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

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

1. Introduction ............................................ 1

2. Conventions ............................................. 4
   2.1 Function Names ...................................... 4
   2.2 Variable Names ..................................... 5
   2.3 Label Names ....................................... 6
   2.4 Other Conventions ................................. 7

3. Control Techniques ...................................... 8
   3.1 INPUT ........................................... 8
   3.2 User Assistance Commands .......................... 14
      3.2.1 HELP ..................................... 14
      3.2.2 ALTER and DATA .............................. 17

4. Algorithms Employed in EVAL ............................ 25
   4.1 Matrix Generation ................................ 25
      4.1.1 DEDFAIL .................................. 26
      4.1.2 NFAIL ..................................... 33
   4.2 EVALUATION ....................................... 40

5. Description of METAPHOR Variables ...................... 47

6. Interdependence of METAPHOR Functions .................. 54

7. METAPHOR Function Descriptions .......................... 62
   BRIEF ........................................... 63
   CALCTRAJPROB ...................................... 64
   CHECKBIN ........................................ 66
   CHECKPOSI ...................................... 67
   CHECKPROB ..................................... 68
   CHECKTRI ...................................... 69
   DECLAREMETAPHOR .................................. 70
   ECHO ........................................... 74
LIST OF FIGURES

1. Flowchart for the function INPUT ......................... 9
2. Typical flowchart section involving a call to the function INPUT ........................................... 16
3. Flowchart for the function COMMANDALTER ............. 20
4. Flowchart for the function COMMANDDATA ............... 21
5. Flowchart for the function GETALTERVECTOR .......... 23
6. Flowchart for the function GETDATAVECTOR ............. 24
7. Transition diagram for the example of Section 4.1.1 .. 28
8. Flowchart for the algorithm DEDFAIL ...................... 30
9. Transition diagram for the example of Section 4.1.2 .. 35
10. Flowchart for the algorithm NFAIL ....................... 37
11. Flowchart for the algorithm EVALUATION ............... 41
12. Interdependence of METAPHOR functions ................. 56
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 \( U_a \) that corresponds to \( a \).

5) For each trajectory set \( U_a \), 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.

METAPHOR (Version 1) Programmer's Guide
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.

1. Introduction
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 GSV, are the same abbreviations used to prefix the name of the information routines associated with the variables, i.e., GAVINFO and GSVINFO 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:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM</td>
<td>NUMBER</td>
</tr>
<tr>
<td>ACCLEV</td>
<td>ACCOMPLISHMENT LEVEL</td>
</tr>
<tr>
<td>PROB</td>
<td>PROBABILITY</td>
</tr>
<tr>
<td>TRI</td>
<td>TRINARY</td>
</tr>
<tr>
<td>TRAJ</td>
<td>TRAJECTORY</td>
</tr>
<tr>
<td>BIN</td>
<td>BINARY</td>
</tr>
<tr>
<td>INFO</td>
<td>INFORMATION</td>
</tr>
<tr>
<td>EVAL</td>
<td>EVALUATE</td>
</tr>
<tr>
<td>COM</td>
<td>COMMENT</td>
</tr>
<tr>
<td>CALC</td>
<td>CALCULATE</td>
</tr>
</tbody>
</table>

2.2 Variable Names
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.

```plaintext
A  COMMANDALTER
CC  COMMANDCOM
CCALC  COMMANDCALC
CHECK  CHECKPOSI
CTP  CALCTRAJPORB
D  COMMANDDATA
E  ENCODE
GAP  GETACCLEVPORB
GAV  GETALTERVECTOR
GBV  GETBASICVARIABLES
GDV  GETDATAVECTOR
GENHM  GENERATEHMATRIX
GENPM  GENERATEPMATRIX
GFV  GETFVECTOR
GG  GGIVEN
CCM  GETGMATRICES
GIV  GETIVECTOR
GN  GNFAIL
GNA  GETNUMACCLEV
GNBV  GETNUMBASICVARIABLES
GNP  GETNUMPHASES
GNTS  GETNUMTRAJSETS
GS  GETSTATES
GVV  GETVVALUES
H  COMMANDHELP
```
IN INPUT
IV INYES
MET METAPHOR
PO 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 cannot 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.

3.1 INPUT
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:
<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>CODE NAME</th>
<th>CODE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETNUMPHASES</td>
<td>GNP</td>
<td>1</td>
</tr>
<tr>
<td>GETSTATES</td>
<td>GS</td>
<td>2</td>
</tr>
<tr>
<td>GENERATEPMATRIX</td>
<td>GPM</td>
<td>3</td>
</tr>
<tr>
<td>GGIN</td>
<td>GG</td>
<td>4</td>
</tr>
<tr>
<td>GFAIL</td>
<td>GN</td>
<td>5</td>
</tr>
<tr>
<td>GDEDFAIL</td>
<td>GD</td>
<td>6</td>
</tr>
<tr>
<td>GENERATEHMAtrIX</td>
<td>GHM</td>
<td>7</td>
</tr>
<tr>
<td>GETNUMBASICVARIABLES</td>
<td>GNBV</td>
<td>8</td>
</tr>
<tr>
<td>GETBASICVARIABLES</td>
<td>GBV</td>
<td>9</td>
</tr>
<tr>
<td>GETNUMACCELVE</td>
<td>GNA</td>
<td>10</td>
</tr>
<tr>
<td>GETVECTOR</td>
<td>GIV</td>
<td>11</td>
</tr>
<tr>
<td>GETGMATRICES</td>
<td>GGM</td>
<td>12</td>
</tr>
<tr>
<td>GETFVECTOR</td>
<td>GFV</td>
<td>13</td>
</tr>
<tr>
<td>GETVECTOR</td>
<td>GVV</td>
<td>14</td>
</tr>
<tr>
<td>GETNUMTRAJSETS</td>
<td>GNTS</td>
<td>15</td>
</tr>
<tr>
<td>GETALTERVECTOR</td>
<td>GAV</td>
<td>16</td>
</tr>
<tr>
<td>GETDATAVECTOR</td>
<td>GDV</td>
<td>17</td>
</tr>
<tr>
<td>METAPHOR</td>
<td>MET</td>
<td>18</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>FLAG</th>
<th>VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFACCLEVEL</td>
<td>[Variables associated with accomplishment level definition]</td>
</tr>
<tr>
<td>DEFBASICVARIABLES</td>
<td>BASICVARIABLES</td>
</tr>
<tr>
<td>DEFF</td>
<td>F</td>
</tr>
<tr>
<td>DEFG</td>
<td>G</td>
</tr>
<tr>
<td>DEFI</td>
<td>I</td>
</tr>
<tr>
<td>DEFPNUMACCLEV</td>
<td>NUMACCELV</td>
</tr>
<tr>
<td>DEFPNUMBASICVARIABLES</td>
<td>NUMBASICVARIABLES</td>
</tr>
<tr>
<td>DEFPNUMPHASES</td>
<td>NUMPHASES</td>
</tr>
<tr>
<td>DEFPNUMSTATES</td>
<td>NUMSTATES</td>
</tr>
<tr>
<td>DEPTRAJSETS</td>
<td>NUMTRAJSETS</td>
</tr>
<tr>
<td>DEFP</td>
<td>P</td>
</tr>
<tr>
<td>DEPPERFORMABILITY</td>
<td>PERFORMABILITY</td>
</tr>
<tr>
<td>DEFV</td>
<td>V</td>
</tr>
</tbody>
</table>

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
Determine first index of ALTERVECTOR with value 1

Set that index of ALTERVECTOR to 0

Print warning

Is that variable defined?

yes

Print that the variable associated with the index is to be altered

no

Print warning

Call the proper functions to obtain the new value for that variable
Determine first index of DATAVECTOR with value 1

Set that index of DATAVECTOR to 0

Is that variable defined?

Print the variable

Print that the variable associated with the index is to be displayed

Figure 4. Flowchart for the function COMMANDDATA
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.
GETALTERVECTOR

Print heading

Print first list of variables which can be altered

INPUT = user's response

COMMAND HELP GDV

yes

INPUT = 'HELP'?

no

Print second list of variables which can be altered

INPUT = INPUT, user response

COMMAND HELP GDV

yes

INPUT = 'HELP'?

no

Set ALTERVECTOR to correspond to X's in INPUT

RETURN

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 \( N \times N \) 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:

\[
\begin{array}{cccc}
(1,1) & (1,0) & (0,1) & (0,0) \\
(1,0) & 0.00E0 & 9.99E-1 & 0.00E0 & 1.00E-3 \\
(1,0) & 0.00E0 & 0.00E0 & 9.99E-1 & 1.00E-3 \\
(0,0) & 0.00E0 & 0.00E0 & 0.00E0 & 1.00E0 \\
\end{array}
\]

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,

\[4.1.1\] DEDFAIL
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.
DEDFAIL

N = number of states
T = length of phase
LAMBDA = component failure rate

Z = NxN matrix of zeros

INDEX =

\[
\begin{bmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & 1 & 1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 1 \\
0 & 0 & 0 & \ldots & 0
\end{bmatrix}
\]

= NxN matrix such that each row is a state

Figure 8. Flowchart for the algorithm DEDFAIL
SIZE = number of 1's in row I that have corresponding 1's in row J

= number of success transitions

FAIL = \( (1 - e^{-\lambda T})^{\text{SIZE}} \)

\( ((\text{Number of 1's in row I}) - \text{SIZE}) \)

= probability that the required number of failures for transferring from state I to state J occur

SUCCESS = \( (e^{-\lambda T})^{\text{SIZE}} \)

= probability that the required number of successes for transferring from state I to state J occur

Figure 8. Flowchart for the algorithm DEDFAIL, continued
\[ Z(I,J) = \begin{cases} 
\text{SUCCESS} \times \text{FAIL}, \text{ if } \text{SIZE} > \text{the number of 1's in row } J, \text{ and if row } I \text{ is not all zeros} \\
\text{when row } J \text{ is not all zeros} \\
0, \text{ otherwise} 
\end{cases} \]

= \begin{cases} 
\text{SUCCESS} \times \text{FAIL}, \text{ if all transitions are possible} \\
0, \text{ otherwise} 
\end{cases}

= \text{probability of transferring to state } J \text{ given that the system was initially in state } I

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 \(N\times N\) 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):

\[
1, \text{ if } j=1 \\
(K(1)+1)(K(2)+1) \ldots (K(j-1)+1), \text{ if } j>1 .
\]

The values ("digits") that the \(j\)-th place can take on are \(0, 1, \ldots, K(j)\). (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 not limited to \(0, 1, \ldots, 9\).) 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 from 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:

\[
\begin{array}{cccccc}
(2,1) & (2,0) & (1,1) & (1,0) & (0,1) & (0,0) \\
(2,0) & 0.000E0 & 9.802E-1 & 0.000E0 & 1.970E-2 & 0.000E0 & 9.901E-5 \\
(1,1) & 0.000E0 & 0.000E0 & 9.802E-1 & 9.851E-3 & 9.851E-3 & 9.901E-5 \\
(1,0) & 0.000E0 & 0.000E0 & 0.000E0 & 9.900E-1 & 0.000E0 & 9.950E-3 \\
(0,1) & 0.000E0 & 0.000E0 & 0.000E0 & 0.000E0 & 9.900E-1 & 9.950E-3 \\
(0,0) & 0.000E0 & 0.000E0 & 0.000E0 & 0.000E0 & 0.000E0 & 1.000E0 \\
\end{array}
\]

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.
\[ N = \text{number of states} \]
\[ T = \text{length of phase} \]
\[ \text{LAMBDA} = \text{component failure rate} \]
\[ \text{GROUPS} = \text{number of groups} \]
\[ \text{NUM} = \text{vector containing the number of components in each group} \]

\[ \text{NUM} = \text{NUM} + 1 \]

\[ Z = N \times N \text{ matrix of zeros} \]

\[
\begin{bmatrix}
K(N) & K(N-1) & \cdots & K(1) \\
K(N) & K(N-1) & \cdots & K(1)-1 \\
\vdots & \vdots & \ddots & \vdots \\
K(N)-1 & K(N-1) & \cdots & K(1) \\
K(N)-1 & K(N-1) & \cdots & K(1)-1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 \\
0 & 0 & \cdots & 0
\end{bmatrix}
\]

= \text{N} \times \text{N} \text{ matrix such that each row is a state, where } K(m) \text{ is the maximum number of components in group } m \]

Figure 10. Flowchart for the algorithm NFAIL
I will denote the row of INDEX containing the initial state under consideration.

J will denote the row of INDEX containing the final state under consideration.

COEF = \( \frac{\text{INDEX}(I,1)}{\text{INDEX}(J,1)} \cdot \ldots \cdot \frac{\text{INDEX}(I,N)}{\text{INDEX}(J,N)} \)

= number of combinations by which state I can fail to state J

FAIL = \( (1 - e^{-\lambda T}) \) ** (Number of failed components in state I) - (Number of nonfailed components in state J))

= probability that the required number of failures for transferring from state I to state J occur

SUCCESS = \( e^{-\lambda T} \) ** (Number of nonfailed components in state J)

= probability that the required number of successes for transferring from state I to state J occur.

Figure 10. Flowchart for the algorithm NFAIL, continued.
$Z(I, J) = \text{COEFF} \times \text{FAIL} \times \text{SUCCESS}$

$= \text{probability of transferring to state } J \text{ given that the system was initially in state } I$
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...
4.2 EVALUATION

Figure 11. Flowchart for the algorithm EVALUATION
42

BASICVARIABLES = the probabilities of each of the constant (i.e., non-time varying) basic variables

NUMACCLEV = the number of accomplishment levels.

PERFORMABILITY = vector of NUMACCLEV zeros

L = 1

L will denote the accomplishment level under consideration

Figure 11. Flowchart for the algorithm EVALUATION, continued

METAPHOR (Version 1) Programmer's Guide
Figure 11. Flowchart for the algorithm EVALUATION, continued
V = characterization of the constant basic variables

\[ \text{TRAJPROB} = I \cdot P(1) \]

Initialze the variable containing the probability of trajectory set T. Performed in CALCTRJPROB.

\[ \text{PHASE} = 2 \]

PHASE will denote the phase under consideration

\[ \text{TRAJPROB} = \text{TRAJPROB} \cdot G(\text{PHASE}-1) \cdot H(\text{PHASE}-1) \cdot P(\text{PHASE}) \]

yes

PHASE < NUMPHASES?

no

\[ \text{TRAJPROB} = \text{TRAJPROB} \cdot F \cdot (V \cdot \text{BASICVARIABLES}) \]

Figure 11. Flowchart for the algorithm EVALUATION, continued
4.2 EVALUATION

Figure 11. Flowchart for the algorithm EVALUATION, continued
Probability = I * P(1) * G(1) * H(1) * P(2) * G(2) * H(2) *
... * H(n) * F * V * 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`.

**METAPHOR** (Version 1) Programmer's Guide
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 in 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 characteristic 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.

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

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

DEFG (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, GETMATRICES, 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.

DEFPV (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, GETVECTOR,
GETGMATRICES, GETHGMATRICES, GETVECTOR, GETPMATRICES,
GETVVALUES, INPUT, GDEDFAIL, GGEVEN, GNFAIL) vector containing the latest data input to METAPHOR.

MAXNUMSTATES (GETSTATES, GETGMATRICES, GETHGMATRICES,
GETVECTOR, 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, GETGMATRICES, 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.

METAPHOR (Version 1) Programmer's Guide
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, GETHMATRICES, GETIVECTOR, GETVALUES, 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.

5. Description of METAPhOR Variables
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.

5. Description of METAPHOR Variables
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:

6. Interdependence of METAPHOR Functions
6. Interdependence of METAPHOR Functions, continued

2 = COMMANDHELP

GNPINFO  GSINFO  GPNINFO  GGINFO  GNINFO  GDINFO  GHMINFO

GNBVINFO  GBVINFO  SNINFO  GIVINFO  GGMINFO  GFVINFO

GVINFO  GNTSINFO  GAVINFO  GDVINFO  METINFO

PRINT  INYES  PRINT
Figure 12. Interdependence of METAPHOR functions, continued
Figure 12. Independence of Metaphor Functions, continued
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: \( \text{result} \leftarrow \text{BRIEF arg} \)

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:
\[
\begin{array}{l}
\text{V Z+\text{BRIEF SWITCH}} \\
[1] \text{a} \\
[2] \text{a ROUTINE TO INPUT THE BRIEF COMMAND} \\
[3] \text{a} \\
[4] \text{a RETURN THE VALUE} \\
[5] \text{Z+\text{BRIEF },SWITCH} \\
\end{array}
\]
CALCTRAJPROB

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{CALCTRAJPROB} \)

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 GETVECTOR, GETGMATRICES, GETIVECTOR, and GETVALUES as well as the global variables to calculate the probability of a trajectory set. The formula used to perform the calculations is:

\[
\text{Probability} = I * P(1) * G(1) * H(1) * P(2) * G(2) * H(2) * \\
\cdots * H(n) * F * V * \text{BASICVARIABLES}
\]

where there are \( n \) phases, and \( * \) denotes matrix multiplication. The probability is returned in \( \text{result} \).

LISTING:

\[
\begin{align*}
V & \leftarrow \text{CALCTRAJPROB} ; J ; \text{PHASE} \\
& \text{CALCULATES THE PROBABILITY OF THE GIVEN TRAJECTORY} \\
& \text{COMPUTE THE INITIAL PHASE PROBABILITY} \\
& \text{INITIALIZE PHASE COUNTER} \\
& \text{FIRST CONSIDER THE SUCCESS STATES MATRIX} \\
& \text{THEN THE INTERPHASE MATRIX} \\
& \text{THEN THE TRANSITION MATRIX}
\end{align*}
\]
INCREMENT PHASE COUNTER AND BRANCH IF APPROPRIATE

PHASE <- PHASE + 1

-> (PHASE = NUMPHASES) / CTPLoop

POST-MULTIPLY BY FINAL STATE

Z+ . x F

FINALLY, MULTIPLY BY THE TIME-INVARIANT BASIC VARIABLE PROBABILITIES

IF NO TIME-INVARIANT BASIC VARIABLES, EXIT

-> (NUMBASICVARIABLES = 0) / 0

Z+ Z+ / V x BASICVARIABLES

7. CALCTRAJPROB METAPHOR FUNCTION DESCRIPTION
CHECKBIN

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{CHECKBIN arg} \)

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

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: \text{GETVECTOR, GETGMATRICES, GETIVECTOR}.

FUNCTIONS CALLED: \text{PRINT}.

COMMENTS: Checks the input argument \text{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 \text{result}.

LISTING:
\begin{verbatim}
V Z<CHECKBIN CHECKNO
[1] a RETURN 0 IF CHECKNO CONTAINS ONLY BINARY ZEROS AND ONES
[2] a ELSE PRINT MESSAGE AND RETURN 1
[3] a CHECK FOR PROPER BINARY ELEMENTS
[4] a V<(~Z)/0
[5] a PRINT 'EACH ENTRY MUST BE EITHER 0 OR 1'
[6] a -0
\end{verbatim}
CHECKPOSI

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE:  \texttt{result} \leftarrow \texttt{CHECKPOSI arg}

PURPOSE:  To check that \texttt{arg} has only positive integer values.

GLOBAL VARIABLES:  None.

CALLING FUNCTIONS:  \texttt{GETNUMACCLEV}, \texttt{GETNUMBASICVARIABLES}, \texttt{GETNUMPHASES}, \texttt{GETNUMTRAJSETS}, \texttt{GETSTATES}.

FUNCTIONS CALLED:  \texttt{PRINT}.

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

LISTING:
\begin{verbatim}
\% Z=CHECKPOSI CHECKNO
[1]  a
[2]  a  RETURN 0 IF CHECKNO IS A POSITIVE INTEGER, ELSE PRINT
MESSAGE AND RETURN 1
[3]  a
[4]  a  CHECK FOR POSITIVENESS
[5]  Z=\sim A/(CHECKNO>0)
[6]  \rightarrow(\sim Z)/CHECKINT
[7]  PRINT 'INPUT NOT POSITIVE'
[8]  \rightarrow 0
[9]  a
[10]  a  CHECK FOR INTEGER
[11]  CHECKINT:Z=\sim A/CHECKNO=\lceil CHECKNO
[12]  \rightarrow(\sim Z)/0
[13]  PRINT 'INPUT NOT AN INTEGER'
\end{verbatim}
CHECKPROB

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE:  \( \text{result} \leftarrow \text{CHECKPROB} \ \text{arg} \)

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

GLOBAL VARIABLES:  None.

CALLING FUNCTIONS:  GETBASICVARIABLES, GGIVEN.

FUNCTIONS CALLED:  PRINT.

COMMENTS:  Checks the input argument \( \text{arg} \) 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 \( \text{result} \).

LISTING:
\[
\begin{align*}
\text{\#1} & \; \text{RETURN 0 IF CHECKNO IS BETWEEN 0 AND 1 INCLUSIVE, ELSE PRINT MESSAGE AND RETURN 1} \\
\text{\#2} & \; \text{CHECK FOR PROPER RANGE} \\
\text{\#3} & \; \text{PRINT 'INPUT NOT BETWEEN 0 AND 1'}
\end{align*}
\]
CHECKTRI

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \texttt{result} \leftarrow \texttt{CHECKTRI \ arg}

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

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GETVVALUES.

FUNCTIONS CALLED: PRINT.

COMMENTS: Checks the input argument \texttt{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 \texttt{result}.

LISTING:
\begin{verbatim}
V Z=CHECKTRI CHECKNO
[1] A RETURN 0 IF CHECKNO CONTAINS ONLY TRINARY ZEROS, ONES, AND TWOS
[2] A ELSE PRINT MESSAGE AND RETURN 1
[3] A CHECK FOR PROPER TRINARY ELEMENTS
[4] A Z=-(^/CHECKNOe 0 1 2)
[5] +(~Z)/0
[6] PRINT 'EACH ENTRY MUST BE EITHER 0,1, OR 2'
[7] \rightarrow 0
\end{verbatim}
DECLAIMEMETAPHOR

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, DEFI, DEFN, DEFNACCEL, DEFNUMBASIC, DEFNUMPHER, DEFNUMSTATES, DEFNUMTRAJSETS, DEFP, DEFFERPERFORMABILITY, DEFX, ECHOINPUT, EVAL, EXIT, GIVEN, HELP, IDENTITY, NFAIL, OFF, ON, GAY, GBV, GD, GDV, GEX, GGM, GG, GHM, GIV, GN, GNA, GNBV, GNP, GNTS, GPM, GEX, 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, DEFACCL and DEFBASE.

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:

V DECLAREMETAPHOR
[1]  a
[2]  a DECLARES AND INITIALIZES THE CONSTANTS USED IN THE 'PERF' FUNCTION SERIES
[3]  a
[4]  a SET THE LIST OF COMMANDS
[5]  HELP='HELP '
[6]  EXIT='EXIT '
[7]  DATA='DATA '
[8]  ALTER='ALTER '
[9]  CALC='CALC '
[10]  COM='COM '
[12]  ON='ON '
[13]  OFF='OFF '
[14]  COMMANDSIZE=6
[15]  COMMANDLIST=(9,COMMANDSIZE)p'HELP ','EXIT ','DATA ','ALTER '

7. DECLAREMETAPHOR METAPHOR FUNCTION DESCRIPTION
SET LIST OF MATRIX GENERATOR TYPES

GIVEN='GIVEN'
DEDFAIL='DEDFAIL'
NFAIL='NFAIL'
IDENTITY='IDENTITY'

PMATRIXLIST=('4 8')p'GIVEN ','DEDFAIL ','NFAIL '
HMATRIXLIST=('2 8')p'GIVEN ','IDENTITY'

SET LIST OF ROUTINES FOR HELP CALLS

GCF=1
GS=2
GPM=3
GC=4
GN=5
GD=6
GHM=7
CNB=8
GBV=9
GNA=10
GIV=11
GGM=12
GFV=13
GVV=14
GNT=15
GAV=16
GDV=17
MET=18

SET NUMBER OF HELP ROUTINES
NUMHELPROUTINES=18

SET VARIABLE DEFINITION SWITCHES. 1 IF VARIABLE DEFINED, 0 IF NOT.
DEFNUMPHASES=0
DEFNUMSTATES=0
DEFF=0
DEFH=0
DEFUMBASICVARIABLES=0
DEFBASICVARIABLES=0
DEFUMACCLEV=0
DEFUMTRAJSETS=0
DEFI=0
DEFG=0
DEFF=0
DEFV=0
DEPERFORMABILITY=0
DEFCACLEVEL=0

MAKE DEFAULT SWITCH SETTINGS
BRIEFOUTPUT=0

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
7. DECLAREMETAPHOR METAPHOR FUNCTION DESCRIPTION
ECHO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{ECHO arg} \)

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 \( \text{arg} \) in \( \text{result} \). For example, after the command ECHO OFF, the vector 'ECHO OFF' is returned in \( \text{result} \). The default is ECHO OFF.

LISTING:

\[ V \quad Z+\text{ECHO SWITCH} \]
\[ \[1\] \ a \ \text{ROUTINE TO INPUT THE ECHO COMMAND} \]
\[ \[2\] \ a \ \text{RETURN THE VALUE} \]
\[ \[5\] \ Z+\text{'ECHO ',SWITCH} \]

\[ V \]
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.

jlocal 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:

```
V Z=M ENCODE N;ROW
[1] a
[2] a
[3] a THE ENCODE FUNCTION EMPLOYED ON MTS WILL NOT ACCEPT VECTORS AS
[4] a ARGUMENTS ON THE RIGHT HAND SIDE. THIS FUNCTION SIMULATES THAT CAPABILITY
[6] a
[7] a TEST FOR A SINGLE INPUT
[8] →(0=pN)/ESINGLE
[9] a INITIALIZE THE ARRAY AND LOOP COUNTER
[10] Z+=(1/1,pM),(pN))p0
[11] COL=1
[13] COL+COL=1
[14] →(COL≤pN)/ELOOP
[15] a EXIT
[16] →0
[17] a
[18] a IF ONLY ONE ARGUMENT TO BE DECODED
```
[19] $ESINGLE: z = M + N$
GENERATEHMATRIX

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{arg1 GENERATEHMATRIX arg2} \)

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

GLOBAL VARIABLES: \( \text{IN, GHM, HMATRIXLIST} \).

