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--ETC(U)
JUN 81 W E BILES, A NOZARI N00014-79-C-0757

UNCLASSIFIED NL

1 OF 2
AD
A101063

LEVEL II

THE PENNSYLVANIA STATE UNIVERSITY
College of Engineering
Department of Industrial and
Management Systems Engineering,
207 Hammond Building
University Park, PA 16802

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

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

Field	Content
1. REPORT NUMBER	
2. GOVT ACCESSION NO.	AD-A101 063
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle)	A Generalized Computer Simulation Language for Naval Systems Modeling.
5. TYPE OF REPORT & PERIOD COVERED	Final Rep[redacted]t, Aug[redacted]79 — Jun[redacted]81
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	Dr. William E. Biles [redacted] Ardavan Nozari
8. CONTRACT OR GRANT NUMBER(s)	N00014-79-C-0757
9. PERFORMING ORGANIZATION NAME AND ADDRESS	The Pennsylvania State University 5 Old Main University Park, PA 16802
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	61153N RR014-11-01 NR277-284
11. CONTROLLING OFFICE NAME AND ADDRESS	Naval Analysis Program (Code 431) Office of Naval Research Arlington, VA 22217
12. REPORT DATE	June [redacted]81
13. NUMBER OF PAGES	117
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	N/A
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.

CONTENTS

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 Enscore 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 <u>external</u> 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
2. Normal
3. Lognormal
4. Gamma
5. Weibull
6. Beta
7. Uniform
8. Poisson
9. Binomial
10. Geometric
11. Negative Binomial
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 subroutines 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 that 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 predetermined 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

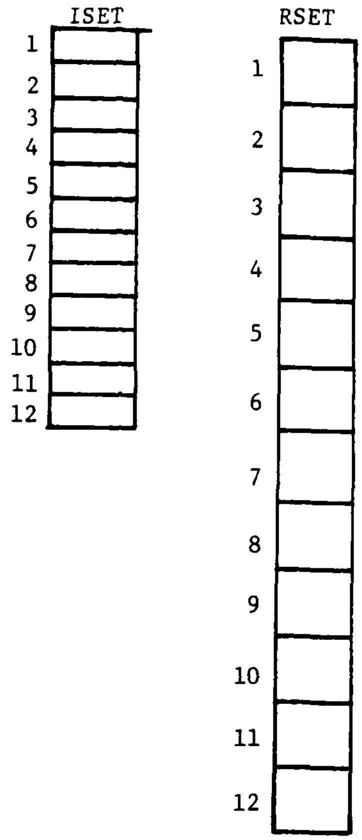


Figure 1. Filing Structure in GASP-IV and SLAM

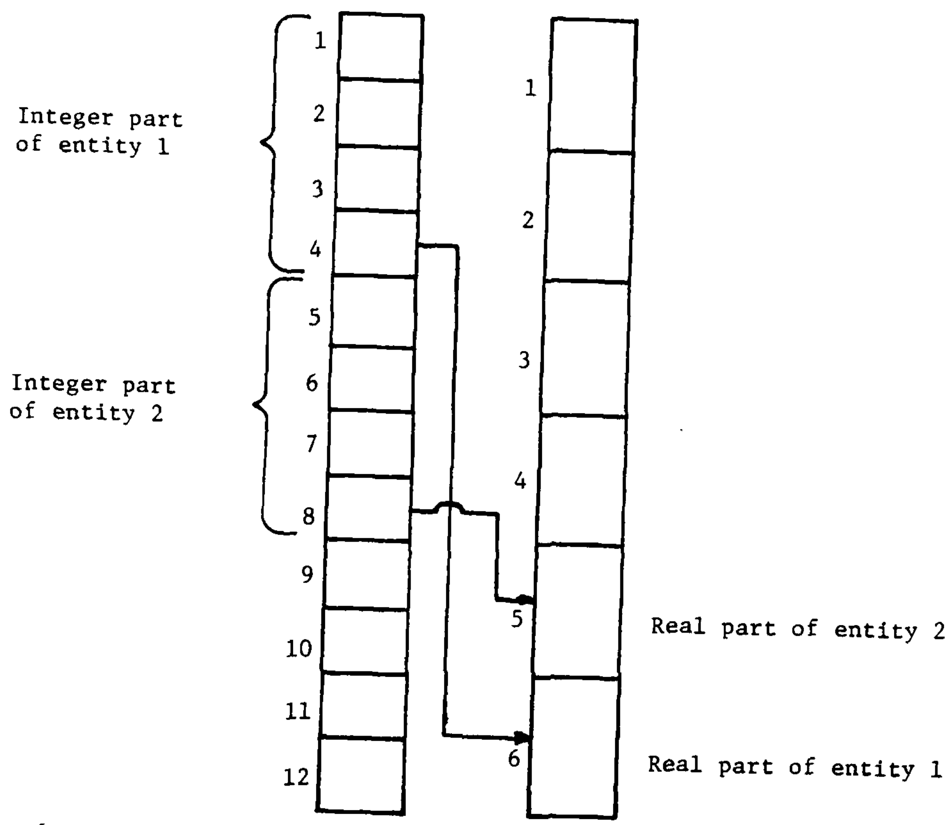


Figure 2. Filing Structure in NAVMAP

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 persistant 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

Integer	Real
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

Integer

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

Integer	Real
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

Integer		Real	
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

Integer		Real	
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

Integer	
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 NCROR, 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/GSC1/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
```

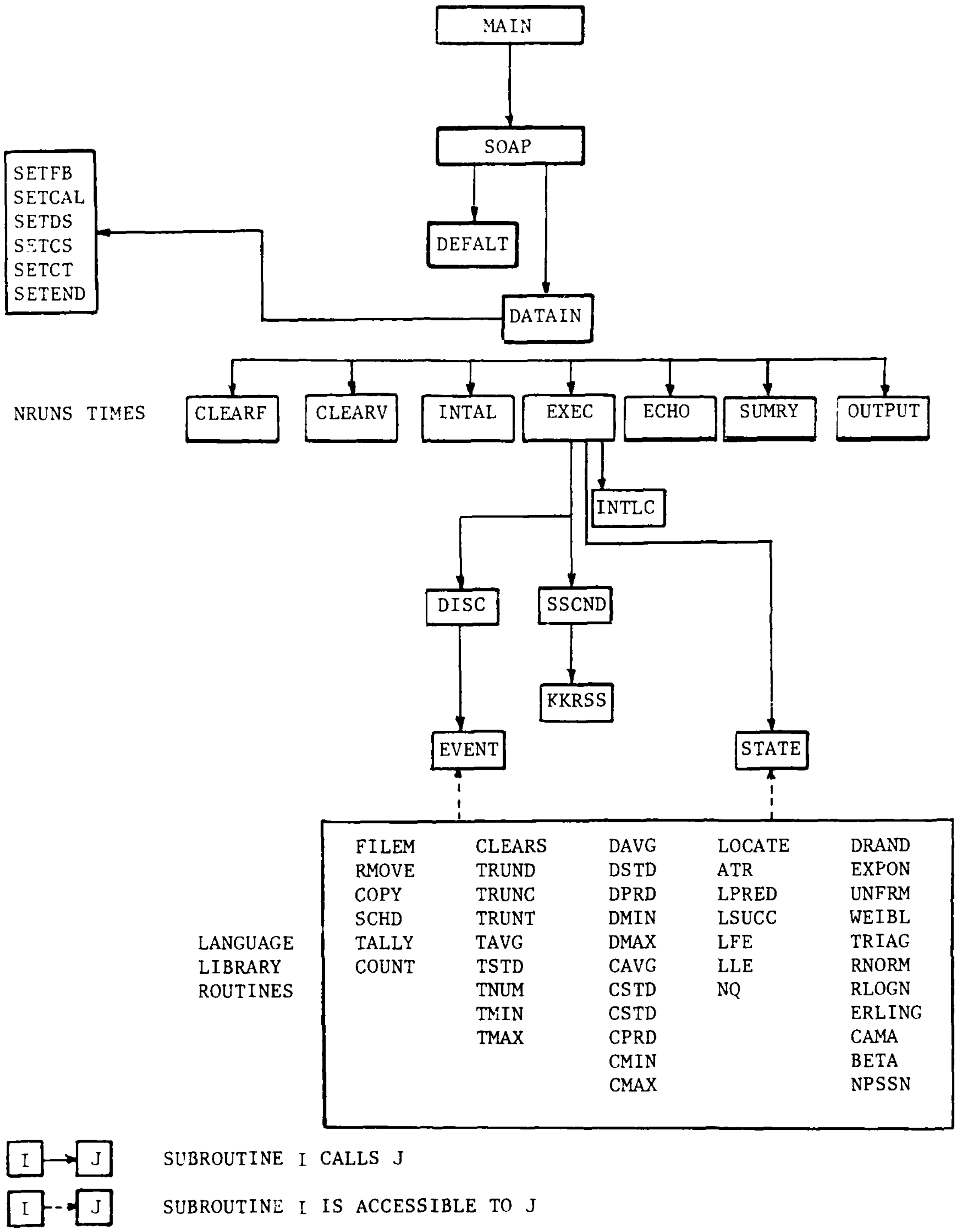


Figure 3. Organization of the NAVMAP Language

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 persistent #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, location 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 oftained 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
ATRIB(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) DEFALT

DEFALT 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,...,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.

REFERENCES

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 Mathemtical 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., "Psendo-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 Emprilcal 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.

Cols	Description	Format
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.

Cols	Description	Format
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.

Cols	Description	Format
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:

1. 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.

Cols	Description	Format
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		
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.

Cols	Description	Format
1 - 4	SEVN	
10 - 11	Event number	I2
20 - 23	Crossing variables +i SS(i) -i 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.

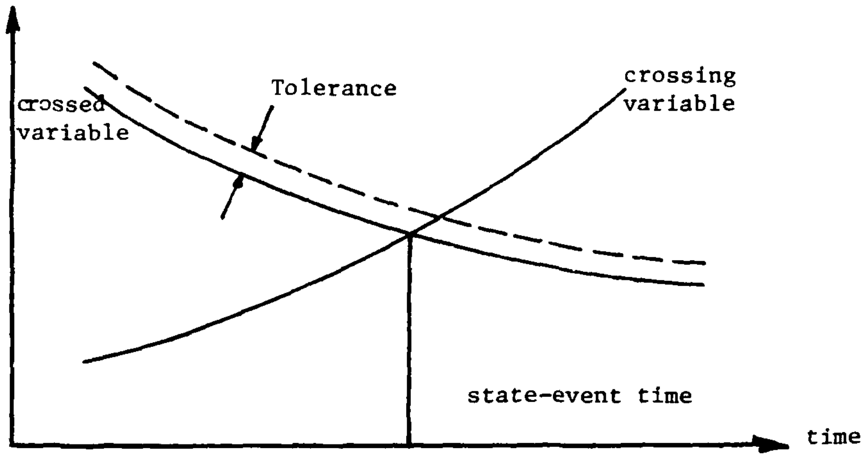
2. Define state-event 2 to occur when SS(2) crosses DD(1) in the negative direction with a tolerance of 0.01.

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

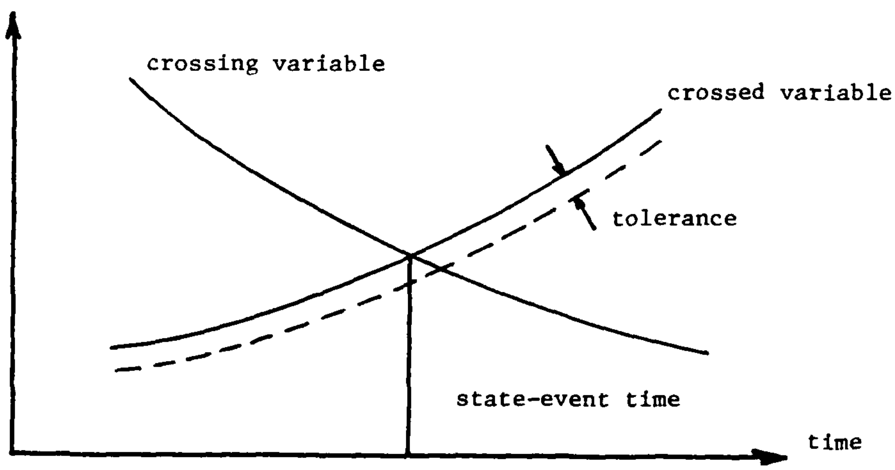
3. Define state-event 3 to occur when DD(4) crosses SS(5) in either direction with tolerance of zero.

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

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



positive crossing



negative 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.

Cols	Description	Format
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.

Cols	Description	Format
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.

Cols	Description	Format
1 - 4	DTPS	
10 - 11	Number	I2
20 - 31	Name	6A2
40 - 42	-i ith XX variable +i Number of entries of file i	I3
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).

Cols	Description	Format
1 - 4	CTPS	
10 - 11	Number	I2
20 - 31	Name	6A2
40 - 42	+i SS(i) -i DD(i)	I3
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.

Cols	Description	Format
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.

Cols	Description	Format
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.

Cols	Description	Format
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.

Cols	Description	Format
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

Cols	Description	Format
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

```
PROJ    FIRST PROB. M/M/1                      A. NOZARI          4/23/1981
DISC    01        1           02
RANK    01        1
CONT
STAT    02        02                           01
TALY    01        TIME IN SYS.
TALY    02        TIME IN QUE.
DTPS    01        NO. IN QUE.                  +01
DTPS    02        EFFICIENCY                   -01
COUN    01        NO. SERVED                   000100
SIRN     1        +1234555599
SIMU    02
INTL    0.0       20.0
```

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

 0001                 SUBROUTINE STATL
 0002                 RETURN
 0003                 END
```

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

 0001                 SUBROUTINE INTIC
 0002                 COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
 0003                 CALL SCHD (1,TNOW,ATRIB)
 0004                 RETURN
 0005                 END
```

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

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

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

                 SUBROUTINE ARVL
                 COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                -DDL(99),TNOW, XX(99),DTNOW,ISTOP
                 T=TNOW+EXPON(1.5,1)
                 CALL SCHD(1,T,ATRIB)
                 ATRIB(1)=TNOW
                 IF(XX(1).EQ.1) GO TO 2
                 XX(1)=1.
                 ATRIB(2)=TNOW
                 T=TNOW+EXPON(1.,2)
                 CALL SCHD (2,T,ATRIB)
                 RETURN
           2     CALL FILEM(1,ATRIB)
                 RETURN
                 END
```

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

                 SUBROUTINE ESRV
                 COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                -DDL(99),TNOW, XX(99),DTNOW,ISTOP
                 TSYS=TNOW-ATRIB(1)
                 TQ=ATRIB(2)-ATRIB(1)
                 CALL TALLY (1,TSYS)
                 CALL TALLY(2,TQ)
                 CALL COUNT(1,1)
                 IF (NQ(1).EQ.0) GO TO 1
                 CALL RMOVE(1,1,ATRIB)
                 ATRIB(2)=TNOW
                 T=TNOW+EXPON(1.,1)
                 CALL SCHD(2,T,ATRIB)
                 RETURN
           1     XX(1)=0
                 RETURN
                 END
```

```
PROJ   FIRST PROB. 4/M/1                    A. NOZALI          4/23/1981
DISC   1       1        2
RANK   1       1        0
CONT   0    0     0        0.0        0.0        0.0        0.0
STAT   2       2        0         1              0.0
TALY   1       TIME IN SYS.       0
TALY   2       TIME IN QUE.       0
DIPS   1       NO. IN QUE.        1        0
DIPS   2       EFFICIENCY        -1        0
COUN   1       NO. SERVED          100
STEM   1        1234555599
SIMU   2
INTL      0.0       20.000     0    0    0    0    0    0          0.0
```

```
                              T H E   E C H O   R E P O R T

THIS IS A DISCRETE MODEL

FILE INFORMATION
----------------

FILE NO.:    1
LOCATION OF FILE BLOCK:      1
RANKING DISCIPLINE IS: 1
         1: FCFS  2: LCFS  3: HVF  4: LVF
LOCATION OF STATISTICS BLOCK:     29

