NASA Contractor Report 165863

CARE III PHASE II REPORT

MAINTENANCE MANUAL

L. A. Bryant and J. J. Stiffler

RAYTHEON COMPANY
Sudbury, Massachusetts 01776

Contract NAS1-15072
September 1982

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF TABLES

1.0 Introduction

CARE III (Computer-Aided Reliability Estimation, version three) is a computer program designed to help estimate the reliability of complex, redundant systems. Although the program can model a wide variety of redundant structures, it was developed specifically for fault-tolerant avionics systems - systems distinguished by the need for extremely reliable performance since a system failure could well result in the loss of human life.

It is usually relatively easy to design enough redundancy into a system to reduce to acceptably small levels the probability that it fails due to inadequate resources. The dominant cause of failure in ultra-reliable systems thus tends to be due not to the exhaustion of resources but rather to the failure to detect and isolate a malfunctioning element before it has caused the system to take an erroneous action. Such failures are called coverage failures. CARE III differs from its predecessors in, among other things, the attention given to coverage failure mechanisms.

The first CARE program, developed at the Jet Propulsion Laboratory in 1971, provided an aid for estimating the reliability of systems consisting of a combination of any of several standard configurations (e.g. standby-replacement configurations, triple-modular redundant configurations, etc.). CARE II was subsequently developed by Raytheon, under contract to the NASA Langley Research Center, in 1974. It substantially generalized the class of redundant configurations that could be accommodated, and included a coverage model to determine the various coverage probabilities as a function of the applicable fault recovery mechanisms (detection delay, diagnostic scheduling interval, isolation and recovery delay, etc.).

CARE III further generalizes the class of system structures that can be modeled and greatly expands the coverage model to take into account such effects as intermittent and transient faults, latent faults, error propagation, etc. In order to accomplish this, it was necessary to depart substantially from the approaches taken in previous reliability modeling efforts. The mathematical reliability and coverage models implemented in CARE III are described in the CARE III PHASE II Report, Mathematical Description.* The following section, taken from that report, is included here in order to put into context the ensuing program maintenance discussion.

*J. J. Stiffler and L. A. Bryant, CARE III PHASE II Report, Mathematical Description, NASA CR-3566, 1982.

2.0 Mathematical Model Description

The following paragraphs describe in detail the mathematical model as it is implemented in CARE III. The system to be modeled is assumed to consist of some number (up to 70) stages with each stage composed of one or more identical interchangeable elements or modules. The modules in each stage are subject to up to five user-defined categories of faults. A fault is characterized in terms of its rate of occurrence and in terms of its coverage model parameters. Fault occurrence rates are constrained to be of the form $\omega\lambda t^{\omega-1}$ (i.e., fault distributions are constrained to be Weibull) with ω and λ user defined. The user can also specify up to five sets of coverage model parameters (α, β, $\rho(t)$, $\varepsilon(t)$, $\delta(t)$, C, P_A, P_B); each such set defines a fault type. (Thus, for example, it is possible to define a permanent fault type, $\alpha = 0$; a transient type, $\alpha \neq 0$, $\beta = 0$; and an intermittent $\alpha \neq 0$, $\beta \neq 0$; each having its own characteristics with regard to detectability, error-propagation, etc.) Fault category x_i then refers to a fault that can affect any module in stage x; it is characterized by the parameters λ_{x_i}, ω_{x_i}, and j with j a fault-type designator.

In addition, the user must specify the number of modules n_x initially available at each stage, the minimum number m_x needed for that stage to function properly, the various combinations of stage failures that constitute a system failure, and the probabilities $b_{xy}(\nu_x, \nu_y)$ that a specific module in stage x forms a critical pair with a specific module in stage y given that ν_x stage-x modules and ν_y stage-y modules are known to have failed and are therefore no longer being used.*

---

*These last two tasks are both accomplished with relative ease through a CARE III user interface incorporating a program called FTREE developed by Boeing Aircraft Co. and described in the CARE III User's Manual.