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 \( \text{arg1xarg2} \). \text{arg1} and \text{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 \rightarrow M \text{ GENERATEH\text{MATRIX \(N; \text{TYPE;}IN}} \]

[1] 

\[ \text{ROUTINE FOR SUPERVISING THE CONSTRUCTION OF THE \(H\) MATRIX FOR THE GIVEN PHASE} \]

[2] 

\[ \text{THE MATRIX IS TO HAVE SHAPE \(M \times N\)} \]

[3] 

\[ \text{GET TYPE OF MATRIX \(A\)} \]

[4] 

\[ \text{PRINT 'A'} \]

[5] 

\[ \text{GENH\text{MIN}; PRINT 'WHAT TYPE OF \(H\) MATRIX?' \(A\)} \]

[6] 

\[ \text{INPUT GHM} \]

[7] 

\[ \text{CHECK FOR COMMAND} \]

[8] 

\[ \rightarrow (1 = ^{\wedge} \text{ 'COMMAND'}) / \text{GENHMIN} \]

[9] 

\[ \text{CHECK FOR TYPE} \]

[10] 

\[ \rightarrow (^{\wedge} / \text{HMATRIXLIST<IN>/GENHM\text{GIVEN,GENHMIDENTITY} \(A\)} \]

[11] 

\[ \text{ELSE ILLEGAL TYPE} \]

[12] 

\[ \text{PRINT 'ILLEGAL \(H\) MATRIX TYPE. TYPE HELP FOR INFORMATION'} \]

[13] 

\[ \rightarrow \text{GENHMIN} \]

[14] 

\[ \text{USER WILL GIVE \(H\) MATRIX VALUES} \]

[15] 

\[ \text{GENHM\text{GIVEN}; Z \rightarrow M \text{ GGIVEN}\text{N}} \]

[16] 

\[ \rightarrow 0 \]

[17] 

\[ \text{IDENTITY MATRIX GENERATOR} \]

[18] 

\[ \text{FIRST CHECK TO MAKE SURE AN IDENTITY MATRIX IS APPROPRIATE HERE} \]

[19] 

\[ \text{GENHMIDENTITY; } \rightarrow (M = N) / \text{GENHMIDENTITYGET} \]

[20] 

\[ \text{PRINT 'THESE TWO PHASES DO NOT HAVE THE SAME NUMBER OF STATES.'} \]

[21] 

\[ \text{PRINT 'AN IDENTITY MATRIX IS INAPPROPRIATE FOR THE INTERPHASE TRANSITION MATRIX.'} \]

[22] 

\[ \rightarrow \text{GENHMMIN} \]

[23] 

\[ \text{GENHMIDENTITYGET; Z \rightarrow GIDENTITY\text{N}} \]

[24] 

\[ \rightarrow 0 \]
GENERATEPMATRIX

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{GENERATEPMATRIX} \ arg \)

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 \( \text{arg} \times \text{arg} \). Arg is typically the number of states in the phase associated with the P matrix.

The P matrix obtained is returned in
LISTING:

`\[\begin{align*}
V & \text{-GENERATE\_MATRIX } N; \text{TYPE;IN} \\
& \text{A} \\
& \text{A} \\
& \text{ROUTINE FOR SUPERVISING THE CONSTRUCTION OF THE P MATRIX} \\
& \text{FOR THE GIVEN PHASE} \\
& \text{THE MATRIX IS TO HAVE SHAPE } N \times N \\
& \text{A} \\
& \text{GET TYPE OF MATRIX} \\
& \text{PRINT ''} \\
& \text{GENPMIN:PRINT 'WHAT TYPE OF P MATRIX?'} \\
& \text{A INPUT GPM} \\
& \text{A CHECK FOR COMMAND} \\
& \text{A } (1 = \& \text{e'} \text{COMMAND'})/\text{GENPMIN} \\
& \text{A CHECK FOR TYPE} \\
& \text{A } (\&/\text{PMATRIXLIST}\text{e}( \\
& \text{IN)))}/\text{GENPMGIVEN,GENPMDEDFAIL,GENPMNFAIL,GENPMIDENTITY} \\
& \text{ELSE ILLEGAL TYPE} \\
& \text{PRINT 'ILLEGAL P MATRIX TYPE. TYPE HELP FOR INFORMATION'} \\
& \text{A GENPMIN} \\
& \text{A USER WILL GIVE P MATRIX VALUES} \\
& \text{GENPMGIVEN:Z-N GGIVEN N} \\
& \text{A} \\
& \text{A} \\
& \text{DEDICATED COMPONENT SYSTEM} \\
& \text{GENPMDEDFAIL:Z-CDEDFAIL N} \\
& \text{IF ROUTINE FAILED, GO BACK TO TRY AGAIN. ELSE EXIT} \\
& \text{ROUTINE} \\
& \text{A } ((Z[1;1]<0),1)/\text{GENPMIN,0} \\
& \text{A} \\
& \text{A} \\
& \text{N GROUPS OF COMPONENTS SYSTEM} \\
& \text{GENPMNFAIL:Z-CNFAIL N} \\
& \text{A} \\
& \text{A} \\
& \text{IDENTITY MATRIX GENERATOR} \\
& \text{GENPMIDENTITY:Z-CIDENTITY N} \\
& \text{A} \\
& \text{V}
\end{align*}\]

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
GETACCLEVPROB

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE:  \[ \text{result} \leftarrow \text{GETACCLEVPROB} \ \text{arg} \]

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

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

CALLING FUNCTIONS:  GETPERFORMABILITY.

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.

7. GETACCLEVPROB METAPHOR Function Description
LISTING:
\[ V \leftarrow \text{GETACCLEVELPROB LEVEL}; T; \text{TRA JPROB}; \text{NUMTRAJSETS}; I; G; F; V \]

\[ 1 \]
\[ \text{ROUTINE FOR DETERMINING THE PROBABILITY OF AN ACCOMPLISHMENT LEVEL} \]

\[ 2 \]
\[ \text{INITIALIZE THE COLLECTION VARIABLE} \]

\[ 3 \]
\[ Z \leftarrow 0 \]

\[ 4 \]
\[ \text{GET THE NUMBER OF TRAJECTORY SETS} \]

\[ 5 \]
\[ \text{NUMTRAJSETS} \leftarrow \text{GETNUMTRAJSETS LEVEL} \]

\[ 6 \]
\[ \text{LOOP THROUGH FOR EACH SET, GETTING ITS VALUES AND CALCULATING ITS PROBABILITY} \]

\[ 7 \]
\[ \text{INITIALIZE COUNTER} \]

\[ 8 \]
\[ T \leftarrow 1 \]

\[ 9 \]
\[ \text{GAPTLOOP:} \]
\[ T \leftarrow (\text{PRINTQUAD 'TRAJECTORY SET '}); \text{PRINTQUAD T}) \]

\[ 10 \]
\[ \text{PRINT ''} \]

\[ 11 \]
\[ \text{GETIVECTOR} \]

\[ 12 \]
\[ \text{GETGMATRICES} \]

\[ 13 \]
\[ \text{GETFVECTOR} \]

\[ 14 \]
\[ \text{GETVVALUES} \]

\[ 15 \]
\[ \text{TRAJPROB} \leftarrow \text{CALCTRAJPROB} \]

\[ 16 \]
\[ \text{ADD THE PROBABILITY TO THE COLLECTION VARIABLE} \]

\[ 17 \]
\[ Z \leftarrow Z + \text{TRAJPROB} \]

\[ 18 \]
\[ \text{INCREMENT TRAJECTORY COUNTER AND BRANCH IF NECESSARY} \]

\[ 19 \]
\[ T \leftarrow T + 1 \]

\[ 20 \]
\[ \rightarrow (T \leftarrow \text{NUMTRAJSETS})/\text{GAPTLOOP} \]

\[ 21 \]
\[ \text{RESET DEFINITION FLAGS} \]

\[ 22 \]
\[ \text{DEFNUMTRAJSETS} \leftarrow \text{DEFI} \leftarrow \text{DEFG} \leftarrow \text{DEFF} \leftarrow \text{DEFV} \leftarrow 0 \]
GETALTERVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \[ \text{result} \leftarrow \text{GETALTERVECTOR} \]

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:

\begin{verbatim}
\v Z+GETALTERVECTOR;GAVINPUT
[1] a ROUTINE FOR GETTING THE ALTER VECTOR FOR CHANGING DATA
[2] a
[3] a PRINT EXPLANATION
[4] GAVIN;PRINT 'PUT AN X BELOW EACH ITEM TO BE CHANGED. HELP
AVAILABLE.'
[5] a GET ALTER REQUESTS, RESHAPING ALONG THE WAY
[6] PRINT 'P H CONST.BAS.VARS ALL.ACC.LEVELS
PRESENT.ACC.LEVEL I G F V NUM.TRAJ.SETS'
[7] GAVINPUT+80p(M,(80p ' '))
[8] a LOOK FOR HELP REQUEST. IF PRESENT, CALL FOR HELP ROUTINE.
[9] (~'HELP'GAVINPUT)/GAVVECTORSET
[10] COMMANDHELP GAV
[12] GAVVECTORSET:Z=0p0
[13] a DETERMINE CHANGE VECTOR
\end{verbatim}

\textit{METAPHOR (VERSION 1) PROGRAMMER'S GUIDE}
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 them 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:

V GETBASICVARIABLES

[1] a
[3] a
[4] a SEE IF IT IS NECESSARY TO INPUT BASIC VARIABLES
[5] \(\text{\#}(\text{NUMBASICVARIABLES}=0)/0\)
[6] a
[7] a INPUT PROBABILITIES AND CHECK VALIDITY
[8] PRINT '!
[10] NUMBASICVARIABLES INPUT GBV
[11] a CHECK FOR COMMAND
[12] \(+(1=a \text{\textquoteleftCOMMAND\textquoteright})/\text{GBVIN}\)
[13] a CHECK FOR PROPER PROBABILITY MAGNITUDE
[14] \(+(\text{CHECKPROB IN})/\text{GBVIN}\)
[15] a SET BASIC VARIABLE PROBABILITIES AND EXIT

7. GETBASICVARIABLES METAPHOR FUNCTION DESCRIPTION
[16] BASICVARIABLES<IN,1-IN
[17] DEFBASICVARIABLES<1
GETDATAVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \[ \text{result} \leftarrow \text{GETDATAVECTOR} \]

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.
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:

V Z+GETDATAVECTOR;GDVINPUT

[1] a

[2] a ROUTINE FOR GETTING THE DATA VECTOR FOR DISPLAYING DATA

[3] a

[4] a PRINT EXPLANATION

[5] GDVIN:PRINT 'PUT AN X BELOW EACH ITEM TO BE DISPLAYED. HELP AVAILABLE.'

[6] a 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+69p(M,(69p ','))

[9] a LOOK FOR HELP REQUEST. IF PRESENT, CALL FOR HELP ROUTINE

[10] →(~'/HELP'EGDVINPUT)/GDVINPUTCONT


[12] →GDVIN

[13] GDVINCONT:PRINT

'NUM.ACC.LEVELS NUM.TRAJ.SEVERE'S I G F V PERF'

[14] GDVINPUT+116p(GDVINPUT,M,(47p ','))

[15] PRINT ''

[16] a LOOK FOR HELP REQUEST. IF PRESENT, CALL FOR HELP ROUTINE

[17] →(~'/HELP'EGDVINPUT)/GDVVECTORSET

[18] COMMANDHELP GDV

[19] →GDVIN

[20] a INITIALIZE DISPLAY VECTOR

[21] GDVVECTORSET:Z+13p0

[22] a DETERMINE DISPLAY VECTOR

[23] Z[1]='X'EGDVINPUT[t10]


[27] Z[5]='X'EGDVINPUT[t30+t18]


[29] Z[7]='X'EGDVINPUT[t69+t14]


[31] Z[9]='X'EGDVINPUT[t101]


[34] Z[12]='X'EGDVINPUT[t110]
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:

[1] ROUTINE FOR FETCHING THE NUMBER OF MISSION ACCOMPLISHMENT LEVELS
[2] INPUT NUMBER OF ACCOMPLISHMENT LEVELS
[3] PRINT 'NUMBER OF ACCOMPLISHMENT LEVELS?'
[4] INPUT GNA
[5] CHECK FOR COMMAND
[6] +(1=\"COMMAND\")/GNA
[7] CHECK VALIDITY OF INPUT
[8] +(CHECKPOSI IN)/GNA
[9] ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT
[10] NUMACCLEV+IN
[11] DEFNUMACCLEV+1

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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] a
[2] a
[3] a ROUTINE FOR FETCHING THE NUMBER OF TIME-IN Variant BASIC VARIABLES
[4] a
[5] a INPUT NUMBER OF TIME-IN Variant BASIC VARIABLES
[6] PRINT ''
[7] GNBVIN:PRINT 'NUMBER OF TIME-IN Variant BASIC VARIABLES?'
[8] 1 INPUT GNBV
[9] a CHECK FOR COMMAND
[10] !(I=A 'COMMAND')/GNBVIN
[12] !(IN=0)/GNBVSET
[13] +(CHECKPOSI IN)/GNBVIN
[14] a ELSE SET THE NUMBER OF TIME-IN Variant BASIC VARIABLES AND EXIT
[15] GNBVSET:NUMBASICVARIABLES+IN
[16] DEFNUMBASICVARIABLES+1

7. GETNUMBASICVARIABLES METAPHOR FUNCTION DESCRIPTION
GETNUMPHASES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETNUMPHASES

PURPOSE: To obtain the number of phases.

GLOBAL VARIABLES: DEFNUMPHASES, IN, NUMPHASES, GNP.

CALLING FUNCTIONS: COMMANDDEVAL.

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:

```
V GETNUMPHASES

ROUTE FOR FETCHING THE NUMBER OF MISSION PHASES

INPUT NUMBER OF PHASES

PRINT ''

GNPIN:PRINT 'NUMBER OF PHASES?'

1 INPUT GNP

CHECK FOR COMMAND

+(1== 'COMMAND')/GNPIN

CHECK VALIDITY OF INPUT

+(CHECKPOSI IN)/GNPIN

ELSE SET THE NUMBER OF PHASES AND EXIT

NUMPHASES+IN

DEFNUMPHASES+1

V
```

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

ROUTINE FOR FETCHING THE NUMBER OF ACCOMPLISHMENT LEVEL TRAJECTORY SETS FOR LEVEL L

INPUT NUMBER OF TRAJECTORY SETS

PRINT ''

(PRINTQUAD 'ACCOMPLISHMENT LEVEL ?);PRINTQUAD L-1)

PRINT ''

GNTSIN:PRINT 'NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT LEVEL?'

1 INPUT GNTS

CHECK FOR COMMAND

+(1=A '&COMMAND')/GNTSIN

CHECK VALIDITY OF INPUT

+(CHECKPOSI IN)/GNTSIN

ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT

Z+IN
[17]  D E F N U N T R A J S E T S + 1

"METAPHOR (VERSION 1) PROGRAMMER'S GUIDE"
GETPERFORMABILITY

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GETPERFORMABILITY

PURPOSE: To determine the performability for the model given METAPHOR

GLOBAL VARIABLES: DEFACCLEVEL, DEFFERFORMABILITY, 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] ROUTINE FOR COMPUTING THE PERFORMABILITY OF THE SYSTEM
[3] PERFORMABILITY+NUMACCLEV+0
[4] DEFACCLEVEL+1
[5] DEFFERFORMABILITY+1
[6] L+1
[7] LOOP THROUGH EACH ACCOMPLISHMENT LEVEL
[8] INITIALIZE LEVEL COUNTER
[9] GET THE PROBABILITY OF EACH ACCOMPLISHMENT LEVEL
[10] GETACCELVPROB; L
```

7. GETPERFORMABILITY METAPHOR FUNCTION DESCRIPTION
INSERT THE PROBABILITY INTO THE PERFORMABILITY VECTOR
PERFORMABILITY[L] += LEVELPROB
INCREMENT THE LEVEL COUNTER AND BRANCH IF NECESSARY
L = L + 1
→ (L ≤ NUMACCLEV)/GPLOOP
ELSE EXIT ROUTINE, SETTING DEFINITION FLAG
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 them 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:

\[ V \text{ GETSTATES} \]
\[ V \]
\[ ^{\cdot} \text{ ROUTINE FOR FETCHING THE NUMBER OF STATES IN EACH PHASE} \]
\[ ^{\cdot} \]
\[ ^{\cdot} \text{ INPUT NUMBER OF STATES AND CHECK VALIDITY} \]
\[ ^{\cdot} \text{ PRINT } \]
\[ ^{\cdot} \text{ GSIN;} \text{ PRINT 'NUMBER OF STATES PER PHASE? (SPACE BETWEEN EACH NUMBER)'} \]
\[ ^{\cdot} \text{ NUMPHASES INPUT GS} \]
\[ ^{\cdot} \text{ CHECK FOR COMMAND} \]
\[ ^{\cdot} \text{ INalling } \text{ NOT } 1} \text{ GSIN} \]
\[ ^{\cdot} \text{ CHECK FOR POSITIVE INTEGER} \]
\[ ^{\cdot} \text{ CHECKPOSI IN) GSIN} \]
\[ ^{\cdot} \text{ SET STATES AND EXIT} \]
\[ ^{\cdot} \text{ STATESPERPHASE+IN} \]
\[ ^{\cdot} \text{ MAXNUMSTATES IN) STATESPERPHASE} \]
\[ ^{\cdot} \text{ DEFNUMSTATES+1} \]
\[ \]

7. GETSTATES METAPHOR FUNCTION DESCRIPTION
GETFVECTOR

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{GETFVECTOR} \)

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

GLOBAL VARIABLES: \( \text{IN, STATESPERPHASE, GFV} \).

CALLING FUNCTIONS: \( \text{GETACCLEVPROB} \).

FUNCTIONS CALLED: \( \text{CHECKBIN, INPUT, PRINT} \).

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:

```
V Z GETFVECTOR
[1] a ROUTINE FOR INPUTTING FINAL STATE VECTOR
[2] a
[3] a
[7] a CHECK FOR COMMAND
[8] \( \rightarrow (1=\& \& \text{'COMMAND'})/GFVIN \)
[9] a CHECK VALIDITY OF INPUT.
[10] \( \rightarrow (\text{CHECKBIN IN})/GFVIN \)
[12] Z(\text{MAXNUMSTATES,1})\&IN,(\text{MAXNUMSTATESp0})
[13] DEFF+1
```

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

v z+GETGMATRICES;PHASE

[1] a
[2] a ROUTINE FOR INPUTTING THE G MATRICES FOR A TRAJECTORY SET
[4] a
[5] a INITIALIZE THE G DIAGONAL MATRIX. ALSO SET DEFINITION FLAG
[6] z+(MAXNUMSTATES,MAXNUMSTATES)p0
[7] DEFG+1
[8] a
[10] a INITIALIZE THE PHASE COUNTER
[11] PHASE+1
[12] a INPUT AND CHECK THE G VECTOR
[13] GGMIN[i]«((PRINTQUAD 'PHASE '):(PRINTQUAD PHASE);PRINTQUAD 'i';')

7. GETGMATRICES METAPHOR FUNCTION DESCRIPTION
PRINT ''
PRINT 'ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):'
STATESPERPHASE[PHASE]INPUT GGM
\nCHECK FOR COMMAND
\n→((1== 'COMMAND')/GGMIN
\nCHECK VALIDITY OF INPUT.
\n→(CHECKBIN IN)/GGMIN
\nPLACE THE INPUT IN THE SET OF G VECTORS
Z[PHASE;\pIn]+IN
\nINCREMENT THE PHASE COUNTER AND BRANCH IF APPROPRIATE
PHASE+PHASE+1
\n→((PHASE<NUMPHASES))/GGMIN
\n
METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

V GETHMATRICES;PHASE;NEXTH
[1] a ROUTINE FOR INPUTTING THE H MATRICES
[2] a ONE MATRIX FOR EACH PHASE
[3] a
[4] a IF NO H MATRICES, SET H+1 AND LEAVE
[5] +(NUMPHASES>2)/GHMMULTIPHASE
[6] H+ 1 1,1p1
[7] a INITIALIZE THE ARRAY OF H MATRICES
[8] GHMMULTIPHASE:H=((NUMPHASES-1),MAXNUMSTATES, MAXNUMSTATES)p0
[9] a
[10] a INPUT AND CHECK THE H MATRICES
[12] PRINT ''
[13] PRINT 'SPECIFY THE H MATRICES FOR EACH PHASE, 1 PHASE AT A TIME'
[14] a

7. GETHMATRICES METAPHOR FUNCTION DESCRIPTION
[17] A
[18] A INITIALIZE PHASE COUNTER
[19] PHASE+2
[20] GHMPHASEIN:PRINT ""
[21] +((PRINTQUAD 'PHASE ')(PRINTQUAD PHASE-1)(PRINTQUAD ' -')(PRINTQUAD PHASE)(PRINTQUAD ' : '))
[22] PRINT ""
[23] A GET H MATRIX FOR PHASE
STATESPERPHASE[PHASE]
[26] H[PHASE-1; STATESPERPHASE[PHASE-1];
\STATESPERPHASE[PHASE]]+NEXTH
[27] A
[28] A INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[29] PHASE+PHASE+1
[30] \(PHASE \leq NUMPHASES) / GHMPHASEIN
[31] A ELSE SET H DEFINITION FLAG AND LEAVE
[32] DEFI-1
\v
GETIVECTOR
METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{GETIVECTOR} \)

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

GLOBAL VARIABLES: \( \text{IN}, \text{STATESPERPHASE}, \text{GIV} \).

CALLING FUNCTIONS: GETACCLEVPROM.

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 \( \text{result} \). If these values are not proper binary quantities (i.e., 0 or 1), the query is repeated. In addition, the definition flag associated with \( \text{I} \) is set so that the \( \text{I} \) vector can be altered or displayed.

LISTING:

\begin{verbatim}
\textbf{ROUTINE FOR INPUTTING INITIAL STATE VECTOR}
\textbf{INPUT AND CHECK THE I VECTOR}
\textbf{GIVIN:PRINT 'ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):'}
\textbf{STATESPERPHASE[i]INPUT GIV}
\textbf{CHECK FOR COMMAND :}
\textbf{\( (i \neq \text{COMMAND'})/\text{GIVIN} \)}
\textbf{CHECK VALIDITY OF INPUT.}
\textbf{\( (\text{CHECKBIN IN})/\text{GIVIN} \)}
\textbf{SET I VECTOR AND LEAVE}
\textbf{\text{Z}+\text{MAXNUMSTATESpIN},(\text{MAXNUMSTATESp0})}
\textbf{DEFI+1}
\end{verbatim}

7. GETIVECTOR METAPHOR FUNCTION DESCRIPTION
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:

```
V GETPMATRICES;PHASE;NEXTP
[1] a
[2] a ROUTINE FOR INPUTTING THE P MATRICES
[3] a
[4] a ONE MATRIX FOR EACH PHASE
[5] a
[7] P=(NUMPHASES,MAXNUMSTATES,MAXNUMSTATES)p0
[8] a
[9] a INPUT AND CHECK THE P MATRICES
[10] PRINT ''
[12] PRINT 'SPECIFY THE P MATRICES FOR EACH PHASE, 1 PHASE AT A TIME'
[13] a
[14] a
[15] a INITIALIZE PHASE COUNTER
[16] PHASE-1
[17] GPMPHASEIN:PRINT ''
```

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINTQUAD 'PHASE '); PRINTQUAD PHASE ); PRINTQUAD ' ; ' )
PRINT ' ;'
GET P MATRIX FOR PHASE
NEXTP + GENERATEPMATRIX STATESPERPHASE[PHASE]
INSERT THE MATRIX INTO THE ARRAY OF MATRICES
P[PHASE; STATESPERPHASE[PHASE];
STATEPERPHASE[PHASE]]) + NEXTP

INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
PHASE + PHASE + 1

ELSE SET P DEFINITION FLAG AND LEAVE
DEFP + 1

7. GETPMATRICES METAPHOR FUNCTION DESCRIPTION
GETVALUES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{GETVALUES} \)

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

GLOBAL VARIABLES: \( \text{IN}, \text{NUMBASICVARIABLES}, \text{GVV} \).

CALLING FUNCTIONS: \( \text{GETACCLEVPROB} \).

FUNCTIONS CALLED: \( \text{CHECKBIN}, \text{INPUT}, \text{PRINT} \).

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:
\[
\text{V Z-GETVALUES;GVVTRUE;GVVFALSE} \\
[1] \text{A} \\
[2] \text{A ROUTINE FOR INPUTTING TIME+INARIANT BASIC VARIABLE PROBABILITIES} \\
[3] \text{A} \\
[4] \text{A INPUT AND CHECK THE V VECTOR} \\
[5] \text{A IF NO TIME-INARIANT BASIC VARIABLES, THEN SET V=0, TURN ON THE DEFINITION FLAG, AND EXIT} \\
[6] \text{Z=0} \\
[7] \text{DEFP+1} \\
[8] \rightarrow(\text{NUMBASICVARIABLES}=0)\,/\,0 \\
[9] \text{GVVIN}:=+((\text{PRINTQUAD 'ENTER THE '});(\text{PRINTQUAD \text{NUMBASICVARIABLES}});\text{PRINTQUAD ' ELEMENT TIME-INARIANT BASIC VARIABLE VECTOR (SPACE BETWEEN EACH ENTRY): '}) \\
[10] \text{PRINT ' '}. \\
[11] \text{NUMBASICVARIABLES INPUT GV} \\
[12] \text{A CHECK FOR COMMAND} \\
[13] \rightarrow(1=\land \epsilon 'COMMAND')/\text{GVVIN} \\
\]

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
\[ 14 \] \textit{CHECK VALIDITY OF INPUT.}\n\[ 15 \] \textit{CHECKTRI IN})/GVVIN\n\[ 16 \] \textit{SET V VECTOR}\n\[ 17 \] \textit{FIRST, SET THE TRUE VECTOR}\n\[ 18 \] GVVTRUE+IN\in \{0,2\}\n\[ 19 \] \textit{THEN SET THE FALSE VECTOR}\n\[ 20 \] GVVFALSE+IN\in \{1,2\}\n\[ 21 \] \textit{COMBINE FOR THE V VECTOR}\n\[ 22 \] Z=GVVTRUE,GVVFALSE
\]
INPUT

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \texttt{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, GENUMACCLEV, GENUMBASICEVARIABLES, GENUMPHASES, GENUMTRAJSETS, GETSTATES, GETVECTOR, GETGMATRICES, GETIVECTOR, GETVALUES, METAPHOR.

FUNCTIONS CALLED: PRINTQUAD, COMMANDALTER, COMMANDBRIEF, COMMANDCALC, COMMANDCOM, COMMANDDATA, COMMANDECHO, COMMANDVAL, 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 \texttt{arg1}. If not, 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 \texttt{arg2} is an indicator of the calling function and is used by the HELP function.

LISTING:

\texttt{v SHAPE INPUT ROUTINE;COMMANDVECTOR}
\begin{verbatim}
[1] a ROUTINE FOR INPUTTING ANSWERS. INPUT CAN BE DATA OR COMMANDS.
[2] a IF NUMERIC, DATA RETURNED IN 'IN', ELSE IF COMMAND, 'COMMAND' RETURNED
[3] a GET INPUT
[4] a ININ:IN=I
[5] a ECHO INPUT IF DESIRED
[6] a (~ECHOINPUT)/INVECTORIZE
\end{verbatim}

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
[9] "-((PRINTQUAD '0: '):PRINTQUAD IN)
[10] PRINT '
[12] a CHANGE ALL SCALAR INPUTS TO VECTORS
[13] INVECTORIZE:IN-,IN
[14] a
[15] a CHECK FOR COMMAND AND SET COMMAND VECTOR
[16] COMMANDVECTOR-=V/COMMANDLISTeIN
[17] a
[18] a
[19] a EXECUTE COMMAND IF PRESENT
[20] INCOMMAND:=(COMMANDVECTOR)/INHELP,INEXIT,INDATA,
INALTER,INCALC,INECHO,INBRIEF,INCOM,INEVAL
[21] a ELSE CHECK THE SHAPE OF THE INPUT. 'SHAPE' IS ALWAYS AN
INTEGRAL SCALAR. IF GOOD, LEAVE.
[22] -(SHAPE=OIN)/O
[23] a OTHERWISE THE INPUT WAS OF THE WRONG DIMENSION. PRINT ERROR
MESSAGE AND TRY AGAIN.
[24] "-((PRINTQUAD 'ENTER ');(PRINTQUAD SHAPE);PRINTQUAD 'ITEMS')
[25] -ININ
[26] -O
[27] a
[28] a
[29] a HELP REQUESTED
[30] INHELP:COMMANDHELP ROUTINE
[31] IN='COMMAND'
[32] -O
[33] a
[34] a
[35] a END THE PROGRAM
[36] INEXIT:
[37] a
[38] a
[39] a DISPLAY OF DATA REQUESTED
[40] INDATA:COMMANDDATA
[41] IN='COMMAND'
[42] -O
[43] a
[44] a
[45] a CHANGE OF DATA REQUESTED
[46] INALTER:COMMANDALTER
[47] IN='COMMAND'
[48] -O
[49] a
[50] a
[51] a CALCULATION OF PERFORMABILITY REQUESTED
[52] INCALC:COMMANDCALC
[53] IN='COMMAND'
[54] -O
[55] a
[56] a SETTING OF THE ECHO SWITCH DESIRED
[57] INECHO:COMMANDECHO IN
IN+ 'COMMAND'
\rightarrow 0

COMMENT DESIRED

INCOM:COMMANDCOM
IN+ 'COMMAND'
\rightarrow 0

SETTING OF BRIEF SWITCH DESIRED

INBRIEF:COMMANDBRIEF IN
IN+ 'COMMAND'
\rightarrow 0

PERFORMABILITY COMPUTATION DESIRED

INEVAL:COMMANDEVAL
\rightarrow 0

\triangledown
INYES

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE:  \texttt{result \leftarrow INYES}

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

GLOBAL VARIABLES:  \texttt{ECHOINPUT, IN}.

CALLING FUNCTIONS:  \texttt{GDEDFAIL, GNFAIL, GAVINFO, GBVINFO, GDINFO, GDVINFO, GFVINFO, GGINFO, GGMINFO, GHMINFO, GIVINFO, GNINFO, GNAINFO, GNEVINFO, GNPIINFO, GNITSINFO, GPMINFO, GSINFO, GVVINFO, METINFO}.

FUNCTIONS CALLED:  \texttt{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,' \texttt{INYES} returns a 1 in \texttt{result}, else if 'NO,' a 0 is returned.

7. INYES METAPHOR Function Description
LISTING:

\[ Z=\text{INYES;IN} \]

[1] 
[2] ROUTINE ASKING YES AND NO ANSWERS. 1 RETURNED IF YES, 0 IF NO

[3] 
[4] PRINT PROMPT AND GET ANSWER

[5] \( \text{ININ}:\text{PRINT '0:'} \)

[6] \( \text{IN}+\text{V} \)

[7] ECHO THE INPUT IF DESIRED

[8] \( \text{ECHOINPUT} / \text{TYSCAN} \)

[9] \( \text{PRINT(IN)} \)

[10] ASSUME YES HAS PRIORITY. LOOK FOR Y OR 1

[11] \( \text{TYSCAN}:Z='Y'v'\text{IN} \)

[12] IF N OR 0 INPUT OR IF YES INPUT, EXIT

[13] \( \text{TYSCAN}:Z='NO'v'\text{IN} / 0 \)

[14] ELSE TRY AGAIN

[15] \( \text{PRINT 'ENTER YES OR NO'} \)

[16] \( \text{TYIN} \)

\[ \text{\textbackslash} \]
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:

```
V METAPHOR
[1] a
[2] a
[3] a
[4] a PROGRAM FOR EVALUATION OF PERFORMABILITIES
[5] a
[6] a
[7] a SET PROGRAM CONSTANTS
[8] DECLAREMETAPHOR
[9] a
[10] a
[12] PRINT ''
[13] PRINT ''
[14] PRINT 'MICHIGAN EVALUATION AID FOR PERFORMABILITY'
[15] PRINT ''
[16] PRINT ''
[17] PRINT ''
[18] PRINT ''
[19] PRINT 'TYPE HELP FOR ASSISTANCE'
```
PROGRAM LOOPS UNTIL EXIT TAKEN
MIN:6 INPUT MET
\rightarrow_{\text{MIN}}
PRINT

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \texttt{result} \leftarrow \texttt{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, GETVECTOR, GETGMATRICES, GETHMATRICES, GETIVECTOR, GETPMATRICES, GETVALUES, INYES, METAPHOR, PRINTQUAD, PRINTPERFORMABILITY, COMMANDALTER, COMMANDBRIEF, COMMANDDATA, COMMANDECHO, GDEDFAIL, GGIVEN, GNFAIL, GAVINFO, GBVINFO, GDINFO, GDVINFO, CPVINFO, CGINFO, GGMINFO, GHMINFO, GIVINFO, GINFO, GAINFO, GNBVINFO, GPINFO, GNTINFO, GPMINFO, GSNINFO, GVVINFO, METINFO.

FUNCTIONS CALLED: NONE.

COMMENTS: \texttt{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:

\begin{verbatim}
V Z=PRINT Q
[1] a
[2] a
[3] a PRINTING ROUTINE FOR METAPHOR
[4] a IF BRIEF=1 NO OUTPUT IS GIVEN
[5] a
[6] a CHECK FOR TERSE INPUT FLAG
[7] a \rightarrow(BRIEFOUTPUT=1)/0
[8] a FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[9] Z=Q
\end{verbatim}

7. PRINT METAPHOR FUNCTION DESCRIPTION
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 not sum to one, the warning cannot be suppressed.

LISTING:

```plaintext
V PRINTPERFORMABILITY
[1] a ROUTINE FOR PRINTING THE PERFORMABILITY ONCE IT IS CALCULATED
[2] a
[4] PRINT '
[5] PRINT ''
[6] PRINT '''
[7] 'PERFORMABILITY FOR THIS MISSION = \$PERFORMABILITY
[8] a
[9] a 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'
```

' METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

V Z=PRINTQUAD Q
[1] a
[2] a
[3] a PRINTING ROUTINE (WITH QUOTE QUAD) FOR METAPHOR
[4] a IF BRIEF+1 NO OUTPUT IS GIVEN
[5] a
[6] a CHECK FOR TERSE INPUT FLAG
[7] +(BRIEFOUTPUT=1)/PQBRIEF
[8] a FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[10] →0
[12] PQBRIEF:Z=''

7. PRINTQUAD METAPHOR FUNCTION DESCRIPTION
**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, GETVALUES, 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:

V COMMANDALTER;ALTERVECTOR

A ROUTINE FOR CHANGING CURRENT DATA
A
A GET DATA TO BE ALTERED
A ALTERVECTOR+GETALTERVECTOR
A
A CHANGE THAT DATA. EXIT WHEN THROUGH.
A ALOOP:ALTERVECTOR/AP,AH,ABASICVARIABLES,AALLACCLEVELS,
A APRESENTACCLEVEL,AI,AG,AF,AV,ANUMTRAJSET
A
A CHANGE THE REQUESTED DATA
A
A CHANGE THE P MATRICES
A AP:=(DEFP=1)/APALTER
A P MATRICES ARE NOTDEFINED AT THIS TIME. PRINT ERROR MESSAGE
A PRINT 'P MATRICES ARE NOT DEFINED AT THIS TIME.'
A ALOOP
A APALTER:PRINT 'ALTERING P'
A GETP MATRICES
A ALOOP:ALTERVECTOR[1]+0
A ALOOP
A
A CHANGE H MATRICES
A AH:=(DEFH=1)/AHALTER
A H MATRICES ARE NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE
A PRINT 'H MATRICES ARE NOT DEFINED AT THIS TIME.'
A A LOOP
A AHALTER:PRINT 'ALTERING H'
A GETH MATRICES
A A LOOP:ALTERVECTOR[2]+0
A A LOOP

7. COMMANDALTER METAPHOR FUNCTION DESCRIPTION
[33] #
[34] # CHANGE THE TIME-IN Variant Basic Variables
[35] #
[36] ABASICVARIABLES:=(
[37] DEFBASICVARIABLES=1)/ABASICVARIABLESALTER
[38] #TIME-IN Variant Basic Variables Are Not Defined At This
[39] # time. Print Error Message
[40] PRINT 'TIME-IN Variant Basic Variables Are Not Defined At
[41] # This Time.'
[42] →ABASICVARIABLESLOOP
[43] ABASICVARIABLESALTER:PRINT 'ALTERING THE NUMBER OF BASIC
[44] # VARIABLES'
[45] GETNUMBASICVARIABLES
[46] GETBASICVARIABLES
[48] →ALoop
[49] #
[50] # CHANGE ALL Accomplishment Levels
[51] AALLACCLEVELS:=(DEFNUMACCLEV=1)/AALLACCLEVELSALTER
[52] # The Accomplishment Levels Are Not Defined At This
[53] # time. Print Error Message
[54] PRINT 'The Accomplishment Levels Are Not Defined At This
[55] # Time.'
[56] →AALLACCLEVELSLOOP
[57] AALLACCLEVELSALTER:PRINT 'ALTERING THE NUMBER OF
[58] # Accomplishment Levels'
[59] GETNUMACCLEV
[60] GETACCLEVPROB
[62] →ALoop
[63] #
[64] # CHANGE Just The Present (Last) Accomplishment Level
[65] APRESENTACCLEVEL:=(DEFACCLEVEL=1)/APRESENTACCLEVELALTER
[66] # An Accomplishment Level Is Not Defined At This
[67] # time. Print Error Message
[68] PRINT 'An Accomplishment Level Is Not Defined At This
[69] # Time.'
[70] →APRESENTACCLEVELLOOP
[71] APRESENTACCLEVELALTER:PRINT 'ALTERING THE PRESENT
[72] # Accomplishment Level'
[73] PERFORMABILITY[1]+GETACCLEVPROB L
[74] APRESENTACCLEVELLOOP:ALTERVECTOR[5]+0
[75] →ALoop
[76] #
[77] # CHANGE The I Vector
[78] AI:=(DEFI=1)/AIALTER
[79] # I Vector Is Not Defined At This Time. Print Error Message
[80] PRINT 'I Vector Is Not Defined At This Time.'
[81] →ALoop
[82] AIALTER:PRINT 'ALTERING THE I Vector'
[83] I+GETIVECTOR
[84] AILoop:ALTERVECTOR[6]+0
[85] →ALoop

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
7. COMMAND ALTER METAPHOR FUNCTION DESCRIPTION

[75] a
[76] a 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] →GGETGMATRICES
[84] ←ALOOP
[85] a
[86] a 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] →FGETFVECTOR
[93] AFLOOP:ALTERVECTOR[8]=0
[94] ←ALOOP
[95] a
[96] a CHANGE THE TIME+INVARIANT BASIC VARIABLE VECTOR
[97] AV:=(DEFF=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] →VGETVALUES
[103] AVLOOP:ALTERVECTOR[9]=0
[104] ←ALOOP
[105] a
[106] a CHANGE THE NUMBER OF TRAJECTORY SETS BEING CONSIDERED
[107] ANUMTRAJSETS:=(DENUMTRAJSETS=1)/ANUMTRAJSETSAFTER
[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] ANUMTRAJSETSAFTER:PRINT 'ALTERING THE NUMBER OF TRAJECTORY SETS'
[111] NUMTRAJSETS=GETNUMTRAJSETS
[112] ALTERVECTOR[10]=0
[113] v
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 be either ON or OFF. A confirmation is also printed.

LISTING:

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

v
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:

```
V COMMANDCALC;CCALCIN

[1] a
[2] a ROUTINE TO UTILIZE THE APL CALCULATOR MODE FROM THE METAPHOR PACKAGE.
[3] a HALTS WHEN THE INPUT IS NULL.
[4] a
[6] U+CCALCIN+]
[7] a LEAVE IF EXIT SPECIFIED. ELSE GET NEXT CALCULATION.
[8] +(1=^/CCALCIN=EXIT)0
[9] +CCALCIN
```

7. COMMANDCALC METAPHOR FUNCTION DESCRIPTION
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:

\[
\text{V COMMANDCOM;CCINPUT}
\]

[1] \(\text{a}\)
[2] \(\text{a}\) ROUTINE FOR ALLOWING THE USER TO PRINT A COMMENT
[3] \(\text{a}\) EACH COMMENT IS PRECEDED BY '***' AND THIS
[4] \(\text{a}\) HALTS WHEN THE INPUT IS NULL
[5] \(\text{a}\)
[6] \(\text{a}\) PRINT PROMPT SYMBOLS
[7] \(\text{CCIN}:=+(\text{PRINTQUAD '***'})\)
[8] \(\text{CCINPUT}-\)
[9] \(\text{a}\) IF COMMENT NOT EMPTY, GET NEXT COMMENT. ELSE LEAVE.
[10] \(\rightarrow(3\neq\text{CCINPUT})/\text{CCIN}\n\]
**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, DEFFPERFORMABILITY, 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.

7. **COMMANDDATA METAPHOR Function Description**
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:

```
V COMMANDDATA;DATAVECTOR
[1] a
[2] a ROUTINE FOR DISPLAYING CURRENT DATA
[3] a
[4] a GET DATA TO BE DISPLAYED
[5] DATAVECTOR=GETDATAVECTOR
[6] a
[7] a DISPLAY THAT DATA. EXIT WHEN THROUGH.
[8] DLOOP:-DATAVECTOR/DNUMPHASES,DNUMSTATES,DP,DH,DNUMBASICVAR
DBASICVARIABLES,DNUMACCLEV,DNUMTRAJSETS,D1,DG,DP,DV,DPERF
[9] ->0
[10] a
[12] a SHOW THE REQUESTED INFORMATION
[13] a
[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 ''
[20] ->DLOOP
[21] a
[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 ''
```

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
7. COMMANDDATA METAPHOR FUNCTION DESCRIPTION
DNUMTRAJSETSLOOP:DATAVECTOR[8]+0

DI:=(DEFI=1)/DIOUT
PRINT 'I VECTOR NOT DEFINED'
DILOOP
DIOUT:=(PRINTQUAD 'THE INITIAL VECTOR IS ');PRINTQUAD I
PRINT ''
DILOOP:DATAVECTOR[9]+0
DILOOP

DG:=(DEFG=1)/DGOUT
PRINT 'G MATRICES NOT DEFINED'
DGLOOP
DGOUT:PRINT 'THE G MATRICES ARE:'
G
DGLOOP:DATAVECTOR[10]+0
DGLOOP

DF:=(DEFF=1)/DFOUT
PRINT 'F VECTOR NOT DEFINED'
DFLOOP
DFOUT:=(PRINTQUAD 'THE FINAL VECTOR IS ');PRINTQUAD F
PRINT ''
DFLOOP

DV:=(DEFV=1)/DVOUT
PRINT 'THE TIME+INVARIANT BASIC VARIABLE VECTOR NOT DEFINED'
DVLOOP
DVOUT:=(PRINTQUAD 'THE TIME+INVARIANT BASIC VARIABLE VECTOR IS ');PRINTQUAD V)
PRINT ''
DVLOOP:DATAVECTOR[12]+0
DVLOOP

DPERF:=(DEPERFORMABILITY=1)/DPERFOUT
PRINT 'PERFORMABILITY NOT DEFINED'
DPERFLOOP
DPERFOUT:=(PRINTQUAD 'THE PERFORMABILITY IS ');PRINTQUAD PERFORMABILITY
DPERFLOOP:DATAVECTOR[13]+0
DPERFLOOP

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:
V COMMANDECHO SWITCH
[1] a ROUTINE FOR TURNING THE ECHO SWITCH ON AND OFF. 'ON' CAUSES
[2] a METAPHOR TO REPEAT EVERY INPUT LINE. 'OFF' SUPPRESSES THE REPETITION.
[3] a
[4] a TURN SWITCH ON IF REQUESTED, ELSE TURN SWITCH OFF.
[5] a ECHOINPUT+\'ON\’;SWITCH
[6] a 'ECHO \';SWITCH
V
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:

V COMMANDEVAL
[1] a
[2] a PERFORMANCE COMPUTATION PORTION OF METAPHOR
[3] a
[4] a FETCH THE MATRICES REQUIRED FOR PERFORMABILITY CALCULATIONS
[6] GETSTATES
[7] GETPMATRICES
[8] GETHMATRICES
[9] GETNUMBASICVARIABLES
[10] GETBASICVARIABLES
[12] a
[13] a FOR EACH ACCOMPLISHMENT LEVEL, FETCH THE TRAJECTORY SETS AND CALCULATE THEIR PROBABILITIES
[14] GETPERFORMABILITY

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT THE RESULTING PERFORMABILITY
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, GBINFO, GDINFO, GDVINFO, GFINFO, GGINFO, GGINFO, GHMINFO, GIVINFO, GGINFO, GAINFO, GNBINFO, C-NBVINFO, GNPOINFO, GNINFO, GNINFO, GNVINFO, METINFO, 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:

[1] a
[2] a ROUTINE PRINTING HELP INFORMATION
[3] a
[4] a GET THE PROPER INFO ROUTINE
[5] +(ROUTINE=NUMHELPROUTINES)/HGNP,HGS,HGPM,HGG HGN,HGD,HGHM,HGMBV,HGBV,HGNA,HGIV, HGGM,HGFM,HGVV,HGNPS,HGAV,HGDV,HMET
[6] a
[7] a 'GET NUMBER OF PHASES' HELP
[8] HGNP:GNPINFO
[9] 0
[10] a
[11] a 'GET STATES' HELP
[12] HGS:GINFO
[13] 0
[14] a
[15] a 'GENERATE P MATRIX' HELP
[16] HGPM:GPMINFO
[17] 0
[18] a
[19] a 'GET GIVEN MATRIX' HELP

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
133

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

7. COMMANDHELP METAPHOR FUNCTION DESCRIPTION
GDEDFAIL

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{GDEDFAIL} \ arg \)

PURPOSE: To generate a DEDFAIL type matrix with shape \( \text{arg} \times \text{arg} \).

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 \( \text{arg} \) 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 \( \text{arg} \times \text{arg} \) 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 \( 2^{\times N} \) states), then the \( i \)-th row or column of the matrix represents the state determined as follows: take the binary representation of \((2^{\times N})-1\)-\( i \). 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:

7. GDEDFAIL METAPHOR Function Description
<table>
<thead>
<tr>
<th></th>
<th>11</th>
<th>10</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.00E0</td>
<td>9.9E-1</td>
<td>0.00E0</td>
<td>1.00E-3</td>
</tr>
<tr>
<td>01</td>
<td>0.00E0</td>
<td>0.00E0</td>
<td>9.99E-1</td>
<td>1.00E-3</td>
</tr>
<tr>
<td>00</td>
<td>0.00E0</td>
<td>0.00E0</td>
<td>0.00E0</td>
<td>1.00E0</td>
</tr>
</tbody>
</table>

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:**

```
V Z+GDEDFAIL N;T;LAMBDA;INDEX;I;J;SIZE;SUCCESS;FAIL
[1] a
[2] b ROUTINE FOR GENERATING THE DEDFAIL TYPE P MATRIX
[3] b THE MATRIX IS TO HAVE SHAPE N×N
[4] a
[5] a CHECK TO MAKE SURE THIS ROUTINE IS APPROPRIATE
[6] +(20N)=(20N)/GDN
[7] a PRINT EXPLANATION, NOTE FAILURE, AND EXIT
[8] PRINT 'THE NUMBER OF STATES IN DEDFAIL MUST BE A POWER OF TWO.'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE```
[9] \[1\text{PRINTQUAD 'THIS PHASE HAS '};\text{PRINTQUAD N};\text{PRINTQUAD STATES. ')}\]
[10] \[\text{PRINT }\]
[11] \[Z \left( N, N \right)^{-1}\] 1\]
[12] \[\rightarrow 0\]
[13] \[\text{a}\]
[14] \[\text{GET PHASE LENGTH}\]
[15] \[\text{GDIN:PRINT 'ENTER PHASE LENGTH'}\]
[16] \[1\text{INPUT GD}\]
[17] \[\text{a CHECK COMMAND}\]
[18] \[+( 1 = \text{'COMMAND'} ) / \text{GDIN}\]
[19] \[\text{CHECK FOR POSITIVE NUMBER}\]
[20] \[+( \text{IN} > 0 ) / \text{GDSETT}\]
[21] \[\text{PRINT 'LENGTH IN TIME. MUST BE POSITIVE'}\]
[22] \[\rightarrow \text{GDIN}\]
[23] \[\text{a}\]
[24] \[\text{SET T TO PHASE LENGTH AND GET FAILURE RATE}\]
[25] \[\text{GDSETT: T \leftarrow IN}\]
[26] \[\text{GDINL:PRINT 'ENTER COMPONENT FAILURE RATE'}\]
[27] \[1\text{INPUT GD}\]
[28] \[\text{a CHECK COMMAND}\]
[29] \[+( 1 = \text{'COMMAND'} ) / \text{GDINL}\]
[30] \[\text{CHECK FOR POSITIVE NUMBER}\]
[31] \[+( \text{IN} > 0 ) / \text{GDCHECKSIZE}\]
[32] \[\text{PRINT 'RATE IN FAILURES PER UNIT TIME. MUST BE POSITIVE'}\]
[33] \[\rightarrow \text{GDINL}\]
[34] \[\text{CHECK REASONABleness OF FAILURE RATE}\]
[35] \[\text{GDCHECKSIZE: \leftarrow \left( \text{IN} \geq 1 \text{E} - 10 \right) \wedge \left( \text{IN} \leq 0.1 \right) / \text{GDSETL}\]
[36] \[\text{PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT}\]
[37] \[\text{PRINTQUAD IN}\]
[38] \[\text{+( \text{IN} \geq 0.1 / \text{GDLBIG}}\]
[39] \[\text{PRINTQUAD 'IS SMALL'}}\]
[40] \[\rightarrow \text{GDYESNOIN}\]
[41] \[\text{GDLBIG: T \leftarrow \text{PRINTQUAD 'IS LARGE'}}\]
[42] \[\text{GDYESNOIN:PRINT ' FOR A FAILURE RATE. DO YOU WANT THIS VALUE?'}\]
[43] \[+( \text{INYES) / } \text{GDINL}\]
[44] \[\text{a}\]
[45] \[\text{SET LAMBDA TO FAILURE RATE AND PERFORM CALCULATIONS}\]
[46] \[\text{GDSETL: LAMBDA \leftarrow IN}\]
[47] \[\text{a}\]
[48] \[\text{INITIALIZE THE P MATRIX}\]
[49] \[Z \left( N, N \right)^{-1}\]
[50] \[\text{a DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX}\]
[51] \[\text{INDEX} \leftarrow q \left( 2 \text{E} N \right) \text{ENCODE(N} - 1 \text{N})\]
[52] \[\text{a}\]
[53] \[\text{LOOP THROUGH INDEX TO CREATE P}\]
[54] \[\text{a INITIALIZE LOOPS}\]
[55] \[i \leftarrow 1\]
[56] \[j \leftarrow 1\]
[57] \[\text{a DETERMINE THE NUMBER OF SUCCESS TRANSITIONS}\]
[58] \[\text{GDLOOP: SIZE} \leftarrow \text{INDEX} \left[ i \right] / \text{INDEX} \left[ j \right]\]
\[ \text{FIND THE SUCCESS AND FAILURE PROBABILITIES} \]
\[ \text{FAIL} = (1 - \star \text{LAMBDA} \times T) \times (\frac{\text{INDEX}[I]}{\text{SIZE}}) \]
\[ \text{SUCCESS} = \star \text{LAMBDA} \times T \times \text{SIZE} \]
\[ \text{Z}[I; J] = \text{SUCCESS} \times \text{FAIL} \times \text{SIZE} \]
\[ \times (\frac{\text{INDEX}[J]}{\text{INDEX}[I]} \times (\text{INDEX}[I] \text{INDEX}[J])) \]
\[ \text{INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE} \]
\[ J = J + 1 \]
\[ \rightarrow (J \leq N) / \text{GDLOOP} \]
\[ \text{RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE} \]
\[ J = 1 \]
\[ I = I + 1 \]
\[ \rightarrow (I \leq N) / \text{GDLOOP} \]
GGIVEN

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \[ \text{result} \leftarrow \text{arg1 \ G GIVEN \ arg2} \]

PURPOSE: To input a given \[ \text{arg1 \times \text{arg2}} \] matrix from the user.

