To the Graduate Council:

I am submitting herewith a dissertation written by Henrique Martini Paula entitled "A Methodology for Time-Dependent Reliability Analysis of Accident Sequences and Complex Reactor Systems." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

Jerry B. Fussell, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

The Graduate School

A METHODOLOGY FOR TIME-DEPENDENT RELIABILITY ANALYSIS OF ACCIDENT SEQUENCES AND COMPLEX REACTOR SYSTEMS

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Henrique Martini Paula

December 1984

I dedicate this work to my wife Célia, our daughters Daniela and Larissa, and to my late father Camilo.

ACKNOWLEDGMENTS

In the course of this dissertation research, a number of scholars have assisted me generously. I am particularly indebted to my Professor Advisor Jerry Bernard Fussell for his consultation and criticism. I extend my sincere appreciation to Professors P. F. Pasqua (head of the department of nuclear engineering), H. L. Dodds, and L. S. Husch, all from the University of Tennessee. I am also indebted to Dr. Jussi K. Vaurio, from the Argonne National Laboratory, for his consultation. Although the participation of these scholars has improved the quality of this dissertation research, they share no responsibility for whatever errors remain in my work. I also express my gratitude for the support of Professor Pedro Bento de Camargo, from the "Instituto de Pesquisas Energéticas e Nucleares (IPEN)," Brazil.

Funding for this research has been provided by IPEN and the "Comissão Nacional de Energia Nuclear," Brazil.

I greatly appreciate the encouragement and support of my beloved wife, Célia Maria.

ABSTRACT

The work presented here is of direct use in Probabilistic Risk Assessment (PRA) and is of value to Utilities as well as the Nuclear Regulatory Commission (NRC).

Specifically, this report presents a methodology and a computer program to calculate the expected number of occurrences for each accident sequence in an event tree. The methodology evaluates the time-dependent (instantaneous) and the average behavior of the accident sequence. The methodology accounts for standby safety system and component failures that occur (a) before they are demanded, (b) upon demand, and (c) during the mission (system operation). With respect to failures that occur during the mission, this methodology is unique in the sense that it models components that can be repaired during the mission. The expected number of system failures during the mission provides an upper bound for the probability of a system failure to run -- the mission unreliability. The basic event modeling includes components that are continuously monitored, periodically tested, and those that are not tested or are otherwise nonrepairable.

The computer program ASA allows practical applications of the method developed. This work represents a required extension of the presently available methodology and allows a more realistic PRA of nuclear power plants.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Nuclear reactor safety concerns the public, the utilities that operate the nuclear power reactors, and the government. Increased public concern creates a need for advanced methods of risk assessment to provide proper information to the decision makers of our society. This dissertation extends the methodologies and computer programs available for the estimation of the potential for nuclear accidents. It also expands the techniques of reliability analysis as applied to large, complex systems such as those in the nuclear and chemical industries.

In particular, the complexity and the size of the protection system of nuclear power reactors require more advanced methods of system reliability analysis. The protection system includes the containment isolation system (CI), the emergency core cooling system (ECCS), the containment-pressure reduction system, the emergency or auxiliary power system, the air filtration system, and the reactor shutdown system . The protection system mitigates the consequences resulting from the occurrence of accident initiating events, such as a loss of coolant accident (LOCA), in nuclear power reactors. In this context, the systems mentioned form a group of standby systems which act when the accident initiating event occurs.

Upon the occurrence of the accident initiating event, each of the standby systems either works or fails (success or failure). Any combination of failures and successes of the standby systems is

called an accident sequence. The expected number of times that an accident sequence occurs over a period of time represents an important safety parameter. This parameter, called the Expected Number of Failures (ENF) for the accident sequence, has direct application in probabilistic risk assessment. The ENF is also an important parameter in reliability analysis, where it provides an important tool in optimization studies.

A standby system's failure may occur in two ways. A standby system may fail to start its intended function, or it may start to function properly but fail to perform adequately during the required period of operation. The term "failure to run" labels this second case. Analysis of these systems' failures may require different reliability failure models (fault trees, reliability block diagrams, and so forth).

I.1 PURPOSE AND SCOPE

This dissertation develops a methodology and a computer program to calculate the expected number of occurrences for each accident sequence in an event tree, where fault trees are available for each branching point in the event tree. The methodology evaluates the time-dependent behavior of a generic accident sequence, allowing a detailed modeling of periodic testing and repair.

The time-dependent analysis of the systems in the accident sequence includes not only the possibility of a system's failure to start at a point in time but also the chance of a system's failure to perform its intended function, following a successful start. Thus,

the analysis accounts for failures that occur before the demand, the failures that occur on demand, and the failures that occur during the mission. With respect to the failures that occur during the mission, the methodology developed in this dissertation is unique in the sense that it allows the system's components to be repairable during the mission time. The following assumptions are used in the methodology:

1. The fault trees representing the system's failures at each branching point in the event tree are coherent.[27]
2. The different basic events in the fault trees are statistically independent in failure and repair.
3. The expected number of accident sequence occurrences during the mission is used to provide a conservative upper bound to the mission unreliability (the probability that the system formed by all the systems in the accident sequence starts and fails to perform adequately during the mission).
4. The repairable systems in the accident sequence or combinations of such systems are assumed to be nonrepairable during the mission (a conservative assumption).

In the analysis of a single system, assumption 4 needs not be made. In some applications, the systems in the accident sequence share no basic events and are independent of each other. In these cases, each system may be treated as an independent event, and the

combination of such independent events becomes a single system for which assumption 4 is unnecessary.

This is the case, for example, presented by Elerath, Vaurio, and Wood.[23] They modeled the common support systems (power supplies) as independent systems in parallel with the front line systems (the cooling systems). In this way, each of the front line and support systems becomes an independent basic event. Chapter II presents the analysis of this same problem using the computer program developed in the study presented here.

Chapter II dicusses assumptions 3 and 4 in more detail. The remainder of this section briefly discusses assumptions 1 and 2 and phased-mission analysis (a related topic).

The fault trees representing the system's failures at each branching point in the event tree are assumed to be coherent as defined by Esary and Prochan.[27] As a consequence of this assumption, the following topics are not directly addressed in this work:

1. Secondary Failures. These are causes of component malfunction for which the component itself is not held accountable; however, the component needs to be repaired before the malfunction is corrected (see reference 72 for a methodology that considers secondary failures).
2. "DELAY" Gates. These represent situations in which the top event occurs if the basic events in at least one MCS exist for a sufficient time period (a definition of an MCS appears on page 9). For

example, a "vessel rupture" results if the malfunction causing a high-pressure condition persist long enough that the pressure can build up to the rupture point.

3. Mutually Exclusive "OR" Gates. These represent situations in which the existence of one of the inputs to the "OR" gate guarantees the nonexistence of another input.

4. Basic events in the fault trees which, when they exist, decrease the existence probability of the top event. This is to say that the top event must be a nondecreasing function of the existence probabilities of the basic events.

In addition to the assumption of coherence, the different basic events in the fault trees are assumed to be statistically independent in both failure and repair. The study of dependencies among components' failures has received increased attention in PRA,[14,15,21,30] where it has become a complex part of the analysis. In some cases, as in flood analysis, special event trees/fault trees are required to study dependencies because accident-sequence issues are different for these scenarios than for accidents initiated from other sources (see, for example, the comments by Budnitz[11] on flood analysis). Parametric methods such as the Beta-factor[30] and the BFR[6] models account for other types of dependencies, those not explicitly considered in the event trees or fault trees.[30]

A third method to study dependencies consists of using computer programs such as COMCAN II[72] and SETS,[102] which identify the root causes of dependencies, called the root cause events (e.g., high temperature and high vibration conditions). In practice, a combination of the methods above provides an adequate inventory of techniques for the Dependent Failure Analysis (DFA). This dissertation does not address, but is complemented by, dependent failure analysis.

In some applications, the system configuration may change at some times. A good example is the Emergency Core Cooling System (ECCS) changing from the High Pressure Coolant Injection (HPCI) to the Low Pressure Coolant Injection (LPCI). At such moments, the accident sequence's and the system's failure probability may change without a component failure or repair. This situation, called "Phased-Mission Analysis," requires special mathematical treatment[13,39,62] and is not addressed in this dissertation.

I.2 STRUCTURE OF THE DISSERTATION

The remainder of this chapter discusses the following topics: 1) important definitions, 2) existing methodologies and computer programs related to the subject of this research, 3) reliability and system reliability analyses, and 4) comments on data.

Chapter II describes the methodology developed in this study. It starts with the derivation of an integral equation for the accident sequence's ENF. The ENF is a function of the initiating event's rate of failure and the accident sequence's

undependability. Then Chapter II defines the system's and accident sequence's time-dependent undependabilities. Sections II.3 and II.4 show how to calculate the accident sequence's undependability as a function of the basic event's unavailabilities. The equations for the basic event's unavailabilities are derived in Section II.5. Chapter II then presents a brief description of the computer program used to implement the methodology developed here. The last section shows an example problem extracted from the available literature.

Chapter III summarizes the report and concludes the research. Recommendations for further studies are also presented. Appendix A contains the information necessary to run the ASA code. Appendix B shows an example of an input to the code, and, in conjunction with Appendix A, should provide enough detail to allow the use of the program. Appendix C shows the program output to the example in Appendix B. The listing of the ASA computer program appears in Appendix D.

I.3 IMPORTANT DEFINITIONS

Event tree and fault tree analyses are major tools in system reliability analysis. The event tree identifies the possible outcomes resulting from accident initiating events. The accident initiating event may be a system failure (such as a ruptured pipe or a thermal transient) or an external event (such as loss of off-site power or an earthquake). The event tree begins with the initiating event. Next, all the systems that affect the subsequent course of

events are identified, defined, and ordered in a convenient form. Each system may either work or fail following the occurrence of the initiating event. Under this binary assumption, each system generates two branches in the event tree. References 59 and 99 provide detailed information about event trees.

Fault trees provide the means to compute the probabilities of failure and successes at each branching point in the event tree. A fault tree is a logical representation of an undesirable event or malfunction of interest, called the "top event," in terms of the basic failures that lead to the top event. The basic failures are called the "basic events." The top event may represent a general statement such as "Release of radioactivity from a nuclear power plant." It can also represent a specific system failure, such as "Failure to insert control rods" or "Motor does not operate."

Given a particular system failure in the event tree, fault trees identify the various combinations of basic failures that lead to the given system failure. The probabilistic information available for the basic events in the fault trees is then combined to provide probabilistic information for the top event. Fault trees are often very large, and computer programs are usually required to perform the desired calculations. Examples of such computer programs are SUPERPOCUS, KITT, FRANTIC, and FRANTIC II-MIT (see references 37, 90, 94, and 20, respectively).

A general time-dependent and quantitative reliability analysis must include not only the possibility of a system's failure to start at a point in time but also the chance of a system's failure

to run following a successful start. The analysis should then account for failures that occur before the demand (predemand), failures that occur on demand (demand-related), and the failures that occur during the mission (postdemand). Table I.1 lists the factors which affect these failures.

In this study, a component is said to have failed to "start" if the failure results from a predemand or from a demand-related factor. If a component's failure results from a postdemand factor, the component is said to have failed to "run."

Some basic events in the fault trees represent component failures that can result from predemand or demand-related as well as postdemand factors (e.g., a diesel generator can fail to start or fail to run). These basic events are here defined as "Type I" basic events. Some basic events represent components' failures to start only. The basic events which represent failures to start only (predemand and demand-related failures) are here defined as "Type II" basic events.

In this dissertation, the approach selected to relate the systems and accident sequences to the basic events in the fault tree makes use of the Minimal Cut Sets (MCSs) and the Minimal Path Sets (MPSs). An MCS is a collection of basic events, such that the simultaneous existence of all these events guarantees the existence of the top event, and removal of any basic event from this set no longer guarantees the existence of the top event.[39] An MPS is the smallest collection of basic events, such that if all the basic

Table I.1 Factors Affecting the Basic Event and the Accident Sequence Unavailabilities.

Factor Affecting the Unavailability	Example
Random failures and repair prior to the demand or during the mission (predemand and postdemand factors).	A pump that fails on standby due to corrosion problems or a pump that fails to run due to normal wear of a moving part.
Unavailability due to testing, repair, and maintenance (predemand factor).	A pump which is unavailable because of the inability to recover from a test, repair, or maintenance work.
Human errors in testing, maintenance, and repair (usually a predemand factor).	A check valve that is installed backwards.
Unavailability due to unrevealed design, manufacturing, installation, and commissioning errors (predemand, demand-related and postdemand effects).	The use of inadequate components, poor welding, component damaged during installation, and so forth.
Unavailability due to extreme environmental conditions such as high temperature, humidity, contamination, vibration, etc. (predemand, demand-related and postdemand effects).	A pump- or valve- seal that fails due to contamination damage.

Table I.1 (continued)

Factor Affecting the Unavailability	Example
Failures due to events such as fire, flooding, earthquakes, etc. (usually a post-demand or demand-related factor).	A building collapsing during an earthquake, resulting in several component failures.
Human errors during abnormal conditions (demand-related or a postdemand factor).	Failure to follow the emergency procedures or failure due to misinterpretation of the information available.
Defective procedures (predemand, demand-related or post-demand factor).	A defective operation procedure that results in reactor shutdown.
Changes in the system (postdemand factor)	A system change during a a phased-mission.

events simultaneously do not exist then the top event is guaranteed not to exist.[39]

An MCS fails to start if all of its basic events fail to start. Any other MCS failure is a failure to run. An accident sequence (or a system) fails to start if at least one of its MCSs fails to start. Any other accident sequence (system) failure is a failure to run. An MCS containing only Type II basic events cannot fail to run; it either fails to start or does not fail. Within this framework of definitions, the same fault tree represents the system's failure to start and the system's failure to run.

I.4 EXISTING METHODOLOGIES AND COMPUTER PROGRAMS FOR QUANTITATIVE RELIABILITY ANALYSIS

Several methodologies and computer programs are currently available for the quantitative reliability analysis of complex systems. Basically, they synthesize probabilistic information for a system from the probabilistic information available for the system's components or parts. Some of the computer programs for quantitative system reliability analysis are briefly described in Table I.2.

Some of the codes in Table I.2 (particularly the FRANTIC family and FRANCALC) evaluate the time-dependent and average unavailability for any general system model. They analyze components with a constant probability of failure, nonrepairable components, monitored components, and periodically tested components. FRANTIC II and FRANTIC II-MIT allow even more detailed treatment of the basic events.[20,96] The unique feature of these computer programs is the

Table I.2 Some Computer Programs for Quantitative System Reliability Analysis.

Computer Program	Comments
FAUNET[a]	Written for minicomputer; uses a top-down pruning and botton-up modularization Boolean substitution algorithm; also calculates top event existence probability.
GO[b]	Uses a Boolean manipulation algorithm; also calculates point estimates of top event characteristics.
KITT 1&2[c]	Calculate time-dependent availability, reliability, and expected number of failures from component failure rates and repair times; also calculate component importance. For some controversy on the computation of the time-dependent reliability, see references 18, 63, 89, and 91 in the list of references.
SUPERPOCUS[d]	Calculates the same time-dependent characteristics as KITT, using bounding approximation methods.
WAM-BAM[e]	Calculates Top event existence probabilities using Boolean manipulation.
FRANTIC[f]	Calculates the effect on the average unavailability and time-dependent instantaneous unavailability due to periodic system testing; testing characteristics include the test interval, test duration time, repair time, and human-caused failure potential associated with the testing. This program requires an analytical expression of the system unavailability as a function of the component unavailabilities as an input.

Table I.2 (continued)

Computer Program	Comments
PROBCALC[g]	Uses a direct evaluation method that does not rely on MCSs and Boolean reduction but employs a numeric technique for calculating the probabilities of series and parallel paths in either success or failure state diagrams or from fault trees.
FRANCALC[h]	Combines the features of FRANTIC and PROBCALC with additional improvements such as a criteria for ordering and terminating conditional probability calculations in order to reduce computing time within a special level of accuracy. It also accepts m-out-of-n:G subsystems. Mutually exclusive and multiple failure modes can also be defined for components.
FRANTIC II and FRANTIC II-MIT[i]	The original FRANTIC code assumes constant failure rates, and its extension (the FRANTIC II code) models burn-in and wear-out periods. Dykes has further improved the code with several modifications and additions including interfacing the code with a cut set generator. The resulting computer program is called FRANTIC II-MIT.

[a] Platz, O. and Olsen, J.V., "FAUNET: A Program Package for Evaluation of Fault Trees and Networks," Report RISO-348, Danish Atomic Energy Commission, September 1976.

[b] Gately, W.Y. and Williams, R.L., "GO methodoloty- System Reliability Assessment and Computer Code Manual," Electric Power Research Institute, EPRI NP-766, May 1978.

Table I.2 (continued)

c
Vesely,W.E. and Narum,R.E.,"PREP and KITT: Computer Codes for the Automatic Evaluation of a Fault Tree," USAEC Rep. IN 1349, Idaho Nuclear Corporation, August 1970.

d
Fussell, J.B., et al.,"SUPERPOCUS- A Computer Program for Calculating System Probabilistic Reliability and Safety Characteristics," NERS-77-01, Knoxville: Nuclear Engineering Department, May 1977.

e
"WAMCUT-II, A Fault Tree Evaluation Program," Electric Power Research Institute, EPRI NP-2421, June 1982.

Leverenz, F.L. and Kirch, H., "User's Guide for the WAM-BAM Computer Code," Electric Power Research Institute, EPRI-217-2-5, January 1976.

f
Vesely,W.E. and Goldberg,F.F.,"FRANTIC- A Computer Code for Time Dependent Unavailability Analysis," NUREG-0193, United States Nuclear Regulatory Commission, March 1977.

g
Elerath, J.G. and Ingram, G.E., "PROBCALC User's Manual: A Computer Program for Probability Calculation," GEFR-00408, General Electric Company, Sunnyvale, California, 1980.

h
Elerath, J.G., Vaurio, J.K., and Wood, A.P., "A Method for Reliability Analysis of Complex Reactor Systems," The ANS/ENS International Meeting on Fast Safety Technology, Lyon, France, July 1982.

i
Dykes, A.A., "Application of Time Dependent Unavailability Analysis to Standby Safety Systems," Ph.D. Dissertation, Department of Nuclear Engineering, MIT, June 1982.

Vesely,W.E., Goldberg,F.F., Powers,J.T., et al.,"FRANTIC II- A Computer Code for Time Dependent Unavailability Analysis," NUREG/CR-1924, April 1981.

detailed, time-dependent modeling of periodic testing and repair. This modeling accounts for the effects of the test downtime, test overrides, detection inefficiencies, test-caused failures, and so forth.

FRANTIC requires a user-defined analytical expression relating the system failure probability to the component failure probabilities. The latest version, FRANTIC II-MIT, which includes an MCS generator, uses MCSs to perform this task. FRANCALC works in a different way, making no use of MCSs nor of Boolean reduction. The approach in FRANCALC consists of computing the system failure probability by evaluating series and parallel arrangements of components. For complex logics, when components appear in more than one path of a block diagram, the code applies the theorem of total probability [23] to the system. To illustrate this approach, suppose that an event E appears in more than one path of a failure block diagram (or, equivalently, in more than one MCS). The system failure probability, $P(\bar{S})$, is then given by

$$P(\bar{S}) = P(\bar{S}|E)\, P(E) + P(\bar{S}|\bar{E})\, P(\bar{E}), \qquad (1.1)$$

which can be applied to any number of such events. In the preceding equation, $P(\bar{E})$ and $P(E)$ are the probabilities of occurrence and nonoccurrence of the event E, respectively.

A quantitative system reliability analysis must include the possibility of a system's failure to start and the chance of a system's failure to run following a successful start. The

methodology developed and implemented by Elerath, Vaurio,and Wood[23] in FRANCALC models the two components of the system's failure probability when the basic events are assumed to be nonrepairable during the mission. The basic events, however, may be repaired at any other time. FRANTIC, FRANTIC II, and FRANTIC II-MIT perform the same analysis under the same assumption.[23]

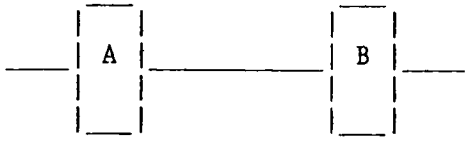
Elerath, Vaurio, and Wood have analyzed the total probability of a system failure by combining the probability that the component starts and fails to run and the probability that the component fails to start. The total component failure probabilities are then input to the fault tree model (or any other reliability model) to compute the probability that the system will fail. Since the components are assumed to be independent in failure, and there is no repair, the probability of a system's failure depends only on the state of the components at the end of the mission. For example, if all the components are working at the end of the mission, the system's failure has not occurred.

This is not the case when component repair is allowed during the mission. Knowing that all the components are working at the end of the mission does not guarantee that some of them were not failed during the mission, causing the top event to exist. The approach by Elerath, Vaurio, and Wood, here defined as the "existing approach," does not apply to the case where the components are repairable during the mission. The existing approach can only be used to provide an upper bound in this case (this is accomplished by assuming that the repairable basic events are nonrepairable).

The approach proposed in the work presented here calculates the probability of system failure by combining the probability that the system starts and fails to run and the probability that the system fails to start. This allows a more realistic solution to the problem. This approach, here defined as the "proposed approach," combines the probabilities of failure to start and failure to run at the system level while the existing approach combines them at the basic event (or component) level.

I.4.1 Comparison of Existing and Proposed Methodologies: A Simple Example

The following simple example shows a major difference between the existing and the proposed approaches. In this example, a system is composed of two components. The components fail independently and are nonrepairable during the mission. The failure block diagram for this system is as following:



When a demand occurs, the system may fail to start. If the system starts, two possibilities remain: the system may perform its required function or the system may fail to run. The notation used in this example is consistent with the notation used in Chapter II and is given below. The component's unreliability is its conditional failure probability, given a successful start:

i	represents components A and B.
$\bar{a}_i$	the unavailability of component i.
$\bar{r}_i$	the unreliability of component i.
$a_i = 1-\bar{a}_i$	the availability of component i.
$r_i = 1-\bar{r}_i$	the reliability of component i.
$P=\bar{a}_i + a_i \bar{r}_i$	the probability that component i fails.

For simplicity, $\bar{a}_A = \bar{a}_B = \bar{a}$ and $\bar{r}_A = \bar{r}_B = \bar{r}$, where $\bar{a}$ and $\bar{r}$ are constants with respect to time. According to the description in reference 23 (pages 2-4), the existing approach calculates the system's undependability as follows:

$$P(\text{system fails}) = P(AB) = P(A)\,P(B) = P^2,$$

where $P=\bar{a}+(1-\bar{a})\bar{r}$. Thus,

$$P(\text{system fails}) = \left(\bar{a}+(1-\bar{a})\bar{r} \right)^2$$

Following the approach proposed in this dissertation,

$$P\,(\text{system fails}) = P\,(\text{system fails to start or system fails to run})$$

$$= P\,(\text{system fails to start}) + P\,(\text{system starts and system fails to run}) \qquad (1.2)$$

The last step above is based on the expression $P(EUF) = P(E) + P(\bar{E}F)$. Each term of the equation above is now evaluated separately. By direct inspection of the failure block diagram,

$$P\left(\begin{array}{c}\text{system}\\ \text{fails}\\ \text{to start}\end{array}\right) = \bar{a}\ \bar{a} = \bar{a}^2 .$$

The last term in Equation 1.2 requires a special treatment. The probability of a system failure to run depends on the condition of the system's components at the time of the demand. Therefore, it is convenient to consider all the possible combinations of component failures and successes at the time when the demand on the system occurs. The number of combinations is given by 2^n, where n is the number of components (n=2 for the system under consideration). Let S_i ($\bar{S}_i$) represent the event "component i (not) available when the demand occurs." Then, the set of all possible combinations of component failures and successes represents the set of possible states for the system. When the demand occurs, the system must be in one of the following states:

S_A	S_B	1
S_A	$\bar{S}_B$	2
$\bar{S}_A$	S_B	3
$\bar{S}_A$	$\bar{S}_B$	4

The system's states 1,2, and 3 represent states for which the system starts to operate. State 4 represents the failure to start. In order for the system to fail to run, it must first be in one of the states 1, 2, or 3. Therefore, the computation of the second term in Equation 1.2 begins by calculating the probability of the system being in one of the states 1, 2, and 3. If the system is indeed in one of these three states, it can perform properly until the end of the mission, or it can fail to run. Thus, the second step in the computation of the second term in Equation 1.2 is to calculate the probability that the system fails to run given that it is in one of the states 1, 2, and 3.

The components' failures are statistically independent. Therefore, the probability of the system being in one of the states 1, 2, and 3 equals the product of the probabilities of the components being in the failed or working states:

System's State i	Probability of the System Being in State i
1	$(1-\bar{a})^2$
2	$(1-\bar{a})\ \bar{a}$
3	$\bar{a}\ (1-\bar{a})$

If the system is in state 1, the probability of a failure to run is $\bar{r}^2$. If the system is in states 2 or 3, this probability is $\bar{r}$. Thus,

$$P\left(\begin{array}{c}\text{system}\\ \text{fails}\\ \text{to run}\end{array}\ \text{and}\ \begin{array}{c}\text{system}\\ \\ \text{starts}\end{array}\right) = (1-\bar{a})\,\bar{r}^{2} + 2\,\bar{a}\,(1-\bar{a})\,\bar{r}\quad,$$

which, when added to the probability that the system starts, gives the following result:

$$P(\text{system fails}) = \bar{a}^{2} + (1-\bar{a})\,\bar{r}^{2} + 2\bar{a}\,(1-\bar{a})\bar{r}$$

$$= (\,\bar{a} + (1-\bar{a})\bar{r}\,)^{2}\,.$$

This is the same result obtained by using the existing approach.

I.5 RELIABILITY ANALYSIS

The discipline reliability analysis models and analyzes random failure processes. System reliability analysis, a subset of reliability analysis, obtains probabilistic information about a system from probabilistic information available for the system's parts or components. System reliability analysis attempts to reduce the expected number of system failures, the expected time the system failures exist, or a combination of these.

I.5.1 Historical Remarks

The concern for reliability has existed for thousands of years. According to Goldberg,[41] "in the structural world reliability meant structures that would not collapse." Reliable structures have been designed and built for thousands of years. The pyramids in

Egypt, the cathedrals in Europe, and other buildings of the past represent the highly reliable structures built in our early days.

Several reliable devices were built before the advent of reliability formalisms or reliability analysis. The undersea telephone cables built by Bell Telephone Laboratories provide a good example. As Goldberg [41] mentions, the cables are still operating today. This high reliability, however, resulted from about twenty years of tests and experience in working with those cables.

The advent of high-technology systems in the 1940s , however, brought about the need for reliability formalisms and reliability analysis. It would take too much time and an unacceptable number of accidents if the complex systems of today were built without reliability analysis techniques. The early 1940s marked the beginning of the formal application of mathematical and statistical theories in reliability analysis.

I.5.2 Fault Tree Analysis

During the early years of reliability analysis, the systems studied were not as large and complex as they are today. Also, the techniques for system reliability analysis were just beginning to be developed. Therefore, the emphasis of reliability analysis was directed towards components and simple systems.

Fault tree analysis (FTA) originated in the early 1960s and soon became one of the principal methods of system reliability analysis. A fault tree is a logical representation of an undesirable

event or malfunction of interest, called the "top event," in terms of basic failures, called the "basic events."

In 1970, Vesely[89] presented the Kinetic Tree Theory (KTT), a major contribution in the field of system reliability analysis. The KTT forms a basis for quantitative fault tree analysis. Later works-including those by Vesely and Goldberg,[93,94,96] Fussell,[31,35,39] Montague,[62] and Dykes[20] -- have expanded the theory and practical applications of fault trees. The following paragraphs illustrate the development of fault trees.

Fault trees originated in 1961 as a technique to perform safety evaluation of the Minuteman Launch Control System. The concept was created by the Bell Telephone Laboratories.[33]

At the Safety Symposium sponsored by the University of Washington and the Boeing Company in 1965, the papers presented showed the advantages of fault tree analysis. According to Fussell, this symposium "marked the beginning of a widespread interest" in the use of fault tree analysis as a reliability tool in the nuclear industry.[33]

In 1970, Vesely[89,90] published the Kinetic Tree Theory (KTT), which forms a basis for quantitative fault tree analysis (milestone suggested by Apostolakis, Garriba, and Volta).[2]

In 1972, Fussell and Vesely[31,33] published a methodology that allows the identification of minimal cut sets of the fault trees. This milestone concerns the logical analysis of fault trees and provides qualitative and quantitative improvements in fault tree analysis (milestone suggested by Apostolakis, Garriba, and Volta).[2]

The publication of the Reactor Safety Study,[99] in 1975, contributed to the development and increased applications of fault trees. It consisted of a major study to assess the risk involved in nuclear power reactors. The study summarized and expanded the fault tree methodology. The study also developed fault trees for all the safety-related systems in a nuclear power plant .

A recent milestone (1980s) has been the implementing of national reliability and risk evaluation programs such as the Interim Reliability Evaluation Program (IREP) and the Risk Methods Integration and Evaluation Program (RMIEP). Within these and other programs, techniques have been developed to further extend the theory and the application of fault trees.

I.5.3 Comments on Data

Quantitative reliability analysis depends on the quality of the reliability data available. Fussell and Arendt[38] have summarized the problems with reliability data for nuclear power plant applications as follows:

> Reliability data are unlike other data such as nuclear cross sections. The basis of cross-section data is a large finite number of nuclei, all with identical properties. The basis of reliability data is stochastic, involving ever-changing components and environmental conditions. In addition, reliability data have the property of describing a statistical distribution rather than a fixed but unknown value. In other words, even under perfect measurement conditions, reliability data for a single event cannot be determined by a single test but rather must be determined from years of observation of equipment in various environments during which time the equipment designs may change. In addition, it is frequently easy to determine how

> many components of a certain type failed in a certain way during a known time period. Unfortunately, it is often not known how many identical components did not fail during this time interval; i.e., the population of components in use is not known. Collecting and evaluating reliability data is very frustrating.

On the other hand, a set of precise data, although desirable, may not be required in many practical applications. Major engineering insights are still available even in the case of uncertain numerical results. The Probabilistic Reactor Accident Analysis for External Initiators (earthquakes, floods, and so forth), for example, provides insights which are "by-and-large qualitative in nature, even though the results are quantitative" (Budnitz).[11] The results of the IREP can aid in simulator design, and emergency-response personnel and operating training. Also, improvements in emergency or maintenance procedures often result from studies such as IREP. In addition, quantitative measures of importance to safety for the reliability of components allows cost-benefit analyses on reliability improvements for components. These, and other applications, do not depend on the exact accident sequence probability of occurrence.

Several sources of reliability data are currently available for application to nuclear power plants. These sources emerge from several years of operating experience in nuclear power plants, coal and oil-fired power plants, and other industries. Table I.3 shows some of the well-known and commonly used data sources for nuclear components. The IEEE Std. 500-1984 is a compilation of most of the nuclear reliability data publicly available for nuclear power plants in the United States. It is the most comprehensive set of nuclear

Table I.3 Some Sources of Nuclear Component Reliability Data.

Data Sources	Remarks[a]
LERs[b] Fault/Failure rates and LERs Common Cause Fault/ Failure rates	Resulted from several studies performed at EG&G, Idaho, based on thousands of LERs.
IEEE Std. 500-1977[c]	Data based on the Delphi method, with participation of more than 200 experts in the nuclear field.
IEEE Std. 500-1984[d]	Compilation of several data banks (see text).
Modified WASH-1400	Data assembled by NRC personnel to be used in specific PRAs.
NPRDS[e]	Data bank sponsored by the utilities and operated by INPO.
IPRDS[f]	Data bank compiled from selected nuclear power plants.
Other NRC-sponsored studies or EPRI-sponsored studies[g]	

[a] This table uses the following acronyms:

LER	Licensee Event Report
IEEE	Institute of Electrical and Electronic Engineers

Table I.3 (continued)

NRC	Nuclear Regulatory Commission
PRA	Probabilistic Risk Assessment
NPRDS	Nuclear Plant Reliability Data System
INPO	Institute for Nuclear Power Operations
IPRDS	In-Plant Reliability Data System
EPRI	Electric Power Research Institute

b

Atwood, C.L. and Steverson, J.A., "Common Cause Fault Rates for Diesel Generators: Estimates Based on Licensee Event Reports at U.S. Commercial Nuclear Power Plants, 1976-1978 ," NUREG/CR-2099, EGG-EA-5359, Revision 1, 1982.

