z←PRINTQUAD Q
[1]    ⍝
[2]    ⍝
[3]    ⍝    PRINTING ROUTINE (WITH QUOTE QUAD) FOR METAPHOR
[4]    ⍝  IF BRIEF←1 NO OUTPUT IS GIVEN
[5]    ⍝
[6]    ⍝ CHECK FOR TERSE INPUT FLAG
[7]    →(BRIEFOUTPUT=1)/PRINTQUADBRIEF
[8]    ⍝ FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[9]    Z←Q
[10]   →0
[11]   ⍝ ELSE RETURN THE EMPTY STRING
[12]  PQBRIEF:Z←''
       ∇
```

APL SUPPORT FUNCTION

```
       ∇ Z←M ENCODE N;ROW
[1]    ⍝
[2]    ⍝
[3]    ⍝   THE ENCODE FUNCTION EMPLOYED ON MTS WILL NOT ACCEPT
       VECTORS AS
[4]    ⍝   ARGUMENTS ON THE RIGHT HAND SIDE.  THIS FUNCTION SIMULATES
       THAT CAPABILITY
[5]    ⍝   THE COMPARABLE APL NOTATION WOULD BE:     M⊤N
[6]    ⍝
[7]    ⍝  TEST FOR A SINGLE INPUT
[8]    →(0=ρρN)/ESINGLE
[9]    ⍝   INITIALIZE THE ARRAY AND LOOP COUNTER
[10]   Z←((⌈/1,ρM),(ρN))ρ0
[11]   COL←1
[12]  ELOOP:Z[;COL]←M⊤N[COL]
[13]   COL←COL+1
[14]   →(COL≤ρN)/ELOOP
[15]   ⍝ EXIT
[16]   →0
[17]   ⍝
[18]   ⍝  IF ONLY ONE ARGUMENT TO BE DECODED
```

```
[19] ESINGLE:Z←MTN
     ∇
```

INDEX