

AD-A101 063

PENNSYLVANIA STATE UNIV UNIVERSITY PARK DEPT OF INDU--ETC F/6 9/2
A GENERALIZED COMPUTER SIMULATION LANGUAGE FOR NAVAL SYSTEMS MO--FTC(11)
JUN 81 W E BILES, A NOZARI

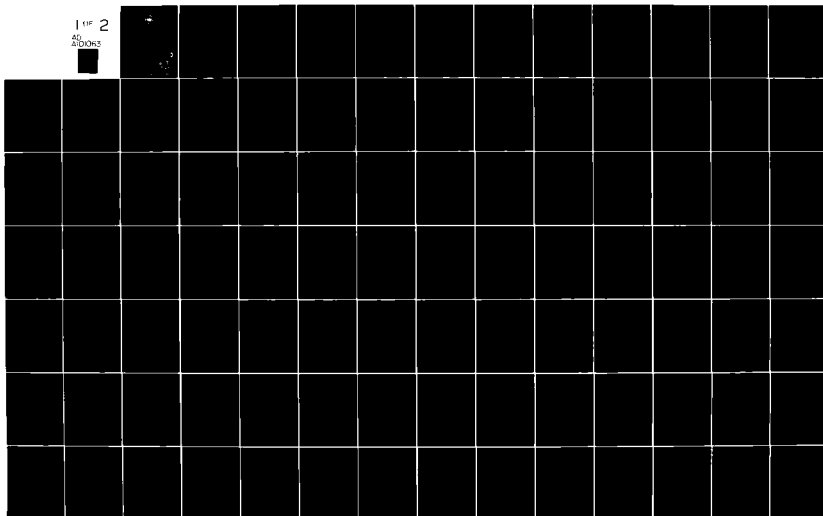
N00014-79-C-0757

NL

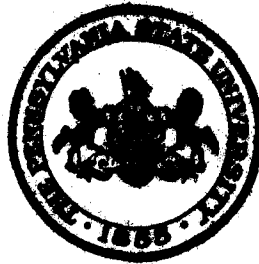
UNCLASSIFIED

1 of 2

AD
A10063



LEVEL I - ②



A

AD A101063

DTIC FILE COPY

DTIC
ELECTE
JUL 7 1981

A

THE PENNSYLVANIA STATE UNIVERSITY

College of Engineering

Department of Industrial and
Management Systems Engineering

207 Monroeville Building

University Park, PA 16802

This document has been approved
for public release and sale in
unlimited quantities.

81 7 06 144

FINAL REPORT

to the

NAVAL ANALYSIS PROGRAM
OFFICE OF NAVAL RESEARCH

ENTITLED

A GENERALIZED COMPUTER SIMULATION LANGUAGE
FOR NAVAL SYSTEMS MODELING (U)

by

Dr. William E. Biles
Principal Investigator

and

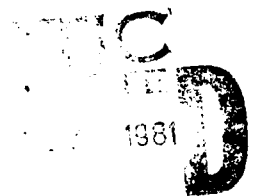
Mr. Ardavan Nozari
Graduate Assistant

Contract N00014-79-C-0757

Task NR277-284

Reproduction in whole or in part is permitted for any purpose of the
United States Government.

Approved for public release; distribution unlimited.



A

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A101 063	
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
A Generalized Computer Simulation Language for Naval Systems Modeling.	Final Report, Aug 1979 - Jun 1981	
7. AUTHOR(s)	6. PERFORMING ORG. REPORT NUMBER	
Dr. William E. Biles and Ardavan/Nozari		
9. PERFORMING ORGANIZATION NAME AND ADDRESS	8. CONTRACT OR GRANT NUMBER(s)	
The Pennsylvania State University 5 Old Main University Park, PA 16802	N00014-79-C-0757	
11. CONTROLLING OFFICE NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
Naval Analysis Program (Code 431) Office of Naval Research Arlington, VA 22217	61153N RR014-11-01 NR277-284	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE	
N/A	130 Jun 1981	
	13. NUMBER OF PAGES	
	117	
	15. SECURITY CLASS. (of this report)	
	Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
	N/A	
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
N/A		
18. SUPPLEMENTARY NOTES		
N/A		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Computer Simulation, Simulation Methodology		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
This report describes NAVMAP, a generalized computer simulation language for naval systems modeling. NAVMAP is designed to be highly flexible, compact and portable. It is a FORTRAN-based language for discrete-event, continuous and state-event modeling. It is designed to easily interface with FORTRAN-based software for statistical methodology and optimization. NAVMAP (Naval Modeling and Analysis Program) is intended to serve as the basis for a consistent simulation modeling approach among naval research laboratories.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

This report describes NAVMAP, a generalized computer simulation language for naval systems modeling. NAVMAP is designed to be highly flexible, compact and portable. It is a FORTRAN-based language for discrete-event, continuous and state-event modeling. It is designed to easily interface with FORTRAN-based software for statistical methodology and optimization. NAVMAP (Naval Modeling and Analysis Program) is intended to serve as the basis for a consistent simulation modeling approach among naval research laboratories.

Accession For	
MRCS	GRA&I
PLS TAB	
Unannounced	
Fortification	
Availability Codes	
Ext	and/or
Special	

A

CONTENTS

	<u>Page</u>
Abstract	1
Contents	ii
Acknowledgements	iii
Introduction	1
Research Objectives	3
Research Approach	4
NAVMAP	9
References	17
APPENDICES:	
Appendix A	33
Appendix B	42
Appendix C	69

ACKNOWLEDGEMENTS

The authors are pleased to acknowledge the able support of several colleagues in the performance of this research. Dr. C. Dennis Pegden was instrumental in the design of the filing structure for NAVMAP. Dr. E. Emory Enscoe assisted in the development of the statistical routines. Ms. Catherine Murphy and Mr. Eugene Yang were Graduate Research Assistants in the first year of this effort, and were helpful in the development of the random variate generation routines as well as the background literature survey. Finally, the skillful typing of Miss Mary Beth Schraf and Mrs. Martha Lowery are greatly appreciated.

INTRODUCTION

In recent years the Office of Naval Research has sponsored numerous industry and university-based research projects aimed at furthering the state-of-the-art of computer simulation methodology. The researchers performing that work are among the most distinguished in the field. The numerous high-quality journal and conference publications generated by that research attest to the highly productive efforts of those researchers and bring great credit to the sponsoring agency.

But these commendable research products have not been sufficiently incorporated into the many computer simulation modeling activities being carried on at various naval research laboratories. In general, naval research simulation programs would receive high "marks" for their programming structure and the extent to which they replicate the physical and operational characteristics of the real systems that they represent. But in many cases they fail to provide adequate program modules for such statistical methodology as variance reduction, statistics computation, the most up-to-date methods in random variate generation, screening experiments, experimental design, response surface methodology, optimization methods, and other proven statistical techniques in computer simulation. Efforts from one naval laboratory to another are very "uneven", and lack a consistent modeling structure and methodological approach.

This research has addressed the development of a simulation modeling language that provides a more consistent modeling structure and integrates the most up-to-date, proven statistical methodology and

optimization approaches. It has sought to "marry" the most applicable results from past and ongoing statistical methodology research with the most promising computer simulation modeling approaches. This "marriage" takes the form of a generalized computer simulation program that:

- Allows the simulation modeler to develop models of specific operations with minimum programming.
- Allows for statistical methodology features with simple program statements.
- Provides a more consistent modeling approach from one laboratory to the next, and thus allow portability of models as needed.

This report describes a generalized computer simulation language called NAVMAP (Naval Modeling and Analysis Program). This language is FORTRAN-based, so as to maximize its potential utility in naval laboratories. It enables discrete-event, continuous, and state-event computer simulation modeling in an event-oriented programming structure. It is designed to be compact and portable, with potential for implementation on minicomputers.

RESEARCH OBJECTIVES

This research will require three years in total, and involve the following tasks:

1. Collection and analysis of appropriate references on (a) statistical methodology and (b) simulation modeling languages that are needed for the research.
2. Design of a simulation modeling structure that incorporates continuous simulation, discrete-event simulation, statistical methodology and optimization techniques.
3. Coding and testing individual program modules.
4. Evaluation of the program package with selected naval simulation models.
5. Transmittal of the research products to naval laboratories.

The first two phases of the project, concluding on June 30, 1981 and covered in this report, essentially involves the first three tasks stated above.

RESEARCH APPROACH

This research project involved two main elements, in terms of a computer simulation modeling language for naval simulations. These two elements were (1) statistical methodology and (2) a simulation modeling and programming structure. The following sections discuss the principal issues embedded in each of these elements.

Statistical Methodology in Computer Simulation

A computer simulation model, particularly one involving stochastic elements, must provide for the following functional capabilities:

1. Random number generation
2. Random variate generation
3. Statistics collection and reporting
4. Variance reduction
5. Input analysis
6. Output analysis
7. Experimental design
8. Optimization

The first four of these functional capabilities are internal to the computer simulation, and must be made available within the simulation modeling structure. For example, in a naval simulation model such as SPEARS [28], it might be necessary to record the range at which "kills" of incoming enemy missiles occur. If we assume that an event MKILL gives rise to such a "kill", and that the kill distance XDISK is computed at the instant in which the "kill" occurs, a single statement can be used to record this value, such as follows:

```
CALL TALLY (4,XDISK)
```

This statement calls a subroutine TALLY which updates the statistics for variable 4, which the modeler has designated as XDISK, the missile kill distance. All statistics collection and summary operations would then be effected automatically through subroutine TALLY. Those statistical operations that are internal to the operation of the simulation model can be incorporated directly into the simulation modeling language, allowing the modeler to call upon powerful statistical capability with a mere handful of program statements. This research has investigated the most efficient techniques for accomplishing these internal statistics operations represented by functional capabilities 1 through 4.

Functional capabilities 5 through 8 above take the form of programs that lie external to the computer simulation model. For instance, "canned" programs would be prepared to allow "goodness-of-fit" testing to sample data, preparatory to having the modeler identify a distribution and parameter set for a specified random variable in the model. At least two statistical procedures, the Chi-Square test and the Kolmogorov-Smirnov test, will be coded in the next phase of research for the following probability distributions:

- | | |
|----------------|-----------------------|
| 1. Exponential | 7. Uniform |
| 2. Normal | 8. Poisson |
| 3. Lognormal | 9. Binomial |
| 4. Gamma | 10. Geometric |
| 5. Weibull | 11. Negative Binomial |
| 6. Beta | 12. Hypergeometric |

Similar "canned" programs will be prepared, or identified, for such statistical procedures as multiple regression, analysis of variance, and

analysis of covariance. These statistical programs must be able to interface with the simulation language described here. Programs for optimization of computer simulations, such as those by Biles [2] and Smith [51-53], will be modified to interface naturally with the simulation modeling structure described in this report.

Computer Simulation Modeling Structure

The development of a generalized computer simulation modeling structure for naval simulation has involved the following subtasks:

1. Identification of the common features of naval simulation models, through interviews with Navy laboratory personnel and analysis of existing naval simulation models (for example [1, 28]).
2. Selection of a base programming language that is compatible with the maximum number of naval laboratory computers (for example, FORTRAN 77).
3. Formulation of modeling approaches for
 - a. Continuous simulation
 - b. Discrete-event simulation
 - c. State-event simulation
 - d. Combined continuous/discrete-event/state-event simulation.
4. Formulation of the basic modeling structure for the overall language.
5. Coding the generalized simulation language.
6. Evaluation and testing of the language.

The crucial feature of simulation modeling structure is the adoption of a "view of the world" from which the modeler operates. Research to date in reviewing naval simulation modeling activity has revealed two major simulation approaches in use by naval personnel: (a) digital simulation, in which the entire simulation logic is represented via a computer program; and (b) real-time simulation, in which there is physical hardware, such as a torpedo or an aircraft, in the loop. This research has focused on digital computer simulation, although some of the software that lies external to the simulation model might very well be applicable with real-time simulation.

A second aspect of the simulation modeling structure is discrete simulation versus continuous simulation. Discrete simulation occurs when the dependent variables change discretely at specified points in simulated time; for example, when a mine detonates, the number of mines in a minefield is decremented by one unit. In continuous simulation the dependent variables may change continuously over simulated time; for instance, the position of a torpedo in a three-dimensional space representing the ocean. Naval simulation modeling definitely requires both discrete and continuous simulation. The generalized simulation language developed during this research project affords both discrete and continuous simulation capability.

A third aspect of the simulation modeling structure's "view of the world" is whether the discrete simulation adopts (a) an event orientation, (b) an activity scanning orientation, or (c) a process orientation.

The NAVMAP language described in this report adopts an event orientation for the discrete simulation. In this orientation, events

occur at instants in time. Events alter the state of the system, and these state changes must be so recorded and the appropriate statistics collected. Thus, the modeler has the task of preparing FORTRAN sub-routines which execute the logical structure of the events which describe the behavior of the system being modeled. NAVMAP automatically calendars the events in proper sequence, and advances time from one event to the next. A NAVMAP model of a given naval system would typically consist of a MAIN program and several event subroutines. The language is described in the ensuing sections.

NAVMAP

The current status of this project, after almost two years of effort under Contract N00014-79-C-0757, is that a combined continuous/discrete-event/state-event simulation language called NAVMAP (Naval Modeling and Analysis Program) has been coded in FORTRAN and evaluated. The objective was that this simulation language should be highly flexible, compact, and portable. Its flexibility is verified by the fact that it can be employed in either discrete-event simulation, continuous simulation, state-event simulation, or a combination of these. Likewise, it can be interfaced with FORTRAN-based statistical and optimization programs. The compactness of this naval systems simulation language is evident in the fact that it contains only 2600 lines of code, far less than the GASP-IV language to which it is comparable in flexibility and capability. The probability of the language lies in the fact that it is FORTRAN-based and compact, which makes it usable on any computer with a FORTRAN compiler, including most minicomputers.

Elements of Design

In the design of NAVMAP, several objectives have been considered. Primary of those are: (1) flexibility for the interfacing with statistical and optimization procedures, (2) portability, and (3) compactness. In this section, each of these goals and the steps for achieving them are discussed separately. Finally, in the last part of this section, the data structure of the language is put into perspective and its features are described.

Interfacing Flexibility

Most of the simulation languages, like SLAM [41] and GASP [40], provide a summary report at the end of each simulation run. One could

obtain the mean, standard deviation, and other statistical measures on each of a number of model variables, as well as histograms and plots of continuous variables. However, more often than not these results are useless with respect to statistical analysis. To make this point clear, let us proceed with a very simple example.

Consider an M/M/1 queueing system, and suppose it is desired to estimate the steady state mean waiting time of the customers. To accomplish this, observation is made on the waiting time of each customer entering the system and the time he completes the service. It is well known that this observation is correlated and therefore the standard deviation provided in the statistical summary report is useless. To obtain histograms, one must specify the parameters of the histogram prior to the run. Now suppose one has made an expensive simulation run to obtain a histogram, and after the run realizes that the histogram is not well shaped because of ill specification of the parameters. To obtain a new histogram, one must completely replicate the expensive simulation run. This could also be true when a plot of the values of a variable is to be obtained.

In the event of correlated observations, one might want to try the batch means procedure [30-34], where one needs to have access to all the observations produced during the simulation. Even in the situation where more advanced statistical analysis is needed, such as regression or analysis of variance, access to individual observation might be required. The issue illustrated here is that one essentially needs to have access to individual observations in order to minimize number of runs for analysis. We take the view that the statistical analysis

and inferences about the unknown parameters of the problem under study must be a separate module from the simulation model. That is, the sole purpose of the simulation must be collection of observations on variables, and the job of analysis and preparation of data must be assigned to other modules. Obviously, the existing simulation languages have shortcomings in this regard. In NAVMAP, we have achieved this goal by letting the user specify a unit number for each of the statistics. This results in recording each individual observation on that unit.

There are three types of statistics collection devices in NAVMAP. The particular type employed in a given instance depends on the nature of the model variable for which statistics are being collected. The first type of statistic is a Tally statistic, which collects information on an observation of a non-time dependent random variable. A random variable is considered non-time dependent when its value at a precise instant in simulated time is recorded without regard either to the length of time that value has persisted or the value it has changed from. For example, if one is collecting observations on the "kill distance" XDISK at which an incoming enemy missile is intercepted, this value is recorded at the instant in time the "kill" occurs, which is marked by the event MKILL. In the simulation of an engagement, observations of XDISK will be recorded each time a kill event occurs. If XDISK is the fourth variable, the observation would be recorded with the statement in MKILL:

```
CALL TALLY (4,XDISK)
```

NAVMAP automatically computes the minimum, maximum, arithmetic mean, standard deviation and number of observations of each Tally variable when one of the functions TMIN, TMAX, TAVG, TSTD or TNUM is accessed.

This information is automatically computed and printed at the conclusion of the simulation run. After all simulation runs have been executed, the user can access the accumulated observations by reading the data from the selected storage unit identified on the Tally data card (see Appendix A). The symbol ** is used to separate data between simulation runs. These observations are recorded unformatted.

Discrete time persistent statistics are employed for observing the values of either XX variables or number of entities of a file as they change and recording their value and the time of their change. If the unit number is specified, the value and the time of the change, respectively, would be dumped on that unit. Again, these are written unformatted and ** is used to separate observations of each run. DMIN, DMAX, DAVG, DSTD, and DPRD are functions that respectively give minimum, maximum, average, standard deviation, and the length of the period of the observations once they are called.

Continuous time persistent statistics do behave exactly as the discrete time persistent except they are for collecting observations on SS, DD variables (continuous variables). Statistical collection can be truncated by several means. Subroutine CLEARS clears all the statistical arrays once it is called. Sometimes it is desirable to clear some variables at some specified time and clear others at some other time. This can be achieved by calling subroutines TRUND(N), TRUNC(N), and TRUNT(N). One can also clear the statistical arrays at a pre-determined time by specifying this time on the Initialization Card. Another means of clearing arrays is to judiciously discard some observation according to one of several proposed methods (for example, Schruben[48]) once the simulation runs are completed.

Portability

Simulation languages which feature real valued attributes like GASP IV[40] and SLAM [41], although claimed, are not totally portable. The problem arises because of the data structure for maintaining the entities. Each entity consists of some integer information, such as pointers, and some real information, such as attributes. In these languages there are two large arrays that are set EQUIVALANCE: one is for maintaining integer information and the other for maintaining the real information associated with each file entry.

On some computers the number of integer words per number of real words is not 1. That is, the real word is comprised of 4 bytes and the integer word is comprised of 2 bytes. In these machines, a problem arises if two arrays, one integer and one real, of the same length are set EQUIVALANCE. To illustrate this difficulty, suppose the integer array is ISET(12) and the real array is RSET(12). Then as Figure 1 shows, location 2 of ISET does not correspond to location 2 of RSET, and location 7 of RSET does not correspond to any location in ISET.

Now let NIR = number of integer words per real word

MSET = dimension of ISET; a multiple of NIR

Then clearly the dimension of RSET must be $\frac{MSET}{NIR}$. If L is an integer address, the corresponding real address should be $\frac{L+NIR-1}{NIR}$. For the above example: NIR = 2, MSET = 10, and the relationship between the integer and real address is clear from Figure 2.

The problem can be resolved by allocating the upper part of the array to integer values and the lower part of the array to the real values and have a pointer in the integer part to give the address of

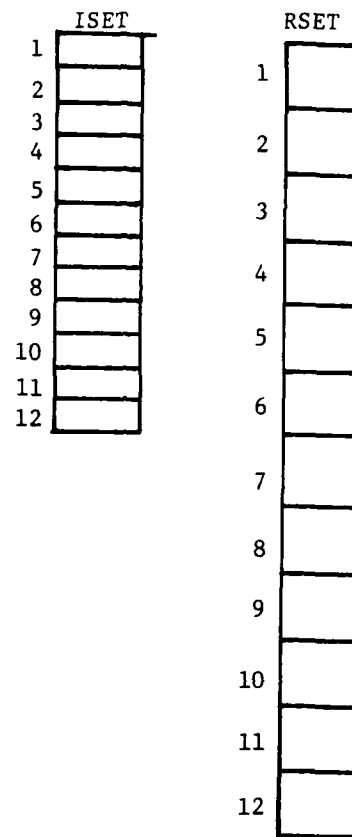


Figure 1. Filing Structure in
GASP-IV and SLAM

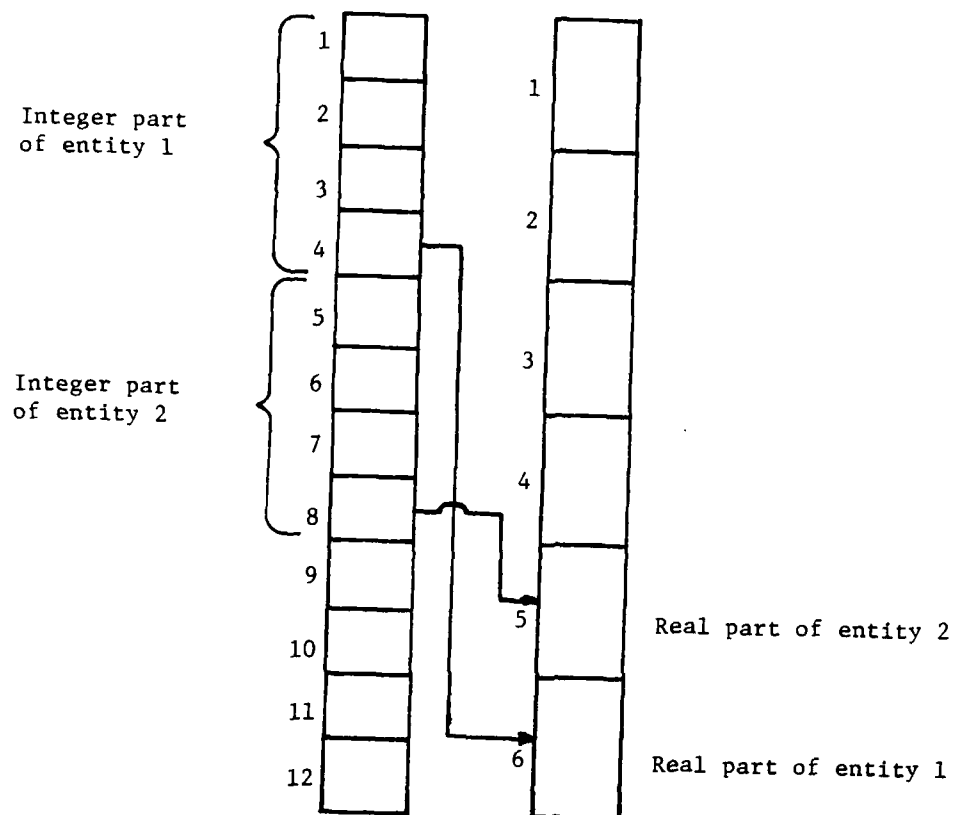


Figure 2. Filing Structure in NAVMAP

the corresponding real part. To illustrate this concept, suppose every entity is comprised of 4 integer variables and 1 real variable, where the last information in the integer part is the pointer to the location of the real part. Then as Figure 2 indicates the problem is alleviated, thus affording a simple basis for designing a completely portable filing structure.

Compactness

In a given simulation model one might or might not need files, tallies, discrete time persistent statistics, continuous time persistent statistics, and counters. In existing simulation languages fixed amount of storage are allocated for all of the above regardless of whether they are used. We take the view that this procedure is extravagant with respect to storage space.

We observe that each of the above data elements integer part information and real part information. We can organize the integer information in one block and the real information in another block, and maintain a pointer to relate the integer part to real part. If we keep a pointer to the location of the first integer block of one type (say files), we can have a dynamic data structure for that type of data element. In the following paragraphs, we define the integer and real information for each type of data element and the manner that they have been organized in NAVMAP.

(a) Entries

<u>Integer</u>	<u>Real</u>
1. Location of predecessor	1. Ranking value
2. Location of successor	2. Attribute 1
3. Event Code	3. Attribute 2
4. Pointer to attributes	.
	.
	N+1. Attribute N

(b) Files

<u>Integer</u>
1. Number of entries in the file
2. Location of first entry
3. Location of last entry
4. Ranking rule
5. Ranking attribute
6. Location of the corresponding discrete time persistent statistics block

(c) Tally

<u>Integer</u>	<u>Real</u>
1. Location of statistical block	1. ΣX
2. Label	2. ΣX^2
3. Output device number	3. Number of observation
	4. Min.
	5. Max.

(d) Discrete time persistent statistics

<u>Integer</u>	<u>Real</u>
1. Location of statistical block	1. $\int X dt$
2. Label	2. $\int X^2 dt$
3. Variable Type	3. XLAST
4. Output device number	4. MIN
	5. MAX
	6. TLAST
	7. TCLEAR

(e) Continuous time persistent statistics

<u>Integer</u>	<u>Real</u>
1. Location of statistical block	1. $\int X dt$
2. Label	2. $\int X^2 dt$
3. Variable type	3. XLAST
4. Output device number	4. MIN
	5. MAX
	6. TLAST
	7. TCLEAR

(f) Counters

<u>Integer</u>
1. Current count
2. Label
3. Limit

This structural feature of NAVMAP enables a highly compact storage of a simulation model.

Organization of NAVMAP

NAVMAP is organized in much the same way as GASP-IV[40], but with quite different subprograms. Figure 3 shows the organization of NAVMAP. Table 1 gives the subroutines used for executive control and statistics collection. Table 2 shows the function subprograms employed for statistics collection. Table 3 presents the random process generation functions. Table 4 lists the key NAVMAP variables.

Description of Routines

1) Main

The user has the choice to write his own main program or use the default program provided in NAVMAP. In this routine, the length of ISET and RSET are set and four variables NCRDR, NPRTR, LFI, and LLR are initialized. Subroutine SOAP is then called. LFI must always be set equal to 1. LLR must always be set equal to the length of RSET plus 1. NCRDR, and NPRTR are respectively number of card reader unit and printer unit at the computer installation. A typical main program is given below. Note that the common blocks are to be written exactly as they are in the sample main program.

```

DIMENSION RSET(5000)
COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
COMMON ISET(5000)
COMMON/GSCI/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
-DDL(99),TNOW,XX(99),DTNOW,ISTOP
EQUIVALENCE (ISET(1),RSET(1))
NCRDR=5
NPRTR=6
LFI=1
LLR=5001
CALL SOAP
STOP
END

```

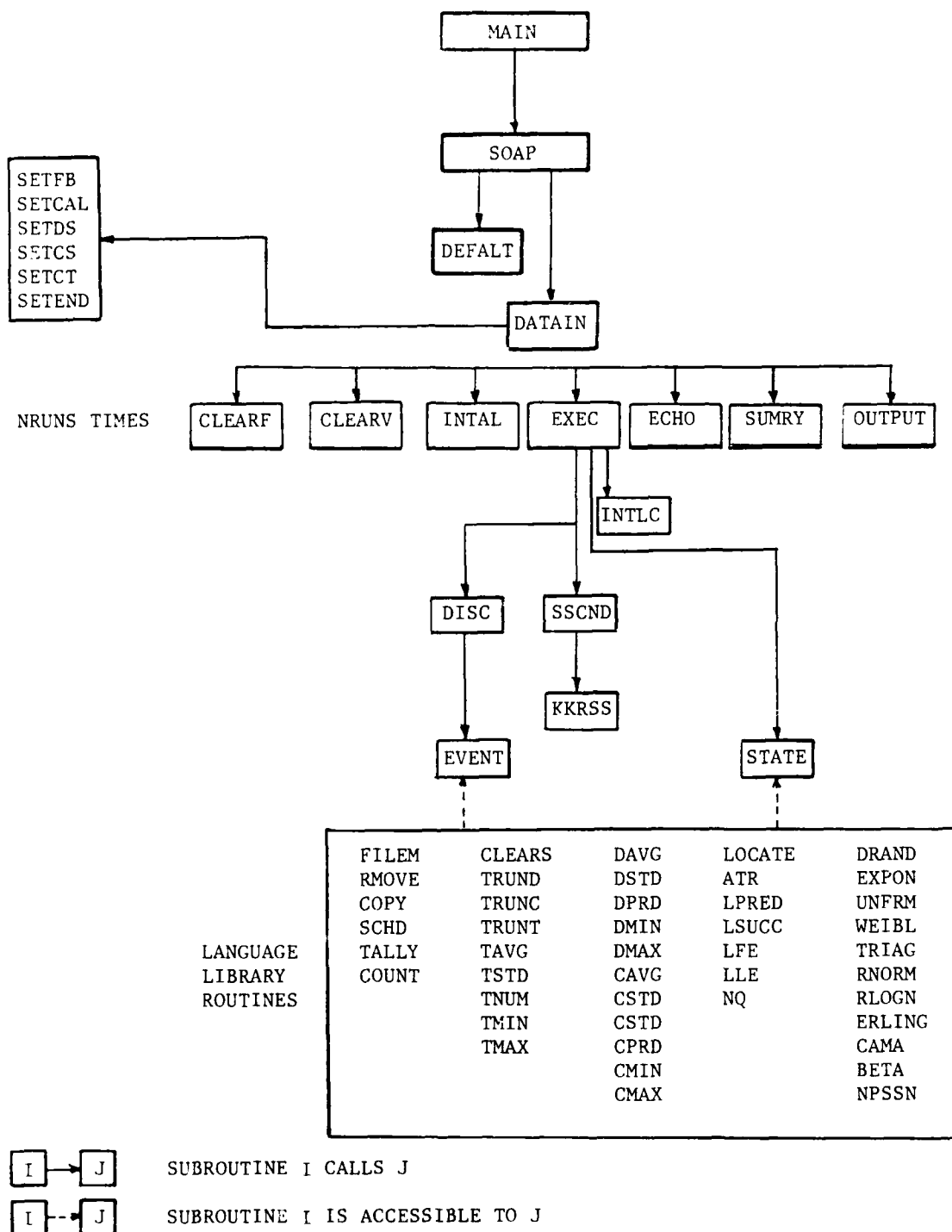


Figure 3. Organization of the NAVMAP Language

Subroutine	Description
FILEM (IFILE, A)	Files the entity with attributes in array A in File IFILE.
RMOVE (I, IFILE, A)	Removes Ith entry of the file IFILE and loads its attributes into array A.
COPY (I, IFILE, A)	Copies the attributes of the Ith entry of file IFILE into array A.
LOCATE (I, IFILE, LENT)	Gives the location of the Ith entry of file IFILE in LENT.
SCHD (IEVENT, TIME, A)	Schedules event IEVENT at time TIME, with attributes stored in vector A.
TALLY (N, VAR)	Collects statistics on Tally #N using VAR.
COUNT (N, INC)	Increments counter #N with INC.
CLEARs	Clears statistical arrays, at TNOW.
TRUND(N)	Truncates discrete time persistent statistics #N at TNOW.
TRUNC(N)	Clears continuous time persistent statistics #N at TNOW.
TRUNT(N)	Clears Tally #N statistics at TNOW.
INTLC	User written for setting initial conditions of the simulation.
EVENT(I)	User written for defining the events.
OUTPUT	User written for special output preparation or any adhoc procedure. It is called after the end of each simulation run.
STATE	User written subroutine for defining the differential and difference equations.