Atwood, C.L., "Common Cause Fault Rates for Pumps," NUREG/CR-2098, EGG-EA-5289, EG&G Idaho, Inc., February 1983.

Atwood, C.L. and Steverson, J.A., "Common Cause Fault Rates for Valves," NUREG/CR-2770, EGG-EA-5485, EG&G Idaho, Inc., February 1983.

Atwood, C.L. and Meachum, T.R., "Common Cause Fault Rates for Instrumentation and Control Assemblies," NUREG/CR-3289, EGG-EA-2258, Idaho National Engineering Laboratory, EG&G Idaho, Inc., May 1983.

Hubble, W.H. and Miller, C.F., "Data Summaries of Licensee Event Reports of Control Rods and Drive Mechanisms at U.S. Commercial Nuclear Power Plants," NUREG/CR-1331, EGG-EA-5079, EG&G Idaho, Inc., February 1980.

Miller, C.F. et al., "Data Summaries of Licensee Event Reports of Selected Instrumentation and Control Components at U.S. Commercial Nuclear Power Plants," NUREG/CR-1363, EGG-EA-5816, Revision 1, October 1982.

Miller, C.F. et al., "Data Summaries of Licensee Event Reports of Valves at U.S. Commercial Nuclear Power Plants from January 1, 1976 to December 31, 1980," EGG-EA-5816, April 1982.

Sams, D.W. and Trojovsky, M., "Data Summaries of Licensee Event Reports of Primary Containment Penetrations at U.S. Commercial Nuclear Power Plants," NUREG/CR-1730, EGG-EA-5188, EG&G Idaho, Inc., September 1980.

Sullivan, W.H. and Poloski, J.P., "Data Summaries of Licensee Event Reports of Pumps at U.S. Commercial Nuclear Power Plants," NUREG/CR-1205, EGG-EA-5044, EG&G Idaho, Inc., January 1980.

Table I.3 (continued)

Trojovsky, M., "Data Summaries of Licensee Event Reports of Pumps at U.S. Commercial Nuclear Power Plants," NUREG/CR-1205, Revision 1, EGG-EA-5524, January 1982.

[c] "IEEE Guide to the Collection and Presentation of Electrical, Electronic, and Sensing Component Reliability Data for Nuclear-Power Generating Stations," IEEE Std 500-1977, Institute of Electrical and Electronics Engineers, Inc., Wiley-Interscience, June 1977.

[d] "IEEE Guide to the Collection and Presentation of Electrical, Electronic, Sensing Component, and Mechanical Equipment Reliability Data for Nuclear-Power Generating Stations," IEEE Std 500-1984, Institute of Electrical and Electronics Engineers, Inc., Wiley-Interscience, 1984.

[e] "Nuclear Plant Reliability Data System (NPRDS) 1980 Annual Reports of Cumulative System and Component Reliability," NUREG/CR-2232.

[f] "In-Plant Reliability Data System (IPRDS) Interim Draft Report on the Reliability Characteristics of Selected Pumps in Four Nuclear Plants," NUREG/CR-2886, May 1982.

[g] Baranowsky, P.W., Kolaczkowski, A.M., and Fedele, M.A., "A Probabilistic Safety Analysis of DC Power Sypply Requirements for Nuclear Power Plants," NUREG-0666, April 1981.

Crooks, J.L. and Vissing, G.S., "Diesel-Generator Operating Experience at U.S. Nuclear Power Plants," USAEC, OOE-ES-002, June 1974.

"Loss of OffSite Power at Nuclear Power Plants: Data and Analysis," Electric Power Research Institute, EPRI NP-2301, March 1982.

Leverenz, F.L., Jr., Koren, J.M., Erdmann, R.C., and Lellouche, G.S., "Electric Power Research Institute, ATWS: A Reappraisal, Part II Frequency of Anticipated Transients," EPRI NP-801, June 1978.

reliability data available in the world. Table I.4 lists and briefly summarizes some of the sources for estimating human error probabilities. The THERP method, developed by Swain,[85] is the most widely used method in the nuclear power industry.

Table I.4 Data Sources and Prediction Techniques for Human Error Probabilities.

Technique/Data Bank	Remarks
American Institute for Research's Data Store (AIR's data store)[a]	Contains information about various commonly found controls and displays for each item within the data store. The information was derived out of a review of over 2,000 research reports.
Technique for Establishing Personnel Performance Standards (TEPPS)[a]	Similar to THERP but assumes optimal conditions (no performance shape factors). Uses Pair-Comparison Ranking Technique to generate data from expert opinion.
Technique for Human Error Reliability Prediction (THERP)[b]	Same philosophy of the AIR data store but uses performance shape factors and recovery factors to allow a more realistic analysis.
Siegel-Wolf Models[a,c] 1-2 man model 4-20 crew size model 20-100 large group model	These are simulation techniques. They use probability distributions as input.
Systems Analysis of Integrated Networks of Tasks (SAINT)[d]	This is a simulation language which allows great flexibility to model the man-machine system.

Table I.4 (continued)

Technique/Data Bank	Remarks
Maintenance Personnel Performance Simulation (MAPPS)[a,e]	This is a simulation technique developed for maintenance applications. It allows dynamic representation of human behavior and behavioral influences on a computer.

[a] Siegel, A.I., Bartter, W.D., Wolf, J.J., Knee, H.E., and Haas, P.M.,"Front-End Analysis for the Nuclear Power Plant Maintenance Personnel Reliability Model," NUREG/CR-2669, ORNL/TM-8300, August 1983.

[b] Swain, A.D., "Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications," NUREG/CR-1278, H.E. Guttman (Ed.), Sandia National Laboratories, Albuquerque, NM, 1980.

[c] Siegel, A.I. and Wolf, J.J., "Man-Machine Simulation Models," John Wiley & Sons, Inc., New York, 1969.

[d] Seifert, D.J. and Chubb, G.P., "SAINT: A Combined Simulation Language for Modelling Large Complex Systems," AMRL-TR-78-48, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, October 1978.

Wortman, D.B., Duket, S.D., Seifert, D.J., Hann, R.L., and Chubb, G.P., " The SAINT User's Manual," AMRL-TR-77-62, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, June 1978.

[e] Knee, H.E., Krois, P.A., Haas, P.M., Siegel, A.I., and Ryan, T.G., "Maintenance Personnel Performance Simulation (MAPPS)- A Model for Predicting Maintenance Performance Reliability in Nuclear Power Plants," 11th Water Reactor Safety Information Meeting, National Bureau of Standards, Washington, D.C., October 24-28, 1983.

CHAPTER II

A METHODOLOGY FOR THE TIME-DEPENDENT ANALYSIS OF ACCIDENT SEQUENCES

This chapter presents a methodology for the time-dependent analysis of accident sequences. It derives equations to calculate the accident sequence's expected number of occurrences in a specified time interval. This chapter also includes a brief description of the ASA computer program. This program allows the implementation of the methodology developed here. In addition, this chapter contains an example problem to illustrate the application of the methodology and the computer program.

II.1 THE ACCIDENT SEQUENCE'S EXPECTED NUMBER OF FAILURES

This section presents the derivation of an expression for the Expected Number of Failures (ENF) for a generic accident sequence. It gives the ENF as a function of the rate of failure of the initiating event and the accident sequence's time-dependent undependability.

The rate of failure of the initiating event is defined as the expected number of occurrences of the initiating event per unit time. The term "undependability" represents the total probability of occurrence of an accident sequence following the occurrence of the initiating event. Section II.2 presents a detailed definition and discussion of the undependability.

The ENF for each accident sequence is here defined as the average number of occurrences of that sequence due to occurrences of the initiating event in the specified period of time (0,t). The ENF has direct application in Probabilistic Risk Assessment (PRA). In PRA, the ENF for an accident sequence may multiply a power of the expected damage,[59] i.e., the magnitude of a consequence resulting from the occurrence of the accident sequence. This product provides a probabilistic measure of risk.

The ENF also provides a useful tool in optimization studies. An optimization study consists of changing the system structure, the maintenance procedures, or a system component in order to minimize a system parameter such as the ENF, cost, and others parameters. The ENF is a more complete optimization parameter than the unavailability or the undependability. As shown later in this section, the ENF includes the effects of the undependability and the rate of failure of the initiating event. If the rate of failure of the initiating event is a constant, then the optimization of the ENF and the optimization of the undependability lead to identical numerical results.

The time-dependent rate of failure of the initiating event, $rof_{IE}(t')$, is defined as the expected number of occurrences of the initiating event per unit time at time t'. In this way,

$$rof_{IE}(t')\,dt' = \begin{array}{l}\text{Expected number of occurrences of the initiating event in the time interval } dt' \\ \text{around } t'.\end{array}$$

Not all the occurrences of the initiating event result in the occurrence of the top event (a specific accident sequence). As already mentioned, the undependability, represented by $\bar{d}_{AS}(t',t'+T)$, is the probability that the accident sequence does occur following the existence of the initiating event. Thus, the ENF of the accident sequence in the time interval from 0 to t due to occurrences of the initiating event in the time interval dt' around t' is given by the following expression:

$$\left[rof_{IE}(t')\, dt' \right] \times \bar{d}_{AS}(t',t'+T) = \text{Contribution to the ENF of the top event in the time interval from 0 to t due to occurrences of the initiating event in the time interval dt' around t'.}$$

Integration over t' from 0 to t gives the ENF of the top event during the time interval from 0 to t:

$$ENF_{TE}(0,t) = \int_0^t rof_{IE}(t')\, \bar{d}_{AS}(t',t'+T)\, dt' \qquad (2.1)$$

The accident sequence's undependability, $\bar{d}_{AS}(t',t'+T)$, consists of the following two parts:

$$\bar{d}_{AS}(t',t'+T) = \bar{a}_{AS}(t') + \int_{t'}^{t'+T} f_{AS}(t''-t';t')\, dt'', \qquad (2.2)$$

where $\bar{a}_{AS}(t')$ represents the accident sequence's unavailability at t', and $f_{AS}(t''-t';t')\,dt''$ represents the probability that the accident sequence starts at t' and occurs for the first time in dt'' around t''. The last term on the right-hand side of the equation above includes the accident sequence unreliability.

In practical applications, $rof_{IE}(t')$ is usually directly estimated from past experience and from analytical models. The accident sequence's undependability depends on the probabilistic parameters of the components and their functional relationship. In this dissertation, fault trees model the accident sequences and provide the basic theory to evaluate these quantities. Section II.3 describes the derivation of expressions to evaluate the accident sequence undependability.

II.1.1 An Alternative Equation for the Expected Number of Failures

The $ENF_{TE}(0,t)$ can also be defined as a function of the rate of failure of the top event as follows:

$$ENF_{TE}(0,t) = \int_0^t rof_{TE}(t')\,dt' , \qquad (2.3)$$

where $rof_{TE}(t')$ is the rate of failure of the top event at time t', defined as the expected number of occurrences of the top event in the time interval dt' around t' per unit time. In general,

$$rof_{TE}(t')\,dt' \neq rof_{IE}(t')\,\bar{d}_{AS}(t',t'+T)\,dt' .$$

An expression for $rof_{TE}(t')$ is derived in the following way. Failures of the top event at t' may result from the occurrence of the initiating event at t'. In this case, the contribution to $rof_{TE}(t')$ is $rof_{IE}(t')\ \bar{a}_{AS}(t')$. Failures of the top event at t' may also result from the occurrence of the initiating event prior to t'. The following sequence must happen in this case:

1. The initiating event occurs at t", $t'-T \leq t'' \leq t'$, with rate of failure $rof_{IE}(t'')$.
2. The accident sequence starts at t" and fails to run at t', with probability $f_{AS}(t'-t";t")\ dt'$.

Thus,

$$rof_{TE}(t')\ dt' = \left[rof_{IE}(t')\ \bar{a}_{AS}(t') + \int_{t'-T}^{t'} rof_{IE}(t'')\ dt''\ f_{AS}(t'-t";t") \right] dt' \qquad (2.4)$$

Although Equations 2.3 and 2.4 provide the same final result as Equation 2.1, the former are more expensive to evaluate in a computer. Therefore, 2.1 has been selected for this dissertation.

II.2 THE ACCIDENT SEQUENCE'S TIME-DEPENDENT UNDEPENDABILITY

An accident sequence consists of a combination of successes and failures of the systems included in the event tree. The probability of a system success, P(S), approachs 1 due to the high reliability of the systems involved. Furthermore, the assumption of

the unit value for P(S) always overpredicts the expected number of failures of the accident sequence. Therefore, setting P(S)=1 results in a realistic and conservative assumption. If this is done, the problem becomes the analysis of a sequence of system failures. These system failures are defined here as $\bar{S}1,\bar{S}2,\bar{S}3,\ldots,\bar{S}n$. In general, $\bar{S}i$ represents the ith system failure (i=1,2,3,...,n). Figure II.1 further exemplifies the notation just defined.

The accident sequence analysis starts by conceptually drawing a fault tree for the accident sequence under consideration. The following top event heads the fault tree: "The accident sequence under study occurs." This fault tree appears in Figure II.2. The top event occurs if the initiating event occurs and the system failures $\bar{S}1,\bar{S}2,\bar{S}3,\ldots,\bar{S}n$ exist simultaneously in the time interval (t',t'+T). The time interval (t',t'+T) (the mission time) starts when the initiating event occurs at time t' and extends from t' to t'+T, with duration T. T represents the average time that the systems in the accident sequence must operate following the occurrence of the initiating event. The period of time of interest represents the time interval for which the ENF for the accident sequence must be computed. The period of time of interest arbitrarily starts at 0 and ends at time t. Figure II.3 illustrates the time sequence for the occurrence of an accident sequence. The top event occurs at t", $t' \leq t'' \leq t'+T$.

Occurrence of the initiating event does not necessarily result in the occurrence of the top event. For example, if any one of the n system failures $\bar{S}1,\bar{S}2,\bar{S}3,\ldots,\bar{S}n$ does not occur in (t',t'+T),

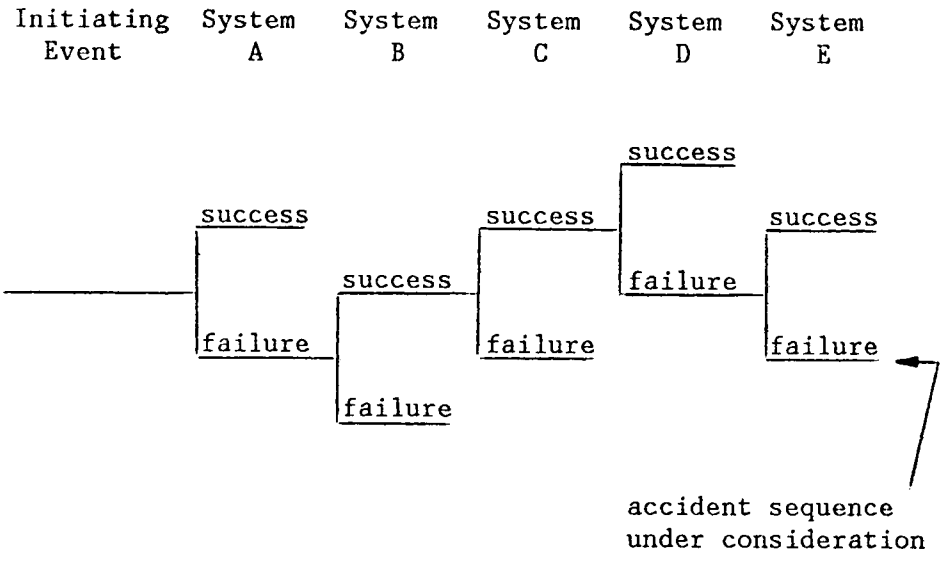


Figure II.1 A Generic Accident Sequence. For this sequence to occur, system A must fail ($\bar{S}1$), systems B and C must operate, system D must fail ($\bar{S}2$), and system E must fail ($\bar{S}3$).

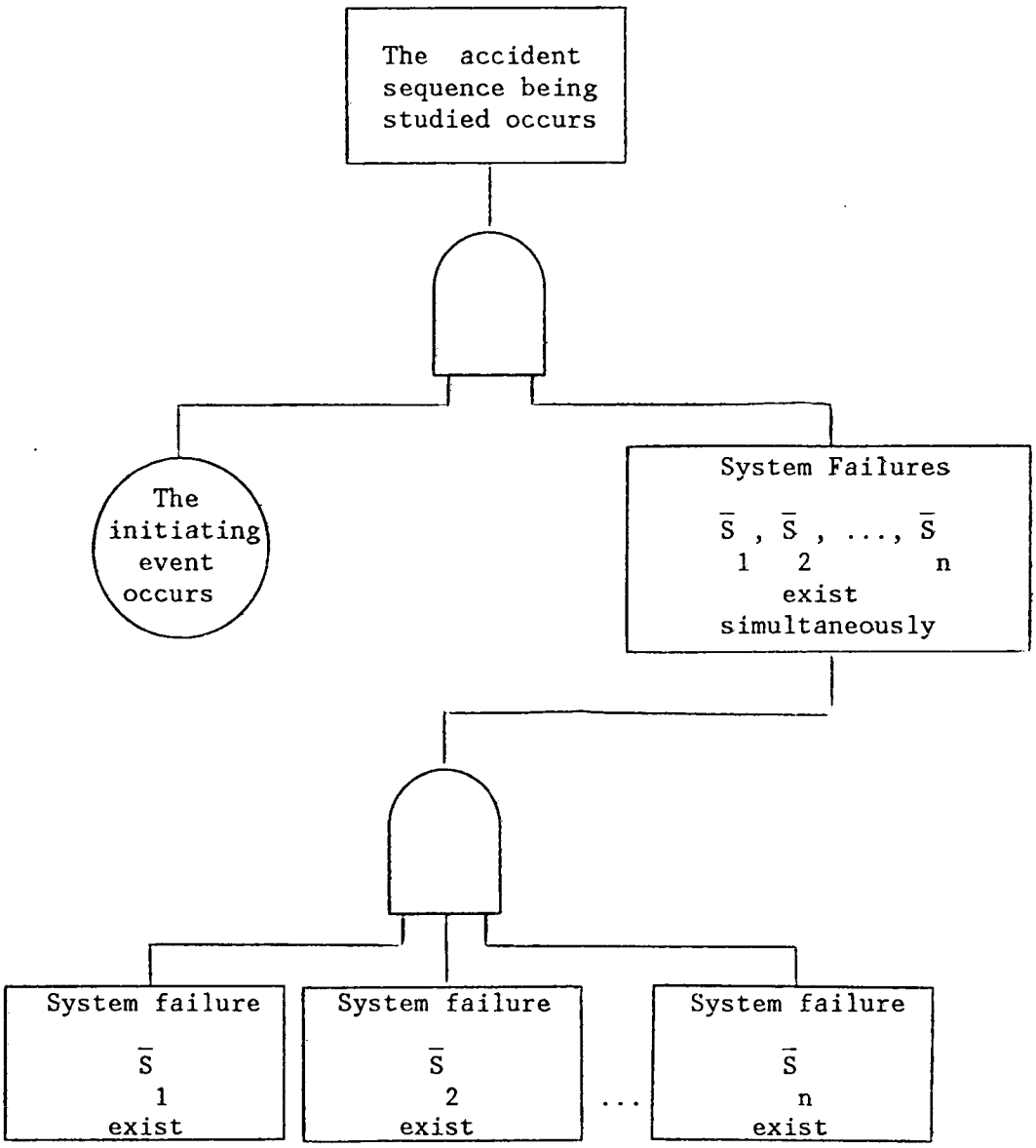


Figure II.2 Fault Tree for a Generic Accident Sequence in an Event Tree.

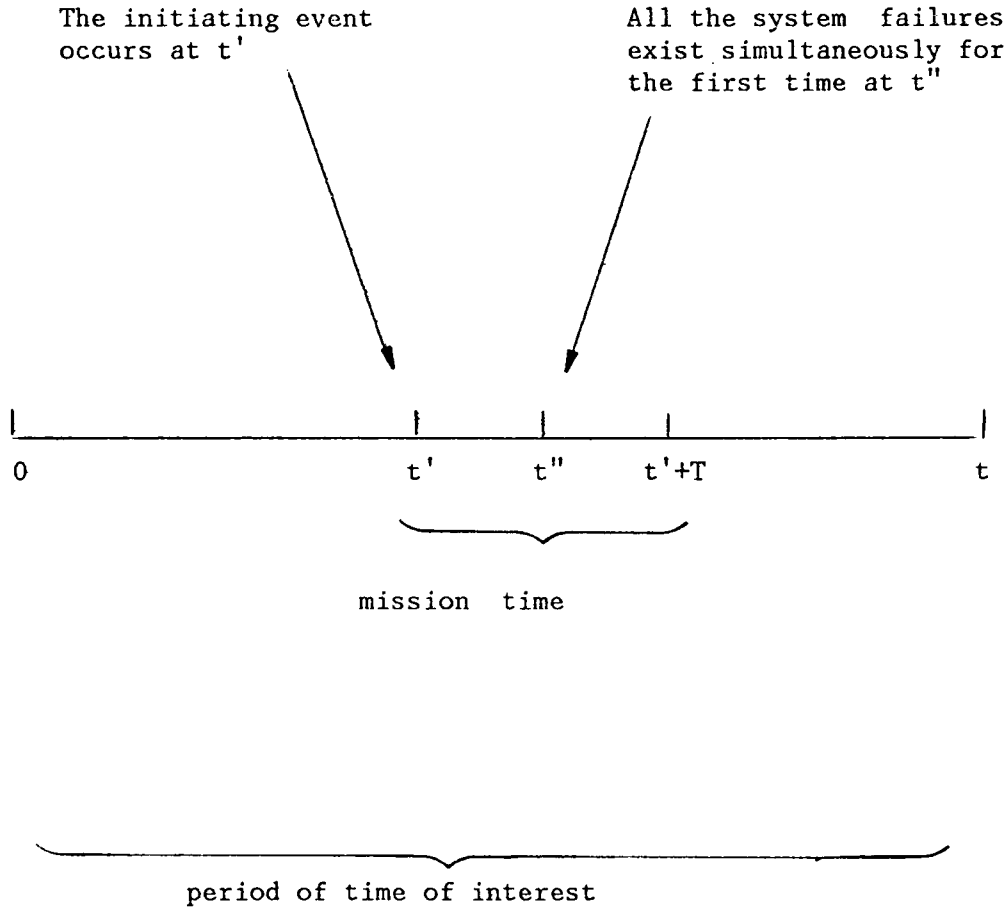


Figure II.3 A Diagram Illustrating the Time Sequence for the Occurrence of an Accident Sequence. The initiating event occurs at t'. The top event occurs at t".

the top event does not occur. Also, they may all occur but not simultaneously. The top event does not occur in this case either. In these cases, the accident sequence under consideration does not happen. Therefore, there is a probability that the accident sequence will occur following the occurrence of the initiating event.

The probability mentioned above clearly equals the probability that $\bar{S}1,\bar{S}2,\bar{S}3,\ldots,\bar{S}n$ exist simultaneously any time between t' and $t'+T$. It represents the probability that the situation described in Figure II.3 happens. It is defined as the time-dependent undependability of the accident sequence, and it is represented by $\bar{d}_{AS}(t',t'+T)$. The undependability depends on the probabilistic parameters of the systems involved and on the nature of the problem as well. The variable $d_{AS}(t',t'+T)=1-\bar{d}_{AS}(t',t'+T)$ represents the probability that the accident sequence does not occur following the existence of the initiating event. The term "dependability" labels this probability.

A definition of undependability, as presented above, includes all the system failures in the accident sequence. A similar definition applies for each system involved. The system time-dependent undependability, represented by $\bar{d}_{Si}(t',t'+T_i)$ for system i, is here defined as the probability that the system will fail in the time interval $(t',t'+T_i)$ following the occurrence of the initiating event at t'. T_i represents the average time that system i must operate following the occurrence of the initiating event.

The undependabilities defined above depend on the nature of the systems and on the particular problem. In general, they have two

components. One, here defined as the time-dependent unavailability, represents the probability of the accident sequence's (system's) failure to start. A second component represents the probability of the accident sequence's (system's) failure to run given that it has started properly. In this work, the term "unreliability" labels this component of the accident sequence (system) undependability. Fault trees, or any other reliability failure model, for the two components of the undependability may differ from each other.

A complex relationship exists between the system's undependability and the accident sequence's undependability. For example, if at least one of the systems is dependable, the accident sequence is dependable. But the accident sequence may still be dependable even if all the systems involved are undependable. This situation arises when all the system failures exist in (t',t'+T) but not simultaneously.

II.3 AN EXPRESSION FOR THE ACCIDENT SEQUENCE'S UNDEPENDABILITY

Section II.1 showed how to obtain $ENF_{TE}(0,t)$ as a function of the accident sequence undependability. This section presents the derivation of an expression to compute the accident sequence undependability as a function of the undependabilities of the system failures in the accident sequence.

As mentioned in Section II.2, a complex relationship exists between the system undependability and the accident sequence undependability. Suppose, as an example, that an accident sequence includes a Core Cooling System (CCS) failure and a Containment

Pressure Reduction System (CPRS) failure. Suppose also that a CCS failure results in core meltdown and a high pressure condition. In this example, if the CPRS fails before the CCS failure, it could be repaired and returned to work without the occurrence of the accident sequence (release of radioactive material in the environment). If the CCS failure occurs first, however, a high-pressure condition will exist. In this second case, if the CPRS fails, the accident sequence occurs (assuming a containment failure under high-pressure conditions). In the first case, an equation for the accident sequence undependability would assume that the CPRS is a repairable system. In the second case, the CPRS must be considered a nonrepairable system. The probability of the occurrence of the accident sequence must then be conditioned on the time sequence of the system's failures, which considerably increases the computational effort needed to solve the problem.

The fault trees representing the repairable system's failures may be combined through an AND gate to form a larger fault tree (the smaller trees are said to be "ANDed" together). If the larger fault tree is assumed to be nonrepairable (a conservative assumption), the complexity described in the last paragraph will no longer exist. Alternatively, the assumption of nonrepairability may be made on each of the repairable system's failures or on combinations of such system's failures. In these cases, the results are more conservative than in the case previously described. The advantage of making these assumptions on each system or groups of systems is that the number of MCSs does not grow to unmanageable sizes.

This dissertation assumes that one of the alternatives explained in the last paragraph is taken. The accident sequence then becomes a combination of nonrepairable system's failures. These system's failures are represented by $\bar{S}1$, $\bar{S}2$, ...,$\bar{S}n$. The accident sequence undependability becomes

$$\bar{d}_{AS}(t',t'+T) = P(\bar{S}1\ \bar{S}2\ \ldots\ \bar{S}n)$$

$$= P(\bar{S}1).P(\bar{S}2|\bar{S}1)\ \ldots\ P(\bar{S}n|\bar{S}1\ \bar{S}2\ \ldots\ \bar{S}n\text{-}1). \quad (2.5)$$

Dependencies among the system's failures result from the system's fault trees sharing common basic events. Two approaches are proposed here to account for these dependencies in the computation of the conditional probabilities in Equation 2.5.

First, Equation 2.5 may be conditioned on the occurrence and nonoccurrence of each of the shared basic events. This approach is essentially the approach used in the computer program FRANCALC,[23] as explained in Chapter I (see Equation 1.1 on page 17). This first approach is accurate but greatly increases the computational effort to solve the problem (if there are k common basic events, the computational effort may be increased by as much as 2^k times). Therefore, this approach should be applied only to those common basic events that have a significant effect on the system's failure probabilities (e.g., common basic events which appear in MCSs with only one basic event). For the other common basic events, a second approach is proposed as explained in the remainder of this section.

For the second approach, define $\bar{S}i^{*}$ as the event system's failure i occurs in $(t',t'+T_i)$ given the existence of all the basic events in the fault trees for $\bar{S}1$, $\bar{S}2,\ldots,\bar{S}i-1$ that also appear in the fault tree for $\bar{S}i$. The following relations result from this definition:

$$P(\bar{S}i^{*}) \geq P(\bar{S}i|\bar{S}1\ \bar{S}2\ \ldots\ \bar{S}i-1) \geq P(\bar{S}i). \qquad (2.6)$$

Therefore, upper and lower bounds for the accident sequence undependability are given by the following equations:

$$\prod_{i=1}^{n} P(\bar{S}i^{*}) \geq \bar{d}_{AS}(t',t'+T) \geq \prod_{i=1}^{n} P(\bar{S}i). \qquad (2.7)$$

The ASA computer program uses the upper bound in Equation 2.7 to provide an estimate of the accident sequence's undependability. If the fault trees representing the system's failures have no common basic events, then $P(\bar{S}i^{*})=P(\bar{S}i)$ for all i, and Equation 2.7 becomes an equality on both sides.

In computing the upper bound in Equation 2.7, the order in which the system failures are considered in the analysis becomes relevant. Although any order will give conservative estimates, the order which produces the smallest value is the most desirable one. A few simple examples were investigated in this study to determine the proper system ordering to produce the smallest upper bound in

Equation 2.7. These examples indicated that the proper ordering depends on the basic event Fussell-Vesely importance.[13] This measure of importance is defined as the probability that the basic event exists given that the top event exists. The examples were restricted to simple systems with nonrepairable basic events. Further research is necessary to provide a general approach to the proper system ordering.

Equation 2.7 provides an upper bound to the accident sequence undependability as a function of $P(\bar{S}i)^*$, which is given by

$$P(\bar{S}i)^* = \bar{a}^*_{Si}(t') + \int_{t'}^{t'+T} f^*_{Si}(t''-t';t')\, dt'', \qquad (2.8)$$

where the definitions of $\bar{a}^*_{Si}(t')$ and $f^*_{Si}(t''-t';t')\, dt''$ are the same as those used in Equation 2.2 but applied for system i. In addition, the superscript "*" indicates the condition defined for the event $\bar{S}i^*$. The next section shows how to obtain $P(\bar{S}i)^*$ as a function of the basic event unavailabilities.

II.4 AN EXPRESSION FOR THE SYSTEM UNDEPENDABILITY

The last section derived an equation relating the accident sequence undependability to the system undependabilities. This section derives an equation for the system undependabilities as a function of the basic event unavailabilities. Section II.5 shows the computation of the basic event unavailabilities from the basic data, which are assumed available.

The superscript "*" implies that a basic event in system i is unavailable if it also appears in any one of the previous systems (1, 2,...,i-1). This section assumes that this condition has already been satisfied before the analysis actually begins. In the ASA computer program, this is achieved by using a subroutine called PREPAR, which simply deletes all such basic events from the MCSs previously to any computation.

The Kinetic Tree Theory (KTT) provides an exact analytical solution to $\bar{a}^{*}_{Si}(t')$. Consider an accident sequence with 'm' MCSs. Equation 40 in reference 89 gives the following expression for $\bar{a}^{*}_{Si}(t')$:

$$\bar{a}^{*}_{Si}(t') = \sum_{r=1}^{m} (-1)^{r+1} \sum_{i_1<i_2<\ldots<i_r} \prod^{+1,\ldots,r} \bar{a}_j(t'), \qquad (2.9)$$

where:

$\bar{a}_j(t')$ is the basic event j unavailability.

$\prod^{+1,\ldots,r}$ is the product of unique basic event quantities where the basic event appears in at least one of the MCSs 1,...,r.

$\sum_{i_1<i_2<\ldots<i_r}$ is taken over all of the $\binom{m}{r}$ possible subsets of the set (1,2,...,m). This definition appears in reference 73.

For r=1, $i_r = i_1$, and the summation over all the $\binom{m}{r} = m$ possible subsets of the set (1,2,...,m) becomes the summation of all the MCSs' unavailabilities. The MCSs' unavailabilities are given by the simple product of the unavailabilities of the basic events in each MCS.

When r=2, the probabilities of two MCSs being unavailable at t' are added together. Since r+1=3 (odd), this term is subtracted from the first term. When r=3, the summation is over the MCSs taken three at a time, and so forth.