On the basis of this user-supplied information, CARE III then determines the system unreliability using the equation

$$\bar{R}(t) = 1 - R(t) = \sum_{\underline{\ell}\varepsilon L} Q_{\underline{\ell}}(t) + \sum_{\underline{\ell}\varepsilon \bar{L}} P_{\underline{\ell}}{}^*(t)$$

with L the set of module failure combinations that would leave the system operational in the absence of a coverage failure, $\bar{L}$ the complementary set, $P_{\underline{\ell}}{}^*(t)$ the probability that the system would be in state $\underline{\ell}$ at time t in the absence of a coverage failure, and

$$Q_{\underline{\ell}}(t) = \int_0^t \left\{ \sum_{Y_j} \left[C_{Y_j}(\tau|\underline{\ell}-\varepsilon_y) P_{\underline{\ell}-\varepsilon_y}{}^*(\tau)(n_y-\ell_y+1)\lambda_{Y_j}(\tau) \right] + A'(\tau|\underline{\ell})P_{\underline{\ell}}{}^*(\tau) + a'(\tau|\underline{\ell})P_{\ell}{}^*(\tau) \right\} d\tau$$

with

$$\mu_{\underline{\ell}}(t) = A'(t|\underline{\ell}) + a'(t|\underline{\ell})$$

$$\lambda^{(2)}_{\underline{j\ell}}(t) = \sum_{Y_j\varepsilon Y} C_{Y_j}(\tau|\underline{\ell}-\varepsilon_y)(n_y-\ell_y+1)\lambda_{Y_j}(\tau)$$

and $\underline{j} = \underline{\ell}-\varepsilon_y$ with ε_y the unit vector denoting a stage-y module.

$\mu_{\underline{\ell}}(t)$ and $\lambda^{(2)}_{\underline{j\ell}}(t)$ are defined in terms of functions of the form

$$\int_0^t p_2(\tau)p_1(t-\tau)d\tau$$

with $p_1(t)$ a measure of the rate at which a certain class of faults occurs and $p_2(\tau)$ a function of the interval τ between that occurrence and the entry of the fault into a particular coverage-model state. Since, typically, faults occur at rates no greater than one fault every several thousand hours, and since coverage-state time constants are usually of the order of fractions of seconds and rarely exceed a few minutes in duration, $p_1(t)$ is a much more slowly varying function of time than is $p_2(\tau)$. Thus, to a very good approximation

$$p_1(t-\tau) \approx a(t) + \tau b(t) + \tau^2 c(t)$$

over the range of τ for which $p_1(\tau)$ is not negligibly small, with $a(t)$, $b(t)$, and $c(t)$ suitably defined. This approximation is used in CARE III with $a(t)$, $b(t)$, and $c(t)$ defined to make the approximation exact at the two end points and at the midpoint of the range of interest of τ. The major advantage of introducing this approximation is that, with it,

$$\int_0^t p_2(\tau)p_1(t-\tau)d\tau \approx a(t)m_2^{\,0}(t) + b(t)m_2^{\,1}(t) + c(t)m_2^{\,2}(t)$$

with

$$m_2^{\,i}(t) = \int_0^t \tau^i p_2(\tau)d\tau$$

Thus, the convolution can be separated into two parts, one part depending only on the reliability-model function $p_1(t)$ and the other involving only the first three moments of the coverage-model function $p_2(\tau)$. Moreover, these moments need be evaluated only at those points of time t relevant to the reliability model. This significantly simplifies the interface between the coverage and reliability models.

With these preliminaries, the reliability model functions used in CARE III are itemized in Table 1 and the coverage model functions in Table 2.