Table 1. Simulation Executive Subroutines

Functions	Description
TAVG(N), TSTD(N), TNUM(N), TMIN(N), TMAX(N)	Gives average, standard deviation, number of observations, minimum, and maximum of Tally #N.
DAVG(N), DSTD(N), DPRD(N), DMIN(N), DMAX(N)	Gives average, standard deviation, length of period of statistical collection, minimum, and maximum of discrete statistical time persistent #N.
CAVG(N), CSTD(N), CPRD(N), CMIN(N), CMAX(N)	Gives average, standard deviation length of period of statistical collection, minimum, and maximum of continuous statistical time persis- tent #N.
ATR(LENT, I)	Gives the value of the Ith attribute of the entry located at LENT.
LPRED (LENT), LSUCC (ENT)	Gives the predecessor and successor of the entry located at LENT.
NQ (IFILE), LFE (IFILE), LLE (IFILE)	Gives the number of entries, loca- tion of first entry, and location of the last entry of file IFILE.

Table 2. Statistics Collection and Reporting Functions

Functions	Description
DRAND(IS)	A pseudo-random number obtained from random number stream IS.
EXPON(XMEAN, IS)	A sample from an exponential distribution with mean XMEAN using random number stream IS.
UNFRM(ULO,UHI, IS)	A sample from a uniform distribution in the interval ULO to UHI using random number stream IS.
WEIBL(BETA,ALPHA, IS)	A sample from a Weibull distribution with scale parameter BETA and shape parameter ALPHA using random number stream IS.
TRIAG(XLO,XMODE,XHI, IS)	A sample from a triangular distribution in the interval XLO to XHI with mode XMODE using random number stream IS.
RNORM(XMN,STD, IS)	A sample from a normal distribution with mean XMN and standard deviation STD using random number stream IS.
RLOGN(XMN,STD, IS)	A sample from a lognormal distribution with mean XMN and standard deviation STD using random number stream IS.
ERLNG(EMN,XK, IS)	A sample from a Erlang distribution which is the sum of XK exponential samples each with mean EMN using random number stream IS.
GAMA(BETA,ALPHA, IS)	A sample from a gamma distribution with parameters BETA and ALPHA using random number stream IS.
BETA(THETA,PHI, IS)	A sample from a beta distribution with parameters THETA and PHI using random number stream IS.
NPSSN(XMN, IS)	A sample from a Poisson distribution with mean XMN using random number stream IS.

Table 3. Random Process Generation Functions

User Variables	Description
NCRDR	Installation card reader unit #
NPRTR	Installation line printer unit #
TNOW	Current time
DTNOW	The step size
ATTRIB(I)	Attribute I of current entity
SS(I)	Value of state variable I at TNOW
SSL(I)	Value of state variable I at TLAST
DD(I)	Value of the derivative of state variable I at TNOW
DDL(I)	Value of the derivative of state variable I at TLAST
XX(I)	Value of global variable I
JJ	An integer global variable
ISTOP	An integer variable that terminates simulation when it is equal to 1

Table 4. Key Simulation Language Variables

2) SOAP

SOAP is the executive routine in NAVMAP.

3) DEFAULT

DEFAULT is used for setting the default values for NAVMAP variables.

4) DATAIN

DATAIN is used for reading the input cards and printing data in the way they are read.

5) SETFB, SETCAL, SETDS, SETCS, SETCT and SETENP

These subroutines are employed for setting up, respectively, a file block, the calendar, the discrete time-persistent block, the continuous time persistent block, a counter, and the entry pool.

6) CLEARF, CLEARV AND INTAL

These subroutines are used for clearing files and variables, and setting up some of the initial conditions prior to each run.

7) EXEC

EXEC is used for the time advance mechanism and for solving differential and difference equations.

8) INTLC

INTLC is a user written subroutine for setting up the initial conditions of each run.

9) DISC and EVENT

These are used for processing events. EVENT is a user written subroutine.

10) SSCND and KKRSS

These subroutines are related to state-events and for determining whether or not a state-event has occurred within the specified tolerance.

11) STATE

Subroutine STATE is developed by the user for specifying the differential and difference equations which describe the continuous component of the model.

12) ECHO and SUMRY

These subroutines are used respectively for printing the Echo and Summary Reports.

13) OUTPUT

This is a user written routine used for special treatment of data at the end of each simulation run.

14) Language Library Routines

Any of the library function subprograms available on a FORTRAN compiler are accessible in NAVMAP.

Data input to NAVMAP consists of punched cards. The input data is arranged in thirteen card types, depending on the particular simulation function that the data supports or initializes. Appendix A describes the data formats for these thirteen types of cards.

As stated above, NAVMAP possesses the capability for discrete-event, continuous, and state-event simulation. Appendix B gives the

data input, user program listing, and simulation output for three relatively simple problems that demonstrate the three modes of simulation capability. Problem 1 is a simple M/M/1 queue; that is, a single-channel queueing problem with Poisson arrivals and exponential service times. The first page of the printout shows the data input cards. The next two pages show the user-written FORTRAN subroutines that describe the M/M/1 queueing system. The two event subroutines in this discrete-event model are ARVL and ESRV, which correspond to the arrival of an entity into the system and the completion of service, respectively. Next is shown the printout of the input data, which provides the analyst a check on the verity of his input to the model. Then two pages give the Echo Report, which shows all data and system conditions at the outset of the execution of the model. Finally, a Summary Report gives the final results of the simulation.

Problem 2 in Appendix B gives input data, user subprograms and output results for a sample continuous simulation problem, the Pilot Ejection Problem from Pritsker and Pegden[41]. Key features here are the initialization of state variables $SS(I)$, $I=1, \dots, 4$ in Subroutine INTLC and representation of the state equations in Subroutine STATE. These equations model the behavior of a pilot ejection module (pilot and seat) which is thrown from a crippled aircraft. The simplicity of the user requirements for a continuous simulation model is illustrated here.

Problem 3 in Appendix B gives input data, user subprograms and output results for a sample continuous/state-event simulation, the Cedar Bog Lake Problem from Pritsker and Pegden[41].

Appendix C gives a complete listing of NAVMAP. The 2607 lines of program code represent a significant compression from that of the GASP-IV language[40] to which it is comparable in capability.

Future efforts will concentrate on three areas:

1. The extension of NAVMAP to include network modeling.
2. The development of FORTRAN-based statistical and optimization programs that would enable preliminary data analysis, regression, correlation, analysis of variance and response surface analysis.
3. Demonstration of NAVMAP with realistic naval systems, including the following:
 - a. Undersea systems vis a vis the NUSC laboratories at Newport, RI and New London, CT.
 - b. Minefield systems vis a vis the ARL at The Pennsylvania State University and the Naval Weapons Laboratory at Dahlgren, VA.

Finally, NAVMAP will be completely documented and transmitted to selected naval laboratories in the form of magnetic tapes.

1. Bailey, J.L., and M. A. Weedon, "The Minefield Analysis with Hunting Evaluation Model (MAYHEM)", NWL Technical Report TR-2742, U.S. Naval Weapons Laboratory, Dahlgren, Virginia, May 1972.
2. Biles, W. E., "Optimization of Multiple-Response Simulation Models", Final Report to the Office of Naval Research, Contract N00014-76-C-1021, NR 277-237, University of Notre Dame, October 1978.
3. Biles, W. E., "A Gradient-Regression Search Procedure for Simulation Experimentation", Proceedings of the 1974 Winter Simulation Conference, Washington, D.C., 1974, p. 491-497.
4. Biles, W. E., and J. J. Swain, "Strategies for Optimization of Multiple Response Simulation Models", Proceedings of the 1977 Winter Simulation Conference, Gaithersburg, Maryland, December 1977.
5. Biles, W. E., and J. J. Swain, "Mathematical Programming and the Optimization of Computer Simulations", Mathematical Programming Study in Engineering Optimization, North-Holland Publishing Company, Amsterdam, Netherlands, 1979.
6. Biles, W. E. and J. J. Swain, Optimization and Industrial Experimentation, Wiley-Interscience, New York, 1979.
7. Box, G. E. P., and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions", Journal of the Royal Statistical Society, B, Vol. 13, 1951, p. 1-45.
8. Box, G. E. P., and J. S. Hunter, "Multifactor Experimental Designs for Exploring Response Surfaces", Annals of Mathematical Statistics, Vol. 28, 1957, p. 242.
9. Box, J. J., "A New Method of Constrained Optimization and Comparison with other Methods", Computer Journal, Vol. 8, 1965.
10. Brooks, S. H., and M. R. Mickey, "Optimization Estimation of Gradient Direction in Steepest Ascent Experiments", Biometrics, Vol. 17, No. 1, 1961.
11. Carter, G. and E. J. Ignall, "A Variance Reduction Technique for Simulation", Management Science, Vol. 21, 1975, pp. 607-616.
12. Conway, R., "Some Tactical Problems in Digital Simulation", Management Science, Vol. 10, 1963, pp. 47-61.
13. Crane, M. A. and D. L. Iglehart, "Simulating Stable Stochastic Systems I: General Multiserver Queues", J. ACM, Vol. 21, 1974, pp. 103-113.
14. Crane, M. A. and A. Lemoine, An Introduction to the Regenerative Method for Simulation Analysis, Technical Report No. 86-23, California Analysis Corporation, Palo Alto, CA, October 1976.
15. Draper, N. R., and H. Smith, Applied Regression Analysis, John Wiley and Sons, New York, 1966.

16. Dudewicz, E. J., "New Procedures for Selection Among (Simulated) Alternatives", Proceedings, 1977 Winter Simulation Conference, 1977, pp. 58-62.
17. Duket, S. and A. A. B. Pritsker, "Examination of Simulation Output Using Spectral Methods", Mathematics and Computers in Simulation, Vol. XX, 1978, pp. 53-60
18. Farrell, W., "Literature Review and Bibliography of Simulation Optimization", Proceedings, 1977 Winter Simulation Conference, 1977, pp. 116-124.
19. Fishman, G. S., "A Study of Bias Considerations in Simulation Experiments", Operations Research, Vol. 20, 1972, pp. 785-790.
20. Fishman, G. S., Concepts and Methods in Discrete Event Digital Simulation, John Wiley, 1973.
21. Fishman, G. S., "Grouping Observations in Digital Simulation", Management Science, Vol. 24, 1978, pp. 510-521.
22. Fishman, G. S., Principles of Discrete Event Simulation, John Wiley, 1978.
23. Fishman, G. S., "Statistical Analysis for Queueing Simulation", Management Science, Vol. 20, 1973, pp. 363-369.
24. Fishman, G. S., "The Allocation of Computer Time in Company Simulation Experiments", Operations Research, Vol. 16, 1968, pp. 280-295/
25. Hunter, J. S., and T. H. Naylor, "Experimental Designs for Computer Simulation Experiments", Management Science, Vol. 16, No. 5, 1970, pp. 422-434.
26. Ignall, E. J., "On Experimental Designs for Computer Simulation Experiments", Management Science, Vol. 18, No. 7, March 1972, pp. 384-388.
27. Kabak, I. W., "Stopping Rules for Queueing Simulations", Operations Research, Vol. 16, 1968, pp. 431-437.
28. Kaplan, D. J., L. C. Davis, M. E. B. Owens II, O. F. Forsyth and J. J. Penick, "SPEARS, an AAW Performance Simulation", NRL Report 7958, Naval Research Laboratory, Washington, D. C., June 1976.
29. Kleijnen, J. P. C., Statistical Techniques in Simulation, Parts I and II, Marcel Dekker, New York, 1975.
30. Law, A. M., Confidence Intervals for Steady-State Simulations, I: A Survey of Fixed Sample Size Procedures, Technical Report 78-5, University of Wisconsin, 1978.

31. Law, A. M., "A Comparison of Two Techniques for Determining the Accuracy of Simulation Output", Research Report (Department of Industrial Engineering, University of Wisconsin - Madison), June, 1975.
32. Law, A. M., "Statistical Analysis of the Output Data from Terminating Simulations", Research Report (Department of Industrial Engineering, University of Wisconsin - Madison), April, 1978.
33. Law, A. M., Carson, J. S., "A Sequential Procedure for Determining the Length of a Steady-State Simulation", Research Report (Dept. of Ind. Engr., University of Wisconsin - Madison), April, 1977.
34. Law, A. M., "Validation of Simulation Models, I: An Overview and Survey of Real-World Practice", Research Report (Department of Industrial Engineering, University of Wisconsin-Madison), August, 1979.
35. Montgomery, D. C., Design and Analysis of Experiments, John Wiley and Sons, New York, 1976.
36. Montgomery, D. C., and V. M. Bettencourt, "Multiple Response Surface Methods in Computer Simulation", Simulation, October, 1977, pp. 113-121.
37. Montgomery, D. C., "Methods for Factor Screening in Computer Simulation Experiments," Research Report on ONR Contract N0014-78-C-0312, Georgia Institute of Technology, Atlanta, GA, March 1975.
38. Pritsker, A. A. B., "GASP", Enclopedia of Computer Science and Technology, J. Belzer, A. G. Holzman, and A. Kent, Editors, Vol. 8, Marcel Dekker, Inc., 1977.
39. Pritsker, A. A. B., Modeling and Analysis Using Q-GERT Networks, Halsted Press and Pritsker & Associates, Inc., 1977.
40. Pritsker, A. A. B., The GASP IV Simulation Language, John Wiley, 1974.
41. Pritsker, A. A. B., and C. D. Pegden, Introduction to Simulation and SLAM, Halsted Press, New York, 1979.
42. Russell, E. C., Simulating with Processes and Resources in SIMSCRIPT II. 5, CACI, Inc., Arlington, VA., 1974.
43. Sabuda, J., F. H. Grant, III and A. A. B. Pritsker, The GASP IV/E User's Manual, Pritsker & Associates, Inc., West Lafayette, IN, 1978.
44. Schmeiser, B. W., Babu, A. J. G., "Beta Variate Generation via Exponential Majorizing Functions" Research Report (Department of Operations Research and Engineering Management, Southern Methodist University), December, 1978.
45. Schmeiser, B. W., Lal, R., "Another Versatile Family of Probability Distributions" Research Report (Department of Operations Research and Engineering Management, Southern Methodist University), December, 1978.
46. Schmeiser, B. W., "Methods for Modeling and Generating Probabilistic Components in Digital Computer Simulation when the Standard Distributions are not Adequate: A Survey" (Department of Operations Research and Engineering Management, Southern Methodist University), December 1977.

47. Schmeiser, B. W., Lal, R., "Computer Generation of Bivariate Gamma Random Vectors" Research Report, (Department of Operations Research, Southern Methodist University), 1979.
48. Schruben, L. W., "A Coverage Function for Interval Estimates, "Research Report (School of Operations Research and Industrial Engineering, Cornell University), February 1979.
49. Schruben, L. W., "Controlling Initialization Bias in Simulation Experiments", Research Report (School of Operations Research and Industrial Engineering, Cornell University), April, 1979.
50. Schruben, L. W., Margolin, B. H., "Pseudo-random Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments", Journal of the American Statistical Association, Vol. 73, No. 363, 1978, pp. 504-520.
51. Smith, D. E., "Requirements of an Optimizer for Computer Simulations", Naval Research Logistics Quarterly, Vol. 20, No. 1, 1973.
52. Smith, D. E., "An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation", Operations Research, Vol. 21, 1973, pp. 475-497.
53. Smith, D. E., "Automated Response Surface Methodology in Digital Computer Simulation," Volumes I and II, Research Report, ONR Contract N00014-74-C-0148, State College, PA, September 1975.
54. Welsh, J. and J. Elder, Introduction to PASCAL, Prentice-Hall, International, London, 1979.

APPENDIX A

DESCRIPTION OF DATA INPUT FORMAT
FOR NAVAL SYSTEMS SIMULATION PROGRAM

Input Cards

1. Project Card

Title of the project, name of the analyst, and the date to appear on the summary report are specified on this card.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	PROJ	2A2
10 - 39	Title of the project	15A2
40 - 59	Analyst name	10A2
60 - 69	Date	5A2

2. Discrete Card

Number of the required files in the model, whether a discrete model or not, and maximum number of attributes per entity are specified on this card.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	DISC	
10 - 11	Number of files	I2
20	{ 1 we have discrete 0 we do not have discrete	I1
30 - 31	Maximum number of attributes per entity	I2

3. Rank Card

The ranking discipline of each of the files are specified on this card. For every file there should be one card.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	RANK	
10 - 11	File number	I2
20	Ranking discipline 1 FIFO 2 LIFO 3 HVF 4 LVF	I1
30 - 31	Number of ranking attribute	I2

Example:

- To specify that file 5 is ranked FIFO we have:

Cols	1	10	20
	↓	↓	↓
	RANK	05	1

2. To specify that file 11 is ranked HVF (High Value First) based on attribute 6 we have:

Cols	1	10	20	30
	↓	↓	↓	↓
RANK	11	3	06	

4. Continuous Card

The information on the continuous part of the model, if any, is provided through this card. If the model does not have a continuous part, this card must still be used but all the parameters are left blank.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	CONT	
10 - 11	No. of Differential Equations (NEQD)	I2
15 - 16	No. of Difference Equations (NEQS)	I2
20 - 21	No. of state Events (NSEV)	I2
30 - 39	Minimum step size (DTMIN)	F10.5
40 - 49	Maximum step size (DTMAX)	F10.5
50 - 59	Absolute Error (AERR)	F10.5
60 - 69	Relative Error (RERR)	F10.5
70	Indicates type of error check in Runge-Kutta integration or in state event crossing detection when a step size smaller than DTMIN is required. If F is specified, a fatal error occurs. If W is specified, a warning message is printed before proceeding. If N is specified, execution proceeds with no warning message given. The default value, that is if the field is left blank, is W.	A1

The numerical integration accuracy is controlled by the specification of AERR and RERR. The Runge-Kutta-Fehlberg integration algorithm used in the language estimates the single step error for each variable defined by a differential equation. The Ith error estimate is compared to TERR where

$$TERR(I) = AERR + ABS(SS(I)) \times RERR.$$

If the error estimate is less than or equal to $TERR(I)$ for each I , the values of $SS(I)$ are accepted. If not, the step size is reduced and the integration algorithm is reapplied. There are no default values for $AERR$ and $RERR$ and they must be specified by the user. The stringent values for these could substantially increase the running times, although they result in better accuracy.

5. State-event Card

The information on state events is provided on this card.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	SEVN	
10 - 11	Event number	I2
20 - 23	Crossing variables +1 $SS(i)$ -1 $DD(i)$	I2
30 - 31	Direction of crossing +1 positive direction 0 either direction -1 negative direction	I2
40 - 43	Crossed variable +1 $SS(i)$ -1 $DD(i)$ 0 a constant value	I3
50 - 59	Constant value; if zero is specified in the last field	F10.5
60 - 69	Tolerance of crossing	F10.5

Example

Below are 3 examples of the SEVN cards:

1. Define state-event 1 to occur when $SS(3)$ crosses 100 in the positive direction with a tolerance of 2.

Cols	1	10	20	30	40	50	60
	↓	↓	↓	↓	↓	↓	↓
	SEVN	01	+03	+1		100.	2.

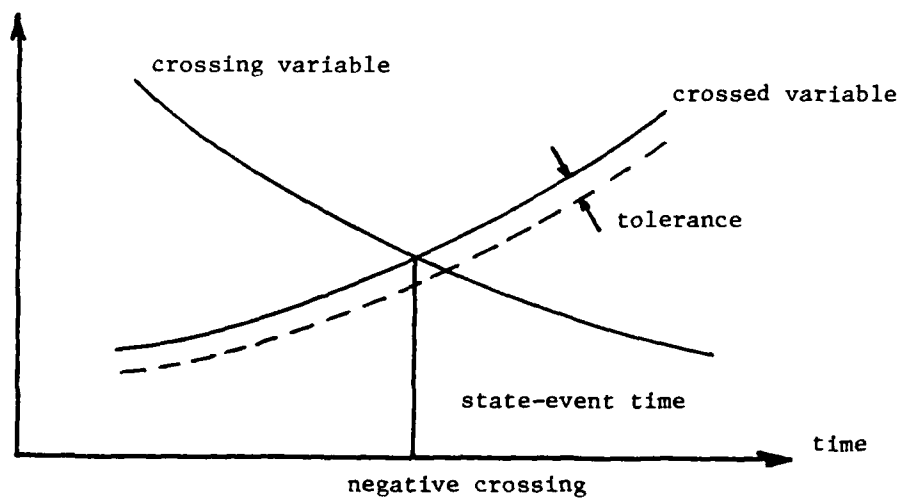
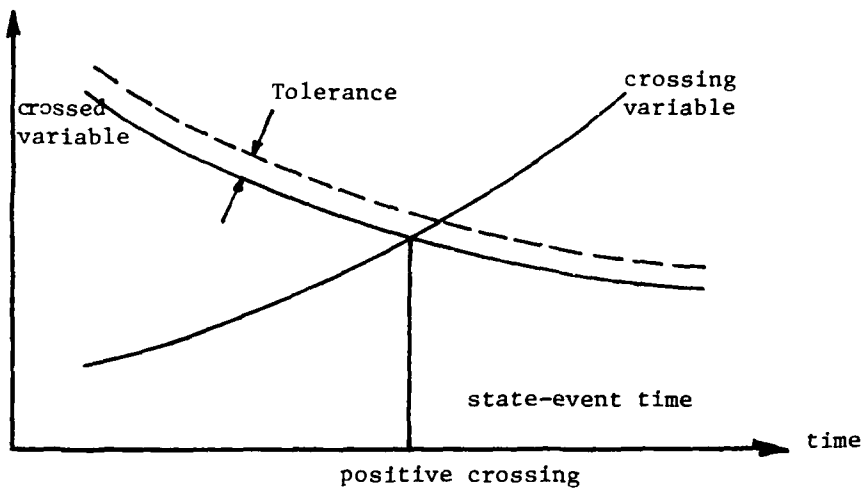
2. Define state-event 2 to occur when $SS(2)$ crosses $DD(1)$ in the negative direction with a tolerance of 0.01.

Cols	1	10	20	30	40	50	60
	↓	↓	↓	↓	↓	↓	↓
	SEVN	02	+02	-1	-01		.01

3. Define state-event 3 to occur when DD(4) crosses SS(5) in either direction with tolerance of zero.

Cols	1	10	20	30	40
	↓	↓	↓	↓	↓
SEVN	03	-04			+05

The following figure illustrates the concept of positive and negative direction of crossing



6. Statistics Card

In this card number of tallies, discrete time persistent statistics, continuous time persistent statistics, counters are specified. Also, DTSAV, time between saving continuous variables is declared on this card.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	STAT	
10 - 11	No. of tallies	I2
20 - 21	No. of discrete time persistent statistics	I2
30 - 31	No. continuous time persistent statistics	I2
40 - 41	No. of counters	I2
50 - 59	The time between observing the continuous time persistent statistics (DTSAV). If it is negative, the statistics are collected every ABS(DTSAV) plus before and after processing any event. In this situation, after processing the event only the value of those continuous variables that are changed within the event are collected.	F10.5

7. Tally Card

For each tally one card must be prepared. Tally number and name are specified on this card. For future use, such as drawing histograms, the analyst might want to have access to every individual observation. By specifying an output unit number in this card, each individual observation will be written unformatted on this unit. Care must be practiced to provide the proper JCL of the computer installation.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	TALY	
10 - 11	Tally number	I2
20 - 31	Tally name	6A2
40 - 41	Unit number	I2

8. Discrete Time Persistent Card

For each discrete time persistent statistic one card must be prepared. Number and name of the discrete time persistent are specified on this card. Also, the output unit number for writing off each individual observation, like the tally card, can be specified on this card. There are two types of discrete time persistent statistics that can be collected. One is on the number of the entries of a file, and the other is on one of the XX variables.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	DTPS	
10 - 11	Number	I2
20 - 31	Name	6A2
40 - 42	-i ith XX variable	I3
	+i Number of entries of file i	
50 - 51	Unit number	I2

Example

Suppose that discrete time persistent number 3 is for collecting statistics on the variation of number of the entries on file 2 and it is to be named "NO. IN QUE.". Plus, it is desired to "dump" out all the changes of the number of the entries of file 2 and the times of changes on output Unit 8. The following card provides this information.

Cols	1	10	20	40	50
	↓	↓	↓	↓	↓
	DTPS	03	NO. IN QUE.	+02	08

9. Continuous Time Persistent Card

For each continuous time persistent statistic one card must be provided. The function of this card is the same as the DTPS card except this is for the continuous variables, i.e. DD(I) or SS(I).

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	CTPS	
10 - 11	Number	I2
20 - 31	Name	6A2
40 - 42	+i SS(i)	I3
	-i DD(i)	
50 - 51	Unit number	I2

Another difference between discrete and continuous time persistent statistics is that, in discrete, all the changes are recorded, but for continuous, every DTSAV the values are recorded. In the case of negative DTSAV, the values of continuous variables are recorded every ABS(DTSAV), before every event and after the event. The latter if the values have changed during the event.

10. Counter Card

This is for defining the counters. For each counter one card must be prepared with specifying number, name, and upper limit of the counter. If at the end of an event the value of one of the counters is larger than its corresponding limit, the simulation would be terminated. Counters can also be used in subroutine STATE. At which case after completion of one step, the value of one of the counters might cause the simulation to be stopped.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	COUN	
10 - 11	Number	I2
20 - 31	Name	6A2
40 - 45	Limit	I6

11. Stream Cards

This is for specifying the initializing seeds of the random number generators. There are a maximum of 10 streams allowed. The language provides the default values. Therefore, only for the streams whose initial seed is to be different from the default values cards must be prepared. One can specify antithetic random numbers by specifying a negative seed.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	STRM	
10 - 11	Number	I2
20 - 30	Seed value	I11

12. Simulate Card

Number of runs to be performed is specified on this card.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	SIMU	
10 - 11	Number of runs	I2

13. Initialization Card

This is for indicating the starting and finishing time of each run of the simulation, whether or not to initialize the seeds between the runs, to suppress the echo report, to suppress the summary report, to clear the files between runs, to clear variables between runs, and finally to clear the statistics and if so, the time to clear.