The second term on the right-hand side of Equation 2.8 involves the integral of the probability density function of the time to first system failure. An exact solution to this integral "is incalculable analytically except in exceedingly simple cases" (Murchland).[63] This integral, however, is bounded by the expected number of system failures in $(t', t'+T_i)$, given no system failure to start at t'.[63,18] As expressed by Clarotti, insofar as reliable systems are considered, no important numerical difference exists between the upper bound and the exact solution. Thus, this dissertation adopts the upper bound mentioned above as the approximation to the integral in Equation 2.8.

The remainder of this section presents the development of an equation for the computation mentioned above. The basis of this approach has been developed by Vesely.[89]

A system failure occurs in dt" if it does not exist at t" and if "r" of the system's MCSs occur in dt", $1 \leq r \leq m$. The probability that a basic event fails in an infinitesimal interval dt" is

proportional to dt". Thus, the probability that two or more basic events fail simultaneously in dt" is zero. Therefore, r MCSs may occur in dt" if and only if there exists at least one basic event that appears in all the r MCSs . Suppose that there are "l" such basic events. IF l=0, the MCSs cannot all fail simultaneously in dt". If l>0, the MCSs can fail simultaneously in dt" only under the following conditions:

1. One of the l common basic events must be available at t" and must fail in dt". This will occur with a probability of $(1-\bar{a}_i(t''))\, h_i(t'')\, dt''$,

where

i =1,2,...,l represents one of the l common, basic events,

$1-\bar{a}_i(t'')$ is the probability that the i basic event is available at t", and

$h_i(t'')$ is the conditional probability of the basic event i failure in dt", given working at t" (the basic event failure or hazard rate).

2. All the remainder basic events in the r MCSs must be unavailable at t", with probability given by

$$\prod_{j\neq i}^{+1,\ldots,r} \bar{a}_j(t'')$$

the definition of this symbol appears in Equation 2.9.

3. The system failure does not exist at time t", given the conditions in (1) and (2) above. Let P_i represent the probability of such condition.

From the above, the following expression results:

$$p\left(\begin{array}{c}\text{system failure}\\ \text{occurs in dt"}\\ \text{due to the occurrence}\\ \text{of one specific}\\ \text{combinations of r MCSs}\end{array}\right) = \sum_{i=1}^{l} P_i \, (1-\bar{a}_i(t")) \, h_i(t") \, dt" \prod_{j\neq i}^{+1,\ldots,r} \bar{a}_j(t") \,, \qquad (2.10)$$

The summation over i accounts for all the common basic events in the r MCSs . A system failure does not exist at time t" if and only if at least one of its MPSs is up at t". A MPS is up at t" if all of its basic events do not exist at t". Therefore, P_i is the probability that at least one MPS is up at t" given the conditions (1) and (2). To account for the conditions (1) and (2) in the computation of P_i, the following facts are considered:

4. Any MPS must contain at least one basic event from each MCS. Thus, any MPS must contain at least one basic event in the set of basic events which appear at least once in the r MCSs. If the MPS contains more than one such basic event, then it must be down at t" and does not contribute to P_i. This results from the conditions (1) and (2).

5. If an MPS contains only one basic event in the set of basic events which appear at least once in the r MCSs, two possibilities remain: the basic event exists at t", or the basic event does not exist at t". In the first case, the MPS again cannot contribute to P_i. Therefore, only the MPSs which contain the basic event 'i' in (1) may contribute to P_i. In addition, the MPS containing 'i' should not contain any of the other basic events which appear in the set of the r MCSs. Thus,

$$P_i = \sum_{s=1}^{g_i} (-1)^{s+1} \sum_{j_1<\ldots<j_s} \prod_{j \neq i}^{j_1,\ldots,j_s} \left(1-\bar{a}_j(t") \right), \quad (2.11)$$

where,

$\bar{a}_j(t")$ is the basic event j unavailability where basic event j appears in at least one of the MPSs $j_1,\ldots,j_s$.

g_i is the number of MPSs which contain the basic event i (as defined in item (1) above) and do not contain any other basic event in the set of basic events which appear at least once in the r MCSs.

$\sum_{j_1<j_2<\ldots<j_s}$ is taken over all the $\binom{g}{i}$ possible subsets of the set $(1,\ldots,g_i)$. The definition of this symbol appears in reference 73.

Therefore, except for the basic event unavailabilities, Equation 2.10 has been completely defined. It applies to the computation of the probability that the system failure will occur in dt" due to the occurrence of one specific combination of r MCSs. There are $\binom{m}{r}$ combinations of m MCSs taken r at a time. Then,

$$p\left(\begin{array}{c}\text{system failure occurs in dt"} \\ \text{due to the occurrence} \\ \text{of r MCSs in dt"}\end{array}\right) = \sum_{i_1<i_2<\ldots<i_r} \text{(Equation 2.10)}.$$

Finally, adding the contributions from all possible r's,

$$f_{Si}(t''-t';t')\,dt'' = \sum_{r=1}^{m} (-1)^{r+1} \sum_{i_1<\ldots<i_r} \text{(Equation 2.10)}. \qquad (2.12)$$

Thus, given the basic event unavailabilities at t' and t", the ENF for the accident sequence can be calculated using Equations 2.8, 2.9, and 2.12. The next section shows how to evaluate the basic event unavailabilities.

II.5 THE BASIC EVENT UNAVAILABILITIES

The methodology developed in this dissertation allows the analysis of the accident sequences as a function of the basic event unavailabilities as shown in the preceding sections. This section presents the basic event unavailability equations implemented in the ASA computer program. For the basic event probabilities of failure to start, the equations used here are the ones used in the FRANTIC and the FRANCALC[23,94] codes. The program includes four types of basic events:

1. Constant Unavailability (CU) basic events,

$$\bar{a}_{CU}(t') = \text{constant}, \quad 0 \leq t' \leq t, \tag{2.13}$$

which is called the "per demand unavailability" or the "cyclic unavailability."

2. Nonrepairable (NR) basic events,

$$\bar{a}_{NR}(t') = h\,t', \tag{2.14}$$

where h is the failure rate (a constant). The nonrepairable basic events can also have a cyclic unavailability contribution.

3. Constantly Monitored (CM) basic events,

$$\bar{a}_{CM}(t') = h\,T_R, \tag{2.15}$$

where h is the failure rate (a constant) and T_R represents the average detection plus repair time. The ASA program also allows the constantly monitored basic events to have a cyclic unavailability contribution.

4. Periodically Tested (PT) basic events.

This type of basic event has its unavailability given by three equations, one for each time period as follows:

4.1 Basic event undergoing test,

$$\bar{a}_{PT}(t') = p_f + (1-p_f)\, q_f + (1-p_f)(1-q_f)\, Q_o \,. \qquad (2.16)$$

4.2 Basic event undergoing repair,

$$\bar{a}_{PT}(t') = p_f + (1-p_f)\, Q_f + (1-p_f)(1-Q)\, 1/2\, h\, T_R \,. \qquad (2.17)$$

4.3 Basic event on standby,

$$\bar{a}_{PT}(t') = h\,(t_{pt} - Tau)\,, \qquad (2.18)$$

where

t_{pt} = the time from the preceding test,

T_1 = first test interval,

T_2 = test interval,

Tau = test period (duration),

T_R = repair period,

p_f = probability of a test caused failure,

q_o = probability of not overriding a test if a demand occurs, and

$Q = h_2 (T - Tau)$.

During the first test interval T_1, Equation 2.18 is modified to $\bar{a}(t') = h_1 t'$, and Q is given by hT_1. To account for periodic test inefficiencies, h_1 in the above equations is substituted by h(1-p), and hpt (the undetected contribution) is added to those equations. The periodically tested basic events may also have a cyclic unavailability contribution.

Following the occurrence of the accident sequence at t', all of the basic events have a probability of being failed or a probability of failing on demand (cyclic unavailability) as given in the equations above. In order to solve the equations in Section II.4, however, the basic event unavailabilities at time t", $t' \leq t'' \leq t'+T$, are also required. For Type I basic events,

$$\bar{a}(t'') = a(\bar{t}') = \text{constant}. \qquad (2.19)$$

For Type II basic events, the ASA computer program has two possibilities currently implemented. The basic event may be nonrepairable, with constant failure rate h, or the basic event may be repairable, with constant failure and repair rates given by h and μ, respectively. The repair rate μ is defined as the conditional probability of the basic event being repaired in dt" per unit time, given that it was failed at t".

For the repairable case, an expression for the basic event unavailability results from a probabilistic balance. The basic event is failed at $(t''+\Delta t'')$ if it was failed at t'' and was not repaired in $\Delta t''$ at t'', or if it was not failed at t'' and failed in $\Delta t''$. Mathematically speaking, with $P(t'')$ representing the unavailability at t'',

$$P(t''+\Delta t'') = P(t'') \ (1-\mu\Delta t'') + (\ 1-P(t'') \ ) \ h\Delta t'' \ .$$

Rearranging this equation,

$$\frac{P(t''+\Delta t'') - P(t'')}{\Delta t''} = - \ (h+\mu) \ P(t'') + h,$$

and taking the limit as $\Delta t''$ goes to zero,

$$\frac{dP(t'')}{dt''} = h - (h+\mu) \ P(t'') \ ,$$

which has the following solution:

$$P(t'')=\bar{a}_{CM}(t'')= \frac{h}{h+\mu}$$

$$+ \left(\bar{a}(t') - \frac{h}{h+\mu} \right) \exp(\ -(h+\mu)(t''-t') \ ). \qquad (2.20)$$

By applying the renewal theory, Caldarola[12] has developed the equation above for two limiting cases, $\bar{a}(t')=0$ and $\bar{a}(t')=1$. For these limiting cases, the equation above agrees with the previous result.

For the nonrepairable case, the basic event unavailability is obtained from equation 2.20 by simply setting μ= zero:[18,63]

$$\bar{a}_{NR}(t'') = 1 - (1-\bar{a}(t')) \exp(- h(t''-t')), \qquad (2.21)$$

which applies for constant h. For time-dependent failure rates, the definition of h should be changed to the conditional probability of failure in dt" per unit time, given no failures in the interval (t',t"-t') (rather than given working at t"). This theoretically important modification was pointed out by Murchland[63] and further clarified by Clarotti.[18]

II.6 THE ASA COMPUTER PROGRAM

The ASA computer program allows the practical application of the methodology developed in this dissertation. The code contains the main program, ten subroutines, and five function subroutines. It is written in FORTRAN and contains approximately 3,300 source statements, about 40% of which are comment lines providing adequate internal documentation of the algorithms used. The next paragraphs describe each of the following routines:

Main Program
START

PREPAR
TIMES
UNDEPE
BASIC1
BASIC2
CUTS
PATHS
RECKON
ROFIE
Functions: FNRA, FNRE, FCMA, FCME and FPTA.

Subroutine START reads and analyzes all the input data, printing part of the input data for checking purposes. Several checks are performed, and messages are given in case of errors.

Subroutine PREPAR has two functions. First, if specified by the user, it will eliminate from a fault tree all the basic events in this fault tree which also appear in any of the preceding fault trees in the accident sequence. In doing so, the program evaluates the upper bound given by Equation 2.7 (page 46). It then identifies the MCSs which represent a failure-to-start only as well as the number of such MCSs.

The main program controls the evaluation of Equation 2.1 (page 35). It calls subroutines START and PREPAR and then computes the integration intervals to numerically integrate Equation 2.8 (page 47). Subroutine TIMES is called several times by the main program to determine the times at which the integrand in Equation 2.1 experiences a discontinuity. The numerical integration of Equation 2.1 can be performed using Simpson's rule, the trapezoidal rule, or the rectangular rule, depending on the user specifications. The computation of the integrand in Equation 2.1 is accomplished by calling ROFIE, BASIC1 and UNDEPE, as described later in this section.

Subroutine TIMES, when called for the first time, will identify all the times at which the unavailability of any of the basic events in the accident sequence experiences a discontinuity. It then stores all such times points in a vector for later use. In subsequent calls, this subroutine will provide such information to the main program.

Subroutine UNDEPE controls the computation of the time-dependent undependability (Equation 2.8--page 47) for each system in the accident sequence. This is accomplished by calling CUTS and RECKON and by performing a numerical integration on the results of RECKON using Simpson's rule. CUTS calculates the probability of an accident sequence's failure to start on demand (Equation 2.9--page 48). RECKON computes the probability of an accident sequence failure-to-run at a time during the mission, given a demand and a successful start (Equation 2.12--page 53). PATHS calculates Equation 2.11 (page 52).

Subroutine BASIC1 controls the computation of the basic event unavailabilities at the time of the demand. For each basic event, it selects the proper equation from those in Section II.5. Subroutine BASIC2 controls the computation of the basic event unavailabilities during the mission, conditioned on the basic event unavailabilities at the time of the demand (computed by BASIC1). For each Type I basic event, BASIC2 selects one of the two currently available equations applicable to nonrepairable or to constantly monitored basic events (Equations 2.21 and 2.20 -- pages 58 and 57).

The subroutine ROFIE calculates the rate of failure, or the frequency of occurrence, of the initiating event as a function of time. It performs a linear interpolation between the data points provided by the user. ROFIE is not called if the initiating event has a constant rate of failure.

Several functions are available for the computation of the basic event unavailabilities. They are used in subroutines BASIC1 and BASIC2. The listing of the ASA code in Appendix D includes a detailed description of each function. These functions apply to the following Equations in Section II.5:

FNRA	Equation 2.14 (page 54)
FNRE	Equation 2.21 (page 58)
FCMA	Equation 2.15 (page 54)
FCME	Equation 2.20 (page 57)
FPTA	Equation 2.16, 2.17, and 2.18 (page 55).

Checking the ASA computer program included three steps: (1) reviews of the code, (2) runs of simple problems which could also be solved by hand calculations, and (3) runs of example problems available in the literature. Several problems were selected from the literature, including one example by Fussell;[35] four examples by Elerath, Vaurio, and Wood;[23] and sixteen examples by Vesely and Goldberg.[94] The examples problems selected from the available literature represent limiting cases for the capability of the ASA computer program. Therefore, these examples problems were used to

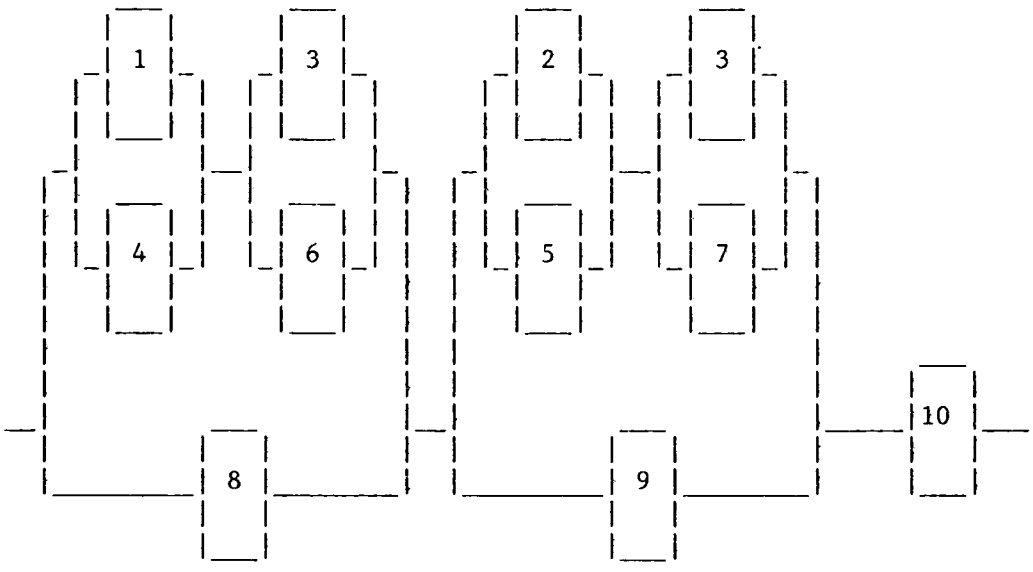
check different portions of the ASA code. For example, the agreement of the results given by ASA and the results obtained by Vesely and Goldberg provides a check of the main program; subroutines START, TIMES, UNDEPE, BASIC1, and CUTS; and some of the function subroutines. The agreement with the results presented by Fussell, on the other hand, provides checking of subroutines RECKON and BASIC2, and the function subroutine FCME. The examples by Elerath, Vaurio, and Wood required the use of most of the ASA computer program, providing a more general test of the ASA code and methodology.

Those portions of the code necessary to reproduce the examples from the literature have been comprehensively checked by the three methods mentioned above. The portions of the computer program which were not used for the examples above (less than 20% of the code) have been reviewed only. The listing of the ASA computer program in Appendix D includes comment cards which give more detailed information about the code.

II.7 APPLICATION OF THE ASA COMPUTER PROGRAM

This section presents the application of the ASA computer program to the Shutdown Heat Removal System (SHRS) analyzed by Elerath, Vaurio, and Wood.[23] Only the initiating event "One-month shutdown for refueling" is illustrated here. Figure II.4 shows the failure block diagram applicable for this accident initiator.

It is clear that the DRACS is the only independent path to successfully remove the decay heat. The other paths have at least one basic event which also appears in a different path. Thus, the



1	Diesel Generator 1
2	Diesel Generator 2
3	Offsite Power
4	Intermediate Reactor Auxiliary Cooling System 1
5	Intermediate Reactor Auxiliary Cooling System 2
6	Steam Loop 1
7	Steam Loop 2
8	Sodium System in Loop 1
9	Sodium System in Loop 2
10	Direct Reactor Auxiliary Cooling System (DRACS)

Figure II.4 Failure Block Diagram: Shutdown Heat Removal System.

accident sequence may be analyzed by assuming two major systems, the DRACS (with only one basic event) and the system formed by deleting basic event 10 from the diagram presented in Figure II.4.

The Sodium Loops (basic events 8 and 9) and the Offsite Power (basic event 3) are also reactor shutdown initiators and, therefore, must be availablle at the time of the shutdown for refueling. The input data for each basic event is shown in Appendix B for the case of staggered testing. The input to the ASA computer program appears in Appendix B, and the output appears in Appendix C.

Two cases are considered in this section. In Case 1, the basic events are assumed to be nonrepairable during the mission, which corresponds to the approach currently available in the literature. Appendixes B and C refer to this case. In Case 2, all the basic events are assumed to be constantly monitored during the mission, except for the DRACS, which is a nonrepairable system. In this second case, the same repair data given for the predemand period have been used for the postdemand period. Figure II.5 shows the time-dependent undependabilities for the two cases at the time points shown in Table II.1.

The undependabilities show large discrepancies. This discrepancies result because the probability of an accident sequence failure to run is greatly overpredicted when using the existing approach. The ASA program gives the following values for the ENF in these two cases:

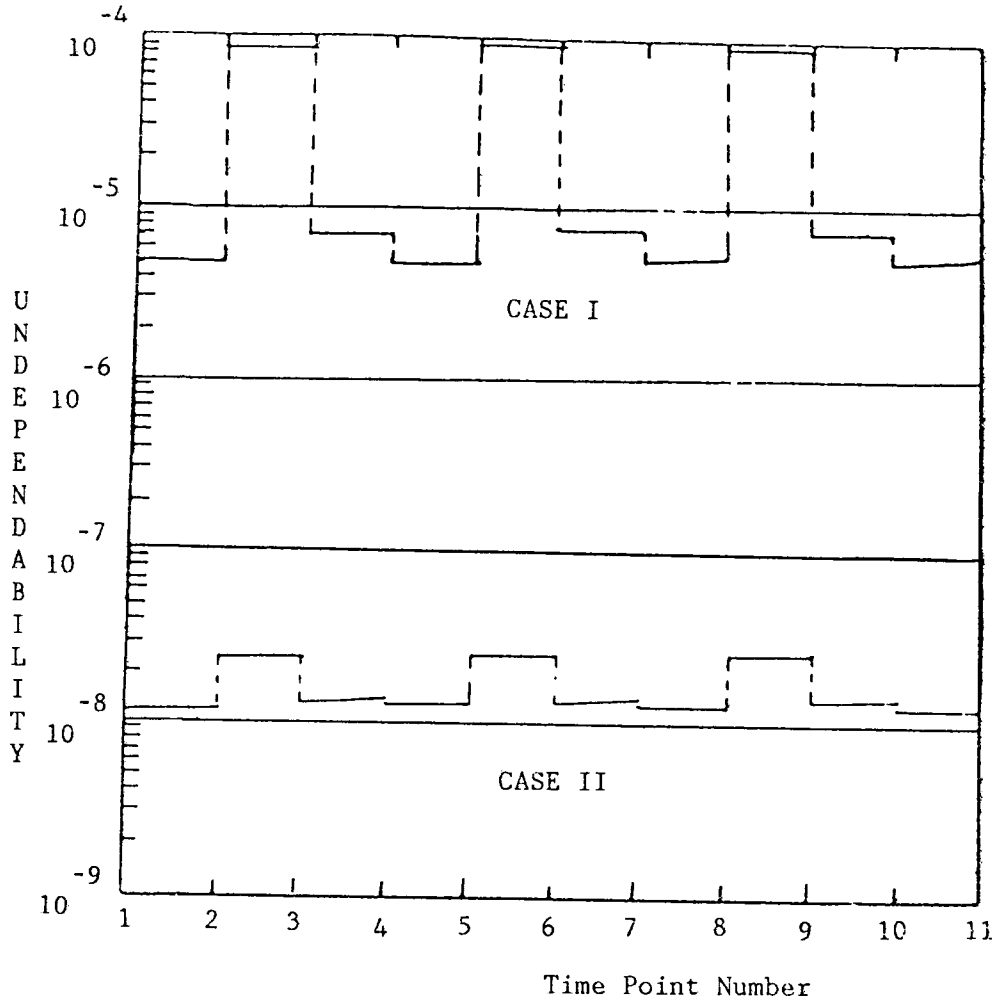


Figure II.5 Accident Sequence Undependability for Cases I and II.

Table II.1 Point Numbers Used in Figure II.5

Point Numbers	Time (hours)
1	7356
2	7665
3	7673
4	7721
5	8030
6	8038
7	8086
8	8395
9	8403
10	8451
11	8760

	CASE 1	CASE 2
Expected Number of Accident Sequence Occurrences in One Year	4.17×10^{-8}	7.67×10^{-11}

This example illustrates the necessity of the methodology developed in this dissertation. This example agrees with the comments by Fussell,[35] which state that a large overprediction results when there is a large degree of redundancy and when the basic event average repair times are small with respect to the corresponding mean times to failure.

CHAPTER III

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

This report presents the development of a methodology to calculate the Expected Number of Failures (ENF) for each accident sequence in an event tree, where fault trees are available for each branching point in the event tree. The methodology evaluates the average and the time-dependent behavior of a generic accident sequence, allowing a detailed modeling of periodic testing and repair. The analysis includes the probability of system failure to start and the probability of system failure to run following a successful start. The method developed here applies to repairable and to nonrepairable components in the accident sequence. In particular, it allows the study of systems with components that may be repaired during the mission. The following assumptions have been adopted in this work:

1. The fault trees representing the system's failures at each branching point in the event tree are coherent.[27]
2. The basic events in the fault trees are statistically independent in failure and repair.
3. The expected number of accident sequence occurrences during the mission is used to provide a conservative upper bound to the mission unreliability (the probability that the system formed by all the systems

in the accident sequence starts and fails to perform adequately during the mission).

4. The repairable systems in the accident sequence or combinations of such systems are assumed to be nonrepairable during the mission (a conservative assumption).

A computer program called ASA (an acronym for Accident Sequence Analysis) allows the practical implementation of the theory developed. It consists of a main program, ten subroutines, and five functions. Checking the ASA computer program included reviews of the code, runs of simple problems which could also be solved by hand calculations, and runs of examples problems available in the literature.

A potential problem has been identified during the course of this study. It refers to the computation of the probability of occurrence of the intersections of two or more MCSs. Two alternative approaches have been identified to solve this potential problem. One possibility consists of using disjoint product methods such as the Sigma Pi method.[19] This approach is general and exactly calculates the probability of occurrence of a system failure during the mission time, under the assumptions listed above. The second alternative consists of evaluating only the intersections of MCSs containing high-unavailability basic events with the remaining MCSs.

The work presented here is a required extension of the presently available methodology and allows a more realistic PRA of

nuclear power plants. It also expands the techniques of reliability analysis as applied to large, complex systems such as those in the nuclear and chemical industries. The following improvements are suggested for further work in this area:

1. The application of the methodology developed here to a nuclear power reactor. This application may reveal problem areas and provide guidelines for future studies.
2. The interfacing of the time-dependent methodology developed here with the existing phased-mission methodology.[39]
3. The generalization of the equations used to calculate the basic event unavailabilities.[20,87]
4. The investigation of the proper system ordering in the accident sequence that will produce the smallest upper bound to the accident sequence's ENF.
5. The investigation of the possibility of using the methodology developed here to analyze systems for which the fault trees for failure to start and failure to run have different structures.

LIST OF REFERENCES

LIST OF REFERENCES

1. Apostolakis, G.,"Probability and Risk Assessment: The Subjectivistic Viewpoint and Some Suggestions," Nuclear Safety 19, pp. 305, 1978.

2. Apostolakis, G.,Garribba, S. and Volta, G., eds.,"Synthesis and Analysis Methods for Safety and Reliability Studies," Plenum Press, New York, 1980.

3. Atwood, C.L. and Steverson, J.A., "Common Cause Fault Rates for Diesel Generators: Estimates Based on Licensee Event Reports at U.S. Commercial Nuclear Power Plants, 1976-1978 ," NUREG/CR-2099, EGG-EA-5359, Revision 1, 1982.

4. Atwood, C.L., "Common Cause Fault Rates for Pumps," NUREG/CR-2098, EGG-EA-5289, EG&G Idaho, Inc., February 1983.

5. Atwood, C.L. and Steverson, J.A., "Common Cause Fault Rates for Valves," NUREG/CR-2770, EGG-EA-5485, EG&G Idaho, Inc., February 1983.

6. Atwood, C.L. and Meachum, T.R., "Common Cause Fault Rates for Instrumentation and Control Assemblies," NUREG/CR-3289, EGG-EA-2258, Idaho National Engineering Laboratory, EG&G Idaho, Inc., May 1983.

7. Baranowsky, P.W., Kolaczkowski, A.M., and Fedele, M.A., "A Probabilistic Safety Analysis of DC Power Sypply Requirements for Nuclear Power Plants," NUREG-0666, April 1981.

8. Barlow,R.E., Fussell,J.B. and Singpurwalla, N.D.,"Reliability and Fault Tree Analysis," SIAM, Philadelphia, 1975.

9. Barlow,R.E. and Proschan,F.,"Statistical Theory of Reliability and Life Testing," Holt, Rinehant and Winston, New York, 1975.

10. Barlow, R.E. and Proschan, F., "Theory of Maintained Systems: Distribution of Time to First System Failure," Mathematics of Operations Research, Vol. 1, pp. 32-42, February 1976.

11. Budnitz, R.J., "Methodology for Probabilistic Reactor Accident Analysis for External Initiators- Earthquakes, Fire, Floods, Winds," Future Resources Associates, Inc., Berkeley, California, Submitted to "Risk Analysis."

12. Caldarola,L.,"A Method for the Calculation of the Cumulative Failure Probability Distribution of Complex Repairable Systems," Nuclear Engineering and Design, Vol. 36, pp. 109-122, 1976.

13. Campbell, D.J.,"A Procedure for Determining the Importance of Basic Events to Systems Undergoing Phased Missions," Master's Thesis, NERS78-07, Department of Nuclear Engineering, University of Tennessee, December 1978.

14. Campbell, D.J. and Ellison, B.C., "Dependent Failure Analysis Procedures Guide," JBFA-105-83, JBF Associates, Inc., Knoxville, TN, June 1983.

15. Campbell, D.J. et al., "Common Cause Failure Analysis of Selected Accident Sequences at Arkansas Nuclear One- Unit 1," JBFA-104-83, JBF Associates, Inc., Knoxville, TN, September 1983.

16. Crooks, J.L. and Vissing, G.S., "Diesel-Generator Operating Experience at U.S. Nuclear Power Plants," USAEC, OOE-ES-002, June 1974.

17. Chow, David K.,"Reliability of Some Redundant Systems with repair," IEEE Trans. Rel., Vol. R-22, No.4, pp.223-228, October 1973.

18. Clarotti,C.A., " Limitations of Minimal Cut-Set Approach in Evaluating Reliability of Systems with Repairable Components,"IEEE Transactions on Reliability, Vol.R-30, No.4, October 1981.

19. Corynen, G.C., "Evaluating the Response of Comlex Systems to Environmental Threats: The Sigma Pi Method," UCRL-53399, May 1983.

20. Dykes, A.A., "Application of Time Dependent Unavailability Analysis to Standby Safety Systems," Ph.D. Dissertation, Department of Nuclear Engineering, MIT, June 1982.

21. Edwards, G.T. and Watson, I.A., "A Study of Common-Mode Failures," Safety and Reliability Directorate, SRD R 146, U.K. Atomic Energy Authority, Wigshaw Lane, Culcheth Warrington, WA3 4NE, July 1979.

22. Elerath, J.G. and Ingram, G.E., "PROBCALC User's Manual: A Computer Program for Probability Calculation," GEFR-00408, General Electric Company, Sunnyvale, California, 1980.

23. Elerath, J.G., Vaurio, J.K., and Wood, A.P., "A Method for Reliability Analysis of Complex Reactor Systems," The ANS/ENS International Meeting on Fast Safety Technology, Lyon, France, July 1982.

24. "Loss of OffSite Power at Nuclear Power Plants: Data and Analysis," Electric Power Research Institute, EPRI NP-2301, March 1982.

25. "WAMCUT-II, A Fault Tree Evaluation Program," Electric Power Research Institute, EPRI NP-2421, June 1982.

26. "Recirculating Pump Seal Investigation," Electric Power Research Institute, EPRI NP-351, Vol. 1, December 1977.

27. Esary,J.D. and Proschan,F.,"Coherent Structures of Nonidentical Components," Technometrics, Vol.5, pp. 191-209, 1963.

28. Esary,J.D. and Marshall,A.W.,"System Structure and the Existence of a System Life," Technometrics, 6, pp. 459-462, 1964.

29. Esary, J.D. and Proschan, F., "A Reliability Bound for Systems of Maintained Interdependent Components," J. American Statistical Association, vol 65, pp. 329-338, March 1970.

30. Fleming, K.N., Mosleh, A. and Kelly, Jr., A.P., "On the Analysis of Dependent Failures in Risk Assessment and Reliability Evaluation," Nuclear Safety, Vol.24, No.5, September-October 1983.

31. Fussell, J.B. and Vesely, W.E.,"A New Methodology for Obtaining Cut Sets for Fault-Trees," Trans. American Nuclear Society, 15, 1972.

32. Fussell, J.B. ,"Synthetic Tree Model- A Formal Methodology for Fault Tree Construction," ANCR-1098, March 1973.

33. Fussell, J.B., "Fault Tree Analysis -- Concepts and Techniques," presented at the NATO Advanced Study Institute on Generic Techniques of System Reliability Assessment, Liverpool, England, July 1973.

34. Fussell,J.B., Henry,E.B. and Marshall,N.H., "MOCUS- A Computer Program to Obtain Minimal Cut Sets from Fault Trees," USAEC Rep. ANCR-1156, Aerojet Nuclear Company, August 1974.

35. Fussell,J.B., "How to Hand-Calculate System Reliability and Safety Characteristics," IEEE Transactions on Reliability, Vol.R-24, NO.3, pp. 169-174, August 1975.

36. Fussell,J.B., "Fault Tree Analysis - Concepts and Techniques," in Generic Techniques in Systems Reliability Assessment (E.J. Henley and J.N. Lynn, eds.), pp. 133-162. Noordhoff, Leyden, 1976.

37. Fussell, J.B., et al.,"SUPERPOCUS- A Computer Program for Calculating System Probabilistic Reliability and Safety Characteristics," NERS-77-01, Knoxville: Nuclear Engineering Department, May 1977.

38. Fussell,J.B. and Arendt, J.S., "System Reliability Engineering Methodology: A Discussion of the State of the Art," Nuclear Safety 20, pp.541, 1979.

39. Fussell,J.B., et. al., "Phased-Mission System Reliability Analysis. Volume1: Methodology," prepared by the University of Tennessee for the Electric Power Research Institute, EPRI NP-1945, Project 1233-2, July 1981.