****LOCATION OF CALENDER IS:      7

TALLY INFORMATION
-----------------

TALLY NO.  1
LOCATION OF THE TALLY BLOCK:     13
LOCATION OF STATISITCAL ARRAY:  4936
LABEL: TIME IN SYS.
OUTPUT DEVICE NO.:    0

TALLY NO.  2
LOCATION OF THE TALLY BLOCK:     21
LOCATION OF STATISITCAL ARRAY:  4991
LABEL: TIME IN QUE.
OUTPUT DEVICE NO.:    0

DISC. TIME PERSISTANT INFORMATION
---------------------------------

DISC. TIME PERSISTANT NO.:    1
LOCATION OF BLOCK:     29
LOCATION OF STATISTICAL ARRAY:  4984
LABEL: NO. IN QUE.
TIME PERSISTANT STATISTICAL COLLECTION OF QUEUE   1
OUTPUT DEVICE NO.:    0

DISC. TIME PERSISTANT NO.:    2
LOCATION OF BLOCK:     [illegible]
LOCATION OF STATISTICAL ARRAY:  [illegible]
LABEL: EFFICIENCY
TIME PERSISTANT STATISTICAL COLLECTION OF XX( 1)
OUTPUT DEVICE NO.:    0
```

```
COUNTER INFORMATION
-------------------

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


LOCATION OF FIRST AVAILABLE ENTRY:     55


NO. OF ENTRIES ALLOCATED:    703


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

SEED( 1)=  1234555599
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
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      0.0
EVENT= 2     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      0.205
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      1.485
EVENT= 2     ATRIB(1)=      1.485    ATRIB(2)=      1.485    TNOW=      1.545
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      5.819
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      5.835
EVENT= 2     ATRIB(1)=      5.819    ATRIB(2)=      5.819    TNOW=      6.979
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      7.082
EVENT= 2     ATRIB(1)=      5.835    ATRIB(2)=      6.979    TNOW=      7.323
EVENT= 2     ATRIB(1)=      7.082    ATRIB(2)=      7.323    TNOW=      7.803
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      8.733
EVENT  1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      9.229
EVENT= 2     ATRIB(1)=      8.733    ATRIB(2)=      8.733    TNOW=      9.626
EVENT= 2     ATRIB(1)=      9.229    ATRIB(2)=      9.626    TNOW=      9.635
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=      9.675
EVENT= 2     ATRIB(1)=      9.675    ATRIB(2)=      9.675    TNOW=     10.128
EVENT= 1     ATRIB(1)=      0.0      ATRIB(2)=      0.0      TNOW=     11.704
EVENT= 1     ATRIB(1)       0.0      ATRIB(2)=      0.0      TNOW=     12.386
```

S U M M A R Y   R E P O R T

PROJECT: FIRST PROB. M/M/1
DATE: 4/23/1981

ANALYST: A. NOZARI
RUN 1 OF 2

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

**** COUNTER INFORMATIN ****

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 QUE.	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

```
EVENT= 1   ATRIB(1) =  0.0     ATRIB(2) =  0.0     TNOW=  0.0
EVENT= 2   ATRIB(1) =  0.0     ATRIB(2) =  0.0     TNOW=  2.047
EVENT= 1   ATRIB(1) =  0.0     ATRIB(2) =  0.0     TNOW=  7.059
EVENT= 1   ATRIB(1) =  0.0     ATRIB(2) =  0.0     TNOW=  8.105
EVENT= [illegible]   [illegible](1) =  7.059   ATRIB(2) =  7.059   TNOW=  8.122
```

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

EXAMPLE PROBLEM #2

PILOT EJECTION PROBLEM

CONTINUOUS SIMULATION

```
PROJ    PILOT EJECTION                 NOZARI A              4/5/81
DISC             1
CONT    04       03        .0001       .01        .00001     .000005    W
SEVN    01       +01       -1                     -60.0
SEVN    01       +02       +1                     30.0
SEVN    02       +02       +1                     4.
STAT                       04                     .01
CTPS    01       X POS.                +01
CTPS    02       Y POS.                +02
CTPS    03       SPEED                 +03
CTPS    04       THETA                 +04
SIMU    02
INTL             4.
900.
500.
```

```
0001            SUBROUTINE INTLC
0002            COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
               -DDL(99),TNOW, XX(99),DTNOW,ISTOP
0003            COMMON /USER1/CD,G,RHO,THED,VA,VE,XM,XS,Y1
0004            CD=1.
0005            G=32.2
0006            RHO=.0023769
0007            THED=15.
0008            VE=40.0
0009            XM=7
0010            XS=10.0
0011            Y1=4.0
0012            READ(NCRDR,101) VA
0013       101  FORMAT(1F10.0)
0014            THE=THED/57.3
0015            VX=VA-VE*SIN(THE)
0016            VY=VE*COS(THE)
0017            SS(3)=SQRT(VX*VX+VY*VY)
0018            SS(4)=ATAN(VY/VX)
0019            XX(1)=0.
0020            CALL SCHD (3,TNOW,ATRIB)
0021            RETURN
0022            END
```

```
FORTRAN IV G LEVEL  21                    STATE              DATE = 81113          23/24/17                PAGE 0

                 SUBROUTINE STATE
                 COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                -DDL(99),TNOW, XX(99),DTNOW,ISTOP
                 COMMON /USER1/CD,G,RHO,THED,VA,VE,XM,XS,Y1
                 DD(1)=SS(3)*COS(SS(4))-VA
                 DD(2)=SS(3)*SIN(SS(4))
                 IF(XX(1).LT.1.) RETURN
                 XD=.5*RHO*CD*XS*SS(3)*SS(3)
                 DD(3)=-XD/XM-G*SIN(SS(4))
                 DD(4)=-G*COS(SS(4))/SS(3)
                 RETURN
                 END
```

```
FORTRAN IV G LEVEL  21                    EVENT              DATE = 81113          23/24/17                PAGE 0

                 SUBROUTINE EVENT (IX)
                 COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                -DDL(99),TNOW, XX(99),DTNOW,ISTOP
                 GO TO (1,2,3) ,IX
           1     ISTOP=1
                 RETURN
           2     XX(1)=1.
                 RETURN
           3     CALL TABLE
                 RETURN
                 END
```

```
FORTRAN IV G LEVEL  21                    TABLE              DATE = 81113          23/24/17                PAGE 0

                 SUBROUTINE TABLE
                 COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                -DDL(99),TNOW, XX(99),DTNOW,ISTOP
                 T=TNOW+.01
                 CALL SCHD (3,T,ATRIB)
                 WRITE (NPRTR,101) TNOW,(SS(I),I=1,4)
         101     FORMAT( 10X,5(F10.5,10X))
                 RETURN
                 END
```

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

THE ECHO REPORT

THIS IS A COMBINED MODEL WITH 4 CONTINUOUS EQUATIONS

FILE INFORMATION

NO FILES ARE DESIGNATED

****LOCATION OF CALENDER 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 (IERR):	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: 4934
LABEL: X POS.
TIME PERSISTANT STATISTICAL COLLECTION ON SS(1)
OUTPUT DEVICE NO.: 0

CONT. TIME PERSISTANT NO.: 2
LOCATION OF BLOCK: 16
LOCATION OF STATISTICAL ARRAY: 4967
LABEL: Y POS.
TIME PERSISTANT STATISTICAL COLLECTION ON SS(2)
OUTPUT DEVICE NO.: 0

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

COMT. TIME PERSISTANT NO.:   4
LOCATION OF BLOCK:     34
LOCATION OF STATISTICAL ARRAY:  4973
LABEL: THETA
TIME PERSISTANT 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             890.48657        0.04340

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

S U M M A R Y   R E P O R T

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)=  0.59E+03        DD( 3)= -0.60E+03
SS( 4)=  0.28E-01        DD( 4)= -0.54E-01
```

EXECUTION IS ATTEMPTED

0.0	0.0	0.0	491.16992	0.07874

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

0.01000	-0.10340	0.38591	491.16992	0.07874
0.02000	-0.20692	0.77228	491.16992	0.07874
0.03000	-0.31045	1.15866	491.16992	0.07874
0.04000	-0.41397	1.54503	491.16992	0.07874
0.05000	-0.51749	1.93140	491.16992	0.07874
0.06000	-0.62101	2.31777	491.16992	0.07874
0.07000	-0.72454	2.70414	491.16992	0.07874
0.08000	-0.82806	3.09051	491.16992	0.07874
0.09000	-0.93158	3.47688	491.16992	0.07874
0.10000	-1.03511	3.86325	491.16992	0.07874

SPECIFIED TOLERANCE EXCEEDED FOR SS(2) AT TIME 0.1036E+00

0.11000	-1.14671	4.24835	488.58521	0.07833
0.12000	-1.29593	4.62749	484.54053	0.07767
0.13000	-1.48490	5.00032	480.56226	0.07701
0.14000	-1.71296	5.36691	476.64844	0.07634
0.15000	-1.97948	5.72734	472.79785	0.07566
0.16000	-2.28386	6.08168	469.00879	0.07498
0.17000	-2.62547	6.43001	465.27979	0.07429
0.18000	-3.00374	6.77238	461.60937	0.07360
0.19000	-3.41808	7.10888	457.99634	0.07290
0.20000	-3.86795	7.43956	454.43921	0.07219
0.21000	-4.35279	7.76449	450.93677	0.07149
0.22000	-4.87207	8.08373	447.48779	0.07077
0.23000	-5.42527	8.39735	444.09106	0.07005
0.24000	-6.01187	8.70541	440.74536	0.06932
0.25000	-6.63137	9.00796	437.44971	0.06859
0.26000	-7.28328	9.30507	434.20288	0.06785
0.27000	-7.96714	9.59678	431.00366	0.06711
0.28000	-8.68246	9.88316	427.85132	0.06636
0.29000	-9.42880	10.16426	424.74463	0.06561
0.30000	-10.20570	10.44012	421.68262	0.06485
0.31000	-11.01273	10.71082	418.66455	0.06409
0.32000	-11.84946	10.97638	415.68921	0.06332
0.33000	-12.71547	11.23687	412.75586	0.06254
0.34000	-13.61035	11.49233	409.86353	0.06176
0.35000	-14.53369	11.74282	407.01147	0.06097
0.36000	-15.48511	11.98836	404.19873	0.06018
0.37000	-16.46422	12.22902	401.42456	0.05938
0.38000	-17.47064	12.46484	398.68823	0.05858
0.39000	-18.50400	12.69585	395.98901	0.05777
0.40000	-19.56393	12.92211	393.32617	0.05695
0.41000	-20.65010	13.14365	390.69873	0.05613
0.42000	-21.76215	13.36052	388.10620	0.05531
0.43000	-22.89972	13.57276	385.54810	0.05448
0.44000	-24.06250	13.78040	383.02344	0.05364
0.45000	-25.25015	13.98348	380.53174	0.05280
0.46000	-26.46236	14.18205	378.07227	0.05195
0.47000	-27.69879	14.37614	375.64429	0.05110
0.48000	-28.95915	14.56578	373.24756	0.05024
0.49000	-30.24315	14.75102	370.88110	0.04937
0.50000	-31.55046	14.93188	368.54468	0.04850
0.51000	-32.88081	15.10840	366.23755	0.04763
0.52000	-34.23392	15.28062	363.95923	0.04675
0.53000	-35.60948	15.44857	361.70923	0.04586
0.54000	-37.00723	15.61228	359.48682	0.04497
0.55000	-38.42691	15.77179	357.29175	0.04407
0.56000	-39.86824	15.92712	355.12329	0.04317
0.57000	-41.33096	16.07829	352.98120	0.04226
0.58000	-42.81482	16.22536	350.86499	0.04135
0.59000	-44.31956	16.36833	348.77417	0.04043
0.60000	-45.84494	16.50726	346.70825	0.03950
0.61000	-47.39072	16.64217	344.66675	0.03857
0.62000	-48.95665	16.77307	342.64941	0.03763
0.63000	-50.54250	16.90001	340.65552	0.03669
0.64000	-52.14803	17.02299	338.68481	0.03574
0.65000	-53.77303	17.14207	336.73706	0.03479
0.66000	-55.41727	17.25726	334.81177	0.03383
0.67000	-57.08054	17.36858	332.90845	0.03287
0.68000	-58.76260	17.47606	331.02686	0.03190

EXAMPLE PROBLEM #3

CEDAR BOG LAKE PROBLEM

STATE-EVENT SIMULATION

```
PROJ      CEDAR BOG                           NOZARI A              4/4/81
DISC
CONT      05     01              .00025       .025       .00001     .00001     W
STAT                             06                      .025
CTPS      01           PLANTS                 0-1
CTPS      02           HERBIVORS              0-2
CTPS      03           CARNIVCRAS             0-3
CIPS      04           ORGANIC                0-4
CTPS      05           ENVIRCNMENT            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              SUBROUTINE INTLC
 0002              COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                  -DDL(99),TNOW, XX(99),DTNOW,ISTOP
 0003              SS(1)=.83
 0004              SS(2)=.003
 0005              SS(3)=.0001
 0006              SS(4)=0.0
 0007              SS(5)=0.0
 0008              RETURN
 0009              END
```

```
FORTRAN IV G LEVEL  21                    EVENT               DATE = 81115          09/05/46              PAGE 000

 0001              SUBROUTINE EVENT(I)
 0002              RETURN
 0003              END
```

```
FORTRAN IV G LEVEL  21                    STATE               DATE = 81115          09/05/46              PAGE 000

 0001              SUBROUTINE STATE
 0002              COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
                  -DDL(99),TNOW, XX(99),DTNOW,ISTOP
 0003              DATA PI/3.14159/
 0004              SS(6)=95.9*(1.+.635*SIN(2.*PI*TNOW))
 0005              DD(1)=SS(6)-4.03*SS(1)
 0006              DD(2)=.48*SS(1)-17.87*SS(2)
 0007              DD(3)=4.85*SS(2)-4.65*SS(3)
 0008              DD(4)=2.55*SS(1)+6.12*SS(2)+1.95*SS(3)
 0009              DD(5)=SS(1)+6.9*SS(2)+2.7*SS(3)
 0010              RETURN
 0011              END
```

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

S U M M A R Y   R E P O R T

PROJECT: FIRST PROB. 4/8/1
DATE: 4/25/1981

ANALYST: A. NOZARI
RUN 2 OF 2

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

**** COUNTER INFORMATIN ****

INDEX	LABEL	LIMIT	CURRENT VALUE
1	NO. SERVED	100	11

**** TALLY STATISTICS ****

INDEX	LABEL	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 PERSISTANT STATISTICS ****

INDEX	LABEL	MEAN	STD. DEV.	MINIMUM	MAXIMUM
1	NO. IN QUE.	0.73E+00	[illegible]	0.0	0.30E+01
2	EFFICIENCY	0.63E+00	[illegible]	0.0	0.10E+01

T H E   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 (RERR):                 0.00001
TIME BETWEEN SAVE POINTS (DTSAV):            0.02500
ACCURACY ERROR SPECIFICATION (IERR):               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: HERBIVORS
TIME PERSISTANT STATISTICAL COLLECTION ON DD( 2)
OUTPUT DEVICE NO.:    0

CONT. TIME PERSISTANT NO.:    3
LOCATION OF BLOCK:     17
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:     31
LOCATION OF STATISTICAL ARRAY:  4966
```

```
LABEL: ENVIRONMENT
TIME PERSISTANT STATISTICAL COLLECTION ON DD( 5)
OUTPUT DEVICE NO.:    0

CONT. TIME PERSISTANT NO.:    6
LOCATION OF BLOCK:      46
LOCATION OF STATISTICAL ARRAY:  4959
LABEL: SCLAR
TIME PERSISTANT STATISTICAL COLLECTION ON SS( 1)
OUTPUT DEVICE NO.:    0