<u>Cols</u>	<u>Description</u>	<u>Format</u>
1 - 4	INTL	
10 - 19	Starting time of the simulation	F10.3
20 - 29	Finishing time of the simulation	F10.3
40	1 to initialize the seeds between the runs 0 otherwise	I1
45	1 to suppress the echo report 0 otherwise	I1
50	1 to suppress the summary report 0 otherwise	I1
55	1 do not clear the files between the runs 0 otherwise	I1

<u>Cols</u>	<u>Description</u>	<u>Format</u>
60	1 do not clear the variables between the runs 0 otherwise	I1
65	1 do not clear statistics 0 otherwise	I1
70 - 79	Time to clear the statistics	F10.3

APPENDIX B

DATA INPUT AND SAMPLE OUTPUT FROM
NAVAL SYSTEMS SIMULATION PROGRAM

EXAMPLE PROBLEM #1

M/M/1 QUEUE

DISCRETE-EVENT SIMULATION

FORTRAN IV G LEVEL 21 STATE DATE = 81115 09/02/43 PAGE 9

0001 SUBROUTINE STATE
0002 RETURN
0003 END

FORTRAN IV G LEVEL 21 INTIC DATE = 81115 09/02/48 PAGE 0

0001 SUBROUTINE INTIC
0002 COMMON/GSC1/RCDR,NPTR,SS(99),DD(99),ATTRIB(99),SSL(39),JJ,
-DDI(99),TNOW,XX(99),DTNOW,ISTOP
0003 CALL SCHED (1,TNOW,ATTRIB)
0004 RETURN
0005 END

FORTRAN IV G LEVEL 21 EVFNT DATE = 81115 09/02/43 PAGE 0

0001 SUBROUTINE EVFNT (I)
0002 COMMON/GSC1/ACDR,NPTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
-DEL(99),TNOW,XX(99),DTNOW,ISTOP
0003 GO TO (1,2),I
0004 WRITE(6,100) I,ATTRIB(1),ATTRIB(2),TNOW
0005 CALL ABVL
0006 RETURN
0007 WRITE(6,100) I,ATTRIB(1),ATTRIB(2),TNOW
0008 CALL ESRV
0009 RETURN
0010 FORMAT (10X,'EVENT=' ,I2,5X,'ATTRIB(1)=' ,F10.3,5X,'ATTRIB(2)=' ,
-F10.3,5X,'TNOW=' ,F10.3)
0011 END

```

0001 SUBROUTINE APVL
0002 COMMON/GSC1/NCADR, FELR, SS(99), DD(99), ATTRIB(99), SSL(99), JJ,
-DDL(99), TNO, XX(99), DINOW, ISTOP
0003 T=INOW+EXPOI(1.5,1)
0004 CALL SCHD(1,1,ATTRIB)
0005 ATTRIB(1)=TEOK
0006 IF (XX(1).EQ.1) GO TO 2
0007 XX(1)=1.
0008 ATTRIB(2)=TNO
0009 T=INOW+EXPOI(1.,2)
0010 CALL SCHD(2,1,ATTRIB)
0011 RETURN
0012 2 CALL FILEH(1,ATTRIB)
0013 RETURN
0014 END

```

```

0001 SUBROUTINE ESRV
0002 COMMON/GSC1/NCADR, NPETR, SS(99), DD(99), ATTRIB(99), SSL(99), JJ,
-DDI(99), TNO, XX(99), FTNO, ISTOP
0003 TSYS=TNO-ATTRIB(1)
0004 TV=ATTRIB(2)-ATTRIB(1)
0005 CALL TALLY(1,TSYS)
0006 CALL TALLY(2,TV)
0007 CALL COUNT(1,1)
0008 IF (NO(1).EQ.0) GO TO 1
0009 CALL REMOVE(1,1,ATTRIB)
0010 ATTRIB(2)=TNO
0011 T=INOW+EXPOI(1.,1)
0012 CALL SCHD(2,1,ATTRIB)
0013 RETURN
0014 1 XX(1)=0
0015 RETURN
0016 END

```

PLCJ	FIRST PROC.	4/11/1	2	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DISC	1	1	1	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RANK	1	1	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
COMP	0	0	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
STAT	2	2	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ITALY	1	1	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ITALY	2	2	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DIPS	1	1	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DIPS	2	2	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
COUN	1	1	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
STEM	1	1	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SIMU	2	2	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
INITL												

A. NGTAL1
 1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 -1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 100 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 1234555599 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 20.000 0.0 0.0 0.0 0.0 0.0 0.0 0.0
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

FILE INFORMATION

*****LOCATION OF CALLER IS: 7

ADDITIONAL INFORMATION:

TALLY NO. 2	21
LOCATION OF THE TALLY FICK:	4991
LOCATION OF STATISTICAL ARMY:	4991
LABEL: LINE IN OUF.	
OUTPUT DEVICE NO.:	0

DISC. THE PRESIDENT'S COLLECTION

DISC. TIME PRESENTLY 10.2
LOCATION OF BLACK: 00
LOCATION OF STATISTICAL AREA: 0000
LABEL: MC. IN CONF.
TIME PRESENTLY STATISTICAL COLLECTION OF 0000
OUTPUT DEVICE NO.: 0

DIST. TIME PERIOD: 1962-63
 LOCATION OF BIRTH: CH
 LOCATION OF DEATH: CHINA
 RELIGION: BUDDHISM
 DATE OF BIRTH: 1900-01-01
 DATE OF DEATH: 1963-01-01

COUNTER INFORMATION

COUNTER NO.: 1
LOCATION OF COUNTER BLOCK:
LABEL: NO. SERVED
LIMIT: 100 47

LOCATION OF FIRST AVAILABLE ENTRY: 55

NO. OF ENTRIES ALLOCATED: 703

RANDOM STREAM INFORMATION

SEED(1) =	1234555599
SEED(2) =	2135124613
SEED(3) =	1743251544
SEED(4) =	1624217675
SEED(5) =	2014632579
SEED(6) =	2036774231
SEED(7) =	1452313571
SEED(8) =	1254204657
SEED(9) =	1410143363
SEED(10) =	2135621895

SEEDS WILL NOT BE INITIALIZED BETWEEN RUNS

TOTAL CP 2 RUNS WILL BE PERFORMED

EXECUTION IS ATTEMPTED			TNUM=		
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	0.0
EVENT= 2	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	0.205
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	1.485
EVENT= 2	ATTRIB(1) =	1.485	ATTRIB(2) =	1.485	1.545
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	5.819
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	5.835
EVENT= 2	ATTRIB(1) =	5.819	ATTRIB(2) =	5.819	6.979
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	7.082
EVENT= 2	ATTRIB(1) =	5.835	ATTRIB(2) =	6.979	7.323
EVENT= 2	ATTRIB(1) =	7.082	ATTRIB(2) =	7.323	7.803
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	8.733
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	9.229
EVENT= 2	ATTRIB(1) =	8.733	ATTRIB(2) =	8.733	9.626
EVENT= 2	ATTRIB(1) =	9.229	ATTRIB(2) =	9.626	9.635
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	9.675
EVENT= 2	ATTRIB(1) =	9.675	ATTRIB(2) =	9.675	10.128
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	11.704
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	12.386

ANALYST: A. NOZAKI
 RUN 1 OF 2

```

SIMULATION STARTED AT TIME: 0.0
STATISTICS CLEARED AT TIME: 0.0
CURRENT TIME: 0.19E+02

```

***** COUNTER INFORMATION *****

INDEX	LABEL	LIMIT	CURRENT VALUE
1	NO. SERVED	100	13

*** TALLY STATISTICS ***

INDEX	LABEL	NO. OF OBS.	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	TIME IN SYS.	13	0.12E+01	0.98E+00	0.60E-01	0.30E+01
2	TIME IN QUP.	13	0.52E+00	0.87E+00	0.0	0.30E+01

**** DISCRETE TIME PERSISTANT STATISTICS ****

INDEX	LABEL	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	NO. IN QUE.	0.35E+00	0.51E+00	0.0	0.20E+01
2	EFFICIENCY	0.49E+00	0.50E+00	0.0	0.10E+01

EXECUTION IS ATTEMPTED

[illegible]

EVENT= 2	ATTRIB(1) =	8.105	ATTRIB(2) =	8.122	TNOV=	8.171
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	8.255
EVENT= 2	ATTRIB(1) =	8.255	ATTRIB(2) =	9.255	TNOV=	8.340
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	9.111
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	9.410
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	9.688
EVENT= 2	ATTRIB(1) =	9.111	ATTRIB(2) =	9.111	TNOV=	10.147
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	10.420
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	10.980
EVENT= 2	ATTRIB(1) =	9.410	ATTRIB(2) =	10.147	TNOV=	11.747
EVENT= 2	ATTRIB(1) =	9.688	ATTRIB(2) =	11.747	TNOV=	12.115
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	12.712
EVENT= 2	ATTRIB(1) =	10.420	ATTRIB(2) =	12.115	TNOV=	13.268
EVENT= 1	ATTRIB(1) =	0.0	ATTRIB(2) =	0.0	TNOV=	13.316
EVENT= 2	ATTRIB(1) =	10.980	ATTRIB(2) =	13.268	TNOV=	15.039
EVENT= 2	ATTRIB(1) =	12.712	ATTRIB(2) =	15.039	TNOV=	15.788
EVENT= 2	ATTRIB(1) =	13.316	ATTRIB(2) =	15.788	TNOV=	15.970

EXAMPLE PROBLEM #2

PILOT EJECTION PROBLEM

CONTINUOUS SIMULATION

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022

SUBROUTINE INTLC

COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,

-DDL(99),TNOW,XX(99),DTNOW,ISTOP

COMMON /USER1/CD,G,RHO,THED,VA,VE,XM,XS,Y1

CD=1.

G=32.2

RHO=.0023769

THED=15.

VE=40.0

XM=7

XS=10.0

Y1=4.0

READ(NCRDR,101) VA

FORMAT(1F10.0)

THE=THED/57.3

VX=VA-VE*SIN(THED)

VY=VE*COS(THED)

SS(3)=SQRT(VX*VX+VY*VY)

SS(4)=ATAN(VY/VX)

XX(1)=0.

CALL SCHED (3,TNOW,ATTRIB)

RETURN

END

101

23/24/17

DATE = 81113

STATE

PORTMAN IV G LEVEL 21

```

0001 SUBROUTINE STATE
0002 COMMON/GSC1/NCRDR, NPRTR, SS(99), DD(99), ATTRIB(99), SSL(99), JJ,
      -DDL(99), TNOW, XX(99), DTNOW, ISTOP
0003 COMMON /USER1/CD, G, RHO, THED, VA, VE, XM, XS, Y1
0004 DD(1) = SS(3)*COS(SS(4)) - VA
0005 DD(2) = SS(3)*SIN(SS(4))
0006 IF (XX(1).LT.1.) RETURN
0007 XD = .5*RHO*CD*XS*SS(3)*SS(3)
0008 DD(3) = -XD/XM-G*SIN(SS(4))
0009 DD(4) = -G*COS(SS(4))/SS(3)
0010 RETURN
0011 END

```

23/24/17

DATE = 81113

EVENT

PORTMAN IV G LEVEL 21

```

0001 SUBROUTINE EVENT (IX)
0002 COMMON/GSC1/NCRDR, NPRTR, SS(99), DD(99), ATTRIB(99), SSL(99), JJ,
      -DDL(99), TNOW, XX(99), DTNOW, ISTOP
0003 GO TO (1,2,3) , IX
0004 ISTOP=1
0005 RETURN
0006 XX(1)=1.
0007 RETURN
0008 CALL TABLE
0009 RETURN
0010 END

```

23/24/17

DATE = 81113

TABLE

PORTMAN IV G LEVEL 21

```

0001 SUBROUTINE TABLE
0002 COMMON/GSC1/NCRDR, NPRTR, SS(99), DD(99), ATTRIB(99), SSL(99), JJ,
      -DDL(99), TNOW, XX(99), DTNOW, ISTOP
0003 T=TNOW+.01
0004 CALL SCHD (3,T,ATTRIB)
0005 WRITE (NPRTR,101) TNOW, (SS(I), I=1,4)
0006 FORMAT( 10X,5(F10.5,10X))
0007 RETURN
0008 END

```

PROJ	PILOT EJECTION		NOZARI A				4/5/81
DISC	0	1	0	0.00010	0.01000	0.00001	0.00000W
CONT	4	0	-1	0	0	-60.00000	0.0
SEVN	1	1	1	0	0	30.00000	0.0
SEVN	1	2	1	0	0	4.00000	0.0
SEVN	2	2	1	0	0	0.01000	0.0
STAT	0	0	4	0	0	0	0
CTPS	1	X POS.			1	0	
CTPS	2	Y POS.			2	0	
CTPS	3	SPEED			3	0	
CTPS	4	THETA			4	0	
SINU	2						
INTL	0.0	4.000	0	0	0	0	0.0

T H F E C H O R P O R T

THIS IS A COMBINED MODEL WITH 4 CONTINUOUS EQUATIONS

FILE INFORMATION

NO FILES ARE DESIGNATED

***LOCATION OF CALPNDER IS: 1

CONTINUOUS INFORMATION

NO. OF DIFFERENTIAL EQUATIONS (NEQD): 4
 NO. OF DIFFERENCE EQUATIONS (NEQS): 0
 MINIMUM STEP SIZE (DTMIN): 0.00010
 MAXIMUM STEP SIZE (DTMAX): 0.01000
 ABSOLUTE ERROR LIMIT (AERR): 0.00001
 RELATIVE ERROR LIMIT (RERR): 0.00010
 TIME BETWEEN SAVE POINTS (DTSAV): 0.01000
 ACCURACY ERROR SPECIFICATION (TERR): W

STATE EVENTS

NUMBER	EVENT	CROSSING VARIABLE	DIRECTION OF CROSSING	CROSSED VALUE	TOLERANCE OF CROSSING
1	1	SS(1)	-1	-60.00000	0.0
2	1	SS(2)	1	30.00000	0.0
3	2	SS(2)	1	4.00000	0.0

CONT. TIME PERSISTANT INFORMATION

CONT. TIME PERSISTANT NO.: 1
 LOCATION OF BLOCK: 7
 LOCATION OF STATISTICAL ARRAY: 4994
 LABEL: X POS.
 TIME PERSISTANT STATISTICAL COLLECTION OR SS(1)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT NO.: 2
 LOCATION OF BLOCK: 16
 LOCATION OF STATISTICAL ARRAY: 4997
 LABEL: Y POS.
 TIME PERSISTANT STATISTICAL COLLECTION OR SS(1)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERISTANT NO.: 3
 LOCATION OF BLOCK: 25
 LOCATION OF STATISTICAL ARRAY: 4980
 LABEL: SPEED
 TIME PERISTANT STATISTICAL COLLECTION ON SS(3)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERISTANT NO.: 4
 LOCATION OF BLOCK: 34
 LOCATION OF STATISTICAL ARRAY: 4973
 LABEL: THETA
 TIME PERISTANT STATISTICAL COLLECTION ON SS(4)
 OUTPUT DEVICE NO.: 0

LOCATION OF FIRST AVAILABLE ENTRY: 43

NO. OF ENTRIES ALLOCATED: 986

----- RANDOM STREAM INFORMATION -----

SEED(1) = 1274321477
 SEED(2) = 2135124613
 SEED(3) = 1743251541
 SEED(4) = 1624217675
 SEED(5) = 2014632579
 SEED(6) = 2036774231
 SEED(7) = 1452313571
 SEED(8) = 1254240657
 SEED(9) = 1410143363
 SEED(10) = 2135621895

SEEDS WILL NOT BE INITIALIZED BETWEEN RUNS

TOTAL OF 2 RUNS WILL BE PERFORMED

EXECUTION IS ATTEMPTED
 0.0

0.0	0.0	0.0	890.48657	0.04340
0.01000	-0.10340	0.38591	890.48657	0.04340
0.02000	-0.20692	0.77229	890.48657	0.04340
0.03000	-0.31044	1.15866	890.48657	0.04340
0.04000	-0.41396	1.54503	890.48657	0.04340
0.05000	-0.51748	1.93140	890.48657	0.04340
0.06000	-0.62100	2.31777	890.48657	0.04340
0.07000	-0.72452	2.70414	890.48657	0.04340
0.08000	-0.82804	3.09052	890.48657	0.04340

SPECIFIED LOCAL ERROR EXCEEDED FOR SS(2) AT TIME 0.5000E-04

SUMMARY REPORT

PROJECT: PILOT EJECTION
DATE: 4/5/81

ANALYST: NOZARI A
RUN 1 OF 2

SIMULATION STARTED AT TIME: 0.0
STATISTICS CLEARED AT TIME: 0.0
CURRENT TIME: 0.43E+00

*** CONTINUOUS TIME PERSISTANT STATISTICS ***

INDEX	LABEL	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	X POS.	-0.17E+02	0.18E+02	-0.60E+02	-0.10E+00
2	Y POS.	0.73E+01	0.38E+01	0.39E+00	0.13E+02
3	SPEED	0.75E+03	0.13E+03	0.59E+03	0.89E+03
4	THETA	0.38E-01	0.63E-02	0.28E-01	0.43E-01

FINAL VALUE OF CONTINUOUS VARIABLES

SS (1) = -0.60E+02	DD (1) = -0.31E+03
SS (2) = 0.13E+02	DD (2) = 0.17E+02
SS (3) = 0.52E+03	DD (3) = -0.60E+02
SS (4) = 0.28E-01	DD (4) = -0.54E-01

$$\begin{aligned} DD(1) &= -0.31E+03 \\ DD(2) &= 0.17E+02 \\ DD(3) &= -0.60E+03 \\ DD(4) &= -0.54E-01 \end{aligned}$$

EXECUTION IS ATTEMPTED

0.0	0.0	0.0	0.0
0.01000	-0.10380	2)	AT TIME
0.02000	-0.20692		0.38591
0.03000	-0.31085		0.77228
0.04000	-0.41397		1.15866
0.05000	-0.51749		1.54503
0.06000	-0.62101		1.93140
0.07000	-0.72454		2.31777
0.08000	-0.82806		2.70414
0.09000	-0.93158		3.09051
0.10000	-1.03511		3.47688
			3.86325

SPECIFIED	TOLERANCE	EXCEEDED FOR SS(	2) AT TIME	0.1036E+00
	0.11000	-1.14671	4.24835	448.58521
	0.12000	-1.29593	4.62749	444.54053
	0.13000	-1.44890	5.00032	480.56226
	0.14000	-1.71296	5.36691	476.64844
	0.15000	-1.97948	5.72734	472.79785
	0.16000	-2.28386	6.08168	469.00879
	0.17000	-2.62547	6.43001	465.27979
	0.18000	-3.00374	6.77238	461.60937
	0.19000	-3.41808	7.10888	457.99634
	0.20000	-3.86795	7.43956	454.43921
	0.21000	-4.35279	7.76449	450.93677
	0.22000	-4.87207	8.08373	447.48779
	0.23000	-5.42527	8.39735	444.09106
	0.24000	-6.01187	8.70541	440.74536
	0.25000	-6.63137	9.00796	437.44971
	0.26000	-7.28328	9.30507	434.20288
	0.27000	-7.96714	9.59678	431.00366
	0.28000	-8.68246	9.88316	427.85132
	0.29000	-9.42880	10.16426	424.74463
	0.30000	-10.20570	10.44012	421.68262
	0.31000	-11.01273	10.71082	418.66455
	0.32000	-11.84946	10.97638	415.68921
	0.33000	-12.71547	11.23687	412.75586
	0.34000	-13.61035	11.49233	409.86353
	0.35000	-14.53369	11.74282	407.01147
	0.36000	-15.48511	11.98836	404.19873
	0.37000	-16.46422	12.22902	401.42456
	0.38000	-17.47064	12.46484	398.68823
	0.39000	-18.50400	12.69585	395.98901
	0.40000	-19.56393	12.92211	393.32617
	0.41000	-20.65010	13.14365	390.69873
	0.42000	-21.76215	13.36052	388.10620
	0.43000	-22.89972	13.57276	385.54810
	0.44000	-24.06250	13.78040	383.02344
	0.45000	-25.25015	13.98348	380.53174
	0.46000	-26.46236	14.18205	378.07227
	0.47000	-27.69879	14.37614	375.64429
	0.48000	-28.95915	14.56578	373.24756
	0.49000	-30.24315	14.75102	370.88110
	0.50000	-31.55046	14.93188	368.54468
	0.51000	-32.88081	15.10840	366.23755
	0.52000	-34.23392	15.28062	363.95923
	0.53000	-35.60948	15.44857	361.70923
	0.54000	-37.00723	15.61228	359.48682
	0.55000	-38.42691	15.77179	357.29175
	0.56000	-39.86824	15.92712	355.12329
	0.57000	-41.33096	16.07829	352.98120
	0.58000	-42.81482	16.22536	350.86499
	0.59000	-44.31956	16.36833	348.77417
	0.60000	-45.84494	16.50726	346.70825
	0.61000	-47.39072	16.64217	344.66675
	0.62000	-48.95665	16.77307	342.64941
	0.63000	-50.54250	16.90001	340.65552
	0.64000	-52.14803	17.02299	338.68481
	0.65000	-53.77303	17.14207	336.73706
	0.66000	-55.41727	17.25726	334.81177
	0.67000	-57.08054	17.36858	332.90845
	0.68000	-58.76260	17.47606	331.02686

0.07333
0.07767
0.07731
0.07534
0.07566
0.07494
0.07429
0.07360
0.07290
0.07219
0.07149
0.07077
0.07005
0.06932
0.06859
0.06785
0.06711
0.06636
0.06561
0.06485
0.06409
0.06332
0.06254
0.06176
0.06097
0.06018
0.05938
0.05858
0.05777
0.05695
0.05613
0.05531
0.05448
0.05364
0.05280
0.05195
0.05110
0.05024
0.04937
0.04850
0.04763
0.04675
0.04586
0.04497
0.04407
0.04317
0.04226
0.04135
0.04043
0.03950
0.03857
0.03763
0.03669
0.03574
0.03479
0.03383
0.03287
0.03190

EXAMPLE PROBLEM #3

CEDAR BOG LAKE PROBLEM

STATE-EVENT SIMULATION

PROJ	CEDAK BUG	NOZART A	4/4/81	W
DISC	05 01	.025	.00001	
CONT		.00025	.00001	
STAT		06	.025	
CTPS	01	PLANTS	0-1	
CTPS	02	HERBIVORS	0-2	
CTPS	03	CARNIVORAS	0-3	
CIPS	04	ORGANIC	0-4	
CTPS	05	ENVIRONMENT	0-5	
CTPS	06	SCLAR	0+6	
SIMU	01			
INTL		2.0		

FORTRAN IV G LEVEL 21

INTLC

DATE = 81115

09/05/46

PAGE 000

0001
0002
0003
0004
0005
0006
0007
0008
0009

SUBROUTINE INTLC

COMMON/GSC1/NCRDR,NPCTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
-DDL(99),TNOW,XX(99),DTNOW,ISTOP

SS(1)=-.83
SS(2)=-.003
SS(3)=-.0001
SS(4)=0.0
SS(5)=0.0
RETURN
END

FORTRAN IV G LEVEL 21

EVENT

DATE = 81115

09/05/46

PAGE 000

0001
0002
0003

SUBROUTINE EVENT(I)

RETURN
END

FORTRAN IV G LEVEL 21

STATE

DATE = 81115

09/05/46

PAGE 000

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011

SUBROUTINE STATE

COMMON/GSC1/NCRDR,NPCTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
-DDL(99),TNOW,XX(99),DTNOW,ISTOP

DATA PI/3.14159/
SS(6)=95.9*(1.+635*SSIN(2.*PI*TNOW))
DD(1)=SS(6)-4.03*SS(1)
DD(2)=.48*SS(1)-17.87*SS(2)
DD(3)=4.85*SS(2)-4.65*SS(3)
DD(4)=2.55*SS(1)+6.12*SS(2)+1.95*SS(3)
DD(5)=SS(1)+6.9*SS(2)+2.7*SS(3)
RETURN
END

PROJ	CIDAR	BOG	NOZARI	A	4/4/81
DISC	0	0	0	0.00001	0.00001W
CONT	5	1	0	0.00025	0.00001W
STAT	0	0	6	0	0.02500
CTPS	1	0	PLANTS	-1	0
CTPS	2	0	HERBIVORS	-2	0
CTPS	3	0	CARNIVORAS	-3	0
CTPS	4	0	ORGANIC	-4	0
CTPS	5	0	ENVIRONMENT	-5	0
CTPS	6	0	SOLAR	6	0
SIMU	1	0.0	2.000	0	0
INTL				0	0.0

SUMMARY REPORT

PROJECT: FIRST PROB. 4/4/1
DATE: 6/15/1981

ANALYST: A. NOZAKI
PUN 2 OF 2

SIMULATION STARTED AT TIME: 0.0
STATISTICS CLEARED AT TIME: 0.0
CURRENT TIME: 0.16E+02

*** COUNTER INFORMATION ***

INDEX	INDEX	LIMIT	CURRENT VALUE
1	NO. SERVED	100	11

*** DAILY STATISTICS ***

INDEX	INDEX	NO. OF OBS.	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	TIME IN SYS.	11	0.20E+01	0.13E+01	0.66E-01	0.41E+01
2	TIME IN QUE.	11	0.11E+01	0.11E+01	0.0	0.25E+01

*** DISCRETE TIME PERSISTENT STATISTICS ***

INDEX	INDEX	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	NO. IN QUE.	0.70E+00	0.07E+00	0.0	0.30E+01
2	EFFICIENCY	0.63E+00	0.00E+00	0.0	0.10E+01

200 F E C H O R E P O R T

THIS IS A CONTINUOUS MODEL WITH 6 CONTINUOUS EQUATIONS

CONTINUOUS INFORMATION

NO. OF DIFFERENTIAL EQUATIONS (NEQD): 5
 NO. OF DIFFERENCE EQUATIONS (NEQS): 1
 MINIMUM STEP SIZE (DTMIN): 0.00025
 MAXIMUM STEP SIZE (DTMAX): 0.02500
 ABSOLUTE ERROR LIMIT (AERR): 0.00001
 RELATIVE ERROR LIMIT (REPR): 0.00001
 TIME BETWEEN SAVE POINTS (DTSAV): 0.02500
 ACCURACY ERROR SPECIFICATION (AERR): W

CONT. TIME PERSISTANT INFORMATION

CONT. TIME PERSISTANT NO.: 1
 LOCATION OF BLOCK: 1
 LOCATION OF STATISTICAL ARRAY: 4994
 LABEL: PLANTS
 TIME PERSISTANT STATISTICAL COLLECTION ON DD(1)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT NO.: 2
 LOCATION OF BLOCK: 10
 LOCATION OF STATISTICAL ARRAY: 4987
 LABEL: HERBIVORES
 TIME PERSISTANT STATISTICAL COLLECTION ON DD(2)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT NO.: 3
 LOCATION OF BLOCK: 19
 LOCATION OF STATISTICAL ARRAY: 4980
 LABEL: CARNIVORAS
 TIME PERSISTANT STATISTICAL COLLECTION ON DD(3)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT NO.: 4
 LOCATION OF BLOCK: 24
 LOCATION OF STATISTICAL ARRAY: 4973
 LABEL: ORGANIC
 TIME PERSISTANT STATISTICAL COLLECTION ON DD(4)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT NO.: 5
 LOCATION OF BLOCK: 37
 LOCATION OF STATISTICAL ARRAY: 4966

LABEL: ENVIRONMENT
 TIME PERSISTANT STATISTICAL COLLECTION ON DD(4)
 OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT RG.: 6
 LOCATION OF BLOCK: 44
 LOCATION OF STATISTICAL ARRAY: 4959
 LABEL: SCALAR
 TIME PERSISTANT STATISTICAL COLLECTION ON SS(1)
 OUTPUT DEVICE NO.: 0

RANCOM STREAM INFORMATION

SEED(1) = 1274321477
 SEED(2) = 2135124613
 SEED(3) = 1743251541
 SEED(4) = 1624217675
 SEED(5) = 2014632579
 SEED(6) = 2036774231
 SEED(7) = 1452313571
 SEED(8) = 1254240657
 SEED(9) = 1410143303
 SEED(10) = 2135621695

SEEDS WILL NOT BE INITIALIZED BETWEEN RUNS

TOTAL OF 1 RUNS WILL BE PERFORMED

EXECUTION IS ATTEMPTED

SPECIFIED LOCAL ERROR EXCEEDED FOR SS(1) AT TIME 0.1250E-03

SUMMARY REPORT

PROJECT: CEDAR BOG
DATE: 4/4/81

ANALYSIS: PUGAPT A
RUN 1 OF 1

SIMULATION STARTED AT TIME: 0.0
STATISTICS CLEARED AT TIME: 0.0
CURRENT TIME: 0.20E+01

*** CONTINUOUS TIME PERSISTANT STATISTICS ***

INDEX	LABEL	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	PLANTS	0.81E+01	0.44E+02	-0.51E+02	0.93E+02
2	HERDIVORES	0.22E+00	0.11E+01	-0.13E+01	0.23E+01
3	CARNIVOMAS	0.28E+00	0.73E+00	-0.80E+00	0.16E+01
4	ORGANIC	0.60E+02	0.19E+02	0.21E+01	0.89E+02
5	ENVIRONMENT	0.27E+02	0.83E+01	0.85E+00	0.40E+02
6	SOLAR	0.96E+02	0.43E+02	0.35E+02	0.16E+03

FINAL VALUE OF CONTINUOUS VARIABLES

SS(1) = 0.17E+02	DD(1) = 0.29E+02
SS(2) = 0.49E+00	DD(2) = 0.30E+00
SS(3) = 0.57E+00	DD(3) = -0.52E+00
SS(4) = 0.12E+03	DD(4) = 0.47E+02
SS(5) = 0.54E+02	DD(5) = 0.21E+02
SS(6) = 0.96E+02	DD(6) = 0.0

APPENDIX C

FORTTRAN LISTING FOR NAVAL SYSTEMS SIMULATION PROGRAM

	1	2	3	4	5	6
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

1.      DIMENSION RSET(5000)
2.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTE,LFAE,LCAL
3.      COMMON ISET(5000)
4.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
5.      -DEL(99),TNOW,XX(99),DTNOW,ISTCF
6.      EQUIVALENCE (ISET(1),RSET(1))
7.      NCRDR=5
8.      NPRTR=6
9.      LFI=1
10.     LLR=5001
11.     CALL SOAP
12.     STOP
13.     END
14.     SUBROUTINE SOAP
15.     DIMENSION RSET(1)
16.     COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTE,LFAE,LCAL
17.     COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
18.     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
19.     COMMON ISET(1)
20.     EQUIVALENCE (ISET(1),RSET(1))
21.     DATA STARS/2H**/
22.     CALL DEFALT
23.     CALL DATAIN
24.     IF(IECO.EQ.1) GO TO 10
25.     CALL ECHO
26. 10    DO 20 I=1,NRUNS
27.         IF(ICLF.EQ.1) GO TO 11
28.         CALL CLEARF
29. 11        IF(ICLV.EQ.1) GO TO 12
30.         CALL CLEARV
31. 12        CALL INTAL
32.         CALL EXEC
33.         IF(ISUM.EQ.1) GO TO 13
34.         CALL SUMRY (I)
35. 13        IF(NDTPST.EQ.0) GO TO 15
36.             DO 14 J=1,NDTPST
37.                 L=LFDSB+(J-1)*9
38.                 NU=ISET(L+8)
39.                 IF(NU.LE.0) GO TO 14
40.                 WRITE(NU) STARS
41. 14                CONTINUE
42. 15        IF(NCTPST.EQ.0) GO TO 19
43.             DO 16 J=1,NCTPST
44.                 L=LPCSE+(J-1)*9
45.                 NU=ISET(L+8)
46.                 IF(NU.LE.0) GO TO 16
47.                 WRITE(NU) STARS
48. 16                CONTINUE
49. 19        CALL OUTPUT
50. 20    CCNTINUE
51.     STOP
52.     END
53.     SUBROUTINE DEFALT
54.     DIMENSION RSET(1),LL(10)
55.     DATA LL/1274321477,2135124613,1743251541,1624217675,2014632579,

```

	1	2	3	4	5	6
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
56.	-2036774231	,1452313571,	1254240657,	1410143363,	2135621895/	
57.	COMMON/LOC/LFI,	ILR,LFFB,LFTB,	LFESB,LFCSE,	LPCTB,LFAE,	LCAL	
58.	COMMON/GEN1/NRUNS,	NFILE,TBEG,	TFIN,ICLF,	ICLV,ICLS,	TCLEAR,IRAN,	
59.	-NTMX,LTMX(99),	NTAL,NDTPST,	NCTPST,NCCUN,	NATR,NENT,	IECO,ISUM	
60.	COMMON/GEN4/ISEED(10),	LSEED(10)				
61.	COMMON/GEN8/NIR,	VALH,VALI				
62.	COMMON ISET(1)					
63.	EQUIVALENCE (ISET(1),	RSET(1))				
64.	C*					
65.	C*	THIS SUBROUTINE ASSIGNS THE DEFAULT VALUES				
66.	C*	TO THE SOAP VARIABLES.				
67.	C*					
68.		NIR=1				
69.		VALH=1.E20				
70.		VALL=-1.E20				
71.		ICLF=0				
72.		ICLV=0				
73.		ICLS=0				
74.		CALL CLEARV				
75.		NFILE=0				
76.		LFPB=0				
77.		LFTB=0				
78.		LFDSB=0				
79.		LFCSE=0				
80.		LFAE=0				
81.		LCAL=0				
82.		NTMX=0				
83.		NENT=0				
84.		NATR=0				
85.		NTAL=0				
86.		NDTPST=0				
87.		NCTPST=0				
88.		N=LLR-1				
89.		DO 1 I=1,N				
90.		ISET(I)=0				
91.	1	CONTINUE				
92.		DO 2 I=1,10				
93.		ISEED(I)=LL(I)				
94.		LSEED(I)=LL(I)				
95.	2	CONTINUE				
96.		RETURN				
97.		END				
98.		SUBROUTINE ECHO				
99.		DIMENSION RSET(1),INFO(3)				
100.		DATA INFO/1HN,1HW,1HF/				
101.		COMMON/LOC/LFI,LLR,LFPB,LFTB,LFDSB,LFCSE,LPCTB,LFAE,LCAL				
102.		COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,				
103.		-NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM				
104.		COMMON/GEN2/NEQD,NEQS,NEQT,DTSV,DTMIN,DTMAX,AERR,RERR,IERR,NSEI				
105.		-ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),				
106.		-ISCD(25)				
107.		COMMON/GEN4/ISEED(10),LSEED(10)				
108.		COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,				
109.		-DDL(99),TNOW,XX(99),DTNOW,ISTCP				
110.		COMMON ISET(1)				

	1	2	3	4	5	6
	123456789012345678901234567890123456789012345678901234567890123456789					
111.						
112.						
113.	95					
114.						
115.						
116.						
117.	100					
118.						
119.						
120.	11					
121.	101					
122.						
123.						
124.	12					
125.	102					
126.	13					
127.	103					
128.						
129.						
130.	110					
131.						
132.	16					
133.						
134.	104					
135.						
136.						
137.	105					
138.						
139.						
140.	106					
141.						
142.						
143.						
144.						
145.	107					
146.	14					
147.						
148.						
149.	108					
150.	10					
151.	15					
152.	109					
153.	C*					
154.	C*					
155.	C*					
156.	20					
157.						
158.	120					
159.						
160.						
161.						
162.	121					
163.						
164.						
165.						