40. Gately, W.Y. and Williams, R.L., "GO methodoloty- System Reliability Assessment and Computer Code Manual," Electric Power Research Institute, EPRI NP-766, May 1978.

41. Goldberg, H., "Extending the Limits of Reliability Theory," John Wiley & Sons, 1981.

42. Green, A.E. and Bourne, A.J.,"Reliability Technology," New York: John Wiley and Sons, 1972.

43. Hatch, S.W., Cybulskis, P., and Wooton, R.O., "Reactor Safety Study Methodology Applications Program: Grand Gulf #1 BWR Power Plant," NUREG/CR-1659, SAND80-1897, Sandia National Laboratories, October 1981.

44. Henley, Ernest J. and Kumamoto, H., "Reliability Engineering and Risk Assessment," Englewood Cliffs, N.J.: Prenctice-Hall, 1981.

45. Hubble, W.H. and Miller, C.F., "Data Summaries of Licensee Event Reports of Control Rods and Drive Mechanisms at U.S. Commercial Nuclear Power Plants," NUREG/CR-1331, EGG-EA-5079, EG&G Idaho, Inc., February 1980.

46. Spectrum, "A Special Issue: RELIABILITY," The Institute of Electrical and Electronics Engineers, Inc., Vol.18, No. 10, October 1981.

47. "IEEE Guide to the Collection and Presentation of Electrical, Electronic, and Sensing Component Reliability Data for Nuclear-Power Generating Stations," IEEE Std 500-1977, Institute of Electrical and Electronics Engineers, Inc., Wiley-Interscience, June 1977.

48. "IEEE Guide to the Collection and Presentation of Electrical, Electronic, Sensing Component, and Mechanical Equipment Reliability Data for Nuclear-Power Generating Stations," IEEE Std 500-1984, Institute of Electrical and Electronics Engineers, Inc., Wiley-Interscience, 1984.

49. "In-Plant Reliability Data System (IPRDS) Interim Draft Report on the Reliability Characteristics of Selected Pumps in Four Nuclear Plants," NUREG/CR-2886, May 1982.

50. Kapur,K.C. and Lamberson,L.R.,"Reliability in Engineering Design," John Wiley & Sons, Inc., 1977.

51. Knee, H.E., Krois, P.A., Haas, P.M., Siegel, A.I., and Ryan, T.G., "Maintenance Personnel Performance Simulation (MAPPS)- A Model for Predicting Maintenance Performance Reliability in Nuclear Power Plants," 11th Water Reactor Safety Information Meeting, National Bureau of Standards, Washington, D.C., October 24-28, 1983.

52. Kolb, G.J., Hatch, S.W., Cybulskis, P., and Wooton, R.O., "Reactor Safety Study Methodology Applications Program: Oconee #3 Laboratories, May 1981.

53. Kolb, G.J. et al.,"Interim Reliability Evaluation Program: Analysis of the Arkansas Nuclear One- Unit 1 Nuclear Power Plant," NUREG/CR-1987, Sandia Laboratories, Albuquerque, New Mexico, June 1982.

54. Kreyszig, Erwin,"Advanced Engineering Mathematics," John Wiley & Sons, Fourth Edition, 1979.

55. Leverenz, F.L. and Kirch, H., "User's Guide for the WAM-BAM Computer Code," Electric Power Research Institute, EPRI-217-2-5, January 1976.

56. Leverenz, F.L., Jr., Koren, J.M., Erdmann, R.C., and Lellouche, G.S., "Electric Power Research Institute, ATWS: A Reappraisal, Part II Frequency of Anticipated Transients," EPRI NP-801, June 1978.

57. Levine, S. and Vesely, W.E., "Important Event-Tree and Fault-Tree Considerations in the Reactor Safety Study," IEEE Trans. Reliability, R-25, 132, 1976.

58. Lie, C.H., Hwang, C.L. and Tillman, F.A., "Availability of Maintained Systems: A State-of-the-Art Survey," AIIE Transactions, American Institute of Industrial Engineers, Inc., Volume 9, Number 3, September 1977.

59. McCormick,N.J., "Reliability and Risk Analysis Methods and Nuclear Power Applications, "Academic Press, New York, 1981.

60. Miller, C.F. et al., "Data Summaries of Licensee Event Reports of Selected Instrumentation and Control Components at U.S. Commercial Nuclear Power Plants," NUREG/CR-1363, EGG-EA-5816, Revision 1, October 1982.

61. Miller, C.F. et al., "Data Summaries of Licensee Event Reports of Valves at U.S. Commercial Nuclear Power Plants from January 1, 1976 to December 31, 1980," EGG-EA-5816, April 1982.

62. Montague, D., "A Procedure for Determining the Expected Number of Failure of a Phased Mission," Master's Thesis, NERS-79-03, Department of Nuclear Engineering, The University of Tennessee, Knoxville, Tennessee, July 1979.

63. Murchland, J.D., "Comment on 'A Time Methodology for Fault Tree Evaluation'," Nuclear Engineering and Design, vol 22, pp 167-169, 1972.

64. Nielsen, D.S., "The Cause/Consequence Diagram Method as a Basis for Quantitative Accident Analysis," Danish Atomic Energy Commission, RISO-M-1374, May 1971.

65. Nielsen, D.S. and Runge,B., "Unreliability of a Standby System with Repair and Imperfect Switching," IEEE Transactions on Reliability, Vol. R-23, No.1, April 1974.

66. "Nonelectric Parts Reliability Data," NPRD-2, Reliability Analysis Center, Rome Air Development Center, Griffiss Air Force Base, NY, Summer 1981.

67. "Nuclear Plant Reliability Data System (NPRDS) 1980 Annual Reports of Cumulative System and Component Reliability," NUREG/CR-2232.

68. Okrent, D., "Risk-Benefit Evaluation for Large Technological Systems," Nuclear Safety, Vol. 20, No. 2, March-April 1979.

69. Pacitti, T. and Atkinson, C.P.,"Programação e Métodos Computacionais," (in Portuguese) Vols. 1 and 2, Livros Técnicos e Científicos Editora S.A., Brazil, 1977.

70. Peebles, Peyton Z.,"Probability, Random Variables, and Random Signal Principles," McGraw-Hill, Inc., 1980.

71. Platz, O. and Olsen, J.V., "FAUNET: A Program Package for Evaluation of Fault Trees and Networks," Report RISO-348, Danish Atomic Energy Commission, September 1976.

72. "PRA Procedures Guide," NRC Report NUREG/CR-2300, prepared under auspices of the American Nuclear Society and the Institute of Electrical and Electronics Engineers, April 1983.

73. Ross, Sheldon M.,"A First Course in Probability," Macmillan Publishing Co., Inc., 1976.

74. Rowe, W.D., "An Anatomy of Risk," John Wiley & Sons, Inc., New York, 1977.

75. Rust,J.H. and Weaver,L.E.,"Nuclear Power Safety," Georgia Institute of Technology Series in Nuclear Engineering, Pergamon Press Inc., pp. 303-349, 1976.

76. Sams, D.W. and Trojovsky, M., "Data Summaries of Licensee Event Reports of Primary Containment Penetrations at U.S. Commercial Nuclear Power Plants," NUREG/CR-1730, EGG-EA-5188, EG&G Idaho, Inc., September 1980.

77. Seifert, D.J. and Chubb, G.P., "SAINT: A Combined Simulation Language for Modelling Large Complex Systems," AMRL-TR-78-48, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, October 1978.

78. Shooman,M.L.,"Probabilistic Reliability: An Engineering Approach," McGraw-Hill Book Co., 1968.

79. Siegel, A.I. and Wolf, J.J., "Man-Machine Simulation Models," John Wiley & Sons, Inc., New York, 1969.

80. Siegel, A.I., Bartter, W.D., Wolf, J.J., Knee, H.E., and Haas, P.M.,"Front-End Analysis for the Nuclear Power Plant Maintenance Personnel Reliability Model," NUREG/CR-2669, ORNL/TM-8300, August 1983.

81. Singh,C.,"Reliability Calculations of Large Systems,"Procedings 1975 Annual Reliability and Maintainability Symposium.

82. Singh,C., "A Cut Set Method for Reliability Evaluation of Systems Having s-Dependent Components," IEEE Transactions on Reliability, Vol.R-29 , No.5, December 1980.

83. Smith, R.L., Westland, R.A., and Blanchard, R.E., "Technique for Establishing Personnel Performance Standards (TEPPS), Technical Manual," Report PTB-70-5, Vols. 1 and 2, Personnel Research Division, Bureau of Naval Personnel, Washington, D.C., December 1969.

84. Sullivan, W.H. and Poloski, J.P., "Data Summaries of Licensee Event Reports of Pumps at U.S. Commercial Nuclear Power Plants," NUREG/CR-1205, EGG-EA-5044, EG&G Idaho, Inc., January 1980.

85. Swain, A.D., "Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications," NUREG/CR-1278, H.E. Guttman (Ed.), Sandia National Laboratories, Albuquerque, NM, 1980.

86. Trojovsky, M., "Data Summaries of Licensee Event Reports of Pumps at U.S. Commercial Nuclear Power Plants," NUREG/CR-1205, Revision 1, EGG-EA-5524, January 1982.

87. Vaurio,J.K., "Unavailability of Components with Inspection and Repair," Nuclear Engineering and Design, Vol. 54, pp. 309-324, 1979.

88. Vaurio,J.K., " Practical Availability Analysis of Standby Systems,"Proceedings Annual Reliability and Maintainability Symposium, pp. 125-131, 1982.

89. Vesely,W.E., "A Time-Dependent Methodology for Fault Tree Evaluation." Nuclear Engineering and Design 13(1970)337-360. North-Holland Publishing Company, 1970.

90. Vesely,W.E. and Narum,R.E.,"PREP and KITT: Computer Codes for the Automatic Evaluation of a Fault Tree," USAEC Rep. IN 1349, Idaho Nuclear Corporation, August 1970.

91. Vesely, W.E., "Reply to J.D. Murchland's 'Comment on 'A Time Dependent Methodology for Fault Tree Evaluation' ' ," Nuclear Engineering and Design, vol. 22, pp 170-172, 1972.

92. Venton,A.O.F., Davidson,A.T.G. and Fraser,G.G., "The Effects of Limited Spares and Scheduled Downtime Upon the Dependability of a Simple System," IEEE Transactions on Reliability, Vol.R-25, No.5, December 1976.

93. Vesely,W.E. and Goldberg,F.F.,"Time Dependent Unavailability Analysis of Nuclear Safety Systems," IEEE Transactions on Reliability, Vol.R-26, NO.4, October 1977.

94. Vesely,W.E. and Goldberg,F.F.,"FRANTIC- A Computer Code for Time Dependent Unavailability Analysis," NUREG-0193, United States Nuclear Regulatory Commission, March 1977.

95. Vesely,W.E., Goldberg,F.F., Roberts,N.H. and Haasl,D.F.,"Fault Tree Handbook," United States Nuclear Regulatory Commission Rep. NUREG-0492, 1981.

96. Vesely,W.E., Goldberg,F.F., Powers,J.T., et al.,"FRANTIC II- A Computer Code for Time Dependent Unavailability Analysis," NUREG/CR-1924, April 1981.

97. Vlajcic, Paul, "Building a Network of Safety," Nuclear News, March 1984.

98. Wagner, David P.,"A Procedure for Qualitative Common Cause Failure Analysis of Complex Systems," Master Thesis, Department of Nuclear Engineering, The University of Tennessee, Knoxville, Tennessee, 1977.

99. "Reactor Safety Study , An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants," United States Nuclear Regulatory Commission, NUREG-75/014, 1975.

100. Winkler, R.L., "An Introduction to Bayesian Inference and Decision," Holt, Rinehart and Winston, Inc., 1972.

101. Wortman, D.B., Duket, S.D., Seifert, D.J., Hann, R.L., and Chubb, G.P., " The SAINT User's Manual," AMRL-TR-77-62, Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio, June 1978.

102. Worrell, R.B. and Stack, D.W., "A SETS User's Manual for the Fault Tree Analyst," SAND77-2051, Sandia National Laboratories, November 1978.

APPENDIXES

APPENDIX A

INPUT TO THE ASA COMPUTER PROGRAM

INPUT DESCRIPTION FOR THE ASA COMPUTER PROGRAM

The ASA computer program, written in FORTRAN IV, determines the expected number of failures (or occurrences) for an accident sequence in an event tree. It allows the implementation of the methodology developed in Chapter II. This Appendix describes the input to the ASA computer program.

The user must supply failure, repair, and maintenance data for each basic event. The MCSs and the MPSs (optional) must be given for each system. Information concerning the initiating event must also be provided. In addition, the program requires parameters such as the time period of interest, the maximum integration interval, the mission duration, and so forth. The program does not provide non-zero default values.

The input data to ASA consist of seven sets of data cards called "data groups." The first card in each data group is a keyword identification card. The last card in each data group usually contains the number "-1" to indicate the end of the data group. The program requires all the data groups to run. In addition, the data groups must appear in the same order in which they are presented in this Appendix. Error messages will indicate a missing or displaced data group. The next sections describe each data group. The following definitions are used in the next sections:

The System Identification Number (SIN).	This is a unique number (from one to the number of systems in the accident sequence) assigned to each

	system. The same SIN must be used for all the input data concerning the same system. The program will consider the system with SIN equal to one first, and SIN equal to two second, and so on. This order is important when subroutine PREPAR is used to delete some of the basic events which are common to two or more systems.
The Basic Event Identification Number (BEIN).	This is a unique number (from one to the number of basic events in the accident sequence) assigned to each basic event. The same BEIN must be used for all the input data concerning the same basic event.

Data Groups

A.1 "TITLE" Data Group

This data group specifies the title for each run. It contains the following three cards:

1. First card:

Contents.....	The keyword "TIT."
Format.....	1X,A3.

2. Second card:

Contents.....	Identification title.
Format	Any alphanumeric informmation describing the title for the to 80 characters.

3. Third card:

Contents..... The number "-1."

Format..... I2.

A.2 "SYSTEM" Data Group

This data group indicates to the program how the systems relate to the basic events by providing the MCSs and, if desired, the MPSs.

1. First card:

Contents..... The keyword "SYS."

Format..... 1X,A3.

2. Second card:

Contents..... The number of systems in the accident sequence.

Format..... I2.

3. The following cards must be provided for each system:

(3.1) Contents..... The SIN and the number of MCSs in the system.

Format..... I2,I3.

(3.2) The following card must be provided for each MCS in this system:

Contents..... The BEIN for each basic event in the MCS.

Format..... 10I5.

(3.3) After completing (3.2) for all MCSs, repeat (3) for the next system.

4. After providing the cards described in (3) for all the systems in the accident sequence, the following card should appear:

Contents..... The number "-1."

Format..... I2.

5. This card indicates whether or not the MPSs are provided with the input data. If the input data contain the MPSs, the program may use (depending on the specifications in A.7) this information to compute the probabilities of the systems being up at the required times. If the MPSs are not provided, the program automatically assumes that this probability is 1.

If a number other than 1 is provided, the program assumes that the input data do not contain the MPSs. In this case, item (6) below does not apply, and the next data group should be provided. If the number 1 is inputed, proceed as indicated in (6). The format for this card is I2.

6. The following cards must be provided for each system:

6.1 Contents..... The SIN and the number of MPSs in the system.

Format..... I2,I3.

6.2 The following cards must be provided for each MPS in the system:

6.2.1 Contents..... The BEIN of the first 10 basic events.

Format..... 10I5.

6.2.2 Contents..... The BEIN of the next 10 basic events in the MPS.

Format..... 10I5.

6.2.3 Item (6.2.2) should be repeated until all the basic events in the MPS have been provided. Then, the following card should appear:

Contents..... The number "-2."

Format..... I2.

6.3 Contents..... The number "-1."

Format..... I2.

A.3 "BASIC EVENT" Data Group

This data group provides information concerning all the basic events in the accident sequence. Any basic event referred to in the MCSs should be properly defined here. The variables and parameters considered in this data group are the following:

NBE	The total number of basic events in the accident sequence.
Basic Event Type	=1 a basic event which represents a component which may fail to start or run. =2 a basic event which represents a component's failure to start only.

Basic Event Pre-demand class	=1 constant unavailability. =2 nonrepairable. =3 constantly monitored. =4 periodically tested.
Basic Event Post-demand class	=2 nonrepairable. =3 constantly monitored.

1. First card:

Contents..... The keyword "BAS."

Format..... 1X,A3.

2. Second card:

Contents..... NBE.

Format..... I5.

3. The following cards should be provided for each basic event in the accident sequence:

3.1 Contents..... BEIN.

Format..... I5.

3.2 Contents..... The basic event type, the hazard rate, the periodic test interval, the first test interval, the test period (duration), and the repair period (duration).

Format..... I5,5E10.5.

3.3 Contents..... The basic event predemand class, the test override unavailability, the probability of a test-caused failure, the detection inefficiency, the undetected hazard rate, and the unavailability per demand.

Format..... I5,5E10.5.

3.4 If the basic event type is not 1, skip this item. Otherwise, the following card should appear:

Contents..... The basic event postdemand class, the postdemand hazard rate, and the postdemand repair time.

Format..... I5,2E10.5.

4. Contents..... The number "-1."

Format..... I2.

A.4 "TIME" Data Group

This data group consists of five cards. The "period of time of interest" is the period of time for which the ENF is to be computed. The "maximum time interval" (to apply the rectangular and the trapezoidal rules) refers to the numerical integration of Equation 2.1 in Chapter II. The "mission duration" is the average time that a system has to run following the occurrence of the initiating event. The "maximum integration interval" applies to the numerical integration of Equation 2.8 in Chapter II.

1. First card:

Contents..... The keyword "TIM."

Format..... 1X,A3.

2. Second card:

Contents..... The period of time of interest, the maximum time interval to apply the rectangular rule, and the maximum time interval to apply the trapezoidal rule.

Format..... 3F8.2.

3. Third card:

Contents..... The mission duration for systems 1, 2, and 3.

Format..... 3F8.2.

4. Fourth card:

Contents..... The maximum integration interval for systems 1, 2, and 3.

Format..... F8.2.

5. Fifth card:

Contents..... The number "-1."

Format..... I2.

A.5 "PRINT" Data Group

Optional output information is not currently available. This data group will be further developed in the future. The following two cards must appear in the input data:

1. Contents..... The keyword "PRI."

   Format..... 1X,A3.

2. Contents..... The number "-1."

Format..... I2.

A.6 "INITIATING EVENT" Data Group

This data group allows the user to specify the information concerning the initiating event. Two alternatives are currently available. The initiating event may have a constant failure (or occurrence) rate or a time-dependent failure rate. In the second case, the failure rate is specified by providing its values at user-defined times. A linear interpolation is automatically performed to compute the failure rate at times for which the failure rate was not specified. The following cards are requested in this data group:

1. Contents..... The keyword "INI."

   Format..... 1X,A3.

2. Contents..... The number "1," if (constant failure rate) or the number "2" (if time-dependent failure rate).

   Format..... I2.

3. If constant failure rate:

   Contents..... The failure rate in 1/hr.

   Format..... E10.5.

If the rate of failure is time-dependent, the user may specify the failure rate at up to 100 time points. Thus, the

cards that follow may be repeated up to 100 times. The time points must be in an increasing order, starting at time zero. The last time point must be at the time corresponding to the end of the period of interest (specified in A.4, item 2).

Contents..... Time and the initiating event rate of failure at this time.

Format..... F10.2,E10.5.

4. Contents..... The number "-1."

Format..... I2.

A.7 "RUN" Data Group

This data group allows the user to control the execution of the ASA computer program. It currently consists of four cards as follows:

1. Contents..... The keyword "RUN."

Format..... 1X,A3.

2. Contents..... The following answers shoud be given (a 1 means YES):

Compute the intersections of the MCSs when evaluating the probabilities of a system failure to start?

Compute the probability that the system is "up" when evaluating the probability of a system failure to run?

Compute the intersection of the MCSs when evaluating the probability of a system failure to run?

		Compute the upper bound in Equation 2.7 (Chapter II, page 48)? Compute the probability that the system fails to start?
	Format.....	10I5.
3.	Contents.....	Maximum overprediction allowed. For values below or equal to this bound, the program will not compute the intersection of MCSs taken three at a time.
	Format.....	F6.5.
4.	Contents.....	The number "-1."
	Format.....	I2.

APPENDIX B

EXAMPLE OF AN INPUT TO THE ASA COMPUTER PROGRAM

```
 TITLE DATA GROUP
730-HOUR SHUTDOWN ( REFUELING ): ASSUMING TWO SYSTEMS - STAGGERED
-1
 SYSTEM DATA GROUP
02
01017
     1     2     3
     1     3     5
     1     3     9
     1     2     6     7
     1     5     6     7
     1     6     9
     2     3     4
     3     4     5
     3     4     9
     2     4     6     7
     4     5     6     7
     4     6     9
     2     3     8
     2     7     8
     3     5     8
     5     7     8
     8     9
02001
    10
-1
 1          ;THE MPS'S ARE PROVIDED IN THE INPUT DATA.
01004
     1     4     8
-2
     3     6     8
-2
     2     5     9
-2
     3     7     9
-2
-1
02001
    10
-2
-1
 BASIC EVENT DATA GROUP
00010
     1                                        ;DIESEL GENERATOR 1
    1.10000E-05.73000E+03.36500E+03.80000E+01.48000E+02  ;TYPE 1
    4.50000E+00.01000E+00.00000E+00.00000E+00.10000E-02  ; PT
    2.10000E-05                                          ; NR
     2                                       ;DIESEL GENERATOR 2
    1.10000E-05.73000E+03.00000E+00.80000E+01.48000E+02  ;TYPE 1
```

```
     4.50000E+00.01000E+00.00000E+00.00000E+00.10000E-02  ; PT
     2.10000E-05                                          ; NR
     3                                  ;OFFSITE POWER
     1.10000E-03                                .48000E+02  ;TYPE 1
     1                                          .00000E+00  ; CU
     2.10000E-03                                            ; NR
     4                              ;IRACS 1
     1.10000E-06.73000E+03.36500E+03.80000E+01.48000E+02  ;TYPE 1
     4.50000E+00.01000E+00.05000E+00.00000E+00.00000E+00  ; PT
     2.10000E-06                                            ; NR
     5                              ;IRACS 2
     1.10000E-06.73000E+03.00000E+00.80000E+01.48000E+02  ;TYPE 1
     4.50000E+00.01000E+00.05000E+00.00000E+00.00000E+00  ; PT
     2.10000E-06                                            ; NR
     6                                  ;LOOP 1- STEAM
     1.10000E-03                                .12000E+03  ;TYPE 1
     3                                                      ; CM
     2.10000E-03                                            ; NR
     7                                  ;LOOP 2- STEAM
     1.10000E-03                                .12000E+03  ;TYPE 1
     3                                                      ; CM
     2.10000E-03                                            ; NR
     8                                  ;LOOP 1- SODIUM
     1.10000E-04                                .24000E+03  ;TYPE 1
     1                                          .00000E+00  ; CU
     2.10000E-04                                            ; NR
     9                                  ;LOOP 2- SODIUM
     1.10000E-04                                .24000E+03  ;TYPE 1
     1                                          .00000E+00  ; CU
     2.10000E-04                                            ; NR
    10                                  ;DRACS
     1.10000E-06                                            ;TYPE 1
     2                                                      ; NR
     2.10000E-06                                            ; NR
-1
 TIME DATA GROUP
08760.0000000.0108760.00
00730.0000730.00
00700.0000700.00
-1
 PRINT DATA GROUP
01
-1
 INITIATING EVENT DATA GROUP
01
.11416E-03
-1
 RUN DATA GROUP
    1    2    1    2    1
.01000
-1
```

APPENDIX C

EXAMPLE OF AN OUTPUT OF THE ASA COMPUTER PROGRAM

TIME (HOURS)	ACCIDENT SEQUENCE UNDEPENDABILITY (U)	RATE OF FAILURE OF THE INITIATING EVENT (ROF) IN (1/HR)	U X ROF
0.000	.7186E-07	.1142E-03	.8204E-11
8.000	.7269E-07	.1142E-03	.8298E-11
8.000	.6011E-08	.1142E-03	.6862E-12
56.000	.6414E-08	.1142E-03	.7322E-12
56.000	.4496E-08	.1142E-03	.5132E-12
365.000	.6355E-08	.1142E-03	.7255E-12
365.000	.1103E-06	.1142E-03	.1260E-10
373.000	.1112E-06	.1142E-03	.1269E-10
373.000	.9157E-08	.1142E-03	.1045E-11
421.000	.9573E-08	.1142E-03	.1093E-11
421.000	.6639E-08	.1142E-03	.7579E-12
730.000	.8546E-08	.1142E-03	.9756E-12
730.000	.1472E-06	.1142E-03	.1680E-10
738.000	.1480E-06	.1142E-03	.1690E-10
738.000	.1226E-07	.1142E-03	.1400E-11
786.000	.1268E-07	.1142E-03	.1448E-11
786.000	.8745E-08	.1142E-03	.9983E-12
1095.000	.1068E-07	.1142E-03	.1220E-11

1095.000	.1840E-06	.1142E-03	.2100E-10
1103.000	.1849E-06	.1142E-03	.2110E-10
1103.000	.1531E-07	.1142E-03	.1748E-11
1151.000	.1574E-07	.1142E-03	.1797E-11
1151.000	.1085E-07	.1142E-03	.1239E-11
1460.000	.1282E-07	.1142E-03	.1464E-11
1460.000	.2208E-06	.1142E-03	.2520E-10
1468.000	.2217E-06	.1142E-03	.2531E-10
1468.000	.1836E-07	.1142E-03	.2096E-11
1516.000	.1879E-07	.1142E-03	.2145E-11
1516.000	.1296E-07	.1142E-03	.1479E-11
1825.000	.1496E-07	.1142E-03	.1708E-11
1825.000	.2576E-06	.1142E-03	.2941E-10
1833.000	.2585E-06	.1142E-03	.2951E-10
1833.000	.2141E-07	.1142E-03	.2444E-11
1881.000	.2185E-07	.1142E-03	.2494E-11
1881.000	.1507E-07	.1142E-03	.1720E-11
2190.000	.1710E-07	.1142E-03	.1952E-11
2190.000	.2944E-06	.1142E-03	.3361E-10
2198.000	.2954E-06	.1142E-03	.3372E-10
2198.000	.2446E-07	.1142E-03	.2792E-11
2246.000	.2490E-07	.1142E-03	.2843E-11
2246.000	.1717E-07	.1142E-03	.1960E-11
2555.000	.1924E-07	.1142E-03	.2196E-11
2555.000	.3313E-06	.1142E-03	.3782E-10

2563.000	.3322E-06	.1142E-03	.3793E-10
2563.000	.2751E-07	.1142E-03	.3141E-11
2611.000	.2796E-07	.1142E-03	.3192E-11
2611.000	.1928E-07	.1142E-03	.2201E-11
2920.000	.2137E-07	.1142E-03	.2440E-11
2920.000	.3681E-06	.1142E-03	.4202E-10
2928.000	.3691E-06	.1142E-03	.4214E-10
2928.000	.3056E-07	.1142E-03	.3489E-11
2976.000	.3102E-07	.1142E-03	.3541E-11
2976.000	.2139E-07	.1142E-03	.2442E-11
3285.000	.2351E-07	.1142E-03	.2684E-11
3285.000	.4050E-06	.1142E-03	.4623E-10
3293.000	.4060E-06	.1142E-03	.4635E-10
3293.000	.3362E-07	.1142E-03	.3838E-11
3341.000	.3408E-07	.1142E-03	.3890E-11
3341.000	.2350E-07	.1142E-03	.2682E-11
3650.000	.2565E-07	.1142E-03	.2929E-11
3650.000	.4418E-06	.1142E-03	.5044E-10
3658.000	.4429E-06	.1142E-03	.5056E-10
3658.000	.3667E-07	.1142E-03	.4186E-11
3706.000	.3714E-07	.1142E-03	.4239E-11
3706.000	.2560E-07	.1142E-03	.2923E-11
4015.000	.2779E-07	.1142E-03	.3173E-11
4015.000	.4787E-06	.1142E-03	.5465E-10
4023.000	.4798E-06	.1142E-03	.5477E-10

4023.000	.3972E-07	.1142E-03	.4535E-11
4071.000	.4020E-07	.1142E-03	.4589E-11
4071.000	.2771E-07	.1142E-03	.3164E-11
4380.000	.2993E-07	.1142E-03	.3417E-11
4380.000	.5156E-06	.1142E-03	.5886E-10
4388.000	.5166E-06	.1142E-03	.5898E-10
4388.000	.4278E-07	.1142E-03	.4883E-11
4436.000	.4325E-07	.1142E-03	.4938E-11
4436.000	.2982E-07	.1142E-03	.3405E-11
4745.000	.3207E-07	.1142E-03	.3662E-11
4745.000	.5525E-06	.1142E-03	.6307E-10
4753.000	.5536E-06	.1142E-03	.6319E-10
4753.000	.4583E-07	.1142E-03	.5232E-11
4801.000	.4632E-07	.1142E-03	.5287E-11
4801.000	.3193E-07	.1142E-03	.3645E-11
5110.000	.3422E-07	.1142E-03	.3906E-11
5110.000	.5894E-06	.1142E-03	.6728E-10
5118.000	.5905E-06	.1142E-03	.6741E-10
5118.000	.4889E-07	.1142E-03	.5581E-11
5166.000	.4938E-07	.1142E-03	.5637E-11
5166.000	.3404E-07	.1142E-03	.3886E-11
5475.000	.3636E-07	.1142E-03	.4151E-11
5475.000	.6263E-06	.1142E-03	.7149E-10
5483.000	.6274E-06	.1142E-03	.7162E-10
5483.000	.5194E-07	.1142E-03	.5930E-11

5531.000	.5244E-07	.1142E-03	.5986E-11
5531.000	.3615E-07	.1142E-03	.4127E-11
5840.000	.3850E-07	.1142E-03	.4395E-11
5840.000	.6632E-06	.1142E-03	.7571E-10
5848.000	.6643E-06	.1142E-03	.7584E-10
5848.000	.5500E-07	.1142E-03	.6279E-11
5896.000	.5550E-07	.1142E-03	.6336E-11
5896.000	.3826E-07	.1142E-03	.4368E-11
6205.000	.4064E-07	.1142E-03	.4640E-11
6205.000	.7001E-06	.1142E-03	.7992E-10
6213.000	.7013E-06	.1142E-03	.8005E-10
6213.000	.5806E-07	.1142E-03	.6628E-11
6261.000	.5856E-07	.1142E-03	.6685E-11
6261.000	.4037E-07	.1142E-03	.4609E-11
6570.000	.4278E-07	.1142E-03	.4884E-11
6570.000	.7370E-06	.1142E-03	.8414E-10
6578.000	.7382E-06	.1142E-03	.8427E-10
6578.000	.6111E-07	.1142E-03	.6977E-11
6626.000	.6162E-07	.1142E-03	.7035E-11
6626.000	.4249E-07	.1142E-03	.4850E-11
6935.000	.4493E-07	.1142E-03	.5129E-11
6935.000	.7739E-06	.1142E-03	.8835E-10
6943.000	.7751E-06	.1142E-03	.8849E-10
6943.000	.6417E-07	.1142E-03	.7326E-11
6991.000	.6469E-07	.1142E-03	.7385E-11

6991.000	.4460E-07	.1142E-03	.5091E-11
7300.000	.4707E-07	.1142E-03	.5373E-11
7300.000	.8109E-06	.1142E-03	.9257E-10
7308.000	.8121E-06	.1142E-03	.9271E-10
7308.000	.6723E-07	.1142E-03	.7675E-11
7356.000	.6775E-07	.1142E-03	.7735E-11
7356.000	.4671E-07	.1142E-03	.5332E-11
7665.000	.4921E-07	.1142E-03	.5618E-11
7665.000	.8478E-06	.1142E-03	.9679E-10
7673.000	.8491E-06	.1142E-03	.9693E-10
7673.000	.7029E-07	.1142E-03	.8024E-11
7721.000	.7082E-07	.1142E-03	.8084E-11
7721.000	.4882E-07	.1142E-03	.5573E-11
8030.000	.5136E-07	.1142E-03	.5863E-11
8030.000	.8848E-06	.1142E-03	.1010E-09
8038.000	.8861E-06	.1142E-03	.1012E-09
8038.000	.7335E-07	.1142E-03	.8373E-11
8086.000	.7388E-07	.1142E-03	.8434E-11
8086.000	.5093E-07	.1142E-03	.5815E-11
8395.000	.5350E-07	.1142E-03	.6108E-11
8395.000	.9218E-06	.1142E-03	.1052E-09
8403.000	.9230E-06	.1142E-03	.1054E-09
8403.000	.7641E-07	.1142E-03	.8723E-11
8451.000	.7695E-07	.1142E-03	.8784E-11
8451.000	.5305E-07	.1142E-03	.6056E-11
8760.000	.5564E-07	.1142E-03	.6352E-11

```
------------------------------------------------------------------------

THE SYSTEM EXPECTED NUMBER OF FAILURES IS .416748E-07

------------------------------------------------------------------------
```