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)=  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
```

S U M M A R Y   R E P O R T

PROJECT: CEDAR BOG
DATE: 4/4/81

ANALYST: FOSARI 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	HERBIVORS	0.22E+00	0.11E+01	-0.13E+01	0.23E+01
3	CARNIVORAS	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)= 0.28E+02
SS(2)= 0.44E+00	DD(2)= 0.36E+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

FORTRAN LISTING FOR NAVAL SYSTEMS SIMULATION PROGRAM

```
                 1         2         3         4         5         6
        123456789012345678901234567890123456789012345678901234567890123456789

              DIMENSION RSET(5000)
              COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
              COMMON ISET(5000)
              COMMON/GSC1/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
              SUBROUTINE SOAP
              DIMENSION RSET(1)
              COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
              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))
              DATA STARS/2H**/
              CALL DEFALT
              CALL DATAIN
              IF(IECO.EQ.1) GO TO 10
              CALL ECHO
        10    DO 20 I=1,NRUNS
                  IF(ICLF.EQ.1) GO TO 11
                  CALL CLEARF
        11        IF(ICLV.EQ.1) GO TO 12
                  CALL CLEARV
        12        CALL INTAL
                  CALL EXEC
                  IF(ISUM.EQ.1) GO TO 13
                  CALL SUMRY (I)
        13        IF(NDTPST.EQ.0) GO TO 15
                      DO 14 J=1,NDTPST
                      L=LFDSB+(J-1)*9
                      NU=ISET(L+8)
                      IF(NU.LE.0) GO TO 14
                      WRITE(NU) STARS
        14            CONTINUE
        15        IF(NCTPST.EQ.0) GO TO 19
                      DO 16 J=1,NCTPST
                      L=LFCSB+(J-1)*9
                      NU=ISET(L+8)
                      IF(NU.LE.0) GO TO 16
                      WRITE(NU) STARS
        16            CONTINUE
        19        CALL OUTPUT
        20    CONTINUE
              STOP
              END
              SUBROUTINE DEFALT
              DIMENSION RSET(1),LL(10)
              DATA LL/1274321477,2135124613,1743251541,1624217675,2014632579,
```

```
                1                   2                   3                   4                   5                   6
       123456789012345678901234567890123456789012345678901234567890123456789

     -2036774231 ,1452313571,1254240657,1410143363,2135621895/
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
      COMMON/GEN4/ISEED(10),LSEED(10)
      COMMON/GEN8/NIR,VALH,VALL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
C*
C*    THIS SUBROUTINE ASSIGNS THE DEFAULT VALUES
C*    TO THE SOAP VARIABLES.
C*
      NIR=1
      VALH=1.E20
      VALL=-1.E20
      ICLF=0
      ICLV=0
      ICLS=0
      CALL CLEARV
      NFILE=0
      LFFB=0
      LFTB=0
      LFDSB=0
      LFCSB=0
      LFAE=0
      LCAL=0
      NTMX=0
      NENT=0
      NATR=0
      NTAL=0
      NDTPST=0
      NCTPST=0
      N=LLR-1
      DO 1 I=1,N
          ISET(I)=0
1     CONTINUE
      DO 2 I=1,10
          ISEED(I)=LL(I)
          LSEED(I)=LL(I)
2     CONTINUE
      RETURN
      END
      SUBROUTINE ECHO
      DIMENSION RSET(1),INFO(3)
      DATA INFO/1HN,1HW,1HF/
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
      COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSE
     -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
     -ISCD(25)
      COMMON/GEN4/ISEED(10),LSEED(10)
      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
      COMMON ISET(1)
```

```
               1         2         3         4         5         6
      123456789012345678901234567890123456789012345678901234567890123456789

      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 CONTINUOUS MODEL WITH ,I4,
     -21H CONTINUOUS EQUATIONS)
      GO TO 20
11    WRITE(NPRTR,101) NEQT
101   FORMAT(//10X,29HTHIS IS A COMBINED 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=LFFB+(I-1)*6
      WRITE(NPRTR,105) L
105   FORMAT(10X,23HLOCATION OF FILE BLOCK:,I6)
      J=L+3
      WRITE(NPRTR,106) ISET(J)
 106   FORMAT(10X,22HRANKING DISCIPLINE IS:,I2/
     -,20X,32H1: FCFS  2: LCFS  3: HVF  4: LVF)
      J=L+4
      IF(ISET(J).EQ.0) GO TO 14
      WRITE(NPRTR,107) ISET(J)
107   FORMAT(10X,21HRANKING ATRIBUTE IS :,I3)
14    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,DTSAV
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,
```

```
                  1         2         3         4         5         6
         123456789012345678901234567890123456789012345678901234567890123456789
            -33HTIME BETWEEN SAVE POINTS (DTSAV):,9X,F10.5)
             IF(IERR.EQ.-1) WRITE(NPRTR,122) INFO(1)
             IF(IERR.EQ.0) WRITE(NPRTR,122) INFO(2)
             IF(IERR.EQ.1) WRITE(NPRTR,122) INFO(3)
     122     FORMAT(10X,36HACCURACY ERROR SPECIFICATION (IERR):15X,A1)
             IF(NSEV.LE.0) GO TO 21
             WRITE(NPRTR,130)
     130     FORMAT(///10X,12HSTATE EVENTS/10X,12(1H-)//10X,6HNUMBER,10X,5HE
            -T,11X,8HCROSSING,10X,9HDIRECTION,10X,7HCROSSED,10X,9HTOLERANCE/
            -,8HVARIABLE,10X,11HOF CROSSING,8X,5HVALUE,12X,11HOF CROSSING//)
             DO 28 I=1,NSEV
             WRITE(NPRTR,131) I,ICSV(I)
     131     FORMAT(12X,I2,13X,I3)
             IF(ICGV(I).LT.0) GO TO 23
             WRITE(NPRTR,132) ICGV(I)
     132     FORMAT(1H+,42X,3HSS(,I2,1H))
             GO TO 24
     23      J=-ICGV(I)
             WRITE(NPRTR,133) J
     133     FORMAT(1H+,42X,3HDD(,I2,1H))
     24      WRITE(NPRTR,134) ICDIR(I)
     134     FORMAT(1H+,64X,I2)
             IF(ICDV(I).LT.0) GO TO 25
             IF(ICDV(I).GT.0) GO TO 26
             WRITE(NPRTR,135) VAL(I)
     135     FORMAT(1H+,79X,F10.5)
             GO TO 27
     25      J=-ICDV(I)
             WRITE(NPRTR,136) J
     136     FORMAT(1H+,79X,3HDD(,I2,1H))
             GO TO 27
     26      WRITE(NPRTR,137) ICDV(I)
     137     FORMAT(1H+,79X,3HSS(,I2,1H))
     27      WRITE(NPRTR,138) TTOL(I)
     138     FORMAT(1H+,96X,F10.5)
     28      CONTINUE
     21      IF(NTAL.EQ.0) GO TO 31
             WRITE(NPRTR,150)
     150     FORMAT(///10X,17HTALLY INFORMATION/10X,17(1H-))
             DO 30 I=1,NTAL
             L=LFTB+(I-1)*8
             WRITE(NPRTR,151) I
     151     FORMAT(//10X,9HTALLY NO., I3)
             WRITE(NPRTR,152) L
     152     FORMAT(10X,28HLOCATION OF THE TALLY BLOCK:,I6)
             WRITE(NPRTR,153) ISET(L)
     153     FORMAT(10X,30HLOCATION OF STATISTICAL ARRAY:,I6)
             J1=L+1
             J2=L+6
             WRITE(NPRTR,154) (ISET(J),J=J1,J2)
     154     FORMAT(10X,7HLABEL: ,6A2)
             J=L+7
             WRITE(NPRTR,155) ISET(J)
     155     FORMAT(10X,18HOUTPUT DEVICE NO.:,I4)
     30      CONTINUE
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345

31    IF(NDTPST.EQ.0) GO TO 41
      WRITE(NPRTR,160)
160   FORMAT(///10X,33HDISC. TIME PERSISTANT INFORMATION/10X,32(1H-))
      DC 40 I=1,NDTPST
      L=LFDSB+(I-1)*9
      WRITE(NPRTR,161) I
161   FORMAT(//10X,27HDISC. TIME PERSISTANT NO.: ,I3)
      WRITE(NPRTR,162) L
162   FORMAT(10X,18HLOCATION OF BLOCK:,I6)
      WRITE(NPRTR,163) ISET(L)
163   FORMAT(10X,30HLOCATION OF STATISTICAL ARRAY: ,I6)
      J1=L+1
      J2=L+6
      WRITE(NPRTR,154) (ISET(J),J=J1,J2)
      J=L+7
      IF(ISET(J).GT.0) GO TO 42
      J=-ISET(J)
      WRITE(NPRTR,165) J
165   FORMAT(10X,45HTIME PERSISTANT STATISTICAL COLLECTION ON XX(
     -,I2,1H))
      GO TO 43
42    WRITE(NPRTR,166) ISET(J)
166   FORMAT(10X,47HTIME PERSISTANT STATISTICAL COLLECTION ON QUEUE,I3)
43    J=L+8
      WRITE(NPRTR,155) ISET(J)
40    CONTINUE
41    IF(NCTPST.EQ.0) GO TO 51
      WRITE(NPRTR,170)
170   FORMAT(///10X,33HCONT. TIME PERSISTANT INFORMATION/10X,33(1H-))
      DC 50 I=1,NCTPST
      L=LFCSB+(I-1)*9
      WRITE(NPRTR,171) I
171   FORMAT(//10X,27HCONT. TIME PERSISTANT NO.: ,I3)
      WRITE(NPRTR,172) L
172   FORMAT(10X,18HLOCATION OF BLOCK:,I6)
      WRITE(NPRTR,173) ISET(L)
173   FORMAT(10X,30HLOCATION OF STATISTICAL ARRAY: ,I6)
      J1=L+1
      J2=L+6
      WRITE(NPRTR,154) (ISET(J),J=J1,J2)
      J=L+7
      IF(ISET(J).GT.0) GO TO 52
      J=-ISET(J)
      WRITE(NPRTR,175) J
175   FORMAT(10X,45HTIME PERSISTANT STATISTICAL COLLECTION ON DD(
     -,I2,1H))
      GO TO 53
52    WRITE(NPRTR,176) ISET(J)
176   FORMAT(10X,45HTIME PERSISTANT STATISTICAL COLLECTION ON SS(
     -,I2,1H))
53    J=L+8
      WRITE(NPRTR,155) ISET(J)
50    CONTINUE
51     IF(NCOUN.EQ.0) GO TO 61
      WRITE(NPRTR,180)
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345
180    FORMAT(///10X,19HCOUNTER INFORMATION/10X,19(1H-))
       DO 60 I=1,NCOUN
       L=LFCTB+(I-1)*8
       WRITE(NPRTR,181) I
181    FORMAT(//10X,12HCOUNTER NO.:,I3)
       WRITE(NPRTR,182) L
182    FORMAT(10X,26HLOCATION OF COUNTER BLOCK: ,I6)
       J1=L+1
       J2=L+6
       WRITE(NPRTR,154) (ISET(J),J=J1,J2)
       J=L+7
       WRITE(NPRTR,183) ISET(J)
183    FORMAT(10X,7HLIMIT :,I7)
60     CONTINUE
61     IF(NENT.EQ.0) GO TO 62
       WRITE(NPRTR,190) LFAE
190    FORMAT(///10X,34HLOCATION OF FIRST AVAILABLE ENTRY:,I6)
       WRITE(NPRTR,191) NENT
191    FORMAT(//10X,25HNO. OF ENTRIES ALLOCATED:,I6)
62     WRITE(NPRTR,140)
140    FORMAT(///10X,25HRANDOM STREAM INFORMATION/10X,25(1H-)//)
       DO 70 I=1,10
       WRITE(NPRTR,141) I,ISEED(I)
141    FORMAT(10X,5HSEED(,I2,2H)=,2X,I10)
70     CONTINUE
       IF (IRAN.EQ. 1) GO TO 80
       WRITE(NPRTR,200)
200    FORMAT(/10X,42HSEEDS WILL NOT BE INITIALIZED BETWEEN RUNS)
       GO TO 81
80     WRITE(NPRTR,201)
201    FORMAT(/10X,38HSEEDS WILL BE INITIALIZED BETWEEN RUNS)
81     WRITE(NPRTR,202) NRUNS
202    FORMAT(///10X,8HTOTAL OF,I4,23H RUNS WILL BE PERFORMED)
       RETURN
       END
       SUBROUTINE CLEARF
C*
C*      THIS ROUTINE CLEARS THE FILES BETWEEN THE RUNS
C*
       DIMENSION RSET(1)
       COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
       COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
      -NIMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
       COMMON ISET(1)
       EQUIVALENCE (ISET(1),RSET(1))
       IF (LCAL.EQ.0) RETURN
       IF (NFILE.EQ.0) GO TO 5
       L=LFFB
       GO TO 6
5      L=LCAL
6      N=NFILE+1
       DO 10 K=1,N
          I=L+(K-1)*6
          ISET(I)=0
C*
```

```
          1         2         3         4         5         6         7
 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345

 C*     JOIN ALL THE ENTRIES TO THE FCCL BY TAKING THE LAST
 C*     ENTRY OF THE FILE AND SETTING ITS SUCCESSOR AS LFAE
 C*     AND UPDATING LFAE
 C*
           I1=I+1
           I2=I+2
           LLFE=ISET(I1)
           LLLE=ISET(I2)
           IF(LLFE.EQ.0) GO TO 10
           J=LLLE+1
           ISET(J)=LFAE
           LFAE=LLFE
           ISET(I1)=0
           ISET(I2)=0
 10     CONTINUE
        RETURN
        END
        SUBROUTINE CLEARV
        COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
       -DDL(99),TNOW, XX(99),DTNOW,ISTOP
        DO 10 I=1,99
           SS(I)=0
           DD(I)=0
           SSL(I)=0
           DDL(I)=0
           XX(I)=0
           ATRIB(I)=0
 10     CONTINUE
        JJ=0
        RETURN
        END
        SUBROUTINE INTAL
 C*
 C*     THIS IS FOR INITIALIZATION BEFORE EACH RUN
 C*
        DIMENSION RSET(1)
        COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
        COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
       -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
        COMMON/GEN4/ISEED(10),LSEED(10)
        COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
       -DDL(99),TNOW, XX(99),DTNOW,ISTOP
        COMMON ISET(1)
        EQUIVALENCE (ISET(1),RSET(1))
        IF(NCOUN.EQ.0) GO TO 2
        DO 1 I=1,NCOUN
           LCTB=LFCTB+(I-1)*8
           ISET(LCTB)=0
 1      CONTINUE
 2      ISTOP=0
        IF(LCAL.EQ.0) GO TO 3
        IF(ICLS.EQ.1) GO TO 3
        CALL SCHD(-1,TCLEAR,ATRIB)
 3      IF(IRAN.EQ.0) RETURN
        DO 4 I=1,10
```

```
                1         2         3         4         5         6         7
       1234567890123456789012345678901234567890123456789012345678901234567890123