LIST OF SYMBOLS

Symbol	Module(s)	Definition
$A(t\|\underline{\ell})$	CARE3	See Table 1.
$A'(t\|\underline{\ell})$	CARE3	See Table 1.
$a_x(t)$	CARE3	See Table 1.
$a_{x_i}(t)$	CARE3	See Table 1.
$a'(t\|\underline{\ell})$	CARE3	See Table 1.
$B_{x_i,y_j}(t\|\underline{\ell})$	CARE3	See Table 1.
$b_{x,y}(\nu_x,\nu_y)$	CAREIN CARE3	Probability that a stage-x fault and a stage-y fault form a critical pair given ν_x known stage-x faults and ν_y known stage-y faults.
C	CAREIN COVRGE	Probability that a propagated error is detected before it causes the system to malfunction.
$C_{y_j}(t\|\underline{\ell})$	CARE3	See Table 1.
$D_{x_i,y}(t\|\underline{\ell})$	CARE3	See Table 1.
$d_{x_i}(t)$	COVRGE	Probability that a category-x_i error has not been detected t time units after its generation.
$e_{x_i}(t)$	COVRGE	Probability that a category-x_i error has not yet propagated t time units after its generation.
$H_X(t\|x_i)$ $X = L,\overline{B},B$	CARE3	See Table 1.
$h_X(t\|x_i)$ $X = DPT,F$	CARE3	See Table 1.
$h_{DF}(t\|x_i,y_j)$	CARE3	See Table 1.

Symbol	Module(s)	Definition
$\underline{\ell}$	CARE3	A vector, the components of which indicate the number of faulty elements in each stage ($\underline{\ell} = \ldots \ell_x, \ell_y, \ldots$).
$\underline{\ell} - \varepsilon_y$	CARE3	$(\ldots\ldots \ell_x, \ell_y - 1, \ell_z, \ldots\ldots)$
ℓ_x	CARE3	Number of stage-x modules that have experienced some fault ($\ell_x = \sum_i \ell_{x_i}$)
ℓ_{x_i}	CARE3	Number of stage-x modules that have experienced category-x_i faults.
m_x	CAREIN CARE3	Minimum number of units needed for stage x to be operational.
n_x	CAREIN CARE3	Number of functional stage-x units available at time t = 0.
$P_{\underline{\ell}}$	CARE3	State of operational system after exactly $\underline{\ell}$ failures.
$P(t)$	CARE3	Element survival probability.
$P'(t)$	CARE3	Rate of change of P(t).
$P_{\underline{\ell}}(t)$	CARE3	Probability that a system has sustained exactly $\underline{\ell}$ failures and is still operational at time t.
$P_{\underline{\ell}}^*(t)$	CARE3	Probability that a system has sustained exactly $\underline{\ell}$ failures by time t.
$P(\mu_x, t \mid \ell_x)$	CARE3	See Table 1.
P_A	CAREIN COVRGE	Probability that a fault detected in the active state is diagnosed as permanent.
P_B	CAREIN COVRGE	Probability that a fault detected in the benign state is diagnosed as permanent.
Q_{ℓ}	CARE3	State of system that has failed after exactly $\underline{\ell}$ element failures.

Symbol	Module(s)	Definition
$Q_{\ell}(t)$	CARE3	Probability that a system has sustained exactly ℓ failures and then malfunctioned by time t.
$r_{x_i}(t)$	COVRGE	Probability that a category-x_i fault has not resulted in an error t time units after it has become active.
$R(t)$	CARE3	Reliability; probability that the system is operational at time t.
$r_{ij}(t)$	———	Transition rate between states S_i and S_j.
$R_x(t)$	CARE3	See Table 1.
$R_{x_i}(t)$	CARE3	See Table 1.
S_i	———	State i in a Markov chain.
$S_i(t)$	———	State i occupancy probability at time t.
$S_i'(t)$	———	Rate of change of $S_i(t)$.
x	CAREIN CARE3	Stage index.
α_{x_i}	CAREIN COVRGE	Rate of transition from active to benign fault states.
β_{x_i}	CAREIN COVRGE	Rate of transition from benign to active fault states.
$\delta_{x_i}(t)/d_{x_i}(t)$	COVRGE	Rate of detection of type-x_i faults t time units after they become active.
$\varepsilon_{x_i}(t)/e_{x_i}(t)$	COVRGE	Rate at which an error caused by a type-x_i fault is propagated following its generation at time $t = 0$.