APPENDIX D

THE LISTING OF THE ASA COMPUTER PROGRAM

```
C ASASASASASASASASASAS    ASASASASASASASASASAS    ASASASASASASASASASAS
C SASASASASASASASASASA    SASASASASASASASASASA    SASASASASASASASASASA
C ASASASASASASASASASAS    ASASASASASASASASASAS    ASASASASASASASASASAS
C SASASASASASASASASASA    SASASASASASASASASASA    SASASASASASASASASASA
C ASASASASASASASASASAS    ASASASASASASASASASAS    ASASASASASASASASASAS
C SASASASASASASASASASA    SASASASASASASASASASA    SASASASASASASASASASA
C ASASASASASASASASASAS    ASASASASASASASASASAS    ASASASASASASASASASAS
C SASASASASASASASASASA    SASASASASASASASASASA    SASASASASASASASASASA
C ASASASASASASASASASAS    ASASA                   ASASASASASASASASASAS
C SASASASASASASASASASA    SASAS                   SASASASASASASASASASA
C ASASA          SASAS    ASASA                   ASASA          SASAS
C SASAS          ASASA    SASAS                   SASAS          ASASA
C ASASA          SASAS    ASASA                   ASASA          SASAS
C SASAS          ASASA    SASAS                   SASAS          ASASA
C ASASA          SASAS    ASASASASASASASASASAS    ASASA          SASAS
C SASAS          ASASA    SASASASASASASASASASA    SASAS          ASASA
C ASASASASASASASASASAS    ASASASASASASASASASAS    ASASASASASASASASASAS
C SASASASASASASASASASA    SASASASASASASASASASA    SASASASASASASASASASA
C ASASASASASASASASASAS    ASASASASASASASASASAS    ASASASASASASASASASAS
C SASASASASASASASASASA    SASASASASASASASASASA    SASASASASASASASASASA
C ASASASASASASASASASAS                   SASAS    ASASASASASASASASASAS
C SASASASASASASASASASA                   ASASA    SASASASASASASASASASA
C ASASA          SASAS                   SASAS    ASASA          SASAS
C SASAS          ASASA                   ASASA    SASAS          ASASA
C ASASA          SASAS                   SASAS    ASASA          SASAS
C SASAS          ASASA                   ASASA    SASAS          ASASA
C ASASA          SASAS    ASASASASASASASASASAS    ASASA          SASAS
C SASAS          ASASA    SASASASASASASASASASA    SASAS          ASASA
C ASASA          SASAS    ASASASASASASASASASAS    ASASA          SASAS
C SASAS          ASASA    SASASASASASASASASASA    SASAS          ASASA
C ASASA          SASAS    ASASASASASASASASASAS    ASASA          SASAS
C SASAS          ASASA    SASASASASASASASASASA    SASAS          ASASA
C ASASA          SASAS    ASASASASASASASASASAS    ASASA          SASAS
C SASAS          ASASA    SASASASASASASASASASA    SASAS          ASASA
C
C   *****************************************************************
C   *                                                               *
C   *    TITLE:          ASA (AN ACRONYM FOR ACCIDENT SEQUENCE ANALYSIS)*
C   *                                                               *
C   *    PURPOSE:        THIS PROGRAM COMPUTES THE EXPECTED  NUMBER  OF *
C   *                    OCCURRENCES (OR FAILURES)  OF   AN   ACCIDENT *
C   *                    SEQUENCE IN AN EVENT TREE, WHERE FAULT  TREES *
C   *                    ARE AVAILABLE FOR EACH BRANCHING POINT IN  THE *
C   *                    EVENT TREE.                                    *
C   *                                                               *
C   *    PROGRAMMER:     THE ASA COMPUTER PROGRAM  WAS  WRITTEN   BY *
C   *                    HENRIQUE  M.  PAULA   AS   A   PART   OF THE *
C   *                    REQUIREMENTS FOR OBTAINING THE  PH. D.  DEGREE *
C   *                    IN NUCLEAR ENGINEERING AT THE UNIVERSITY   OF *
C   *                    TENNESSEE IN KNOXVILLE, TENNESSEE.             *
C   *****************************************************************
```

```
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),              DLTPR (3)
      COMMON ROF
      DIMENSION FSIMP(200),STORE4(2000,4)
C
C     READ THE INPUT DATA
C
      CALL START
C
C             BEGIN THE EXECUTION OF THE ASA COMPUTER PROGRAM
C
              CALL PREPAR
C
C                     COMPUTE THE NUMBER OF POINTS (IDELTA) FOR THE
C                     NUMERICAL INTEGRATION OF THE DENSITY FUNCTION
C                     FOR THE TIME TO FIRST FAILURE.
C
                      DO 90000 I=1,NS
                              I1=INT(TRUN(I)/DLTPR(I))+1
                              IF((FLOAT(I1)/2.).NE.FLOAT(INT(FLOAT(I1)
     1                        /2.)))
     2                                              GO TO 2000
                              JDELTA(I)=I1+1
                              GO TO 2010
 2000                         CONTINUE
                              JDELTA(I)=I1
 2010                         CONTINUE
                              IF(JDELTA(I).GE.3)      GO TO 2011
                              JDELTA(I)=3
 2011                         CONTINUE
                              IF(JDELTA(I).GT.101)    GO TO 20030
C
C                             COMPUTE THE ACTUAL INTEGRATION INTERVAL
C
                              DLTPR(I)=TRUN(I)/(JDELTA(I)-1)
90000                 CONTINUE
C
C     START THE COMPUTATION OF THE EXPECTED NUMBER OF FAILURES
C
      ENF=0.
      IN=1
      ISTORE=1
 701  CONTINUE
```

```
C
C       SUBROUTINE TIMES
C
        CALL TIMES (IN)
        IN=IN+1
        EPS=.001
        IF(NIM.EQ.1) GO TO 11
        IF(NIM.EQ.2) GO TO 12
C*************************************************************************
C                                                                        *
C       NUMERICAL INTEGRATION  --  SIMPSON'S RULE                        *
C                                                                        *
C*************************************************************************
        IF(NDP.GT.201)                                    GO TO 14000
        HELP=DELTA/FLOAT(NDP-1)
C
C       COMPUTE THE FIRST DATA POINT FOR THE NUMERICAL INTEGRATION.
C
        TPRIME=FIRST+EPS
        IF(TPRIME.EQ.FIRST) GO TO 20000
        UNDEP=1.
C
C               COMPUTE THE UNAVAILABILITIES OF THE BASIC EVENTS
C               ( THE VECTOR BASE( 300) ).
C
                CALL BASIC1
        DO 10000 J=1,NS
        IDELTA=JDELTA(J)
        DELTAP=DLTPR(J)
        JJ=J
        IF(TRUN(JJ).NE.0)                         GO TO 15000
                IND05=2
                GO TO 15010
15000   CONTINUE
                IND05=1
15010   CONTINUE
        CALL UNDEPE (JJ,UNDE)
        UNDEP=UNDEP*UNDE
10000   CONTINUE
C
C               COMPUTE THE RATE OF FAILURE OF THE INITIATING EVENT:
C
        IF(IND06.EQ.1)                            GO TO 11010
                CALL ROFIE(ROF)
11010   CONTINUE
        FSIMP(1)=UNDEP*ROF
        STORE4(ISTORE,1) = FIRST
        STORE4(ISTORE,2) = UNDEP
        STORE4(ISTORE,3) = ROF
        STORE4(ISTORE,4) = FSIMP(1)
        ISTORE=ISTORE+1
```

```
C
C       COMPUTE THE REMAINING DATA POINTS FOR THE NUMERICAL INTEGRATION.
C
        DO 10010 I=2,NDP-1
        UNDEP=1.
        TPRIME=HELP*(I-1)+TFINAL
C
C               COMPUTE THE UNAVAILABILITIES OF THE BASIC EVENTS
C               ( THE VECTOR BASE(300) ).
C
                CALL BASIC1
        DO 10015 J=1,NS
        IDELTA=JDELTA(J)
        DELTAP=DLTPR(J)
        JJ=J
        IF(TRUN(JJ).NE.0)                       GO TO 15040
                IND05=2
                GO TO 15050
15040   CONTINUE
                IND05=1
15050   CONTINUE
        CALL UNDEPE (JJ,UNDE)
        UNDEP=UNDEP*UNDE
10015   CONTINUE
C
C               COMPUTE THE RATE OF FAILURE OF THE INITIATING EVENT:
C
        IF(IND06.EQ.1)                          GO TO 11030
                CALL ROFIE(ROF)
11030   CONTINUE
        FSIMP(I)=ROF*UNDEP
        STORE4(ISTORE,1) = TPRIME
        STORE4(ISTORE,2) = UNDEP
        STORE4(ISTORE,3) = ROF
        STORE4(ISTORE,4) = FSIMP(I)
        ISTORE=ISTORE+1
10010   CONTINUE
C
C       COMPUTE THE LAST DATA POINT FOR THE NUMERICAL INTEGRATION.
C
        TPRIME=FINAL-EPS
        UNDEP=1.
C
C               COMPUTE THE UNAVAILABILITIES OF THE BASIC EVENTS
C               ( THE VECTOR BASE( 300) ).
C
                CALL BASIC1
        DO 10020 J=1,NS
        IDELTA=JDELTA(J)
        DELTAP=DLTPR(J)
        JJ=J
```

```
      IF(TRUN(JJ).NE.0)                          GO TO 15020
              IND05=2
              GO TO 15030
15020 CONTINUE
              IND05=1
15030 CONTINUE
      CALL UNDEPE(JJ,UNDE)
      UNDEP=UNDEP*UNDE
10020 CONTINUE
C
C             COMPUTE THE RATE OF FAILURE OF THE INITIATING EVENT:
C
      IF(IND06.EQ.1)                             GO TO 11020
              CALL ROFIE(ROF)
11020 CONTINUE
      FSIMP(NDP)= UNDEP*ROF
      STORE4(ISTORE,1) = FINAL
      STORE4(ISTORE,2) = UNDEP
      STORE4(ISTORE,3) = ROF
      STORE4(ISTORE,4) = FSIMP(NDP)
      ISTORE=ISTORE+1
C
C     PERFORM THE INTEGRATION USING SIMPSON'S RULE.
C
      FEVEN=0.0
      FODD=0.0
      DO 10030 I=2,NDP-1,2
10030 FEVEN=FEVEN+FSIMP(I)
      DO 10040 I=3,NDP-2,2
10040 FODD=FODD+FSIMP(I)
      ENF=ENF+(HELP/3.)*(FSIMP(1)+4.*FEVEN+2.*FODD+FSIMP(NDP))
      IF(FINAL.NE.TFINAL) GO TO 701
      GO TO 20010
C*************************************************************************
C                                                                        *
C     NUMERICAL INTEGRATION  --  RECTANGULAR RULE                        *
C                                                                        *
C*************************************************************************
 11   CONTINUE
      TPRIME=(FINAL+FIRST)/2.
C
C             COMPUTE THE UNAVAILABILITIES OF THE BASIC EVENTS
C             ( THE VECTOR BASE( 300) ).
C
              CALL BASIC1
      UNDEP=1.
      DO 20 I=1,NS
      IDELTA=JDELTA(I)
      DELTAP=DLTPR(I)
      II=I
      IF(TRUN(II).NE.0)                          GO TO 15060
```

```
                IND05=2
                GO TO 15070
15060   CONTINUE
                IND05=1
15070   CONTINUE
        CALL UNDEPE (II,UNDE)
        UNDEP=UNDEP*UNDE
 20     CONTINUE
C
C               COMPUTE THE RATE OF FAILURE OF THE INITIATING EVENT:
C
        IF(IND06.EQ.1)                          GO TO 11040
                CALL ROFIE(ROF)
11040   CONTINUE
        F=ROF*UNDEP
        STORE4(ISTORE,1) = TPRIME
        STORE4(ISTORE,2) = UNDEP
        STORE4(ISTORE,3) = ROF
        STORE4(ISTORE,4) = F
        ISTORE=ISTORE+1
        ENF=ENF+F*(FINAL-FIRST)
        IF(FINAL.NE.TFINAL) GO TO 701
        GO TO 20010
C*************************************************************************
C                                                                        *
C       NUMERICAL INTEGRATION  --  TRAPEZOIDAL RULE                      *
C                                                                        *
C*************************************************************************
 12     CONTINUE
        TPRIME=FIRST+EPS
        IF(TPRIME.EQ.FIRST)                     GO TO 20000
 13     CONTINUE
C
C               COMPUTE THE UNAVAILABILITIES OF THE BASIC EVENTS
C               ( THE VECTOR BASE( 300) ).
C
                CALL BASIC1
        UNDEP=1.
        DO 21 I=1,NS
        IDELTA=JDELTA(I)
        DELTAP=DLTPR(I)
        II=I
        IF(TRUN(II).NE.0.)                      GO TO 15080
                IND05=2
                GO TO 15090
15080   CONTINUE
                IND05=1
15090   CONTINUE
        CALL UNDEPE (II,UNDE)
        UNDEP=UNDEP*UNDE
 21     CONTINUE
```

```
C
C               COMPUTE THE RATE OF FAILURE OF THE INITIATING EVENT:
C
      IF(IND06.EQ.1)                                  GO TO 11050
               CALL ROFIE(ROF)
11050 CONTINUE
      F=ROF*UNDEP
      IF(TPRIME.EQ.(FIRST+EPS))                       GO TO 11051
      STORE4(ISTORE,1) = FINAL
      GO TO 11052
11051 CONTINUE
      STORE4(ISTORE,1) = FIRST
11052 CONTINUE
      STORE4(ISTORE,2) = UNDEP
      STORE4(ISTORE,3) = ROF
      STORE4(ISTORE,4) = F
      ISTORE=ISTORE+1
      IF(TPRIME.NE.FIRST+EPS) GO TO 19
      TPRIME=FINAL-EPS
      FA=F
      GO TO 13
 19   CONTINUE
      ENF=ENF+((FA+F)/2.)*(FINAL-FIRST)
      IF(FINAL.NE.TFINAL) GO TO 701
20010          CONTINUE
                       WRITE(21,1020)
                       WRITE(21,1031)
                       WRITE(21,1060)
                       WRITE(21,1070)
                       WRITE(21,1060)
                       WRITE(21,1080)
                       WRITE(21,1060)
                       WRITE(21,1090)
                       WRITE(21,1060)
                       WRITE(21,1100)
                       WRITE(21,1060)
                       WRITE(21,1031)
                       WRITE(21,1060)
                       WRITE(21,1060)
                       DO 888 IFIZ=1,ISTORE-1
                          WRITE(21,1110) (STORE4(IFIZ,LFIZ),LFIZ=1,4)
                          WRITE(21,1060)
  888                  CONTINUE
                       WRITE(21,1031)
               WRITE(21,1020)
               WRITE(21,1030)
               WRITE(21,1040) ENF
               WRITE(5 ,1040) ENF
               WRITE(21,1030)
               STOP
20030                  CONTINUE
```

```
                      WRITE(21,1030)
                      WRITE(21,1020)
                      WRITE(21,1050) JDELTA(I),I
                      WRITE(21,1010)
                      STOP
20000                         CONTINUE
                              WRITE(21,1000) EPS
                              WRITE(21,1010)
                              STOP
14000  CONTINUE
       WRITE(21,1031)
       WRITE(21,1120) NDP
       WRITE(21,1010)
       WRITE(21,1031)
       STOP
C  ***************************************************************************
C                                                                             *
C                       FORMATS                                               *
C                                                                             *
C  ***************************************************************************
 1000  FORMAT('1 THE TIME INTERVAL EPS=',E15.5,' IS TOO SMALL.')
 1010  FORMAT('0 THE PROGRAM WILL NOT BE EXECUTED')
 1020  FORMAT('1')
 1030  FORMAT('0-------------------------------------------------------
     1----------------')
 1031  FORMAT(' -------------------------------------------------------
     1----------------')
 1040  FORMAT('0THE SYSTEM EXPECTED NUMBER OF FAILURES IS ',E11.6)
 1050  FORMAT('0THE NUMBER OF DATA POINTS (',I2,') REQUIRED TO NUMERICA
     1LLY COMPUTE THE PROBABILITY OF ',//,'FAILURE TO RUN FOR SYSTEM (',
     2I2,') HAS EXCEEDED THE LIMIT (101).')
1060    FORMAT(' ','|',10X,'|',19X,'|',18X,'|',20X,'|')
1070    FORMAT(' ','|',3X,'TIME',3X,'|',1X,'ACCIDENT SEQUENCE',1X,'|',1X
     1,'RATE OF FAILURE',2X,'|',7X,'U',12X,'|')
1080    FORMAT(' ','|',2X,'(HOURS)',1X,'|',2X,'UNDEPENDABILITY',2X,'|',4
     1X,'OF THE',8X,'|',7X,'X',12X,'|')
1090    FORMAT(' ','|',10X,'|',7X,'(U)',9X,'|',1X,'INITIATING EVENT',1X,
     1'|',6X,'ROF',11X,'|')
1100    FORMAT(' ','|',10X,'|',19X,'|',1X,'(ROF) IN (1/HR)',2X,'|',20X,'
     1|')
1110    FORMAT(' ','|',1X,F8.3,1X,'|',4X,E9.4,6X,'|',4X,E9.4,5X,'|',6X,E
     19.4,5X,'|')
1120    FORMAT('0THE NUMBER OF DATA POINTS (',I3,') REQUIRED TO NUMERICA
     1LLY COMPUTE THE "ENF" HAS EXCEEDED THE LIMIT (201).')
        END
```

```
      SUBROUTINE START
      COMMON  INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON     IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON        NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON           MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON              TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON              DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON              DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON              NBE, IND02, IND03, IND04 , IND05
      COMMON              IND10, IND09, IND08, IND07,IND06
      COMMON              JDELTA(3),             DLTPR (3)
      COMMON ROF
      DIMENSION PNAME(20),ILET(6)
C
C                     INITIALIZATION
C
                      DO 100 I=1,6
                              ILET(I)=0
00100                 CONTINUE
C
C     READ THE KEY WORD
C
00010 CONTINUE
      READ(20,1000) KEY
      IF(KEY.NE.'TIT') GO TO 20000
10000         CONTINUE
C*************************************************************************
C                                                                        *
C             THE TITLE DATA GROUP                                       *
C                                                                        *
C*************************************************************************
              ILET(1)=1
              READ(20,10500) (PNAME(I),I=1,20)
              WRITE(21,10510)
              READ(20,10505) IDUMMY
              IF(IDUMMY.EQ.-1) GO TO 10020
                      WRITE(21,10560)
                      WRITE(21,10570)
                      STOP
10020         CONTINUE
              DO 10010 J=1,12
                      WRITE(21,10550)
10010         CONTINUE
              WRITE(21,10520)
              WRITE(21,10530)
              WRITE(21,10540) (PNAME(I),I=1,20)
              WRITE(21,10520)
              GO TO 10
20000 CONTINUE
      IF(KEY.NE.'SYS') GO TO 30000
      IF(ILET(1).EQ.0) GO TO 110
```

```
C*************************************************************************
C                                                                        *
C              THE SYSTEM DATA GROUP                                     *
C                                                                        *
C*************************************************************************
               ILET(2)=1
C
C                      INITIALIZATION
C
                       DO 20010 I1=1,3
                          DO 20020 I2=1,20
                             DO 20030 I3=1,20
                                IF(I3.GT.5) GO TO 20040
                                   INFORM(I1,I2,I3)=0
20040                              CONTINUE
                                   JNFORM(I1,I2,I3)=0
20030                           CONTINUE
20020                      CONTINUE
20010                   CONTINUE
C
C      LOAD THE VECTOR INFORM(3,20,5).
C
               READ(20,20500) NS
               WRITE(21,10510)
               WRITE(21,10520)
               WRITE(21,20510) NS
               WRITE(21,10520)
               DO 20050 I=1,NS
C
C                      READ THE SYSTEM IDENTIFICATION (I1) AND THE
C                      NUMBER OF MCS'S IN EACH SYSTEM (I1).
C
                       READ(20,20520) I1,NUMMCS(I)
                       IF(I1.NE.I) GO TO 20060
C
C                      READ THE BASIC EVENTS IN EACH MCS OF THE SYSTEM
C                      'I1'.
C
                       DO 20070 I2=1,NUMMCS(I)
                              READ(20,20540) (INFORM(I1,I2,J),J=1,5)
20070                  CONTINUE
20050          CONTINUE
C
C              READ THE NEXT CARD AFTER THE DATA FOR INFORM(3,20,5).
C
               READ(20,20500) ITEST
               IF(ITEST.NE.-1) GO TO 20080
C
C              LOAD THE VECTOR JNFORM(3,20,20).
C                      (IF APPLICABLE)
C
```

```
                READ(20,20500) IND01
                IF(IND01.NE.1) GO TO 20090
                DO 20100 I=1,NS
C
C                  READ THE SYSTEM IDENTIFICATION (I1) AND THE NUMBER
C                  OF MPS'S IN EACH SYSTEM (I1).
C
                   READ(20,20520) I1,NUMMPS(I)
                   IF(I1.NE.I) GO TO 20140
C
C                  READ THE BASIC EVENTS IN EACH MPS OF SYSTEM 'I1'.
C
                   DO 20110 I2=1,NUMMPS(I)
                      I4=1
                      DO 20120 I3=1,2
                         READ(20,20540) (JNFORM(I1,I2,J),J=I4,I4+9)
                         IF(JNFORM(I1,I2,I4).LT.0) GO TO 20130
                            I4=I4+10
20120                 CONTINUE
20130                 CONTINUE
                      JNFORM(I1,I2,I4)=0
20110              CONTINUE
C
C                  READ THE NEXT CARD AFTER THE DATA FOR
C                  JNFORM(3,20,20).
C
                   READ(20,20500) ITEST
                   IF(ITEST.NE.-1) GO TO 20150
20100           CONTINUE
20090           CONTINUE
                GO TO 20091
C
C                       TYPE ERROR MESSAGES:
C
20060                                  CONTINUE
                                       WRITE(21,10510)
                                       WRITE(21,20530)
                                       WRITE(21,20531) I1,I
                                       WRITE(21,10570)
                                       STOP
20080                                  CONTINUE
                                       WRITE(21,10510)
                                       WRITE(21,20530)
                                       WRITE(21,20550)
                                       WRITE(21,10570)
                                       STOP
20140                                  CONTINUE
                                       WRITE(21,10510)
                                       WRITE(21,20530)
                                       WRITE(21,20532) I1,I
                                       WRITE(21,10570)
                                       STOP
```

```
20150                                   CONTINUE
                                        WRITE(21,10510)
                                        WRITE(21,20530)
                                        WRITE(21,20551)
                                        WRITE(21,10570)
                                        STOP
20091           CONTINUE
C
C       ECHO-PRINTING THE VECTORS INFORM(3,20,5) AND JNFORM(3,20,20).
C
        DO 20160 LL=1,NS
                WRITE(21,10510)
                WRITE(21,10520)
                WRITE(21,20560) LL
                WRITE(21,10520)
                WRITE(21,20570)
                WRITE(21,10550)
                DO 20170 KK=1,NUMMCS(LL)
                        WRITE(21,20580) KK,(INFORM(LL,KK,M),M=1,5)
20170           CONTINUE
                IF(IND01.NE.1)                          GO TO 20160
                WRITE(21,10510)
                DO 20180 KK=1,NUMMPS(LL)
                        WRITE(21,20590) KK
                        JAUX=1
                        DO 20190 JJ=1,2
                                WRITE(21,10550)
                                IF(JNFORM(LL,KK,JAUX).EQ.0) GO TO 20200
                                WRITE(21,20540)(JNFORM(LL,KK,L),L=JAUX,
     1                          JAUX+9)
                                JAUX=JAUX+10
20190                   CONTINUE
20200                   CONTINUE
20180           CONTINUE
20160   CONTINUE
        GO TO 10
30000   CONTINUE
        IF(KEY.NE.'BAS') GO TO 40000
        IF(ILET(2).EQ.0) GO TO 111
C*************************************************************************
C                                                                        *
C               THE BASIC EVENT DATA GROUP                               *
C                                                                        *
C*************************************************************************
                ILET(3)=1
                READ(20,30510) NBE
C
C               THE BASIC EVENTS IN INFORM(ISYS,IMCS,IBAS)    AND
C               JNFORM(ISYS,IMPS,IBAS) SHOULD HAVE IDENTIFICATION
C               NUMBERS BETWEEN 1 AND "NBE".
C
```

```
C
C                       CHECK INFORM(ISYS,IMCS,IBAS):
C
      DO 36000                                      LL1=1,NS
         DO 36010                                   LL2=1,NUMMCS(LL1)
            DO 36020                                LL3=1,5
               IF(INFORM(LL1,LL2,LL3).EQ.0)              GO TO 36010
               IF((INFORM(LL1,LL2,LL3).LT.1).OR.
     1            ( INFORM(LL1,LL2,LL3).GT.NBE))         GO TO 36030
36020       CONTINUE
36010    CONTINUE
36000 CONTINUE
C
C                       CHECK JNFORM(ISYS,IMPS,IBAS):
C
      DO 36100                                      LL1=1,NS
         DO 36110                                   LL2=1,NUMMPS(LL1)
            DO 36120                                LL3=1,20
              IF(JNFORM(LL1,LL2,LL3).EQ.0)              GO TO 36110
              IF((JNFORM(LL1,LL2,LL3).LT.1).OR.
     1           (JNFORM(LL1,LL2,LL3).GT.NBE))        GO TO 36130
36120       CONTINUE
36110    CONTINUE
36100 CONTINUE
C
C                       INITIALIZATION.
C
                        DO 30010          I1=1,NBE
                           DO 30020       I2=1,   12
                           IF(I2.GT.3) GO TO 30030
                              IBE(I1,I2)=        0
30030                      CONTINUE
                              RBE(I1,I2)=        0
30020                      CONTINUE
30010                   CONTINUE
C
C               LOAD THE VECTORS IBE(NBE,3) AND RBE(NBE,12).
C
                DO 30040 JJ1=1,300
                        READ(20,30510) IDUMMY
                        IF(IDUMMY.LT.0) GO TO 30041
                        IF(IDUMMY.NE.JJ1 ) GO TO 30050
                        READ(20,30520)IBE(JJ1,1),(RBE(JJ1,L),L=1,5)
                        READ(20,30520)IBE(JJ1,2),(RBE(JJ1,L),L=6,10)
                        IF(IBE(JJ1,1).EQ.2)             GO TO 39040
                        READ(20,30520)IBE(JJ1,3),(RBE(JJ1,L),L=11,12)
39040                   CONTINUE
30040           CONTINUE
                READ(20,30510) IDUMMY
                IF(IDUMMY.NE.-1)                GO TO 30070
30041           CONTINUE
                GO TO 30080
```

```
C
C                              TYPE ERROR MESSAGES
C
36030                          CONTINUE
                               WRITE(21,10510)
                               WRITE(21,30530)
                               WRITE(21,30810)LL1,LL2,LL3,INFORM(
     1                                        LL1,LL2,LL3)
                               WRITE(21,10570)
                               STOP
36130                          CONTINUE
                               WRITE(21,10510)
                               WRITE(21,30530)
                               WRITE(21,30820)LL1,LL2,LL3,JNFORM(
     1                                        LL1,LL2,LL3)
                               WRITE(21,10570)
                               STOP
30050                          CONTINUE
                               WRITE(21,10510)
                               WRITE(21,30530)
                               WRITE(21,30540) JJ1
                               WRITE(21,10570)
                               STOP
30070                          CONTINUE
                               WRITE(21,10510)
                               WRITE(21,30530)
                               WRITE(21,30560)
                               WRITE(21,10570)
                               STOP
30080  CONTINUE
C
C              ECHO-PRINTING THE BASIC EVENT DATA:
C
       WRITE(21,10510)
       DO 30100 IBAS=1,NBE
              WRITE(21,10550)
              WRITE(21,30580)
              WRITE(21,30570) IBAS
              WRITE(21,30580)
              WRITE(21,10550)
              IF(IBE(IBAS,1).EQ.1)              GO TO 30110
              IF(IBE(IBAS,1).EQ.2)              GO TO 30110
C
C                      THE BASIC EVENT TYPE IS NOT ACCEPTABLE.
C                      THE PROGRAM CANNOT BE EXECUTED.
C
                       WRITE(21,30530)
                       WRITE(21,30590) IBAS
                       WRITE(21,10510)
                       STOP
```

```
C
C               THE PRE-DEMAND DATA:
C
30110           CONTINUE
                WRITE(21,30600) IBE(IBAS,1)
                IF(IBE(IBAS,2).EQ.1)                        GO TO 30120
                IF(IBE(IBAS,2).EQ.2)                        GO TO 30130
                IF(IBE(IBAS,2).EQ.3)                        GO TO 30140
                IF(IBE(IBAS,2).EQ.4)                        GO TO 30150
                WRITE(21,30530)
                WRITE(21,30610) IBE(IBAS,2)
                WRITE(21,10570)
                STOP
30120                   CONTINUE
                        WRITE(21,30620)
                        WRITE(21,30660) RBE(IBAS,10)
                        GO TO 30200
30130                   CONTINUE
                        WRITE(21,30630)
                        WRITE(21,30670) RBE(IBAS,1)
                        WRITE(21,30671) RBE(IBAS,10)
                        GO TO 30200
30140                   CONTINUE
                        WRITE(21,30640)
                        WRITE(21,30680) RBE(IBAS,1),RBE(IBAS,5)
                        WRITE(21,30671) RBE(IBAS,10)
                        GO TO 30200
30150                   CONTINUE
                        WRITE(21,30650)