GLOBAL VARIABLES: \text{IN, GG}.

CALLING FUNCTIONS: \text{GENERATEHMATRIX, GENERATEPMATRIX}.

FUNCTIONS CALLED: \text{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 \[ \text{arg2} \] elements.

LISTING:

```
V Z+M G GIVEN N; ROW

\[ (\text{result} \leftarrow \text{arg1 \ G GIVEN \ arg2}) \]

\text{PURPOSE: To input a given \[ \text{arg1 \times \text{arg2}} \] matrix from the user.}

\text{GLOBAL VARIABLES: IN, GG.}

\text{CALLING FUNCTIONS: GENERATEHMATRIX, GENERATEPMATRIX.}

\text{FUNCTIONS CALLED: INPUT, PRINT, PRINTQUAD.}

\text{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 \[ \text{arg2} \] elements.}

\text{LISTING:}

\text{\texttt{v z+M G GIVEN N; ROW}}

\text{\texttt{\[ (\text{result} \leftarrow \text{arg1 \ G GIVEN \ arg2}) \]}}

\text{\texttt{\text{PURPOSE: To input a given \[ \text{arg1 \times \text{arg2}} \] matrix from the user.}}}

\text{\texttt{\text{GLOBAL VARIABLES: IN, GG.}}}

\text{\texttt{\text{CALLING FUNCTIONS: GENERATEHMATRIX, GENERATEPMATRIX.}}}

\text{\texttt{\text{FUNCTIONS CALLED: INPUT, PRINT, PRINTQUAD.}}}

\text{\texttt{\text{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 \[ \text{arg2} \] elements.}}}
```

7. GGIVEN METAPHOR FUNCTION DESCRIPTION
[25] →GGIN
[26] a
[27] a INSERT THE ROW INTO THE MATRIX
[28] GGINSERT: Z[ROW;]→IN
[29] a
[30] a INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
[31] ROW→ROW+1 .
[32] →(ROW≤M)/GGIN
[33] a ELSE LEAVE

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
GIDENTITY

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \( \text{result} \leftarrow \text{GIDENTITY arg} \)

PURPOSE: To generate an \( \text{argxarg} \) identity matrix.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: GENERATEHMATRIX, GENERATEPMATRIX.

FUNCTIONS CALLED: None.

COMMENTS: Returns an \( \text{argxarg} \) identity matrix in \( \text{result} \).

LISTING:
\[
\begin{array}{l}
\text{V Z=GIDENTITY N} \\
\text{[1] a} \\
\text{[2] \text{routine for generating an N\times N identity matrix}} \\
\text{[3] a} \\
\text{[4] \text{return the matrix}} \\
\text{[5] Z+(N,N)p(1,(Np0))} \\
\end{array}
\]
GNFAIL

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: \[ \text{result} \leftarrow \text{GNFAIL arg} \]

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] function. Each digit of a number from the system denotes the state of the corresponding group, e.g., 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:

<table>
<thead>
<tr>
<th></th>
<th>(2,1)</th>
<th>(2,0)</th>
<th>(1,1)</th>
<th>(1,0)</th>
<th>(0,1)</th>
<th>(0,0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2,0)</td>
<td>0.000E0</td>
<td>9.802E-1</td>
<td>0.000E0</td>
<td>1.970E-2</td>
<td>0.000E0</td>
<td>9.901E-5</td>
</tr>
<tr>
<td>(1,1)</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>9.802E-1</td>
<td>9.851E-3</td>
<td>9.901E-5</td>
<td>9.901E-3</td>
</tr>
<tr>
<td>(1,0)</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>9.900E-1</td>
<td>0.000E0</td>
<td>9.950E-3</td>
</tr>
<tr>
<td>(0,1)</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>9.900E-1</td>
<td>9.950E-3</td>
</tr>
<tr>
<td>(0,0)</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>0.000E0</td>
<td>1.000E0</td>
</tr>
</tbody>
</table>

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 asked 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

7. GNFAIL METAPHOR Function Description
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:

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

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
A CHECK COMMAND
\((1=\& \in \text{\'COMMAND\'})/\text{GN}\)
A CHECK FOR POSITIVE INTEGER
\((\text{CHECKPOSI IN})/\text{GN}\)
A SET GROUPS TO NUMBER OF GROUPS AND GET COMPONENTS PER GROUP
\(\text{GROUPS IN}\)
\(\text{PRINT 'ENTER NUMBER OF COMPONENTS PER GROUP (SPACE BETWEEN EACH NUMBER):'}\)
\(\text{GROUPS INPUT GN}\)
A CHECK COMMAND
\((1=\& \in \text{\'COMMAND\'})/\text{GN}\)
A CHECK FOR POSITIVE INTEGER
\((\text{CHECKPOSI IN})/\text{GN}\)
A CHECK FOR THE PROPER NUMBER OF STATES
\((N=x/IN+)/\text{GNCALC}\)
\((\text{PRINTQUAD 'THERE ARE '};(\text{PRINTQUAD N});\text{PRINTQUAD ' STATES IN THIS PHASE.'};\text{THE PRODUCT OF [EACH COMPONENT']}\)
PRINT '
PRINT 'NUMBER PLUS 1] MUST BE THE NUMBER OF STATES.'
PRINT 'HOW MANY GROUPS'
\((\text{GROUPS INPUT GN})/\text{GN}\)
A SET NUM TO THE NUMBER OF POSITIONS FOR THE COMPONENTS IN THE GROUPS AND CALCULATE THE P MATRIX
\(\text{GNCALC:NUM=IN+1}\)
A .
A INITIALIZE THE P MATRIX
\((N,N)pO\)
A DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX
\(\text{INDEX=Q(\text{NUM})ENCODE(N-IN)}\)
A .
A LOOP THROUGH INDEX TO CREATE P
A INITIALIZE LOOPS
\(I=1\)
\(J=1\)
A FIRST DETERMINE THE PROPER COEFFICIENT FOR THE TERM UNDER CONSIDERATION
\(\text{GNLOOP:COEFF\times (INDEX[J];INDEX[I];)}\)
A MULTIPLY THE COEFFICIENT WITH THE PROPER EXPONENTIALS
\((1-\times\text{LAMBDAXT}\times (+/\text{INDEX}[I];)-+/\text{INDEX}[J];)\)
\(\text{SUCCESS\times \times \times \text{LAMBDAXT}\times (+/\text{INDEX}[J];)}\)
\(Z[I,J]=\text{COEFF\times FAIL\times SUCCESS}\)
A INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE
\(J=J+1\)
\((J<\text{N})/\text{GNLOOP}\)
A RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
\(J=1\)
\(I=I+1\)

7. GNUFAL METAPHOR FUNCTION DESCRIPTION
\[ 88 \] \rightarrow (I \leq N) / GnLoop
\[ \downarrow \]
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:

V GAVINFO
[1] a  ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE DISPLAYED
[2] a  PRINT 'PLACE AN X BELOW EACH ENTRY IN THE GIVEN LIST WHICH YOU WANT TO '
[3] a  PRINT 'CHANGE. THE ABBREVIATIONS ARE AS FOLLOWS:'
[4] a  PRINT '
[5] a  PRINT P THE INTRAPHASE STATE TRANSITION
[6] a  PRINT (P) MATRICES'
[7] a  PRINT H THE INTERPHASE STATE TRANSITION
[8] a  PRINT (H) MATRICES'
[9] a  PRINT ' CONST.BAS.VARS THE NUMBER OF TIME-INVARIANT
[10] a  PRINT BASIC VARIABLES AND' THEIR ASSOCIATED
[12] a  PRINT ' ALL.ACC.LEVELS USING THE PRESENT H AND P
[13] a  PRINT MATRICES AND THE' PRESENT TIME-INVARIANT BASIC
[14] a  PRINT VARIABLE INFORMATION.' DETERMINE THE PERFORMABILITY OF
[15] a  PRINT THE SYSTEM.' METAPHOR WILL ASK FOR THE
[16] a  PRINT '
[17]
APPROPRIATE INFORMATION' PRINT ' REGARDING THE ACCOMPLISHMENT LEVELS.' PRINT '' PRINT ' PRESENT.ACC.LEVEL ALTER ONLY THE ACCOMPLISHMENT LEVEL ' PRINT ' CONSIDERATION.' PRINT '' PRINT ' I INITIAL VECTOR' PRINT '' PRINT ' G CHARACTERISTIC MATRICES' PRINT '' PRINT ' F CHARACTERISTIC VECTOR' PRINT ' V VECTOR CHARACTERIZING THE TIME+
INVARIANT BASIC VARIABLES' PRINT '' PRINT ' NUM.TRAJ.SETS ALTER THE NUMBER OF TRAJECTORY SETS DESCRIBING' PRINT ' CONSIDERATION' PRINT '' PRINT ' IF AN ITEM IS UNDEFINED WHEN AN ALTERATION IS REQUESTED, AN ERROR '
PRINT 'MESSAGE WILL BE PRINTED AND THAT ALTERATION SUPPRESSED. MORE THAN'
PRINT 'ONE ITEM MAY BE CHANGED WITH A SINGLE ALTER COMMAND.' PRINT ' EXAMPLE:' PRINT '' PRINT 'P H CONST.BAS.VARS ALL.ACC.LEVELS PRESENT.ACC.LEVEL I G V NUM.TRAJ.SETS'
PRINT 'X X X'
PRINT '' PRINT ' THIS INFORMS METAPHOR THAT THE P AND H MATRICES ARE TO BE CHANGED AND THAT THE '
PRINT 'PERFORMABILITY IS TO BE CALCULATED. IF YOU WISH TO CHANGE THE NUMBER OF PHASES '
PRINT 'OR ASSOCIATED STATES, TYPE END AND BEGIN METAPHOR AGAIN' PRINT '' PRINT ' DO YOU WANT REFERENCES?' PRINT ' LEAVE IF NOT
PRINT ' FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT '' PRINT ' J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
7. GAVINFO METAPHOR FUNCTION DESCRIPTION
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:

V GBVINFO

[1] a
[2] a ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF BASIC VARIABLES
[3] a
[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.'
[12] PRINT ' .2, 0.3 .4,0.1'
[13] PRINT 'THIS INFORMS METAPHOR THAT THE PROBABILITIES OF THE FOUR TIME+IN Variant VARIABLES ARE 0.2, 0.3, 0.4, AND 0.1 RESPECTIVELY. THE NUMBER'
[14] PRINT 'OF TIME-IN Variant VARIABLES DECLARED EARLIER MUST HAVE BEEN FOUR OR AN'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT 'ERROR MESSAGE WILL RESULT.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
PRINT 'LEAVE IF NOT ~INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH ASSOCIATES, INC., CHICAGO, 1972.'
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:

V GDINFO
[1] a
[2] a ROUTINE GIVING HELP ON DEDFAIL TYPE P MATRIX
[3] a
[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.'
[10] PRINT 'THE COMPONENTS. THE NUMBER OF STATES DECLARED FOR THE
PHASE MUST BE '  
PHASE; ENTER A'
[12] PRINT 'SINGLE POSITIVE INTEGER. NEXT YOU WILL BE PROMPTED FOR
THE FAILURE'
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] a LEAVE IF NOT
[18] ~(INYES)/0
7. GDINFO METAPHOR FUNCTION DESCRIPTION
**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:**

```plaintext
V GDVINFO
[1] a
[2] a ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE ALTERED
[3] a
[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 ''
[12] PRINT ''
[13] PRINT 'NUM.CONST.BAS.VARS THE NUMBER OF TIME-IN Variant BASIC VARIABLES'
[14] PRINT ''
[15] PRINT 'PROB.CONST.BAS.VARS THE PROBABILITIES OF EACH OF THE TIME-IN Variant BASIC VARIABLES'
[16] PRINT ''
[17] PRINT ''
[18] PRINT 'NUM.ACC.LEVELS THE NUMBER OF ACCOMPLISHMENT LEVELS'
[19] PRINT ''
[20] PRINT 'NUM.TRAJ.SETS THE NUMBER OF TRAJECTORY SETS
```

*METAPHOR (VERSION 1) PROGRAMMER'S GUIDE*
ASSOCIATED WITH
PRINT ' THE ACCOMPLISHMENT LEVEL UNDER
CONSIDERATION'
PRINT ' THE INITIAL VECTOR FOR THE
TRAJECTORY SET'
PRINT ' UNDER CONSIDERATION'
PRINT ' G THE CHARACTERISTIC MATRICES FOR
THE TRAJECTORY'
PRINT ' SET UNDER CONSIDERATION'
PRINT ' I THE INITIAL VECTOR FOR THE
TRAJECTORY'
PRINT ' UNDER CONSIDERATION'
PRINT ' F THE CHARACTERISTIC VECTOR FOR
THE TRAJECTORY'
PRINT ' SET UNDER CONSIDERATION'
PRINT ' V THE VECTOR CHARACTERIZING THE
TIME+INVARIANT 
TRAJECTORY SET '
PRINT ' UNDER CONSIDERATION'
PRINT ' PERF THE PERFORMABILITY'
PRINT ' 
PRINT 'IF AN ITEM IS UNDEFINED WHEN A DISPLAY IS REQUESTED, AN
ERROR MESSAGE'
PRINT 'WILL BE PRINTED AND THAT DISPLAY WILL BE
SUPPRESSED. MORE THAN ONE ITEM '
PRINT 'MAY BE DISPLAYED WITH A SINGLE DATA COMMAND.'
PRINT 'EXAMPLE:'
PRINT '
PRINT 'NUM.PHASES NUM.STATES P H NUM.CONST.BAS.VARS PROB.CONST.B
PRINT 'X X
PRINT 'NUM.ACC.LEVELS NUM.TRAJ.SETS I G F V PERF'
PRINT '
PRINT 'This informs metaphor that the number of phases,
states, and accomplishment'
PRINT 'levels as well as the probabilities of the time+
invariant basic variables'
PRINT 'and the performability are to be displayed.'
PRINT '
PRINT 'Do you want references?'
LEAVE IF NOT
-(~YES)/0
PRINT 'For further information on performability modeling and
analysis, see'
PRINT '
PRINT 'J. F. Meyer, "Models and techniques for evaluating

7. GDVINFO METAPHOR FUNCTION DESCRIPTION
THE 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT.'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE:
PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT.'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE:
PRINT 'S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE
PRINT RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT '}

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
GFVINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GFVINFO

PURPOSE: To generate the GETSERVER HELP information.

GLOBAL VARIABLES: None.

CALLING FUNCTIONS: COMMANDHELP.

FUNCTIONS CALLED: INYES, PRINT.

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

LISTING:

\[ \text{GFVINFO} \]

\[ \text{ROUTINE GIVING HELP ON INPUTTING THE F VECTORS} \]

\[ \text{PRINT 'ENTER THE CHARACTERISTIC (F) VECTOR FOR THE TRAJECTORY SET UNDER CONSIDERATION.'} \]

\[ \text{PRINT 'EACH ENTRY SHOULD BE EITHER 0 OR 1 AND SHOULD BE SEPARATED FROM THE OTHER ENTRIES.'} \]

\[ \text{PRINT 'BY SPACES AND/OR COMMAS. THE NUMBER OF ENTRIES SHOULD BE THE SAME AS THE NUMBER.'} \]

\[ \text{PRINT 'OF STATES OF THE FINAL PHASE MODEL. ALSO THE ORDER OF THE ENTRIES SHOULD'} \]

\[ \text{PRINT 'CORRESPOND TO THE ORDER OF THE STATES AS CONSIDERED ELSEWHERE IN THE'} \]

\[ \text{PRINT 'METAPHOR PACKAGE FOR THE FINAL PHASE.'} \]

\[ \text{PRINT 'EXAMPLE:'} \]

\[ \text{PRINT '1 \ 0 \ 1 \ ,0'} \]

\[ \text{PRINT 'THIS INFORMS METAPHOR THAT THE CHARACTERISTIC VECTOR FOR THIS TRAJECTORY'} \]

\[ \text{PRINT 'SET IS'} \]

\[ \text{PRINT '1'} \]

\[ \text{PRINT '0'} \]

\[ \text{PRINT '1'} \]

\[ \text{PRINT '0'} \]

\[ \text{PRINT 'DO YOU WANT REFERENCES?'} \]
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT 
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT 
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT 
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT 
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT 
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT 

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

V GGINFO

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

7. GGINFO METAPHOR FUNCTION DESCRIPTION
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, ''APL\360 REFERENCE MANUAL,' SCIENCE
RESEARCH'
PRINT 'ASSOCIATES,INC., CHICAGO, 1972.'
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:

V  GGMINFO
[1] a
[2] a ROUTINE GIVING HELP ON INPUTTING G MATRICES
[3] a
[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'
[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 1 0 0 0'
[16] PRINT '  0 0 0 1 0'
[17] PRINT '  0 0 0 0 1'
[18] PRINT ''
[19] PRINT 'DO YOU WANT REFERENCES?'

7. GGMINFO METAPHOR FUNCTION DESCRIPTION
~ LEAVE IF NOT (~INYES)/O
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

V GHMINFO
[1] a
[2] a ROUTINE GIVING HELP ON INPUTTING GIVEN H MATRICES
[3] a
[4] PRINT 'TYPE ONE OF: GIVEN, IDENTITY'
[5] PRINT 'DO YOU WANT MORE HELP?'
[6] aLEAVE 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.'
[12] PRINT ' IDENTITY METAPHOR WILL GENERATE AN H MATRIX
[13] PRINT 'assin THE MATRIX' ASSUMING THE MATRIX
[14] PRINT ' REPRESENTS A SYSTEM IN WHICH THERE IS NO STATE CHANGE'
[15] PRINT ' BETWEEN PHASES. THUS, METAPHOR
[16] PRINT GENERATES AN IDENTITY MATRIX.'
[17] PRINT ''
[18] PRINT ''
[19] PRINT 'DO YOU WANT REFERENCES?'
[20] a LEAVE IF NOT
[21] +(~INYES)/0
[22] PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'

PRINT ' J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE"
PRINT ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT ' NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT '
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT '
PRINT ' J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE"
PRINT ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT ' NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT '
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT '
PRINT ' S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH'
PRINT ' ASSOCIATES, INC., CHICAGO, 1972.'
PRINT '

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
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:

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

GIVINFO METAPHOR FUNCTION DESCRIPTION
\texttt{METAPHOR (VERSION 1) PROGRAMMER'S GUIDE}
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:

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

7. GNINFO METAPHOR FUNCTION DESCRIPTION
COMPONENTS.  
PRINT 'AGAIN ENTER A SINGLE POSITIVE NUMBER. IF THIS NUMBER 
IS NOT BETWEEN  
PRINT '1E-1 AND 1E-10, YOU WILL BE ASKED FOR 
CONFIRMATION. YOU WILL THEN '  
PRINT 'BE ASKED THE NUMBER OF GROUPS; ENTER THIS AS A SINGLE 
POSITIVE INTEGER.'  
PRINT 'FINALLY, METAPHOR WILL REQUEST THE NUMBER OF COMPONENTS 
IN EACH GROUP.'  
PRINT 'THIS SHOULD BE INPUT AS A ROW OF POSITIVE INTEGERS 
SEPARATED BY '  
PRINT 'SPACES OR COMMAS.'  
PRINT ''  
PRINT 'DO YOU WANT REFERENCES?'  
PRINT '  
LEAVE IF NOT  
¬((~INYES)/O  
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND 
ANALYSIS, SEE'  
PRINT ''  
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING 
THE '  
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA 
GRANT'  
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'  
PRINT ''  
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'  
PRINT ''  
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING 
THE '  
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA 
GRANT'  
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'  
PRINT ''  
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'  
PRINT ''  
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE 
RESEARCH'  
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'  
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:

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

7. GNAINFO METAPHOR FUNCTION DESCRIPTION
PRINT ' J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ' FOR FURTHER INFORMATION REGARDING APL, SEE S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH ASSOCIATES, INC., CHICAGO, 1972.'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
GNBINFO

METAPHOR FUNCTION DESCRIPTION

CALLING SEQUENCE: GNBINFO

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, GNBINFO also prints a small selection of references.

LISTING:

V GNBINFO

ROUTINE GIVING HELP ON INPUTTING BASIC VARIABLES

PRINT 'ENTER THE NUMBER OF BASIC VARIABLES WHOSE PROBABILITIES

REMAIN TIME-INVARIANT'

PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF

TIME-INVARIANT BASIC VARIABLES.)'

PRINT 'THE NUMBER SHOULD BE A SINGLE POSITIVE INTEGER.'

PRINT 'EXAMPLE:'

PRINT '2'

PRINT 'THIS INFORMS METAPHOR THAT TWO TIME-INVARIANT BASIC

VARIABLES ARE CONSIDERED IN THE'

PRINT 'MODEL.'

PRINT '

PRINT 'DO YOU WANT REFERENCES?'

PRINT 'LEAVE IF NOT

PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND

ANALYSIS, SEE'

PRINT '

PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING

THE'

PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA

GRANT'

PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'

7. GNBINFO METAPHOR FUNCTION DESCRIPTION
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:

V GNPINFO

[1] a
[2] a ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF PHASES
[3] a
[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] a LEAVE IF NOT
[11] ~(INYES)/0
[12] PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
[13] PRINT '
[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

7. GNPINFO METAPHOR FUNCTION DESCRIPTION
THE ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT'
PRINT ' NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT ' S. PAKIN, 'APL360 REFERENCE MANUAL,' SCIENCE RESEARCH'
PRINT ' ASSOCIATES, INC., CHICAGO, 1972.'
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, GNTSINFO also prints a small selection of references.

LISTING:

V GNTSINFO

[1] a
[2] a ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF TRAJECTORIES IN A GIVEN ACCOMPLISHMENT LEVEL
[3] a
[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 '
[12] a 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
[18] PRINT ' GRANT'
[19] PRINT '

7. GNTSINFO METAPHOR FUNCTION DESCRIPTION
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT'
PRINT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE RESEARCH'
PRINT ASSOCIATES, INC., CHICAGO, 1972.'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
**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:**

```
V GPMINFO
   a ROUTINE GIVING HELP ON INPUTTING P MATRICES
   a
   PRINT 'TYPE ONE OF: GIVEN, DEDFAIL, NFAIL, IDENTITY'
   PRINT 'DO YOU WANT MORE HELP?'
   LEAVE IF NO MORE HELP WANTED
   +(~INYES)/0
   PRINT 'ENTER ONE OF THE FOLLOWING TYPES FOR THE STATE
   TRANSITION (P) MATRIX:'
   PRINT ''
   PRINT '    GIVEN    YOU WILL INPUT A P MATRIX, ONE ROW AT A
   TIME.'
   PRINT ''
   PRINT '    DEDFAIL  METAPHOR WILL GENERATE A P MATRIX
   ASSUMING THE MATRIX'
   PRINT '   EACH FAILING'
   PRINT '    REPRESENTS A SYSTEM HAVING N COMPONENTS,
   DISTINGUISHABLE. THE STATE'
   PRINT '    INDEPENDENTLY AND EACH
   THE COMPONENTS.'
   PRINT '    OF THE SYSTEM IS THE STATE OF EACH OF
   PHASE AND THE FAILURE'
   PRINT '    YOU WILL BE ASKED THE LENGTH OF THE
   RATE OF THE COMPONENTS.'
```
METAPHOR WILL GENERATE A P MATRIX

ASSUMING THE MATRIX REPRESENTS A SYSTEM HAVING M GROUPS OF
K(M) COMPONENTS EACH.

THE STATE OF THE FAILED) COMPONENTS' NUMBER OF GROUPS,' THE LENGTH'
METAPHOR GENERATES THE COMPONENTS.'

IDENTITY MAT'IX.

METAPHOR WILL GENERATE A P MATRIX

REPRESENTS A SYSTEM IN WHICH THERE IS NO
FAILURE, I.E., NO CHANGES IN STATES ARE MADE. THUS,
AN IDENTITY MATRIX.'