Symbol	Module(s)	Definition
$\Lambda_{\ell}(t,\tau)$	CARE3	$\int_{\tau}^{t} \lambda_{\ell}(\eta)d\eta$
$\lambda_{j\ell}^{(1)}(t)$	———	Rate of occurrence of failures that take the system from state P_j to state P_{ℓ}.
$\lambda_{j\ell}^{(2)}(t)$	———	Rate of occurrence of failures that take the system from state P_j to Q_{ℓ}.
$\lambda_{j\ell}^{*}(t)$	———	$\lambda_{j\ell}^{(1)}(t) + \lambda_{j\ell}^{(2)}(t)$
$\lambda_{\ell}(t)$	———	Transition rate out of state P_{ℓ}.
$\lambda_{\ell}^{*}(t)$	———	$\sum_{j \neq \ell} \lambda_{\ell j}^{*}(t)$
$\lambda_{x_i}(t)$	CAREIN COVRGE CARE3	Rate of occurrence of category-x_i faults ($\lambda_{x_i}(t)=\omega_{x_i}\lambda_{x_i} t^{\omega_{x_i}-1}$).
$\underline{\mu}$	CARE3	A vector, the components of which indicate the current number of latent faults in each stage ($\underline{\mu} = \ldots \mu_x, \mu_y, \ldots,$).
$\rho_{x_i}(t)/r_{x_i}(t)$	COVRGE	Error generation rate of a type-x_i fault t time units after it becomes active.
$\mu_{\ell}(t)$	CARE3	Transition rate from system state P_{ℓ} to system state Q_{ℓ} ($\mu_{\ell}(t) = \mu_{\ell}^{(1)}(t) + \mu_{\ell}^{(2)}(t)$).
$\mu_{\ell}^{(1)}(t)$	———	Transition rate from state P_{ℓ} to state Q_{ℓ} due to single-fault coverage failures.
$\mu_{\ell}^{(2)}(t)$	———	Transition rate from state P_{ℓ} to state Q_{ℓ} due to double-fault coverage failures.

Table 1

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION		DEFINITION
$R_{x_i}(t)$	$e^{-\Lambda_{x_i}(t)}$	PERMANENT	PROBABILITY THAT A GIVEN STAGE x MODULE HAS NOT EXPERIENCED A CATEGORY-x_i FAULT BY TIME t
	$e^{-H_{DPT}(\tau \mid x_i)d\tau}$	TRANSIENT	
$R_x(t)$	$\prod_i R_{x_i}(t)$		RELIABILITY OF A STAGE x MODULE
$a_{x_i}(t)$	$\frac{H_L(t \mid x_i)}{1-R_x(t)}$	PERMANENT	PROBABILITY THAT A GIVEN STAGE x MODULE HAS A CATEGORY-x_i LATENT PERMANENT (TRANSIENT) FAULT AT TIME t GIVEN THAT IT HAS (NOT) EXPERIENCED A PERMANENT OR LEAKY TRANSIENT FAULT BY TIME t
	$H_L(t \mid x_i)$	TRANSIENT	
$a_x(t)$	$\sum_i a_{x_i}(t)$ (PERMANENT)		PROBABILITY THAT A GIVEN STAGE x MODULE HAS A LATENT (PERMANENT) FAULT AT TIME t GIVEN THAT IT HAS EXPERIENCED SOME PERMANENT FAULT BY TIME t

Table 1 (Continued)