                        WRITE(21,30690) RBE(IBAS,1),RBE(IBAS,2)
                        WRITE(21,30700) RBE(IBAS,3),RBE(IBAS,4)
                        WRITE(21,30710) RBE(IBAS,5),RBE(IBAS,6)
                        WRITE(21,30720) RBE(IBAS,7),RBE(IBAS,8)
                        WRITE(21,30730) RBE(IBAS,9),RBE(IBAS,10)
C
C               THE POST-DEMAND DATA:
C
30200           CONTINUE
                IF(IBE(IBAS,1).EQ.2)                        GO TO 30100
                IF(IBE(IBAS,3).EQ.2)                        GO TO 30220
                IF(IBE(IBAS,3).EQ.3)                        GO TO 30230
                WRITE(21,30740) IBAS,IBE(IBAS,3)
                WRITE(21,10570)
                STOP
30220                   CONTINUE
                        WRITE(21,30760)
                        WRITE(21,30790) RBE(IBAS,11)
                        GO TO 30100
30230                   CONTINUE
                        WRITE(21,30770)
                        WRITE(21,30800) RBE(IBAS,11),RBE(IBAS,12)
                        GO TO 30100
```

```
30100  CONTINUE
       GO TO 10
40000  CONTINUE
       IF(KEY.NE.'TIM')                          GO TO 50000
       IF(ILET(3).EQ.0)                          GO TO   112
C*************************************************************************
C                                                                        *
C                        THE TIME  DATA  GROUP                           *
C                                                                        *
C*************************************************************************
       ILET(4)=1
       READ(20,40510) TFINAL,DELTA1,DELTA2
       READ(20,40510) (TRUN(LL1),LL1=1,NS)
       READ(20,40510) (DLTPR(LL2),LL2=1,NS)
       READ(20,10505) IDUMMY
       IF(IDUMMY.NE.-1)                          GO TO 40040
C
C      CHECK FOR INVALID ENTRIES:
C
       DO 40010 I4=1,NS
          IF(TRUN(I4).LT.0)                      GO TO 40020
          IF(DLTPR(I4).LE.0.)                    GO TO 40030
40010  CONTINUE
       GO TO 10
40020  CONTINUE
       WRITE(21,40520)
       WRITE(21,40530) I4
       WRITE(21,10570)
       STOP
40030  CONTINUE
       WRITE(21,40520)
       WRITE(21,40540) I4
       WRITE(21,10570)
       STOP
40040  CONTINUE
       WRITE(21,40550)
       WRITE(21,10570)
       STOP
50000  CONTINUE
       IF(KEY.NE.'PRI')                          GO TO 60000
       IF(ILET(4).EQ.0)                          GO TO   113
C*************************************************************************
C                                                                        *
C                        THE PRINT DATA GROUP                            *
C                                                                        *
C*************************************************************************
       ILET(5)=1
       READ(20,10505) IND10
       READ(20,10505) IDUMMY
       IF(IDUMMY.NE.-1)                                    GO TO 50020
       IF(IND10.EQ.1)                            GO TO 50010
       WRITE(21,50510)
```

```
      GO TO 10
50010 CONTINUE
      WRITE(21,50520)
      GO TO 10
50020 CONTINUE
      WRITE(21,50530)
      WRITE(21,10570)
      STOP
60000 CONTINUE
      IF(KEY.NE.'INI')                          GO TO 70000
      IF(ILET(5).EQ.0)                          GO TO   114
C*************************************************************************
C                                                                        *
C                      THE INITIATING EVENT DATA GROUP                   *
C                                                                        *
C*************************************************************************
      ILET(6)=1
      READ(20,10505) IND06
      IF(IND06.NE.1) GO TO 60100
      READ(20,60502) ROF
      WRITE(21,60620) ROF
      READ(20,10505) IDUMMY
      IF(IDUMMY.NE.-1)                                  GO TO 60200
      GO TO 10
60200 CONTINUE
      WRITE(21,60610)
      WRITE(21,10570)
      STOP
60100 CONTINUE
      READ(20,60501) ROFVCT(1,1),ROFVCT(1,2)
C
C     THE FIRST TIME DATA POINT MUST BE ZERO:
C
      IF(ROFVCT(1,1).NE.0.)                             GO TO 60010
      DO 60020 I=2,100
         READ(20,60501) ROFVCT(I,1),ROFVCT(I,2)
         IF(ROFVCT(I,1).EQ.-1.)                         GO TO 60021
         KOUNT=I-1
60020 CONTINUE
      READ(20,60500) DUMMY
      IF(DUMMY.NE.-1.)                                  GO TO 60030
      KOUNT=100
60021 CONTINUE
C
C     THE LAST TIME DATA POINT MUST BE AT TFINAL:
C
      IF(ROFVCT(KOUNT,1).EQ.TFINAL)                     GO TO 10
C
C
C     ERROR MESSAGES:
C
```

```
      WRITE(21,60510)
      WRITE(21,60540) ROFVCT(KOUNT,1)
      WRITE(21,60530)
      STOP
60010 CONTINUE
      WRITE(21,60510)
      WRITE(21,60520) ROFVCT(1,1)
      WRITE(21,60530)
      STOP
60030 CONTINUE
      WRITE(21,60510)
      WRITE(21,60560)
      WRITE(21,60570)
      WRITE(21,60530)
      STOP
70000 CONTINUE
      IF(KEY.NE.'RUN')                                    GO TO 70010
      IF(ILET(6).NE.1)                                    GO TO 116
C*************************************************************************
C                                                                        *
C             THE "RUN" DATA GROUP.                                      *
C                                                                        *
C*************************************************************************
      READ(20,70520) IND02,IND03,IND04,IND09,IND08
      READ(20,70540) BOUND
      READ(20,10505) IDUMMY
      IF(IDUMMY.EQ.-1)                                    GO TO 70020
      WRITE(21,10510)
      WRITE(21,70530) IDUMMY
      WRITE(21,10570)
      STOP
C*************************************************************************
C                                                                        *
C             ERROR MESSAGES INDICATING A MISSING OR DISPLACED           *
C                                                                        *
C             DATA GROUP.                                                *
C                                                                        *
C*************************************************************************
00110 CONTINUE
C
C     THE "TITLE DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,10580)
      WRITE(21,10570)
      STOP
00111 CONTINUE
C
C     THE "SYSTEM DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,20600)
```

```
      WRITE(21,10570)
      STOP
00112 CONTINUE
C
C     THE "BASIC EVENT DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,30500)
      WRITE(21,10570)
      STOP
00113 CONTINUE
C
C     THE "TIME DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,40500)
      WRITE(21,10570)
      STOP
00114 CONTINUE
C
C     THE "PRINT DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,50500)
      WRITE(21,10570)
      STOP
00115 CONTINUE
C
C     THE "INITIATOR DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,60500)
      WRITE(21,10570)
      STOP
00116 CONTINUE
C
C     THE "RUN DATA GROUP" IS EITHER MISSING OR DISPLACED.
C
      WRITE(21,10510)
      WRITE(21,70500)
      WRITE(21,10570)
      STOP
70010 CONTINUE
      WRITE(21,10510)
      WRITE(21,70510) KEY
      WRITE(21,10570)
      STOP
70020 CONTINUE
      RETURN
```

```
C*************************************************************************
C                                                                        *
C               FORMATS                                                  *
C                                                                        *
C*************************************************************************
01000   FORMAT(1X,A3)
C
C               10000 LEVEL
C
10500   FORMAT(20A4)
10505   FORMAT(3I2)
10510   FORMAT('1')
10520   FORMAT('0------------------------------------------------------
     1-----------------------------')
10530   FORMAT('0 THE ACCIDENT SEQUENCE IDENTIFICATION TITLE:')
10540   FORMAT('0',20A4)
10550   FORMAT('0')
10560   FORMAT('1 THE LAST CARD IN THE "TITLE DATA GROUP" MUST CONTAIN T
     1HE NUMBER -1')
10570   FORMAT('0 THE PROGRAM WILL NOT BE EXECUTED')
10580   FORMAT('0THE "TITLE DATA GROUP" IS EITHER MISSING OR DISPLACED')
C
C               20000 LEVEL
C
20500   FORMAT(I2)
20510   FORMAT('0THE NUMBER OF SYSTEMS IN THE ACCIDENT SEQUENCE: ',I2)
20520   FORMAT(I2,I3)
20530   FORMAT('0AN ERROR HAS BEEN DETECTED IN THE "SYSTEM DATA GROUP".
     1  ')
20531   FORMAT('0THE PROGRAM IS READING ',I2,' AS THE IDENTIFICATION NUM
     1BER FOR SYSTEM ',I2,' (MCS).')
20532   FORMAT('0THE PROGRAM IS READING ',I2,' AS THE IDENTIFICATION
     1NUMBER FOR SYSTEM ',I2,' (MPS).')
20540   FORMAT(10I5)
20550   FORMAT('0AFTER THE DATA FOR "INFORM(I,J,K)", THE PROGRAM SHOU
     1LD READ A -1.')
20551   FORMAT('0AFTER THE DATA FOR "JNFORM(3,20,20)", THE PROGRAM
     1SHOULD READ A "-1".')
20560   FORMAT('0SYSTEM NUMBER ',I3)
20570   FORMAT('0MCS ',20X,' BASIC EVENTS')
20580   FORMAT(2X,I2,20X,10I5)
20590   FORMAT(' MINIMAL PATH SET ',I3,' CONTAINS THE FOLLOWING BASIC
     1EVENTS:')
20600   FORMAT('0THE "SYSTEM DATA GROUP" IS EITHER MISSING    OR  DISPLA
     1CED.')
C
C               30000 LEVEL
C
30500   FORMAT('0THE "BASIC EVENT DATA GROUP" IS EITHER MISSING OR DISPL
     1ACED.')
30510   FORMAT(I5)
30520   FORMAT(I5,5E10.5)
```

```
30530  FORMAT('0AN ERROR HAS BEEN DETECTED IN THE "BASIC EVENT DATA GRO
     1UP".')
30540  FORMAT('0BASIC EVENT ',I4,' IS EITHER INVALID OR OUT OF ORDER.')
30560  FORMAT('01000 BASIC EVENTS HAVE BEEN PROVIDED.  THE NEXT DATA
     1CARD MUST CONTAIN THE NUMBER -1.')
30570  FORMAT(' BASIC EVENT # ',I4)
30580  FORMAT('0--------------------')
30590  FORMAT('0THE TYPE OF THE BASIC EVENT ',I4,' SHOULD BE EITHER  1
     1OR 2.')
30600  FORMAT('0TYPE OF BASIC EVENT:',I2)
30610  FORMAT('0THE PRE-DEMAND CLASS OF THIS BASIC EVENT SHOULD BE EITH
     1ER 1, 2, 3, OR 4, NOT ',I2,'.')
30620  FORMAT('0BASIC EVENT PRE-DEMAND CLASS: 1  (CONSTANT UNAVAILABILI
     1TY).')
30630  FORMAT('0BASIC EVENT PRE-DEMAND CLASS: 2  (NONREPAIRABLE).')
30640  FORMAT('0BASIC EVENT PRE-DEMAND CLASS: 3  (CONSTANTLY MONITORED)
     1.')
30650  FORMAT('0BASIC EVENT PRE-DEMAND CLASS: 4  (PERIODICALLY TESTED).
     1')
30660  FORMAT('0PRE-DEMAND UNAVAILABILITY= ',E12.5)
30670  FORMAT('0PRE-DEMAND HAZARD RATE = ',E12.5,'   1/HR.')
30671  FORMAT('0UNAVAILABILITY PER DEMAND = ',E12.5)
30680  FORMAT('0PRE-DEMAND HAZARD RATE = ',E12.5,' 1/HR',10X,'PRE-DEMAND
     1 REPAIR TIME = ',F8.2,' HRS.')
30690  FORMAT('0PRE-DEMAND HAZARD RATE = ',E12.5,' 1/HR',10X,'PERIODIC T
     1EST INTERVAL = ',F8.2,' HRS.')
30700  FORMAT('0FIRST TEST INTERVAL = ',F8.2,' HRS,',10X,'TEST PERIOD =
     1',F8.2,' HRS.')
30710  FORMAT('0REPAIR PERIOD = ',F8.2,' HRS,',10X,'OVERRIDE UNAVAILABI
     1LITY = ',E12.5)
30720  FORMAT('0PROBABILITY OF A TEST CAUSED FAILURE = ',E12.5,10X,'DETE
     1CTION INEFFICIENCY = ',E12.5)
30730  FORMAT('0UNDETECTED HAZARD RATE = ',E12.5,' 1/HR,',10X,'UNAVAILAB
     1ILITY PER DEMAND = ',E12.5)
30740  FORMAT('0THE POST-DEMAND CLASS FOR BASIC EVENT #',I4,' (',I3,')
     1IS NOT ACCEPTABLE.')
30750  FORMAT('0BASIC EVENT POST-DEMAND CLASS: 1 (CONSTANT UNAVAILABILI
     1TY).')
30760  FORMAT('0BASIC EVENT POST-DEMAND CLASS: 2 (NONREPAIRABLE).')
30770  FORMAT('0BASIC EVENT POST-DEMAND CLASS : 3 (CONSTANTLY MONITORED)
     1.')
30780  FORMAT('0POST-DEMAND CONSTANT UNAVAILABILITY = ',E12.5)
30790  FORMAT('0POST-DEMAND HAZARD RATE = ',E12.5,' 1/HR.')
30800  FORMAT('0POST-DEMAND HAZARD RATE = ',E12.5,' 1/HR,',10X,'POST-DE
     1MAND REPAIR TIME = ',E12.5,' HRS.')
30810  FORMAT('0THE BASIC EVENT IN INFORM(',I2,',',I3,',',I2,') = ',I5,
     1' IS NOT ACCEPTABLE.')
30820  FORMAT('0THE BASIC EVENT IN JNFORM(',I2,',',I3,',',I2,') = ',I5,
     1' IS NOT ACCEPTABLE.')
C
C                        4000 LEVEL
C
```

```
40500  FORMAT('0THE "TIME" DATA GROUP IS EITHER MISSING OR DISPLACED.')
40510  FORMAT(10F8.2)
40520  FORMAT('0AN ERROR HAS BEEN DETECTED IN THE "TIME" DATA GROUP.')
40530  FORMAT('0THE MISSION TIME FOR SYSTEM ',I2,' MUST BE GREATER THAN
     1OR EQUAL TO ZERO.')
40540  FORMAT('0THE MINIMAL INTEGRATION INTERVAL FOR SYSTEM ',I2,' MUST
     1 BE GREATER THAN ZERO.')
40550  FORMAT('0THE "TIME" DATA GROUP MUST BE FOLLOWED BY THE NUMBER -1
     1.')
C
C                        5000 LEVEL
C
50500  FORMAT('0THE "PRINT" DATA GROUP IS EITHER MISSING OR DISPLACED.')
50510  FORMAT('1THE DISCONTINUITY TIME POINTS WILL NOT BE PRINTED.')
50520  FORMAT('1THE DISCONTINUITY TIME POINTS WILL BE PRINTED.')
50530  FORMAT('0THE "PRINT" DATA GROUP MUST BE FOLLOWED BY THE NUMBER -
     11.')
C
C                        6000 LEVEL
C
60500  FORMAT('0THE "INITIATING EVENT" DATA GROUP IS EITHER MISSING OR
     1DISPLACED.')
60501  FORMAT(F10.2,E10.5)
60502  FORMAT(E10.5)
60510  FORMAT('1AN ERROR HAS BEEN DETECTED BY SUBROUTINE "START".')
60520  FORMAT('0THE FIRST DATA POINT FOR THE INITIATING EVENT MUST BE Z
     1ERO, NOT ',E10.5)
60530  FORMAT('0THE PROGRAM CANNOT BE EXECUTED.')
60540  FORMAT('0THE LAST DATA POINT FOR THE INITIATING EVENT MUST BE AT
     1 "TFINAL", NOT ',E10.5)
60560  FORMAT('0100 TIME POINTS HAVE BEEN ENTERED FOR THE INITIATING EV
     1ENT FREQUENCY.')
60570  FORMAT('0THE NEXT CARD MUST CONTAIN THE NUMBER -1, NOT ',F10.2)
60600  FORMAT(E10.5)
60610  FORMAT('0THE "INITIATING EVENT" DATA GROUP MUST BE FOLLOWED BY T
     1HE NUMBER -1.')
60620  FORMAT('0THE ROF FOR THE INITIATING EVENT IS ',E10.5,' .')
C
C                        7000 LEVEL
C
70500  FORMAT('0THE "RUN" DATA GROUP IS EITHER MISSING OR DISPLACED.')
70510  FORMAT('0THE KEYWORD "',A3,'" IS NOT ACCEPTABLE.')
70520  FORMAT(1015)
70530  FORMAT('0AFTER THE DATA FOR THE "RUN" DATA GROUP, THE PROGRAM MU
     1ST READ A -1, NOT ',I2)
70540  FORMAT(F6.5)
       END
```

```
      SUBROUTINE PREPAR
      COMMON  INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON     IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON        NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON           MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON              TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON              DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON              DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON              NBE, IND02, IND03, IND04 , IND05
      COMMON              IND10, IND09, IND08, IND07,IND06
      COMMON              JDELTA(3),              DLTPR (3)
C*********************************************************************
C                                                                     *
C     SHOULD THE COMMON BASIC EVENTS BE DELETED BY SUBROUTINE PREPAR?*
C                                                                     *
C*********************************************************************
      IF((IND09.EQ.1).AND.(NS.GT.1))                  GO TO 20000
      IF(NS.EQ.1)                                     GO TO 10520
      WRITE(21,1010)
      WRITE(21,1011)
      GO TO 10520
20000 CONTINUE
      WRITE(21,1020)
      WRITE(21,1021)
C
C     DELETE THE COMMON BASIC EVENTS IN THE VECTOR INFORM(3,20,5).
C
C
C                       INITIALIZE THE VECTOR IHELP(300).
C
                        INDIC1=1
                        IHELP(1)=INFORM(1,1,1)
                        DO 20010 IMCS=1,20
                           IF(INFORM(1,IMCS,1).EQ.0)    GO TO 20011
                           DO 20020 IBAS=1,5
                              IF(INFORM(1,IMCS,IBAS).EQ.0)
     1                                                 GO TO 20010
                              DO 20030 II=1,INDIC1
                                 IF(INFORM(1,IMCS,IBAS).EQ.IHELP(II))
     1                                                 GO TO 20020
20030                         CONTINUE
                              INDIC1=INDIC1+1
                              IHELP(INDIC1)=INFORM(1,IMCS,IBAS)
20020                      CONTINUE
20010                   CONTINUE
20011                   CONTINUE
                        INDIC2=INDIC1
C
C     DELETE THE BASIC EVENTS IN SYSTEM ISYS AND IMPLEMENT THE VECTOR
C     IHELP(300).
C
```

```
      DO 20040 ISYS=2,3
         IEVENT=0
         IF(INFORM(ISYS,1,1).EQ.0)                    GO TO 20041
         DO 20050 IMCS=1,20
            IF(INFORM(ISYS,IMCS,1).EQ.0)              GO TO 20051
            DO 20060 IBAS=1,5
21061          CONTINUE
               IF(INFORM(ISYS,IMCS,IBAS).EQ.0)        GO TO 20050
               DO 20070 II=1,INDIC1
                  IF(INFORM(ISYS,IMCS,IBAS).EQ.IHELP(II))
     1                                                GO TO 20080
                  GO TO 20070
C
C     THE BASIC EVENT APPEARS IN AT LEAST ONE OF THE PRECEDING
C
C     SYSTEMS.  DELETE IT FROM THE CURRENT SYSTEM.
C
20080             CONTINUE
                  IEVENT=INFORM(ISYS,IMCS,IBAS)
                  DO 21080 JJ=IBAS,4
                     INFORM(ISYS,IMCS,JJ)=INFORM(ISYS,IMCS,JJ+1)
21080             CONTINUE
                  INFORM(ISYS,IMCS,10)=0
                  IF(INFORM(ISYS,IMCS,1).NE.0)        GO TO 21060
                  WRITE(21,1030)
                  WRITE(21,1050)
                  WRITE(21,1051)
                  WRITE(21,1060) IMCS,ISYS
                  WRITE(21,1070)
                  WRITE(21,1080)
                  STOP
21060             CONTINUE
C
C                 DELETE THE MPS'S WHICH CONTAIN THE BASIC EVENT
C                 DELETED IN THE MCS 'IMCS' (INFORM(ISYS,IMCS,IBAS)).
C
                  IF(IND01.NE.1)              GO TO 21061
                  DO 30000 JMPS=1,20
                     IF(JNFORM(ISYS,JMPS,1).EQ.0)     GO TO 30001
                     DO 30010 JBAS=1,20
                        IF(JNFORM(ISYS,JMPS,JBAS).EQ.0)
     1                                                GO TO 30011
                        IF(JNFORM(ISYS,JMPS,JBAS).EQ.IEVENT)
     1                                                GO TO 30020
30010                CONTINUE
30011                CONTINUE
C
C                       THE MPS 'JMPS' DOES NOT CONTAIN THE BASIC
C                       EVENT INFORM(ISYS,IMCS,IBAS).  SELECT THE
C                       NEXT 'MPS'.
C
                                                    GO TO 30000
```

```
C
C                       THE MPS 'JMPS' CONTAINS THE BASIC EVENT
C                       INFORM(ISYS,IMCS,IBAS).  DELETE 'JMPS'.
C
30020                   CONTINUE
                        DO 30030 LL1=JMPS,19
                           IF(JNFORM(ISYS,LL1,1).EQ.0)
     1                                                 GO TO 30031
                           DO 30040 LL2=1,20
                              JNFORM(ISYS,LL1,LL2)=JNFORM(ISYS,
     1                        LL1+1,LL2)
                              IF(JNFORM(ISYS,LL1,LL2).EQ.0)
     1                                                 GO TO 30030
30040                      CONTINUE
30030                   CONTINUE
30031                   CONTINUE
                        JNFORM(ISYS,20,1)=0
30000             CONTINUE
30001             CONTINUE
                                                       GO TO 21061
20070          CONTINUE
C
C      THE BASIC EVENT DOES NOT APPEAR IN ANY OF THE PRECEDING
C
C      SYSTEMS.
C
                INDIC2=INDIC2+1
                IHELP(INDIC2)=INFORM(ISYS,IMCS,IBAS)
20060        CONTINUE
20050     CONTINUE
20051     CONTINUE
          INDIC1=INDIC2
20040  CONTINUE
20041  CONTINUE
10520  CONTINUE
C*************************************************************************
C                                                                        *
C               LOAD THE VECTOR MCSTYP(3,20).                            *
C                                                                        *
C*************************************************************************
C
C                       INITIALIZE THE VECTOR MCSTYP(3,20).
C
                        KOUNT=0
                        DO 10500 ISYS=1,3
                           DO 10510 IMCS=1,20
                              MCSTYP(ISYS,IMCS)=1
10510                      CONTINUE
10500                   CONTINUE
C
C      CHECK FOR THE MCS'S WHICH CONTAIN BASIC EVENTS OF TYPE 2 ONLY
C      (FAILURE TO START ONLY).
C
```

```
      DO 10000 ISYS=1,NS
         IF(INFORM(ISYS,1,1).EQ.0)                        GO TO 10001
         DO 10010 IMCS=1,NUMMCS(ISYS)
            IF(INFORM(ISYS,IMCS,1).EQ.0)                  GO TO 10011
            DO 10020 IBAS=1,10
               IF(INFORM(ISYS,IMCS,IBAS).EQ.0)            GO TO 10021
               IF(IBE(INFORM(ISYS,IMCS,IBAS),1).EQ.1)     GO TO 10010
10021        CONTINUE
10020       CONTINUE
            KOUNT=KOUNT+1
            MCSTYP(ISYS,IMCS)=2
10010    CONTINUE
10011    CONTINUE
         WRITE(21,1000) KOUNT,ISYS
         WRITE(21,1001)
      NUMRUN(ISYS)=NUMMCS(ISYS)-KOUNT
         KOUNT=0
10000 CONTINUE
10001 CONTINUE
      RETURN
C*************************************************************************
C                                                                        *
C                        FORMATS                                         *
C                                                                        *
C*************************************************************************
 1000  FORMAT('1 THERE ARE ',I3,' MCS''S IN SYSTEM ',I2,' WHICH  ')
 1001  FORMAT('0 REPRESENT A SYSTEM FAILURE TO START ONLY.')
 1010  FORMAT('0 THE COMMON BASIC EVENTS HAVE NOT BEEN DELETED BY ',
     1 'SUBROUTINE PREPAR.')
 1011  FORMAT('0 THE RESULT IS A LOWER BOUND TO THE "ENF".')
 1020  FORMAT('0 THE COMMON BASIC EVENTS HAVE BEEN DELETED BY ',
     1 'SUBROUTINE PREPAR.')
 1021  FORMAT('0 THE RESULT IS AN UPPER BOUND TO THE "ENF".')
 1030  FORMAT('1')
 1040  FORMAT('0')
 1050  FORMAT(' THE DELETING OF THE BASIC EVENTS WHICH ARE SHARED '
     1 ,'BY TWO OR MORE FAULT TREES HAS ')
 1051  FORMAT('0 CAUSED ONE OF THE TOP EVENTS TO EXIST.')
 1060  FORMAT('0 THE MCS # ',I3,' OF THE FAULT TREE # ',I2,' IS DOWN.')
 1070  FORMAT('0 EITHER THE FAULT TREE ORDER MUST BE CHANGED OR THE ',
     1 'FAULT TREE MUST NOT APPEAR IN THE INPUT DATA.')
 1080  FORMAT('0 THE PROGRAM WILL NOT BE EXECUTED.')
       END
```

```
      SUBROUTINE TIMES (IN)
C
C           THE SUBROUTINE "TIMES" CALCULATES AND STORES ALL THE
C     TIME POINTS AT WHICH THE ACCIDENT SEQUENCE  UNDEPENDABILITY
C     EXPERIENCES A DISCONTINUITY.  "TIMES" ALSO SELECTS      THE
C     NUMERICAL INTEGRATION METHOD TO BE USED IN THE  COMPUTATION
C     OF THE EXPECTED NUMBER OF FAILURES IN THE MAIN PROGRAM.   IF
C     THE NUMERICAL METHOD IS     SIMPSON'S RULE, "TIMES" COMPUTES
C     THE NUMBER OF DATA POINTS TO BE USED  IN  THE  INTEGRATION.
C     THE VARIABLES USED IN THIS SUBROUTINE ARE THE FOLLOWING:
C
C           DISDAT(I)        THIS VECTOR STORES THE TIME POINTS AT
C                            WHICH THE SYSTEM      UNDEPENDABILITY
C                            EXPERIENCES A DISCONTINUITY.
C
C           MOD              THE DISCONTINUITY TIME POINTS     FOR
C                            EACH BASIC EVENT REPEAT    THEMSELVES
C                            PERIODICALLY STARTING  ON  THE SECOND
C                            TEST INTERVAL.  "MOD" CONTAINS    THE
C                            MAXIMUM NUMBER OF SUCH REPETITIONS.
C
C           HELP             THIS IS AN AUXILIARY VARIABLE USED TO
C                            STORE THE DISCONTINUITY TIME   POINTS
C                            WHEN THE VECTOR    "DISDAT(I)"     IS
C                            ORDERED TO FORM AN INCREASING  SET OF
C                            DATA POINTS.
C
C           KFINAL           THE NUMBER OF NON-ZERO POINTS IN  THE
C                            VECTOR " DISDAT(I) ".
C
C           IHELP, AHELP,    THESE ARE AUXILIARY VARIABLES USED TO
C           AND RHELP        DETERMINE IF "NDP" IS ODD.
C
C           MM               A COUNTER USED WHEN LOADING DISDAT(I).
C
C           ALL OTHER VARIABLES AND VECTORS USED IN THIS SUBROUTINE
C     HAVE BEEN DEFINED IN THE MAIN PROGRAM.
C
      COMMON  INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON     IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON        NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON           MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON              TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON              DELTA1, DELTA2, BOUND ,   IDELTA
```

```
      COMMON              DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON              NBE, IND02, IND03, IND04 , IND05
      COMMON              IND10, IND09, IND08, IND07,IND06
      COMMON              JDELTA(3),              DLTPR (3)
      DIMENSION DISDAT(1000)
      IF(IN.NE.1) GO TO 10000
C
C     THE SUBROUTINE "TIMES" HAS BEEN CALLED FOR THE FIRST TIME
C
C
C
C                    INITIALIZE THE VECTOR DISDAT(I)
C
                     DO 15 I=1,1000
                     DISDAT(I)=0.0
 15                  CONTINUE
C
C     LOAD THE VECTOR DISDAT(I) WITH THE FIRST TEST INTERVAL
C
      J=1
      DO 1000 I=1,NBE
      IF(IBE(I,2).NE.4) GO TO 1000
      DISDAT(J)=RBE(I,3)
      DISDAT(J+1)=RBE(I,3)+RBE(I,4)
      DISDAT(J+2)=RBE(I,3)+RBE(I,4)+RBE(I,5)
      J=J+3
 1000 CONTINUE
      IF(J.EQ.1)                                GO TO 80000
C
C     IMPLEMENT THE VECTOR DISDAT(I)
C
      MM=1
      DO 20 I=1,10
      IF(IBE(I,2).NE.4) GO TO 20
      DISDAT(  J)=DISDAT(  MM)+RBE(I,2)
      DISDAT(J+1)=DISDAT(MM+1)+RBE(I,2)
      DISDAT(J+2)=DISDAT(MM+2)+RBE(I,2)
      MM=MM+3
      J=  J+3
      MOD=INT((TFINAL-RBE(I,3))/RBE(I,2))
      IF(MOD.LE.1) GO TO 20
      DO 30 K=1,MOD-1
      DISDAT(  J)=RBE(I,2)+DISDAT(J-3)
      DISDAT(J+1)=RBE(I,2)+DISDAT(J-2)
      DISDAT(J+2)=RBE(I,2)+DISDAT(J-1)
      J=J+3
 30   CONTINUE
 20   CONTINUE
C
C     SET TO ZERO THE UNNECESSARY TIME POINTS
C
```

```
                DO 40 I=1,J-1
                DO 50 II=I+1,J-1
                IF(DISDAT(II).NE.DISDAT(I)) GO TO 50
                DISDAT(II)=0.0
 50             CONTINUE
 40             CONTINUE
C
C       DELETE DATA POINTS GREATER THAN OR EQUAL TO TFINAL
C
                        DO 9000 K1=1,J
                        IF(DISDAT(K1).LT.TFINAL) GO TO 9000
                        DISDAT(K1)=0.0
 9000                   CONTINUE
C
C       COUNT THE NUMBER OF TIME POINTS
C
                                KK=0
                                DO 70 I=1,J-1
                                IF(DISDAT(I).EQ.0.0) GO TO 70
                                KK=KK+1
 70                             CONTINUE
C
C       DISREGARD THE ZEROS IN THE VECTOR
C
                        DO 80 I=1,J
                        IF(I.GE.KK) GO TO 80
 81                     IF(DISDAT(I).NE.0.0) GO TO 80
                        DO 90 II=I,J
                        DISDAT(II)=DISDAT(II+1)
 90                     CONTINUE
                        GO TO 81
 80                     CONTINUE
C
C       REORDER THE TIME POINTS TO FORM AN INCREASING SET
C
 100            L=0
                JJ=KK-1
                DO 110 I=1,JJ
                IF(DISDAT(I).LT.DISDAT(I+1)) GO TO 110
                HELP=DISDAT(I)
                DISDAT(I)=DISDAT(I+1)
                DISDAT(I+1)=HELP
                L=1
 110            CONTINUE
                IF(L.NE.0) GO TO 100
C
C       INPUT THE LAST ELEMENT IN VECTOR DISDAT(I)
C
        DISDAT(KK+1)=TFINAL
        KFINAL=KK+1
        GO TO 80100
80000   CONTINUE
```

```
      DISDAT(1)=TFINAL
80100 CONTINUE
C
C     PRINT THE VECTOR DISDAT(I)  (IF ASKED).
C
      IF(IND10.NE.1) GO TO 10600
            WRITE(21,10090)
            WRITE(21,10100)
            WRITE(21,10110)
            WRITE(21,10120)
            WRITE(21,10100)
            WRITE(21,10130)
            DO 10500 LL=1,KFINAL
                  WRITE(21,10140) DISDAT(LL)
10500       CONTINUE
10600 CONTINUE
C
C     SELECT THE FIRST INTEGRATION INTERVAL
C
                  FIRST=0.0
                  FINAL=DISDAT(1)
                  GO TO 115
C
C     "TIMES" HAS BEEN CALLED AT LEAST TWO "TIMES"
C
10000 CONTINUE
      FIRST=DISDAT(IN-1)
      FINAL=DISDAT(IN  )
 115  CONTINUE
C
C     SELECT THE NUMERICAL INTEGRATION METHOD
C
      DELTA=(FINAL-FIRST)
      IF(DELTA.LE.DELTA1) GO TO 120
      IF(DELTA.LE.DELTA2) GO TO 130
C*************************************************************************
C                                                                        *