                    ISEED(I)=LSEED(I)
       4        CONTINUE
                RETURN
                END
                SUBROUTINE EXEC
                DIMENSION A2(100),A3(100),A4(100),A5(100)
                DIMENSION RSET(1)
                COMMON/LOC/LFI,LLR,LFFR,LFTB,LFDSB,LFCSB,LFCTB,LFAE,ICAL
                COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
               -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
                COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
               -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
               -ISCD(25)
                COMMON/GEN8/NIR,VALH,VALL
                COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
               -DDL(99),TNOW, XX(99),DTNOW,ISTOP
                COMMON ISET(1)
                EQUIVALENCE(ISET(1),RSET(1))
                WRITE(NPRTR,200)
       200      FORMAT(//10X,22HEXECUTION IS ATTEMPTED)
       C*
       C*       INITIALIZATION
       C*
                TTSAV=VALH
                RESLS=.01 * ABS(DTSAV)
                RESLS = AMIN1(DTMIN,RESLS)
                DTACC=DTMAX
                TNOW=TBEG
                FRMUL=.5
                TMUL=.9
                ISEES=0
                SCALL=1.0
                IF(RERR.GT.0.) SCALL=2./RERR
                AEM=SCALL*AERR
                CALL INTLC
                IF(NDT   T.EQ.0) GO TO 10
                DC 5 I=  NDTPST
                    LDBL=LFDSB+(I-1)*9
                    CALL DCLCT(LDBL)
       5        CONTINUE
       10       IF(NEQT.EQ.0) GO TO 30
                TLAST=TNOW
                DTNOW=0.0
                IF(NSEV.GT.0) CALL SSCND
                DC 15 I=1,NEQT
                    DDL(I)=DD(I)
                    SSL(I)=SS(I)
                  NTINUE
                      TATE
                      AV.E..0) GO TO 16
                        AV
                          AB (DTSAV)
```

```
                   1         2         3         4         5         6         7
          1234567890123456789012345678901234567890123456789012345678901234567890123 4
                IF(ISCD(I).EQ.0.OR.ISCD(I)*ICDIR(I).LT.0) GO TO 24
                IX=ICSV(I)
                CALL SCHD(IX,TNOW,ATRIB)
       24       ISCD(I)=0
       25    CONTINUE
      C*
      C*     NEW ITERATION
      C*
       30    TNEXT=VALH
             IF(LCAL.EQ.0) GO TO 40
             IF(ISET(LCAL).EQ.0) GO TO 35
             J=LCAL+1
             LLFE=ISET(J)
             LATR=ISET(LLFE+3)
             TNEXT=RSET(LATR)
       35    IF(NEQT.GT.0) GO TO 40
             IF(TNEXT.GT.TFIN. OR.ISTOP.EQ.1) GO TO 1000
             TNOW=TNEXT
             CALL DISC
             GO TO 30
       40    IF(TNEXT.GT.TFIN.AND.TNOW.EQ.TFIN) GO TO 1000
             IF(TNEXT.GT.TFIN) TNEXT=TFIN
       45    IF(ISTOP.EQ.1) GO TO 1000
      C*
      C*     SAVE CONTINUOUS VARIABELS IF IT IS SAVING TIME
      C*
             IF(ABS(TTSAV-TNOW)-RESLS) 50,50,55
       50    CALL SSAVE
             TTSAV=TTSAV+ABS(DTSAV)
       55    IF  (TNEXT-TNOW) 60,65,70
       60    WRITE(NPRTR,960) TNEXT,TNOW
             CALL ERROR(9,1)
       65    IF(DTSAV.LT.0)CALL SSAVE
             CALL DISC
             IF(DTSAV.GE.0) GO TO 30
      C*
      C*     SAVE THE VALUE OF THE CTPST VAR. IF THEY HAVE CHANGED
      C*     IN THE EVENT
      C*
             DO 68 I=1,NCTPST
             J=LFCSB+(I-1)*9
             LSTAT=ISET(J)
             ITYPE=ISET(J+7)
             IF(ITYPE.LT.0) GO TO 66
             X=SS(ITYPE)
             GO TO 67
       66    ITYPE=-ITYPE
             X=DD(ITYPE)
       67    IF(X.EQ.RSET(LSTAT)) GO TO 68
             NU=ISET(J+8)
             CALL CTPST(X,LSTAT,NU)
       68    CONTINUE
             GO TO 30
       70    IF(NEQD.GT.0) GO  TO 75
             DTFUL=DTMAX
```

```
              1         2         3         4         5         6         7
     1234567890123456789012345678901234567890123456789012345678901234567890123
      GC TO 80
75    DTFUL=DTACC
80    IF(TLAST+DTFUL.GT.TTSAV) DTFUL=TTSAV-TLAST
      IF(TLAST+DTFUL.GT.TNEXT) DTFUL=TNEXT-TLAST
      IF(NEQD.GT.0) GO TO 240
85    DTNOW=DTFUL
      TNOW=TLAST+DTNOW
      CALL STATE
90    IF(NSEV.GT.0) CALL SSCND
      IF (ISEES) 380,95,450
C*
C*    NC STATE EVENT HAS CCCURED
C*
95    TLAST=TNOW
      DO 100 I=1,NEQT
          DDL(I)=DD(I)
          SSL(I)=SS(I)
100   CONTINUE
      IF(TNOW.EQ.TFIN) GO TO 30
      GO TO 45
C*
C*    STATE EVENT HAS PASSED; REDUCE THE STEP  SIZE
C*
380   IF(DTFUL-DTMIN) 381,381,382
381   IF (IERR) 450,401,401
401   IF (ISEES+1000)  410,400,400
400   I=-ISEES
      WRITE(NPRTR,980) I,TNOW
      GC TO 420
410   I=-ISEES-1000
      WRITE(NPRTR,990)  I,TNOW
420   IF(IERR.EQ.0) GO TO 450
      CALL ERROR (17,1)
382   DTFUL=FRMUL*DTFUL
      IF(DTFUL.LT.DTMIN) DTFUL=DTMIN
      ISEES=0
      GO TO 240
C*
C*    AT LEAST ONE STATE EVENT HAS OCCURED; SCHEDULE THEM ON
C*    THE CALENDER ; ACCEPT THE SS AND  DD VALUES; START A NEW STEP
C*
450   ISEES=0
      DC 451 I=1,NSEV
      IF(ISCD(I).EQ.0.OR.ISCD(I)*ICDIR(I).LT.0) GO TO 453
      IX=ICSV(I)
      CALL SCHD(IX,TNOW,ATRIB)
453   ISCD(I)=0
451   CCNTINUE
      TLAST=TNOW
      DC 452 I=1,NEQT
          SSL(I)=SS(I)
          DDL(I)=DD(I)
452   CONTINUE
      GC TO 30
C*
```

```
          1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234

C*    RUNGE-KUTA INTEGRATICN
C*
240   DTNOW=.25*DTFUL
      DC 250 I=1,NEQD
250   SS(I)=SSL(I)+DTNOW*DDL(I)
      TNOW=TLAST+DTNOW
      CALL STATE
      CH=3.0*DTFUL/32.0
      DO 260 I=1,NEQD
      A2(I)=DD(I)
260   SS(I)=SSL(I)+CH*(DDL(I)+3.0*A2(I))
      DTNOW=.375*DTFUL
      TNOW=TLAST+DTNOW
      CALL STATE
      CH=DTFUL/2197.0
      DO 270 I=1,NEQD
      A3(I)=DD(I)
270   SS(I)=SSL(I)+CH*(1932.0*DDL(I)+(7296.0*A3(I)-7200.0*A2(I)))
      DTNOW=12.0*DTFUI/13.0
      TNOW=TLAST+DTNOW
      CALL STATE
      CH=DTFUL/4104.
      DO 280 I=1,NEQD
      A4(I)=DD(I)
280   SS(I)=SSL(I)+CH*((8341.0*DDL(I)-845.0*A4(I))+(29440.0*A3(I)
     --32832.0*A2(I)))
      DTNOW=DTFUL
      TNOW=TLAST+DTNOW
      CALL STATE
      CH=DTFUL/20520.0
      DC 290 I=1,NEQD
      A5(I)=DD(I)
290   SS(I)=SSL(I)+CH*((-6080.*DDL(I)+(9295.0*A4(I)-5643.0*A5(I)))
     -+(41040.0*A2(I)-28352.0*A3(I)))
      DTNOW=.5*DTFUL
      TNOW=TLAST+DTNOW
      CALL STATE
      CH=DTFUL/7618050.0
      EEOET=0.0
      DO 300 I=1,NEQD
      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)))
      IF(DTFUL.LT.DTMIN) GO TO 300
      TERR=ABS(SSL(I))+ABS(SS(I))+AEM
      IF (TERR.LE.0.0) GO TO 300
      EERR=ABS((-2090.0*DDL(I)+(21970.0*A4(I)-15048.0*A5(I)))+
     -(22528.0*A3(I)-27360.0*DD(I)))
      IF (EEOET.GE.EERR/TERR) GO TO 300
      EEOET=EERR/TERR
      IIR=I
300   CONTINUE
      IF (DTFUL.LT.DTMIN) GO TO 85
      ESTOL=DTFUL*EEOET*SCALL/752400.0
      IF(ESTOL.LE.1.0) GO TO 310
C*
```

```
              1         2         3         4         5         6         7
     12345678901234567890123456789012345678901234567890123456789012345678901234

     C*    ACCURACY IS NOT MET; MUST REDUCE THE STEP SIZE
     C*
           IF (DTFUL.LE.DTMIN) GO TO 330
           FRACI=.1
           IF (ESTOL.LT.59049.0) FRACI=TMUL/ESTOL**.2
           DTACC=FRACI*DTFUL
           IF (DTACC.LT.DTMIN) DTACC=DTMIN
           DTFUL=DTACC
           GO TO 240
     C*
     C*    ACCURACY IS ACCPTABLE; INCREASE THE STEP SIZE
     C*
     310   FRACI=5.0
           IF (ESTOL.GT.1.889568E-4) FRACI=TMUL/ESTOL**.2
           DTACC=FRACI*DTFUL
           IF (DTACC.GT.DTMAX) DTACC=DTMAX
           IF (DTACC.LT.DTMIN) DTACC=DTMIN
           GO TO 85
     330   IF(IERR) 90,350,340
     340   WRITE(NPRTR,970) ILR,TNOW
           CALL ERROR(18,1)
     350   WRITE(NPRTR,970) ILR,TNOW
           GO TO 85
     C*
     C*    RUN IS COMPLETED
     C*
     1000  IF(NDTPST.EQ.0) GO TO 1003
     1001  DO 1002 I=1,NDTPST
              LDBL=LFDSB+(I-1)*9
              CALL DCLCT(LDBL)
     1002  CONTINUE
     1003  IF(NCTPST.EQ.0) RETURN
           CALL SSAVE
           RETURN
     960   FORMAT (///36X,6HTNEXT=,E17.9,5X,5HTNOW=,E17.9)
     970   FORMAT (/2X,38HSPECIFIED LOCAL ERROR EXCEEDED FOR SS(,I3,
          -9H) AT TIME,E12.4)
     980   FORMAT (/2X,38HSPECIFIED  TOLERANCE  EXCEEDED FOR SS(,I3,
          -9H) AT TIME,E12.4)
     990   FORMAT (/2X,38HSPECIFIED  TOLERANCE  EXCEEDED FOR DD(,I3,
          -9H) AT TIME,E12.4)
           END
           SUBROUTINE DISC
           DIMENSION RSET(1)
           COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTP,LFAE,LCAL
           COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
          -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
           COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
          -DDL(99),TNOW, XX(99),DTNOW,ISTOP
           COMMON ISET(1)
           EQUIVALENCE(ISET(1),RSET(1))
     C*
     C*    TAKE THE FIRST ENTRY OF THE CALENDER
     C*
           J=LCAL+1
```

```
                 1         2         3         4         5         6         7
        1234567890123456789012345678901234567890123456789012345678901234567890123
              LENT=ISET(J)
              L=LCAL
              CALL ULINK(L,LENT)
        C*
        C*    TAKE ITS EVENT CODE AND PROCESS IT
        C*
              J=LENT+2
              ICODE=ISET(J)
              IF(ICODE.NE.-1) GO TO 11
              CALL CLEARS
              RETURN
        C*
        C*    COPY THE ATRIBUTES INTO ATRIB ARRAY
        C*
        11    IF(NATR.EQ.0) GO TO 16
              J=LENT+3
              LATR=ISET(J)
              DO 15 K=1,NATR
                 J=LATR+K
                 ATRIB(K)=RSET(J)
        15    CONTINUE
        16    CALL EVENT(ICODE)
        C*
        C*    FOR ALL THE XX DTPST SEE IF THEIR VALUE IS CHANGED
        C*    IF SO COLLECT STATISTICS
        C*
              IF(NTMX.EQ.0) RETURN
              DO 10 K=1,NTMX
                  L=LTMX(K)
                  J=L+7
                  I=-ISET(J)
                  J=L+8
                  NU=ISET(J)
                  LSTAT=ISET(L)
                  J=LSTAT+2
                  IF(XX(I).EQ.RSET(J)) RETURN
                  X=XX(I)
                  CALL DTPST(X,LSTAT,NU)
        10    CONTINUE
              RETURN
              END
              SUBROUTINE SSCND
              COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
             -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
             -ISCD(25)
              DO 10 J=1,NSEV
                 ISCD(J)=KKRSS(ICGV(J),ICDV(J),VAL(J),ICDIR(J),TTOL(J))
        10    CONTINUE
              RETURN
              END
              FUNCTION KKRSS(IKRSG,IKRSD,CADD,LDIR,TCL)
              COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
             -DDL(99),TNOW, XX(99),DTNOW,ISTCP
              COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
             -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
```

```
                  1         2         3         4         5         6         7
         1234567890123456789012345678901234567890123456789012345678901234567890

               -ISCD(25)
                DIMENSION IKRSG(1),IKRSD(1),CADD(1),LDIR(1),TOL(1)
                JKRSG=IKRSG(1)
                JKRSD=IKRSD(1)
                IF (JKRSG) 10,260,20
           10   JKRSG=-JKRSG
                CRSGL=DDL(JKRSG)
                CRSGN=DD(JKRSG)
                GC TO 30
           20   CRSGL=SSL(JKRSG)
                CRSGN=SS(JKRSG)
           30   IF (JKRSD) 40,50,60
           40   JKRSD=-JKRSD
                CRSDL=DDL(JKRSD)
                CRSDN=DD(JKRSD)
                GC TO 70
           50   CRSDL=CADD(1)
                CRSDN=CADD(1)
                GC TO 70
           60   CRSDL=SSL(JKRSD)
                CRSDN=SS(JKRSD)
           70   IF (CRSGL-CRSDL) 80,260,90
           80   IF (CRSGN-CRSDN) 260,100,100
           90   IF (CRSGN-CRSDN) 130,130,260
          100   IF (LDIR(1))  260,110,110
          110   IF (CRSGN-CRSDN-TOL(1)) 120,120,200
          120    KKRSS=1
                GO TO 160
          130   IF (LDIR(1)) 140,140,260
          140   IF (CRSGN-CRSDN+TOL(1)) 210,150,150
          150   KKRSS=-1
          160   IF (ISEES) 270,170,270
          170   IF (IKRSG(1)) 180,270,190
          180   ISZES=JKRSG+1000
                GO TO 270
          190   ISEES=JKRSG
                GC TO 270
          200   KKRSS=2
                GC TO 220
          210   KKRSS=-2
          220   IF (ISEES) 270,230,230
          230   IF (IKRSG(1)) 240,270,250
          240   ISEES=-JKRSG-1000
                GO TO 270
          250   ISEES=-JKRSG
                GO TO 270
          260   KKRSS=0
          270   RETURN
                END
                SUBROUTINE SSAVE
                DIMENSION RSET(1)
                COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,ICAL
                COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
               -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
                COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
```