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$P(\mu_x,t\|\ell_x)$	$\binom{\ell_x}{\mu_x}(1-a_x(t))^{\ell_x-\mu_x}\, a_x^{\mu_x}(t)$	PROBABILITY THAT A SUBSYSTEM CONTAINS μ_x STAGE x LATENT PERMANENT FAULTS GIVEN THAT IT HAS ℓ_x STAGE x PERMANENT FAULTS
$B_{x_i,y_j}(t\|\underline{\ell})$	$\sum_{\mu_x,\mu_y} b_{x,y}(\ell_x-\mu_x,\ell_y-\mu_y)P(\mu_x,t\|\ell_x)$ $P(\mu_y,t\|\ell_y)C(x_i,y_j)a_{x_i}(t)a_{y_j}(t)$	EXPECTED NUMBER OF x_i,y_j-CRITICAL FAULTS AT TIME t GIVEN $\underline{\ell}$ PERMANENT FAULTS
$C(x_i,y_j)$	$\frac{\mu_x\mu_y}{a_x(t)a_y(t)}$ x_i,y_j = PERMANENT, $x \neq y$ $\frac{\mu_x(\mu_x-1)}{a_x^2(t)}$ x_i,y_j = PERMANENT, $x = y$ $\frac{\mu_x(n_y-\ell_y)}{a_x(t)}$ x_i = PERMANENT, y_j = TRANSIENT	

Table 1 (Continued)

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$C(x_i, y_j)$ (CONT.)	$\frac{(n_x-\ell_x)\mu_y}{a_y(t)}$ $\quad x_i$ = TRANSIENT, y_j = PERMANENT	
	$(n_x-\ell_x)(\mu_y-\ell_y)$ $\quad x_i, y_j$ = TRANSIENT, $x \neq y$	
	$(n_x-\ell_x)(n_x-\ell_x-1)$ $\quad x_i, y_j$ = TRANSIENT, $x = y$	
$D_{x_i,y}(t\|\underline{\ell})$	$\sum_{\mu_x,\mu_y} b_{x,y}(\ell_x-\mu_x,\ell_y-\mu_y)P(\mu_x,t\|\ell_x)P(\mu_y,t\|\ell_y)\begin{cases}\mu_x \frac{a_{x_i}(t)}{a_x(t)} & x_i = \text{PERMANENT} \\ (n_x-\ell_x)a_{x_i}(t) & x_i = \text{TRANSIENT}\end{cases}$	EXPECTED NUMBER OF $x_i y$-CRITICAL FAULTS, GIVEN $\underline{\ell}$ PERMANENT FAULTS, THAT WOULD BE CREATED AS THE RESULT OF A STAGE y FAULT AT TIME t
$C_{y_j}(t\|\underline{\ell})$	$\sum_{x_i} \frac{H_{\bar{B}}(t\|x_i)}{H_L(t\|x_i)} D_{x_i,y_j}(t\|\underline{\ell})$	PROBABILITY THAT A CATEGORY y_j FAULT WOULD PRODUCE A SYSTEM FAILURE AT TIME t GIVEN $\underline{\ell}$ FAULTS AT TIME t^-

Table 1 (Continued)