C           SIMPSON'S RULE                                               *
C                                                                        *
C*************************************************************************
            NIM=3
C
C           SELECT THE NUMBER OF DATA POINTS
C
            NDP=INT(DELTA/DELTA2)+1
            AHELP=FLOAT(INT(NDP/2.))
            RHELP=NDP/2.
            IF(RHELP.NE.AHELP) GO TO 140
            NDP=NDP+1
 140        CONTINUE
            RETURN
```

```
C*************************************************************************
C                                                                        *
C                        RECTANGULAR RULE                                *
C                                                                        *
C*************************************************************************
 120                     CONTINUE
                         NIM=1
                         RETURN
C*************************************************************************
C                                                                        *
C                        TRAPEZOIDAL RULE                                *
C                                                                        *
C*************************************************************************
 130                             CONTINUE
                                 NIM=2
                                 RETURN
C
C       FORMATS
C
10090   FORMAT('1')
10100   FORMAT('0----------------------------------')
10110   FORMAT('0 THE DISCONTINUITY TIME POINTS')
10120   FORMAT('0 ( TIME IN HOURS )')
10130   FORMAT('0')
10140   FORMAT(1X,F10.5)
        END
```

```
      SUBROUTINE UNDEPE(ISYS,UI)
C
C           THE SUBROUTINE "UNDEPE" CONTROLS THE COMPUTATION  OF
C     THE TIME-DEPENDENT UNDEPENDABILITY FOR EACH SYSTEM  IN   THE
C     ACCIDENT SEQUENCE.  THIS IS ACCOMPLISHED  BY CALLING    SUB-
C     ROUTINES "CUTS" AND "RECKON" AND  BY PERFORMING  A  NUMERICAL
C     INTEGRATION USING     SIMPSON'S RULE.  THE NOTATION USED  IN
C     THIS SUBROUTINE IS THE FOLLOWING:
C
C
C           ISYS     THE SYSTEM IDENTIFICATION NUMBER (1,...,NS).
C
C           UI       THE SYSTEM UNDEPENDABILITY AT "TPRIME".
C
C           SUM      THIS VARIABLE HAS BEEN DEFINED IN SUBROUTINE
C                    "CUTS."
C
C           PDF(I)   THE PROBABILITY DENSITY FUNCTION   FOR   THE
C                    TIME TO FIRST SYSTEM FAILURE (SYSTEM "ISYS")
C
C           PFRO     AUXILIARY VARIABLES USED   IN  THE NUMERICAL
C           AND      INTEGRATION OF PDF(I).
C           PFRE
C
C           PFR      THE PROBABILITY OF A SYSTEM FAILURE TO   RUN
C                    GIVEN THAT A DEMAND OCCURS AT "TPRIME"   AND
C                    THAT THE SYSTEM STARTS.
C
C           PFS      THE PROBABILITY OF A SYSTEM FAILURE TO START
C                    ON DEMAND AT "TPRIME".
C
C
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),             DLTPR (3)
      DIMENSION PDF(101)
      EPS=.001
                  IF(((TPRIME+EPS).NE.8403.).AND.
     1               ((TPRIME+EPS).NE.8038.).AND.
```

```
     2                    ((TPRIME+EPS).NE.7673.).AND.
     3                    ((TPRIME-EPS).NE.8403.).AND.
     4                    ((TPRIME-EPS).NE.8038.).AND.
     5                    ((TPRIME+EPS).NE.7308.).AND.
     6                    ((TPRIME-EPS).NE.7308.).AND.
     7                    ((TPRIME+EPS).NE.6578.).AND.
     8                    ((TPRIME-EPS).NE.6578.).AND.
     9                    ((TPRIME-EPS).NE.7673.)) GO TO 44447
                       IF(TPRIME.LT.7670) GO TO 44447
                       DO 44444 I=1,NBE
                              WRITE(21,44445) I,BASE(I)
44444                  CONTINUE
44447                  CONTINUE
44445  FORMAT('0    BASE ( ',I4,' ) = ',E12.7)
C
C                              INITIALIZATION:
C
                               PFS=0.0
                               PFR=0.0
                               DO 20000 LL=1,101
                                 PDF(LL)=0.0
20000                          CONTINUE
C
C      COMPUTE THE PROBABILITY THAT THE SYSTEM FAILS TO START
C      AT T'.  THIS IS ACCOMPLISHED BY CALLING SUBROUTINE "CUTS."
C
       IF(IND08.NE.1)                                        GO TO 1000
       CALL CUTS (ISYS,SUM)
       PFS=SUM
       IF(IND05.EQ.1)                                        GO TO 1000
              UI=PFS
              RETURN
1000   CONTINUE
C
C      COMPUTE THE VALUE OF THE DENSITY FUNCTION AT EACH ONE OF
C      THE IDELTA TIME POINTS.
C
       DO 2000 I=1,IDELTA
       TDIFF=(I-1)*DELTAP
       IF(I.NE.IDELTA)                                       GO TO 2001
       TDIFF=TRUN(ISYS)
 2001  CONTINUE
       CALL RECKON (SUM,ISYS)
       PDF(I)=SUM
 2000  CONTINUE
C
C              IMPLEMENTATION OF  SIMPSON'S RULE
C
               PFRO=0.0
               PFRE=0.0
               DO 3000 LL=2,IDELTA-1,2
               PFRE=PFRE+PDF(LL)
```

```
 3000           CONTINUE
                IF(IDELTA.LT.5)                                  GO TO 3011
                DO 3010 LL=3,IDELTA-2,2
                PFRO=PFRO+PDF(LL)
 3010           CONTINUE
 3011           CONTINUE
C
C       COMPUTE THE PROBABILITY THAT THE SYSTEM FAILS TO RUN
C       DURING THE MISSION GIVEN THAT A DEMAND OCCURS AT  T',
C       AND THE SYSTEM STARTS AT T'.
C
        PFR=(DELTAP/3.)*(PDF(1)+4.*PFRE+2.*PFRO+PDF(IDELTA))
C
C       COMPUTE THE UNDEPENDABILITY AT T'.
C
        UI=PFS+PFR
        WRITE(21,55501) TPRIME,PFS,PFR
55501   FORMAT(' PFS AND PFR AT TIME ',F8.2,'=', E12.7,5X,E12.7)
        RETURN
        END
```

```
      SUBROUTINE BASIC1
C             SUBROUTINE "BASIC1" CONTROLS THE COMPUTATION OF  THE
C
C     THE BASIC EVENT UNAVAILABILITIES AT TIME "TPRIME" (T').  ALL
C
C     THE VARIABLES USED IN THIS SUBROUTINE HAVE   ALREADY    BEEN
C
C     DEFINED IN THE MAIN PROGRAM.
C
C
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),             DLTPR (3)
C
C                     INITIALIZE THE VECTOR BASE(1000).
C
                      DO 10000 I=1,NBE
                      BASE(I)=0.0
10000                 CONTINUE
C
C     SELECT THE BASIC EVENT (K).
C
      DO 20000 K=1,NBE
            IF(IBE(K,2).EQ.1) GO TO 20100
            IF(IBE(K,2).EQ.2) GO TO 20200
            IF(IBE(K,2).EQ.3) GO TO 20300
            IF(IBE(K,2).EQ.4) GO TO 20400
C
C     K IS A CONSTANT UNAVAILABILITY BASIC EVENT.
C
20100                                 CONTINUE
                                      BASE(K)=RBE(K,10)
                                      GO TO 20000
C
C     K IS A NONREPAIRABLE BASIC EVENT.
C
20200                         CONTINUE
                              BASE(K)=FNRA(RBE(K,1),TPRIME,RBE(K,10))
                              GO TO 20000
C
C     K IS A CONSTANTLY MONITORED BASIC EVENT.
C
20300                 CONTINUE
                      BASE(K)=FCMA(RBE(K,1),RBE(K,5),RBE(K,10))
                      GO TO 20000
```

```
C
C       K IS A PERIODICALLY TESTED BASIC EVENT.
C
20400           CONTINUE
                BASE(K)=FPTA(RBE(K,1),RBE(K,2),RBE(K,3),RBE(K,4),
     1          RBE(K,5),RBE(K,6),RBE(K,7),RBE(K,8),RBE(K,9),
     2          RBE(K,10),TPRIME)
C
20000   CONTINUE
C
C       THE UNAVAILABILITY HAS BEEN COMPUTED FOR 'NBE' BASIC EVENTS.
C       RETURN TO THE MAIN PROGRAM.
C
        RETURN
        END
```

```
      SUBROUTINE BASIC2 (NUMBER,BEUNAV)
C
C             THE SUBROUTINE "BASIC2" CONTROLS THE COMPUTATION OF
C
C     THE BASIC EVENT UNAVAILABILITIES AT TIME "DPRIME"      (T")
C
C     FOLLOWING THE OCCURRENCE OF THE INITIATING EVENT AT    TIME
C
C     "TPRIME" (T').  ALL THE VARIABLES AND VECTORS USED  IN THIS
C
C     SUBROUTINE HAVE ALREADY BEEN DEFINED IN THE MAIN PROGRAM.
C
C
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),   TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,    IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),              DLTPR (3)
C
C             WHAT IS THE TYPE OF BASIC EVENT?
C
      IF(IBE(NUMBER,1).NE.1)                          GO TO 11000
C
C     THE BASIC EVENT IS OF TYPE 1.  WHAT IS THE BASIC EVENT POST-
C     DEMAND CLASS?
C
             IF(IBE(NUMBER,3).EQ.2)                   GO TO 20000
C
C     THE BASIC EVENT POSTDEMAND CLASS IS 3:
C
C     THE BASIC EVENT IS REPAIRABLE (CONSTANTLY MONITORED).
C
             BEUNAV=FCME(RBE(NUMBER,11),RBE(NUMBER,12),BASE(NUM
     1       BER))
             RETURN
20000        CONTINUE
C
C     THE BASIC EVENT POSTDEMAND CLASS IS 2:
C
C     THE BASIC EVENT IS NONREPAIRABLE.
C
             BEUNAV=FNRE(RBE(NUMBER,11),TDIFF,BASE(NUMBER))
             RETURN
11000  CONTINUE
C
C     THE BASIC EVENT IS OF TYPE 2.
C
```

```
BEUNAV=BASE(NUMBER)
RETURN
END
```

```
      SUBROUTINE RECKON (SUM,ISYS)
C
C
C              SUBROUTINE "RECKON" COMPUTES THE PROBABILITY   OF  A
C
C     SYSTEM FAILURE TO RUN AT TIME "DPRIME" (T") GIVEN THAT   THE
C
C     THE INITIATING EVENT OCCURRED AT TIME "TPRIME" (T')      AND
C
C     THE SYSTEM STARTED AT "TPRIME" (T').
C
C
C
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,    IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),            DLTPR (3)
C*************************************************************************
C                                                                        *
C     'NUMRUN(ISYS)' MCS'S TAKEN ONE AT A TIME                           *
C                                                                        *
C*************************************************************************

      SUM1=0.0
         SUM2=0.0
      SUM3=0.0
      III=1
      DO 1020 J=1,NUMRUN(ISYS)
      IF(MCSTYP(ISYS,J).EQ.2) GO TO 1020
      IF(INFORM(ISYS,J,1).EQ.0) GO TO 1021
      UNAMCS=0.0
      DO 1030 K=1,5
      IF(INFORM(ISYS,J,K).EQ.0) GO TO 1040
C
C     SKIP THE BASIC EVENTS OF TYPE 2.  THEY WILL NOT FAIL IN DT".
C
      IF(IBE(INFORM(ISYS,J,K),1).EQ.2)        GO TO 1030
C
C     THE PROBABILITY THAT THE BASIC EVENT INFORM(ISYS,J,K) IS UP
C     AT T".
C
      CALL BASIC2(INFORM(ISYS,J,K),BEUNAV)
      IF(BEUNAV.EQ.1.) GO TO 1030
      PUNA=1.-BEUNAV
```

```
C
C       THE PROBABILITY THAT THE BASIC EVENT INFORM(ISYS,J,K) FAILS
C       IN DT" GIVEN THAT IT IS UP AT T".
C
        PUNA=PUNA*RBE(INFORM(ISYS,J,K),11)
C
C       THE PROBABILITY THAT ALL THE REMAINING BASIC EVENTS IN MCS
C       J ARE DOWN AT T".
C
        DO 1050 KP=1,5
        IF(KP.EQ.K) GO TO 1050
        IF(INFORM(ISYS,J,KP).EQ.0) GO TO 1060
        CALL BASIC2(INFORM(ISYS,J,KP),BEUNAV)
        PUNA=PUNA*BEUNAV
 1050   CONTINUE
 1060   CONTINUE
C
C       COMPUTE THE PROBABILITY THAT THE SYSTEM IS UP AT T".
C
        IF(IND03.EQ.1.AND.IND01.EQ.1.AND.NUMRUN(ISYS).GT.1) GO TO 1070
        GO TO 1080
 1070   CONTINUE
C
C       SELECT THE MPS'S WHICH CONTAIN THE BASIC EVENT INFORM(ISYS,J,K).
C
C
C                       INITIALIZE THE VECTOR MPSVCT(10,20):
C
                        DO 10705 INITI=1,10
                            DO 10706 INITI1=1,20
                                MPSVCT(INITI,INITI1)=0
10706                       CONTINUE
10705                   CONTINUE
        DO 10710 LL2=1,NUMMPS(ISYS)
        DO 10720 LL3=1,20
        IF(JNFORM(ISYS,LL2,LL3).EQ.0) GO TO 10710
        IF(JNFORM(ISYS,LL2,LL3).NE.INFORM(ISYS,J,K)) GO TO 10720
C
C       THE MPS 'LL2' CONTAINS THE BASIC EVENT INFORM(ISYS,J,K).
C
C
C               IS THE VECTOR MPSVCT(10,20) OVERLOADED?
C
                IF(III.GT.10) GO TO 1080
C
C       DOES THE MPS CONTAIN ANY OF THE REMAINING BASIC EVENTS IN
C       MCS 'J'?
C
        DO 10730 LL4=1,5
        IF(INFORM(ISYS,J,LL4).EQ.0)                         GO TO 10731
        IF(LL4.EQ.K) GO TO 10730
        DO 10740 LL5=1,20
```

```
      IF(INFORM(ISYS,J,LL4).NE.JNFORM(ISYS,LL2,LL5)) GO TO 10740
C
C     MPS LL2 IS DOWN AT T". SELECT THE NEXT MPS.
C
      GO TO 10710
10740 CONTINUE
10730 CONTINUE
10731 CONTINUE
C
C             IS THE MPS LL2 A 1 BASIC EVENT MPS?
C
                    INTAR=0
                    DO 10753 LLL6=1,20
                       IF(JNFORM(ISYS,LL2,LLL6).EQ.0)  GO TO 10754
                       INTAR=INTAR+1
10753               CONTINUE
10754               CONTINUE
                    IF(INTAR.EQ.1)                        GO TO 1080
C
C     MPS LL2 DOES NOT CONTAIN ANY OF THE REMAINING BASIC EVENTS
C     IN MCS 'J'. INSERT MPS LL2 IN THE VECTOR MPSVCT(10,20) AND
C     DELETE THE BASIC EVENT INFORM(ISYS,J,K).
C
      IDUMMY=0
      DO 10750 LL6=1,19
      IF(JNFORM(ISYS,LL2,LL6).EQ.0) GO TO 10709
      IF(IDUMMY.EQ.1) GO TO 10760
      IF(JNFORM(ISYS,LL2,LL6).EQ.INFORM(ISYS,J,K)) GO TO 10760
      MPSVCT(III,LL6)=JNFORM(ISYS,LL2,LL6)
      GO TO 10750
10760 CONTINUE
      IDUMMY=1
      MPSVCT(III,LL6)=JNFORM(ISYS,LL2,LL6+1)
10750 CONTINUE
      III=III+1
      GO TO 10710
10720 CONTINUE
10709 CONTINUE
      III=III+1
10710 CONTINUE
      IF(III.EQ.1)                                        GO TO 1080
C
C             CALL SUBROUTINE "PATHS."
C
              CALL PATHS(PROB,III-1)
              PUNA=PUNA*PROB
 1080         CONTINUE
      III=1
      UNAMCS=UNAMCS+PUNA
 1030 CONTINUE
 1040 CONTINUE
      SUM1=SUM1+UNAMCS
```

```
 1020  CONTINUE
 1021  CONTINUE
       IF(SUM1.EQ.0.)                                          GO TO 1022
       IF(NUMRUN(ISYS).EQ.1)                                   GO TO 1022
C
C      ARE THE SECOND AND THIRD SUMMATIONS REQUESTED?
C
       IF(IND04.EQ.1) GO TO 1090
       SUM=SUM1
       RETURN
C
C      ARE THE SECOND AND THIRD SUMMATIONS FEASIBLE?
C
 1090  CONTINUE
       IF(NUMRUN(ISYS).LE.20) GO TO 1100
       SUM=SUM1
       RETURN
C*************************************************************************
C                                                                        *
C      'NUMRUN(ISYS)' MCS'S TAKEN TWO AT A TIME                          *
C                                                                        *
C*************************************************************************
 1100  CONTINUE
       III=1
       SUM2=0.0
C
C      SELECT THE MCS'S I1 AND I2
C
       DO 2000 I1=1,NUMRUN(ISYS)-1
       IF(MCSTYP(ISYS,I1).EQ.2)   GO TO 2000
       DO 2010 I2=I1+1,NUMRUN(ISYS)
       IF(MCSTYP(ISYS,I2).EQ.2)   GO TO 2010
C
C      LOAD THE VECTOR IHELP(0-5)
C
C
C                              INITIALIZATION
C
                       DO 2020 KK=1,15
                       IHELP(KK)=0
 2020                  CONTINUE
                       IIV=0
C
C      FIND THE BASIC EVENTS COMMON TO THE TWO MCS'S.
C
       DO 2030 M1=1,5
       IF(INFORM(ISYS,I1,M1).EQ.0) GO TO 2031
       DO 2040 M2=1,5
       IF(INFORM(ISYS,I2,M2).EQ.0) GO TO 2030
       IF(INFORM(ISYS,I1,M1).NE.INFORM(ISYS,I2,M2)) GO TO 2040
       IIV=IIV+1
       IHELP(IIV)=INFORM(ISYS,I1,M1)
```

```
      GO TO 2030
 2040 CONTINUE
 2030 CONTINUE
 2031 CONTINUE
C
C                       COUNT THE NUMBER OF COMMON BASIC EVENTS
C
                        NUMBER=0
                        DO 2050 KK=1,5
                        IF(IHELP(KK).EQ.0) GO TO 2051
                        NUMBER=NUMBER+1
 2050                   CONTINUE
 2051                   CONTINUE
C
C       CHECK THE NUMBER OF COMMON BASIC EVENTS.
C
        IF(NUMBER.NE.0) GO TO 2060
C
C       THE MCS'S I1 AND I2 SHARE NO BASIC EVENTS.
C
        GO TO 2010
 2060   CONTINUE
C
C       THE MCS'S I1 AND I2 SHARE 'NUMBER' BASIC EVENTS.  LOAD THE
C       REMAINING PART OF THE VECTOR IHELP(6-15)
C
        KK=6
C
C               TAKE ALL THE BASIC EVENTS IN I1 WHICH DO NOT SHOW IN I2.
C
                DO 2070 K1=1,5
                IF(INFORM(ISYS,I1,K1).EQ.0) GO TO 2071
                DO 2080 K2=1,NUMBER
                IF(INFORM(ISYS,I1,K1).EQ.IHELP(K2)) GO TO 2070
 2080           CONTINUE
                IHELP(KK)=INFORM(ISYS,I1,K1)
                KK=KK+1
 2070           CONTINUE
 2071           CONTINUE
C
C               TAKE ALL THE BASIC EVENTS IN I2 WHICH DO NOT SHOW IN I1.
C
                DO 2090 K1=1,5
                IF(INFORM(ISYS,I2,K1).EQ.0) GO TO 2091
                DO 2100 K2=1,NUMBER
                IF(INFORM(ISYS,I2,K1).EQ.IHELP(K2)) GO TO 2090
 2100           CONTINUE
                IHELP(KK)=INFORM(ISYS,I2,K1)
                KK=KK+1
 2090           CONTINUE
 2091           CONTINUE
```

```
C
C                       INITIALIZE THE UNAVAILABILITY OF THE
C                       INTERSECTION OF THE MCS'S I1 AND I2.
C
                        UNAINT=0.0
C
      DO 2110 NC=1,NUMBER
C
C     SKIP THE BASIC EVENTS OF TYPE 2.  THEY WILL NOT FAIL IN DT".
C
      IF(IBE(IHELP(NC),1).EQ.2)                GO TO 2110
C
C     THE PROBABILITY THAT THE COMMON BASIC EVENT 'NC' IS UP AT T".
C
      CALL BASIC2(IHELP(NC),BEUNAV)
      IF(BEUNAV.EQ.1.) GO TO 2110
      PUNA=1.-BEUNAV
C
C     THE PROBABILITY THAT THE BASIC EVENT IHELP(NC) FAILS IN DT"
C     GIVEN THAT IT IS WORKING AT T".
C
      PUNA=PUNA*RBE(IHELP(NC),11)
C
C     THE PROBABILITY THAT ALL THE REMAINING BASIC EVENTS IN
C     IHELP(15) ARE DOWN AT T".
C
      DO 2120 NCP=1,15
      IF(NCP.EQ.NC) GO TO 2120
      IF(IHELP(NCP).EQ.0) GO TO 2120
      CALL BASIC2(IHELP(NCP),BEUNAV)
      PUNA=PUNA*BEUNAV
 2120 CONTINUE
C
C     THE PROBABILITY THAT THE SYSTEM IS UP AT T".
C
      IF(IND03.EQ.1.AND.IND01.EQ.1.AND.NUMRUN(ISYS).GT.2) GO TO 2130
      GO TO 2140
 2130 CONTINUE
C
C     SELECT THE MPS'S WHICH CONTAIN THE BASIC EVENTS 'NC'.
C
      DO 20000 LL1=1,NUMMPS(ISYS)
      DO 20010 LL2=1,20
      IF(JNFORM(ISYS,LL1,LL2).EQ.0) GO TO 20000
      IF(JNFORM(ISYS,LL1,LL2).NE.IHELP(NC)) GO TO 20010
C
C     THE MPS 'LL1' CONTAINS THE BASIC EVENT 'NC'.
C
C
C             IS THE VECTOR MPSVCT(10,20) OVERLOADED?
C
```

```
                IF(III.GT.10) GO TO 2140
C
C       DOES THE MPS 'LL1' CONTAINS ANY OF THE REMAINING BASIC
C       EVENTS IN IHELP(30)?
C
        DO 20020 LL4=1,15
        IF(LL4.EQ.NC) GO TO 20020
        DO 20030 LL5=1,20
        IF(IHELP(LL4).NE.JNFORM(ISYS,LL1,LL5)) GO TO 20030
C
C       THE MPS 'LL1' IS DOWN AT T".  SELECT THE NEXT MPS.
C
        GO TO 20000
20030   CONTINUE
20020   CONTINUE
C
C       THE MPS 'LL1' DOES NOT CONTAIN ANY OF THE REMAINING BASIC
C       EVENTS IN IHELP(15).  INSERT MPS 'LL1' IN     THE  VECTOR
C       MPSVCT(10,20) AND DELETE THE BASIC EVENT IHELP(NC).
C
        IDUMMY=0
        DO 20040 LL6=1,19
        IF(JNFORM(ISYS,LL1,LL6).EQ.0) GO TO 20000
        IF(IDUMMY.EQ.1) GO TO 20060
        IF(JNFORM(ISYS,LL1,LL6).EQ.IHELP(NC)) GO TO 20060
        MPSVCT(III,LL6)=JNFORM(ISYS,LL1,LL6)
        GO TO 20040
20060   CONTINUE
        IDUMMY=1
        MPSVCT(III,LL6)=JNFORM(ISYS,LL1,LL6+1)
20040   CONTINUE
        III=III+1
        GO TO 20000
20010   CONTINUE
20000   CONTINUE
C
C               CALL SUBROUTINE PATHS
C
                CALL PATHS(PROB,III-1)
                PUNA=PUNA*PROB
 2140           CONTINUE
        UNAINT=UNAINT+PUNA
 2110   CONTINUE
        SUM2=SUM2+UNAINT
 2010   CONTINUE
 2000   CONTINUE
        IF(NUMRUN(ISYS).NE.2)                                   GO TO 2002
        SUM=SUM1-SUM2
        RETURN
2002    CONTINUE
C
C       IS THE THIRD SUMMATION NECESSARY?
C
```

```
      BOUNDP=(SUM2/(SUM1-SUM2))*100.
      IF(BOUNDP.GT.BOUND.AND.BOUNDP.GT.0.)          GO TO 3000
      SUM=SUM1
      RETURN
C
C*************************************************************************
C                                                                        *
C     'NUMRUN(ISYS)' MCS'S TAKEN THREE AT A TIME                         *
C                                                                        *
C*************************************************************************
 3000 CONTINUE
      SUM3=0.0
C
C     SELECT THE MCS'S I1, I2, AND I3.
C
      DO 3010 I1=1,NUMRUN(ISYS)-2
      IF(MCSTYP(ISYS,I1).EQ.2) GO TO 3010
      DO 3020 I2=I1+1,NUMRUN(ISYS)-1
      IF(MCSTYP(ISYS,I2).EQ.2) GO TO 3020
      DO 3030 I3=I2+1,NUMRUN(ISYS)
      IF(MCSTYP(ISYS,I3).EQ.2) GO TO 3030
C
C     LOAD THE VECTOR IHELP(0-5) WITH THE BASIC EVENTS COMMON
C
C     TO MCS'S I1, I2, AND I3:
C
C
C                      INITIALIZATION
C
                       DO 3040 KK=1,20
                       IHELP(KK)=0
 3040                  CONTINUE
                       IIV=0
C
      DO 3050 M1=1,5
      IF(INFORM(ISYS,I1,M1).EQ.0) GO TO 3051
      DO 3060 M2=1,5
      IF(INFORM(ISYS,I2,M2).EQ.0) GO TO 3050
      IF(INFORM(ISYS,I1,M1).NE.INFORM(ISYS,I2,M2)) GO TO 3060
      DO 3070 M3=1,5
      IF(INFORM(ISYS,I3,M3).EQ.0) GO TO 3050
      IF(INFORM(ISYS,I1,M1).NE.INFORM(ISYS,I3,M3)) GO TO 3070
      IIV=IIV+1
      IHELP(IIV)=INFORM(ISYS,I1,M1)
      GO TO 3050
 3070 CONTINUE
      GO TO 3050
 3060 CONTINUE
 3050 CONTINUE
 3051 CONTINUE
C
C                      COUNT THE NUMBER OF COMMON BASIC EVENTS.
C
```

```
                        NUMBER=0
                        DO 3080 KK=1,5
                        IF(IHELP(KK).EQ.0) GO TO 3081
                        NUMBER=NUMBER+1
 3080                   CONTINUE
 3081                   CONTINUE
C
C       CHECK THE  NUMBER OF COMMON BASIC EVENTS.
C
        IF(NUMBER.NE.0) GO TO 3090
C
C       THE MCS'S I1, I2, AND I3 SHARE NO BASIC EVENTS.
C
        GO TO 3030
 3090   CONTINUE
C
C       THE MCS'S I1, I2, AND I3 SHARE 'NUMBER' BASIC EVENTS.  LOAD
C       THE REMAINING PART OF THE VECTOR IHELP(20).
C
        DO 3100 K1=1,5
                IHELP(K1+ 5)=INFORM(ISYS,I1,K1)
                IHELP(K1+10)=INFORM(ISYS,I2,K1)
                IHELP(K1+15)=INFORM(ISYS,I3,K1)
3100    CONTINUE
C
C       DELETE ALL THE REDUNDANT BASIC EVENTS IN THE VECTOR
C       IHELP(20).
C
        DO 3110 L1=1,19
                IF(IHELP(L1).EQ.0)              GO TO 3110
                DO 3120 L2=L1+1,20
                        IF(IHELP(L2).EQ.0)      GO TO 3120
                        IF(IHELP(L1).NE.IHELP(L2))
     1                                         GO TO 3120
                        IHELP(L2)=0
3120            CONTINUE
3110    CONTINUE
C
C                       INITIALIZE THE UNAVAILABILITY    OF    THE
C                       INTERSECTION OF THE MCS'S I1, I2, AND  I3.
C
                        UNAINT=0.0
C
        DO 3180 NC=1,NUMBER
        IF(IHELP(NC).EQ.0)                      GO TO 3180
C
C       SKIP THE BASIC EVENTS OF TYPE 2.  THEY WILL NOT FAIL IN DT".
C
        IF(IBE(IHELP(NC),1).EQ.2)               GO TO 3180
C
C       THE PROBABILITY THAT THE COMMON BASIC EVENT 'NC' IS WORKING
C       AT T".
C
```

```
      CALL BASIC2(IHELP(NC),BEUNAV)
      IF(BEUNAV.EQ.1.) GO TO 3180
      PUNA=1.-BEUNAV
C
C     THE PROBABILITY THAT THE BASIC EVENT IHELP(NC) FAILS IN DT"
C     GIVEN THAT IT IS WORKING AT T".
C
      PUNA=PUNA*RBE(IHELP(NC),11)
C
C     THE PROBABILITY THAT ALL THE REMAINING BASIC   EVENTS   IN
C     IHELP(20) ARE DOWN AT T".
C
      DO 3190 NCP=1,20
      IF(NCP.EQ.NC) GO TO 3190
      IF(IHELP(NCP).EQ.0) GO TO 3190
      CALL BASIC2(IHELP(NCP),BEUNAV)
      PUNA=PUNA*BEUNAV
 3190 CONTINUE
C
C     THE PROBABILITY THAT THE SYSTEM IS WORKING AT T".
C
      IF(IND03.EQ.1.AND.IND01.EQ.1.AND.NUMRUN(ISYS).GT.3) GO TO 3200
      GO TO 3210
 3200 CONTINUE
C
C     SELECT THE MPS'S WHICH CONTAIN THE BASIC EVENT 'NC'.
C
      III=1
      DO 30000 LL1=1,NUMMPS(ISYS)
      DO 30010 LL2=1,20
      IF(JNFORM(ISYS,LL1,LL2).EQ.0) GO TO 30000
      IF(JNFORM(ISYS,LL1,LL2).NE.IHELP(NC)) GO TO 30010
C
C     THE MPS 'LL1' CONTAINS THE BASIC EVENT 'NC'.
C
C
C             IS THE VECTOR MPSVCT(10,20) OVERLOADED?
C
              IF(III.GT.10) GO TO 3210
C
C     DOES MPS 'LL1' CONTAIN ANY OF THE REMAINING BASIC EVENTS IN
C     IHELP(20)?
C
      DO 30020 LL4=1,20
      IF(LL4.EQ.NC) GO TO 30020
      IF(IHELP(LL4).EQ.0)                        GO TO 30020
      DO 30030 LL5=1,20
      IF(JNFORM(ISYS,LL1,LL5).EQ.0)              GO TO 30020
      IF(IHELP(LL4).NE.JNFORM(ISYS,LL1,LL5)) GO TO 30030
C
C     THE MPS 'LL1' IS DOWN AT T".  SELECT THE NEXT MPS.
C
```

```
      GO TO 30000
30030 CONTINUE
30020 CONTINUE
C
C     THE MPS 'LL1' DOES NOT CONTAIN ANY OF THE REMAINING BASIC
C     EVENTS IN IHELP(20).  INSERT MPS 'LL1'   IN   THE  VECTOR
C     MPSVCT(10,20) AND DELETE THE BASIC EVENT IHELP(NC).
C
      IDUMMY=0
      DO 30040 LL6=1,19
      IF(JNFORM(ISYS,LL1,LL6).EQ.0) GO TO 30000
      IF(IDUMMY.EQ.1) GO TO 30060
      IF(JNFORM(ISYS,LL1,LL6).EQ.IHELP(NC)) GO TO 30060
      MPSVCT(III,LL6)=JNFORM(ISYS,LL1,LL6)
      GO TO 30040
30060 CONTINUE
      IDUMMY=1
      MPSVCT(III,LL6)=JNFORM(ISYS,LL1,LL6+1)
30040 CONTINUE
      III=III+1
      GO TO 30000
30010 CONTINUE
30000 CONTINUE
C
C             CALL SUBROUTINE "PATHS."
C
              CALL PATHS (PROB,III-1)
              PUNA=PUNA*PROB
 3210         CONTINUE
      UNAINT=UNAINT+PUNA
 3180 CONTINUE
      SUM3=SUM3+UNAINT
 3030 CONTINUE
 3020 CONTINUE
 3010 CONTINUE
 3011 CONTINUE
C
C     COMPUTE THE SECOND UPPER BOUND ( IT INCLUDES THE FIRST THREE
C     SUMMATIONS SUM1, SUM2, AND SUM3 ).
C
 1022 CONTINUE
      SUM=SUM1-SUM2+SUM3
      RETURN
      END
```

```
      SUBROUTINE PATHS (PROB,NMPS)
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),              DLTPR (3)
      DIMENSION IPATH(200)
      DOUBLE PRECISION SUM1, SUM2, SUM3, SUM4, SUM5, SUM6, SUM7
      DOUBLE PRECISION PROB,                   SUM8, SUM9, SUM10
C
C                     INITIALIZATION OF THE SUMMATIONS
C
                      SUM1=0.0
                       SUM2=0.0
                        SUM3=0.0
                         SUM4=0.0
                          SUM5=0.0
                          SUM6=0.0
                         SUM7=0.0
                        SUM8=0.0
                       SUM9=0.0
                      SUM10=.0
C
C             INITIALIZE THE COUNTER IND
C
              IND=1
10            CONTINUE
C
C     SELECT THE MINIMAL PATH SETS
C
      DO 100 I1=1   ,(NMPS-(IND-1))
      IF(IND.EQ.1) GO TO 1100
      DO 200 I2=I1+1,(NMPS-(IND-2))
      IF(IND.EQ.2) GO TO 2100
      DO 300 I3=I2+1,(NMPS-(IND-3))
      IF(IND.EQ.3) GO TO 3100
      DO 400 I4=I3+1,(NMPS-(IND-4))
      IF(IND.EQ.4) GO TO 4100
      DO 500 I5=I4+1,(NPMS-(IND-5))
      IF(IND.EQ.5) GO TO 5100
      DO 600 I6=I5+1,(NMPS-(IND-6))
      IF(IND.EQ.6) GO TO 6100
      DO 700 I7=I6+1,(NMPS-(IND-7))
      IF(IND.EQ.7) GO TO 7100
      DO 800 I8=I7+1,(NMPS-(IND-8))
      IF(IND.EQ.8) GO TO 8100
      DO 900 I9=I8+1,(NMPS-(IND-9))
```

```
      IF(IND.EQ.9) GO TO 9100
      I10=10
C*************************************************************************
C                                                                        *
C     THE MPS'S TAKEN TEN AT A TIME                                      *
C                                                                        *
C*************************************************************************
C
C     LOAD THE VECTOR IPATH(200).
C
                              DO 10200 K=1,20
                              IPATH(K    )=MPSVCT(I1 ,K)
                              IPATH(K+ 20)=MPSVCT(I2 ,K)
                              IPATH(K+ 40)=MPSVCT(I3 ,K)
                              IPATH(K+ 60)=MPSVCT(I4 ,K)
                              IPATH(K+ 80)=MPSVCT(I5 ,K)
                              IPATH(K+100)=MPSVCT(I6 ,K)
                              IPATH(K+120)=MPSVCT(I7 ,K)
                              IPATH(K+140)=MPSVCT(I8 ,K)
                              IPATH(K+160)=MPSVCT(I9 ,K)
                              IPATH(K+180)=MPSVCT(I10,K)
10200                         CONTINUE
C
C     SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                    DO 10300 L1=1,199
                    IF(IPATH(L1).EQ.0) GO TO 10300
                    DO 10400 L2=L1+1,200
                    IF(IPATH(L1).NE.IPATH(L2)) GO TO 10400
                    IPATH(L2)=0
10400               CONTINUE
10300               CONTINUE
C
C     COMPUTE THE AVAILABILITY OF THE INTERSECTION OF MPS'S I1,I2,
C     I3,I4,I5,I6,I7,I8,I9, AND I10
C
            AV=1.
            DO 10500 M1=1,200
            IF(IPATH(M1).EQ.0) GO TO 10500
            CALL BASIC2(IPATH(M1),BEUNAV)
            AV=AV*(1.-BEUANV)
10500       CONTINUE
C
C     COMPUTE THE TENTH SUMMATION
C
      SUM10=SUM10+AV
      GO TO 900
C*************************************************************************
C                                                                        *
C     THE MPS'S TAKEN NINE AT A TIME                                     *
C                                                                        *
```

```
C*************************************************************************
 9100  CONTINUE
C