```

EQUIVALENCE(ISET(1),RSET(1))
WRITE(NPRTR,95)
95  FORMAT(1H1,52X,29HT H E   E C H O   R E P O R T)
    IF(NENT.GT. 0.AND.NEQT.GT.0) GO TO 11
    IF(NENT.GT.0) GO TO 12
    WRITE(NPRTR,100) NEQT
100  FORMAT(//10X,31HTHIS IS A CCNTINUOUS MODEL WITH ,I4,
-21H CONTINUOUS EQUATIONS)
    GO TO 20
11  WRITE(NPRTR,101) NEQT
101  FORMAT(//10X,29HTHIS IS A CCMBINED MODEL WITH,I4,
-21H CONTINUOUS EQUATIONS)
    GO TO 13
12  WRITE(NPRTR,102)
102  FORMAT(//10X,24HTHIS IS A DISCRETE MODEL)
13  WRITE(NPRTR,103)
103  FORMAT(//10X,16HFILE INFORMATION/10X,16(1H-))
    IF(NFILE.GT.0) GO TO 16
    WRITE(NPRTR,110)
110  FORMAT(10X,23HNO FILES ARE DESIGNATED)
    GO TO 15
16  DO 10 I=1,NFILE
    WRITE(NPRTR,104) I
104  FORMAT(//10X,9HFILE NO.: ,I4)
    L=L+1
    WRITE(NPRTR,105) L
105  FORMAT(10X,23HLOCATION OF FILE BLOCK: ,I6)
    J=L+3
    WRITE(NPRTR,106) ISET(J)
106  FORMAT(10X,22H RANKING DISCIPLINE IS: ,I2/
-,20X,32H1: PCFS 2: LCFS 3: HVP 4: LVP)
    J=L+4
    IF(ISET(J).EQ.0) GO TO 14
    WRITE(NPRTR,107) ISET(J)
107  FORMAT(10X,21H RANKING ATTRIBUTE IS : ,I3)
    J=L+5
    IF(ISET(J).EQ.0) GO TO 15
    WRITE(NPRTR,108) ISET(J)
108  FORMAT(10X,29HLOCATION OF STATISTICS BLOCK: ,I6)
10  CONTINUE
15  WRITE(NPRTR,109) LCAL
109  FORMAT(//10X,28H****LOCATION OF CALENDER IS: ,I6)
C*
C*  PRINT CONTINUOUS INFORMATION
C*
20  IF (NEQT.LE.0) GO TO 21
    WRITE(NPRTR,120) NEQD,NEQS
120  FORMAT(///10X,22HCONTINUOUS INFORMATION/10X,22(1H-)//
-10X,37HNO. OF DIFFERENTIAL EQUATIONS (NEQD): ,10X,I5/10X,
-37HNO. OF DIFFERENCE EQUATIONS (NEQS): ,10X,I5)
    WRITE(NPRTR,121) DTMIN,DTMAX,AERR,RERR,DTSV
121  FORMAT(10X,26HMINIMUM STEP SIZE (DTMIN): ,16X,F10.5/10X,
-26HMAXIMUM STEP SIZE (DTMAX): ,16X,F10.5,/10X,
-28HABSOLUTE ERROR LIMIT (AERR): ,14X,F10.5/10X,
-28HRELATIVE ERROR LIMIT (RERR): ,14X,F10.5/10X,

```

```

1234567890123456789012345678901234567890123456789012345678901234567890
166.      -33H TIME BETWEEN SAVE POINTS (DTSAV):,9X,F10.5)
167.      IF (IERR.EQ.-1) WRITE(NPRTR,122) INFO(1)
168.      IF (IERR.EQ.0) WRITE(NPRTR,122) INFO(2)
169.      IF (IERR.EQ.1) WRITE(NPRTR,122) INFO(3)
170. 122    FORMAT(10X,36H ACCURACY ERROR SPECIFICATION (IERR):15X,A1)
171.      IF (NSEV.LE.0) GO TO 21
172.      WRITE(NPRTR,130)
173. 130    FORMAT(///10X,12H STATE EVENTS/10X,12(1H-)//10X,6H NUMBER,10X,5H E
174.      -T,11X,8H CROSSING,10X,9H DIRECTION,10X,7H CROSSED,10X,9H TOLERANCE/
175.      -,8H VARIABLE,10X,11H CFC CROSSING,8X,5H VALUE,12X,11H OF CROSSING//)
176.      DO 28 I=1,NSEV
177.      WRITE(NPRTR,131) I,ICSV(I)
178. 131    FORMAT(12X,I2,13X,I3)
179.      IF (ICGV(I).LT.0) GO TO 23
180.      WRITE(NPRTR,132) ICGV(I)
181. 132    FORMAT(1H+,42X,3H SS(,I2,1H))
182.      GO TO 24
183. 23     J=-ICGV(I)
184.      WRITE(NPRTR,133) J
185. 133    FORMAT(1H+,42X,3H DD(,I2,1H))
186. 24     WRITE(NPRTR,134) ICDIR(I)
187. 134    FORMAT(1H+,64X,I2)
188.      IF (ICDV(I).LT.0) GO TO 25
189.      IF (ICDV(I).GT.0) GO TO 26
190.      WRITE(NPRTR,135) VAL(I)
191. 135    FORMAT(1H+,79X,F10.5)
192.      GO TO 27
193. 25     J=-ICDV(I)
194.      WRITE(NPRTR,136) J
195. 136    FORMAT(1H+,79X,3H DD(,I2,1H))
196.      GO TO 27
197. 26     WRITE(NPRTR,137) ICDV(I)
198. 137    FORMAT(1H+,79X,3H SS(,I2,1H))
199. 27     WRITE(NPRTR,138) TTOL(I)
200. 138    FORMAT(1H+,96X,F10.5)
201. 28     CONTINUE
202. 21     IF (NTAL.EQ.0) GO TO 31
203.      WRITE(NPRTR,150)
204. 150    FORMAT(///10X,17H TALLY INFORMATION/10X,17(1H-))
205.      DO 30 I=1,NTAL
206.      L=LFTB+(I-1)*8
207.      WRITE(NPRTR,151) I
208. 151    FORMAT(//10X,9H TALLY NO., I3)
209.      WRITE(NPRTR,152) L
210. 152    FORMAT(10X,28H LOCATION OF THE TALLY BLOCK:,I6)
211.      WRITE(NPRTR,153) ISET(L)
212. 153    FORMAT(10X,30H LOCATION OF STATISTICAL ARRAY:,I6)
213.      J1=L+1
214.      J2=L+6
215.      WRITE(NPRTR,154) (ISET(J),J=J1,J2)
216. 154    FORMAT(10X,7H LABEL: ,6A2)
217.      J=L+7
218.      WRITE(NPRTR,155) ISET(J)
219. 155    FORMAT(10X,18H OUTPUT DEVICE NO.: ,I4)
220. 30     CONTINUE

```

	1	2	3	4	5	6	7
	12345678901234567890123456789012345678901234567890123456789012345						
21.	31	IF (NDTPST.EQ.0)	GO TO 41				
22.		WRITE (NPRTR,160)					
23.	160	FCRMT (///10X,33HDISC. TIME PERSISTANT INFORMATION/10X,32(1H-))					
24.		DC 40 I=1,NDTPST					
25.		L=LFDSB+(I-1)*9					
26.		WRITE (NPRTR,161) I					
27.	161	FCRMT (//10X,27HDISC. TIME PERSISTANT NO.: ,I3)					
28.		WRITE (NPRTR,162) L					
29.	162	FORMAT (10X,18HLOCATION OF BLOCCK: ,I6)					
30.		WRITE (NPRTR,163) ISET(L)					
31.	163	FCRMT (10X,30HLOCATION OF STATISTICAL ARRAY: ,I6)					
32.		J1=L+1					
33.		J2=L+6					
34.		WRITE (NPRTR,154) (ISET(J),J=J1,J2)					
35.		J=L+7					
36.		IF (ISET(J).GT.0) GO TO 42					
37.		J=-ISET(J)					
38.		WRITE (NPRTR,165) J					
39.	165	FORMAT (10X,45HTIME PERSISTANT STATISTICAL COLLECTION ON XX(					
40.		-,I2,1H))					
41.		GO TO 43					
42.	42	WRITE (NPRTR,166) ISET(J)					
43.	166	FORMAT (10X,47HTIME PERSISTANT STATISTICAL COLLECTION ON QUEUE,I3)					
44.	43	J=L+8					
45.		WRITE (NPRTR,155) ISET(J)					
46.	40	CONTINUE					
47.	41	IF (NCTPST.EQ.0) GO TO 51					
48.		WRITE (NPRTR,170)					
49.	170	FORMAT (///10X,33HCONT. TIME PERSISTANT INFORMATION/10X,33(1H-))					
50.		DC 50 I=1,NCTPST					
51.		L=LFCSB+(I-1)*9					
52.		WRITE (NPRTR,171) I					
53.	171	FORMAT (//10X,27HCONT. TIME PERSISTANT NO.: ,I3)					
54.		WRITE (NPRTR,172) L					
55.	172	FCRMT (10X,18HLOCATION OF BLOCK: ,I6)					
56.		WRITE (NPRTR,173) ISET(L)					
57.	173	FCRMT (10X,30HLOCATION OF STATISTICAL ARRAY: ,I6)					
58.		J1=L+1					
59.		J2=L+6					
60.		WRITE (NPRTR,154) (ISET(J),J=J1,J2)					
61.		J=L+7					
62.		IF (ISET(J).GT.0) GO TO 52					
63.		J=-ISET(J)					
64.		WRITE (NPRTR,175) J					
65.	175	FORMAT (10X,45HTIME PERSISTANT STATISTICAL COLLECTION ON DD(					
66.		-,I2,1H))					
67.		GO TO 53					
68.	52	WRITE (NPRTR,176) ISET(J)					
69.	176	FORMAT (10X,45HTIME PERSISTANT STATISTICAL COLLECTION ON SS(					
70.		-,I2,1H))					
71.	53	J=L+8					
72.		WRITE (NPRTR,155) ISET(J)					
73.	50	CONTINUE					
74.	51	IF (NCOUN.EQ.0) GO TO 61					
75.		WRITE (NPRTR,180)					

	1	2	3	4	5	6	7
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	12345

```

76. 180  FORMAT(///10X,19HCOUNTER INFORMATION/10X,19(1H-))
77.      DC 60 I=1,NCOUN
78.      L=LFCTB+(I-1)*8
79.      WRITE(NPRTR,181) I
80. 181  FORMAT(//10X,12HCOUNTER NO.: ,I3)
81.      WRITE(NPRTR,182) L
82. 182  FORMAT(10X,26HLOCATION OF COUNTER BLOCK: ,I6)
83.      J1=L+1
84.      J2=L+6
85.      WRITE(NPRTR,154) (ISET(J),J=J1,J2)
86.      J=L+7
87.      WRITE(NPRTR,183) ISET(J)
88. 183  FORMAT(10X,7HLIMIT : ,I7)
89. 60   CONTINUE
90. 61   IF(NENT.EQ.0) GO TO 62
91.      WRITE(NPRTR,190) LFAE
92. 190  FORMAT(///10X,34HLOCATION OF FIRST AVAILABLE ENTRY: ,I6)
93.      WRITE(NPRTR,191) NENT
94. 191  FORMAT(//10X,25HNO. OF ENTRIES ALLOCATED: ,I6)
95. 62   WRITE(NPRTR,140)
96. 140  FORMAT(///10X,25HRANDOM STREAM INFORMATION/10X,25(1H-)//)
97.      DC 70 I=1,10
98.      WRITE(NPRTR,141) I,ISEED(I)
99. 141  FORMAT(10X,5HSEED( ,I2,2H)= ,2X,I10)
00. 70   CONTINUE
01.      IF (IRAN.EQ. 1) GO TO 80
02.      WRITE(NPRTR,200)
03. 200  FORMAT(/10X,42HSEEDS WILL NOT BE INITIALIZED BETWEEN RUNS)
04.      GO TO 81
05. 80   WRITE(NPRTR,201)
06. 201  FORMAT(/10X,38HSEEDS WILL BE INITIALIZED BETWEEN RUNS)
07. 81   WRITE(NPRTR,202) NRUNS
08. 202  FORMAT(///10X,8HTOTAL OF ,I4,23H RUNS WILL BE PERFORMED)
09.      RETURN
10.      END
11.      SUBROUTINE CLEARF
12.  C*
13.  C*      THIS ROUTINE CLEARS THE FILES BETWEEN THE RUNS
14.  C*
15.      DIMENSION RSET(1)
16.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCB,LFCTB,LFAE,LCAL
17.      COMMON/GEN1/NRUNS,NFILE,TREG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
18.      -NIMX,LTMX(99),NTAL,NDTPST,NCIPST,NCOUN,NATR,NENT,IECO,ISUM
19.      COMMON ISET(1)
20.      EQUIVALENCE (ISET(1),RSET(1))
21.      IF (LCAL.EQ.0) RETURN
22.      IF (NFILE.EQ.0) GO TO 5
23.      L=LFFB
24.      GO TO 6
25. 5     L=LCAL
26. 6     N=NFILE+1
27.      DO 10 K=1,N
28.          I=L+(K-1)*6
29.          ISET(I)=0
30.  C*

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
31.	C*	JOIN ALL THE ENTRIES TO THE ECCL BY TAKING THE LAST					
32.	C*	ENTRY OF THE FILE AND SETTING ITS SUCCESSOR AS LFAE					
33.	C*	AND UPDATING LFAE					
34.	C*						
35.		I1=I+1					
36.		I2=I+2					
37.		LLFE=ISET(I1)					
38.		LLLE=ISET(I2)					
39.		IF(LLFE.EQ.0) GO TO 10					
40.		J=LLLE+1					
41.		ISET(J)=LFAE					
42.		LFAE=LLFE					
43.		ISET(I1)=0					
44.		ISET(I2)=0					
45.	10	CONTINUE					
46.		RETURN					
47.		END					
48.		SUBROUTINE CLEARV					
49.		COMMON/GSC1/NCRRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,					
50.		-DDL(99),TNOW,XX(99),DTNOW,ISTCP					
51.		DC 10 I=1,99					
52.		SS(I)=0					
53.		DD(I)=0					
54.		SSL(I)=0					
55.		DDL(I)=0					
56.		XX(I)=0					
57.		ATTRIB(I)=0					
58.	10	CONTINUE					
59.		JJ=0					
60.		RETURN					
61.		END					
62.		SUBROUTINE INTAL					
63.	C*						
64.	C*	THIS IS FOR INITIALIZATION BEFORE EACH RUN					
65.	C*						
66.		DIMENSION RSET(1)					
67.		COMMON/LOC/LFI,LLR,LFPB,LPTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL					
68.		COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,					
69.		-NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM					
70.		COMMON/GEN4/ISEED(10),LSEED(10)					
71.		COMMON/GSC1/NCRRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,					
72.		-DDL(99),TNOW,XX(99),DTNOW,ISTOP					
73.		COMMON ISET(1)					
74.		EQUIVALENCE (ISET(1),RSET(1))					
75.		IF(NCOUN.EQ.0) GO TO 2					
76.		DO 1 I=1,NCOUN					
77.		LCTB=LFCTB+(I-1)*8					
78.		ISET(LCTB)=0					
79.	1	CONTINUE					
80.	2	ISTOP=0					
81.		IF(LCAL.EQ.0) GO TO 3					
82.		IF(ICLS.EQ.1) GO TO 3					
83.		CALL SCHD(-1,TCLEAR,ATTRIB)					
84.	3	IF(IRAN.EQ.0) RETURN					
85.		DC 4 I=1,10					