DO YOU WANT REFERENCES?

LEAVE IF NOT

FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE

J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'

FOR FURTHER INFORMATION REGARDING METAPHOR, SEE

J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'

FOR FURTHER INFORMATION REGARDING APL, SEE'

S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE
RESEARCH'

ASSOCIATES, INC., CHICAGO, 1972.'
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:

V GSINFO
[1] a
[2] a ROUTINE FOR GIVING HELP ON INPUTTING THE NUMBER OF STATES IN EACH PHASE
[3] a
[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
[9] PRINT '  3,5'
[10] PRINT 'THIS INDICATES TO METAPHOR THAT THE FIRST PHASE HAS 4 STATES, THE SECOND
[12] PRINT 'METAPHOR CHECKS TO MAKE SURE THAT THE NUMBER OF GROUPS OF STATES
[13] PRINT 'MATCHES THE NUMBER OF PHASES INPUT EARLIER. AN ERROR MESSAGE WILL BE
[14] PRINT 'PRINTED IF THEY DO NOT MATCH.'
[15] a LEAVE IF NOT
[16] \~(-INYES)/0
[17] PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'

FOR FURTHER INFORMATION REGARDING METAPHOR, SEE


FOR FURTHER INFORMATION REGARDING APL, SEE

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:

1. GVVINFO
2. ROUTINE GIVING HELP ON INPUTTING THE TIME-IN Variant BASIC VARIABLE VECTOR
3. PRINT 'ENTER A VECTOR OF 0'S, 1'S, AND 2'S TO INDICATE WHETHER THE 'TIME-IN Variant BASIC VARIABLE''S OCCURRENCES OR NON-OCCURRENCES SHOULD BE' CONSIDERED IN THE TRAJECTORY SET. THE CODING IS AS FOLLOWS:'
4. PRINT ''
5. PRINT ' 0 THE CORRESPONDING BASIC VARIABLE''S OCCURRENCE SHOULD'
6. PRINT ' 1 BE CONSIDERED'
7. PRINT ' 2 THE CORRESPONDING BASIC VARIABLE''S NON-OCCURRENCE SHOULD BE CONSIDERED'
8. PRINT ' EITHER THE OCCURRENCE OR NON-OCCURRENCE
9. PRINT ' OF THE '
10. PRINT ' CONSIDERED'
11. PRINT ' (I.E., THE BASIC VARIABLE IS A 'DON'T'T CARE')'

7. GVVINFO METAPHOR FUNCTION DESCRIPTION
PRINT ''
PRINT 'ENTER A ROW OF 0'S, 1'S, AND 2'S, SEPARATING EACH ENTRY BY SPACES'
PRINT 'AND/OR COMMAS. THE NUMBER OF ENTRIES SHOULD BE THE SAME AS THE '
PRINT 'NUMBER OF TIME-IN Variant BASIC VARIABLES DECLARED EARLIER. ALSO THE ORDER'
PRINT 'OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER OF THE BASIC VARIABLES'
PRINT 'AS CONSIDERED ELSEWHERE IN THE METAPHOR PACKAGE.'
PRINT 'EXAMPLE:'
PRINT ' 0,1 1,2'
PRINT 'THIS INDICATES TO METAPHOR THAT FOR THIS TRAJECTORY SET, THE OCCURRENCE'
PRINT 'OF THE FIRST TIME-IN Variant BASIC VARIABLE IS IMPORTANT TO THE TRAJECTORY'
PRINT 'SET, THE NON-OCCURRENCE OF THE SECOND AND THIRD TIME-'
PRINT 'IN Variant BASIC VARIABLES IS IMPORTANT, AND THAT THE FOURTH TIME-IN Variant BASIC '
PRINT 'VARIABLE IS IRRELEVANT.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
LEAVE IF NOT (~INYES)
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
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:

V METINFO
[1] a ROUTINE GIVING HELP ON METAPHOR
[2] a
[3] a
[4] PRINT 'METAPHOR IS AN INTERACTIVE SOFTWARE PACKAGE AIDING THE
[5] PRINT 'MODELING AND ANALYSIS OF PERFORMABILITY. AT PRESENT, METAPHOR
[6] PRINT 'IS CAPABLE OF EVALUATING CERTAIN PERFORMABILITY MODELS.'
[7] PRINT 'THE COMMANDS PRESENTLY AVAILABLE ARE: EVAL, HELP, DATA,
[8] PRINT 'ALTER, CALC,'
[9] PRINT 'COM, BRIEF [ON|OFF], ECHO [ON|OFF], AND EXIT.'
[10] PRINT 'DO YOU WANT MORE HELP?'
[12] ~(~INYES)/O
[13] PRINT 'THE COMMANDS CAN BE ENTERED AT ANY TIME EXCEPT IN
[14] PRINT 'RESPONSE TO A YES/NO QUESTION. THE COMMANDS ARE AS FOLLOWS:'
[15] PRINT ''
[16] PRINT 'EVAL EVALUATE A USER SUPPLIED PERFORMABILITY MODEL'
[17] PRINT ''
[18] PRINT 'HELP GIVE MORE INFORMATION ABOUT THE QUESTION BEING ASKED'
[19] PRINT ''

7. METINFO METAPHOR FUNCTION DESCRIPTION
PRINT 'DISPLAY VARIABLE INFORMATION AND MODEL PARAMETERS'
PRINT ''
PRINT 'ALTER MODEL PARAMETERS'
PRINT ''
PRINT 'CALC 'EXIT' TO LEAVE.'
PRINT ''
PRINT 'COM ENTER COMMENTS ON THE OUTPUT'
PRINT ''
PRINT 'BRIEF [ON|OFF] TURN BRIEF OUTPUT ON OR OFF'
PRINT ''
PRINT 'ECHO [ON|OFF] TURN INPUT ECHO ON OR OFF'
PRINT ''
PRINT 'EXIT LEAVE METAPHOR'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
PRINT ''
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH'
PRINT ''
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''
8. REFERENCES


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

Matrix Generator Functions

GETHMATRICES
GETNUMBASICVARIABLES
GNBVINFO
GETBASICVARIABLES
GBVINFO
GETNUMACCLEV
GNAINFO
GETPERF
PUTPERFORMABILITY

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

\[ \text{METAPHOR} \]

1. PROGRAM FOR EVALUATION OF PERFORMABILITIES
2. SET PROGRAM CONSTANTS
3. DECLAREMETAPHOR
4. PRINT HEADING
5. PRINT ''
6. PRINT ''
7. PRINT 'MICHIGAN EVALUATION AID FOR PERFORMABILITY'
8. PRINT ''
9. PRINT ''
10. PRINT ''
11. PRINT 'TYPE HELP FOR ASSISTANCE'
12. PROGRAM LOOPS UNTIL EXIT TAKEN
13. MIN:6 INPUT MET
14. PROGRAM LOOPS UNTIL EXIT TAKEN
15. DECLAREMETAPHOR

\[ \text{DECLAREMETAPHOR} \]

1. DECLARES AND Initializes THE CONSTANTS USED IN THE 'PERF FUNCTION SERIES
2. SET THE LIST OF COMMANDS
3. HELP=HELP
4. EXIT=EXIT
5. DATA=DATA
6. ALTER=ALTER
7. CALC=CALC
8. COM=COM
9. EVAL=EVAL
10. ON=ON
11. OFF=OFF
12. COMMANDSIZE=6
13. COMMANDLIST=(9,COMMANDSIZE)'HELP','EXIT','DATA','ALTER'
14. 'CALC','ECHO','BRIEF','COM','EVAL'
15. SET LIST OF MATRIX GENERATOR TYPES
16. GIVEN=GIVEN
17. DEDFAIL=DEDFAIL

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
\begin{verbatim}
SET LIST OF ROUTINES FOR HELP CALLS
GNP+1
GS+2
GPM+3
GG+4
GN+5
GD+6
GHM+7
GBV+8
GEV+9
GNA+10
GIV+11
GCN+12
GFV+13
GTV+14
GNS+15
GAV+16
GDV+17
MET+18

SET NUMBER OF HELP ROUTINES
NUMHELPROUTINES+18

SET VARIABLE DEFINITION SWITCHES. 1 IF VARIABLE DEFINED, 0 IF NOT.
DEFNUMPHASES+0
DEFNUMSTATES+0
DEFP+0
DEFH+0
DEFNUMBASICVARIABLES+0
DEFBASICVARIABLES+0
DEFNUMACCLEV+0
DEFNUMTRAJSETS+0
DEFI+0
DEFC+0
DEFF+0
DEFT+0
DEFPERFORMABILITY+0
DEFACCELLEV+0

MAKE DEFAULT SWITCH SETTINGS
BRIEFOUTPUT+0
ECHOINPUT+0
\end{verbatim}

APPENDIX: LISTING OF METAPHOR
v METINFO

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

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS, NASA
GRANT'
PRINT ' NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ' FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ' J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE'
PRINT ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS, NASA
GRANT'
PRINT ' NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ' FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ' S. PAKIN, 'APL/360 REFERENCE MANUAL,' SCIENCE
RESEARCH'
PRINT ' ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ' COMMAND FUNCTIONS

\* COMMANDALTER; ALTERVECTOR

\n
A Routines FOR CHANGING CURRENT DATA

A GET DATA TO BE ALTERED

A ALTERVECTOR+GETALTERVECTOR

A CHANGE THAT DATA. EXIT WHEN THROUGH.

ALoop: = ALTERVECTOR/AP, AH, ABASICVARIABLES, AALACCLEVELS,
APRESENTACCLEVEL, AT, AG, AF, AV, ANUMTRAJSETS

A CHANGE THE REQUESTED DATA

A CHANGE THE P MATRICES

AP: = (DEFP=1)/APALTER

A P MATRICES ARE NOT DEFINED AT THIS TIME. PRINT ERROR
MESSAGE

PRINT ' P MATRICES ARE NOT DEFINED AT THIS TIME.'

APLoop

APALTER: = PRINT 'ALTERING P'

GETPMATRICES

APLoop: = ALTERVECTOR[1] = 0

A ALOOP

A CHANGE H MATRICES

AH: = (DEFH=1)/AHALTER

A H MATRICES ARE NOT DEFINED AT THIS TIME. PRINT ERROR
MESSAGE

APPENDIX: LISTING OF METAPHOR
PRINT 'H MATRICES ARE NOT DEFINED AT THIS TIME.'

AHALTER: PRINT 'ALTERING H'

GETHMATRICES

AHLOOP: ALTERTENSOR[2] = 0

→ ALoop

CHANGE THE TIME-INVARIANT BASIC VARIABLES

ABASICVARIABLES: = (DEFBASICVARIABLES = 1) / ABASICVARIABLESALTER

TIME-INVARIANT BASIC VARIABLES ARE NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE

PRINT 'TIME-INVARIANT BASIC VARIABLES ARE NOT DEFINED AT THIS TIME.'

→ ABASICVARIABLESLOOP

ABASICVARIABLESALTER: PRINT 'ALTERING THE NUMBER OF BASIC VARIABLES'

GETNUMBASICVARIABLES

GETBASICVARIABLES

ABASICVARIABLESLOOP: ALTERTENSOR[3] = 0

→ ALoop

CHANGE ALL ACCOMPLISHMENT LEVELS

AALLACCLEVELS: = (DEFNUMACCEL = 1) / AALLACCLEVELSALTER

THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE

PRINT 'THE ACCOMPLISHMENT LEVELS ARE NOT DEFINED AT THIS TIME.'

→ AALLACCLEVELSLOOP

AALLACCLEVELSALTER: PRINT 'ALTERING THE NUMBER OF ACCOMPLISHMENT LEVELS'

GETNUMACCEL

GETACCELPROB

AALLACCLEVELSLOOP: ALTERTENSOR[4] = 0

→ ALoop

CHANGE JUST THE PRESENT (LAST) ACCOMPLISHMENT LEVEL

APRESENTACCEL: = (DEFACCEL = 1) / APRESENTACCELALTER

AN ACCOMPLISHMENT LEVEL IS NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE

PRINT 'AN ACCOMPLISHMENT LEVEL IS NOT DEFINED AT THIS TIME.'

→ APRESENTACCELLOOP

APRESENTACCELALTER: PRINT 'ALTERING THE PRESENT ACCOMPLISHMENT LEVEL'

PERFORMABILITY[1] + GETACCELPROB L

APRESENTACCELLOOP: ALTERTENSOR[5] = 0

→ ALoop

CHANGE THE I VECTOR

AI: = (DEFI = 1) / AAlter

I VECTOR IS NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE

PRINT 'I VECTOR IS NOT DEFINED AT THIS TIME.'

→ AILOOP
AIALTER:PRINT 'ALTERING THE I VECTOR'
I GETIVECTOR
ALOOP:ALTERVECTOR[6]←0
→ALOOP

a

CHANGE THE G MATRICES
AG:=(DEFG=1)/AGALTER
G MATRICES ARE NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE
PRINT 'G MATRICES ARE NOT DEFINED AT THIS TIME.'
AGLOOP
AGALTER:PRINT 'ALTERING THE G MATRICES'
G GETGMATRICES
AGLOOP:ALTERVECTOR[7]←0
→ALOOP

CHANGE THE F VECTOR
AF:=(DEFF=1)/AFALTER
F VECTOR IS NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE
PRINT 'F VECTOR IS NOT DEFINED AT THIS TIME.'
AFLOOP
AFALTER:PRINT 'ALTERING THE F VECTOR'
F GETFVECTOR
AFLOOP:ALTERVECTOR[8]←0
→ALOOP

CHANGE THE TIME-INARIANT BASIC VARIABLE VECTOR
AV:=(DEFV=1)/AVALTER
THE TIME-INARIANT BASIC VARIABLE VECTOR IS NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE
PRINT 'THE TIME-INARIANT BASIC VARIABLE VECTOR IS NOT DEFINED AT THIS TIME.'
AVLOOP
AVALTER:PRINT 'ALTERING THE V VECTOR'
V GETVVALUES
AVLOOP:ALTERVECTOR[9]←0
→ALOOP

CHANGE THE NUMBER OF TRAJECTORY SETS BEING CONSIDERED
NUMTRAJSETS:=(DEFNUMTRAJSETS=1)/NUMTRAJSETS ALTER
THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT THIS TIME. PRINT ERROR MESSAGE
PRINT 'THE NUMBER OF TRAJECTORY SETS IS NOT DEFINED AT THIS TIME.'
NUMTRAJSETS ALTER:PRINT 'ALTERING THE NUMBER OF TRAJECTORY SETS'
NUMTRAJSETS GETNUMTRAJSETS
ALTERVECTOR[10]←0
→ALOOP

APPENDIX: LISTING OF METAPHOR
\textbf{v \textit{COMMAND\textsc{brief switch}}}

1. ROUTINE FOR TUR\textsc{n}ING THE BRIEF SWITCH ON AND OFF. 'ON' CAUSES
   2. METAPHOR TO USE A TERSE FORM OF OUTPUT. 'OFF' CAUSE THE
   3. NORMAL FULL OUTPUT.

4. TURN SWITCH ON IF REQUESTED, ELSE TURN SWITCH OFF.
5. BRIEFOUTPUT=~/'ON'\textsc{ switch}
6. PRINT CONFIRMATION REGARDLESS OF BRIEF SWITCH
7. 'BRIEF ';SWITCH

\textbf{v \textit{COMMAND\textsc{calc;ccalcinp}}}

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

\textbf{v \textit{COMMAND\textsc{com;ccin}}}.

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

\textbf{v \textit{COMMAND\textsc{data;datavector}}}

1. ROUTINE FOR DISPLAYING CURRENT DATA
2. GET DATA TO BE DISPLAYED
3. DATAVECTOR=GETDATAVECTOR
4. DISPLAY THAT DATA. EXIT WHEN THROUGH.
5. DLOOP:=DATAVECTOR/DNUMPHASES,DNUMSTATES,DP,DP,DNUMBASICVAR

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
DBASICVARIABLES,DNUMACCLEV,DNUMTRAJSETS,DI,DC,DF,DV,DPERF

SHOW THE REQUESTED INFORMATION

DNUMPHASES: (DEFNUMPHASES=1)/DNUMPHASESOUT
PRINT 'NUMBER OF PHASES HAS NOT BEEN DEFINED'
+DNUMPHASESLOOP
DNUMPHASESOUT: ((PRINTQUAD 'NUMBER OF PHASES IS ');
PRINTQUAD NUMPHASES)

PRINT '
DNUMPHASESLOOP: DVECTOR[1] = 0
DLOOP

DNUMSTATES: (DEFNUMSTATES=1)/DNUMSTATESOUT
PRINT 'NUMBER OF STATES HAS NOT BEEN DEFINED'
+DNUMSTATESLOOP
DNUMSTATESOUT: ((PRINTQUAD 'NUMBER OF STATES PER PHASE IS ');
PRINTQUAD STATESPERPHASE)

PRINT '
DNUMSTATESLOOP: DVECTOR[2] = 0
DLOOP

DP: (DEFP=1)/DPOUT
PRINT 'P MATRICES HAVE NOT BEEN DEFINED'
+DPLOOP
DPOUT: PRINT 'THE P MATRICES ARE:

P
DPLOOP: DVECTOR[3] = 0
DLOOP

DH: (DEFH=1)/DHOUT
PRINT 'H MATRICES HAVE NOT BEEN DEFINED'
+DHLOOP
DHOUT: PRINT 'THE H MATRICES ARE:

H
DHLOOP: DVECTOR[4] = 0
DLOOP

DBASICVARIABLES: (DEFBASICVARIABLES=1)/DBASICVARIABLESOUT
PRINT 'THE NUMBER OF BASIC VARIABLES HAS NOT BEEN DEFINED'
+DBASICVARIABLESLOOP
DBASICVARIABLESOUT: ((PRINTQUAD 'THE NUMBER OF TIME-IN Variant BASIC VARIABLES IS ');
PRINTQUAD NUMBASICVARIABLES)

PRINT '
DBASICVARIABLESLOOP: DVECTOR[5] = 0
DLOOP

DBASICVARIABLES: (DEFBASICVARIABLES=1)/DBASICVARIABLESOUT
PRINT 'THE TIME-IN Variant BASIC VARIABLES HAVE NOT BEEN

APPENDIX: LISTING OF METAPHOR
DEFINED'

DBASICVARIABLESLOOP

DBASICVARIABLESOUT:PRINT 'THE TIME-INVARIANT BASE VARIABLES HAVE PROBABILITIES:

BASICVARIABLES

DBASICVARIABLESLOOP:DATAVECTOR[6] ← 0

DBASICVARIABLESLOOP:

DNUMACCELV:← (DEFFNUMACCELV=1)/DNUMACCELVOUT

PRINT 'THE NUMBER OF ACCOMPLISHMENT LEVELS NOT DEFINED'

DNUMACCELVLOOP:

DNUMACCELVOUT:3-((PRINTQUAD 'THE NUMBER OF ACCOMPLISHMENT LEVELS IS ');PRINTQUAD NUMACCELV)

PRINT ''

DNUMACCELVLOOP:DATAVECTOR[7] ← 0

DNUMACCELVLOOP:

DNUMTRAJSETS:← (DEFFNUMTRAJSETS=1)/DNUMTRAJSETSOUT

PRINT 'THE NUMBER OF TRAJECTORY SETS NOT DEFINED'

DNUMTRAJSETSLOOP:

DNUMTRAJSETSOUT:PRINT 'THE NUMBER OF TRAJECTORY SETS IS:

NUMTRAJSETS

DNUMTRAJSETSLOOP:DATAVECTOR[8] ← 0

DNUMTRAJSETSLOOP:

DI:← (DEFFI=1)/DIOUT

PRINT 'I VECTOR NOT DEFINED'

DILOOP:

DIOUT:3-((PRINTQUAD 'THE INITIAL VECTOR IS ');PRINTQUAD I)

PRINT ''

DILOOP:DATAVECTOR[9] ← 0

DILOOP:

DG:← (DEFG=1)/DGOUT

PRINT 'G MATRICES NOT DEFINED'

DGLOOP:

DGOUT:PRINT 'THE G MATRICES ARE:

DGLOOP:DATAVECTOR[10] ← 0

DGLOOP:

DF:← (DEFF=1)/DFOUT

PRINT 'F VECTOR NOT DEFINED'

DFLOOP:

DFOUT:3-((PRINTQUAD 'THE FINAL VECTOR IS ');PRINTQUAD F)

PRINT ''


DFLOOP:

DV:← (DEFV=1)/DVOUT

PRINT 'THE TIME-INVARIANT BASIC VARIABLE VECTOR NOT DEFINED'

DVLOOP

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
/commandecho switch

routine for turning the echo switch on and off. 'on' causes metaphor to repeat every input line. 'off' suppresses the repetition.

turn switch on if requested, else turn switch off.

/echo ;switch

/commandeval

performability computation portion of metaphor

fetch the matrices required for performability calculations

getnumphases

getstates

getpmatrices

getmmatrices

getnumbasicvariables

getbasicvariables

getnumacclev

for each accomplishment level, fetch the trajectory sets and calculate their probabilities

getperformability

print the resulting performability

printperformability

appendix: listing of metaphor
V COMMANDHELP ROUTINE

[1] ROUTINE PRINTING HELP INFORMATION
[2] GET THE PROPER INFO ROUTINE
[3] (ROUTINE=\NUMHELPROUTINES)/HGMP,HGS,HGPM,HGG,
HG,HGD,HGM,HGNBV,HGBV,HGNA,HGIV,
HGCM,HGFM,HGVV,HGTS,HGAV,HGAV,HEMT
[4] 'GET NUMBER OF PHASES' HELP
[6] 'GET STATES' HELP
[7] HGS:GSINFO
[8] 'GENERATE P MATRIX' HELP
[9] HGPM:GPMINFO
[10] 'GET GIVEN MATRIX' HELP
[12] 'GET NFAIL MATRIX' HELP
[14] 'GET DEDFAIL MATRIX' HELP
[16] 'GENERATE H MATRIX' HELP
[17] HGHM:GHMINFO
[18] 'GET NUMBER OF TIME INVARIANT BASIC VARIABLES' HELP
[19] HGNNBV:GNBVINFO
[20] 'GET TIME INVARIANT BASIC VARIABLES' HELP
[21] HGBV:GBVINFO
[22] 'GET NUMBER OF ACCOMPLISHMENT LEVELS' HELP
[23] HGNA:GNAINFO
[24] 'GET I VECTOR' HELP
[25] HGIV:GIVINFO

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
A 'GET G MATRIX' HELP
HGGM:GGMINFO

A 'GET F VECTOR' HELP
HGFV:GFVINFO

A 'GET V VECTOR' HELP
HGVV:GVVINFO

A 'GET NUMBER OF TRAJECTORY SETS' HELP
HGNTS:GNTSINFO

A 'GET ALTER VECTOR VECTOR' HELP
HGAIV:GAVINFO

A 'GET DATA VECTOR VECTOR' HELP
HGDV:GDVINFO

A 'METAPHOR' HELP
HMET:METINFO

COMMAND SUPPORT FUNCTIONS

Z=BRIEF SWITCH

ROUTINE TO INPUT THE BRIEF COMMAND
RETURN THE VALUE
Z='BRIEF ',SWITCH

Z=ECHO SWITCH

ROUTINE TO INPUT THE ECHO COMMAND
RETURN THE VALUE
Z='ECHO ',SWITCH

APPENDIX: LISTING OF METAPHOR
\[ Z + \text{GETALTERVECTOR; GAVINPUT} \]

\[ \text{ROUTINE FOR GETTING THE ALTER VECTOR FOR CHANGING DATA} \]

\[ \text{PRINT EXPLANATION} \]

\[ \text{GAVIN:PRINT 'PUT AN X BELOW EACH ITEM TO BE CHANGED. HELP AVAILABLE.'} \]

\[ \text{GET ALTER REQUESTS, RESHAPING ALONG THE WAY} \]

\[ \text{PRINT 'P H CONST.BAS.VARS ALL.ACC.LEVELS PRESENT.ACC.LEVEL I G F V NUM.TRAJ.SETS'} \]

\[ \text{GAVINPUT+80pM(80p' )} \]

\[ \text{LOOK FOR HELP REQUEST. IF PRESENT, CALL FOR HELP ROUTINE.} \]

\[ \text{COMMANDHELP GAV} \]

\[ \text{GAVINFO} \]

\[ \text{ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE DISPLAYED} \]

\[ \text{PRINT 'PLACE AN X BELOW EACH ENTRY IN THE GIVEN LIST WHICH YOU WANT TO'} \]

\[ \text{PRINT 'CHANGE. THE ABBREVIATIONS ARE AS FOLLOWS:'} \]

\[ \text{PRINT ''} \]

\[ \text{PRINT '(P) MATRICES'} \]

\[ \text{PRINT ''} \]

\[ \text{PRINT '(H) MATRICES'} \]

\[ \text{PRINT ''} \]

\[ \text{PRINT 'CONST.BAS.VARS THE NUMBER OF TIME-INVARIANT BASIC VARIABLES AND'} \]

\[ \text{PRINT 'PROBABILITIES'} \]

\[ \text{PRINT ''} \]

\[ \text{PRINT 'ALL.ACC.LEVELS USING THE PRESENT H AND P MATRICES AND THE'} \]

\[ \text{PRINT 'PRESENT TIME-INVARIANT BASIC} \]

\[ \text{METAPHOR (VERSION 1) PROGRAMMER'S GUIDE} \]
VARIABLE INFORMATION,' PRINT 'THE SYSTEM,' PRINT 'APPROPRIATE INFORMATION' PRINT 'LEVELS.' PRINT 'PRESENT.ACC.LEVEL LEVEL:' PRINT 'CONSIDERATION.' PRINT '' PRINT 'INITIAL VECTOR' PRINT 'CHARACTERISTIC MATRICES' PRINT 'CHARACTERISTIC VECTOR' (AT PRESENT, THIS ALTER OPERATION IS NOT EXECUTABLE.)' PRINT '' PRINT 'VECTOR CHARACTERIZING THE TIME-
VECTOR DESCRIBING THE
ALTER THE NUMBER OF TRAJECTORY
THE ACCOMPLISHMENT LEVEL UNDER
CONSIDERATION'
PRINT '' PRINT 'IF AN ITEM IS UNDEFINED WHEN AN ALTERATION IS REQUESTED, AN ERROR'
PRINT 'MESSAGE WILL BE PRINTED AND THAT ALTERATION SUPPRESSED. MORE THAN'
PRINT 'ONE ITEM MAY BE CHANGED WITH A SINGLE ALTER COMMAND.'
PRINT 'EXAMPLE:'
PRINT ''
PRINT 'P H CONST.BAS.VARS ALL.ACC.LEVELS PRESENT.ACC.LEVEL I G
V NUM.TRAJ.SETS'
PRINT 'X X X X'
PRINT ''
PRINT 'THIS INFORMS METAPHOR THAT THE P AND H MATRICES ARE TO BE CHANGED AND THAT THE'
PRINT 'PERFORMABILITY IS TO BE CALCULATED. IF YOU WISH TO CHANGE THE NUMBER OF PHASES'
PRINT 'OR ASSOCIATED STATES, TYPE END AND BEGIN METAPHOR AGAIN'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
PRINT 'LEAVE IF NOT