```
              1         2         3         4         5         6         7
     1234567890123456789012345678901234567890123456789012345678901234567890123
           -DCL(99),TNOW, XX(99),DTNOW,ISTOP
            COMMON ISET(1)
            EQUIVALENCE(ISET(1),RSET(1))
            DO 10 N=1,NCTPST
              LCBL=LFCSB+(N-1)*9
              LSTAT=ISET(LCBL)
              ITYPE=ISET(LCBL+7)
              IF(ITYPE.LT.0) GO TO 15
              X=SS(ITYPE)
              GO TO 16
     15       X=DD(-ITYPE)
     16       NU=ISET(LCBL+8)
              CALL CTPST(X,LSTAT,NU)
     10     CONTINUE
            RETURN
            END
            SUBROUTINE DATAIN
     C*
     C*     THIS SUBROUTINE READS THE INPUT CARDS AND SETS
     C*     UP THE 'ENVIRONMENT' FOR THE MODEL. FOR THE TIMEBEING
     C*     THE CARDS HAVE FIXED FORMAT AND ORDER.
     C*
            DIMENSION KARD(13),RSET(1),NAME(6),INFO(4)
            DATA INFO /1HN,1HW,1H ,1HF/
            DATA KARD /4HPROJ,4HDISC,4HRANK,4HCONT,4HSEVN,4HSTAT,4HTALY,
           -4HDTPS,4HCTPS,4HCOUN,4HSTRM,4HSIMU,4HINTL/
            COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
            COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
           -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
            COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
           -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
           -ISCD(25)
            COMMON/GEN3/ITITLE(15),IANAL(10),IDATE(5)
            COMMON/GEN4/ISEED(10),LSEED(10)
            COMMON/GEN8/NIR,VALH,VALL
            COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
           -DCL(99),TNOW, XX(99),DTNOW,ISTOP
            COMMON ISET(1)
            EQUIVALENCE(ISET(1),RSET(1))
     C*
     C*     READ THE PROJECT CARD
     C*
            READ(NCRDR,1001) LABEL,(ITITLE(I),I=1,15),(IANAL(I),I=1,10),
           -(IDATE(I),I=1,5)
            WRITE(NPRTR,2001) LABEL,(ITITLE(I),I=1,15),(IANAL(I),I=1,10),
           -(IDATE(I),I=1,5)
            IF(LABEL.NE.KARD(1)) GO TO 100
     C*
     C*     READ THE DISCRETE CARD
     C*
            READ(NCRDR,1002) LABEL,NFILE,NENT,NATR
            WRITE(NPRTR,2002)LABEL,NFILE,NENT,NATR
            IF(LABEL.NE.KARD(2)) GO TO 101
            LFFB=LFI
            IF(NFILE.EQ.0.AND.NENT.EQ.0) GO TO 19
```

```
                1         2         3         4         5         6         7
       1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234

                IF(NENT.EQ.0) GO TO 117
                IF(NFILE.EQ.0) GO TO 11
                DO 10 I=1,NFILE
       C*
       C*       READ RANK CARDS
       C*
                    READ(NCRDR,1003)LABEL,IFL,IRANK,NRAT
                    WRITE(NPRTR,2003)LABEL,IFL,IRANK,NRAT
                    IF(LABEL.NE.KARD(3)) GO TO 102
       C*
       C*       SET THE FILE BLOCK
       C*
                    CALL SETFB(IRANK,NRAT)
       10       CONTINUE
       C*
       C*       SET UP THE CALANDER AND THE ENTRY POOL
       C*
        11      CALL SETCAL
       C*
       C*       READ THE CONTINUOUS CARD
       C*
       19       READ(NCRDR,1004) LABEL,NEQD,NEQS,NSEV,DTMIN,DTMAX,AERR,RERR,IER
                WRITE(NPRTR,2004)LABEL,NEQD,NEQS,NSEV,DTMIN,DTMAX,AERR,RERR,IER
                IF(LABEL.NE.KARD(4)) GO TO 103
                IF(NEQD.LT.0.OR.NEQS.LT.0) GO TO 188
                NEQT=NEQD+NEQS
                IF(NEQT.EQ.0.AND.NENT.EQ.0) GO TO 189
                IF(NEQT.EQ.0) GO TO 29
                IERR=2
                IF(IER.EQ.INFO(1)) IERR=-1
                IF(IER.EQ.INFO(2).OR.IER.EQ.INFO(3))  IERR=0
                IF(IER.EQ.INFO(4)) IERR=1
                IF(IERR.EQ.2) GO TO 122
       C*
       C*       READ STATE EVENT CARDS
       C*
                IF(NSEV.LT.0) GO TO 104
                IF(DTMIN.LT.0.OR.DTMAX.LT.0.OR.AERR.LT.0.OR.IERR.LT.0) GO TO 123
                IF(NSEV.EQ.0) GO TO 29
                DO 20 I=1,NSEV
                    READ(NCRDR,1005) LABEL,ICSV(I),ICGV(I),ICDIR(I),ICDV(I),
               -    VAL(I),TTOL(I)
                    WRITE(NPRTR,2005)LABEL,ICSV(I),ICGV(I),ICDIR(I),ICDV(I),
               -    VAL(I),TTOL(I)
                    IF(LABEL.NE.KARD(5)) GO TO 104
                    IF(ICSV(I).EQ.0) GO TO 190
                    R=ICGV(I)
                    IF(ABS(R).GT.25.OR.ICGV(I).EQ.0) GO TO 190
                    IF(ICDIR(I).EQ.1.OR.ICDIR(I).EQ.0.OR.ICDIR(I).EQ.-1) GO TO 22
                    GO TO 191
       22           IF(ICDV(I).EQ.0) GO TO 20
                    R=ICDV(I)
                    IF(ABS(R).GT.25) GO TO 192
       20       CONTINUE
       C*
```

```
                 1         2         3         4         5         6         7
        1234567890123456789012345678901234567890123456789012345678901234567890123456

      C*    READ STAT CARD
      C*
       29   READ(NCRDR,1006) LABEL,NTAL,NDTPST,NCTPST,NCOUN,DTSAV
            WRITE(NPRTR,2006) LABEL,NTAL,NDTPST,NCTPST,NCOUN,DTSAV
            IF(LABEL.NE.KARD(6)) GO TO 105
      C*
      C*    READ TALY CARDS
      C*
            IF(NTAL.LT.0.OR.NDTPST.LT.0.OR.NCTPST.LT.0.OR.NCOUN.LT.0)GO TO 124
            IF(NTAL.EQ.0) GO TO 39
            LFTB=LFI
            DO 30 I=1,NTAL
                READ(NCRDR,1007)LABEL,N,(NAME(J),J=1,6),NU
                WRITE(NPRTR,2007)LABEL,N,(NAME(J),J=1,6),NU
                IF(LABEL.NE.KARD(7)) GO TO 106
                CALL SETTAL(NAME,NU)
       30   CONTINUE
      C*
      C*    READ DTPS CARDS
      C*
       39   IF(NDTPST.EQ.0) GO TO 49
            LFDSB=LFI
            DO 40 I=1,NDTPST
                READ(NCRDR,1008)LABEL,N,(NAME(J),J=1,6),ITYPE,NU
                WRITE(NPRTR,2008)LABEL,N,(NAME(J),J=1,6),ITYPE,NU
                IF(LABEL.NE.KARD(8)) GO TO 107
                CALL SETDS (NAME,ITYPE,NU)
       40   CONTINUE
      C*
      C*    READ CTPS CARDS
      C*
       49   IF(NCTPST.EQ.0) GO TO 59
            LFCSB=LFI
            DO 50 I=1,NCTPST
                READ(NCRDR,1008) LABEL,N,(NAME(J),J=1,6) ,ITYPE,NU
                WRITE(NPRTR,2008)LABEL,N,(NAME(J),J=1,6),ITYPE,NU
                IF(LABEL.NE.KARD(9)) GO TO 108
                CALL SETCS (NAME,ITYPE,NU)
       50   CONTINUE
      C*
      C*    READ COUN CARDS
      C*
       59   IF(NCOUN.EQ.0) GO TO 69
            LFCTB=LFI
            DO 60 I=1,NCOUN
                READ(NCRDR,1009)LABEL,N,(NAME(J),J=1,6) ,LIMIT
                WRITE(NPRTR,2009 LABEL,N,(NAME(J),J=1,6) ,LIMIT
                IF(LABEL.NE.KARD(10)) GO TO 109
                CALL SETCT(NAME,LIMIT)
       60   CONTINUE
       69   IF(NENT.EQ.1) CALL SETENP
      C*
      C*
      C*    READ STRM CARDS
      C*
```

```
                 1         2         3         4         5         6         7
        1234567890123456789012345678901234567890123456789012345678901234567890123456

              DC 70 I=1,10
                 READ(NCRDR,1010) LABEL,IR,IS
                 IF(LABEL.NE.KARD(11))  GO TO 71
                 WRITE(NPRTR,2010) LABEL,IR,IS
                 IF(IR.LT.1.OR.IR.GT.10) CALL ERROR(13,IR)
                 LSEED(IR)=IS
                 IF(IS.LT.0) IS=-IS
                 ISEED(IR)=IS
        70    CONTINUE
              GO TO 80
        71    WRITE(NPRTR,2011)LABEL,IR
              IF(LABEL.NE.KARD(12)) GO TO 111
              NRUNS=IR
              GO TO 81
        C*
        C*
        C*    READ SIMU CARD
        C*
        80    READ(NCRDR,1011)LABEL,NRUNS
              WRITE(NPRTR,2011)LABEL,NRUNS
              IF(LABEL.NE.KARD(12)) GO TO 111
        C*
        C*    READ INTL CARD
        C*
        81    READ(NCRDR,1012)LABEL,TBEG,TFIN,IRAN,IECO,ISUM,ICLF,ICLV,ICLS,
             -TCLEAR
              WRITE(NPRTR,2012)LABEL,TBEG,TFIN,IRAN,IECO,ISUM,ICLF,ICLV,ICLS,
             -TCLEAR
              IF(LABEL.NE.KARD(13)) GO TO 110
              IF(IRAN.NE.1.AND.IRAN.NE.0) GO TO 121
              IF(IECO.NE.1.AND.IECO.NE.0) GO TO 119
              IF(ISUM.NE.1.AND.ISUM.NE.0) GO TO 120
              IF(TFIN.LE.TBEG) TFIN=VALH
              IF(ICLF.NE.1.AND.ICLF.NE.0) GO TO 113
              IF(ICLV.NE.1.AND.ICLV.NE.0) GO TO 118
              IF(ICLS.NE.1.AND.ICLS.NE.0) GO TO 114
              IF(ICLS.EQ.1)RETURN
              IF(TCLEAR.GT.TFIN.OR.TCLEAR.LT.TBEG) GO TO 115
              RETURN
        1001  FORMAT(A4,5X,15A2,10A2,5A2)
        1002  FORMAT(A4,5X,I2,8X,I1,9X,I2)
        1003  FORMAT(A4,5X,I2,8X,I1,9X,I2)
        1004  FORMAT(A4,5X,I2,3X,I2,3X,I2,8X,4F10.5,A1)
        1005  FORMAT(A4,5X,I2,8X,I3,7X,I2,8X,I3,7X,2F10.5)
        1006  FORMAT(A4,5X,4(I2,8X),F10.5)
        1007  FORMAT(A4,5X,I2,8X,6A2,8X,I2)
        1008  FORMAT(A4,5X,I2,8X,6A2,8X,I3,7X,I2)
        1009  FORMAT(A4,5X,I2,8X,6A2,8X,I6)
        1010  FORMAT(A4,5X,I2,8X,I11)
        1011  FORMAT(A4,5X,I2)
        1012  FORMAT(A4,5X,2F10.3,5X,6(I1,4X),5X,F10.3)
        2001  FORMAT(10X,A4,5X,15A2,10A2,5A2)
        2002  FORMAT(10X,A4,5X,I2,8X,I1,9X,I2)
        2003  FORMAT(10X,A4,5X,I2,8X,I1,9X,I2)
        2004  FORMAT(10X,A4,5X,I2,3X,I2,3X,I2,8X,4F10.5,A1)
```

```
                 1         2         3         4         5         6         7
        1234567890123456789012345678901234567890123456789012345678901234567890123456789012345
      2005  FORMAT(10X,A4,5X,I2,8X,I3,7X,I2,8X,I3,7X,2F10.5)
      2006  FORMAT(10X,A4,5X,4(I2,8X),F10.5)
      2007  FORMAT(10X,A4,5X,I2,8X,6A2,8X,I2)
      2008  FORMAT(10X,A4,5X,I2,8X,6A2,8X,I3,7X,I2)
      2009  FORMAT(10X,A4,5X,I2,8X,6A2,8X,I6)
      2010  FORMAT(10X,A4,5X,I2,8X,I11)
      2011  FORMAT(10X,A4,5X,I2)
      2012  FORMAT(10X,A4,5X,2F10.3,5X,6(I1,4X),5X,F10.3)
      100   WRITE(NPRTR,200)
      200   FORMAT(10X,50H*****ERROR: THE PORJ CARD IS MISSING                 )
            GO TO 1000
      101   WRITE(NPRTR,201)
      201   FORMAT(10X,50H*****ERROR: THE DISC CARD IS MISSING                 )
            GO TO 1000
      102   WRITE(NPRTR,202)
      202   FORMAT(10X,50H*****ERROR: NOT ENOUGH RANK CARDS                    )
            GO TO 1000
      103   WRITE(NPRTR,203)
      203   FORMAT(10X,50H*****ERROR: THE CONT CARD IS MISSING                 )
            GO TO 1000
      104   WRITE(NPRTR,204)
      204   FORMAT(10X,50H*****ERROR: NOT ENOUGH SEVN CARD  OR NSEV.LT.0       )
            GO TO 1000
      188   WRITE(NPRTR,288)
      288   FORMAT(10X,34H*****ERROR: NEQD.LT.0.OR.NEQS.LT.0)
            GO TO 1000
      189   WRITE(NPRTR,289)
      289   FORMAT(10X,38H*****ERROR: THIS IS NEITHER CONTINUOUS,
           -20H NOR DISCRETE MODEL )
            GO TO 1000
      190   WRITE(NPRTR,290)
      290   FORMAT(10X,50H*****ERROR: IMPROPER VALUE FOR THE INDEX OF          ,
           -22X,20HCROSSING VARIABLE      )
            GO TO 1000
      191   WRITE(NPRTR,291)
      291   FORMAT(10X,50H*****ERROR: IMPROPER DIRECTION                       )
            GO TO 1000
      192   WRITE(NPRTR,292)
      292   FORMAT(10X,50H*****ERROR: IMPROPER CROSSED VARIABLE                )
            GO TO 1000
      105   WRITE(NPRTR,205)
      205   FORMAT(10X,50H*****ERROR: STAT CARD IS MISSING                     )
            GO TO 1000
      106   WRITE(NPRTR,206)
      206   FORMAT(10X,50H*****ERROR: NOT ENOUGH TALY CARDS                    )
            GO TO 1000
      107   WRITE(NPRTR,207)
      207   FORMAT(10X,50H*****ERROR: NOT ENOUGH DTPS CARDS                    )
            GO TO 1000
      108   WRITE(NPRTR,208)
      208   FORMAT(10X,50H*****ERROR: NOT ENOUGH CTPS CARDS                    )
            GO TO 1000
      109   WRITE(NPRTR,209)
      209   FORMAT(10X,50H*****ERROR: NOT ENOUGH COUN CARDS                    )
            GO TO 1000
```

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012345

  110   WRITE(NPRTR,210)
  210   FORMAT(10X,50H*****ERROR: INTL CARD IS MISSING                    )
        GO TO 1000
  111   WRITE(NPRTR,211)
  211   FORMAT(10X,50H*****ERROR: SIMU CARD IS MISSING                    )
        GO TO 1000
  113   WRITE(NPRTR,213)
  213   FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR CLEARING THE           ,
       -22X,5HFILES)
        GO TO 1000
  114   WRITE(NPRTR,214)
  214   FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR CLEARING THE           ,
       -22X,20HSTATISTICAL ARRAYS      )
        GO TO 1000
  115   WRITE(NPRTR,215)
  215   FORMAT(10X,50H*****ERROR: TCLEAR.GT.TFIN .OR.TCLEAR.LT.TBEG       )
        GO TO 1000
  117   WRITE(NPRTR,217)
  217   FORMAT(10X,50H*****ERROR: DISCRETE EVENT BUT NO ENTITY            )
        GO TO 1000
  118   WRITE(NPRTR,218)
  218   FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR CLEARING VARIABLES     )
        GO TO 1000
  119   WRITE(NPRTR,219)
  219   FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR DELETING ECHO REPORT )
        GO TO 1000
  120   WRITE(NPRTR,220)
  220   FORMAT(10X,50H*****ERROR: ILLEGAL CODE FOR DELETING SUMMARY REPO
      -,2HRT)
        GO TO 1000
  121   WRITE(NPRTR,221)
  221   FORMAT(10X,49H*****ERROR: ILLEGAL CODE FOR INITIALIZING RANDOM
       - ,24HNO. STREAMS BETWEEN RUNS)
        GO TO 1000
  122   WRITE(NPRTR,222)
  222   FORMAT(10X,44H*****ERROR: ILLEGAL CODE FOR ERROR INDICATOR)
        GO TO 1000
  123   WRITE(NPRTR,223)
  223   FORMAT(10X,50H*****ERROR: DTMIN.LT.0.OR.DTMAX.LT.0.OR.AERR.LT.0.,
       -12HOR.RERR.LT.0)
        GO TO 1000
  124   WRITE(NPRTR,224)
  224   FORMAT(10X,50H,*****ERROR:NTAL.LT.0.OR.NDTPST.LT.0.OR.NCTPST.LT0,
       -14H.OR.NCOUN.LT.0)
  1000  I=SQRT(-1.)
        STOP
        END
        SUBROUTINE SETCAL
        DIMENSION RSET(1)
        COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,ICAL
        COMMON/GEN8/NIR,VALH,VALI
        COMMON ISET(1)
        EQUIVALENCE (ISET(1),RSET(1))
        II=LFI+6
        I=(II+NIR-1)/NIR
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123

C*
C*    ASSIGN THE RANKING RULE
C*
      IF(I.GT.LLR) CALL ERROR (1,1)
      LCAL=LFI
      LFI=II
C*
C*    ASSIGN THE RANKING RULE
C*
      J=LCAL+3
      ISET(J)=4
      J=LCAL+4
      ISET(J)=1
      RETURN
      END
      SUBROUTINE SETFB(IRANK,NRAT)
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
      COMMON/GEN8/NIR,VALH,VALL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      II=LFI+6
      I=(II+NIR-1)/NIR
      IF(I.GT.LLR) CALL ERROR (1,1)
      IF(IRANK.LT.1.OR.IRANK.GT.4) CALL ERROR (3,1)
      I=LFI
      J=I+3
      ISET(J)=IRANK
      IF(IRANK.LT.3) GO TO 1
      IF(NRAT.GT.NATR.OR.NRAT.LT.1) CALL ERROR (2,1)
      J=I+4
      ISET(J)=NRAT+1
1     LFI=I+6
      RETURN
      END
      SUBROUTINE SETENP
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
      COMMON/GEN8/NIR,VALH,VALL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      KK=0
      I=LFI+4
      J=LLR-(NATR+1)
      II=(I+NIR-1)/NIR
      IF(II.GT.J) CALL ERROR (1,1)
      LFAE=LFI
10    KK=KK+1
      L=LFI+1
      ISET(L)=I
      L=LFI+3
      ISET(L)=J
      LFI=I
      LLR=J
```