	1	2	3	4	5	6	7
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234

```

386.      ISEED(I)=LSEED(I)
387. 4      CONTINUE
388.      RETURN
389.      END
390.      SUBROUTINE EXEC
391.      DIMENSION A2(100),A3(100),A4(100),A5(100)
392.      DIMENSION RSET(1)
393.      COMMON/LOC/LFI,LLR,LFFP,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
394.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
395.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
396.      COMMON/GEN2/NEQD,NEQS,NEQT,DTSAB,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
397.      -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
398.      -ISCD(25)
399.      COMMON/GEN8/NIR,VALH,VALL
400.      COMMON/GSC1/NCRRD,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
401.      -DDL(99),TNOW,XX(99),DINOW,ISTCP
402.      COMMON ISET(1)
403.      EQUIVALENCE(ISET(1),RSET(1))
404.      WRITE(NPRTR,200)
405. 200      FORMAT(/10X,22HEXECUTION IS ATTEMPTED)
406.  C*
407.  C*      INITIALIZATION
408.  C*
409.      TTSAV=VALH
410.      RESLS=.01 * ABS(DTSAV)
411.      RESLS = AMIN1(DTMIN,RESLS)
412.      DTACC=DTMAX
413.      TNOW=TBEG
414.      PRMUL=.5
415.      TMUL=.9
416.      ISEES=0
417.      SCALL=1.0
418.      IF(RERR.GT.0.) SCALL=2./RERR
419.      AEM=SCALL*AERR
420.      CALL INTLC
421.      IF(NDTPST.EQ.0) GO TO 10
422.      DO 5 I=1,NDTPST
423.          LDBL=LFDSB+(I-1)*9
424.          CALL DCLCT(LDBL)
425. 5      CCNTINUE
426. 10      IF(NEQT.EQ.0) GO TO 30
427.      TLAST=TNOW
428.      DTNOW=0.0
429.      IF(NSEV.GT.0) CALL SSCND
430.      DO 15 I=1,NEQT
431.          DDL(I)=DD(I)
432.          SSL(I)=SS(I)
433.          AERR=0.0
434.          AEM=0.0
435.          AEM=ABS(DTSAV)
436.          RESLS=AMIN1(DTMIN,RESLS)
437.          DTACC=DTMAX
438.          TNOW=TBEG
439.          PRMUL=.5
440.          TMUL=.9
441.          ISEES=0
442.          SCALL=1.0
443.          IF(RERR.GT.0.) SCALL=2./RERR
444.          AEM=SCALL*AERR
445.          CALL INTLC
446.          IF(NDTPST.EQ.0) GO TO 10
447.          DO 5 I=1,NDTPST
448.              LDBL=LFDSB+(I-1)*9
449.              CALL DCLCT(LDBL)
450. 5      CCNTINUE
451.      IF(NEQT.EQ.0) GO TO 30
452.      TLAST=TNOW
453.      DTNOW=0.0
454.      IF(NSEV.GT.0) CALL SSCND
455.      DO 15 I=1,NEQT
456.          DDL(I)=DD(I)
457.          SSL(I)=SS(I)
458.          AERR=0.0
459.          AEM=0.0
460.          AEM=ABS(DTSAV)
461.          RESLS=AMIN1(DTMIN,RESLS)
462.          DTACC=DTMAX
463.          TNOW=TBEG
464.          PRMUL=.5
465.          TMUL=.9
466.          ISEES=0
467.          SCALL=1.0
468.          IF(RERR.GT.0.) SCALL=2./RERR
469.          AEM=SCALL*AERR
470.          CALL INTLC
471.          IF(NDTPST.EQ.0) GO TO 10
472.          DO 5 I=1,NDTPST
473.              LDBL=LFDSB+(I-1)*9
474.              CALL DCLCT(LDBL)
475. 5      CCNTINUE
476.      IF(NEQT.EQ.0) GO TO 30
477.      TLAST=TNOW
478.      DTNOW=0.0
479.      IF(NSEV.GT.0) CALL SSCND
480.      DO 15 I=1,NEQT
481.          DDL(I)=DD(I)
482.          SSL(I)=SS(I)
483.          AERR=0.0
484.          AEM=0.0
485.          AEM=ABS(DTSAV)
486.          RESLS=AMIN1(DTMIN,RESLS)
487.          DTACC=DTMAX
488.          TNOW=TBEG
489.          PRMUL=.5
490.          TMUL=.9
491.          ISEES=0
492.          SCALL=1.0
493.          IF(RERR.GT.0.) SCALL=2./RERR
494.          AEM=SCALL*AERR
495.          CALL INTLC
496.          IF(NDTPST.EQ.0) GO TO 10
497.          DO 5 I=1,NDTPST
498.              LDBL=LFDSB+(I-1)*9
499.              CALL DCLCT(LDBL)
500. 5      CCNTINUE
501.      IF(NEQT.EQ.0) GO TO 30
502.      TLAST=TNOW
503.      DTNOW=0.0
504.      IF(NSEV.GT.0) CALL SSCND
505.      DO 15 I=1,NEQT
506.          DDL(I)=DD(I)
507.          SSL(I)=SS(I)
508.          AERR=0.0
509.          AEM=0.0
510.          AEM=ABS(DTSAV)
511.          RESLS=AMIN1(DTMIN,RESLS)
512.          DTACC=DTMAX
513.          TNOW=TBEG
514.          PRMUL=.5
515.          TMUL=.9
516.          ISEES=0
517.          SCALL=1.0
518.          IF(RERR.GT.0.) SCALL=2./RERR
519.          AEM=SCALL*AERR
520.          CALL INTLC
521.          IF(NDTPST.EQ.0) GO TO 10
522.          DO 5 I=1,NDTPST
523.              LDBL=LFDSB+(I-1)*9
524.              CALL DCLCT(LDBL)
525. 5      CCNTINUE
526.      IF(NEQT.EQ.0) GO TO 30
527.      TLAST=TNOW
528.      DTNOW=0.0
529.      IF(NSEV.GT.0) CALL SSCND
530.      DO 15 I=1,NEQT
531.          DDL(I)=DD(I)
532.          SSL(I)=SS(I)
533.          AERR=0.0
534.          AEM=0.0
535.          AEM=ABS(DTSAV)
536.          RESLS=AMIN1(DTMIN,RESLS)
537.          DTACC=DTMAX
538.          TNOW=TBEG
539.          PRMUL=.5
540.          TMUL=.9
541.          ISEES=0
542.          SCALL=1.0
543.          IF(RERR.GT.0.) SCALL=2./RERR
544.          AEM=SCALL*AERR
545.          CALL INTLC
546.          IF(NDTPST.EQ.0) GO TO 10
547.          DO 5 I=1,NDTPST
548.              LDBL=LFDSB+(I-1)*9
549.              CALL DCLCT(LDBL)
550. 5      CCNTINUE
551.      IF(NEQT.EQ.0) GO TO 30
552.      TLAST=TNOW
553.      DTNOW=0.0
554.      IF(NSEV.GT.0) CALL SSCND
555.      DO 15 I=1,NEQT
556.          DDL(I)=DD(I)
557.          SSL(I)=SS(I)
558.          AERR=0.0
559.          AEM=0.0
560.          AEM=ABS(DTSAV)
561.          RESLS=AMIN1(DTMIN,RESLS)
562.          DTACC=DTMAX
563.          TNOW=TBEG
564.          PRMUL=.5
565.          TMUL=.9
566.          ISEES=0
567.          SCALL=1.0
568.          IF(RERR.GT.0.) SCALL=2./RERR
569.          AEM=SCALL*AERR
570.          CALL INTLC
571.          IF(NDTPST.EQ.0) GO TO 10
572.          DO 5 I=1,NDTPST
573.              LDBL=LFDSB+(I-1)*9
574.              CALL DCLCT(LDBL)
575. 5      CCNTINUE
576.      IF(NEQT.EQ.0) GO TO 30
577.      TLAST=TNOW
578.      DTNOW=0.0
579.      IF(NSEV.GT.0) CALL SSCND
580.      DO 15 I=1,NEQT
581.          DDL(I)=DD(I)
582.          SSL(I)=SS(I)
583.          AERR=0.0
584.          AEM=0.0
585.          AEM=ABS(DTSAV)
586.          RESLS=AMIN1(DTMIN,RESLS)
587.          DTACC=DTMAX
588.          TNOW=TBEG
589.          PRMUL=.5
590.          TMUL=.9
591.          ISEES=0
592.          SCALL=1.0
593.          IF(RERR.GT.0.) SCALL=2./RERR
594.          AEM=SCALL*AERR
595.          CALL INTLC
596.          IF(NDTPST.EQ.0) GO TO 10
597.          DO 5 I=1,NDTPST
598.              LDBL=LFDSB+(I-1)*9
599.              CALL DCLCT(LDBL)
600. 5      CCNTINUE
601.      IF(NEQT.EQ.0) GO TO 30
602.      TLAST=TNOW
603.      DTNOW=0.0
604.      IF(NSEV.GT.0) CALL SSCND
605.      DO 15 I=1,NEQT
606.          DDL(I)=DD(I)
607.          SSL(I)=SS(I)
608.          AERR=0.0
609.          AEM=0.0
610.          AEM=ABS(DTSAV)
611.          RESLS=AMIN1(DTMIN,RESLS)
612.          DTACC=DTMAX
613.          TNOW=TBEG
614.          PRMUL=.5
615.          TMUL=.9
616.          ISEES=0
617.          SCALL=1.0
618.          IF(RERR.GT.0.) SCALL=2./RERR
619.          AEM=SCALL*AERR
620.          CALL INTLC
621.          IF(NDTPST.EQ.0) GO TO 10
622.          DO 5 I=1,NDTPST
623.              LDBL=LFDSB+(I-1)*9
624.              CALL DCLCT(LDBL)
625. 5      CCNTINUE
626.      IF(NEQT.EQ.0) GO TO 30
627.      TLAST=TNOW
628.      DTNOW=0.0
629.      IF(NSEV.GT.0) CALL SSCND
630.      DO 15 I=1,NEQT
631.          DDL(I)=DD(I)
632.          SSL(I)=SS(I)
633.          AERR=0.0
634.          AEM=0.0
635.          AEM=ABS(DTSAV)
636.          RESLS=AMIN1(DTMIN,RESLS)
637.          DTACC=DTMAX
638.          TNOW=TBEG
639.          PRMUL=.5
640.          TMUL=.9
641.          ISEES=0
642.          SCALL=1.0
643.          IF(RERR.GT.0.) SCALL=2./RERR
644.          AEM=SCALL*AERR
645.          CALL INTLC
646.          IF(NDTPST.EQ.0) GO TO 10
647.          DO 5 I=1,NDTPST
648.              LDBL=LFDSB+(I-1)*9
649.              CALL DCLCT(LDBL)
650. 5      CCNTINUE
651.      IF(NEQT.EQ.0) GO TO 30
652.      TLAST=TNOW
653.      DTNOW=0.0
654.      IF(NSEV.GT.0) CALL SSCND
655.      DO 15 I=1,NEQT
656.          DDL(I)=DD(I)
657.          SSL(I)=SS(I)
658.          AERR=0.0
659.          AEM=0.0
660.          AEM=ABS(DTSAV)
661.          RESLS=AMIN1(DTMIN,RESLS)
662.          DTACC=DTMAX
663.          TNOW=TBEG
664.          PRMUL=.5
665.          TMUL=.9
666.          ISEES=0
667.          SCALL=1.0
668.          IF(RERR.GT.0.) SCALL=2./RERR
669.          AEM=SCALL*AERR
670.          CALL INTLC
671.          IF(NDTPST.EQ.0) GO TO 10
672.          DO 5 I=1,NDTPST
673.              LDBL=LFDSB+(I-1)*9
674.              CALL DCLCT(LDBL)
675. 5      CCNTINUE
676.      IF(NEQT.EQ.0) GO TO 30
677.      TLAST=TNOW
678.      DTNOW=0.0
679.      IF(NSEV.GT.0) CALL SSCND
680.      DO 15 I=1,NEQT
681.          DDL(I)=DD(I)
682.          SSL(I)=SS(I)
683.          AERR=0.0
684.          AEM=0.0
685.          AEM=ABS(DTSAV)
686.          RESLS=AMIN1(DTMIN,RESLS)
687.          DTACC=DTMAX
688.          TNOW=TBEG
689.          PRMUL=.5
690.          TMUL=.9
691.          ISEES=0
692.          SCALL=1.0
693.          IF(RERR.GT.0.) SCALL=2./RERR
694.          AEM=SCALL*AERR
695.          CALL INTLC
696.          IF(NDTPST.EQ.0) GO TO 10
697.          DO 5 I=1,NDTPST
698.              LDBL=LFDSB+(I-1)*9
699.              CALL DCLCT(LDBL)
700. 5      CCNTINUE
701.      IF(NEQT.EQ.0) GO TO 30
702.      TLAST=TNOW
703.      DTNOW=0.0
704.      IF(NSEV.GT.0) CALL SSCND
705.      DO 15 I=1,NEQT
706.          DDL(I)=DD(I)
707.          SSL(I)=SS(I)
708.          AERR=0.0
709.          AEM=0.0
710.          AEM=ABS(DTSAV)
711.          RESLS=AMIN1(DTMIN,RESLS)
712.          DTACC=DTMAX
713.          TNOW=TBEG
714.          PRMUL=.5
715.          TMUL=.9
716.          ISEES=0
717.          SCALL=1.0
718.          IF(RERR.GT.0.) SCALL=2./RERR
719.          AEM=SCALL*AERR
720.          CALL INTLC
721.          IF(NDTPST.EQ.0) GO TO 10
722.          DO 5 I=1,NDTPST
723.              LDBL=LFDSB+(I-1)*9
724.              CALL DCLCT(LDBL)
725. 5      CCNTINUE
726.      IF(NEQT.EQ.0) GO TO 30
727.      TLAST=TNOW
728.      DTNOW=0.0
729.      IF(NSEV.GT.0) CALL SSCND
730.      DO 15 I=1,NEQT
731.          DDL(I)=DD(I)
732.          SSL(I)=SS(I)
733.          AERR=0.0
734.          AEM=0.0
735.          AEM=ABS(DTSAV)
736.          RESLS=AMIN1(DTMIN,RESLS)
737.          DTACC=DTMAX
738.          TNOW=TBEG
739.          PRMUL=.5
740.          TMUL=.9
741.          ISEES=0
742.          SCALL=1.0
743.          IF(RERR.GT.0.) SCALL=2./RERR
744.          AEM=SCALL*AERR
745.          CALL INTLC
746.          IF(NDTPST.EQ.0) GO TO 10
747.          DO 5 I=1,NDTPST
748.              LDBL=LFDSB+(I-1)*9
749.              CALL DCLCT(LDBL)
750. 5      CCNTINUE
751.      IF(NEQT.EQ.0) GO TO 30
752.      TLAST=TNOW
753.      DTNOW=0.0
754.      IF(NSEV.GT.0) CALL SSCND
755.      DO 15 I=1,NEQT
756.          DDL(I)=DD(I)
757.          SSL(I)=SS(I)
758.          AERR=0.0
759.          AEM=0.0
760.          AEM=ABS(DTSAV)
761.          RESLS=AMIN1(DTMIN,RESLS)
762.          DTACC=DTMAX
763.          TNOW=TBEG
764.          PRMUL=.5
765.          TMUL=.9
766.          ISEES=0
767.          SCALL=1.0
768.          IF(RERR.GT.0.) SCALL=2./RERR
769.          AEM=SCALL*AERR
770.          CALL INTLC
771.          IF(NDTPST.EQ.0) GO TO 10
772.          DO 5 I=1,NDTPST
773.              LDBL=LFDSB+(I-1)*9
774.              CALL DCLCT(LDBL)
775. 5      CCNTINUE
776.      IF(NEQT.EQ.0) GO TO 30
777.      TLAST=TNOW
778.      DTNOW=0.0
779.      IF(NSEV.GT.0) CALL SSCND
780.      DO 15 I=1,NEQT
781.          DDL(I)=DD(I)
782.          SSL(I)=SS(I)
783.          AERR=0.0
784.          AEM=0.0
785.          AEM=ABS(DTSAV)
786.          RESLS=AMIN1(DTMIN,RESLS)
787.          DTACC=DTMAX
788.          TNOW=TBEG
789.          PRMUL=.5
790.          TMUL=.9
791.          ISEES=0
792.          SCALL=1.0
793.          IF(RERR.GT.0.) SCALL=2./RERR
794.          AEM=SCALL*AERR
795.          CALL INTLC
796.          IF(NDTPST.EQ.0) GO TO 10
797.          DO 5 I=1,NDTPST
798.              LDBL=LFDSB+(I-1)*9
799.              CALL DCLCT(LDBL)
800. 5      CCNTINUE
801.      IF(NEQT.EQ.0) GO TO 30
802.      TLAST=TNOW
803.      DTNOW=0.0
804.      IF(NSEV.GT.0) CALL SSCND
805.      DO 15 I=1,NEQT
806.          DDL(I)=DD(I)
807.          SSL(I)=SS(I)
808.          AERR=0.0
809.          AEM=0.0
810.          AEM=ABS(DTSAV)
811.          RESLS=AMIN1(DTMIN,RESLS)
812.          DTACC=DTMAX
813.          TNOW=TBEG
814.          PRMUL=.5
815.          TMUL=.9
816.          ISEES=0
817.          SCALL=1.0
818.          IF(RERR.GT.0.) SCALL=2./RERR
819.          AEM=SCALL*AERR
820.          CALL INTLC
821.          IF(NDTPST.EQ.0) GO TO 10
822.          DO 5 I=1,NDTPST
823.              LDBL=LFDSB+(I-1)*9
824.              CALL DCLCT(LDBL)
825. 5      CCNTINUE
826.      IF(NEQT.EQ.0) GO TO 30
827.      TLAST=TNOW
828.      DTNOW=0.0
829.      IF(NSEV.GT.0) CALL SSCND
830.      DO 15 I=1,NEQT
831.          DDL(I)=DD(I)
832.          SSL(I)=SS(I)
833.          AERR=0.0
834.          AEM=0.0
835.          AEM=ABS(DTSAV)
836.          RESLS=AMIN1(DTMIN,RESLS)
837.          DTACC=DTMAX
838.          TNOW=TBEG
839.          PRMUL=.5
840.          TMUL=.9
841.          ISEES=0
842.          SCALL=1.0
843.          IF(RERR.GT.0.) SCALL=2./RERR
844.          AEM=SCALL*AERR
845.          CALL INTLC
846.          IF(NDTPST.EQ.0) GO TO 10
847.          DO 5 I=1,NDTPST
848.              LDBL=LFDSB+(I-1)*9
849.              CALL DCLCT(LDBL)
850. 5      CCNTINUE
851.      IF(NEQT.EQ.0) GO TO 30
852.      TLAST=TNOW
853.      DTNOW=0.0
854.      IF(NSEV.GT.0) CALL SSCND
855.      DO 15 I=1,NEQT
856.          DDL(I)=DD(I)
857.          SSL(I)=SS(I)
858.          AERR=0.0
859.          AEM=0.0
860.          AEM=ABS(DTSAV)
861.          RESLS=AMIN1(DTMIN,RESLS)
862.          DTACC=DTMAX
863.          TNOW=TBEG
864.          PRMUL=.5
865.          TMUL=.9
866.          ISEES=0
867.          SCALL=1.0
868.          IF(RERR.GT.0.) SCALL=2./RERR
869.          AEM=SCALL*AERR
870.          CALL INTLC
871.          IF(NDTPST.EQ.0) GO TO 10
872.          DO 5 I=1,NDTPST
873.              LDBL=LFDSB+(I-1)*9
874.              CALL DCLCT(LDBL)
875. 5      CCNTINUE
876.      IF(NEQT.EQ.0) GO TO 30
877.      TLAST=TNOW
878.      DTNOW=0.0
879.      IF(NSEV.GT.0) CALL SSCND
880.      DO 15 I=1,NEQT
881.          DDL(I)=DD(I)
882.          SSL(I)=SS(I)
883.          AERR=0.0
884.          AEM=0.0
885.          AEM=ABS(DTSAV)
886.          RESLS=AMIN1(DTMIN,RESLS)
887.          DTACC=DTMAX
888.          TNOW=TBEG
889.          PRMUL=.5
890.          TMUL=.9
891.          ISEES=0
892.          SCALL=1.0
893.          IF(RERR.GT.0.) SCALL=2./RERR
894.          AEM=SCALL*AERR
895.          CALL INTLC
896.          IF(NDTPST.EQ.0) GO TO 10
897.          DO 5 I=1,NDTPST
898.              LDBL=LFDSB+(I-1)*9
899.              CALL DCLCT(LDBL)
900. 5      CCNTINUE
901.      IF(NEQT.EQ.0) GO TO 30
902.      TLAST=TNOW
903.      DTNOW=0.0
904.      IF(NSEV.GT.0) CALL SSCND
905.      DO 15 I=1,NEQT
906.          DDL(I)=DD(I)
907.          SSL(I)=SS(I)
908.          AERR=0.0
909.          AEM=0.0
910.          AEM=ABS(DTSAV)
911.          RESLS=AMIN1(DTMIN,RESLS)
912.          DTACC=DTMAX
913.          TNOW=TBEG
914.          PRMUL=.5
915.          TMUL=.9
916.          ISEES=0
917.          SCALL=1.0
918.          IF(RERR.GT.0.) SCALL=2./RERR
919.          AEM=SCALL*AERR
920.          CALL INTLC
921.          IF(NDTPST.EQ.0) GO TO 10
922.          DO 5 I=1,NDTPST
923.              LDBL=LFDSB+(I-1)*9
924.              CALL DCLCT(LDBL)
925. 5      CCNTINUE
926.      IF(NEQT.EQ.0) GO TO 30
927.      TLAST=TNOW
928.      DTNOW=0.0
929.      IF(NSEV.GT.0) CALL SSCND
930.      DO 15 I=1,NEQT
931.          DDL(I)=DD(I)
932.          SSL(I)=SS(I)
933.          AERR=0.0
934.          AEM=0.0
935.          AEM=ABS(DTSAV)
936.          RESLS=AMIN1(DTMIN,RESLS)
937.          DTACC=DTMAX
938.          TNOW=TBEG
939.          PRMUL=.5
940.          TMUL=.9
941.          ISEES=0
942.          SCALL=1.0
943.          IF(RERR.GT.0.) SCALL=2./RERR
944.          AEM=SCALL*AERR
945.          CALL INTLC
946.          IF(NDTPST.EQ.0) GO TO 10
947.          DO 5 I=1,NDTPST
948.              LDBL=LFDSB+(I-1)*9
949.              CALL DCLCT(LDBL)
950. 5      CCNTINUE
951.      IF(NEQT.EQ.0) GO TO 30
952.      TLAST=TNOW
953.      DTNOW=0.0
954.      IF(NSEV.GT.0) CALL SSCND
955.      DO 15 I=1,NEQT
956.          DDL(I)=DD(I)
957.          SSL(I)=SS(I)
958.          AERR=0.0
959.          AEM=0.0
960.          AEM=ABS(DTSAV)
961.          RESLS=AMIN1(DTMIN,RESLS)
962.          DTACC=DTMAX
963.          TNOW=TBEG
964.          PRMUL=.5
965.          TMUL=.9
966.          ISEES=0
967.          SCALL=1.0
968.          IF(RERR.GT.0.) SCALL=2./RERR
969.          AEM=SCALL*AERR
970.          CALL INTLC
971.          IF(NDTPST.EQ.0) GO TO 10
972.          DO 5 I=1,NDTPST
973.              LDBL=LFDSB+(I-1)*9
974.              CALL DCLCT(LDBL)
975. 5      CCNTINUE
976.      IF(NEQT.EQ.0) GO TO 30
977.      TLAST=TNOW
978.      DTNOW=0.0
979.      IF(NSEV.GT.0) CALL SSCND
980.      DO 15 I=1,NEQT
981.          DDL(I)=DD(I)
982.          SSL(I)=SS(I)
983.          AERR=0.0
984.          AEM=0.0
985.          AEM=ABS(DTSAV)
986.          RESLS=AMIN1(DTMIN,RESLS)
987.          DTACC=DTMAX
988.          TNOW=TBEG
989.          PRMUL=.5
990.          TMUL=.9
991.          ISEES=0
992.          SCALL=1.0
993.          IF(RERR.GT.0.) SCALL=2./RERR
994.          AEM=SCALL*AERR
995.          CALL INTLC
996.          IF(NDTPST.EQ.0) GO TO 10
997.          DO 5 I=1,NDTPST
998.              LDBL=LFDSB+(I-1)*9
999.              CALL DCLCT(LDBL)
1000. 5      CCNTINUE
1001.      IF(NEQT.EQ.0) GO TO 30
1002.      TLAST=TNOW
1003.      DTNOW=0.0
1004.      IF(NSEV.GT.0) CALL SSCND
1005.      DO 15 I=1,NEQT
1006.          DDL(I)=DD(I)
1007.          SSL(I)=SS(I)
1008.          AERR=0.0
1009.          AEM=0.0
1010.          AEM=ABS(DTSAV)
1011.          RESLS=AMIN1(DTMIN,RESLS)
1012.          DTACC=DTMAX
1013.          TNOW=TBEG
1014.          PRMUL=.5
1015.          TMUL=.9
1016.          ISEES=0
1017.          SCALL=1.0
1018.          IF(RERR.GT.0.) SCALL=2./RERR
1019.          AEM=SCALL*AERR
1020.          CALL INTLC
1021.          IF(NDTPST.EQ.0) GO TO 10
1022.          DO 5 I=1,NDTPST
1023.              LDBL=LFDSB+(I-1)*9
1024.              CALL DCLCT(LDBL)
1025. 5      CCNTINUE
1026.      IF(NEQT.EQ.0) GO TO 30
1027.      TLAST=TNOW
1028.      DTNOW=0.0
1029.      IF(NSEV.GT.0) CALL SSCND
1030.      DO 15 I=1,NEQT
1031.          DDL(I)=DD(I)
1032.          SSL(I)=SS(I)
1033.          AERR=0.0
1034.          AEM=0.0
1035.          AEM=ABS(DTSAV)
1036.          RESLS=AMIN1(DTMIN,RESLS)
1037.          DTACC=DTMAX
1038.          TNOW=TBEG
1039.          PRMUL=.5
1040.          TMUL=.9
1041.          ISEES=0
1042.          SCALL=1.0
1043.          IF(RERR.GT.0.) SCALL=2./RERR
1044.          AEM=SCALL*AERR
1045.          CALL INTLC
1046.          IF(NDTPST.EQ.0) GO TO 10
1047.          DO 5 I=1,NDTPST
1048.              LDBL=LFDSB+(I-1)*9
1049.              CALL DCLCT(LDBL)
1050. 5      CCNTINUE
1051.      IF(NEQT.EQ.0) GO TO 30
1052.      TLAST=TNOW
1053.      DTNOW=0.0
1054.      IF(NSEV.GT.0) CALL SSCND
1055.      DO 15 I=1,NEQT
1056.          DDL(I)=DD(I)
1057.          SSL(I)=SS(I)
1058.          AERR=0.0
1059.          AEM=0.0
1060.          AEM=ABS(DTSAV)
1061.          RESLS=AMIN1(DTMIN,RESLS)
1062.          DTACC=DTMAX
1063.          TNOW=TBEG
1064.          PRMUL=.5
1065.          TMUL=.9
1066.          ISEES=0
1067.          SCALL=1.0
1068.          IF(RERR.GT.0.) SCALL=2./RERR
1069.          AEM=SCALL*AERR
1070.          CALL INTLC
1071.          IF(NDTPST.EQ.0) GO TO 10
1072.          DO 5 I=1,NDTPST
1073.              LDBL=LFDSB+(I-1)*9
1074.              CALL DCLCT(LDBL)
1075. 5      CCNTINUE
1076.      IF(NEQT.EQ.0) GO TO 30
1077.      TLAST=TNOW
1078.      DTNOW=0.0
1079.      IF(NSEV.GT.0) CALL SSCND
1080.      DO 15 I=1,NEQT
1081.          DDL(I)=DD(I)
1082.          SSL(I)=SS(I)
1083.          AERR=0.0
1084.          AEM=0.0
1085.          AEM=ABS(DTSAV)
1086.          RESLS=AMIN1(DTMIN,RESLS)
1087.          DTACC=DTMAX
1088.          TNOW=TBEG
1089.          PRMUL=.5
1090.          TMUL=.9
1091.          ISEES=0
1092.          SCALL=1.0
1093.          IF(RERR.GT.0.) SCALL=2./RERR
1094.          AEM=SCALL*AERR
1095.          CALL INTLC
1096.          IF(NDTPST.EQ.0) GO TO 10
1097.          DO 5 I=1,NDTPST
1098.              LDBL=LFDSB+(I-1)*9
1099.              CALL DCLCT(LDBL)
1100. 5      CCNTINUE
1101.      IF(NEQT.EQ.0) GO TO 30
1102.      TLAST=TNOW
1103.      DTNOW=0.0
1104.      IF(NSEV.GT.0) CALL SSCND
1105.      DO 15 I=1,NEQT
1106.          DDL(I)=DD(I)
1107.          SSL(I)=SS(I)
1108.          AERR=0.0
1109.          AEM=0.0
1110.          AEM=ABS(DTSAV)
1111.          RESLS=AMIN1(DTMIN,RESLS)
1112.          DTACC=DTMAX
1113.          TNOW=TBEG
1114.          PRMUL=.5
1115.          TMUL=.9
1116.          ISEES=0
1117.          SCALL=1.0
1118.          IF(RERR.GT.0.) SCALL=2./RERR
1119.          AEM=SCALL*AERR
1120.          CALL INTLC
1121.          IF(NDTPST.EQ.0) GO TO 10
1122.          DO 5 I=1,NDTPST
1123.              LDBL=LFDSB+(I-1)*9
1124.              CALL DCLCT(LDBL)
1125. 5      CCNTINUE
1126.      IF(NEQT.EQ.0) GO TO 30
1127.      TLAST=TNOW
1128.      DTNOW=0.0
1129.      IF(NSEV.GT.0) CALL SSCND
1130.      DO 15 I=1,NEQT
1131.          DDL(I)=DD(I)
1132.          SSL(I)=SS(I)
1133.          AERR=0.0
1134.          AEM=0.0
1135.          AEM=ABS(DTSAV)
1136.          RESLS=AMIN1(DTMIN,RESLS)
1137.          DTACC=DTMAX
1138.          TNOW=TBEG
1139.          PRMUL=.5
1140.          TMUL=.9
1141.          ISEES=0
1142.          SCALL=1.0
1143.          IF(RERR.GT.0.) SCALL=2./RERR
1144.          AEM=SCALL*AERR
1145.          CALL INTLC
1146.          IF(NDTPST.EQ.0) GO TO 10
1147.          DO 5 I=1,NDTPST
1148.              LDBL=LFDSB+(I-1)*9
1149.              CALL DCLCT(LDBL)
1150. 5      CCNTINUE
1151.      IF(NEQT.EQ.0) GO TO 30
1152.      TLAST=TNOW
1153.      DTNOW=0.0
1154.      IF(NSEV.GT.0) CALL SSCND
1155.      DO 15 I=1,NEQT
1156.          DDL(I)=DD(I)
1157.          SSL(I)=SS(I)
1158.          AERR=0.0
1159.          AEM=0.0
1160.          AEM=ABS(DTSAV)
1161.          RESLS=AMIN1(DTMIN,RESLS)
1162.          DTACC=DTMAX
1163.          TNOW=TBEG
1164.          PRMUL=.5
1165.          TMUL=.9
1166.          ISEES=0
1167.          SCALL=1.0
1168.          IF(RERR.GT.0.) SCALL=2./RERR
1169.          AEM=SCALL*AERR
1170.          CALL INTLC
1171.          IF(NDTPST.EQ.0) GO TO 10
1172.          DO 5 I=1,NDTPST
1173.              LDBL=LFDSB+(I-1)*9
1174.              CALL DCLCT(LDBL)
1175. 5      CCNTINUE
1176.      IF(NEQT.EQ.0) GO TO 30
1177.      TLAST=TNOW
1178.      DTNOW=0.0
1179.      IF(NSEV.GT.0) CALL SSCND
1180.      DO 15 I=1,NEQT
1181.          DDL(I)=DD(I)
1182.          SSL(I)=SS(I)
1183.          AERR=0.0
1184.          AEM=0.0
1185.          AEM=ABS(DTSAV)
1186.          RESLS=AMIN1(DTMIN,RESLS)
1187.          DTACC=DTMAX
1188.          TNOW=TBEG
1189.          PRMUL=.5
1190.          TMUL=.9
1191.          ISEES=0
1192.          SCALL=1.0
1193.          IF(RERR.GT.0.) SCALL=2./RERR
1194.          AEM=SCALL*AERR
1195.          CALL INTLC
1196.          IF(NDTPST.EQ.0) GO TO 10
1197.          DO 5 I=1,NDTPST
1198.              LDBL=LFDSB+(I-1)*9
1199.              CALL DCLCT(LDBL)
1200. 5      CCNTINUE
1201.      IF(NEQT.EQ.0) GO TO 30
1202.      TLAST=TNOW
1203.      DTNOW=0.0
1204.      IF(NSEV.GT.0) CALL SSCND
1205.      DO 15 I=1,NEQT
1206.          DDL(I)=DD(I)
1207.          SSL(I)=SS(I)
1208.          AERR=0.0
1209.          AEM=0.0
1210.          AEM=ABS(DTSAV)
1211.          RESLS=AMIN1(DTMIN,RESLS)
1212.          DTACC=DTMAX
1213.          TNOW=TBEG
1214.          PRMUL=.5
1215.          TMUL=.9
1216.          ISEES=0
1217.          SCALL=1.0
1218.          IF(RERR.GT.0.) SCALL=2./RERR
12
```

	1	2	3	4	5	6	7
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234

```

441.          IF (ISCD(I).EQ.0.OR.ISCD(J)*ICDIR(I).LT.0) GO TO 24
442.          IX=ICSV(I)
443.          CALL SCHD(IX,TNOW,ATRIE)
444.  24       ISCD(I)=0
445.  25       CONTINUE
446.  C*
447.  C*       NEW ITERATION
448.  C*
449.  30       TNEXT=VALH
450.          IF(LCAL.EQ.0) GO TO 40
451.          IF(ISET(LCAL).EQ.0) GO TO 35
452.          J=LCAL+1
453.          LIFE=ISET(J)
454.          LATR=ISET(LIFE+3)
455.          TNEXT=RSET(LATR)
456.  35       IF(NEQT.GT.0) GO TO 40
457.          IF(TNEXT.GT.TFIN. OR.ISTOP.EQ.1) GO TO 1000
458.          TNOW=TNEXT
459.          CALL DISC
460.          GC TO 30
461.  40       IF(TNEXT.GT.TFIN.AND.TNOW.EQ.TFIN) GO TO 1070
462.          IF(TNEXT.GT.TFIN) TNEXT=TFIN
463.  45       IF(ISTOP.EQ.1) GO TO 1000
464.  C*
465.  C*       SAVE CONTINUOUS VARIABLES IF IT IS SAVING TIME
466.  C*
467.          IF(ABS(TTSAV-TNOW)-RESLS) 50,50,55
468.  50       CALL SSAVE
469.          TTSAV=TTSAV+ABS(DTSAV)
470.  55       IF (TNEXT-TNOW) 60,65,70
471.  60       WRITE(NPRTR,960) TNEXT,TNOW
472.          CALL ERROR(9,1)
473.  65       IF(DTSAV.LT.0) CALL SSAVE
474.          CALL DISC
475.          IF(DTSAV.GE.0) GO TO 30
476.  C*
477.  C*       SAVE THE VALUE OF THE CTPST VAR. IF THEY HAVE CHANGED
478.  C*       IN THE EVENT
479.  C*
480.          DC 68 I=1,NCTPST
481.          J=LPCSB+(I-1)*9
482.          LSTAT=ISET(J)
483.          ITYPE=ISET(J+7)
484.          IF(ITYPE.LT.0) GO TO 66
485.          X=SS(ITYPE)
486.          GC TO 67
487.  66       ITYPE=-ITYPE
488.          X=DD(ITYPE)
489.  67       IF(X.EQ.RSET(LSTAT)) GO TO 68
490.          NU=ISET(J+8)
491.          CALL CTPST(X,LSTAT,NU)
492.  68       CONTINUE
493.          GO TO 30
494.  70       IF(NEQD.GT.0) GO TO 75
495.          DTFUL=DTMAX

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

496.      GC TO 80
497.      75      DTFUL=DTACC
498.      80      IF (TLAST+DTFUL.GT.TTSAV) DTFUL=TTSAV-TLAST
499.      IF (TLAST+DTFUL.GT.TNEXT) DTFUL=TNEXT-TLAST
500.      IF (NEQD.GT.0) GO TO 240
501.      85      DTNOW=DTFUL
502.      TNOW=TLAST+DTNOW
503.      CALL STATE
504.      90      IF (NSEV.GT.0) CALL SSCND
505.      IF (ISEES) 380,95,450
506.      C*
507.      C*      NC STATE EVENT HAS OCCURED
508.      C*
509.      95      TLAST=TNOW
510.      DO 100 I=1,NEQT
511.      DDL(I)=DD(I)
512.      SSL(I)=SS(I)
513.      100     CCNTINUE
514.      IF (TNOW.EQ.TPIN) GO TO 30
515.      GO TO 45
516.      C*
517.      C*      STATE EVENT HAS PASSED; REDUCE THE STEP SIZE
518.      C*
519.      380     IF (DTFUL-DTMIN) 381,381,382
520.      381     IF (IERR) 450,401,401
521.      401     IF (ISEES+1000) 410,400,400
522.      400     I=-ISEES
523.      WRITE(NPRTR,980) I,TNOW
524.      GC TO 420
525.      410     I=-ISEES-1000
526.      WRITE(NPRTR,990) I,TNOW
527.      420     IF (IERR.EQ.0) GO TO 450
528.      CALL ERROR (17,1)
529.      382     DTFUL=FRMUL*DTFUL
530.      IF (DTFUL.LT.DTMIN) DTFUL=DTMIN
531.      ISEES=0
532.      GO TO 240
533.      C*
534.      C*      AT LEAST ONE STATE EVENT HAS OCCURED; SCHEDULE THEM ON
535.      C*      THE CALENDER ; ACCEPT THE SS AND DD VALUES; START A NEW STEP
536.      C*
537.      450     ISEES=0
538.      DC 451 I=1,NSEV
539.      IF (ISCD(I).EQ.0.OR.ISCD(I)*ICDIR(I).LT.0) GO TO 453
540.      IX=ICSV(I)
541.      CALL SCHD (IX,TNOW,ATRI)
542.      453     ISCD(I)=0
543.      451     CCNTINUE
544.      TLAST=TNOW
545.      DC 452 I=1,NEQT
546.      SSL(I)=SS(I)
547.      DDL(I)=DD(I)
548.      452     CONTINUE
549.      GC TO 30
550.      C*

```

		1	2	3	4	5	6	7
		1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234
551.	C*	RUNGE-KUTA INTEGRATION						
552.	C*							
553.	240	DTNOW=.25*DTFUL						
554.		DC 250 I=1,NEQD						
555.	250	SS(I)=SSL(I)+DTNOW*DDL(I)						
556.		TNOW=TLAST+DTNOW						
557.		CALL STATE						
558.		CH=3.0*DTFUL/32.0						
559.		DO 260 I=1,NEQD						
560.		A2(I)=DD(I)						
561.	260	SS(I)=SSL(I)+CH*(DDL(I)+3.0*A2(I))						
562.		DTNOW=.375*DTFUL						
563.		TNOW=TLAST+DTNOW						
564.		CALL STATE						
565.		CH=DTFUL/2197.0						
566.		DO 270 I=1,NEQD						
567.		A3(I)=DD(I)						
568.	270	SS(I)=SSL(I)+CH*(1932.0*DDL(I)+(7296.0*A3(I)-7200.0*A2(I)))						
569.		DTNOW=12.0*DTFUL/13.0						
570.		TNOW=TLAST+DTNOW						
571.		CALL STATE						
572.		CH=DTFUL/4104.						
573.		DO 280 I=1,NEQD						
574.		A4(I)=DD(I)						
575.	280	SS(I)=SSL(I)+CH*((8341.0*DDL(I)-845.0*A4(I))+(29440.0*A3(I)-32832.0*A2(I)))						
576.		DTNOW=DTFUL						
577.		TNOW=TLAST+DTNOW						
578.		CALL STATE						
579.		CH=DTFUL/20520.0						
580.		DO 290 I=1,NEQD						
581.		A5(I)=DD(I)						
582.	290	SS(I)=SSL(I)+CH*((-6080.0*DDL(I)+(9295.0*A4(I)-5643.0*A5(I)))+(41040.0*A2(I)-28352.0*A3(I)))						
583.		DTNOW=.5*DTFUL						
584.		TNOW=TLAST+DTNOW						
585.		CALL STATE						
586.		CH=DTFUL/7618050.0						
587.		EEOET=0.0						
588.		DO 300 I=1,NEQD						
589.		SS(I)=SSL(I)+CH*((902880.0*DDL(I)+(3855735.0*A4(I)-1371249.0*A5(I)))+(3953664.0*A3(I)+277020.0*DD(I)))						
590.		IF(DTFUL.LT.DTMIN) GO TO 300						
591.		TERR=ABS(SSL(I))+ABS(SS(I))+AEM						
592.		IF(TERR.LE.0.0) GO TO 300						
593.		EERR=ABS((-2090.0*DDL(I)+(21970.0*A4(I)-15048.0*A5(I)))+(22528.0*A3(I)-27360.0*DD(I)))						
594.		IF(EEOET.GE.EERR/TERR) GO TO 300						
595.		EEOET=EERR/TERR						
596.		IIR=I						
597.	300	CONTINUE						
598.		IF(DTFUL.LT.DTMIN) GO TO 85						
599.		ESTOL=DTFUL*EEOET*SCALL/752400.0						
600.		IF(ESTOL.LE.1.0) GO TO 310						
601.	C*							