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$A'(t\|\underline{\ell})$	$\sum_{x_i,y_j} \frac{h_{DF}(t\|x_i,y_j)}{H_L(t\|x_i)H_L(t\|y_j)} B_{x_i,y_j}(t\|\underline{\ell})$	RATE WHICH SYSTEMS HAVING $\underline{\ell}$ FAULTS FAIL AT TIME t DUE TO CRITICAL FAULT CONDITIONS
$a'(t\|\underline{\ell})$	$\sum_{x_i} \frac{\ell_x h_F(t\|x_i)}{1-R_x(t)}$ PERMANENT $+\sum_{x_i}(n_x-\ell_x)h_F(t\|x_i)$ TRANSIENT	RATE AT WHICH SYSTEMS HAVING $\underline{\ell}$ FAULTS FAIL AT TIME t DUE TO ERROR PROPAGATION
$H_L(t\|x_i)$	$a_L(t\|x_i)M_L^0(t\|x_i)+b_L(t\|x_i)M_L^1(t\|x_i)$ $+c_L(t\|x_i)M_L^2(t\|x_i)$	PROBABILITY OF A LATENT CATEGORY x_i FAULT AT TIME t
$H_{\overline{B}}(t\|x_i)$	$a_{\overline{B}}(t\|x_i)M_{\overline{B}}^0(t\|x_i)+b_{\overline{B}}(t\|x_i)M_{\overline{B}}^1(t\|x_i)$ $+c_{\overline{B}}(t\|x_i)M_{\overline{B}}^2(t\|x_i)$	PROBABILITY OF A NON-BENIGN LATENT FAULT AT TIME t

Table 1 (Continued)

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$H_B(t\|x_i)$	$a_B(t\|x_i)M_B^0(t\|x_i)+b_B(t\|x_i)M_B^1(t\|x_i)$ $+ c_B(t\|x_i)M_B^2(t\|x_i)$	PROBABILITY OF A BENIGN LATENT FAULT AT TIME t
$H_{DPT}(t\|x_i)$	$a_{DP}(t\|x_i)m_{DP}^0(t\|x_i)+b_{DP}(t\|x_i)m_{DP}^1(t\|x_i)$ $+ c_{DP}(t\|x_i)m_{DP}^2(t\|x_i)$	PROBABILITY THAT A CATEGORY x_i TRANSIENT FAULT IS DETECTED AS PERMANENT
$h_{DF}(t\|x_i,y_j)$	$a_{DF}(t\|x_i,y_j)m_{DF}^0(t\|x_i,y_j)$ $+ b_{DF}(t\|x_i,y_j)m_{DF}^1(t\|x_i,y_j)$ $+ c_{DF}(t\|x_i,y_j)m_{DF}^2(t\|x_i,y_j)$	RATE AT WHICH AN x_iy_j-CRITICAL FAULT CAUSES SYSTEM FAILURE
$h_F(t\|x_i)$	$a_F(t\|x_i)m_F^0(t\|x_i)+b_F(t\|x_i)m_F^1(t\|x_i)$ $+ c_F(t\|x_i)m_F^2(t\|x_i)$	RATE OF ERROR PROPAGATION FAILURE DUE TO A CATEGORY x_i FAULT

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$a_X(t\|x_i)$ $a_X(t\|x_i,y_j)$	$f(t)=\begin{cases}\lambda_{x_i}(t) & X = DP,L,B,\bar{B},F \\ & x_i = \text{TRANSIENT} \\ \lambda_{x_i}(t)\,R_X(t) & X = L,B,\bar{B},F \\ & x_i = \text{NON-TRANSIENT} \\ H_B(t\|x_i)\,\lambda_{y_j}(t) & X = DF \\ & y_j = \text{TRANSIENT} \\ H_B(t\|x_i)\,\lambda_{y_j}(t)\,r_Y(t) & X = DF \\ & y_j = \text{NON-TRANSIENT}\end{cases}$	WEIGHT FUNCTIONS USED IN CONVOLUTIONAL APPROXIMATION (CF. EQUATION 25)
$b_X(t\|x_i)$ $b_X(t\|x_i,y_j)$ all X	$\begin{cases}-\dfrac{f(t)-f(0)}{t} & t = \Delta t_r \\ -\dfrac{(3k^2-1)f(t)-2k^2\left[f\left(\frac{t+\Delta t_r}{2}\right)+f\left(\frac{t-\Delta t_r}{2}\right)\right]+(k^2+1)f(0)}{(k^2-1)t} & t = k\Delta t_r, < t_n,\ k \text{ odd} \\ -\dfrac{3f(t)-4f(t/2)+f(0)}{t} & t = k\Delta t_r, < t_n,\ k \text{ even} \\ -\dfrac{3f(t)-4f(t-t_n/2)+f(t-t_n)}{t_n} & t \geq t_n\end{cases}$	