APPENDIX: LISTING OF METAPHOR
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

V Z+GETDATAVECTOR;GDVINPUT

1 a ROUTINE FOR GETTING THE DATA VECTOR FOR DISPLAYING DATA
2 a PRINT EXPLANATION
3 a PRINT EXPLANATION
4 a 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
8 GDVINPUT=69p(M,(69p' '))
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=116p(GDVINPUT,«47p' '))
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=13p0
22 DETERMINE DISPLAY VECTOR

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
ROUTINE GIVING HELP ON CHOOSING THE DATA TO BE ALTERED

PRINT 'ENTER AN X BELOW EACH ITEM IN THE GIVEN LIST WHICH YOU WANT TO'

PRINT 'DISPLAY. THE ABBREVIATIONS ARE AS FOLLOWS:'

PRINT ''

PRINT 'NUM.PHASES' THE NUMBER OF PHASES'

PRINT ''

PRINT 'NUM.STATES' THE NUMBER OF STATES'

PRINT ''

PRINT 'P' THE INTRAPHASE TRANSITION (P)

PRINT ''

PRINT 'NUM.CONST.BAS.VARS BASIC VARIABLES' THE NUMBER OF TIME-ININVARIANT

PRINT ''

PRINT 'PROB.CONST.BAS.VARS THE PROBABILITIES OF EACH OF'

PRINT 'THE TIME-ININVARIANT'

PRINT ''

PRINT 'NUM.ACC.LEVELS 'THE NUMBER OF ACCOMPLISHMENT'

PRINT ''

PRINT 'NUM.TRAJ.SETS 'THE NUMBER OF TRAJECTORY SETS ASSOCIATED WITH'

PRINT 'THE ACCOMPLISHMENT LEVEL UNDER'

PRINT 'CONSIDERATION'

PRINT ''

PRINT 'I THE INITIAL VECTOR FOR THE'

PRINT UNDER CONSIDERATION'

PRINT ''

PRINT 'G THE CHARACTERISTIC MATRICES FOR'

APPENDIX: LISTING OF METAPHOR
PRINT "SET UNDER CONSIDERATION''
PRINT "F THE CHARACTERISTIC VECTOR FOR
THE TRAJECTORY''
PRINT "SET UNDER CONSIDERATION''
PRINT "V THE VECTOR CHARACTERIZING THE
TIME-INVARIANT''
PRINT "BASIC VARIABLES FOR THE
TRAJECTORY SET''
PRINT "UNDER CONSIDERATION''
PRINT "PERF THE PERFORMABILITY''
PRINT "IF AN ITEM IS UNDEFINED WHEN A DISPLAY IS REQUESTED, AN
ERROR MESSAGE''
PRINT "WILL BE PRINTED AND THAT DISPLAY WILL BE
SUPPRESSED. MORE THAN ONE ITEM''
PRINT "MAY BE DISPLAYED WITH A SINGLE DATA COMMAND.''
PRINT "EXAMPLE:''
PRINT ''
PRINT 'NUM.PHASES NUM.STATES P H NUM.CONST.BAS.VARS PROB.CONST.B'
PRINT 'X' X
PRINT 'X' X
PRINT 'NUM.ACC.LEVELS NUM.TRAJ.SETS I G F V PERF'
PRINT 'X' X
PRINT ''
PRINT 'THIS INFORMS METAPHOR THAT THE NUMBER OF PHASES,
STATES, AND ACCOMPLISHMENT'
PRINT 'LEVELS AS WELL AS THE PROBABILITIES OF THE TIME-
INVARIANT BASIC VARIABLES'
PRINT 'AND THE PERFORMABILITY ARE TO BE DISPLAYED.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
READ IF NOT
~(~INYES)/O
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
THE'"
PRINT '" EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT'
PRINT '" NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING
THE'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
COMMAND EVAL IMPLEMENTATION FUNCTIONS

GETNUMPHASES

ROUTINE FOR FETCHING THE NUMBER OF MISSION PHASES

INPUT NUMBER OF PHASES

PRINT ' ' GNPIN:PRINT 'NUMBER OF PHASES?'

1 INPUT GNP

CHECK FOR COMMAND

→(1=\e COMMAND')/GNPIN

CHECK VALIDITY OF INPUT

→(CHECKPOS IN)/GNPIN

ELSE SET THE NUMBER OF PHASES AND EXIT

NUMPHASES+IN

DEENUMPHASES+1

GNPINFO

ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF PHASES

PRINT 'ENTER THE NUMBER OF PHASES IN THE FINITE PHASE MODEL AS A SINGLE'

PRINT 'POSITIVE INTEGER.'

PRINT 'EXAMPLE:'

PRINT ' 3'

PRINT 'THIS INDICATES TO METAPHOR THAT THE MODEL TO BE EVALUATED HAS 3 PHASES'

PRINT 'DO YOU WANT REFERENCES?'

LEAVE IF NOT

→(~INYES)/0

PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'

PRINT '

PRINT ' J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING

APPENDIX: LISTING OF METAPHOR
THE 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT 'S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT 'V GETSTATES

a ROUTINE FOR FETCHING THE NUMBER OF STATES IN EACH PHASE
a INPUT NUMBER OF STATES AND CHECK VALIDITY
PRINT 'GSIN:PRINT 'NUMBER OF STATES PER PHASE? (SPACE BETWEEN EACH NUMBER)'
NUMPHASES INPUT GS
a CHECK FOR COMMAND
+(IN=-1)/GSIN
a CHECK FOR POSITIVE INTEGER
+(CHECKPOSI IN)/GSIN
a SET STATES AND EXIT
STATESPERPHASE IN
MAXNUMSTATES/STATESPERPHASE
DEFNUMSTATES-1

a ROUTINE FOR GIVING HELP ON INPUTTING THE NUMBER OF STATES IN EACH PHASE
PRINT 'ENTER THE NUMBER OF STATES FOR EACH PHASE IN THE FINITE PHASE MODEL.'
PRINT 'TYPE A POSITIVE INTEGER FOR EACH PHASE, SEPARATING EACH WITH SPACES AND/OR COMMAS.'
PRINT 'THE NUMBER OF STATES MUST BE A POSITIVE INTEGER.'
PRINT 'EXAMPLE:'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT '4 3,5'
PRINT 'THIS INDICATES TO METAPHOR THAT THE FIRST PHASE HAS 4
STATES, THE SECOND'
PRINT 'PHASE HAS 3 STATES, AND THE THIRD PHASE HAS 5
STATES.'
PRINT 'METAPHOR CHECKS TO MAKE SURE THAT THE NUMBER OF GROUPS
OF STATES'
PRINT 'MATCHES THE NUMBER OF PHASES INPUT EARLIER. AN ERROR
MESSAGE WILL BE'
PRINT 'PRINTED IF THEY DO NOT MATCH.'
PRINT 'DO YOU WANT REFERENCES?'
\$ LEAVE IF NOT
\S \S \S \S \S \S \S \S \S
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, 'MODEL AND TECHNIQUES FOR EVALUATING
THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, 'MODEL AND TECHNIQUES FOR EVALUATING
THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA GRANT
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE RESEARCH'
PRINT '''
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT '''

GETPMATRICES;PHASE;NEXTP

ROUTINE FOR INPUTTING THE P MATRICES
ONE MATRIX FOR EACH PHASE

INITIALIZE THE ARRAY OF P MATRICES
P=(NUMPHASES,MAXNUMSTATES,MAXNUMSTATES)p0

INPUT AND CHECK THE P MATRICES
PRINT ''
PRINT ''
PRINT 'SPECIFY THE P MATRICES FOR EACH PHASE, 1 PHASE AT A
INITIALIZE PHASE COUNTER

PHASE+1

GPMPHASEIN:PRINT ''

PRINT ''

GET P MATRIX FOR PHASE

NEXTP=GENERATEPMATRIX STATESPERPHASE[PHASE]

INSERT THE MATRIX INTO THE ARRAY OF MATRICES

P[PHASE;STATESPERPHASE[PHASE];STATESPERPHASE[PHASE]]=NEXTP

INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE

PHASE=PHASE+1

=(PHASE=NUMPHASES)/GPMPHASEIN.

ELSE SET P DEFINITION FLAG AND LEAVE

DEFP+1

GETHMATRICES;PHASE;NEXTH

ROUTINE FOR INPUTTING THE H MATRICES

ONE MATRIX FOR EACH PHASE

IF NO H MATRICES, SET H+1 AND LEAVE

=(NUMPHASES>2)/GHMMULTIPHASE

H+1

INITIALIZE THE ARRAY OF H MATRICES

GHMMULTIPHASE:H+((NUMPHASES-1),MAXNUMSTATES,MAXNUMSTATES)0

INPUT AND CHECK THE H MATRICES

PRINT ''

PRINT ''

PRINT 'SPECIFY THE H MATRICES FOR EACH PHASE, 1 PHASE AT A TIME'

INITIALIZE PHASE COUNTER

PHASE+2

GPMPHASEIN:PRINT ''

PRINT ''

GET H MATRIX FOR PHASE

NEXTH=STATESPERPHASE[PHASE-1]GENERATEHMATRIX STATESPERPHASE[PHASE]

INSERT THE MATRIX INTO THE ARRAY OF MATRICES

H[PHASE-1;STATESPERPHASE[PHASE-1];
\text{STATESPERPHASE}[\text{PHASE}] \leftarrow \text{NEXTH}

\text{INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE}
\text{PHASE} \leftarrow \text{PHASE} + 1

\text{\textasciitilde}(\text{PHASE} = \text{NUMPHASES})/\text{GCMPHASEIN}

\text{ELSE SET H DEFINITION FLAG AND LEAVE}
\text{DEFH} \leftarrow 1

\text{\textasciitilde} \text{GETNUMBASICVARIABLES}

\text{ROUTINE FOR FETCHING THE NUMBER OF TIME-IN Variant) BASIC VARIABLES}

\text{INPUT NUMBER OF TIME-IN Variant BASIC VARIABLES}
\text{PRINT 'NUMBER OF TIME-IN Variant BASIC VARIABLES?'}
\text{INPUT GNBV}

\text{CHECK FOR COMMAND}
\text{\textasciitilde}(1 = ^')/\text{GNBVIN}

\text{CHECK VALIDITY OF INPUT}
\text{\textasciitilde}(\text{IN} = 0)/\text{GNBVSET}

\text{ELSE SET THE NUMBER OF TIME-IN Variant BASIC VARIABLES AND EXIT}
\text{\textasciitilde} \text{GNBVSET} = \text{NUMBASICVARIABLES} + \text{IN}

\text{\textasciitilde} \text{DEFNUMBASICVARIABLES} + 1

\text{\textasciitilde} \text{GNBVINFO}

\text{ROUTINE GIVING HELP ON INPUTTING BASIC VARIABLES}

\text{PRINT 'ENTER THE NUMBER OF BASIC VARIABLES WHOSE PROBABILITIES REMAIN TIME-IN Variant.'}
\text{PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF TIME-IN Variant BASIC VARIABLES.)'}
\text{PRINT 'THE NUMBER SHOULD BE A SINGLE POSITIVE INTEGER.'}
\text{PRINT 'EXAMPLE: '}
\text{PRINT '2'}

\text{PRINT 'THIS INFORMS METAPHOR THAT TWO TIME-IN Variant BASIC VARIABLES ARE CONSIDERED IN THE'}
\text{PRINT 'MODEL.'}
\text{PRINT '}'

\text{PRINT 'DO YOU WANT REFERENCES?'}
\text{LEAVE IF NOT}
\text{\textasciitilde}(\text{\textasciitilde} \text{INYES})/0

\text{PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND}

\text{APPENDIX: LISTING OF METAPHOR}
ANALYSIS, SEE'

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 ''

V GET BASIC VARIABLES

a ROUTINE FOR FETCHING THE PROBABILITIES OF EACH OF THE
TIME-INVARIANT BASIC VARIABLES

a SEE IF IT IS NECESSARY TO INPUT BASIC VARIABLES

(NUMBASICVARIABLES=0)/0

INPUT PROBABILITIES AND CHECK VALIDITY

GBVIN:PRINT 'PROBABILITIES OF EACH TIME-INVARIANT BASIC
VARIABLE? (SPACE BETWEEN EACH NUMBER)'

NUMBASICVARIABLES INPUT GBV

CHECK FOR COMMAND

(1=A 'COMMAND')/GBVIN

CHECK FOR PROPER PROBABILITY MAGNITUDE

((CHECKPROB IN)/GBVIN

SET BASIC VARIABLE PROBABILITIES AND EXIT

BASICVARIABLES=IN,1-IN

DEF BASICVARIABLES+1

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF BASIC VARIABLES

PRINT 'ENTER THE PROBABILITIES OF THE BASIC VARIABLES WHOSE PROBABILITIES REMAIN CONSTANT'
PRINT 'THROUGHOUT THE MISSION INTERVAL (I.E., THE NUMBER OF TIME-IN Variant BASIC VARIABLES.)'
PRINT 'THE PROBABILITIES SHOULD BE ENTERED AS A ROW OF POSITIVE NUMBERS'
PRINT 'BETWEEN ZERO AND ONE, INCLUSIVE. THE NUMBERS SHOULD BE'
PRINT 'SEPARATED BY SPACES AND/OR COMMAS. THE ORDER OF THE NUMBERS SHOULD'
PRINT 'CORRESPOND TO THE ORDER OF THE TIME-IN Variant BASIC VARIABLE VECTORS'
PRINT 'WHICH WILL BE ASKED FOR LATER.'
PRINT 'EXAMPLE:'
PRINT '.2, 0.3, 0.4, 0.1'
PRINT 'THIS INFORMS METAPHOR THAT THE PROBABILITIES OF THE
FOUR TIME-IN Variant'
PRINT 'BASIC VARIABLES ARE 0.2, 0.3, 0.4, AND 0.1 RESPECTIVELY. THE NUMBER'
PRINT 'OF TIME-IN Variant VARIABLES DECLARED EARLIER MUST HAVE BEEN FOUR OR AN'
PRINT 'ERROR MESSAGE WILL RESULT.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
LEAVE IF NOT
(\~INYES)/0
FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH'

APPENDIX: LISTING OF METAPHOR
Routine for Fetching the Number of Mission Accomplishment Levels

```
; GETNUMACCLEV

; ROUTINE FOR FETCHING THE NUMBER OF MISSION ACCOMPLISHMENT LEVELS

; INPUT NUMBER OF ACCOMPLISHMENT LEVELS

; PRINT 'NUMBER OF ACCOMPLISHMENT LEVELS?

; INPUT GNA

; CHECK FOR COMMAND

; CHECK VALIDITY OF INPUT

; ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT

; DEFNUMACCLEV=IN

```

Routine Giving Help on Inputting the Number of Accomplishment Levels

```
; GNAINFO

; ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF ACCOMPLISHMENT LEVELS

; PRINT 'ENTER THE NUMBER OF ACCOMPLISHMENT LEVELS FOR THIS MODEL AS A SINGLE POSITIVE INTEGER.'

; PRINT 'EXAMPLE:

; PRINT '5'

; PRINT 'THIS INDICATES TO METAPHOR THAT THE MODEL IT IS EVALUATING HAS 5 ACCOMPLISHMENT LEVELS.'

; PRINT ''

; PRINT 'DO YOU WANT REFERENCES?'

; LEAVE IF NOT

; +(~INYES)/0

; PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'

; PRINT '

; PRINT 'J. P. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'

; PRINT ''

; PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'

; PRINT ''
```

Metaphor (Version 1) Programmer's Guide


GETPERFORMABILITY; LEVELPROB; L

ROUTE FOR COMPUTING THE PERFORMABILITY OF THE SYSTEM

INITIALIZE THE PERFORMABILITY VECTOR AND SET DEFINITION FLAGS

PERFORMABILITY = NUMACCLEV / 0
DEFACCELEVEL + 1
DEFPERFORMABILITY + 1

LOOP THROUGH EACH ACCOMPLISHMENT LEVEL

L + 1

GET THE PROBABILITY OF EACH ACCOMPLISHMENT LEVEL

GPLOOP: LEVELPROB = GETACCLEVPROB L

INSERT THE PROBABILITY INTO THE PERFORMABILITY VECTOR

PERFORMABILITY[L] = LEVELPROB

INCREMENT THE LEVEL COUNTER AND BRANCH IF NECESSARY

L = L + 1

+(L = NUMACCLEV) / GPLOOP

ELSE EXIT ROUTINE, SETTING DEFINITION FLAG

DEFACCELEVEL = 0

APPENDIX: LISTING OF METAPHOR
MATRIX GENERATOR FUNCTIONS:

\[ Z \rightarrow M \text{ GENERATE H MATRIX } N; \text{TYPE;IN} \]

[1] a
[2] a
[4] a THE MATRIX IS TO HAVE SHAPE M\times N
[5] a
[6] a GET TYPE OF MATRIX
[7] PRINT ''
[8] GENHMIN:PRINT 'WHAT TYPE OF H MATRIX?'
[9] a INPUT GHM
[10] a CHECK FOR COMMAND
[11] (1=^A \epsilon 'COMMAND')/GENHMIN
[12] a CHECK FOR TYPE
[13] (\wedge/\text{MATRIXLIST} \epsilon (IN))/\text{GENHM GIVEN,GENHМИДENTITY}
[14] a ELSE ILLEGAL TYPE
[15] PRINT 'ILLEGAL H MATRIX TYPE. TYPE HELP FOR INFORMATION'
[16] a GENHMIN
[17] a
[18] a USER WILL GIVE H MATRIX VALUES
[19] GENHM GIVEN:Z-M \_GGIVEN N
[20] a
[21] a
[22] a
[23] a
[24] a
[25] a FIRST CHECK TO MAKE SURE AN IDENTITY MATRIX IS APPROPRIATE HERE
[26] GENHМИДENTITY:+(M=N)/GENHМИДENTITY GET
[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] a GENHMIN
[30] GENHМИДENTITY GET:Z-M \_IDENTITY N
[31] a

\[ \text{GENHINFO} \]

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

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
Print 'GIVEN YOU WILL INPUT AN $H$ MATRIX, ONE ROW AT A TIME.'
Print 'IDENTITY METAPHOR WILL GENERATE AN $H$ MATRIX assuming the matrix represents a system in which there is no state change. Thus, METAPHOR generates an identity matrix.'
Print 'Do you want references?'
Print 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
Print 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
Print 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
Print 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
Print 'FOR FURTHER INFORMATION REGARDING APL, SEE'
Print 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE RESEARCH ASSOCIATES, INC., CHICAGO, 1972.'

Z+GENERATEPMATRIX N;TYPE;IN

Routine for supervising the construction of the $P$ matrix for the given phase.

The matrix is to have shape $N \times N$.

Get type of matrix.

Print 'What type of $P$ matrix?

Input $GPM$.

Appendix: Listing of Metaphor
216

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

¥ GPINFO

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

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT 'INDEPENDENTLY AND EACH OF THE SYSTEM IS THE STATE OF EACH OF THE COMPONENTS.'
PRINT 'YOU WILL BE ASKED THE LENGTH OF THE PHASE AND THE FAILURE RATE OF THE COMPONENTS.'
PRINT ''
PRINT 'IDENTITY METAPHOR WILL GENERATE A P MATRIX ASSUMING THE MATRIX REPRESENTS A SYSTEM IN WHICH THERE IS NO FAILURE, I.E., NO CHANGES IN STATES ARE MADE. THUS, METAPHOR GENERATES AN IDENTITY MATRIX.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
LEAVE IF NOT (-INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE:
J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE:
J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA GRANT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'

APPENDIX: LISTING OF METAPHOR
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT ' S. PAKIN, "APL/360 REFERENCE MANUAL," SCIENCE
RESEARCH'
PRINT ' ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

\[ Z \in (G) \text{DEDFAIL N; T; LAMBDA; INDEX; I; J; SIZE; SUCCESS; FAIL} \]

1. ROUTINE FOR GENERATING THE DEDFAIL TYPE P MATRIX
2. THE MATRIX IS TO HAVE SHAPE N x N
3. CHECK TO MAKE SURE THIS ROUTINE IS APPROPRIATE
4. -((2^N) = [(2^N)]) / DGIN
5. PRINT EXPLANATION, NOTE FAILURE, AND EXIT
6. PRINT 'THE NUMBER OF STATES IN DEDFAIL MUST BE A POWER OF TWO.'
7. -((PRINTQUAD 'THIS PHASE HAS '); (PRINTQUAD N); PRINTQUAD ' STATES.')
8. PRINT ''
9. Z = (N, N)^{-1}
10. T = -((N, N)^{-1})
11. GET PHASE LENGTH
12. GDIN: PRINT 'ENTER PHASE LENGTH'
13. INPUT GD
14. CHECK COMMAND
15. -((I = ^ (COMM'ND') / GDIN)
16. CHECK FOR POSITIVE NUMBER
17. -((IN > 0) / GDSETT
18. PRINT 'LENGTH IN TIME. MUST BE POSITIVE'
19. GDIN
20. SET T TO PHASE LENGTH AND GET FAILURE RATE
21. GDSETT: T = IN
22. GDINL: PRINT 'ENTER COMPONENT FAILURE RATE'
23. INPUT GD
24. CHECK COMMAND
25. -((I = ^ (COMM'ND') / GDINL
26. CHECK FOR POSITIVE NUMBER
27. -((IN > 0) / GDCHECKSIZE
28. PRINT 'RATE IN FAILURES PER UNIT TIME. MUST BE POSITIVE'
29. GDINL
30. CHECK REASONABleness OF FAILURE RATE
31. GDCHECKSIZE: -((IN = 1E^{-10}) ^ (IN <= 0.1)) / GDSETL
32. PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT
33. GDINL
34. PRINT QUAD IN
35. GDDBIG
36. PRINT QUAD 'IS SMALL'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
(40) +GDLYESNOIN
(41) GDLIBG:ISTRYNQUAD 'IS LARGE'
(42) GDLYESNOIN:PRINT ' FOR A FAILURE RATE. DO YOU WANT THIS VALUE?'
(43) "(~INYES)/GDINL
(44) a
(45) a SET LAMBDA TO FAILURE RATE AND PERFORM CALCULATIONS
(46) GDSETL:LAMBDA+IN
(47) a
(48) a INITIALIZE THE P MATRIX
(49) Z+(N,N)p0
(50) a DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX
(51) INDEX=Q((2*N)p2)ENCODE(N-1
(52) a
(53) a LOOP THROUGH INDEX TO CREATE P
(54) a INITIALIZE LOOPS
(55) I=1
(56) J=1
(57) a DETERMINE THE NUMBER OF SUCCESS TRANSITIONS
(58) GDLOOP:SIZE+=INDEX[I;]/INDEX[J;]
(59) a FIND THE SUCCESS AND FAILURE PROBABILITIES
(60) FAIL+=(1-+LAMBDAXT)*((+/INDEX[I;]/)-SIZE)
(61) SUCCESS+=LAMBDAXT*SIZE
(62) Z[I;J]+=SUCCESS*FAIL*(SIZE+/
(63) a INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE
(64) J+=1
(65) +(J<=N)/GDLOOP
(66) a RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
(67) J=1
(68) I=I+1
(69) +(I<=N)/GDLOOP

V GDINFO

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

APPENDIX: LISTING OF METAPHOR
PHASE; ENTER A
PRINT 'SINGLE POSITIVE INTEGER. NEXT YOU WILL BE PROMPTED FOR
THE FAILURE'
PRINT 'RATE OF THE COMPONENTS. AGAIN ENTER A SINGLE POSITIVE
NUMBER. IF
PRINT 'THIS NUMBER IS NOT BETWEEN 1E-1 AND 1E-10, YOU WILL BE
ASKED FOR CONFIRMATION.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
IF (~INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE
EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT
NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE
EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT
NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE
 RESEARCH
ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