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

606. C*   ACCURACY IS NOT MET; MUST REDUCE THE STEP SIZE
607. C*
608.   IF (DTFUL.LE.DTMIN) GO TO 330
609.   FRACI=.1
610.   IF (ESTOL.LT.59049.0) FRACI=TMUL/ESTOL**.2
611.   DTACC=FRACI*DTFUL
612.   IF (DTACC.LT.DTMIN) DTACC=DTMIN
613.   DTFUL=DTACC
614.   GO TO 240
615. C*
616. C*   ACCURACY IS ACCEPTABLE; INCREASE THE STEP SIZE
617. C*
618. 310  FRACI=5.0
619.   IF (ESTOL.GT.1.889568E-4) FRACI=TMUL/ESTOL**.2
620.   DTACC=FRACI*DTFUL
621.   IF (DTACC.GT.DTMAX) DTACC=DTMAX
622.   IF (DTACC.LT.DTMIN) DTACC=DTMIN
623.   GO TO 85
624. 330  IF(IERR) 90,350,340
625. 340  WRITE(NPRTR,970) ILR,TNOW
626.   CALL ERROR(18,1)
627. 350  WRITE(NPRTR,970) ILR,TNOW
628.   GO TO 85
629. C*
630. C*   RUN IS COMPLETED
631. C*
632. 1000 IF(NDTPST.EQ.0) GO TO 1003
633. 1001 DO 1002 I=1,NDTPST
634.     LDBL=LPDSB+(I-1)*9
635.     CALL DCLCT(LDBL)
636. 1002 CONTINUE
637. 1003 IF(NCTPST.EQ.0) RETURN
638.   CALL SSAVE
639.   RETURN
640. 960  FORMAT (///36X,6HTNEXT=,E17.9,5X,5HTNOW=,E17.9)
641. 970  FORMAT (/2X,38HSPECIFIED LOCAL ERROR EXCEEDED FOR SS(,I3,
642. -9H) AT TIME,E12.4)
643. 980  FORMAT (/2X,38HSPECIFIED TOLERANCE EXCEEDED FOR SS(,I3,
644. -9H) AT TIME,E12.4)
645. 990  FORMAT (/2X,38HSPECIFIED TOLERANCE EXCEEDED FOR DD(,I3,
646. -9H) AT TIME,E12.4)
647.   END
648.   SUBROUTINE DISC
649.   DIMENSION RSET(1)
650.   COMMON/LOC/LFI,LLR,LFPB,LFTB,LPDSB,LFCSB,LFCTP,LFAE,LCAL
651.   COMMON/GEN1/NRUNS,NFILE,TEEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
652. -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
653.   COMMON/GSC1/NCRRD,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
654. -DDL(99),TNOW,IX(99),DTNOW,ISTOP
655.   COMMON ISET(1)
656.   EQUIVALENCE(ISET(1),RSET(1))
657. C*
658. C*   TAKE THE FIRST ENTRY OF THE CALENDER
659. C*
660.   J=LCAL+1

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

661.      LENT=ISET(J)
662.      L=LCAL
663.      CALL ULINK(L,LENT)
664.      C*
665.      C*      TAKE ITS EVENT CODE AND PROCESS IT
666.      C*
667.      J=LENT+2
668.      ICODE=ISET(J)
669.      IF(ICODE.NE.-1) GO TO 11
670.      CALL CLEARS
671.      RETURN
672.      C*
673.      C*      COPY THE ATTRIBUTES INTO ATRIE ARRAY
674.      C*
675.      11      IF(NATR.EQ.0) GO TO 16
676.      J=LENT+3
677.      LATR=ISET(J)
678.      DO 15 K=1,NATR
679.          J=LATR+K
680.          ATRIB(K)=RSET(J)
681.      15      CONTINUE
682.      16      CALL EVENT(ICODE)
683.      C*
684.      C*      FOR ALL THE XX DTPST SEE IF THEIR VALUE IS CHANGED
685.      C*      IF SO COLLECT STATISTICS
686.      C*
687.      IF(NTMX.EQ.0) RETURN
688.      DO 10 K=1,NTMX
689.          L=LTMX(K)
690.          J=L+7
691.          I=-ISET(J)
692.          J=L+8
693.          NU=ISET(J)
694.          LSTAT=ISET(L)
695.          J=LSTAT+2
696.          IF(XX(I).EQ.RSET(J)) RETURN
697.          X=XX(I)
698.          CALL DTPST(X,LSTAT,NU)
699.      10      CONTINUE
700.      RETURN
701.      END
702.      SUBROUTINE SSCND
703.      COMMON/GEN2/NEQD,NEOS,NEQT,DISAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
704.      -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
705.      -ISCD(25)
706.      DO 10 J=1,NSEV
707.          ISCD(J)=KKRSS(ICGV(J),ICDV(J),VAL(J),ICDIR(J),TTOL(J))
708.      10      CONTINUE
709.      RETURN
710.      END
711.      FUNCTION KKRSS(IKRSR,IKRSD,CARD,LDIR,TCL)
712.      COMMON/GSC1/MCRDR,NPRTR,SS(99),DD(99),ATRI(99),SSL(99),JJ,
713.      -DDL(99),TNOW,XX(99),DTNOW,ISTCP
714.      COMMON/GEN2/NEQD,NEOS,NEQT,DISAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
715.      -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

716.      -ISCD(25)
717.      DIMENSION IKRSG(1),IKRSD(1),CADD(1),LDIR(1),TOL(1)
718.      JKRS=IKRSG(1)
719.      JKRS=IKRSD(1)
720.      IF (JKRS) 10,260,20
721.      10 JKRS=-JKRS
722.      CRSG=DDL(JKRS)
723.      CRSG=DD(JKRS)
724.      GO TO 30
725.      20 CRSG=SSL(JKRS)
726.      CRSG=SS(JKRS)
727.      30 IF (JKRS) 40,50,60
728.      40 JKRS=-JKRS
729.      CRSD=DDL(JKRS)
730.      CRSD=DD(JKRS)
731.      GO TO 70
732.      50 CRSD=CADD(1)
733.      CRSD=CADD(1)
734.      GO TO 70
735.      60 CRSD=SSL(JKRS)
736.      CRSD=SS(JKRS)
737.      70 IF (CRSG-CRSD) 80,260,90
738.      80 IF (CRSG-CRSD) 260,100,100
739.      90 IF (CRSG-CRSD) 130,130,260
740.      100 IF (LDIR(1)) 260,110,110
741.      110 IF (CRSG-CRSD-TOL(1)) 120,120,200
742.      120 KKR=1
743.      GO TO 160
744.      130 IF (LDIR(1)) 140,140,260
745.      140 IF (CRSG-CRSD+TOL(1)) 210,150,150
746.      150 KKR=-1
747.      160 IF (ISEE) 270,170,270
748.      170 IF (IKRS(1)) 180,270,190
749.      180 ISEE=JKRS+1000
750.      GO TO 270
751.      190 ISEE=JKRS
752.      GO TO 270
753.      200 KKR=2
754.      GO TO 220
755.      210 KKR=-2
756.      220 IF (ISEE) 270,230,230
757.      230 IF (IKRS(1)) 240,270,250
758.      240 ISEE=-JKRS-1000
759.      GO TO 270
760.      250 ISEE=-JKRS
761.      GO TO 270
762.      260 KKR=0
763.      270 RETURN
764.      END
765.      SUBROUTINE SSAVE
766.      DIMENSION RSET(1)
767.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCB,LFCB,LFAE,ICAL
768.      COMMON/GEN1/NRNS,NFILE,TEEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
769.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
770.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,

```

```

771.      -DEL(99),TNOW, XX(99),DTNOW,ISTOP
772.      COMMON ISET(1)
773.      EQUIVALENCE(ISET(1),RSET(1))
774.      DC 10 N=1,NCTPST
775.          LCBL=LFCSB+(N-1)*9
776.          LSTAT=ISET(LCBL)
777.          ITYPE=ISET(LCBL+7)
778.          IF(ITYPE.LT.0) GO TO 15
779.          X=SS(ITYPE)
780.          GO TO 16
781.      15      X=DD(-ITYPE)
782.      16      NU=ISET(LCBL+8)
783.          CALL CTPST(X,LSTAT,NU)
784.      10      CONTINUE
785.          RETURN
786.          END
787.          SUBROUTINE DATAIN
788.      C*
789.      C*      THIS SUBROUTINE READS THE INPUT CARDS AND SETS
790.      C*      UP THE 'ENVIRONMENT' FOR THE MODEL. FOR THE TIMEBEING
791.      C*      THE CARDS HAVE FIXED FORMAT AND ORDER.
792.      C*
793.          DIMENSION KARD(13),RSET(1),NAME(6),INFO(4)
794.          DATA INFO /1HN,1HW,1H ,1HF/
795.          DATA KARD /4HPROJ,4HDISC,4HRANK,4HCCNT,4HSEVN,4HSTAT,4HTALY,
796.          -4HDTPS,4HCTPS,4HCOUN,4HSTRM,4HSINU,4HINTL/
797.          COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,ICAL
798.          COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
799.          -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
800.          COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
801.          -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
802.          -ISCD(25)
803.          COMMON/GEN3/ITITLE(15),IANAL(10),IDATE(5)
804.          COMMON/GEN4/ISEED(10),LSEED(10)
805.          COMMON/GEN8/NIR,VALH,VALL
806.          COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
807.          -DEL(99),TNOW, XX(99),DTNOW,ISTOP
808.          COMMON ISET(1)
809.          EQUIVALENCE(ISET(1),RSET(1))
810.      C*
811.      C*      READ THE PROJECT CARD
812.      C*
813.          READ(NCRDR,1001) LABEL,(ITITLE(I),I=1,15),(IANAL(I),I=1,10),
814.          -(IDATE(I),I=1,5)
815.          WRITE(NPRTR,2001) LABEL,(ITITLE(I),I=1,15),(IANAL(I),I=1,10),
816.          -(IDATE(I),I=1,5)
817.          IF(LABEL.NE.KARD(1)) GO TO 100
818.      C*
819.      C*      READ THE DISCRETE CARD
820.      C*
821.          READ(NCRDR,1002) LABEL,NFILE,NENT,NATR
822.          WRITE(NPRTR,2002) LABEL,NFILE,NENT,NATR
823.          IF(LABEL.NE.KARD(2)) GO TO 101
824.          LFFB=LFI
825.          IF(NFILE.EQ.0.AND.NENT.EQ.0) GO TO 19

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234
826.							
827.							
828.							
829.	C*						
830.	C*						
831.	C*						
832.							
833.							
834.							
835.	C*						
836.	C*						
837.	C*						
838.							
839.	10						
840.	C*						
841.	C*						
842.	C*						
843.	11						
844.	C*						
845.	C*						
846.	C*						
847.	19						
848.							
849.							
850.							
851.							
852.							
853.							
854.							
855.							
856.							
857.							
858.							
859.	C*						
860.	C*						
861.	C*						
862.							
863.							
864.							
865.							
866.							
867.	-						
868.							
869.	-						
870.							
871.							
872.							
873.							
874.							
875.							
876.	22						
877.							
878.							
879.	20						
880.	C*						

```

1. C* READ STAT CARD
2. C*
3. 29 READ(NCRDR,1006) LABEL,N,(NAME(J),J=1,6),NU
4. WRITE(NPRTR,2006) LABEL,N,(NAME(J),J=1,6),NU
5. IF(LABEL.NE.KARD(6)) GO TO 105
6. C*
7. C* READ TALLY CARDS
8. C*
9. IF(NTAL.LT.0.OR.NDTPST.LT.0.CR.NCTPST.LT.0.OR.NCCUN.LT.0) GO TO 124
10. IF(NTAL.EQ.0) GO TO 39
11. LFTB=LFI
12. DO 30 I=1,NTAL
13. READ(NCRDR,1007) LABEL,N,(NAME(J),J=1,6),NU
14. WRITE(NPRTR,2007) LABEL,N,(NAME(J),J=1,6),NU
15. IF(LABEL.NE.KARD(7)) GO TO 106
16. CALL SETTAL(NAME,NU)
17. 30 CCNTINUE
18. C*
19. C* READ DTPS CARDS
20. C*
21. 39 IF(NDTPST.EQ.0) GO TO 49
22. LFDSB=LFI
23. DO 40 I=1,NDTPST
24. READ(NCRDR,1008) LABEL,N,(NAME(J),J=1,6),ITYPE,NU
25. WRITE(NPRTR,2008) LABEL,N,(NAME(J),J=1,6),ITYPE,NU
26. IF(LABEL.NE.KARD(8)) GO TO 107
27. CALL SETDS (NAME,ITYPE,NU)
28. 40 CONTINUE
29. C*
30. C* READ CTPS CARDS
31. C*
32. 49 IF(NCTPST.EQ.0) GO TO 59
33. LFCSB=LFI
34. DO 50 I=1,NCTPST
35. READ(NCRDR,1008) LABEL,N,(NAME(J),J=1,6),ITYPE,NU
36. WRITE(NPRTR,2008) LABEL,N,(NAME(J),J=1,6),ITYPE,NU
37. IF(LABEL.NE.KARD(9)) GO TO 108
38. CALL SETCS (NAME,ITYPE,NU)
39. 50 CONTINUE
40. C*
41. C* READ COUN CARDS
42. C*
43. 59 IF(NCOUN.EQ.0) GO TO 69
44. LFCTB=LFI
45. DO 60 I=1,NCOUN
46. READ(NCRDR,1009) LABEL,N,(NAME(J),J=1,6),LIMIT
47. WRITE(NPRTR,2009) LABEL,N,(NAME(J),J=1,6),LIMIT
48. IF(LABEL.NE.KARD(10)) GO TO 109
49. CALL SETCT(NAME,LIMIT)
50. 60 CONTINUE
51. 69 IF(NENT.EQ.1) CALL SETENP
52. C*
53. C*
54. C* READ STRM CARDS
55. C*

```

```

6.      DC 70 I=1,10
7.          READ(NCRDR,1010) LABEL,IR,IS
8.          IF(LABEL.NE.KARD(11)) GO TO 71
9.          WRITE(NPRTR,2010) LABEL,IR,IS
10.         IF(IR.LT.1.OR.IR.GT.10) CALL ERROR(13,IR)
11.         LSEED(IR)=IS
12.         IF(IS.LT.0) IS=-IS
13.         ISEED(IR)=IS
14.     70    CONTINUE
15.         GC TO 80
16.     71    WRITE(NPRTR,2011) LABEL,IR
17.         IF(LABEL.NE.KARD(12)) GO TO 111
18.         NRUNS=IR
19.         GO TO 81
20.
21.     C*
22.     C*
23.     C*    READ SIMU CARD
24.     C*
25.     80    READ(NCRDR,1011) LABEL,NRUNS
26.         WRITE(NPRTR,2011) LABEL,NRUNS
27.         IF(LABEL.NE.KARD(12)) GO TO 111
28.
29.     C*
30.     C*    READ INTL CARD
31.     C*
32.     81    READ(NCRDR,1012) LABEL,TBEG,TFIN,IRAN,IECO,ISUM,ICLF,ICLV,ICLS,
33.         -TCLEAR
34.         WRITE(NPRTR,2012) LABEL,TBEG,TFIN,IRAN,IECO,ISUM,ICLF,ICLV,ICLS,
35.         -TCLEAR
36.         IF(LABEL.NE.KARD(13)) GO TO 110
37.         IF(IRAN.NE.1.AND.IRAN.NE.0) GO TO 121
38.         IF(IECO.NE.1.AND.IECO.NE.0) GO TO 119
39.         IF(ISUM.NE.1.AND.ISUM.NE.0) GO TO 120
40.         IF(TFIN.LE.TBEG) TFIN=VALH
41.         IF(ICLF.NE.1.AND.ICLF.NE.0) GO TO 113
42.         IF(ICLV.NE.1.AND.ICLV.NE.0) GO TO 118
43.         IF(ICLS.NE.1.AND.ICLS.NE.0) GO TO 114
44.         IF(ICLS.EQ.1) RETURN
45.         IF(TCLEAR.GT.TFIN.OR.TCLEAR.JT.TBEG) GO TO 115
46.         RETURN
47.
48.     1001   FORMAT(A4,5X,15A2,10A2,5A2)
49.     1002   FORMAT(A4,5X,I2,8X,I1,9X,I2)
50.     1003   FORMAT(A4,5X,I2,8X,I1,9X,I2)
51.     1004   FORMAT(A4,5X,I2,3X,I2,3X,I2,8X,4F10.5,A1)
52.     1005   FORMAT(A4,5X,I2,8X,I3,7X,I2,8X,I3,7X,2F10.5)
53.     1006   FORMAT(A4,5X,4(I2,8X),F10.5)
54.     1007   FORMAT(A4,5X,I2,8X,6A2,8X,I2)
55.     1008   FORMAT(A4,5X,I2,8X,6A2,8X,I3,7X,I2)
56.     1009   FORMAT(A4,5X,I2,8X,6A2,8X,I6)
57.     1010   FORMAT(A4,5X,I2,8X,I11)
58.     1011   FORMAT(A4,5X,I2)
59.     1012   FORMAT(A4,5X,2F10.3,5X,6(I1,4X),5X,F10.3)
60.     2001   FORMAT(10X,A4,5X,15A2,10A2,5A2)
61.     2002   FORMAT(10X,A4,5X,I2,8X,I1,9X,I2)
62.     2003   FORMAT(10X,A4,5X,I2,8X,I1,9X,I2)
63.     2004   FORMAT(10X,A4,5X,I2,3X,I2,3X,I2,8X,4F10.5,A1)

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
01.	2005	FORMAT(10X,A4,5X,I2,8X,I3,7X,I2,8X,I3,7X,2F10.5)					
02.	2006	FORMAT(10X,A4,5X,4(I2,8X),F10.5)					
03.	2007	FORMAT(10X,A4,5X,I2,8X,6A2,8X,I2)					
04.	2008	FORMAT(10X,A4,5X,I2,8X,6A2,8X,I3,7X,I2)					
05.	2009	FORMAT(10X,A4,5X,I2,8X,6A2,8X,I6)					
06.	2010	FORMAT(10X,A4,5X,I2,8X,I11)					
07.	2011	FORMAT(10X,A4,5X,I2)					
08.	2012	FORMAT(10X,A4,5X,2F10.3,5X,6(I1,4X),5X,F10.3)					
09.	100	WRITE(NPRTR,200)					
10.	200	FORMAT(10X,50H*****ERROR: THE PORJ CARD IS MISSING					)
11.		GO TO 1000					
12.	101	WRITE(NPRTR,201)					
13.	201	FORMAT(10X,50H*****ERROR: THE DISC CARD IS MISSING					)
14.		GO TO 1000					
15.	102	WRITE(NPRTR,202)					
16.	202	FORMAT(10X,50H*****ERROR: NCT ENOUGH RANK CARDS					)
17.		GO TO 1000					
18.	103	WRITE(NPRTR,203)					
19.	203	FORMAT(10X,50H*****ERROR: THE CONT CARD IS MISSING					)
20.		GO TO 1000					
21.	104	WRITE(NPRTR,204)					
22.	204	FORMAT(10X,50H*****ERROR: NCT ENOUGH SEVN CARD OR NSEV.LT.0					)
23.		GO TO 1000					
24.	188	WRITE(NPRTR,288)					
25.	288	FORMAT(10X,34H*****ERROR: NEQD.LT.0.OR.NEQS.LT.0)					
26.		GO TO 1000					
27.	189	WRITE(NPRTR,289)					
28.	289	FORMAT(10X,38H*****ERROR: THIS IS NEITHER CONTINUOUS,					
29.		-20H NOR DISCRETE MODEL)					
30.		GO TO 1000					
31.	190	WRITE(NPRTR,290)					
32.	290	FORMAT(10X,50H*****ERROR: IMPROPER VALUE FOR THE INDEX OF					,
33.		-22X,20HCROSSING VARIABLE)					
34.		GO TO 1000					
35.	191	WRITE(NPRTR,291)					
36.	291	FORMAT(10X,50H*****ERROR: IMPROPER DIRECTION					)
37.		GO TO 1000					
38.	192	WRITE(NPRTR,292)					
39.	292	FORMAT(10X,50H*****ERROR: IMPROPER CROSSED VARIABLE					)
40.		GO TO 1000					
41.	105	WRITE(NPRTR,205)					
42.	205	FORMAT(10X,50H*****ERROR: STAT CARD IS MISSING					)
43.		GO TO 1000					
44.	106	WRITE(NPRTR,206)					
45.	206	FORMAT(10X,50H*****ERROR: NCT ENOUGH TAYL CARDS					)
46.		GO TO 1000					
47.	107	WRITE(NPRTR,207)					
48.	207	FORMAT(10X,50H*****ERROR: NOT ENOUGH DTPS CARDS					)
49.		GO TO 1000					
50.	108	WRITE(NPRTR,208)					
51.	208	FORMAT(10X,50H*****ERROR: NCT ENOUGH CTPS CARDS					)
52.		GO TO 1000					
53.	109	WRITE(NPRTR,209)					
54.	209	FORMAT(10X,50H*****ERROR: NOT ENOUGH COUN CARDS					)
55.		GO TO 1000					

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	12345
46.	110	WRITE(NPRTR,210)					
47.	210	FORMAT(10X,50H*****ERROR: INTL CARD IS MISSING					)
48.		GO TO 1000					
49.	111	WRITE(NPRTR,211)					
50.	211	FORMAT(10X,50H*****ERROR: SIMU CARD IS MISSING					)
51.		GO TO 1000					
52.	113	WRITE(NPRTR,213)					
53.	213	FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR CLEARING THE					,
54.		-22X,5HFILES)					
55.		GO TO 1000					
56.	114	WRITE(NPRTR,214)					
57.	214	FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR CLEARING THE					,
58.		-22X,20HSTATISTICAL ARRAYS					)
59.		GO TO 1000					
60.	115	WRITE(NPRTR,215)					
61.	215	FORMAT(10X,50H*****ERROR: TCLEAR.GT.TFIN .OR.TCLEAR.LT.TBEG					)
62.		GO TO 1000					
63.	117	WRITE(NPRTR,217)					
64.	217	FORMAT(10X,50H*****ERROR: DISCRETE EVENT BUT NO ENTITY					)
65.		GO TO 1000					
66.	118	WRITE(NPRTR,218)					
67.	218	FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR CLEARING VARIABLES					)
68.		GO TO 1000					
69.	119	WRITE(NPRTR,219)					
70.	219	FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR DELETING ECHO REPORT					)
71.		GO TO 1000					
72.	120	WRITE(NPRTR,220)					
73.	220	FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR DELETING SUMMARY REPO					
74.		- ,2HRT)					
75.		GO TO 1000					
76.	121	WRITE(NPRTR,221)					
77.	221	FORMAT(10X,49H*****ERROR: ILLEGAL CODE FOR INITIALIZING RANDOM					
78.		- ,24HNO. STREAMS BETWEEN RUNS)					
79.		GO TO 1000					
80.	122	WRITE(NPRTR,222)					
81.	222	FORMAT(10X,44H*****ERROR: ILLEGAL CODE FOR ERROR INDICATOR)					
82.		GO TO 1000					
83.	123	WRITE(NPRTR,223)					
84.	223	FORMAT(10X,50H*****ERROR: DTMIN.LT.0.OR.DTMAX.LT.0.OR.AERR.LT.0.,					
85.		-12HOR.RERR.LT.0)					
86.		GO TO 1000					
87.	124	WRITE(NPRTR,224)					
88.	224	FORMAT(10X,50H,*****ERROR:NTAL.LT.0.OR.NDTPST.LT.0.OR.NCTPST.LT0,					
89.		-14H.OR.NCOUN.LT.0)					
90.	1000	I=SQRT(-1.)					
91.		STOP					
92.		END					
93.		SUBROUTINE SETCAL					
94.		DIMENSION RSET(1)					
95.		COMMON/LOC/LFI,LLR,LFFB,LFTB,LPDSB,LPCSB,LFCSTB,LFAE,LCAL					
96.		COMMON/GEN8/NIR,VALH,VALI					
97.		COMMON ISET(1)					
98.		EQUIVALENCE (ISET(1),RSET(1))					
99.		II=LFI+6					
0.		I=(II+NIR-1)/NIR					

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234
101.							
102.							
103.							
104.	C*						
105.	C*						
106.	C*						
107.							
108.							
109.							
110.							
111.							
112.							
113.							
114.							
115.							
116.							
117.							
118.							
119.							
120.							
121.							
122.							
123.							
124.							
125.							
126.							
127.							
128.							
129.							
130.							
131.							
132.	1						
133.							
134.							
135.							
136.							
137.							
138.							
139.							
140.							
141.							
142.							
143.							
144.							
145.							
146.							
147.							
148.							
149.	10						
150.							
151.							
152.							
153.							
154.							
155.							

AD-A101 063

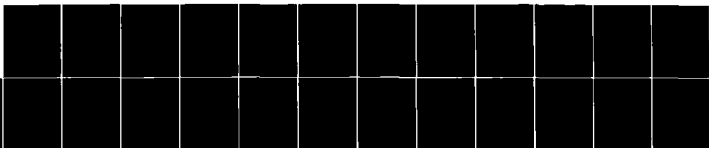
PENNSYLVANIA STATE UNIV UNIVERSITY PARK DEPT OF INDU--ETC F/G 9/2
A GENERALIZED COMPUTER SIMULATION LANGUAGE FOR NAVAL SYSTEMS MO--FTC(1)
JUN 81 W E BILES, A NOZARI
N00014-79-C-0757

NL

UNCLASSIFIED

2nd 2

AD
ADONIS



END

DATE

FILED

7 81

DTIC

1234567890123456789012345678901234567890123456789012345678901234

```

156.      I=LFI+4
157.      J=LLR-(NATR+1)
158.      II=(I+NIR-1)/NIR
159.      IF(II.LE.J) GO TO 10
160.      I=I-3
161.      ISET(I)=0
162.      NENT=KK
163.      RETURN
164.      END
165.      SUBROUTINE SETTAL(NAME,IU)
166.      DIMENSION NAME(6)
167.      DIMENSION RSET(1)
168.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
169.      COMMON/GEN8/NIR,VALH,VALL
170.      COMMON ISET(1)
171.      EQUIVALENCE(ISET(1),RSET(1))
172.      I=LFI+8
173.      J=LLR-5
174.      II=(I+NIR-1)/NIR
175.      IF(II.GT.J) CALL ERROR (1,1)
176.      ISET(LFI)=J
177.      II=LFI+1
178.      KK=LFI+6
179.      DO 10 L=II, KK
180.          LL=L-II+1
181.          ISET(L)=NAME(LL)
182. 10      CONTINUE
183.      II=LFI+7
184.      ISET(II)=IU
185.      II=J+3
186.      RSET(II)=VALH
187.      II=J+4
188.      RSET(II)=VALL
189.      LFI=I
190.      LLR=J
191.      RETURN
192.      END
193.      SUBROUTINE SETCS (NAME, ITYPE, NU)
194.      DIMENSION NAME(6)
195.      DIMENSION RSET(1)
196.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
197.      COMMON/GEN1/NRUNS,NFILE,TREG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
198.      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,LECO,ISUM
199.      COMMON/GEN8/NIR,VALH,VALL
200.      COMMON ISET(1)
201.      EQUIVALENCE(ISET(1),RSET(1))
202.      I=LFI+9
203.      J=LLR-7
204.      II=(I+NIR-1)/NIR
205.      IF(II.GT.J) CALL ERROR (1,1)
206.      ISET(LFI)=J
207.      II=LFI+1
208.      KK=LFI+6
209.      DO 10 L=II, KK
210.          LL=L-II+1

```

1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234

```

211.      ISET(L)=NAME(LL)
212. 10    CONTINUE
213.      R=ITYPE
214.      IF(ITYPE.EQ.0.OR.ABS(R).GT.99.) CALL ERROR (19,ITYPE)
215.      II=LFI+7
216.      ISET(II)=ITYPE
217.      II=LFI+8
218.      ISET(II)=NU
219.      II=J+3
220.      RSET(II)=VALH
221.      II=J+4
222.      RSET(II)=VALL
223.      LFI=I
224.      LIR=J
225.      RETURN
226.      END
227.      SUBROUTINE SETDS (NAME,ITYPE,NU)
228.      DIMENSION NAME(6)
229.      DIMENSION RSET(1)
230.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSP,LFCTB,LFAE,LCAI
231.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLP,ICLV,ICLS,TCLEAR,IRAN,
232.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
233.      COMMON/GEN8/NIR,VALH,VALL
234.      COMMON ISET(1)
235.      EQUIVALENCE(ISET(1),RSET(1))
236.      I=LFI+9
237.      J=LLR-7
238.      II=(I+NIR-1)/NIR
239.      IF(II.GT.J) CALL ERROR (1,1)
240.      ISET(LFI)=J
241.      II=LFI+1
242.      KK=LFI+6
243.      DO 10 L=II,KK
244.          LL=L-II+1
245.          ISET(L)=NAME(LL)
246. 10    CONTINUE
247.      IF(ITYPE.GT.0.AND.ITYPE.LE.NFILE) GO TO 11
248.      IF(ITYPE.EQ.0.OR.ITYPE.LT.-99.CR.ITYPE.GT.NFILE)
249.      -CALL ERROR (19,ITYPE)
250.      NTMX=NTMX+1
251.      LTMX(NTMX)=LFI
252.      GO TO 12
253. 11    LPBL=LFFB+(ITYPE-1)*6
254.      II=LPBL+5
255.      ISET(II)=LFI
256. 12    II=LFI+7
257.      ISET(II)=ITYPE
258.      II=LFI+8
259.      ISET(II)=NU
260.      II=J+3
261.      RSET(II)=VALH
262.      II=J+4
263.      RSET(II)=VALL
264.      LFI=I
265.      LLR=J

```

1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234