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--ETC(U)
JUN 81 W E BILES, A NOZARI N00014-79-C-0757

UNCLASSIFIED NL

2 OF 2
AD
A101063

END
DATE
FILMED
7-81
DTIC

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234

      I=LFI+4
      J=LLR-(NATR+1)
      II=(I+NIR-1)/NIR
      IF(II.LE.J) GO TO 10
      I=I-3
      ISET(I)=0
      NENT=KK
      RETURN
      END
      SUBROUTINE SETTAL(NAME,IU)
      DIMENSION NAME(6)
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN8/NIR,VALH,VALL
      COMMON ISET(1)
      EQUIVALENCE(ISET(1),RSET(1))
      I=LFI+8
      J=LLR-5
      II=(I+NIR-1)/NIR
      IF(II.GT.J) CALL ERROR (1,1)
      ISET(LFI)=J
      II=LFI+1
      KK=LFI+6
      DO 10 L=II,KK
          LL=L-II+1
          ISET(L)=NAME(LL)
   10 CONTINUE
      II=LFI+7
      ISET(II)=IU
      II=J+3
      RSET(II)=VALH
      II=J+4
      RSET(II)=VALL
      LFI=I
      LLR=J
      RETURN
      END
      SUBROUTINE SETCS (NAME,ITYPE,NU)
      DIMENSION NAME(6)
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
      COMMON/GEN8/NIR,VALH,VALL
      COMMON ISET(1)
      EQUIVALENCE(ISET(1),RSET(1))
      I=LFI+9
      J=LLR-7
      II=(I+NIR-1)/NIR
      IF(II.GT.J) CALL ERROR (1,1)
      ISET(LFI)=J
      II=LFI+1
      KK=LFI+6
      DO 10 L=II,KK
          LL=L-II+1
```

```
               1         2         3         4         5         6         7
      1234567890123456789012345678901234567890123456789012345678901234567890123
                ISET(L)=NAME(LL)
   10     CONTINUE
          R=ITYPE
          IF(ITYPE.EQ.0.OR.ABS(R).GT.99.) CALL ERROR (19,ITYPE)
          II=LFI+7
          ISET(II)=ITYPE
          II=LFI+8
          ISET(II)=NU
          II=J+3
          RSET(II)=VALH
          II=J+4
          RSET(II)=VALL
          LFI=I
          LLR=J
          RETURN
          END
          SUBROUTINE SETDS (NAME,ITYPE,NU)
          DIMENSION NAME(6)
          DIMENSION RSET(1)
          COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSP,LFCTB,LFAE,LCAL
          COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
         -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
          COMMON/GEN8/NIR,VALH,VALL
          COMMON ISET(1)
          EQUIVALENCE(ISET(1),RSET(1))
          I=LFI+9
          J=LLR-7
          II=(I+NIR-1)/NIR
          IF(II.GT.J) CALL ERROR (1,1)
          ISET(LFI)=J
          II=LFI+1
          KK=LFI+6
          DO 10 L=II,KK
                LL=L-II+1
                ISET(L)=NAME(LL)
   10     CONTINUE
          IF(ITYPE.GT.0.AND.ITYPE.LE.NFILE) GO TO 11
          IF(ITYPE.EQ.0.OR.ITYPE.LT.-99.OR.ITYPE.GT.NFILE)
         -CALL ERROR (19,ITYPE)
          NTMX=NTMX+1
          LTMX(NTMX)=LFI
          GO TO 12
   11     LFBL=LFFB+(ITYPE-1)*6
          II=LFBL+5
          ISET(II)=LFI
   12     II=LFI+7
          ISET(II)=ITYPE
          II=LFI+8
          ISET(II)=NU
          II=J+3
          RSET(II)=VALH
          II=J+4
          RSET(II)=VALL
          LFI=I
          LLR=J
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123 4
      RETURN
      END
      SUBROUTINE ERROF (ICODE,I)
      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
      IS=0
      IF(ICODE.LT.1000) GO TO 100
      ICODE=1000-ICODE
      IS=1
  100 GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,
     -114,115,116,117,118), ICODE
  101 WRITE(NPRTR,201)
  201 FORMAT(10X,37H*****ERROR: THERE IS NOT ENOUGH SPACE)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
  102 WRITE(NPRTR,202)
  202 FORMAT(10X,44H*****ERROR: RANKING ATRIBUTE IS OUT OF RANGE)
      GO TO 999
  103 WRITE(NPRTR,203)
  203 FORMAT(10X,33H*****ERROR: RANKING CODE IS WRONG)
      GO TO 999
  104 WRITE(NPRTR,204) I
  204 FORMAT(10X,44H*****ERROR: THE ENITY THAT IS TO BE REMOVED,
     -,27H COPIED, OR LOCATED IN FILE,I4,15H DOES NOT EXIST)
      GO TO 998
  105 WRITE(NPRTR,205) I
  205 FORMAT(10X,29H*****ERROR: THERE IS NO CTPST,I4)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
  106 WRITE(NPRTR,206) I
  206 FORMAT(10X,29H*****ERROR: THERE IS NO DTPST,I4)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
  107 WRITE(NPRTR,207) I
  207 FORMAT(10X,29H*****ERROR: THERE IS NO TALLY,I4)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
  108 WRITE(NPRTR,208)
  208 FORMAT(10X,36H*****ERROR: NO CONTINUOUS NO DICRETE)
      GO TO 999
  109 WRITE(NPRTR,209)
  209 FORMAT(10X,36H*****ERROR: TNEXT<TNOW                         )
      GO TO 998
  110 WRITE(NPRTR,210) I
  210 FORMAT(10X,29H*****ERROR: THERE IS NO FILE ,I3)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
  111 WRITE(NPRTR,211) I
  211 FORMAT(10X,33H,*****ERROR: THERE IS NO COUNTER ,I4)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
  112 WRITE(NPRTR,212) I
  212 FORMAT(10X,33H,*****ERROR: THERE IS NO ATRIBUTE,I4)
      IF(IS.EQ.1) GO TO 998
      GO TO 999
```

```
                  1         2         3         4         5         6         7
         1234567890123456789012345678901234567890123456789012345678901234567890123

     113  WRITE(NPRTR,213) I
     213  FORMAT(10X,38H*****ERROR: STREAM NO. IS OUT OF RANGE,I4)
          IF (IS.EQ.1) GO TO 998
          GO TO 999
     114  WRITE(NPRTR,214)
     214  FORMAT(10X,25H*****ERROR: DRAND IS ZERO)
          GO TO 998
     115  WRITE(NPRTR,215)
     215  FORMAT(10X,45H*****ERROR: INCORRECT PARAMETER SPECIFICATION)
          GO TO 998
     116  WRITE(NPRTR,216)
     216  FORMAT(10X,45H*****ERROR: INCORRECT CUMULATIVE PROBABILITY ,
         -13HSPECIFICATION)
          GO TO 998
     117  WRITE(NPRTR,217)
     217  FORMAT(10X,45H*****ERROR: FATAL INTEGRATION TOLERANCE ERROR)
          GO TO 998
     118  WRITE(NPRTR,218)
     218  FORMAT(10X,45H*****ERROR: FATAL INTEGRATION ACCURACY  ERROR)
          GO TO 998
     119  WRITE(NPRTR,219) ITYPE
     219  FORMAT(10X,11H*****ERROR:,I4,22H IS OUT OF RANGE INDEX  )
          GO TO 999
     998  CALL SUMRY(0)
     999  I=SQRT(-1.)
          STOP
          END
          SUBROUTINE SETCT(NAME,LIMIT)
          DIMENSION NAME(6)
          DIMENSION RSET(1)
          COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
          COMMON/GEN8/NIR,VALH,VALL
          COMMON ISET(1)
          EQUIVALENCE(ISET(1),RSET(1))
          I=LFI+8
          II=(I+NIR-1)/NIR
          IF(II.GT.LLR) CALL ERROR (1,1)
          II=LFI+1
          KK=LFI+6
          DO 10 L=II,KK
              LL=L-II+1
              ISET(L)=NAME(LL)
     10   CONTINUE
          II=LFI+7
          ISET(II)=LIMIT
          LFI=I
          RETURN
          END
          SUBROUTINE FILEM(IFILE,A)
          DIMENSION A(99)
          DIMENSION RSET(1)
          COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
          COMMON/GEN1/NRUNS,NFILE,IBEG,TFIN,ICLP,ICLV,ICLS,TCLEAR,IRAN,
         -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
          COMMON/GEN5/IEVENT,TIME
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
C*
C*    TAKE ONE ENTRY FROM THE PCOL
C*
      IF(LFAE.EQ.0) CALL ERROR (1001,1)
      LENT=LFAE
      J=LENT +1
      LS=ISET(J)
      LFAE=LS
C*
C*    FIND THE ATTRIBITES
C*
      J=LENT+3
      LATR=ISET(J)
C*
C*    LOAD A INTO THE ATTRIBUTE ZONE
C*
      IF(NATR.EQ.0) GO TO 12
      DO 10 K=1,NATR
          J=LATR+K
          RSET(J)=A(K)
10    CONTINUE
C*
C*    FIND  THE FILE BLOCK; FIND THE RANKING ATTRIBUTE;
C*    COPY IT IN THE LATR POSITION;
C*    IF NRAT IS 1 THEN FILE IS CALANDER AND TIME AND EVENT
C*    MUST BE COPIED
C*
12    IF(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
      LFBL=LFFB+(IFILE-1)*6
      J=LFBL+3
      IRANK=ISET(J)
      GO TO (1,2,3,3) ,IRANK
C*
C*    FIFO
C*
1     L=0
      GO TO 40
C*
C*    LIFO; ADD IT TO THE TOP
C*
2     J=LFBL+1
      L=ISET(J)
      GO TO 40
C*
C*    COPY THE VALUE OF THE RANKING ATTRIBUTE IN ISET(LATR)
C*
3     J=LFBL+4
      NRAT=ISET(J)
      IF(NRAT.EQ.1) GO TO 5
      K=LATR+(NRAT-1)
      RSET(LATR)=RSET(K)
      GO TO 6
5     CONTINUE
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123
C*
C*    THIS IS EVENT CALANDER
C*
          RSET(LATR)=TIME
          J=LENT+2
          ISET(J)=IEVENT
6     CONTINUE
C*
C*    FIND THE POSITION
C*
          J=LFBL+1
          LLFE=ISET(J)
          IF(LLFE.NE.0) GO TO 7
          L=0
          GO TO 40
7         CONTINUE
              L=LLFE
              IF(IRANK.EQ.4) GO TO 4
C*
C*            THIS IS HVF
C*
11            I=L+3
              J=ISET(I)
              IF(RSET(J).LT.RSET(LATR)) GO TO 40
              I=L+1
              L=ISET(I)
              IF(L.EQ.0) GO TO 40
              GO TO 11
C*
C*            THIS IS LVF
C*
4             I=L+3
              J=ISET(I)
              IF(RSET(J).GT.RSET(LATR)) GO TO 40
              I=L+1
              L=ISET(I)
              IF(L.EQ.0) GO TO 40
              GO TO 4
40    CALL LINK(LFBL,L,LENT)
      RETURN
      END
      SUBROUTINE LINK(LFBL,L,LENT)
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
C*
C*    THIS SUBROUTINE LINKS THE ENTITY AT LENT TO THE FILE AT
C*    LFBL, AT ADDRESS L.
C*    IF L=0 THE ENTITY IS LINKED TO THE BOTTOM OF THE FILE
C*    ALSO THEN IT MUST BE CHECKED IF THER IS ANY ENTITY IN
C*    ALREADY IN THE FILE.
C*
      IF(L.NE.0) GO TO 1
      J=LFBL+1
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123

      IF(ISET(J).GT.0) GO TO 2
C*
C*    THERE IS NO ENTITY IN THE FILE
C*
      ISET(J)=LENT
      J=LFBL+2
      ISET(J)=LENT
      ISET(LENT)=0
      J=LENT+1
      ISET(J)=0
      GO TO 4
2     CONTINUE
C*
C*    LINK THE ENTITY TO THE BOTTOM OF THE FILE
C*
      J=LFBL+2
      LLLE=ISET(J)
      ISET(J)=LENT
      ISET(LENT)=LLLE
      J=LENT+1
      ISET(J)=0
      J=LLLE+1
      ISET(J)=LENT
      GO TO 4
1     CONTINUE
C*
C*    ADD IT TO THE LIST BUT SEE IF IT IS TOP OF THE LIST
C*
      J=LFBL+1
      LLFE=ISET(J)
      IF(L.NE.LLFE) GO TO 3
C*
C*    THIS IS TOP OF THE LIST
C*
      ISET(J)=LENT
      ISET(LENT)=0
      J=LENT+1
      ISET(J)=LLFE
      ISET(LLFE)=LENT
      GO TO 4
3     CONTINUE
C*
C*    LINK IT TO THE FILE AND DON'T WORRY ABOUT THE LFE & LLE
C*
      LF=ISET(L)
      ISET(LENT)=LF
      J=LENT+1
      ISET(J)=L
      J=LF+1
      ISET(J)=LENT
      ISET(L)=LENT
4     CONTINUE
C*
C*    COLLECT STATISTICS ON THE FILE, IF NECESSARY
C*
```