\v
\v Z=M \text{ GIVEN } N; \text{ ROW}
\text{ ROUTINE FOR INPUTTING CONSTANT MATRIX OF PROBABILITIES}
\text{ EACH ROW MUST SUM TO ONE. THE MATRIX IS TO HAVE SHAPE}
M \times N
\text{ INITIALIZE THE MATRIX}
\text{ Z=}(M,N) \cdot 0
\text{ INPUT AND CHECK THE MATRIX}
\text{ PRINT 'ENTER THE MATRIX, 1 ROW AT A TIME'
\text{ INITIALIZE ROW COUNTER}
\text{ ROW}=1
\text{ PRINT ''

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
GGIN:((PRINTQUAD 'ROW ');
(PRINTQUAD ROW);PRINTQUAD ' : ')
PRINT ''
GET ROW DATA
N INPUT GG
CHECK FOR COMMAND
+(1=A e 'COMMAND')/GGIN
CHECK VALIDITY OF INPUT. FIRST CHECK FOR PROBABILITIES
+(CHECKPROB IN)/GGIN
CHECK THAT EACH ROW SUMS TO ONE. IF NOT, ASK AGAIN.
+((+/IN)=1)/GGININSERT
PRINT 'THE SUM OF PROBABILITIES IN EACH ROW MUST BE 1'
GGIN
INSERT THE ROW INTO THE MATRIX
GGINSERT: Z[ROW;]+IN
INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE
ROW=ROW+1
+(ROW<=N)/GGIN
ELSE LEAVE

GGINFO
ROUTINE GIVING HELP ON INPUTTING GIVEN P MATRICES
PRINT 'ENTER AN MxN ARRAY, ONE ROW AT A TIME. EACH ENTRY MUST'
PRINT 'BE BETWEEN 0 AND 1 INCLUSIVE AND THE ENTRIES OF EACH ROW MUST SUM'
PRINT 'TO ONE. ENTER EACH ROW AS A SERIES OF N NUMBERS WITH SPACES AND/OR'
PRINT 'COMMAS BETWEEN EACH.'
PRINT 'EXAMPLE:'
PRINT '.25 0.5,.1 .015'
PRINT 'HERE, THE MATRIX HAS FOUR ENTRIES PER ROW.'
PRINT ''
DO YOU WANT REFERENCES?
LEAVE IF NOT
+(~INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT '
J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS.' NASA
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT '
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT '
APPENDIX: LISTING OF METAPHOR
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE'
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT '
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT '
PRINT 'S. PAKIN, "APL/360 REFERENCE MANUAL," SCIENCE
RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT '

\[ Z = \text{GIDENTITY N} \]
\[ \text{ROUTINE FOR GENERATING AN } N \times N \text{ IDENTITY MATRIX} \]
\[ \text{RETURN THE MATRIX} \]
\[ Z = (N, N, p(1, (N \times 0)) \]

\[ Z = \text{GNFAIL N; T; LAMBDA; GROUPS; NUM; INDEX; I; J; COEFF; FAIL; SUCCESS} \]
\[ \text{ROUTINE FOR GENERATING THE NFAIL TYPE P MATRIX} \]
\[ \text{THE MATRIX IS TO HAVE SHAPE } N \times N \]
\[ \text{GET PHASE LENGTH} \]
\[ \text{GNINT: PRINT 'ENTER PHASE LENGTH'} \]
\[ 1 \text{ INPUT GN} \]
\[ \text{CHECK COMMAND} \]
\[ \text{ modifying COMMAND')/GNINT} \]
\[ \text{CHECK FOR POSITIVE NUMBER} \]
\[ \text{GNSETT: T \times \text{IN}} \]
\[ \text{PRINTER 'LENGTH IN TIME. MUST BE POSITIVE'} \]
\[ \text{GNINT} \]
\[ \text{SET T TO PHASE LENGTH AND GET FAILURE RATE} \]
\[ \text{GNSETT: T = \text{IN}} \]
\[ \text{GNINT: PRINT 'ENTER COMPONENT FAILURE RATE'} \]
\[ 1 \text{ INPUT GN} \]
\[ \text{CHECK COMMAND} \]
\[ \text{modifying COMMAND')/GNINT} \]
\[ \text{CHECK FOR POSITIVE NUMBER} \]
\[ \text{GNSETT: T \times \text{IN}} \]
\[ \text{PRINTER 'RATE IN FAILURES PER UNIT TIME. MUST BE POSITIVE'} \]
\[ \text{GNINT} \]
\[ \text{CHECK REASONABLENESS OF FAILURE RATE} \]
\[ \text{GNCHECKSIZE: }((\text{IN} \geq 1E-10) \land (\text{IN} \leq 0.1))/\text{GNSETL} \]

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT MESSAGE CHECK TO MAKE SURE THE VALUE IS CORRECT

PRINTQUAD IN

IF (IN > 0.1) GNLIG

PRINTQUAD 'IS SMALL'

GNYESNOIN

CNLIG: PRINT 'IS LARGE'

GNYESNOIN: PRINT 'FOR A FAILURE RATE. DO YOU WANT THIS VALUE?'

I =NO)/GNINL

SET LAMBDA TO FAILURE RATE AND GET NUMBER OF GROUPS

CNSEL: LAMBDAIN

GNING: PRINT 'ENTER NUMBER OF GROUPS'

INPUT GN

CHECK COMMAND

(1 = \epsilon 'COMMAND') / CNING

CHECK FOR POSITIVE INTEGER

(CHECKPOS IN )/GNING

SET GROUPS TO NUMBER OF GROUPS AND GET COMPONENTS PER GROUP

GROUPS IN

GNINN: PRINT 'ENTER NUMBER OF COMPONENTS PER GROUP (SPACE BETWEEN EACH NUMBER):'

GROUPS INPUT GN

CHECK COMMAND

(1 = \epsilon 'COMMAND') / GNINN

CHECK FOR POSITIVE INTEGER

(CHECKPOS IN )/GNINN

CHECK FOR THE PROPER NUMBER OF STATES

(N = \times / IN+1 )/ GNCALC

PRINT

PRINT 'NUMBER PLUS 1] MUST BE THE NUMBER OF STATES.'

PRINT 'HOW MANY GROUPS'

GNING

SET NUM TO THE NUMBER OF POSITIONS FOR THE COMPONENTS IN THE GROUPS AND CALCULATE THE P MATRIX

GNCALC: NUM = IN+1

INITIALIZE THE P MATRIX

Z = (N, N) \rho

DETERMINE AN INDEX VECTOR FOR CALCULATING THE P MATRIX

INDEX = Q(NUM) ENCODE(N-1, N)

LOOP THROUGH INDEX TO CREATE P

INITIALIZE LOOPS

I = 1

J = 1

APPENDIX: LISTING OF METAPHOR
FIRST DETERMINE THE PROPER COEFFICIENT FOR THE TERM UNDER CONSIDERATION

\[ \text{GNLOOP: COEFF} \times (\text{INDEX}[J;] \times \text{INDEX}[I;]) \]

MULTIPLY THE COEFFICIENT WITH THE PROPER EXPONENTIALS

\[ \text{FAIL} \times (1 - \text{LAMBDA} \times t) \times (+/\text{INDEX}[I;]) - (+/\text{INDEX}[J;]) \]

\[ \text{SUCCESS} \times \text{LAMBDA} \times t \times (+/\text{INDEX}[J;]) \]

\[ Z[I;J] \times \text{COEFF} \times \text{FAIL} \times \text{SUCCESS} \]

INCREMENT COLUMN COUNTER AND BRANCH IF APPROPRIATE

\[ J + J + 1 \]

\[ \text{IVAL} \leq N \text{/GNLOOP} \]

RESET COLUMN COUNTER, INCREMENT ROW COUNTER AND BRANCH IF APPROPRIATE

\[ J + 1 \]

\[ I + I + 1 \]

\[ \text{IVAL} \leq N \text{/GNLOOP} \]

\[ \text{GNNINFO} \]

ROUTINE GIVING HELP ON NFAIL TYPE P MATRIX

PRINT 'METAPHOR WILL GENERATE A P MATRIX ASSUMING THE MATRIX REPRESENTS A'

PRINT 'SYSTEM HAVING M GROUPS OF K(N) COMPONENTS EACH, WHERE K IS A '

PRINT 'FUNCTION OF THE GROUP. THE COMPONENTS FAIL INDEPENDENTLY'

PRINT 'AND ARE ASSUMED TO HAVE A POISSON DISTRIBUTION. ALSO, ONCE'

PRINT 'A COMPONENT HAS FAILED, IT CANNOT BECOME GOOD AGAIN.'

PRINT 'THE STATE OF THE SYSTEM IS THE NUMBER OF ACTIVE COMPONENTS'

PRINT 'IN EACH GROUP. THE NUMBER OF STATES DECLARED'

PRINT 'FOR THE PHASE MUST BE THE PRODUCT OF [THE NUMBER OF COMPONENTS'

PRINT 'IN EACH GROUP PLUS ONE]. FOR EXAMPLE, IF THE SYSTEM HAS 3 GROUPS'

PRINT 'CONTAINING RESPECTIVELY 2, 5, AND 7 COMPONENTS, THEN THE PHASE HAS'

PRINT '((2+1) \times (5+1) \times (7+1) = 144 STATES. '

PRINT '

PRINT 'YOU WILL BE ASKED THE LENGTH OF THE PHASE; ENTER A SINGLE POSITIVE INTEGER.'

PRINT 'NEXT YOU WILL BE PROMPTED FOR THE FAILURE RATE OF THE COMPONENTS.'

PRINT 'AGAIN ENTER A SINGLE POSITIVE NUMBER. IF THIS NUMBER IS NOT BETWEEN'

PRINT '1E-1 AND 1E-10, YOU WILL BE ASKED FOR CONFIRMATION. YOU WILL THEN'

PRINT 'BE ASKED THE NUMBER OF GROUPS; ENTER THIS AS A SINGLE POSITIVE INTEGER.'
PRINT 'FINALLY, METAPHOR WILL REQUEST THE NUMBER OF COMPONENTS IN EACH GROUP.'
PRINT 'THIS SHOULD BE INPUT AS A ROW OF POSITIVE INTEGERS SEPARATED BY '.
PRINT 'SPACES OR COMMAS.'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
LEAVE IF NOT
->(~INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA GRANT NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, ''MODELS AND TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,'' NASA GRANT NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, ''APL\360 REFERENCE MANUAL,'' SCIENCE RESEARCH ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

TRAJECTORY SET EVALUATION FUNCTIONS

Z+GETACCLEVPROB LEVEL;T;TRAJPROB;NUMTRAJSETS;I;G;F;V

a ROUTINE FOR DETERMINING THE PROBABILITY OF AN ACCOMPLISHMENT LEVEL
a
a INITIALIZE THE COLLECTION VARIABLE
Z+0
a
a GET THE NUMBER OF TRAJECTORY SETS
NUMTRAJSETS+GETNUMTRAJSETS LEVEL
a
a LOOP THROUGH FOR EACH SET, GETTING ITS VALUES AND CALCULATING ITS PROBABILITY
a
a INITIALIZE COUNTER
T+1
GAPTLOOP: T+->((PRINTQUAD 'TRAJECTORY SET ');PRINTQUAD T)
PRINT ';'

I=GETIVECTOR
G=GETGMATRICES
F=GETFVECTOR
V=GETVVALUES
TRAJPROB=CALCTRAJPROB

ADD THE PROBABILITY TO THE COLLECTION VARIABLE
Z+Z+TRAJPROB
INCREMENT TRAJECTORY COUNTER AND BRANCH IF NECESSARY
T=T+1
(T<=NUMTRAJSETS)/GAPTLOOP
RESET DEFINITION FLAGS
DEFNUMTRAJSETS+DEFI+DEFG+DEFV+0

Z+GETNUMTRAJSETS L

ROUTINE FOR FETCHING THE NUMBER OF ACCOMPLISHMENT LEVEL
TRAJECTORY SETS FOR LEVEL L

INPUT NUMBER OF TRAJECTORY SETS
PRINT '
PRINT (PRINTQUAD 'ACCOMPLISHMENT LEVEL ');PRINTQUAD L-1)
PRINT '
GNTSIN:PRINT 'NUMBER OF TRAJECTORY SETS FOR THIS ACCOMPLISHMENT
LEVEL?'
1 INPUT GNTS
CHECK FOR COMMAND
(1='COMMAND')/GNTSIN
CHECK VALIDITY OF INPUT
(CHECKPOS IN)/GNTSIN
ELSE SET THE NUMBER OF ACCOMPLISHMENT LEVELS AND EXIT
Z+IN
DEFNUMTRAJSETS+1

GNTSINFO
ROUTINE GIVING HELP ON INPUTTING THE NUMBER OF
TRAJECTORIES IN A GIVEN ACCOMPLISHMENT LEVEL
PRINT 'ENTER THE NUMBER OF TRAJECTORY SETS ASSOCIATED WITH
THIS ACCOMPLISHMENT'
PRINT 'LEVEL AS A SINGLE POSITIVE INTEGER.'
PRINT 'EXAMPLE:'

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
PRINT 'THIS INFORMS METAPHOR THAT THE ACCOMPLISHMENT LEVEL HAS
FOUR TRAJECTORY'
PRINT 'SETS DESCRIBING IT.'
PRINT 'DO YOU WANT REFERENCES?'
PRINT ~(~INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'
PRINT '
PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT '
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT '
PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE '
PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT '
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT '
PRINT 'S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE
RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

V

Z<GETIVECTOR

ROUTE FOR INPUTTING INITIAL STATE VECTOR

GIVIN: PRINT 'ENTER THE I VECTOR (SPACE BETWEEN EACH ENTRY):'
STATESPERPHASE[1]INPUT GIV
CHECK FOR COMMAND
->(1\^e'COMMAND')/GIVIN
CHECK VALIDITY OF INPUT.
->(CHECKBIN IN)/GIVIN
SET I VECTOR AND LEAVE
Z<MAXNUMSTATES\in,(MAXNUMSTATES\rho0)
DEFI+1

APPENDIX: LISTING OF METAPHOR
ROUTINE GIVING HELP ON INPUTTING I VECTORS

PRINT 'ENTER THE INITIAL PROBABILITY DISTRIBUTION FOR THIS
TRAJECTORY.'

PRINT 'TYPE A PROBABILITY BETWEEN ZERO AND ONE INCLUSIVE
CORRESPONDING TO EACH'

PRINT 'STATE"S INITIAL PROBABILITY. SEPARATE EACH NUMBER
WITH SPACES AND/OR COMMAS.'

PRINT 'THE ORDER OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER
OF THE INITIAL STATES.'

PRINT 'THE NUMBER OF ENTRIES SHOULD BE THE SAME AS THE NUMBER
OF STATES'

PRINT 'IN THE FIRST PHASE MODEL.'

PRINT 'EXAMPLE:

.3 0.5,.2'

PRINT 'THIS INFORMS METAPHOR THAT FOR THE TRAJECTORY SET UNDER
CONSIDERATION, THE

PRINT 'PROBABILITY THE SYSTEM BEGINS IN THE FIRST STATE OF
PHASE 1 IS 0.2, FOR THE'

PRINT 'SECOND STATE, THE PROBABILITY IS 0.5, AND FOR THE THIRD
STATE, THE PROBABILITY'

PRINT 'IS 0.2.'

PRINT ''

PRINT 'DO YOU WANT REFERENCES?'

LEAVE IF NOT

PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'

PRINT ''

PRINT ' J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE'

PRINT ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'

PRINT ' NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'

PRINT ''

PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'

PRINT ''

PRINT ' J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE'

PRINT ' EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'

PRINT ' NSG 1306, STATUS REPORT NO. 4, JULY 1978.'

PRINT ''

PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'

PRINT ''

PRINT ' S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE
RESEARCH'

PRINT ' ASSOCIATES, INC., CHICAGO, 1972.'

PRINT''
\[ V \text{GETGMATRICES; PHASE} \]

[1] \( \text{ROUTINE FOR INPUTTING THE G MATRICES FOR A TRAJECTORY SET} \)

[2] \( \text{EACH G MATRIX WILL BE REPRESENTED AS A VECTOR OF ITS} \)

[3] \( \text{DIAGONAL ELEMENTS} \)

[4] \( \text{INITIALIZE THE G DIAGONAL MATRIX. ALSO SET DEFINITION} \)

[5] \( \text{FLAG} \)

[6] \( \text{GET}(\text{MAXNUMSTATES, MAXNUMSTATES})p0 \)

[7] \( \text{DEFG}=1 \)

[8] \( \text{LOOP THROUGH EACH PHASE EXCEPT THE LAST, GETTING THE G} \)

[9] \( \text{MATRICES} \)

[10] \( \text{INITIALIZE THE PHASE COUNTER} \)

[11] \( \text{PHASE}=1 \)

[12] \( \text{INPUT AND CHECK THE G VECTOR} \)

[13] \( \text{GGMIN}:=((\text{PRINTQUAD 'PHASE '});(\text{PRINTQUAD PHASE});\text{PRINTQUAD} \)

[14] \( \text{'}END')GGM \)

[15] \( \text{PRINT 'ENTER THE G DIAGONAL (SPACE BETWEEN EACH ENTRY):'} \)

[16] \( \text{STATESPERPHASE[PHASE]}\text{INPUT GGM} \)

[17] \( \text{CHECK FOR COMMAND} \)

[18] \( \text{'+(1=\)} \)

[19] \( \text{COMMAND')/GGMIN} \)

[20] \( \text{CHECK VALIDITY OF INPUT.} \)

[21] \( \text{+(CHECKBIN IN)/GGMIN} \)

[22] \( \text{PLACE THE INPUT IN THE SET OF G VECTORS} \)

[23] \( \text{Z[PHASE: \text{IN}]=IN} \)

[24] \( \text{INCREMENT THE PHASE COUNTER AND BRANCH IF APPROPRIATE} \)

[25] \( \text{PHASE}=\text{PHASE+1} \)

\[ V \text{GGMINFO} \]

[1] \( \text{ROUTINE GIVING HELP ON INPUTTING G MATRICES} \)

[2] \( \text{PRINT 'ENTER THE DIAGONAL OF THE CHARACTERISTIC (G) MATRIX FOR} \)

[3] \( \text{THE PHASE UNDER CONSIDERATION.'} \)

[4] \( \text{PRINT 'EACH ENTRY SHOULD BE EITHER 0 OR 1 AND SHOULD BE} \)

[5] \( \text{SEPARATED FROM THE OTHER'} \)

[6] \( \text{PRINT 'ENTRIES BY SPACES AND/OR COMMAS. THE NUMBER OF ENTRIES} \)

[7] \( \text{SHOULD BE THE SAME'} \)

[8] \( \text{PRINT 'AS THE NUMBER OF STATES FOR THE PHASE. ALSO THE ORDER} \)

[9] \( \text{OF THE ENTRIES SHOULD'} \)

[10] \( \text{PRINT 'CORRESPOND TO THE ORDER OF THE STATES AS CONSIDERED} \)


[12] \( \text{PRINT 'METAPHOR PACKAGE FOR THE PHASE.'} \)

\text{APPENDIX: LISTING OF METAPHOR}
PRINT 'THIS INFORMS METAPHOR THAT THE CHARACTERISTIC MATRIX
FOR THIS PHASE IS'
PRINT '0 0 0 0 0'
PRINT '0 0 0 0 0'
PRINT '0 0 1 0 0'
PRINT '0 0 0 1 0'
PRINT '0 0 0 0 1'
PRINT ''
PRINT 'DO YOU WANT REFERENCES?'
PRINT 'NO' IF NOT
PRINT '~(IN YES)/0'
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT ''
PRINT 'J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT'
PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT ''
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT ''
PRINT 'S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE
RESEARCH'
PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'
PRINT ''

V Z+GETVECTOR

ROUTINE FOR INPUTTING FINAL STATE VECTOR

INPUT AND CHECK THE F VECTOR
GEVIN: PRINT 'ENTER THE F VECTOR (SPACE BETWEEN EACH ENTRY):'
STATESPERPHASE[NUMPHASES] INPUT GENV
CHECK FOR COMMAND
="COMMAND")/GEVIN
CHECK VALIDITY OF INPUT.
=CHECKBIN IN)/GEVIN
SET F VECTOR AND LEAVE
Z+(MAXNUMSTATES,1)PIN,(MAXNUMSTATES@0)
DEFF+1

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
ROUTE GIVING HELP ON INPUTTING THE F VECTORS

PRINT 'ENTER THE CHARACTERISTIC (F) VECTOR FOR THE TRAJECTORY
SET UNDER CONSIDERATION.'
PRINT 'EACH ENTRY SHOULD BE EITHER 0 OR 1 AND SHOULD BE
SEPARATED FROM THE OTHER ENTRIES.'
PRINT 'BY SPACES AND/OR COMMAS. THE NUMBER OF ENTRIES SHOULD
BE THE SAME AS THE NUMBER'
PRINT 'OF STATES OF THE FINAL PHASE MODEL. ALSO THE ORDER OF
THE ENTRIES SHOULD'
PRINT 'CORRESPOND TO THE ORDER OF THE STATES AS CONSIDERED
ELSEWHERE IN THE'
PRINT 'METAPHOR PACKAGE FOR THE FINAL PHASE.'
PRINT 'EXAMPLE:'
PRINT '  1 0, 1 , 0'
PRINT 'THIS INFORMS METAPHOR THAT THE CHARACTERISTIC VECTOR
FOR THIS TRAJECTORY'
PRINT 'SET IS'
PRINT '  1'
PRINT '  0'
PRINT '  1'
PRINT '  0 ,'
PRINT 'DO YOU WANT REFERENCES?'

LEAVE IF NOT
\(~INYES)/0
PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS, SEE'
PRINT '
PRINT '  J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE '
PRINT '  EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT
PRINT '  NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'
PRINT '  '
PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'
PRINT '
PRINT '  J. F. MEYER, "MODELS AND TECHNIQUES FOR EVALUATING
THE '
PRINT '  EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS," NASA
GRANT
PRINT '  NSG 1306, STATUS REPORT NO. 4, JULY 1978.'
PRINT '  '
PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'
PRINT '
PRINT '  S. PAKIN, "APL\360 REFERENCE MANUAL," SCIENCE
RESEARCH'
PRINT '
PRINT '  ASSOCIATES, INC., CHICAGO, 1972.'

APPENDIX: LISTING OF METAPHOR
V Z-GETVVALUES; GVVTURE; GVVFALSE
[1] R ROUTINE FOR INPUTTING TIME-INARIANT BASIC VARIABLE PROBABILITIES
[2] R
[4] R IF NO TIME-INARIANT BASIC VARIABLES, THEN SET V=0, TURN ON
[6] Z+0
[7] DEFV+1
[8] +(NUMBASICVARIABLES=0)/0
[10] NUMBASICVARIABLES INPUT GV
[11] R CHECK FOR COMMAND
[12] +(1=\n[13] e 'COMMAND'/GVVIN
[14] R CHECK VALIDITY OF INPUT.
[15] +(CHECKTRI IN)/GVVIN
[16] R SET V VECTOR
[17] R FIRST, SET THE TRUE VECTOR
[18] GVVTURE+IN 0 2
[19] R THEN SET THE FALSE VECTOR
[20] GVVFALSE+IN 1 2
[21] R COMBINE FOR THE V VECTOR
[22] Z+GVVTURE,GVVFALSE

V GVVININFO
[1] R ROUTINE GIVING HELP ON INPUTTING THE TIME-INARIANT BASIC VARIABLE VECTOR
[2] R
[3] R PRINT 'ENTER A VECTOR OF 0'S, 1'S, AND 2'S TO INDICATE
[5] R PRINT 'TIME-INARIANT BASIC VARIABLE'S OCCURRENCES OR NON-
[6] R OCCURRENCES SHOULD BE'
[7] R PRINT 'CONSIDERED IN THE TRAJECTORY SET. THE CODING IS AS
[8] R PRINT '
[9] R PRINT ' 0 THE CORRESPONDING BASIC VARIABLE'S
[10] R PRINT ' OCCURRENCE SHOULD'
[12] R PRINT ' 1 THE CORRESPONDING BASIC VARIABLE'S NON-
[13] R OCCURRENCE'
[14] R PRINT ' SHOULD BE CONSIDERED'
[15] R PRINT ' 2 EITHER THE OCCURRENCE OR NON-OCCURRENCE

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
OF THE 'CORRESPONDING BASIC VARIABLE SHOULD BE
CONSIDERED'.

PRINT 'I.E., THE BASIC VARIABLE IS A 'DON'T' CARE')',

PRINT 'ENTER A ROW OF 0''S, 1''S, AND 2''S, SEPARATING EACH
ENTRY BY SPACES'.

PRINT 'AND/OR COMMAS. THE NUMBER OF ENTRIES SHOULD BE THE
SAME AS THE '.

PRINT 'NUMBER OF TIME-ININVARIANT BASIC VARIABLES DECLARED
EARLIER. ALSO THE ORDER'.

PRINT 'OF THE ENTRIES SHOULD CORRESPOND TO THE ORDER OF THE
BASIC VARIABLES'.

PRINT 'AS CONSIDERED ELSEWHERE IN THE METAPHOR PACKAGE.'

PRINT 'EXAMPLE:'

PRINT ' 0, 1, 2'

PRINT 'THIS INDICATES TO METAPHOR THAT FOR THIS TRAJECTORY
SET, THE OCCURRENCE'

PRINT 'OF THE FIRST TIME-ININVARIANT BASIC VARIABLE IS IMPORTANT
TO THE TRAJECTORY'

PRINT 'SET, THE NON-OCCURRENCE OF THE SECOND AND THIRD TIME-
ININVARIANT BASIC VARIABLES'

PRINT 'IS IMPORTANT, AND THAT THE FOURTH TIME-ININVARIANT BASIC
variable IS IRRELEVANT'.

PRINT '

PRINT 'DO YOU WANT REFERENCES?'