```

266.      RETURN
267.      END
268.      SUBROUTINE ERROF (ICCODE,I)
269.      COMMON/GSC1/NCRDR,NPRTR,SS(99),ED(99),ATRI(99),SSL(99),JJ,
270.      -DDL(99),TNOW,XX(99),DTNOW,ISTOP
271.      IS=0
272.      IF(ICODE.LT.1000) GO TO 100
273.      ICCODE=1000-ICCODE
274.      IS=1
275. 100      GC TO (101,102,103,104,105,106,107,108,109,110,111,112,113,
276.      -114,115,116,117,118), ICCDE
277. 101      WRITE(NPRTR,201)
278. 201      FORMAT(10X,37H*****ERROR: THERE IS NOT ENOUGH SPACE)
279.      IF(IS.EQ.1) GO TO 998
280.      GC TO 999
281. 102      WRITE(NPRTR,202)
282. 202      FORMAT(10X,44H*****ERROR: RANKING ATTRIBUTE IS OUT OF RANGE)
283.      GC TO 999
284. 103      WRITE(NPRTR,203)
285. 203      FORMAT(10X,33H*****ERROR: RANKING CODE IS WRONG)
286.      GC TO 999
287. 104      WRITE(NPRTR,204) I
288. 204      FORMAT(10X,44H*****ERROR: THE ENITY THAT IS TO BE REMOVED,
289.      -,27H COPIED, OR LOCATED IN FILE,I4,15H DOES NOT EXIST)
290.      GC TO 998
291. 105      WRITE(NPRTR,205) I
292. 205      FCRMAT(10X,29H*****ERROR: THERE IS NO CTPST,I4)
293.      IF(IS.EQ.1) GO TO 998
294.      GC TO 999
295. 106      WRITE(NPRTR,206) I
296. 206      FORMAT(10X,29H*****ERROR: THERE IS NO DTPST,I4)
297.      IF(IS.EQ.1) GO TO 998
298.      GC TO 999
299. 107      WRITE(NPRTR,207) I
300. 207      FORMAT(10X,29H*****ERROR: THERE IS NO TALLY,I4)
301.      IF(IS.EQ.1) GO TO 998
302.      GC TO 999
303. 108      WRITE(NPRTR,208)
304. 208      FCRMAT(10X,36H*****ERROR: NO CONTINUOUS NO DICRETE)
305.      GC TO 999
306. 109      WRITE(NPRTR,209)
307. 209      FORMAT(10X,36H*****ERROR: TNEXT<TNOW
308.      GC TO 998
309. 110      WRITE(NPRTR,210) I
310. 210      FORMAT(10X,29H*****ERROR: THERE IS NO FILE ,I3)
311.      IF(IS.EQ.1) GO TO 998
312.      GC TO 999
313. 111      WRITE(NPRTR,211) I
314. 211      FCRMAT(10X,33H,*****ERROR: THERE IS NO COUNTER ,I4)
315.      IF(IS.EQ.1) GO TO 998
316.      GC TO 999
317. 112      WRITE(NPRTR,212) I
318. 212      FORMAT(10X,33H,*****ERROR: THERE IS NO ATTRIBUTE,I4)
319.      IF(IS.EQ.1) GO TO 998
320.      GC TO 999

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

1321. 113 WRITE(NPRTR,213) I
1322. 213 FORMAT(10X,38H*****ERROR: STRFAM NO. IS OUT OF RANGE,I4)
1323. IF (IS.EQ.1) GO TO 998
1324. GC TO 999
1325. 114 WRITE(NPRTR,214)
1326. 214 FORMAT(10X,25H*****ERROR: IRAND IS ZERO)
1327. GC TO 998
1328. 115 WRITE(NPRTR,215)
1329. 215 FORMAT(10X,45H*****ERROR: INCORRECT PARAMETER SPECIFICATION)
1330. GC TO 998
1331. 116 WRITE(NPRTR,216)
1332. 216 FORMAT(10X,45H*****ERROR: INCORRECT CUMULATIVE PROBABILITY ,
1333. -13HSPECIFICATION)
1334. GC TO 998
1335. 117 WRITE(NPRTR,217)
1336. 217 FORMAT(10X,45H*****ERROR: FATAL INTEGRATION TOLERANCE ERROR)
1337. GC TO 998
1338. 118 WRITE(NPRTR,218)
1339. 218 FORMAT(10X,45H*****ERROR: FATAL INTEGRATION ACCURACY ERROR)
1340. GC TO 998
1341. 119 WRITE(NPRTR,219) ITYPE
1342. 219 FORMAT(10X,11H*****ERROR:,I4,22H IS OUT OF RANGE INDEX )
1343. GO TO 999
1344. 998 CALL SUMRY(0)
1345. 999 I=SQRT(-1.)
1346. STOP
1347. END
1348. SUBROUTINE SETCT(NAME,LIMIT)
1349. DIMENSION NAME(6)
1350. DIMENSION RSET(1)
1351. COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1352. COMMON/GEN8/NIR,VALH,VALL
1353. COMMON ISET(1)
1354. EQUIVALENCE(ISET(1),RSET(1))
1355. I=LFI+8
1356. II=(I+NIR-1)/NIR
1357. IF (II.GT.LLR) CALL ERROR (1,1)
1358. II=LFI+1
1359. KK=LFI+6
1360. DO 10 L=II,KK
1361. LL=L-II+1
1362. ISET(L)=NAME(LL)
1363. 10 CCNTINUE
1364. II=LFI+7
1365. ISET(II)=LIMIT
1366. LFI=I
1367. RETURN
1368. END
1369. SUBROUTINE FILEM(IFILE,A)
1370. DIMENSION A(99)
1371. DIMENSION RSET(1)
1372. COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1373. COMMON/GEN1/NRUNS,NFILE,IBEG,TFIN,ICLP,ICLV,ICLS,TCLEAR,IRAN,
1374. -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
1375. COMMON/GEN5/IEVENT,TIME

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	123
1376.							
1377.							
1378.	C*						
1379.	C*						
1380.	C*						
1381.							
1382.							
1383.							
1384.							
1385.							
1386.	C*						
1387.	C*						
1388.	C*						
1389.							
1390.							
1391.	C*						
1392.	C*						
1393.	C*						
1394.							
1395.							
1396.							
1397.							
1398.	10						
1399.	C*						
1400.	C*						
1401.	C*						
1402.	C*						
1403.	C*						
1404.	C*						
1405.	12						
1406.							
1407.							
1408.							
1409.							
1410.	C*						
1411.	C*						
1412.	C*						
1413.	1						
1414.							
1415.	C*						
1416.	C*						
1417.	C*						
1418.	2						
1419.							
1420.							
1421.	C*						
1422.	C*						
1423.	C*						
1424.	3						
1425.							
1426.							
1427.							
1428.							
1429.							
1430.	5						

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
1431.	C*						
1432.	C*	THIS IS EVENT CAIANDER					
1433.	C*						
1434.		RSET(LATR)=TIME					
1435.		J=LENT+2					
1436.		ISET(J)=IEVENT					
1437.	6	CONTINUE					
1438.	C*						
1439.	C*	FIND THE POSITION					
1440.	C*						
1441.		J=LFBL+1					
1442.		LLFE=ISET(J)					
1443.		IF(LLFE.NE.0) GO TO 7					
1444.		L=0					
1445.		GO TO 40					
1446.	7	CONTINUE					
1447.		L=LLFE					
1448.		IF(IRANK.EQ.4) GO TO 4					
1449.	C*						
1450.	C*	THIS IS HVF					
1451.	C*						
1452.	11	I=L+3					
1453.		J=ISET(I)					
1454.		IF(RSET(J).LT.RSET(LATR)) GO TO 40					
1455.		I=L+1					
1456.		L=ISET(I)					
1457.		IF(L.EQ.0) GO TO 40					
1458.		GO TO 11					
1459.	C*						
1460.	C*	THIS IS LVF					
1461.	C*						
1462.	4	I=L+3					
1463.		J=ISET(I)					
1464.		IF(RSET(J).GT.RSET(LATR)) GO TO 40					
1465.		I=L+1					
1466.		L=ISET(I)					
1467.		IF(L.EQ.0) GO TO 40					
1468.		GO TO 4					
1469.	40	CALL LINK(LFBL,L,LENT)					
1470.		RETURN					
1471.		END					
1472.		SUBROUTINE LINK(LFBL,L,LENT)					
1473.		DIMENSION RSET(1)					
1474.		COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFACTB,LFAE,ICAL					
1475.		COMMON ISET(1)					
1476.		EQUIVALENCE (ISET(1),RSET(1))					
1477.	C*						
1478.	C*	THIS SUBROUTINE LINKS THE ENTITY AT LENT TO THE FILE AT					
1479.	C*	LFBL, AT ADDRESS L.					
1480.	C*	IF L=0 THE ENTITY IS LINKED TO THE BOTTOM OF THE FILE					
1481.	C*	ALSO THEN IT MUST BE CHECKED IF THER IS ANY ENTITY IN					
1482.	C*	ALREADY IN THE FILE.					
1483.	C*						
1484.		IF(L.NE.0) GO TO 1					
1485.		J=LFBL+1					

1 2 3 4 5 6 7
12345678901234567890123456789012345678901234567890123

```

1486.      IF(ISET(J).GT.0) GO TO 2
1487.      C*
1488.      C*      THERE IS NO ENTITY IN THE FILE
1489.      C*
1490.      ISET(J)=LENT
1491.      J=LPBL+2
1492.      ISET(J)=LENT
1493.      ISET(LENT)=0
1494.      J=LENT+1
1495.      ISET(J)=0
1496.      GC TO 4
1497.      2      CONTINUE
1498.      C*
1499.      C*      LINK THE ENTITY TO THE BOTTOM OF THE FILE
1500.      C*
1501.      J=LPBL+2
1502.      LLLE=ISET(J)
1503.      ISET(J)=LENT
1504.      ISET(LENT)=LLE
1505.      J=LENT+1
1506.      ISET(J)=0
1507.      J=LLLE+1
1508.      ISET(J)=LENT
1509.      GO TO 4
1510.      1      CONTINUE
1511.      C*
1512.      C*      ADD IT TO THE LIST BUT SEE IF IT IS TOP OF THE LIST
1513.      C*
1514.      J=LPBL+1
1515.      LLFE=ISET(J)
1516.      IF(L.NE.LLFE) GO TO 3
1517.      C*
1518.      C*      THIS IS TOP OF THE LIST
1519.      C*
1520.      ISET(J)=LENT
1521.      ISET(LENT)=0
1522.      J=LENT+1
1523.      ISET(J)=LLFE
1524.      ISET(LLFE)=LENT
1525.      GC TO 4
1526.      3      CONTINUE
1527.      C*
1528.      C*      LINK IT TO THE FILE AND DCN'T WORRY ABOUT THE LFE & LLE
1529.      C*
1530.      LF=ISET(L)
1531.      ISET(LENT)=LP
1532.      J=LENT+1
1533.      ISET(J)=L
1534.      J=LP+1
1535.      ISET(J)=LENT
1536.      ISET(L)=LENT
1537.      4      CONTINUE
1538.      C*
1539.      C*      COLLECT STATISTICS ON THE FILE, IF NECESSARY
1540.      C*

```

```

      ISET (LFBL) = ISET (LFBL) + 1
      J = LFBL + 5
      LDTB = ISET (J)
      IF (LDTB.EQ.0) RETURN
      LSTAT = ISET (LDTB)
      X = ISET (LFBL)
      J = LDTB + 8
      NU = ISET (J)
      CALL DTPST (X, LSTAT, NU)
      RETURN
      END
      SUBROUTINE RMOVE ( I, IFILE, A)

C*
C*      THIS SUBROUTINE REMOVES THE ITH ENTRY OF THE FILE IFILE
C*      AND LOADS ITS ATRIBUTES INTO ARRAY A.
C*

      DIMENSION A (1)
      DIMENSION RSET (1)
      COMMON/LOC/LFI, LLR, LFFB, LFTB, LFDSB, LFCSB, LFCTB, LFAE, LCAL
      COMMON/GEN6/LFBL, LENT
      COMMON ISET (1)
      EQUIVALENCE (ISET (1), RSET (1))
      CALL COPY (I, IFILE, A)
      CALL ULINK (LFBL, LENT)
      RETURN
      END
      SUBROUTINE COPY (I, IFILE, A)

C*
C*      THIS SUBROUTINE COPIES THE ATRIBUTES OF THE ITH ENTRY
C*      OF FILE IFILE INTO A.
C*

      DIMENSION A (1)
      DIMENSION RSET (1)
      COMMON/GEN1/NRUNS, NFILE, TBEG, TFIN, ICLF, ICLV, ICLS, TCLEAR, IRAN,
-NTMX, LTMX (99), NTAL, NDTPST, NCTPST, NCOUN, NATR, NENT, IECO, ISUM
      COMMON ISET (1)
      EQUIVALENCE (ISET (1), RSET (1))
      IF (NATR.LE.0) RETURN
      CALL LOCATE (I, IFILE, LENT)
      J = LENT + 3
      LATR = ISET (J)
      DO 6 J = 1, NATR
          K = LATR + J
          A (J) = RSET (K)
6      CONTINUE
      RETURN
      END
      SUBROUTINE LOCATE (I, IFILE, LENT)

C*
C*      THIS SUBROUTINE FINDS THE LOCATION OF THE ITH ENTRY
C*      IN FILE IFILE. THE LOCATION IS RETURNED IN LENT.
C*

      DIMENSION A (1)
      DIMENSION RSET (1)
      COMMON/LOC/LFI, LLR, LFFB, LFTB, LFDSB, LFCSB, LFCTB, LFAE, LCAL

```

1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012

```

1596.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1597.      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
1598.      COMMON/GEN6/LFBL,L
1599.      COMMON ISET(1)
1600.      EQUIVALENCE (ISET(1),RSET(1))
1601.      IF(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERRCLR(1010,IFILE)
1602.      LFBL=LFPB+(IFILE-1)*6
1603.      IF(I.GT.ISET(LFBL).OR.I.LT.1) CALL ERROR(4,IFILE)
1604.      J=ISET(LFBL)/2
1605.      IF(I.GT.J) GO TO 2
1606.      C*
1607.      C*      START FROM THE TOP
1608.      C*
1609.      J=LFBL+1
1610.      LENT=ISET(J)
1611.      K=1
1612.      4      CONTINUE
1613.          IF(K.EQ.I) GO TO 6
1614.          J=LENT+1
1615.          LENT=ISET(J)
1616.          K=K+1
1617.          GO TO 4
1618.      2      CONTINUE
1619.      C*
1620.      C*      START FROM THE BOTTOM
1621.      C*
1622.      J=LFBL+2
1623.      LENT=ISET(J)
1624.      K=ISET(LFBL)
1625.      5      CCNTINUE
1626.          IF(K.EQ.I) GO TO 6
1627.          LENT=ISET(LENT)
1628.          K=K-1
1629.          GO TO 5
1630.      6      L=LENT
1631.      RETURN
1632.      END
1633.      SUBROUTINE ULINK (LFBL,LENT)
1634.      C*
1635.      C*      THIS SUBROUTINE UNLINKS ENTRY AT LENT OF FILE AT LFBL
1636.      C*      AND LINKS IT TO THE ENTRY PCCL.
1637.      C*
1638.      DIMENSION RSET(1)
1639.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAI
1640.      COMMON ISET(1)
1641.      EQUIVALENCE (ISET(1),RSET(1))
1642.      LP=ISET(LENT)
1643.      J=LENT+1
1644.      LS=ISET(J)
1645.      C*
1646.      C*      LINK IT TO THE POOL
1647.      C*
1648.      ISET(J)=LFAE
1649.      LFAE=LENT
1650.      ISET(LENT)=0

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	123
1651.							
1652.							
1653.	C*						
1654.	C*						
1655.	C*						
1656.							
1657.							
1658.							
1659.							
1660.	10						
1661.							
1662.	C*						
1663.	C*						
1664.	C*						
1665.							
1666.							
1667.							
1668.							
1669.	25						
1670.	C*						
1671.	C*						
1672.	C*						
1673.							
1674.							
1675.							
1676.							
1677.							
1678.	20						
1679.	C*						
1680.	C*						
1681.	C*						
1682.							
1683.							
1684.							
1685.							
1686.	30						
1687.	C*						
1688.	C*						
1689.	C*						
1690.							
1691.							
1692.							
1693.							
1694.							
1695.							
1696.							
1697.							
1698.							
1699.							
1700.							
1701.							
1702.	C*						
1703.	C*						
1704.	C*						
1705.	C*						

	1	2	3	4	5	6	7
1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

1706. C*      MUST BE DUMPED.
1707. C*
1708.      DIMENSION RSET(1)
1709.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
1710.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
1711.      -DDL(99),TNOW,XX(99),DTNOW,ISTOP
1712.      COMMON ISET(1)
1713.      EQUIVALENCE(ISET(1),RSET(1))
1714.      LP1=L+1
1715.      LP2=L+2
1716.      LP3=L+3
1717.      LP4=L+4
1718.      LP5=L+5
1719.      XL=RSET(LP2)
1720.      RSET(LP2)=X
1721.      IF(X.LT.RSET(LP3)) RSET(LP3)=X
1722.      IF(X.GT.RSET(LP4)) RSET(LP4)=X
1723.      DT=TNOW-RSET(LP5)
1724.      IF(DT.LE.0) RETURN
1725.      XDT=XL*DT
1726.      RSET(L)=RSET(L)+XDT
1727.      RSET(LP1)=RSET(LP1)+XL*XDT
1728.      RSET(LP5)=TNOW
1729.      IF(NU.LE.0) RETURN
1730.      WRITE(NU) TNOW,X
1731.      RETURN
1732.      END
1733.      SUBROUTINE CTPST(X,L,NU)
1734. C*
1735. C*      UPDATES THE INFORMATION ON CONTINUOUS TIME PERSISTANT STATISTICS
1736. C*      WITH X. L IS THE LOCATION OF THE CORRESPONDING STATISTICS'
1737. C*      BLOCK. NU IS THE UNIT NO ON WHICH THE STATISTIC
1738. C*      MUST BE DUMPED.
1739. C*
1740.      DIMENSION RSET(1)
1741.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
1742.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
1743.      -DDL(99),TNOW,XX(99),DTNOW,ISTOP
1744.      COMMON ISET(1)
1745.      EQUIVALENCE(ISET(1),RSET(1))
1746.      LP1=L+1
1747.      LP2=L+2
1748.      LP3=L+3
1749.      LP4=L+4
1750.      LP5=L+5
1751.      XL=RSET(LP2)
1752.      RSET(LP2)=X
1753.      IF(X.LT.RSET(LP3)) RSET(LP3)=X
1754.      IF(X.GT.RSET(LP4)) RSET(LP4)=X
1755.      DT=TNOW-RSET(LP5)
1756.      IF(DT.LE.0) RETURN
1757.      XDT=(XL+X)*DT/2.
1758.      X2DT=(XL*XL+X*X)*DT/2.
1759.      RSET(L)=RSET(L)+XDT
1760.      RSET(LP1)=RSET(LP1)+X2DT

```

	1	2	3	4	5	6	7
123456789012345678901234567890123456789012345678901234567890123							

```

1761.      RSET(LP5) =TNOW
1762.      IF (NU.LE.0) RETURN
1763.      WRITE (NU) TNOW,X
1764.      RETURN
1765.      END
1766.      SUBROUTINE SCHD (IEVENT,TIME,A)
1767.      C*
1768.      C* SCHEDULES EVENT IEVENT AT TIME TIME ON CALANDER
1769.      C*
1770.      DIMENSION A(99)
1771.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1772.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1773.      COMMON/GEN5/II,TT
1774.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATFIB(99),SSL(99),JJ,
1775.      -IEL(99),TNOW,XX(99),DTNOW,ISTCP
1776.      II=IEVENT
1777.      TT=TIME
1778.      N=NFILE+1
1779.      CALL FILEM(N,A)
1780.      RETURN
1781.      END
1782.      SUBROUTINE TALLY (N,VAR)
1783.      C*
1784.      C* COLLECTS STATISTICS ON TALLY # N USING VAR
1785.      C*
1786.      DIMENSION RSET(1)
1787.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1788.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
1789.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSR,LFCTB,LFAE,LCAL
1790.      COMMON ISET(1)
1791.      EQUIVALENCE (ISET(1),RSET(1))
1792.      IF (N.GT.NTAL.OR.N.LT.1) CALL ERROR (7,N)
1793.      LTBL=LFTB+(N-1)*8
1794.      L=ISET(LTBL)
1795.      RSET(L)=RSET(L)+VAR
1796.      J=L+1
1797.      RSET(J)=RSET(J)+VAR*VAR
1798.      J=L+2
1799.      RSET(J)=RSET(J)+1.
1800.      J=L+3
1801.      IF (VAR.LT.RSET(J)) RSET(J)=VAR
1802.      J=L+4
1803.      IF (VAR.GT.RSET(J)) RSET(J)=VAR
1804.      J=LTBL+7
1805.      IF (ISET(J).EQ.0) RETURN
1806.      NU=ISET(J)
1807.      WRITE (NU) VAR
1808.      RETURN
1809.      END
1810.      SUBROUTINE COUNT(N,INC)
1811.      DIMENSION RSET(1)
1812.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSR,LFCTB,LFAE,LCAL
1813.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1814.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1815.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATFIB(99),SSL(99),JJ,

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
1816.							
1817.							
1818.							
1819.							
1820.							
1821.							
1822.							
1823.							
1824.							
1825.							
1826.							
1827.							
1828.							
1829.							
1830.							
1831.							
1832.							
1833.	10						
1834.	100						
1835.							
1836.							
1837.	20						
1838.	200						
1839.							
1840.							
1841.	30						
1842.							
1843.							
844.							
845.	C*						
1846.	C*						
1847.	C*						
848.							
1849.							
1850.							
851.							
852.							
1853.							
854.							
855.							
1856.							
1857.							
858.							
859.							
1860.							
861.							
862.							
863.							
864.							
865.							
866.							
867.							
868.							
869.							
870.							

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123

```

1871.      J=LSTAT+6
1872.      RSET(J)=TNOW
1873.      RETURN
1874.      END
1875.      SUBROUTINE TRUNC(N)
1876. C*
1877. C*      TRUNCATES CONTINUOUS TIME PERSISTANT STATISTICS
1878. C*
1879.      DIMENSION RSET(1)
1880.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1881.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1882.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1883.      COMMON/GEN8/NIR,VALH,VALI
1884.      COMMON/GSC1/NCRDR,NPRT,SS(99),DD(99),ATPFB(99),SSL(99),JJ,
1885.      -DDL(99),TNOW,XX(99),DTNOW,ISTOP
1886.      COMMON ISET(1)
1887.      EQUIVALENCE (ISET(1),RSET(1))
1888.      IF(N.GT.NCTPST.OR.N.LT.1) CALL ERROR(1005,N)
1889.      LCBL=LFCSE+(N-1)*9
1890.      LSTAT=ISET(LCBL)
1891.      RSET(LSTAT)=0
1892.      J=LSTAT+1
1893.      RSET(J)=0
1894.      J=LSTAT+2
1895.      RSET(J)=0
1896.      J=LSTAT+3
1897.      RSET(J)=VALH
1898.      J=LSTAT+4
1899.      RSET(J)=VALI
1900.      J=LSTAT+5
1901.      RSET(J)=TNOW
1902.      J=LSTAT+6
1903.      RSET(J)=TNOW
1904.      RETURN
1905.      END
1906.      SUBROUTINE TRUNT(N)
1907. C*
1908. C*      TRUNCATES STATISTICS OF TALLY N
1909. C*
1910.      DIMENSION RSET(1)
1911.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1912.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1913.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1914.      COMMON/GEN8/NIR,VALH,VALI
1915.      COMMON ISET(1)
1916.      EQUIVALENCE (ISET(1),RSET(1))
1917.      IF(N.GT.NTAL.OR.N.LT.1) CALL ERROR(1007,N)
1918.      LTBL=LFTB+(N-1)*8
1919.      LSTAT=ISET(LTBL)
1920.      RSET(LSTAT)=0
1921.      J=LSTAT+1
1922.      RSET(J)=0
1923.      J=LSTAT+2
1924.      RSET(J)=0
1925.      J=LSTAT+3

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234567890123

```

1926.      RSET(J)=VALH
1927.      J=LSTAT+4
1928.      RSET(J)=VALL
1929.      RETURN
1930.      END
1931.      FUNCTION TAVG(I)
1932.  C*
1933.  C*      FINDS THE AVERAGE OF TALLY I
1934.  C*
1935.      DIMENSION RSET(1)
1936.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1937.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1938.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECC,ISUM
1939.      COMMON ISET(1)
1940.      EQUIVALENCE (ISET(1),RSET(1))
1941.      IF(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
1942.      LTBL=LFTB+(I-1)*8
1943.      LSTAT=ISET(LTBL)
1944.      J=LSTAT+2
1945.      TAVG=RSET(LSTAT)/RSET(J)
1946.      RETURN
1947.      END
1948.      FUNCTION TSTD(I)
1949.  C*
1950.  C*      FINDS THE STANDARD DEVIATION OF TALLY I
1951.  C*
1952.      DIMENSION RSET(1)
1953.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1954.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1955.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1956.      COMMON ISET(1)
1957.      EQUIVALENCE (ISET(1),RSET(1))
1958.      IF(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
1959.      LTBL=LFTB+(I-1)*8
1960.      LSTAT=ISET(LTBL)
1961.      J=LSTAT+2
1962.      RN=RSET(J)
1963.      J=LSTAT+1
1964.      TSTD=(RSET(J)-RN*RSET(LSTAT))/(RN-1)
1965.      IF(TSTD.LT.0) TSTD=0.0
1966.      TSTD=SQRT(TSTD)
1967.      RETURN
1968.      END
1969.      FUNCTION TNUM (I)
1970.  C*
1971.  C*      FINDS THE NUMBER OF TALLIES IN TALLY I
1972.  C*
1973.      DIMENSION RSET(1)
1974.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
1975.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1976.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1977.      COMMON ISET(1)
1978.      EQUIVALENCE (ISET(1),RSET(1))
1979.      IF(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
1980.      LTBL=LFTB+(I-1)*8

```

123456789012345678901234567890123456789012345678901234567890123

```

1981.      LSTAT=ISET(LTBL)
1982.      J=LSTAT+2
1983.      TNUM=RSET(J)
1984.      RETURN
1985.      END
1986.      FUNCTION TMIN (I)
1987.  C*
1988.  C*      FINDS THE MINIMUM OF TALLY I
1989.  C*
1990.      DIMENSION RSET(1)
1991.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAF,LCAL
1992.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
1993.      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
1994.      COMMON ISET(1)
1995.      EQUIVALENCE (ISET(1),RSET(1))
1996.      IF(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
1997.      LTBL=LFTB+(I-1)*8
1998.      LSTAT=ISET(LTBL)
1999.      J=LSTAT+3
2000.      TMIN=RSET(J)
2001.      RETURN
2002.      END
2003.      FUNCTION TMAX (I)
2004.  C*
2005.  C*      FINDS THE MAXIMUM OF TALLY I
2006.  C*
2007.      DIMENSION RSET(1)
2008.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAF,LCAL
2009.      COMMON/GEN1/NRUNS,NFILE,TEEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
2010.      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
2011.      COMMON ISET(1)
2012.      EQUIVALENCE (ISET(1),RSET(1))
2013.      IF(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
2014.      LTBL=LFTB+(I-1)*8
2015.      LSTAT=ISET(LTBL)
2016.      J=LSTAT+4
2017.      TMAX=RSET(J)
2018.      RETURN
2019.      END
2020.      SUBROUTINE DCLCT (LDBL)
2021.  C*
2022.  C*      THIS IS FOR COLLECTING THE DISCRETE TIME PERSISTANT
2023.  C*      STATISTICS
2024.  C*
2025.      DIMENSION RSET(1)
2026.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
2027.      -DEL(99),TNOW,XX(99),DTNOW,ISTOP
2028.      COMMON ISET(1)
2029.      EQUIVALENCE (ISET(1),RSET(1))
2030.      LSTAT=ISET(LDBL)
2031.      ITYPE=ISET(LDBL+7)
2032.      IF(ITYPE.LT.0) GO TO 10
2033.      X=NQ(ITYPE)
2034.      GO TO 20
2035.  10      X=XX(-ITYPE)