Table 1 (Continued)

Reliability Model Functions

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$c_X(t\|x_i)$ $c_X(t\|x_i,y_j)$	$\begin{cases} 0 & t = \Delta t_r \\ \dfrac{f(t) - f\left(\frac{t+\Delta t_r}{2}\right) + f\left(\frac{t-\Delta t_r}{2}\right) + f(0)}{[t^2 - (\Delta t_r)^2]/2} & t = k\Delta t_r < t_n \text{, } k \text{ odd} \\ \dfrac{f(t) - 2f(t/2) + f(0)}{t^2/2} & t = k\Delta t_r < t_n \text{, } k \text{ even} \\ \dfrac{f(t) - 2f(t-t_n/2) + f(t-t_n)}{t_n^2/2} & t \geq t_n \end{cases}$	
t_n	min $t = n\ t_r$, n even, such that $p_X(t) \leq \theta$ or $P_X(t) \leq \theta$ with θ a user-defined threshold	

Table 2a

Single-Fault Model Equations

FUNCTION	MATHEMATICAL EXPRESSION*	DEFINITION
$\phi(t)$	$\alpha e^{-\beta t}\int_0^t e^{-(\alpha-\beta)\tau} r(\tau)d(\tau)d\tau$	β^{-1} TIMES THE PROBABILITY INTENSITY OF RE-ENTERING STATE A EXACTLY t TIME UNITS AFTER THE PREVIOUS ENTRY
$P_a(t)$	$e^{-\alpha t} r(t)d(t) + \beta\int_0^t \phi(t-\tau)P_a(\tau)d\tau$	PROBABILITY OF BEING IN STATE A AT TIME t WHEN $P_A = P_B = 1$
$P_b(t)$	$\phi(t) + \beta\int_0^t \phi(t-\tau)P_b(\tau)d\tau$	PROBABILITY OF BEING IN STATE B AT TIME WHEN $P_A = P_B = 1$
$P_e(t)$	$\int_0^t e^{-\alpha\tau}\rho(\tau)d(\tau)e(t-\tau)d\tau + \beta\int_0^t \phi(t-\tau)P_e(\tau)d\tau$	PROBABILITY OF BEING IN STATE A_E OR B_E AT TIME t WHEN $P_A = P_B = 1$

* t HERE IS A MEASURE OF THE TIME SINCE THE ENTRY INTO STATE A.

Table 2a (Continued)

Single-Fault Model Equations

FUNCTION	MATHEMATICAL EXPRESSION*	DEFINITION
$p_e(t)$	$e^{-\alpha t}\rho(t)d(t) + \beta\int_0^t \phi(t-\tau)p_e(\tau)d\tau$	INTENSITY OF ENTRY INTO STATE A_E AT TIME t WHEN $P_A = P_B = 1$
$p_{\bar{e}}(t)$	$e^{-\alpha t}\delta(t)r(t) + \beta\int_0^t \phi(t-\tau)p_{\bar{e}}(\tau)d\tau$	INTENSITY OF ENTRY INTO STATE A_D FROM STATE A AT TIME t WHEN $P_A = P_B = 1$
$p_f(t)$	$(1-C)\int_0^t p_e(\tau)\varepsilon(t-\tau)d\tau$	INTENSITY OF ENTRY INTO STATE F AT TIME t WHEN $P_A = P_B = 1$
$\psi_A(t)$	$C\int_0^t p_e(\tau)\varepsilon(t-\tau)\left(\frac{\beta + \alpha e^{-(\alpha+\beta)(t-\tau)}}{\alpha+\beta}\right)d\tau + p_{\bar{e}}(t)$	INTENSITY OF ENTRY INTO STATE A_D AT TIME t FOR THE FIRST TIME

* t HERE IS A MEASURE OF THE TIME SINCE THE ENTRY INTO STATE A.