PRINT 'LEAVE IF NOT

PRINT 'YES)/0

PRINT 'FOR FURTHER INFORMATION ON PERFORMABILITY MODELING AND
ANALYSIS. SEE'

PRINT '

PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE'

PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'

PRINT 'NSG 1306, STATUS REPORT NO. 3, NOVEMBER 1977.'

PRINT '

PRINT 'FOR FURTHER INFORMATION REGARDING METAPHOR, SEE'

PRINT '

PRINT 'J. F. MEYER, 'MODELS AND TECHNIQUES FOR EVALUATING
THE'

PRINT 'EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS,' NASA
GRANT'

PRINT 'NSG 1306, STATUS REPORT NO. 4, JULY 1978.'

PRINT '

PRINT 'FOR FURTHER INFORMATION REGARDING APL, SEE'

PRINT '

PRINT 'S. PAKIN, 'APL\360 REFERENCE MANUAL,' SCIENCE
RESEARCH'

PRINT 'ASSOCIATES, INC., CHICAGO, 1972.'

APPENDIX: LISTING OF METAPHOR
CALCULATES THE PROBABILITY OF THE GIVEN TRAJECTORY.

COMPUTE THE INITIAL PHASE PROBABILITY

LOOP THROUGH THE OTHER PHASES

INITIALIZE PHASE COUNTER

FIRST CONSIDER THE SUCCESS STATES MATRIX

THEN THE INTERPHASE MATRIX

THEN THE TRANSITION MATRIX

INCREMENT PHASE COUNTER AND BRANCH IF APPROPRIATE

POST-MULTIPLY BY FINAL STATE

FINALLY, MULTIPLY BY THE TIME-INVARIANT BASIC VARIABLE PROBABILITIES

IF NO TIME-INVARIANT BASIC VARIABLES, EXIT

IF NUMERIC, DATA RETURNED IN 'IN', ELSE IF COMMAND, 'COMMAND' RETURNED

GET INPUT

ECHO INPUT IF DESIRED

CHANGE ALL SCALAR INPUTS TO VECTORS
A CHECK FOR COMMAND AND SET COMMAND VECTOR
COMMANDVECTOR=COMMANDLISTerIN
A EXECUTE COMMAND IF PRESENT
 INCOMMAND:=(COMMANDVECTOR)/INHELP,INEXIT,INDATA,
INALTER,INCALC,INECHO,INBRIEF,INCOM,INEVAL
A ELSE CHECK THE SHAPE OF THE INPUT. 'SHAPE' IS ALWAYS AN INTEGER SCALAR. IF GOOD, LEAVE.
A =(SHAPE=PIN)/0
A OTHERWISE THE INPUT WAS OF THE WRONG DIMENSION. PRINT ERROR MESSAGE AND TRY AGAIN.
A (+PRINTQUAD 'ENTER '),(PRINTQUAD SHAPE),PRINTQUAD ',ITEMS')
A =ININ
A =0
A HELP REQUESTED
A INHELP:COMMANDHELP ROUTINE
A IN='COMMAND'
A =0
A END THE PROGRAM
A INEXIT:=
A DISPLAY OF DATA REQUESTED
A INDATA:COMMANDDATA
A IN='COMMAND'
A =0
A CHANGE OF DATA REQUESTED
A INALTER:COMMANDALTER
A IN='COMMAND'
A =0
A SETTING OF THE ECHO SWITCH DESIRED
A INECHO:COMMANDECHO IN
A IN='COMMAND'
A =0
A COMMENT DESIRED

APPENDIX: LISTING OF METAPHOR
INCOM:COMMAND

IN+ 'COMMAND'

→0

a

SETTING OF BRIEF SWITCH DESIRED

INBRIEF:COMMANDBRIEF IN

→0

a

PERFORMABILITY COMPUTATION DESIRED

INDEVAL:COMMANDEVAL

→0

→ INYIN YES; IN

ROUTE ALL ASKING YES AND NO ANSWERS. 1 RETURNED IF YES, 0 IF NO

PRINT PROMPT AND GET ANSWER

IYIN:PRINT '0:'

IN+IN

ECHO THE INPUT IF DESIRED

→(~ECHOINPUT)/IYSCAN

PRINT(IN)

ASSUME YES HAS PRIORITY. LOOK FOR Y OR 1

IYSCAN:Z+V/ 'Y1'eIN

IF N OR 0 INPUT OR IF YES INPUT, EXIT

→(ZVV/'NO'eIN)/0

ELSE TRY AGAIN

PRINT 'ENTER YES OR NO'

→IYIN

→ IN Z+CHECKBIN CHECKNO

RETURN 0 IF CHECKNO CONTAINS ONLY BINARY ZEROS AND ONES

ELSE PRINT MESSAGE AND RETURN 1

CHECK FOR PROPER BINARY ELEMENTS

Z+~(\_/CHECKNO e 0 1)

→(Z)/0

PRINT 'EACH ENTRY MUST BE EITHER 0 OR 1'

→0

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
\[ Z + \text{CHECKPOSI CHECKNO} \]

[1] RETURN 0 IF CHECKNO IS A POSITIVE INTEGER, ELSE PRINT MESSAGE AND RETURN 1
[2] 
[3] CHECK FOR POSITIVENESS
[4] \( Z \leftarrow \neg \neg \neg / (\text{CHECKNO} \geq 0) \)
[5] \( \rightarrow (\neg Z) / \text{CHECKINT} \)
[6] PRINT 'INPUT NOT POSITIVE'
[7] \( \rightarrow 0 \)
[8] 
[9] CHECK FOR INTEGER
[10] \( \text{CHECKINT} : Z \leftarrow \neg \neg \neg / \text{CHECKNO} = \{ \text{CHECKNO} \) 
[11] \( \rightarrow (\neg Z) / 0 \)
[12] PRINT 'INPUT NOT AN INTEGER'

\[ Z + \text{CHECKPROB CHECKNO} \]

[1] RETURN 0 IF CHECKNO IS BETWEEN 0 AND 1 INCLUSIVE, ELSE PRINT MESSAGE AND RETURN 1
[2] 
[3] CHECK FOR PROPER RANGE
[4] \( Z \leftarrow \neg \neg \neg / \text{CHECKNO} \geq 0 \land \neg \neg \neg / \text{CHECKNO} \leq 1 \)
[5] \( \rightarrow (\neg Z) / 0 \)
[6] PRINT 'INPUT NOT BETWEEN 0 AND 1'

\[ Z + \text{CHECKTRI CHECKNO} \]

[1] RETURN 0 IF CHECKNO CONTAINS ONLY TRINARY ZEROS, ONES, AND TWOS
[2] ELSE PRINT MESSAGE AND RETURN 1
[3] CHECK FOR PROPER TRINARY ELEMENTS
[4] \( Z \leftarrow \neg \neg \neg / \text{CHECKNO} \in \{ 0, 1, 2 \} \)
[5] \( \rightarrow (\neg Z) / 0 \)
[6] PRINT 'EACH ENTRY MUST BE EITHER 0, 1, OR 2'
[7] \( \rightarrow 0 \)


\nV Z-PRINT Q

[1] a
[2] a PRINTING ROUTINE FOR METAPHOR
[3] a IF BRIEF+1 NO OUTPUT IS GIVEN
[4] a
[5] a CHECK FOR TERSE INPUT FLAG
[6] a (BRIEFOUTPUT=1)/O
[7] a FULL OUTPUT DESIRED, GIVE IT AND LEAVE
[8] Z+Q

V

\nV Z-PRINTQUAD Q

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

\nAPL SUPPORT FUNCTION

\nV Z+M ENCODE N;ROW

[1] a
[2] a THE ENCODE FUNCTION EMPLOYED ON MTS WILL NOT ACCEPT
[3] a VECTORS AS
[4] a ARGUMENTS ON THE RIGHT HAND SIDE. THIS FUNCTION SIMULATES
[7] a (O=\n)/ESINGLE
[8] a INITIALIZE THE ARRAY AND LOOP COUNTER
[9] Z+((1/\n),(\n))\n0

METAPHOR (VERSION 1) PROGRAMMER'S GUIDE
INDEX

A, prefix for labels in the function COMMANDALTER, 6
Abbreviations, of definition flags, 19
of labels, 5-6
ACCLEV, abbreviation for ACCOMPLISHMENT LEVEL, 5
ACCLEVEL, definition flag for, 19
Accomplishment level, 2
abbreviation for, 5
Algorithm, EVAL, 25
Alphabetical ordering, 3
ALTER, control and information exchange techniques for, 17
description of METAPHOR variable, 47
METAPHOR variable, 64, 70
ALTERVECTOR, index code for, 83
APL, 2, 6
APL support function, 187

Ballance, R. A., 1
Base model, 1
Base model trajectory set, 2
BASICVARIABLES, definition flag for, 19
description of METAPHOR variable, 47
METAPHOR variable, 64, 85
BIN, abbreviation for BINARY, 5
Binary, abbreviation for, 5
BRIEF, description of METAPHOR variable, 47
METAPHOR command, 11-12, 47, 63, 70, 110, 115, 117, 184, 190, 194, 199, 236, 238
METAPHOR function, 4, 10, 12, 51, 54, 63, 186, 199
METAPHOR function description, 63
BRIEFOUTPUT, description of METAPHOR variable, 47
METAPHOR variable, 70, 115-117, 119, 122

CALC, abbreviation for CALCULATE, 5
description of METAPHOR variable, 47
METAPHOR variable, 70
CALCTRIPROB, abbreviation for labels in, 6
METAPHOR function, 6, 47, 49-51, 64, 82, 187, 226, 234
METAPHOR function description, 64
Calculate, abbreviation for, 5
Capability function, 2
CC, prefix for labels in the function COMMANDCALC, 6
CCALC, prefix for labels in the function COMMANDCALC, 6
CHECK, prefix for labels in the function CHECKPOS1, 6
Prefix for METAPHOR function names, 4
CHECKBIN, METAPHOR function, 7, 47, 51, 66, 98, 100, 103, 106, 115, 187, 227, 230, 236
METAPHOR function description, 66
CHECKNO, description of METAPHOR variable, 47
CHECKPOS1, abbreviation for METAPHOR variable, 47
CHECKPOS1, description of METAPHOR function, 6
CHECKPROB, METAPHOR function, 4, 47, 65, 85, 115, 139, 187, 21
METAPHOR function description, 68
CHECKTRI, METAPHOR function, 47, 51, 69, 107, 115, 187, 232
METAPHOR function description, 69
Clarity, 8
Code, index code for ALTERVECTOR, 17, 83

METAPHOR (Version 1) Programmer's Guide
index code for DATAVECTOR, 18,87
Code names, for functions (for use with the HELP command), 15
Code values, for functions (for use with the HELP command), 15

COM, abbreviation for COMMENT, 5
description of METAPHOR variable, 47
METAPHOR variable, 70

COMMAND, Prefix for METAPHOR function names, 4
COMMANDALTER, abbreviation for labels in, 6
METAPHOR function description, 118

COMMANDBRIEF, METAPHOR function, 12,47,108,110,115,117,122
METAPHOR function description, 122
COMMANDCALC, abbreviation for labels in, 6
METAPHOR function, 6,108-109,117,123,186,194,235
METAPHOR function description, 123

COMMANDCOM, abbreviation for labels in, 6
METAPHOR function, 6,108,110,117,124,186,194
METAPHOR function description, 124
COMMANDDATA, abbreviation for labels in, 6
METAPHOR function, 6,18-19,21,48-49,87,108-109,115,117,125-126,186,194,235
METAPHOR function description, 125

COMMANDDECHO, METAPHOR function, 49,108-109,115,129,186,197,235
METAPHOR function description, 129

COMMANDDEVAL, METAPHOR function,
METAPHOR variable, 70
DATAVECTOR, index code for, 17-18, 87
DECLAREMETAPHOR, METAPHOR function, 47-53, 70-71, 113, 186, 188
METAPHOR function description, 70
DEDFAIL, description of METAPHOR variable, 48
flowchart for, 32
METAPHOR command, 48, 79, 133, 177, 198, 216
METAPHOR function, 25
METAPHOR matrix generation algorithm, 25-26, 28-29, 32, 35, 70, 135-136, 152, 177, 216, 218-219
METAPHOR variable, 70, 72, 188
DEF, Prefix for METAPHOR variable names, 5
DEFACCLEVEL, definition flag for the METAPHOR variable ACCLEVEL, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 95, 118, 125-126
DEFBASICVARIABLES, definition flag for the METAPHOR variable BASICVARIABLES, 19
METAPHOR variable, 70, 85, 118, 125-126
DEFBASICVARS, description of METAPHOR variable, 48
DEFF, definition flag for the METAPHOR variable F, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 81-82, 118, 125-126
DEFG, definition flag for the METAPHOR variable C, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 81-82, 118, 125-126
DEFH, description of METAPHOR variable, 48
METAPHOR variable, 70, 101, 125-126
DEFI, definition flag for the METAPHOR variable I, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 81-82, 118, 125-126
Definition flags, 19
abbreviations of, 19
DEFNUMACCLEVEL, definition flag for the METAPHOR variable NUMACCLEVEL, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 90
DEFNUMBASICVARIABLES, definition flag for the METAPHOR variable NUMBASICVARIABLES, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 91, 125-126
DEFNUMPHASES, definition flag for the METAPHOR variable NUMPHASES, 19
description of METAPHOR variable, 48
METAPHOR variable, 70, 92
DEFNUMSTATES, definition flag for the METAPHOR variable NUMSTATES, 19
description of METAPHOR variable, 49
METAPHOR variable, 70, 125-126
DEFNUMTRAJSETS, definition flag for the METAPHOR variable NUMTRAJSETS, 19
description of METAPHOR variable, 49
METAPHOR variable, 70, 81-82, 93, 97, 118, 125-126
DEFP, definition flag for the METAPHOR variable P, 19
description of METAPHOR variable, 49
METAPHOR variable, 70, 104, 118, 125-126
DEFFPERFROMABILITY, definition flag for the METAPHOR variable PERFROMABILITY, 19

METAPHOR (Version 1) Programmer's Guide
INDEX
prefix for labels in the function GETDATAVECTOR, 6
GDVINFO, METAPHOR function, 111, 115, 132-133, 154, 186, 199, 203
METAPHOR function description, 154
GENERATEH MATRIX, abbreviation for labels in, 6
METAPHOR function, 6, 15, 48, 50-53, 77-78, 101-102, 108, 115, 139, 141, 186, 2
METAPHOR function description, 77
GENERATEPMATRIX, abbreviation for labels in, 6
METAPHOR function, 6-7, 15, 50-53, 79-80, 104-105, 115, 135, 139, 141-142, 186, 208, 215
METAPHOR function description, 79
GENHM, prefix for labels in the function GENERATEH MATRIX, 6
GENPM, prefix for labels in the function GENERATEPMATRIX, 6
GET, Prefix for METAPHOR function names, 4
GETACCLEVPROB, abbreviation for labels in, 6
METAPHOR function, 4, 6, 15, 17, 22-23, 50-51, 83-84, 115, 118-119, 147, 186, 191, 200
METAPHOR function description, 83
GETALTERVECTOR, abbreviation for labels in, 6
flowchart of, 23
METAPHOR function, 4, 6, 15, 17, 22-23, 50-51, 83-84, 115, 118-119, 147, 186, 191, 200
METAPHOR function description, 83
GETBASICVARIABLES, abbreviation for labels in, 6
METAPHOR function, 6, 15, 47-48, 50-51, 68, 85, 108, 115, 118, 120
METAPHOR function description, 85
GETDATAVECTOR, abbreviation for labels in, 6
flowchart of, 24
METAPHOR function, 6, 15, 18, 22, 24, 51, 87-88, 115, 125-126, 154, 186, 194, 202
METAPHOR function description, 87
GETFVECTOR, abbreviation for labels in, 6
METAPHOR function, 6, 15, 50-51, 64, 66, 81-82, 98, 108, 115, 118, 121, 157
METAPHOR function description, 98
GETGMATRICES, abbreviation for labels in, 6
METAPHOR function, 6, 15, 50-52, 64, 66, 81-82, 99, 108, 115, 117-118, 121, 151, 186, 193, 226, 23
METAPHOR function description, 99
GETH MATRICES, METAPHOR function, 49-52, 77, 101, 115, 117-119, 130, 163, 186, 192, 197, 227
METAPHOR function description, 101
GETIVECTOR, abbreviation for labels in, 6
METAPHOR function, 6, 15, 19, 50-52, 64, 66, 81-82, 103, 108, 115, 118, 120, 16227
METAPHOR function description, 103
GETNUMACCLEVEL, abbreviation for labels in, 6
METAPHOR function, 6, 15, 48, 50-52, 67, 90, 108, 1
METAPHOR function description, 90
GETNUMBASICVARIABLES, abbreviation for labels in, 6
METAPHOR function, 6, 15, 48, 50-52, 67, 91, 108, 1
METAPHOR function description, 91
GETNUMPHASES, abbreviation for
INDEX
133,165,186,198,228
METAPHOR function
description, 165
Global labels, 6
GN, description of METAPHOR
variable, 52
METAPHOR variable, 70,142,144
prefix for labels in the
function GNFAIL, 6
GNA, description of METAPHOR
variable, 52
METAPHOR variable, 70,93
prefix for labels in the
function GETNUMBASICVARIABLES, 6
GNINFO, METAPHOR function, 111,115,132-
133,169,186,198,212
METAPHOR function
description, 169
GNBV, description of METAPHOR
variable, 52
METAPHOR variable, 70,91
prefix for labels in the
function GETNUMBASICVARIABLES, 6
GNBVINFO, METAPHOR function, 111,115,132-
133,171,186,198,209
METAPHOR function
description, 171
GNFAIL, abbreviation for labels
in, 6
description of METAPHOR
variable, 50
METAPHOR command, 50,79
METAPHOR function, 6,15,39,50-52,75,79-
80,111,115,117,136,142-
144,167,186,215,222
METAPHOR function
description, 142
METAPHOR variable, 70
GNINFO, METAPHOR function, 111,115,132-
133,167,186,198,224
METAPHOR function
description, 167
GNP, description of METAPHOR
variable, 52
METAPHOR variable, 70,92
prefix for labels in the
function GETNUMPHASES, 6
GNPINFO, METAPHOR function, 111,115,132,173,186,198,2
METAPHOR function
description, 173
GNTS, description of METAPHOR
variable, 52
METAPHOR variable, 70,93
prefix for labels in the
function GETNUMTRAJSETS, 6
GNTSINFO, METAPHOR function, 111,115,132-
133,175,186,199,226
METAPHOR function
description, 175
GPM, description of METAPHOR
variable, 52
METAPHOR variable, 70,79-80
GPMINFO, METAPHOR function, 111,115,132,177,186,198,2
METAPHOR function
description, 177
GS, description of METAPHOR
variable, 52
METAPHOR variable, 70
prefix for labels in the
function GETSTATES, 6
GSINFO, METAPHOR function, 111,115,132,179,186,198
METAPHOR function
description, 179
GVV, description of METAPHOR
variable, 52
METAPHOR variable, 70,106
prefix for labels in the
function GETVVALUES, 6
GVVINFO, METAPHOR function, 111,115,132-
133,181,187,199,232
METAPHOR function
description, 181
H, description of METAPHOR
variable, 49
METAPHOR variable, 64,101
prefix for labels in the
function COMMANDHELP, 6
HELP, control and information
exchange techniques for, 14
description of METAPHOR
variable, 49
flowchart for, 17
METAPHOR command, 2
METAPHOR variable, 70
Hierarchy, model, 1-2
HMATRIXLIST, description of
METAPHOR variable, 53
METAPHOR variable, 70,77-78
I, definition flag for, 19
description of METAPHOR variable, 49
IDENTITY, description of METAPHOR variable, 50
METAPHOR matrix generation algorithm, 7,25
METAPHOR variable, 70
IN, conventions for, 8
description of METAPHOR variable, 50
METAPHOR variable, 77-80,85,90-93,97-99,103,105,108,111-112,135-136,139,142,144
Postfix for METAPHOR label names, 7
prefix for labels in the function INPUT, 6
Prefix for METAPHOR function names, 4
Index code, for ALTERVECTOR, 17,83
for DATAVECTOR, 18,87
INFO, abbreviation for INFORMATION, 5
Postfix for METAPHOR function names, 4
Information, abbreviation for, 5
Information exchange, 8
Information exchange techniques, ALTER, 17
DATA, 17
HELP, 14
INPUT, 8
user assistance commands, 14
INPUT, abbreviation for labels in, 6
APL quad, 9
control and information exchange techniques for the METAPHOR function, 8
flowchart of, 8
METAPHOR function description, 108
Interdependence of METAPHOR functions, 54,61
figure illustrating, 54
Interlevel translations, 2
INYES, abbreviation for labels in, 7
METAPHOR function
METAPHOR function description, 111
I/O and Checking functions, 187
Iverson, K. E., 2
IV, prefix for labels in the function INYES, 7
Labels, abbreviations of, 5-6
conventions for, 6
for loops, 7
global, 6
local, 6
Lexicographical ordering, 3
Local labels, 6
LOOP, Postfix for METAPHOR label names, 7
Loops, labels for, 7
Main functions, 186
Matrix generation, 25
Matrix generator functions, 186
Matrix generators, names of, 7
MAXNUMSTATES, description of
METAPHOR variable, 50
METAPHOR variable, 101, 104
MET, description of METAPHOR
variable, 53
METAPHOR variable, 70, 113
prefix for labels in the
function METAPHOR, 7
METAPHOR, abbreviation for
labels in, 7
APL support function, 187
command EVAL implementation
functions, 186
command functions, 186
command support functions, 186
I/O and Checking functions,
187
listing of, 186
main functions, 186
matrix generator functions, 186
METAPHOR function,
METAPHOR function
description, 113
METAPHOR software package,
1-6, 8, 10-12, 15, 18, 22, 25-
26, 28, 32, 39-40, 45-
47, 50, 54, 61, 63, 70-
71, 77, 79, 95, 113, 115, 117, 1
performance computation
functions, 186
METAPHOR functions, figure
illustrating the
interdependence of, 54
interdependence of, 54, 61
METINFO, METAPHOR function,
111, 115, 132, 134, 183, 186, 1
METAPHOR function
description, 183
Meyer, J. F., 1, 40, 45, 148-
149, 151, 153, 155-156, 158-
160, 162, 164, 166, 168-
173, 175-
176, 178, 180, 182, 184, 190-
191, 202, 204-207, 210-
213, 215, 217, 220-
222, 225, 227-229, 230-
231, 233
Model hierarchy, 1-2
Modeling, 2

Names, conventions for, 4-5
length of command names, 11
of commands, 5, 7
of functions, 4
of matrix generators, 7
of variables, 5
NFAIL, description of METAPHOR
variable, 50
flowchart for, 39
METAPHOR matrix generation
algorithm, 7, 25-
26, 28, 33, 35-36, 39, 142
METAPHOR variable, 70
NUM, abbreviation for NUMBER, 5
NUMACCLEV, definition flag for, 19
description of METAPHOR
variable, 50
METAPHOR variable, 90
NUMBASICVARIABLES, definition
flag for, 19
description of METAPHOR
variable, 50
METAPHOR variable, 64, 85, 91, 106
Number, abbreviation for, 5
NUMPHASES, definition flag for, 19
description of METAPHOR
variable, 50
METAPHOR variable, 92
NUMSTATES, definition flag for, 19
description of METAPHOR
variable, 50
METAPHOR variable, 81-
82, 93, 95, 97, 101, 104
OFF, description of METAPHOR
variable, 50
METAPHOR variable, 70
ON, description of METAPHOR
variable, 50
METAPHOR variable, 70
P, definition flag for, 19
description of METAPHOR
variable, 50
METAPHOR variable, 64,104
akin, S., 2,149,151,153,156,158,160
207,210-
211,213,215,218,220,222,2
228,230-231,233
arometer, Z, 7
erformanceability, 1-2
definition flag for, 19
description of METAPHOR variable, 50
METAPHOR variable, 95,116
erformanceability computation functions, 186
ATRIXLIST, description of METAPHOR variable, 53
METAPHOR variable, 70,79-80
, prefix for labels in the function PRINTQUAD, 7
RINT, METAPHOR function,
4,47,50-51,66-69,77-
85,87-88,90-93,97-
106,108-109,111-113,115-
132,135-137,139,142-
145,147-184,187-188,190-
198,200-234,236-238
METAPHOR function
description, 115
Prefix for METAPHOR function names, 4
RINTPERFORMABILITY, METAPHOR function,
4,50,115-
116,130-131,197
METAPHOR function
description, 116
RINTQUAD, abbreviation for labels in, 7
METAPHOR function,
4,7,47,51,81-
82,93,99,101-
102,104,106,108-
109,115,117,123-
128,135,137,139,142,144-
145,187,194-197,208,218-
220,223,225-
226,229,232,234-235,238
METAPHOR function
description, 117
ROB, abbreviation for PROBABILITY, 5
robability, abbreviation for, 5
Q, description of METAPHOR variable, 51
REPRESENTATION, APL function, 33
ROUTINE, description of METAPHOR variable, 51
SHAPE, description of METAPHOR variable, 51
STATESPERPHASE, description of METAPHOR variable, 51
METAPHOR variable, 97-
99,101,103-104
SWITCH, description of METAPHOR variable, 51
TRAJ, abbreviation for TRAJECTORY, 5
Trajectory, abbreviation for, 5
Trajectory set, 2
Translations, interlevel, 2
TRI, abbreviation for TRINARY, 5
Trinary, abbreviation for, 5
Tutorial facilities, 2
User assistance commands, 14
control and information exchange techniques for, 14
V, definition flag for, 19
description of METAPHOR variable, 51
Wu, L. T., 1
Z, description of METAPHOR variable, 51
parameter, 7