```

		1	2	3	4	5	6	7
		123456789012345678901234567890123456789012345678901234567890123						
2036.	20	NU=ISET (LDBL+8)						
2037.		CALL DTPST (X,LSTAT,NU)						
2038.		RETURN						
2039.		END						
2040.		SUBROUTINE CCLCT (LCBL)						
2041.	C*							
2042.	C*	THIS IS FOR COLLECTING THE CONTINUOUS TIME PERSISTANT						
2043.	C*	STATISTICS						
2044.	C*							
2045.		DIMENSION RSET(1)						
2046.		COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,						
2047.		-DEL(99),TNOW,XX(99),DTNOW,ISTCP						
2048.		COMMON ISET(1)						
2049.		EQUIVALENCE (ISET(1),RSET(1))						
2050.		LSTAT=ISET (LCBL)						
2051.		ITYPE=ISET (LDBL+7)						
2052.		IF(ITYPE.LT.0) GO TO 10						
2053.		X=SS(ITYPE)						
2054.		GO TO 11						
2055.	10	X=DD(-ITYPE)						
2056.	11	NU=ISET (LCBL+8)						
2057.		CALL CTPST (X,LSTAT,NU)						
2058.		RETURN						
2059.		END						
2060.		FUNCTION DAVG (I)						
2061.	C*							
2062.	C*	FINDS THE AVERAGE OF DISCRETE TIME PERSISTANT STATISTICS						
2063.	C*							
2064.		DIMENSION RSET(1)						
2065.		COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCBS,LFCTB,LFAE,LCAL						
2066.		COMMON/GEN1/NRUNS,NFILE,TEEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,						
2067.		-NMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM						
2068.		COMMON ISET(1)						
2069.		EQUIVALENCE (ISET(1),RSET(1))						
2070.		IF(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)						
2071.		LCBL=LFDSB+(I-1)*9						
2072.		LSTAT=ISET (LDBL)						
2073.		J=LSTAT+6						
2074.		T=TNOW-RSET (J)						
2075.		CALL DCLCT (LDBL)						
2076.		DAVG=RSET (LSTAT)/T						
2077.		RETURN						
2078.		END						
2079.		FUNCTION CAVG (I)						
2080.	C*							
2081.	C*	FINDS THE AVERAGE OF CONTINUOUS TIME PERSISTANT STATISTICS						
2082.	C*							
2083.		DIMENSION RSET(1)						
2084.		COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCBS,LFCTB,LFAE,LCAL						
2085.		COMMON/GEN1/NRUNS,NFILE,TEEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,						
2086.		-NMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM						
2087.		COMMON ISET(1)						
2088.		EQUIVALENCE (ISET(1),RSET(1))						
2089.		IF(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)						
2090.		LCBL=LFCBS+(I-1)*9						

	1	2	3	4	5	6	7
123456789012345678901234567890123456789012345678901234567890123							

```

2091.      LSTAT=ISET(LCBL)
2092.      J=LSTAT+6
2093.      T=TNOW-RSET(J)
2094.      CALL CCLCT (LCBL)
2095.      CAVG=RSET(LSTAT)/T
2096.      RETURN
2097.      END
2098.      FUNCTION DSTD (I)
2099.  C*
2100.  C*  FINDS THE STANDARD DEVIATION OF DISCRETE TIME PERSISTANT STATISTIC
2101.  C*
2102.      DIMENSION RSET(1)
2103.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
2104.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
2105.      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
2106.      COMMON ISET(1)
2107.      EQUIVALENCE (ISET(1),RSET(1))
2108.      IF(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
2109.      LDBL=LFDSB+(I-1)*9
2110.      CALL DCLCT (LDBL)
2111.      LSTAT=ISET(LDBL)
2112.      J=LSTAT+6
2113.      T=TNOW-RSET(J)
2114.      AV=RSET(LSTAT)/T
2115.      J=LSTAT+1
2116.      DSTD=RSET(J)/T-AV*AV
2117.      IF(DSTD.LT.0) DSTD=0
2118.      DSTD=SQRT(DSTD)
2119.      RETURN
2120.      END
2121.      FUNCTION CSTD (I)
2122.  C*
2123.  C*  FINDS THE STANDARD DEVIATION OF CONTINUOUS TIME PERSISTANT STATISTIC
2124.  C*
2125.      DIMENSION RSET(1)
2126.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LFDSB,LFCSE,LFCTB,LFAE,LCAL
2127.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
2128.      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
2129.      COMMON ISET(1)
2130.      EQUIVALENCE (ISET(1),RSET(1))
2131.      IF(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
2132.      LCBL=LFCSE+(I-1)*9
2133.      CALL CCLCT (LCBL)
2134.      LSTAT=ISET(LCBL)
2135.      J=LSTAT+6
2136.      T=TNOW-RSET(J)
2137.      AV=RSET(LSTAT)/T
2138.      J=LSTAT+1
2139.      CSTD=RSET(J)/T-AV*AV
2140.      IF(CSTD.LT.0) CSTD=0
2141.      CSTD=SQRT(CSTD)
2142.      RETURN
2143.      END
2144.      FUNCTION DPRD (I)
2145.  C*

```

```

146. C* FINDS THE PERIOD OF DISCRETE TIME PERSISTANT STATISTIC
147. C*
148. DIMENSION RSET(1)
149. COMMON/LOC/LFI,LLR,LFFB,LFTB,LFESB,LFCSB,LFCTB,LFAE,LCAL
150. COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLFAR,IRAN,
151. -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NFNT,IECO,ISUM
152. COMMON ISET(1)
153. EQUIVALENCE (ISET(1),RSET(1))
154. IF(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
155. LDBL=LFDSB+(I-1)*9
156. LSTAT=ISET(LDBL)
157. J=LSTAT+6
158. DFRD=TNOW-RSET(J)
159. RETURN
160. END
161. FUNCTION CPRD (I)
162. C*
163. C* FINDS THE PERIOD OF CONTINUOUS TIME PERSISTANT STATISTIC
164. C*
165. DIMENSION RSET(1)
166. COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
167. COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLFAR,IRAN,
168. -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
169. COMMON ISET(1)
170. EQUIVALENCE (ISET(1),RSET(1))
171. IF(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
172. LCBL=LFCSB+(I-1)*9
173. LSTAT=ISET(LCBL)
174. J=LSTAT+6
175. CPRD=TNOW-RSET(J)
176. RETURN
177. END
178. FUNCTION DMIN (I)
179. C*
180. C* FINDS THE MINIMUM OF DISCRETE TIME PERSISTANT STATISTIC
181. C*
182. DIMENSION RSET(1)
183. COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
184. COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLFAR,IRAN,
185. -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
186. COMMON ISET(1)
187. EQUIVALENCE (ISET(1),RSET(1))
188. IF(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
189. LDBL=LFDSB+(I-1)*9
190. CALL DCLCT (LDBL)
191. LSTAT=ISET(LDBL)
192. J=LSTAT+3
193. DMIN=RSET(J)
194. RETURN
195. END
196. FUNCTION CMIN (I)
197. C*
198. C* FINDS THE MINIMUM OF CONTINUOUS TIME PERSISTANT STATISTIC
199. C*
200. DIMENSION RSET(1)

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

2201.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
2202.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
2203.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
2204.      COMMON ISET(1)
2205.      EQUIVALENCE (ISET(1),RSET(1))
2206.      IF(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
2207.      LCBP=LFCSB+(I-1)*9
2208.      CALL CCLCT (LCBP)
2209.      LSTAT=ISET(LCBP)
2210.      J=LSTAT+3
2211.      CMIN=RSET(J)
2212.      RETURN
2213.      END
2214.      FUNCTION DMAX (I)
2215.  C*
2216.  C*      FINDS THE MAXIMUM OF DISCRETE TIME PERSISTANT STATISTIC
2217.  C*
2218.      DIMENSION RSET(1)
2219.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
2220.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
2221.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
2222.      COMMON ISET(1)
2223.      EQUIVALENCE (ISET(1),RSET(1))
2224.      IF(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
2225.      LDBP=LFDSB+(I-1)*9
2226.      CALL DCLCT (LDBP)
2227.      LSTAT=ISET(LDBP)
2228.      J=LSTAT+4
2229.      DMAX=RSET(J)
2230.      RETURN
2231.      END
2232.      FUNCTION CMAX (I)
2233.  C*
2234.  C*      FINDS THE MAXIMUM OF CONTINUOUS TIME PERSISTANT STATISTIC
2235.  C*
2236.      DIMENSION RSET(1)
2237.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
2238.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
2239.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
2240.      COMMON ISET(1)
2241.      EQUIVALENCE (ISET(1),RSET(1))
2242.      IF(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
2243.      LCBP=LFCSB+(I-1)*9
2244.      CALL CCLCT (LCBP)
2245.      LSTAT=ISET(LCBP)
2246.      J=LSTAT+4
2247.      CMAX=RSET(J)
2248.      RETURN
2249.      END
2250.      FUNCTION ATR(LENT,I)
2251.  C*
2252.  C*      FIDS THE VALUE OF ITH ATTRIBUTE OF THE ENTRY AT LOCATION LENT
2253.  C*
2254.      DIMENSION RSET(1)
2255.      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL

```

1 2 3 4 5 6 7
1234567890123456789012345678901234567890123456789012345678901234

```

256.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
257.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
258.      COMMON ISET(1)
259.      EQUIVALENCE (ISET(1),RSET(1))
260.      IF(I.GT.NATR.OR.I.LT.1) CALL ERROR(1012,I)
261.      J=LENT+3
262.      LATR=ISET(J)
263.      J=LATR+I
264.      ATR=RSET(J)
265.      RETURN
266.      END
267.      FUNCTION LPRED (LENT)
268.  C*
269.  C*      GIVES THE PREDECESSOR OF THE ENTRY LOCATED AT LENT
270.  C*
271.      DIMENSION RSET(1)
272.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LPDSB,LPCSB,LFCTB,LPAE,LCAL
273.      COMMON ISET(1)
274.      EQUIVALENCE (ISET(1),RSET(1))
275.      LPRED=ISET(LENT)
276.      RETURN
277.      END
278.      FUNCTION LSUCC (LENT)
279.  C*
280.  C*      GIVES THE SUCCESSOR OF THE ENTRY LOCATED AT LENT
281.  C*
282.      DIMENSION RSET(1)
283.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LPDSB,LPCSB,LFCTB,LPAE,LCAL
284.      COMMON ISET(1)
285.      EQUIVALENCE (ISET(1),RSET(1))
286.      J=LENT+1
287.      LSUCC=ISET(J)
288.      RETURN
289.      END
290.      FUNCTION NQ (IFILE)
291.  C*
292.  C*      GIVES NO. OF ENTITIES IN FILE I
293.  C*
294.      DIMENSION RSET(1)
295.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LPDSB,LPCSB,LFCTB,LPAE,LCAL
296.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
297.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
298.      COMMON ISET(1)
299.      EQUIVALENCE (ISET(1),RSET(1))
300.      IF(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
301.      LPBL=LFPB+(IFILE-1)*6
302.      NQ=ISET(LPBL)
303.      RETURN
304.      END
305.      FUNCTION LFE (I)
306.  C*
307.  C*      GIVES THE LOCATION OF FIRST ENTRY OF FILE I
308.  C*
309.      DIMENSION RSET(1)
310.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LPDSB,LPCSB,LFCTB,LPAE,LCAL

```

1 2 3 4 5 6 7
123456789012345678901234567890123456789012345678901234

```

311.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
312.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
313.      COMMON ISET(1)
314.      EQUIVALENCE (ISET(1),RSET(1))
315.      IF(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
316.      LFBL=LFPB+(I-1)*6
317.      J=LFBL+1
318.      LFE=ISET(J)
319.      RETURN
320.      END
321.      FUNCTION LLE (I)
322.  C*
323.  C*      GIVES THE LOCATION OF LAST ENTRY OF FILE I
324.  C*
325.      DIMENSION RSET(1)
326.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
327.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
328.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
329.      COMMON ISET(1)
330.      EQUIVALENCE (ISET(1),RSET(1))
331.      IF(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
332.      LFBL=LFPB+(I-1)*6
333.      J=LFBL+2
334.      LLE=ISET(J)
335.      RETURN
336.      END
337.      SUBROUTINE OUTPUT
338.      RETURN
339.      END
340.      SUBROUTINE SUMRY (NNN)
341.      DIMENSION RSET(1)
342.      COMMON/LOC/LFI,LLR,LFPB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
343.      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
344.      -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
345.      COMMON/GEN2/NEQD,NEOS,NEQT,DISAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
346.      -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
347.      -ISCD(25)
348.      COMMON/GEN3/ITITLE(15),IANAL(10),IDATE(5)
349.      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATTRIB(99),SSL(99),JJ,
350.      -DDL(99),TNOW,XX(99),DTNOW,ISTCP
351.      COMMON ISET(1)
352.      EQUIVALENCE(ISET(1),RSET(1))
353.      WRITE(NPRTR,10)
354. 10      FORMAT(1H1,///50X,28HSUMMARYREPORT)
355.      WRITE(NPRTR,11) (ITITLE(I),I=1,15), (IANAL(I),I=1,10), (IDATE(I),
356.      -I=1,5), NNN, NRUNS
357. 11      FORMAT(//25X,9HPROJECT: ,15A2,15X,9HANALYST: ,10A2/
358.      -,25X,6HDATE: ,5A2,38X,3HRUN,I3,3HOF,I3)
359.      WRITE(NPRTR,12) TBEG,TCLEAR,TNOW
360. 12      FORMAT(//25X,28HSIMULATION STARTED AT TIME: ,E11.2/,25X,
361.      -28HSTATISTICS CLEARED AT TIME: ,E11.2,
362.      -/25X,14HCURRENT TIME: ,E11.2)
363.      IF(NCOUN.EQ.0) GO TO 19
364.      WRITE(NPRTR,15)
365. 15      FORMAT(///50X,28H*** COUNTER INFORMATION ****///15X,5HINDEX,

```

```

2366.      -10X,5HLABEL,20X,5SHLIMIT,10X,13HCPRENT VALUE/15X,5(1H-),
2367.      -10X,5(1H-),20X,5(1H-),10X,13(1H-))
2368.      DC 16 I=1,NCOUN
2369.      LCTB=LFCTB+(I-1)*8
2370.      J1=LCTB+1
2371.      J2=LCTB+6
2372.      J3=LCTB+7
2373.      WRITE(NPRTR,17) I,(ISET(J),J=J1,J2),ISET(J3),ISET(LCTB)
2374.      17  FORMAT(/16X,I3,11X,6A2,12X,I6,13X,I6)
2375.      16  CONTINUE
2376.      19  IF(NTAL.EQ.0) GO TO 30
2377.      WRITE(NPRTR,20)
2378.      20  FORMAT(/////50X,26H**** TALIY STATISTICS ****///15X,5HINDEX,8X,
2379.      -5HLABEL,9X,11HNO. OF OBS.,12X,4HMEAN,11X,9HSTD. DEV.,10X,
2380.      -7EMINIMUM,10X,7HMAXIMUM)
2381.      WRITE(NPRTR,201)
2382.      201  FORMAT(15X,5(1H-),8X,5(1H-),9X,11(1H-),12X,4(1H-),11X,9(1H-),
2383.      -10X,7(1H-),10X,7(1H-))
2384.      DC 21 I=1,NTAL
2385.      LTBL=LFTB+(I-1)*8
2386.      J1=LTBL+1
2387.      J2=LTBL+6
2388.      WRITE(NPRTR,22) I,(ISET(K),K=J1,J2)
2389.      22  FORMAT(/16X,I3,6X,6A2)
2390.      LSTAT=ISET(LTBL)
2391.      J=LSTAT+2
2392.      N=RSET(J)
2393.      AV=RSET(LSTAT)/RSET(J)
2394.      STD=(RSET(LSTAT+1)-RSET(J)*AV*AV)/(RSET(J)-1.)
2395.      IF(STD.LT.0) STD=0
2396.      STD=SQRT(STD)
2397.      RMN=RSET(LSTAT+3)
2398.      RMX=RSET(LSTAT+4)
2399.      WRITE(NPRTR,23) N,AV,STD,RMN,RMX
2400.      23  FORMAT(1H+,43X,I8,8X,E11.2,6X,E11.2,7X,E11.2,6X,E11.2)
2401.      21  CONTINUE
2402.      30  IF(NDTPST.EQ.0) GO TO 40
2403.      WRITE(NPRTR,31)
2404.      31  FORMAT(/////43X,45H**** DISCRETE TIME PERSISTANT STATISTICS ****
2405.      -///15X,5HINDEX,8X,5HLABEL,10X,4HMEAN,11X,9HSTD. DEV.,10X,
2406.      -7HMINIMUM,10X,7HMAXIMUM/15X,5(1H-),8X,5(1H-),10X,4(1H-),11X,
2407.      -9(1H-),10X,7(1H-),10X,7(1H-))
2408.      DO 32 I=1,NDTPST
2409.      LBL=LPDSB+(I-1)*9
2410.      J1=LDBL+1
2411.      J2=LDBL+6
2412.      WRITE(NPRTR,22) I,(ISET(K),K=J1,J2)
2413.      LSTAT=ISET(LDBL)
2414.      T=TNOW-RSET(LSTAT+6)
2415.      AV=RSET(LSTAT)/T
2416.      STD=RSET(LSTAT+1)/T-AV*AV
2417.      IF(STD.LT.0) STD=0
2418.      STD=SQRT(STD)
2419.      RMN=RSET(LSTAT+3)
2420.      RMX=RSET(LSTAT+4)

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
2421.							
2422.	34						
2423.	32						
2424.	40						
2425.							
2426.	41						
2427.							
2428.							
2429.							
2430.							
2431.							
2432.							
2433.							
2434.							
2435.							
2436.							
2437.							
2438.							
2439.							
2440.							
2441.							
2442.							
2443.							
2444.	44						
2445.	42						
2446.	50						
2447.							
2448.	130						
2449.							
2450.							
2451.							
2452.	131						
2453.	51						
2454.	60						
2455.							
2456.							
2457.							
2458.							
2459.							
2460.							
2461.							
2462.	10						
2463.	90						
2464.	100						
2465.							
2466.							
2467.							
2468.							
2469.							
2470.							
2471.							
2472.							
2473.							
2474.							
2475.							

```

WRITE(NPRTR,34) AV,STD,RMN,RMX
FORMAT(1H+,38X,E11.2,6X,E11.2,7X,E11.2,6X,E11.2)
CCONTINUE
IF(NCTPST.EQ.0) GO TO 50
WRITE(NPRTR,41)
FORMAT(/////43X,47H**** CONTINUOUS TIME PERSISTANT STATISTICS ****
-///15X,5HINDEX,8X,5HLABEL,10X,4HMEAN,11X,9HSTD. DEV.,10X,
-7HMINIMUM,10X,7HMAXIMUM/15X,5(1H-),8X,5(1H-),10X,4(1H-),11X,
-9(1H-),10X,7(1H-),10X,7(1H-))
DC 42 I=1,NCTPST
LCBL=LPCSB+(I-1)*9
J1=LCBL+1
J2=LCBL+6
WRITE(NPRTR,22) I,(ISET(K),K=J1,J2)
LSTAT=ISET(LCBL)
T=TNOW-RSET(LSTAT+6)
AV=RSET(LSTAT)/T
STD=RSET(LSTAT+1)/T-AV*AV
IF(STD.LT.0) STD=0
STD=SQRT(STD)
RMN=RSET(LSTAT+3)
RMX=RSET(LSTAT+4)
WRITE(NPRTR,44) AV,STD,RMN,RMX
FCRMT(1H+,38X,E11.2,6X,E11.2,7X,E11.2,6X,E11.2)
CONTINUE
IF(NEQT.EQ.0) GO TO 60
WRITE(NPRTR,130)
FORMAT(/////43X,35HFINAL VALUE OF CONTINUOUS VARIABLES/,43X,
-35(1H-)//)
DO 51 I=1,NEQT
WRITE(NPRTR,131) I,SS(I),I,DD(I)
FCRMT(35X,3HSS(,I2,2H)=,E10.2,18X,3HDD(,I2,2H)=,E10.2)
CONTINUE
RETURN
END
FUNCTION DRAND(ISTRM)
COMMON/GEN4/ISEED(10),LSEED(10)
DATA MULT/65539/
IF(ISTRM.LT.1.OR.ISTRM.GT.10) CALL ERROR(1013,ISTRM)
ISEED(ISTRM)=ISEED(ISTRM)*MULT
IF(ISEED(ISTRM)) 90,10,100
CALL ERROR(1014,1)
ISEED(ISTRM)=ISEED(ISTRM)+2147483647+1
DRAND=ISEED(ISTRM)
DRAND=DRAND*.4656613E-9
IF(LSEED(ISTRM).LT.0) DRAND=1.-DRAND
RETURN
END
FUNCTION UNIF(A,B,IST)
IF(A.GT.B) CALL ERROR(1015,1)
UNIF=A+(B-A)*DRAND(IST)
RETURN
END
FUNCTION ERLNG(B,A,IST)
K=A

```

	1	2	3	4	5	6	7
	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
476.							
477.							
478.							
479.							
480.	10						
481.							
482.							
483.							
484.							
485.							
486.							
487.							
488.							
489.							
490.							
491.							
492.							
493.							
494.							
495.							
496.							
497.							
498.							
499.							
500.							
501.							
502.							
503.							
504.							
505.							
506.	10						
507.							
508.							
509.							
510.							
511.							
512.							
513.							
514.							
515.	20						
516.							
517.	30						
518.							
519.							
520.							
521.							
522.							
523.							
524.							
525.							
526.	10						
527.							
528.	20						
529.							
530.							

```

IF (K.LT.1.OR.B.LT.0) CALL ERROR(1015,1)
R=1
DC 10 I=1,K
  R=R*DRAND(IST)
10 CONTINUE
  ERLNG=-B*ALOG(R)
  RETRN
END
FUNCTION EXPON(AV,IST)
IF (AV.LT.0) CALL ERROR(1015,1)
Y=DRAND(IST)
EXPON=-AV*ALOG(Y)
RETURN
END
FUNCTION RLOGN(QM,OS,IS)
IF (QS.LT.0.OR.QM.LE.0) CALL ERROR(1015,1)
QSS=QS*QS
QMS=QM*QM
XSS=ALOG(QSS/QMS+1.)
XS=SQRT(XSS)
XM=ALOG(QM)-.5*XSS
VA=RNORM(XM,XS,IS)
RLOGN=EXP(VA)
RETURN
END
FUNCTION RNORM(XMN,STD,ISTRM)
DIMENSION ENORM(10), IEVEN(10)
DATA IEVEN/10*1/
504 IF (STD.LT.0) CALL ERROR(1015,1)
505 IF (IEVEN(ISTRM).GT.1) GO TO 20
506 10 UA=2.*DRAND(ISTRM)-1.
507    UB=2.*DRAND(ISTRM)-1.
508    IEVEN(ISTRM)=2
509    W=UA*UA+UB*UB
510    IF (W.GT.1.0) GO TO 10
511    W=SQRT(-2.*ALOG(W)/W)
512    RNORM=UA*W
513    ENORM(ISTRM)=UB*W
514    GO TO 30
515 20 IEVEN(ISTRM)=1
516    RNORM=ENORM(ISTRM)
517 30 RNORM=RNORM*STD+XMN
518    RETURN
519    END
520 FUNCTION TRIAG (XL,XM,XH,ISTRM)
521 IF (XL.GT.XM.OR.XL.GT.XH.OR.XM.GT.XH) CALL ERROR(1015,1)
522 RN=DRAND(ISTRM)
523 BMA=XH-XL
524 CMA=XH-XL
525 IF (RN-BMA/CMA) 10,10,20
526 10 TRIAG=XL+SQRT(BMA*CMA*RN)
527    RETURN
528 20 TRIAG=XH-SQRT(CMA*(1.-RN)*(XH-XM))
529    RETURN
530    END

```

```

2531.      FUNCTION WEIBUL(B,A,ISTRM)
2532.      WEIBUL=(-B*ALOG(DRAND(ISTRM)))*(1./A)
2533.      RETURN
2534.      END
2535.      FUNCTION BETA(A,B,ISTRM)
2536.      X=GAMA(1.,A,ISTRM)
2537.      BETA=X/(X+GAMA(1.,B,ISTRM))
2538.      RETURN
2539.      END
2540.      FUNCTION GAM(AK,ISTRM)
2541.      K=AK
2542.      FK=K
2543.      GAM=0
2544.      IF(K) 30,30,10
2545.      PROD=1.0
2546.      DO 20 I=1,K
2547.      20 PROD=PROD*DRAND(ISTRM)
2548.      GAM=-ALOG(PROD)
2549.      30 DG=AK-FK
2550.      IF (DG-.015) 100,100,40
2551.      40 IF (DG-.985) 60,50,50
2552.      50 W=1.
2553.      GO TO 90
2554.      60 A=1./DG
2555.      B=1./(1.-DG)
2556.      70 X=DRAND(ISTRM)**A
2557.      Y=DRAND(ISTRM)**B*X
2558.      IF(Y-1.) 80,80,70
2559.      80 W=X/Y
2560.      90 Y=-ALOG(DRAND(ISTRM))
2561.      GAM=GAM+W*Y
2562.      100 RETURN
2563.      END
2564.      FUNCTION GAMA(B,A,ISTRM)
2565.      IF(A.GE.1.0) GO TO 10
2566.      GAMA=GAM(A,ISTRM)*B
2567.      RETURN
2568.      10 KA=A
2569.      AK=KA
2570.      C=A-AK
2571.      IF(A.LT.5.) GO TO 20
2572.      IF(DRAND(ISTRM).LT.C) KA=KA+1
2573.      20 PR=1.0
2574.      DO 30 I=1,KA
2575.      30 PR=PR*DRAND(ISTRM)
2576.      GAMA=-ALOG(PR)
2577.      IF(A.GE.5.) GO TO 40
2578.      TEST=(GAMA/AK)**C*EXP(-C*(GAMA/A-1.))
2579.      IF(DRAND(ISTRM).GT.TEST) GO TO 20
2580.      GAMA=GAMA*(A/AK)
2581.      40 GAMA=GAMA*B
2582.      RETURN
2583.      END
2584.      FUNCTION NPSSN(P,ISTRM)
2585.      IF(P.LT.0) CALL ERRCR(1015,1)

```

117

```

586.      NPSSN=0
587.      IF ( P-9) 10,10,40
588.      10    Y=EXP (-P)
589.      X=1.0
590.      20    X=X*DRAND (ISTRM)
591.      IF (X-Y) 50,30,30
592.      30    NPSSN=NPSSN+1
593.      GO TO 20
594.      40    Z=SQRT(P)
595.      NPSSN=RNORM (P,Z,ISTRM) +.5
596.      50    RETURN
597.      END
598.      FUNCTION DPROB (CPROB,VALUE,NVAR,ISTRM)
599.      DIMENSION CPROB (1),VALUE (1)
600.      RN=DRAND (ISTRM)
601.      DO 10 I=1,NVAR
602.      IF (RN.LE.CPROB (I)) GO TO 20
603.      10    CONTINUE
604.      CALL ERROR (1016,1)
605.      20    DPROB=VALUE (I)
606.      RETURN
607.      END

```

DISTRIBUTION LIST

18 Jun 1981

"A Generalized Simulation Language for Naval Systems Modeling"

All addressees receive one copy unless otherwise indicated.

Dr. B. Schmeiser
Department of Industrial Engineering
Purdue University
West Lafayette, IN 47907

Dr. T. Schriber
School of Business Administration
University of Michigan
Ann Arbor, MI 48109

Dr. R. E. Shannon
Department of Industrial & Systems
Engineering
University of Alabama, Huntsville
Box 1247
Huntsville, AL 35807

Mr. Roger Willis
Mr. Keith Thorp
U.S. Army TRASANA
White Sands Missile Range, NM 88002

Dr. R. Launer
Army Research Office
Durham, N.C.

Dr. Seymour Goldstein
David Taylor Naval Ship Research
& Development Center (DTNSRDC)
Carderock
Bethesda, MD 20084

Mr. Tony Rouse
U.S. Army Material Systems
Analysis Activity
Aberdeen Proving Grounds, MD 21005

Mr. Robert Flum
ASW Systems Project Office
PM-4
Department of the Navy
Washington, D.C. 20360

Mr. Douglas Kinney
Code 122
Naval Weapons Center
China Lake, CA 93555

Defense Technical Information Center
Cameron Station
Alexandria, VA 22314 2 copies

Center for Naval Analyses
2000 North Beauregard Street
Alexandria, VA 22311

Office of Naval Research
Code 434
800 N. Quincy Street
Arlington, VA 22217 2 copies

Naval Postgraduate School
Technical Library
Monterey, CA 93940

Dr. Richard E. Nance
Computer Science, Head
Virginia Polytechnic Institute
& State University
Blacksburg, VA 24061

Dr. A. Alan B. Pritsker
Department of Industrial Engineering
Purdue University
West Lafayette, IN 47907

Dr. J. William Schmidt
Department of Industrial Engineering
& Operations Research
VPT & SU
Blacksburg, VA 24061

Dr. Lee Schruben
College of Engineering
Cornell University
Ithaca, NY 14853

Distribution List (Cont'd)

All addressees receive one copy unless otherwise indicated.

Dr. S. S. Lavenberg
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 00598

Naval Ocean Systems Center
Code 16
San Diego, CA 92132

George S. Fishman
Curriculum in Operations Research
& Systems Analysis
Phillips Annex
University of North Carolina, Chapel Hill
Chapel Hill, NC 27514

Newport Laboratory
Naval Underwater Systems Center
Newport, RI 02840

New London Laboratory
Naval Underwater Systems Center
New London, CT 06320

Toke Jayachandran
Naval Postgraduate School
Monterey, CA 93940

Naval Weapons Laboratory (Code K-30)
Dahlgren, VA 22448 2 copies

Dr. D. L. Iglehart
Department of Operations Research
Stanford University
Stanford, CA 94305

Dr. Donald P. Gaver
Naval Postgraduate School
Operations Research Department
Monterey, CA 93940

Dr. Averill M. Law
1513 University Avenue
Department of Industrial Engineering
University of Wisconsin
Madison, WI 53706

Mr. Carl Bates
MATH-STAT Team, MOCA-MRA
U.S. Army Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, MD 20014

Dr. Peter A. W. Lewis
Naval Postgraduate School
Department of Administrative
Sciences
Monterey, CA 93940

Naval Air Development Center
Warminster, PA 18974

Naval Research Laboratory
Washington, D.C. 20375
Code 2627 (1 copy)
Code 5308 (1 copy)
Code 7932 (1 copy)

END

DATE
FILMED

7-81

DTIC