C       LOAD THE VECTOR IPATH(180)
C
                                DO 9200 K=1,20
                                IPATH(K    )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
                                IPATH(K+ 40)=MPSVCT(I3,K)
                                IPATH(K+ 60)=MPSVCT(I4,K)
                                IPATH(K+ 80)=MPSVCT(I5,K)
                                IPATH(K+100)=MPSVCT(I6,K)
                                IPATH(K+120)=MPSVCT(I7,K)
                                IPATH(K+140)=MPSVCT(I8,K)
                                IPATH(K+160)=MPSVCT(I9,K)
 9200                           CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 9300 L1=1,179
                        IF(IPATH(L1).EQ.0) GO TO 9300
                        DO 9400 L2=L1+1,180
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 9400
                        IPATH(L2)=0
 9400                   CONTINUE
 9300                   CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF MPS'S I1,I2,
C       I3,I4,I5,I6,I7,I8, AND I9
C
                AV=1.
                DO 9500 M1=1,180
                IF(IPATH(M1).EQ.0) GO TO 9500
                CALL BASIC2(IPATH(M1),BEUNAV)
                AV=AV*(1.-BEUNAV)
 9500           CONTINUE
C
C       COMPUTE THE NINTH SUMMATION
C
        SUM9=SUM9+AV
  900   CONTINUE
        GO TO 800
C*************************************************************************
C                                                                        *
C       THE MPS'S TAKEN EIGHT AT A TIME                                  *
C                                                                        *
C*************************************************************************
 8100  CONTINUE
C
C       LOAD THE VECTOR IPATH(160)
C
```

```
                                DO 8200 K=1,20
                                IPATH(K    )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
                                IPATH(K+ 40)=MPSVCT(I3,K)
                                IPATH(K+ 60)=MPSVCT(I4,K)
                                IPATH(K+ 80)=MPSVCT(I5,K)
                                IPATH(K+100)=MPSVCT(I6,K)
                                IPATH(K+120)=MPSVCT(I7,K)
                                IPATH(K+140)=MPSVCT(I8,K)
 8200                           CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 8300 L1=1,159
                        IF(IPATH(L1).EQ.0) GO TO 8300
                        DO 8400 L2=L1+1,160
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 8400
                        IPATH(L2)=0
 8400                   CONTINUE
 8300                   CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF MPS'S I1,I2,
C       I3,I4,I5,I6,I7, AND I8.
               AV=1.
               DO 8500 M1=1,160
               IF(IPATH(M1).EQ.0) GO TO 8500
               CALL BASIC2(IPATH(M1),BEUNAV)
               AV=AV*(1.-BEUNAV)
 8500          CONTINUE
C
C       COMPUTE THE EIGHTH SUMMATION
C
        SUM8=SUM8+AV
   800  CONTINUE
        GO TO 700
C*************************************************************************
C                                                                        *
C       THE MPS'S TAKEN SEVEN AT A TIME                                  *
C                                                                        *
C*************************************************************************
 7100   CONTINUE
C
C       LOAD THE VECTOR IPATH(140)
C
                                DO 7200 K=1,20
                                IPATH(K    )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
                                IPATH(K+ 40)=MPSVCT(I3,K)
                                IPATH(K+ 60)=MPSVCT(I4,K)
                                IPATH(K+ 80)=MPSVCT(I5,K)
                                IPATH(K+100)=MPSVCT(I6,K)
                                IPATH(K+120)=MPSVCT(I7,K)
```

```
 7200                          CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 7300 L1=1,139
                        IF(IPATH(L1).EQ.0) GO TO 7300
                        DO 7400 L2=L1+1,140
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 7400
                        IPATH(L2)=0
 7400                   CONTINUE
 7300                   CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF MPS'S I1,I2,
C       I3,I4,I5,I6, AND I7.
C
               AV=1.
               DO 7500 M1=1,140
               IF(IPATH(M1).EQ.0) GO TO 7500
               CALL BASIC2(IPATH(M1),BEUNAV)
               AV=AV*(1.-BEUNAV)
 7500          CONTINUE
C
C       COMPUTE THE SEVENTH SUMMATION
C
       SUM7=SUM7+AV
  700  CONTINUE
       GO TO 600
C*************************************************************************
C                                                                        *
C       THE MPS'S TAKEN SIX AT A TIME                                    *
C                                                                        *
C*************************************************************************
 6100  CONTINUE
C
C       LOAD THE VECTOR IPATH(120)
C
                                DO 6200 K=1,20
                                IPATH(K    )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
                                IPATH(K+ 40)=MPSVCT(I3,K)
                                IPATH(K+ 60)=MPSVCT(I4,K)
                                IPATH(K+ 80)=MPSVCT(I5,K)
                                IPATH(K+100)=MPSVCT(I6,K)
 6200                           CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 6300 L1=1,119
                        IF(IPATH(L1).EQ.0) GO TO 6300
                        DO 6400 L2=L1+1,120
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 6400
                        IPATH(L2)=0
```

```
 6400                  CONTINUE
 6300                  CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF MPS'S I1,I2,
C       I3,I4,I5, AND I6.
C
               AV=1.
               DO 6500 M1=1,120
               IF(IPATH(M1).EQ.0) GO TO 6500
               CALL BASIC2(IPATH(M1),BEUNAV)
               AV=AV*(1.-BEUNAV)
 6500          CONTINUE
C
C       COMPUTE THE SIXTH SUMMATION
C
        SUM6=SUM6+AV
   600  CONTINUE
        GO TO 500
C*************************************************************************
C                                                                        *
C       THE MPS'S TAKEN FIVE AT A TIME                                   *
C                                                                        *
C*************************************************************************
 5100   CONTINUE
C
C       LOAD THE VECTOR IPATH(100)
C
                              DO 5200 K=1,20
                              IPATH(K    )=MPSVCT(I1,K)
                              IPATH(K+ 20)=MPSVCT(I2,K)
                              IPATH(K+ 40)=MPSVCT(I3,K)
                              IPATH(K+ 60)=MPSVCT(I4,K)
                              IPATH(K+ 80)=MPSVCT(I5,K)
 5200                         CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 5300 L1=1,99
                        IF(IPATH(L1).EQ.0) GO TO 5300
                        DO 5400 L2=L1+1,100
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 5400
                        IPATH(L2)=0
 5400                   CONTINUE
 5300                   CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF THE MPS'S
C       I1,I2,I3,I4, AND I5
C
               AV=1.
               DO 5500 M1=1,100
               IF(IPATH(M1).EQ.0) GO TO 5500
               CALL BASIC2(IPATH(M1),BEUNAV)
```

```
                AV=AV*(1.-BEUNAV)
 5500           CONTINUE
C
C       COMPUTE THE FIFTH SUMMATION
C
        SUM5=SUM5+AV
  500   CONTINUE
        GO TO 400
C*************************************************************************
C                                                                        *
C       THE MPS'S TAKEN FOUR AT A TIME                                   *
C                                                                        *
C*************************************************************************
 4100   CONTINUE
C
C       LOAD THE VECTOR IPATH(80)
C
                                DO 4200 K=1,20
                                IPATH(K    )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
                                IPATH(K+ 40)=MPSVCT(I3,K)
                                IPATH(K+ 60)=MPSVCT(I4,K)
 4200                           CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 4300 L1=1,79
                        IF(IPATH(L1).EQ.0) GO TO 4300
                        DO 4400 L2=L1+1,80
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 4400
                        IPATH(L2)=0
 4400                   CONTINUE
 4300                   CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF THE MPS'S
C       I1,I2,I3, AND I4
C
                AV=1.
                DO 4500 M1=1,80
                IF(IPATH(M1).EQ.0) GO TO 4500
                CALL BASIC2(IPATH(M1),BEUNAV)
                AV=AV*(1.-BEUNAV)
 4500           CONTINUE
C
C       COMPUTE THE FOURTH SUMMATION
C
        SUM4=SUM4+AV
  400   CONTINUE
        GO TO 300
```

```
C*************************************************************************
C                                                                        *
C      THE MPS'S TAKEN THREE AT A TIME                                   *
C                                                                        *
C*************************************************************************
 3100  CONTINUE
C
C      LOAD THE VECTOR IPATH(60)
C
                                DO 3200 K=1,20
                                IPATH(K     )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
                                IPATH(K+ 40)=MPSVCT(I3,K)
 3200                           CONTINUE
C
C      SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                         DO 3300 L1=1,59
                         IF(IPATH(L1).EQ.0) GO TO 3300
                         DO 3400 L2=L1+1,60
                         IF(IPATH(L1).NE.IPATH(L2)) GO TO 3400
                         IPATH(L2)=0
 3400                    CONTINUE
 3300                    CONTINUE
C
C      COMPUTE THE AVAILABILITY OF THE INTERSECTION OF THE MPS'S
C      I1,I2, AND I3.
C
               AV=1.
               DO 3500 M1=1,60
               IF(IPATH(M1).EQ.0) GO TO 3500
               CALL BASIC2(IPATH(M1),BEUNAV)
               AV=AV*(1.-BEUNAV)
 3500          CONTINUE
C
C      COMPUTE THE THIRD SUMMATION
C
       SUM3=SUM3+AV
  300  CONTINUE
       GO TO 200
C*************************************************************************
C                                                                        *
C      THE MPS'S TAKEN TWO AT A TIME                                     *
C                                                                        *
C*************************************************************************
 2100  CONTINUE
C
C      LOAD THE VECTOR IPATH(40)
C
                                DO 2200 K=1,20
                                IPATH(K     )=MPSVCT(I1,K)
                                IPATH(K+ 20)=MPSVCT(I2,K)
```

```
 2200                          CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 2300 L1=1,39
                        IF(IPATH(L1).EQ.0) GO TO 2400
                        DO 2500 L2=21,40
                        IF(IPATH(L2).EQ.0) GO TO 2300
                        IF(IPATH(L1).NE.IPATH(L2)) GO TO 2500
                        IPATH(L2)=0
 2500                   CONTINUE
 2300                   CONTINUE
 2400  CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE INTERSECTION OF THE MPS'S
C       I1 AND I2.
C
               AV=1.
               DO 2600 M1=1,40
               IF(IPATH(M1).EQ.0) GO TO 2600
               CALL BASIC2(IPATH(M1),BEUNAV)
               AV=AV*(1.-BEUNAV)
 2600          CONTINUE
C
C       COMPUTE THE SECOND SUMMATION
C
       SUM2=SUM2+AV
  200  CONTINUE
       GO TO 100
C*************************************************************************
C                                                                        *
C       THE MPS'S TAKEN ONE AT A TIME                                    *
C                                                                        *
C*************************************************************************
 1100  CONTINUE
C
C       LOAD THE VECTOR IPATH(20)
C
                                DO 1200 K=1,20
                                IF(MPSVCT(I1,K).EQ.0) GO TO 1300
                                IPATH(K)=MPSVCT(I1,K)
 1200                           CONTINUE
 1300                           CONTINUE
C
C       COMPUTE THE AVAILABILITY OF THE MPS I1
C
                        AV=1.
                        DO 1400 M1=1,20
                        IF(IPATH(M1).EQ.0) GO TO 1500
                        CALL BASIC2(IPATH(M1),BEUNAV)
                        AV=AV*(1.-BEUNAV)
```

```
 1400                    CONTINUE
 1500                    CONTINUE
C
C       COMPUTE THE FIRST SUMMATION
C
        SUM1=SUM1+AV
  100   CONTINUE
        IND=IND+1
        IF(IND.LE.NMPS) GO TO 10
C
C       COMPUTE THE PROBABILITY THAT THE SYSTEM IS WORKING AT T" GIVEN
C       THAT THE BASIC EVENT K CAUSES A SYSTEM FAILURE IN DT".
C
        PROB=SUM1-SUM2+SUM3-SUM4+SUM5-SUM6+SUM7-SUM8+SUM9-SUM10
        IF(PROB.GT.1.)                              GO TO 101
        RETURN
101     CONTINUE
        WRITE(21,1000) TPRIME,TDIFF
        WRITE(21,1010) PROB
        PROB=1.
        RETURN
1000    FORMAT('1THE PROBABILITY THAT THE SYSTEM IS UP HAS EXCEEDED THE
     1 LIMIT 1. ( TPRIME=',F10.3,'  AND TDIFF=',F10.5)
1010    FORMAT('0THE PROBABILITY THAT THE SYSTEM IS UP AT THE GIVEN TIME
     1S IS ',E10.5)
        END
```

```
      SUBROUTINE CUTS (ISYS,SUM)
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),              DLTPR (3)
C
C                     INITIALIZE THE SUMMATIONS.
C
                      SUM1=0.0
                        SUM2=0.0
                          SUM3=0.0
                        SUM4=0.0
                      SUM5=0.0
C
C
C*************************************************************************
C                                                                        *
C     'NUMMCS(ISYS)' MCS'S TAKEN ONE AT A TIME                           *
C                                                                        *
C*************************************************************************
      DO 120 J=1,NUMMCS(ISYS)
             UNAMCS=1.
             DO 130 K=1,5
                    IF(INFORM(ISYS,J,K).EQ.0) GO TO 140
                    UNAMCS=UNAMCS*BASE(INFORM(ISYS,J,K))
 130         CONTINUE
 140         CONTINUE
      IF((TPRIME+.001).NE.8038.) GO TO 77770
77770 CONTINUE
             SUM1=SUM1+UNAMCS
 120  CONTINUE
C
C     FIRST UPPER BOUND
C
      SUM=SUM1
      IF(NUMMCS(ISYS).EQ.1)                    GO TO 251
C
C     ARE THE SECOND AND THIRD SUMMATIONS REQUESTED?
C
      IF(IND02.EQ.1) GO TO 150
      GO TO 251
C
C     ARE THE SECOND AND THIRD SUMMATIONS FEASIBLE?
C
 150  CONTINUE
      IF(NUMMCS(ISYS).LE.20)          GO TO 160
      GO TO 251
```

```
C*************************************************************************
C                                                                        *
C      'NUMMCS(ISYS)' MCS'S TAKEN TWO AT A TIME                          *
C                                                                        *
C*************************************************************************
 160   CONTINUE
C
C      SELECT THE MCS'S 'I1' AND 'I2'.
C

       DO 170 I1=1,NUMMCS(ISYS)-1
       DO 180 I2=I1+1,NUMMCS(ISYS)
C
C                              LOAD THE VECTOR IHELP(10).
C
                               DO 190 I3=1,5
                               IHELP(I3   )=INFORM(ISYS,I1,I3)
                               IHELP(I3+5)=INFORM(ISYS,I2,I3)
 190                           CONTINUE
C
C      SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                      DO 200 J1=1,5
                      DO 210 J2=6,10
                      IF(IHELP(J1).EQ.0) GO TO 220
                      IF(IHELP(J1).NE.IHELP(J2)) GO TO 210
                      IHELP(J2)=0
 210                  CONTINUE
 200                  CONTINUE
 220                  CONTINUE
C
C      COMPUTE THE UNAVAILABILITY OF THE INTERSECTION OF MCS'S I1
C      AND I2 -- UNAINT
C
              UNAINT=1.
              DO 230 L1=1,10
              IF(IHELP(L1).EQ.0) GO TO 230
              UNAINT=UNAINT*BASE(IHELP(L1))
 230          CONTINUE
       IF((TPRIME+.001).NE.8038.) GO TO 77771
77771  CONTINUE
C
C      COMPUTE THE UNAVAILABILITY OF THE INTERSECTION OF THE MCS'S
C      TAKEN TWO AT A TIME.
C
       SUM2=SUM2+UNAINT
 180   CONTINUE
 170   CONTINUE
       IF(NUMMCS(ISYS).NE.2)                              GO TO 171
       SUM=SUM1-SUM2
       GO TO 251
  171  CONTINUE
```

```
C
C       IS THE THIRD SUMMATION NECESSARY?
C
        IF(SUM1.NE.SUM2)                                        GO TO 239
        BOUNDP=100.0
        GO TO 238
 239    CONTINUE
        BOUNDP=(SUM2/(SUM1-SUM2))*100.
 238    CONTINUE
        IF((BOUNDP.GT.BOUND).AND.(BOUNDP.GT.0.))                GO TO 240
        GO TO 251
C*************************************************************************
C                                                                        *
C       'NUMMCS(ISYS)' MCS'S TAKEN THREE AT A TIME                       *
C                                                                        *
C*************************************************************************
 240    CONTINUE
C
C       SELECT THE MCS'S 'M1', 'M2', AND 'M3'.
C
        DO 250 M1=1,NUMMCS(ISYS)-2
        DO 260 M2=M1+1,NUMMCS(ISYS)-1
        DO 270 M3=M2+1,NUMMCS(ISYS)
C
C                               LOAD THE VECTOR IHELP(15).
C
                                DO 280 M4=1,5
                                IHELP(M4   )=INFORM(ISYS,M1,M4)
                                IHELP(M4+5)=INFORM(ISYS,M2,M4)
                                IHELP(M4+10)=INFORM(ISYS,M3,M4)
 280                            CONTINUE
C
C       SET TO ZERO THE REDUNDANT BASIC EVENTS
C
                        DO 290 N1=1,5
                        IF(IHELP(N1).EQ.0) GO TO 310
                        DO 300 N2=6,15
                        IF(IHELP(N1).NE.IHELP(N2)) GO TO 300
                        IHELP(N2)=0
 300                    CONTINUE
 290                    CONTINUE
 310                    CONTINUE
                        DO 320 N3=6,10
                        DO 330 N4=11,15
                        IF(IHELP(N3).NE.IHELP(N4)) GO TO 330
                        IHELP(N4)=0
 330                    CONTINUE
 320                    CONTINUE
C
C       COMPUTE THE UNAVAILABILITY OF THE INTERSECTION OF MCS'S
C       M1,M2, AND M3 -- UNAINT
C
```

```
      UNAINT=1.
              DO 350 N5=1,15
              IF(IHELP(N5).EQ.0) GO TO 350
              UNAINT=UNAINT*BASE(IHELP(N5))
 350          CONTINUE
      IF((TPRIME).NE.8038.)   GO TO 77772
77772 CONTINUE
C
C     COMPUTE THE UNAVAILABILITY OF THE INTERSECTION OF THE MCS'S
C     TAKEN THREE AT A TIME.
C
      SUM3=SUM3+UNAINT
 270  CONTINUE
 260  CONTINUE
 250  CONTINUE
 251  CONTINUE
C
C     SECOND UPPER BOUND  ( THIRD SUMMATION )
C
      SUM=SUM1-SUM2+SUM3
      RETURN
      END
```

```
      SUBROUTINE ROFIE(ROF)
C
C
C              THIS SUBROUTINE COMPUTES THE RATE OF FAILURE   OF   THE
C
C      INITIATING EVENT AS A FUNCTION OF TIME.  IT PERFORMS A   LINEAR
C
C      INTERPOLATION BETWEEN THE DATA POINTS PROVIDED.  THE   NOTATION
C
C      USED IN THIS SUBROUTINE IS AS FOLLOWS:
C
C
C              ROFVCT(I,J)      THIS VECTOR STORES THE MAGNITUDE OF
C                               THE RATE OF       FAILURE   OF  THE
C                               INITIATING EVENT IN (1/HR.) AS    A
C
C
C                       I  -    DATA POINT IDENTIFICATION.
C
C                       J  -    =1 TIME, HRS.
C                               =2 MAGNITUDE OF THE ROF  (T).
C                                                    IE
C
C
C              DUMMY1,           THESE ARE AUXILIARY VARIABLES USED TO
C              DUMMY2,AND        PERFORM INTERMEDIATE CALCULATIONS.
C              DYMM3
C
C
      COMMON   INFORM( 3, 20,  5), JNFORM(3, 20, 20), RBE( 300,12), NDP
      COMMON      IBE( 300,3), MCSTYP(3, 20), NUMMCS(3), NUMMPS ( 3)
      COMMON         NUMRUN(3), IHELP(1000), TRUN(3), BASE( 300 )
      COMMON            MPSVCT(10, 20), ROFVCT(100,2),  TFINAL
      COMMON               TPRIME, DPRIME, FIRST, FINAL, NS
      COMMON               DELTA1, DELTA2, BOUND ,   IDELTA
      COMMON               DELTAP, TDIFF, DELTA, NIM, IND01
      COMMON               NBE, IND02, IND03, IND04 , IND05
      COMMON               IND10, IND09, IND08, IND07,IND06
      COMMON               JDELTA(3),              DLTPR (3)
C
C
C*************************************************************************
C                                                                        *
C      SELECT THE TIME INTERVAL WHICH CONTAINS T'                        *
C                                                                        *
C*************************************************************************
      DO 10000 I=1,100
              IF(ROFVCT(I,1).EQ.TPRIME)              GO TO 20000
              IF(ROFVCT(I,1).GT.TPRIME)              GO TO 30000
10000 CONTINUE
```

```
C*************************************************************************
C                                                                        *
C       THE VALUE OF 'ROF' IS AVAILABLE AT T'.                           *
C                                                                        *
C       NO INTERPOLATION IS REQUIRED.                                    *
C                                                                        *
C*************************************************************************
20000   CONTINUE
        ROF=ROFVCT(I,2)
        RETURN
C*************************************************************************
C                                                                        *
C       THE VALUE OF 'ROF' IS NOT AVAILABLE AT T'.  A LINEAR             *
C                                                                        *
C       INTERPOLATION WILL BE USED TO ESTIMATE 'ROF(TPRIME)'.            *
C                                                                        *
C*************************************************************************
30000   CONTINUE
        DUMMY1= ROFVCT (I,2) - ROFVCT(I-1,2)
        DUMMY2= ROFVCT (I,1) - TPRIME
        DUMMY3= ROFVCT (I,1) - ROFVCT(I-1,1)
        ROF   = ROFVCT (I,2) - DUMMY1 * DUMMY2 / DUMMY3
        RETURN
        END
```

```
C*************************************************************************
C                                                                        *
C      F U N C T I O N          S U B R O U T I N E S                    *
C                                                                        *
C*************************************************************************
C
C
      FUNCTION FNRA(X1,X2,X3)
C
C     THIS FUNCTION COMPUTES THE UNAVAILABILITY UPPER BOUND
C     APPLICABLE TO NONREPAIRABLE BASIC EVENTS.
C
C     THE NOTATION USED HERE IS THE FOLLOWING:
C
C                                  -1
C             X1 -- HAZARD RATE, HR   .
C
C             X2 -- TIME, HR.
C
C             X3 -- UNAVAILABILITY PER DEMAND, RBE(I,10).
C
      FNRA=X1*X2
      FNRA=FNRA+(1.-FNRA)*X3
      RETURN
      END
C
C
C
      FUNCTION FNRE(X1,X2,X3)
C
C     THIS FUNCTION COMPUTES THE EXACT VALUE OF THE BASIC EVENT
C     UNAVAILABILITY FOR NONREPAIRABLE BASIC EVENTS.
C
C     THE NOTATION USED HERE IS THE FOLLOWING:
C
C                                  -1
C             X1  -- HAZARD RATE, HR   .
C
C             X2  -- TIME INTERVAL, (T"-T'), HR.
C
C             X3  -- UNAVAILABILITY AT T'.
C
      DUMMY1=1.-X3
      DUMMY2=EXP(-X1*X2)
      FNRE=1.-DUMMY1*DUMMY2
      RETURN
      END
C
C
C
      FUNCTION FCMA(X1,X2,X3)
```

```
C
C       THIS FUNCTION COMPUTES THE UNAVAILABILITY UPPER BOUND
C       APPLICABLE TO CONSTANTLY MONITORED BASIC EVENTS.
C
C       THE NOTATION USED HERE IS THE FOLLOWING:
C
C                                      -1
C               X1  -- HAZARD RATE, HR   .
C
C               X2  -- THE AVERAGE TIME TO REPAIR, HR.
C
C               X3  -- UNAVAILABILITY PER DEMAND, RBE(I,10).
C
        FCMA=X1*X2
        FCMA=FCMA+(1.-FCMA)*X3
        RETURN
        END
C
C
C
        FUNCTION FCME(X1,X2,X3,X4)
C
C       THIS FUNCTION COMPUTES THE UNAVAILABILITY OF CONSTANTLY
C       MONITORED BASIC EVENTS.
C
C       THE NOTATION USED HERE IS THE FOLLOWING:
C
C                                      -1
C               X1  -- HAZARD RATE, HR   .
C
C               X2  -- THE AVERAGE TIME TO REPAIR, HR.
C
C               X3  -- TIME INTERVAL, (T"-T'), HR.
C
C               X4  -- UNAVAILABILITY AT T'.

        DUMMY1=(X1+1./X2)
        DUMMY2=X1/DUMMY1
        DUMMY3=EXP(-DUMMY1*X3)
        FCME=DUMMY2+(X4-DUMMY2)*DUMMY3
        RETURN
        END
C
C
C
        FUNCTION FPTA(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11)
C
C       THIS FUNCTION COMPUTES AN APPROXIMATION TO THE UNAVAILABILITY
C       OF PERIODICALLY TESTED BASIC EVENTS.
```

```
C
C       THE NOTATION USED HERE IS THE FOLLOWING:
C
C                                                    -1
C               X1   -- HAZARD RATE, RBE(I,1), HR    .
C
C               X2   -- PERIODIC TEST INTERVAL, RBE(I,2), HR.
C
C               X3   -- FIRST TEST INTERVAL, RBE(I,3), HR.
C
C               X4   -- TEST PERIOD, RBE(I,4), HR.
C
C               X5   -- REPAIR PERIOD, RBE(I,5), HR.
C
C               X6   -- OVERRIDE UNAVAILABILITY, RBE(I,6).
C
C               X7   -- PROBABILITY OF A TEST CAUSED FAILURE, RBE(I,7).
C
C               X8   -- DETECTION INEFFICIENCY, RBE(I,8).
C                                                         -1
C               X9   -- UNDETECTED HAZARD RATE, RBE(I,9), HR   .
C
C               X10  -- UNAVAILABILITY PER DEMAND, RBE(I,10).
C
C               X11  -- TIME, TP, HR.
C
      HAZ1=X1*(1.-X8)
      HAZ2=X1*X8+X9
      IF(X11.GT.X3) GO TO 15000
C
C     FIRST TEST INTERVAL
C
      FPTA=(X1+X9)*X11
      FPTA=FPTA+(1.-FPTA)*X10
      RETURN
C
C     REMAINING TEST INTERVALS
C
15000 CONTINUE
      INDIC1=INT((X11-X3)/X2)
      EFFTIM=X11-INDIC1*X2-X3
      IF(EFFTIM.LT.X4) GO TO 15010
      IF(EFFTIM.LT.(X5+X4)) GO TO 15020
C
C     BASIC EVENT ON STANDBY
C
      FPTA=HAZ2*X11+(1.-HAZ2*X11)*HAZ1*(EFFTIM-X4)
      FPTA=FPTA+(1.-FPTA)*X10
      RETURN
C
C     BASIC EVENT BEING TESTED
C
```

```
15010  CONTINUE
       IF(INDIC1.EQ.0) GO TO 15030
       Q=HAZ1*(X2-X4)
       GO TO 15040
15030  CONTINUE
       Q=HAZ1*X3
15040  CONTINUE
       FPTA=X7+(1.-X7)*X6+(1.-X7)*(1.-X6)*Q
       FPTA=FPTA+(1.-FPTA)*HAZ2*X11
       FPTA=FPTA+(1.-FPTA)*X10
       RETURN
C
C      BASIC EVENT UNDERGOING REPAIR
C
15020  CONTINUE
       IF(INDIC1.EQ.0) GO TO 15050
       Q=HAZ1*(X2-X4)
       GO TO 15060
15050  CONTINUE
       Q=HAZ1*X3
15060  CONTINUE
       FPTA=X7+(1.-X7)*Q+(1.-X7)*(1.-Q)*.5*HAZ1*X5
       FPTA=FPTA+(1.-FPTA)*HAZ2*X11
       FPTA=FPTA+(1.-FPTA)*X10
       RETURN
       END
```

VITA

Henrique Martini Paula was born in the city of Campinas, Brazil on June 17, 1956. He is the son of the late Camilo Marques Paula and Esmeralda Martini Paula. He attended elementary and high schools in the city of Indaiatuba, Brazil.

Henrique had his college education at the "Instituto Nacional de Telecomunicações (INATEL)," from which he graduated with a major in Electrical Engineering in December 1977. He entered the graduate program in Nuclear Engineering at the "Universidade de São Paulo" in March 1978 and received a Master of Science degree in November 1979. During the year of 1979, Henrique has also worked at the "Instituto de Pesquisas Energéticas e Nucleares" in the city of São Paulo, Brazil.

Henrique married Célia Maria on December 1, 1979 and moved to Knoxville, Tennessee. He entered the graduate program in Nuclear Engineering at The University of Tennessee, Knoxville on January 4, 1980. He has recently joined JBF Associates, Inc. of Knoxville, Tennessee, as a research engineer. After receiving the Doctor of Philosophy degree in 1984, Henrique will continue his work in the field of Nuclear Systems Reliability and Risk.