```
          1         2         3         4         5         6         7
 1234567890123456789012345678901234567890123456789012345678901234567890123

      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 INTC 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,ICLP,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
      CCMMON 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
        123456789012345678901234567890123456789012345678901234567890123456789012

      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
      COMMON/GEN6/LFBL,L
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      IF(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
      LFBL=LFFB+(IFILE-1)*6
      IF(I.GT.ISET(LFBL).OR.I.LT.1) CALL ERROR(4,IFILE)
      J=ISET(LFBL)/2
      IF(I.GT.J) GO TO 2
C*
C*    START FROM THE TOP
C*
      J=LFBL+1
      LENT=ISET(J)
      K=1
4     CONTINUE
          IF(K.EQ.I)  GO TO 6
          J=LENT+1
          LENT=ISET(J)
          K=K+1
          GO TO 4
2     CONTINUE
C*
C*    START FROM THE BOTTOM
C*
      J=LFBL+2
      LENT=ISET(J)
      K=ISET(LFBL)
5     CONTINUE
          IF(K.EQ.I)  GO TO 6
          LENT=ISET(LENT)
          K=K-1
          GO TO 5
6     L=LENT
      RETURN
      END
      SUBROUTINE ULINK (LFBL,LENT)
C*
C*    THIS SUBROUTINE UNLINKS ENTRY AT LENT OF FILE AT LFBL
C*    AND LINKS IT TO THE ENTRY POOL.
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      LP=ISET(LENT)
      J=LENT+1
      LS=ISET(J)
C*
C*    LINK IT TO THE POOL
C*
      ISET(J)=LFAE
      LFAE=LENT
      ISET(LENT)=0
```

```
              1         2         3         4         5         6         7
     1234567890123456789012345678901234567890123456789012345678901234567890123
           IF(LP.EQ.0) GO TO 10
           IF(LS.EQ.0) GO TO 20
     C*
     C*    UNLINK FROM THE MIDDLE
     C*
           ISET(LS)=LP
           J=LP+1
           ISET(J)=LS
           GO TO 30
     10    CONTINUE
           IF(LS.EQ.0) GO TO 25
     C*
     C*    UNLINK THE FIRST ONE
     C*
           ISET(LS)=0
           J=LFBL+1
           ISET(J)=LS
           GO TO 30
     25    CONTINUE
     C*
     C*    UNLINK THE ONLY ITEM
     C*
           J=LFBL+1
           ISET(J)=0
           J=LFBL+2
           ISET(J)=0
           GO TO 30
     20    CONTINUE
     C*
     C*    UNLINK THE LAST ITEM
     C*
           J=LP+1
           ISET(J)=0
           J=LFBL+2
           ISET(J)=LP
     30    CONTINUE
     C*
     C*    COLLECT STATISTICS ON THE FILE, IF NECESSARY
     C*
           ISET(LFBL)=ISET(LFBL)-1
           J=LFBL+5
           LDTB=ISET(J)
           IF(LDTB.EQ.0) RETURN
           X=ISET(LFBL)
           LSTAT=ISET(LDTB)
           J=LDTB+8
           NU=ISET(J)
           CALL DTPST(X,LSTAT,NU)
           RETURN
           END
           SUBROUTINE DTPST(X,L,NU)
     C*
     C*    UPDATES THE INFORMATION  ON DISCRETE TIME PERSISTANT STATISTICS
     C*    WITH X. L IS THE LOCATION OF THE CORRESPONDING  STATISTICS'
     C*    BLOCK. NU IS THE UNIT NO ON WHICH THE STATISTIC
```

```
                    1         2         3         4         5         6         7
          1234567890123456789012345678901234567890123456789012345678901234567890123
C*    MUST BE DUMPED.
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
      COMMON ISET(1)
      EQUIVALENCE(ISET(1),RSET(1))
      LP1=L+1
      LP2=L+2
      LP3=L+3
      LP4=L+4
      LP5=L+5
      XL=RSET(LP2)
      RSET(LP2)=X
      IF(X.LT.RSET(LP3))  RSET(LP3)=X
      IF(X.GT.RSET(LP4))  RSET(LP4)=X
      DT=TNOW-RSET(LP5)
      IF(DT.LE.0)  RETURN
      XDT=XL*DT
      RSET(L)=RSET(L)+XDT
      RSET(LP1)=RSET(LP1)+XL*XDT
      RSET(LP5)=TNOW
      IF(NU.LE.0)  RETURN
      WRITE(NU)  TNOW,X
      RETURN
      END
      SUBROUTINE CTPST(X,L,NU)
C*
C*    UPDATES THE INFORMATION  ON CONTINUOUS TIME PERSISTANT STATISTICS
C*    WITH X. L IS THE LOCATION OF THE CORRESPONDING  STATISTICS'
C*    BLOCK. NU IS THE UNIT NO ON WHICH THE STATISTIC
C*    MUST BE DUMPED.
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
      COMMON ISET(1)
      EQUIVALENCE(ISET(1),RSET(1))
      LP1=L+1
      LP2=L+2
      LP3=L+3
      LP4=L+4
      LP5=L+5
      XL=RSET(LP2)
      RSET(LP2)=X
      IF(X.LT.RSET(LP3))  RSET(LP3)=X
      IF(X.GT.RSET(LP4))  RSET(LP4)=X
      DT=TNOW-RSET(LP5)
      IF(DT.LE.0)  RETURN
      XDT=(XL+X)*DT/2.
      X2DT=(XL*XL+X*X)*DT/2.
      RSET(L)=RSET(L)+XDT
      RSET(LP1)=RSET(LP1)+X2DT
```

```
                   1         2         3         4         5         6         7
         1234567890123456789012345678901234567890123456789012345678901234567890123

               RSET(LP5)=TNOW
               IF(NU.LE.0) RETURN
               WRITE(NU) TNOW,X
               RETURN
               END
               SUBROUTINE SCHD(IEVENT,TIME,A)
         C*
         C*    SCHEDULES EVENT IEVENT AT TIME TIME ON CALANDER
         C*
               DIMENSION A(99)
               COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
              -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
               COMMON/GEN5/II,TT
               COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
              -DDL(99),TNOW, XX(99),DTNOW,ISTOP
               II=IEVENT
               TT=TIME
               N=NFILE+1
               CALL FILEM(N,A)
               RETURN
               END
               SUBROUTINE TALLY (N,VAR)
         C*
         C*    COLLECTS STATISTICS ON TALLY # N USING VAR
         C*
               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/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
               COMMON ISET(1)
               EQUIVALENCE (ISET(1),RSET(1))
               IF(N.GT.NTAL.OR.N.LT.1) CALL ERROR (7,N)
               LTBL=LFTB+(N-1)*8
               L=ISET(LTBL)
               RSET(L)=RSET(L)+VAR
               J=L+1
               RSET(J)=RSET(J)+VAR*VAR
               J=L+2
               RSET(J)=RSET(J)+1.
               J=L+3
               IF(VAR.LT.RSET(J)) RSET(J)=VAR
               J=L+4
               IF(VAR.GT.RSET(J)) RSET(J)=VAR
               J=LTBL+7
               IF(ISET(J).EQ.0) RETURN
               NU=ISET(J)
               WRITE(NU) VAR
               RETURN
               END
               SUBROUTINE COUNT(N,INC)
               DIMENSION RSET(1)
               COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
               COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
              -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
               COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
```

```
                   1         2         3         4         5         6         7
          1234567890123456789012345678901234567890123456789012345678901234567890123
     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      IF(N.GT.NCOUN.OR.N.LT.1) CALL ERROR (1011,1)
      LCTB=LFCTB+(N-1)*8
      ISET(LCTB)=ISET(LCTB)+INC
      J=LCTB+7
      IF(ISET(LCTB).LT.ISET(J)) RETURN
      ISTOP=1
      RETURN
      END
      SUBROUTINE CLEARS
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
      IF(NTAL.EQ.0) GO TO 100
      DO 10 I=1,NTAL
          CALL TRUNT(I)
 10   CONTINUE
 100  IF(NDTPST.EQ.0) GO TO 200
      DO 20 I=1,NDTPST
          CALL TRUND (I)
 20   CONTINUE
 200  IF(NCTPST.EQ.0)  RETURN
      DO 30 I=1,NCTPST
          CALL TRUNC (I)
 30   CONTINUE
      RETURN
      END
      SUBROUTINE TRUND(N)
C*
C*    TRUNCATES DISCRETE TIME PERSISTANT STATISTICS
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
     -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCCUN,NATR,NENT,IECO,ISUM
      COMMON/GEN8/NIR,VALH,VALL
      COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
     -DDL(99),TNOW, XX(99),DTNOW,ISTOP
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      IF(N.GT.NDTPST.OR.N.LT.1) CALL ERROR(1006,N)
      LDBL=LFDSB+(N-1)*9
      LSTAT=ISET(LDBL)
      RSET(LSTAT)=0
      J=LSTAT+1
      RSET(J)=0
      J=LSTAT+2
      RSET(J)=0
      J=LSTAT+3
      RSET(J)=VALH
      J=LSTAT+4
      RSET(J)=VALL
      J=LSTAT+5
      RSET(J)=TNOW
```

```
               1         2         3         4         5         6         7
      1234567890123456789012345678901234567890123456789012345678901234567890123

         J=LSTAT+6
         RSET(J)=TNOW
         RETURN
         END
         SUBROUTINE TRUNC(N)
C*
C*       TRUNCATES CONTINUOUS TIME PERSISTANT STATISTICS
C*
         DIMENSION RSET(1)
         COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
         COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
        -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
         COMMON/GEN8/NIR,VALH,VALI
         COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATPIB(99),SSL(99),JJ,
        -DDL(99),TNOW, XX(99),DTNOW,ISTOP
         COMMON ISET(1)
         EQUIVALENCE (ISET(1),RSET(1))
         IF(N.GT.NCTPST.OR.N.LT.1) CALL ERROR(1005,N)
         LCBL=LFCSB+(N-1)*9
         LSTAT=ISET(LCBL)
         RSET(LSTAT)=0
         J=LSTAT+1
         RSET(J)=0
         J=LSTAT+2
         RSET(J)=0
         J=LSTAT+3
         RSET(J)=VALH
         J=LSTAT+4
         RSET(J)=VALL
         J=LSTAT+5
         RSET(J)=TNOW
         J=LSTAT+6
         RSET(J)=TNOW
         RETURN
         END
         SUBROUTINE TRUNT(N)
C*
C*       TRUNCATES STATISTICS OF TALLY N
C*
         DIMENSION RSET(1)
         COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
         COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
        -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
         COMMON/GEN8/NIR,VALH,VALL
         COMMON ISET(1)
         EQUIVALENCE (ISET(1),RSET(1))
         IF(N.GT.NTAL.OR.N.LT.1) CALL ERROR(1007,N)
         LTBL=LFTB+(N-1)*8
         LSTAT=ISET(LTBL)
         RSET(LSTAT)=0
         J=LSTAT+1
         RSET(J)=0
         J=LSTAT+2
         RSET(J)=0
         J=LSTAT+3
```

```
                1                   2                   3                   4                   5                   6                   7
       1234567890123456789012345678901234567890123456789012345678901234567890123

      RSET(J)=VALH
      J=LSTAT+4
      RSET(J)=VALL
      RETURN
      END
      FUNCTION TAVG(I)
C*
C*    FINDS THE AVERAGE OF TALLY I
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
      LTBL=LFTB+(I-1)*8
      LSTAT=ISET(LTBL)
      J=LSTAT+2
      TAVG=RSET(LSTAT)/RSET(J)
      RETURN
      END
      FUNCTION TSTD(I)
C*
C*    FINDS THE STANDARD DEVIATION OF TALLY I
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
      LTBL=LFTB+(I-1)*8
      LSTAT=ISET(LTBL)
      J=LSTAT+2
      RN=RSET(J)
      J=LSTAT+1
      TSTD=(RSET(J)-RN*RSET(LSTAT))/(RN-1)
      IF(TSTD.LT.0) TSTD=0.0
      TSTD=SQRT(TSTD)
      RETURN
      END
      FUNCTION TNUM (I)
C*
C*    FINDS THE NUMBER OF TALLIES IN TALLY I
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
      LTBL=LFTB+(I-1)*8
```

```
                 1         2         3         4         5         6         7
        1234567890123456789012345678901234567890123456789012345678901234567890123

              LSTAT=ISET(LTBL)
              J=LSTAT+2
              TNUM=RSET(J)
              RETURN
              END
              FUNCTION TMIN (I)
        C*
        C*    FINDS THE MINIMUM OF TALLY I
        C*
              DIMENSION RSET(1)
              COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
              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(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
              LTBL=LFTB+(I-1)*8
              LSTAT=ISET(LTBL)
              J=LSTAT+3
              TMIN=RSET(J)
              RETURN
              END
              FUNCTION TMAX (I)
        C*
        C*    FINDS THE MAXIMUM OF TALLY I
        C*
              DIMENSION RSET(1)
              COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
              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(I.GT.NTAL.OR.I.LT.1) CALL ERROR(1007,I)
              LTBL=LFTB+(I-1)*8
              LSTAT=ISET(LTBL)
              J=LSTAT+4
              TMAX=RSET(J)
              RETURN
              END
              SUBROUTINE DCLCT (LDBL)
        C*
        C*    THIS IS FOR COLLECTING THE DISCRETE TIME PERSISTANT
        C*    STATISTICS
        C*
              DIMENSION RSET(1)
              COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
             -DDL(99),TNOW, XX(99),DTNOW,ISTOP
              COMMON ISET(1)
              EQUIVALENCE (ISET(1),RSET(1))
              LSTAT=ISET(LDBL)
              ITYPE=ISET(LDBL+7)
              IF(ITYPE.LT.0) GO TO 10
              X=NQ(ITYPE)
              GO TO 20
        10    X=XX(-ITYPE)
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123

 20      NU=ISET(LDBL+8)
         CALL DTPST(X,LSTAT,NU)
         RETURN
         END
         SUBROUTINE CCLCT (LCBL)
C*
C*       THIS IS FOR COLLECTING THE CONTINUOUS TIME PERSISTANT
C*       STATISTICS
C*
         DIMENSION RSET(1)
         COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
        -DDL(99),TNOW, XX(99),DTNOW,ISTOP
         COMMON ISET(1)
         EQUIVALENCE (ISET(1),RSET(1))
         LSTAT=ISET(LCBL)
         ITYPE=ISET(LDBL+7)
         IF(ITYPE.LT.0) GO TO 10
         X=SS(ITYPE)
         GO TO 11
 10      X=DD(-ITYPE)
 11      NU=ISET(LCBL+8)
         CALL CTPST(X,LSTAT,NU)
         RETURN
         END
         FUNCTION DAVG (I)
C*
C*       FINDS THE AVERAGE OF DISCRETE TIME PERSISTANT STATISTICS
C*
         DIMENSION RSET(1)
         COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
         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(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
         LDBL=LFDSB+(I-1)*9
         LSTAT=ISET(LDBL)
         J=LSTAT+6
         T=TNOW-RSET(J)
         CALL DCLCT (LDBL)
         DAVG=RSET(LSTAT)/T
         RETURN
         END
         FUNCTION CAVG (I)
C*
C*       FINDS THE AVERAGE OF CONTINUOUS TIME PERSISTANT STATISTICS
C*
         DIMENSION RSET(1)
         COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
         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(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
         LCBL=LFCSB+(I-1)*9
```

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
      LSTAT=ISET(LCBL)
      J=LSTAT+6
      T=TNOW-RSET(J)
      CALL CCLCT (LCBL)
      CAVG=RSET(LSTAT)/T
      RETURN
      END
      FUNCTION DSTD (I)
C*
C*  FINDS THE STANDARD DEVIATION OF DISCRETE TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
      LDBL=LFDSB+(I-1)*9
      CALL DCLCT (LDBL)
      LSTAT=ISET(LDBL)
      J=LSTAT+6
      T=TNOW-RSET(J)
      AV=RSET(LSTAT)/T
      J=LSTAT+1
      DSTD=RSET(J)/T-AV*AV
      IF(DSTD.LT.0) DSTD=0
      DSTD=SQRT(DSTD)
      RETURN
      END
      FUNCTION CSTD (I)
C*
C* FINDS THE STANDARD DEVIATION OF CONTINUOUS TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
      LCBL=LFCSB+(I-1)*9
      CALL CCLCT (LCBL)
      LSTAT=ISET(LCBL)
      J=LSTAT+6
      T=TNOW-RSET(J)
      AV=RSET(LSTAT)/T
      J=LSTAT+1
      CSTD=RSET(J)/T-AV*AV
      IF(CSTD.LT.0) CSTD=0
      CSTD=SQRT(CSTD)
      RETURN
      END
      FUNCTION DPRD (I)
C*
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123
C*    FINDS THE PERIOD OF DISCRETE TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
      LDBL=LFDSB+(I-1)*9
      LSTAT=ISET(LDBL)
      J=LSTAT+6
      DPRD=TNOW-RSET(J)
      RETURN
      END
      FUNCTION CPRD (I)
C*
C*    FINDS THE PERIOD OF CONTINUOUS TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
      LCBL=LFCSB+(I-1)*9
      LSTAT=ISET(LCBL)
      J=LSTAT+6
      CPRD=TNOW-RSET(J)
      RETURN
      END
      FUNCTION DMIN (I)
C*
C*    FINDS THE MINIMUM OF DISCRETE TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
      LDBL=LFDSB+(I-1)*9
      CALL DCLCT (LDBL)
      LSTAT=ISET(LDBL)
      J=LSTAT+3
      DMIN=RSET(J)
      RETURN
      END
      FUNCTION CMIN (I)
C*
C*    FINDS THE MINIMUM OF CONTINUOUS TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123

      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
      LCBL=LFCSB+(I-1)*9
      CALL CCLCT (LCBL)
      LSTAT=ISET(LCBL)
      J=LSTAT+3
      CMIN=RSET(J)
      RETURN
      END
      FUNCTION DMAX (I)
C*
C*    FINDS THE MAXIMUM OF DISCRETE TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NDTPST.OR.I.LT.1) CALL ERROR(1006,I)
      LDBL=LFDSB+(I-1)*9
      CALL DCLCT (LDBL)
      LSTAT=ISET(LDBL)
      J=LSTAT+4
      DMAX=RSET(J)
      RETURN
      END
      FUNCTION CMAX (I)
C*
C*    FINDS THE MAXIMUM OF CONTINUOUS TIME PERSISTANT STATISTIC
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(I.GT.NCTPST.OR.I.LT.1) CALL ERROR(1005,I)
      LCBL=LFCSB+(I-1)*9
      CALL CCLCT (LCBL)
      LSTAT=ISET(LCBL)
      J=LSTAT+4
      CMAX=RSET(J)
      RETURN
      END
      FUNCTION ATR(LENT,I)
C*
C*    FIDS THE VALUE OF ITH ATRIBUTE OF THE ENTRY AT LOCATION LENT
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123
      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(I.GT.NATR.OR.I.LT.1) CALL ERROR(1012,I)
      J=LENT+3
      LATR=ISET(J)
      J=LATR+I
      ATR=RSET(J)
      RETURN
      END
      FUNCTION LPRED (LENT)
C*
C*    GIVES THE PREDECESSOR OF THE ENTRY LOCATED AT LENT
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      LPRED=ISET(LENT)
      RETURN
      END
      FUNCTION LSUCC (LENT)
C*
C*    GIVES THE SUCCESSOR OF THE ENTRY LOCATED AT LENT
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      COMMON ISET(1)
      EQUIVALENCE (ISET(1),RSET(1))
      J=LENT+1
      LSUCC=ISET(J)
      RETURN
      END
      FUNCTION NQ (IFILE)
C*
C*    GIVES NO. OF ENTITIES IN FILE I
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
      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(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
      LFBL=LFFB+(IFILE-1)*6
      NQ=ISET(LFBL)
      RETURN
      END
      FUNCTION LFE (I)
C*
C*    GIVES THE LOCATION OF FIRST ENTRY OF FILE I
C*
      DIMENSION RSET(1)
      COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
```

```
           1         2         3         4         5         6         7
  1234567890123456789012345678901234567890123456789012345678901234567890123
          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(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
          LFBL=LFFB+(I-1)*6
          J=LFBL+1
          LFE=ISET(J)
          RETURN
          END
          FUNCTION LLE (I)
  C*
  C*      GIVES THE LOCATION OF LAST ENTRY OF FILE I
  C*
          DIMENSION RSET(
          COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
          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(IFILE.GT.(NFILE+1).OR.IFILE.LT.1) CALL ERROR(1010,IFILE)
          LFBL=LFFB+(I-1)*6
          J=LFBL+2
          LLE=ISET(J)
          RETURN
          END
          SUBROUTINE OUTPUT
          RETURN
          END
          SUBROUTINE SUMRY (NNN)
          DIMENSION RSET(1)
          COMMON/LOC/LFI,LLR,LFFB,LFTB,LFDSB,LFCSB,LFCTB,LFAE,LCAL
          COMMON/GEN1/NRUNS,NFILE,TBEG,TFIN,ICLF,ICLV,ICLS,TCLEAR,IRAN,
         -NTMX,LTMX(99),NTAL,NDTPST,NCTPST,NCOUN,NATR,NENT,IECO,ISUM
          COMMON/GEN2/NEQD,NEQS,NEQT,DTSAV,DTMIN,DTMAX,AERR,RERR,IERR,NSEV,
         -ISEES,ICSV(25),ICGV(25),ICDIR(25),ICDV(25),VAL(25),TTOL(25),
         -ISCD(25)
          COMMON/GEN3/ITITLE(15),IANAL(10),IDATE(5)
          COMMON/GSC1/NCRDR,NPRTR,SS(99),DD(99),ATRIB(99),SSL(99),JJ,
         -DDL(99),TNOW, XX(99),DTNOW,ISTOP
          COMMON ISET(1)
          EQUIVALENCE(ISET(1),RSET(1))
          WRITE(NPRTR,10)
  10      FORMAT(1H1,///50X,28HS U M M A R Y    R E P O R T)
          WRITE(NPRTR,11) (ITITLE(I),I=1,15),(IANAL(I),I=1,10),(IDATE(I),
         -I=1,5),NNN,NRUNS
  11      FORMAT(//25X,9HPROJECT: ,15A2,15X,9HANALYST: ,10A2/
         -,25X,6HDATE: ,5A2,38X,3HRUN,I3,3H OF,I3)
          WRITE(NPRTR,12) TBEG,TCLEAR,TNOW
  12      FORMAT(//25X,28HSIMULATION STARTED AT TIME: ,E11.2/,25X,
         -28HSTATISTICS CLEARED AT TIME: ,E11.2,
  12     -/25X,14HCURRENT TIME: ,E11.2)
          IF(NCOUN.EQ.0) GO TO 19
          WRITE(NPRTR,15)
  15      FORMAT(////50X,28H**** COUNTER INFORMATIN ****///15X,5HINDEX,
```

```
                 1         2         3         4         5         6         7
        1234567890123456789012345678901234567890123456789012345678901234567890123
     -10X,5HLABEL,20X,5HLIMIT,10X,13HCURRENT VALUE/15X,5(1H-),
     -10X,5(1H-),20X,5(1H-),10X,13(1H-))
      DO 16 I=1,NCOUN
      LCTB=LFCTB+(I-1)*8
      J1=LCTB+1
      J2=LCTB+6
      J3=LCTB+7
      WRITE(NPRTR,17) I,(ISET(J),J=J1,J2),ISET(J3),ISET(LCTB)
17    FORMAT(/16X,I3,11X,6A2,12X,I6,13X,I6)
16    CONTINUE
19    IF(NTAL.EQ.0) GO TO 30
      WRITE(NPRTR,20)
20    FORMAT(/////50X,26H**** TALLY STATISTICS ****///15X,5HINDEX,8X,
     -5HLABEL,9X,11HNO. OF OBS.,12X,4HMEAN,11X,9HSTD. DEV.,10X,
     -7HMINIMUM,10X,7HMAXIMUM)
      WRITE(NPRTR,201)
201   FORMAT(15X,5(1H-),8X,5(1H-),9X,11(1H-),12X,4(1H-),11X,9(1H-),
     -10X,7(1H-),10X,7(1H-))
      DO 21 I=1,NTAL
      LTBL=LFTB+(I-1)*8
      J1=LTBL+1
      J2=LTBL+6
      WRITE(NPRTR,22) I,(ISET(K),K=J1,J2)
22    FORMAT(/16X,I3,6X,6A2)
      LSTAT=ISET(LTBL)
      J=LSTAT+2
      N=RSET(J)
      AV=RSET(LSTAT)/RSET(J)
      STD=(RSET(LSTAT+1)-RSET(J)*AV*AV)/(RSET(J)-1.)
      IF(STD.LT.0) STD=0
      STD=SQRT(STD)
      RMN=RSET(LSTAT+3)
      RMX=RSET(LSTAT+4)
      WRITE(NPRTR,23) N,AV,STD,RMN,RMX
23    FORMAT(1H+,43X,I8,8X,E11.2,6X,E11.2,7X,E11.2,6X,E11.2)
21    CONTINUE
30    IF(NDTPST.EQ.0) GO TO 40
      WRITE(NPRTR,31)
31    FORMAT(/////43X,45H**** DISCRETE 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-))
      DO 32 I=1,NDTPST
      LDBL=LFDSB+(I-1)*9
      J1=LDBL+1
      J2=LDBL+6
      WRITE(NPRTR,22) I,(ISET(K),K=J1,J2)
      LSTAT=ISET(LDBL)
      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)
```

```
                   1         2         3         4         5         6         7
         1234567890123456789012345678901234567890123456789012345678901234567890123

               WRITE(NPRTR,34)AV,STD,RMN,RMX
         34    FORMAT(1H+,38X,E11.2,6X,E11.2,7X,E11.2,6X,E11.2)
         32    CONTINUE
         40    IF(NCTPST.EQ.0) GO TO 50
               WRITE(NPRTR,41)
         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-))
               DO 42 I=1,NCTPST
               LCBL=LFCSB+(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
         44    FORMAT(1H+,38X,E11.2,6X,E11.2,7X,E11.2,6X,E11.2)
         42    CONTINUE
         50    IF(NEQT.EQ.0) GO TO 60
               WRITE(NPRTR,130)
         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)
         131   FORMAT(35X,3HSS(,I2,2H)=,E10.2,18X,3HDD(,I2,2H)=,E10.2)
         51    CONTINUE
         60    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
         10    CALL ERROR(1014,1)
         90    ISEED(ISTRM)=ISEED(ISTRM)+2147483647+1
         100   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
 1234567890123456789012345678901234567890123456789012345678901234567890123

            IF(K.LT.1.OR.B.LT.0) CALL ERROR(1015,1)
            R=1
            DO 10 I=1,K
               R=R*DRAND(IST)
   10       CONTINUE
            ERLNG=-B*ALOG(R)
            RETURN
            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,QS,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/
            IF(STD.LT.0) CALL ERROR(1015,1)
            IF(IEVEN(ISTRM).GT.1) GO TO 20
   10       UA=2.*DRAND(ISTRM)-1.
            UB=2.*DRAND(ISTRM)-1.
            IEVEN(ISTRM)=2
            W=UA*UA+UB*UB
            IF(W.GT.1.0) GO TO 10
            W=SQRT(-2.*ALOG(W)/W)
            RNORM=UA*W
            ENORM(ISTRM)=UB*W
            GO TO 30
   20       IEVEN(ISTRM)=1
            RNORM=ENORM(ISTRM)
   30       RNORM=RNORM*STD+XMN
            RETURN
            END
            FUNCTION TRIAG (XL,XM,XH,ISTRM)
            IF(XL.GT.XM.OR.XL.GT.XH.OR.XM.GT.XH) CALL ERROR(1015,1)
            RN=DRAND(ISTRM)
            BMA=XM-XL
            CMA=XH-XL
            IF(RN-BMA/CMA) 10,10,20
     10     TRIAG=XL+SQRT(BMA*CMA*RN)
            RETURN
   20       TRIAG=XH-SQRT(CMA*(1.-RN)*(XH-XM))
            RETURN
            END
```

```
                  1         2         3         4         5         6         7
         1234567890123456789012345678901234567890123456789012345678901234567890123
               FUNCTION WEIBUL(B,A,ISTRM)
               WEIBUL=(-B*ALOG(DRAND(ISTRM)))**(1./A)
               RETURN
               END
               FUNCTION BETA(A,B,ISTRM)
               X=GAMA(1.,A,ISTRM)
               BETA=X/(X+GAMA(1.,B,ISTRM))
               RETURN
               END
               FUNCTION GAM(AK,ISTRM)
               K=AK
               FK=K
               GAM=0
               IF(K) 30,30,10
         10    PROD=1.0
               DO 20 I=1,K
         20    PROD=PROD*DRAND(ISTRM)
               GAM=-ALOG(PROD)
         30    DG=AK-FK
               IF (DG-.015) 100,100,40
         40    IF (DG-.985) 60,50,50
         50    W=1.
               GO TO 90
         60    A=1./DG
               B=1./(1.-DG)
         70    X=DRAND(ISTRM)**A
               Y=DRAND(ISTRM)**B+X
               IF(Y-1.) 80,80,70
         80    W=X/Y
         90    Y=-ALOG(DRAND(ISTRM))
               GAM=GAM+W*Y
         100   RETURN
               END
               FUNCTION GAMA(B,A,ISTRM)
               IF(A.GE.1.0) GO TO 10
               GAMA=GAM(A,ISTRM)*B
               RETURN
         10    KA=A
               AK=KA
               C=A-AK
               IF(A.LT.5.) GO TO 20
               IF(DRAND(ISTRM).LT.C) KA=KA+1
         20    PR=1.0
               DO 30 I=1,KA
         30    PR=PR*DRAND(ISTRM)
               GAMA=-ALOG(PR)
               IF(A.GE.5.) GO TO 40
               TEST=(GAMA/AK)**C*EXP(-C*(GAMA/A-1.))
               IF(DRAND(ISTRM).GT.TEST) GO TO 20
               GAMA=GAMA*(A/AK)
         40    GAMA=GAMA*B
               RETURN
               END
               FUNCTION NPSSN(P,ISTRM)
               IF(P.LT.0) CALL ERRCR(1015,1)
```

```
                  1         2         3         4         5         6         7
         1234567890123456789012345678901234567890123456789012345678901234567890123

               NPSSN=0
               IF( P-9) 10,10,40
         10    Y=EXP(-P)
               X=1.0
         20    X=X*DRAND(ISTRM)
               IF (X-Y) 50,30,30
         30    NPSSN=NPSSN+1
               GO TO 20
         40    Z=SQRT(P)
               NPSSN=RNORM(P,Z,ISTRM)+.5
         50    RETURN
               END
               FUNCTION DPROB(CPROB,VALUE,NVAR,ISTRM)
               DIMENSION CPROB(1),VALUE(1)
               RN=DRAND(ISTRM)
               DO 10 I=1,NVAR
                  IF(RN.LE.CPROB(I)) GO TO 20
         10    CONTINUE
               CALL ERROR(1016,1)
         20    DPROB=VALUE(I)
               RETURN
               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

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

Toke Jayachandran
Naval Postgraduate School
Monterey, CA 93940

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

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

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)

Naval Ocean Systems Center
Code 16
San Diego, CA 92132

Newport Laboratory
Naval Underwater Systems Center
Newport, RI 02840

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

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

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

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

END

DATE FILMED

7-81

DTIC