Table 2a (Continued)

Single-Fault Model Equations

FUNCTION	MATHEMATICAL EXPRESSION*	DEFINITION
$\Psi_B(t)$	$\frac{\alpha C}{\alpha+\beta}\int_0^t p_e(\tau)(1 - e^{-(\alpha+\beta)(t-\tau)})\varepsilon(t-\tau)d\tau$	INTENSITY OF ENTRY INTO STATE B_0 AT TIME t FOR THE FIRST TIME
$\chi_B(t)$	$\int_0^t \Psi_B(\tau)e^{-\beta(t-\tau)}d\tau$	PROBABILITY OF HAVING ENTERED STATE B_D FOR THE FIRST TIME AND THEN REMAINING IN THE BENIGN STATE UNTIL TIME t
$P_{dp}(t)$	$P_A\int_0^t \Psi_A(\tau)d\tau + P_B\int_0^t \Psi_B(\tau)d\tau$	PROBABILITY THAT A FAULT HAS BEEN DIAGNOSED AS PERMANENT BY TIME t
$F_X(t)$	$F_x(t) + \int_0^t [(1-P_A)\Psi_A(t-\tau) + (1-P_B)\beta\chi_B(t-\tau)]F_X(\tau)d\tau$	FUNCTION RELATING PROBABILITIES AND INTENSITIES DERIVED WHEN $P_A = P_B = 1$ TO THOSE SAME QUANTITIES WHEN P_A & P_B ARE ARBITRARY

* t HERE IS A MEASURE OF THE TIME SINCE THE ENTRY INTO STATE A.

Table 2a (Continued)

Single-Fault Model Equations

FUNCTION	MATHEMATICAL EXPRESSION*	DEFINITION
$P_B(t)$	$F_X(t)$ with $F_x(t) = P_b(t) + X_B(t)$	PROBABILITY OF BEING IN STATE B AT TIME t
$P_{\bar{B}}(t)$	$F_X(t)$ with $F_x(t) = P_a(t) + P_e(t)$	PROBABILITY OF BEING IN A NON-BENIGN STATE AT TIME t
$P_L(t)$	$F_X(t)$ with $F_x(t) = \begin{cases} P_b(t) + X_B(t) \\ + P_a(t) + P_e(t) \\ \text{PERMANENT FAULTS} \\ P_a(t) + P_e(t) \\ \text{TRANSIENT FAULTS} \end{cases}$	PROBABILITY OF A LATENT FAULT OR UNDETECTED ERROR AT TIME t
$P_{DP}(t)$	$F_X(t)$ with $F_x(t) = P_{dp}(t)$	PROBABILITY THAT A FAULT HAS BEEN DIAGNOSED AS PERMANENT BY TIME t

* t HERE IS A MEASURE OF THE TIME SINCE THE ENTRY INTO STATE A.

Table 2b

Double-Fault Model Equations

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$c_i(t)$ $i = 1,2$ $j = 3-i$	$\beta_i(t)d_j(t)r_j(t)a_j(t)$ + $(1-P_{A_j})b_i(t)\,\delta_j(t)r_j(t)a_j(t)$ + $b_i(t)d_j(t)\rho_j(t)a_j(t)$	TRANSITION RATE FROM STATE A_jB_i TO STATE F
$f_i(t)$ $i = 1,2$ $j = 3-i$	$\alpha_j(t)b_i(t)d_i(t)r_i(t)$	TRANSITION RATE FROM STATE A_jB_i TO STATE B_1B_2
$c_4(t)$	$\int_0^t [c_1(t-\tau)\beta_2(\tau)b_1(\tau)$ + $c_2(t-\tau)\,\beta_1(\tau)b_2(\tau)]d\tau$	INTENSITY OF ENTRY INTO STATE F t TIMEUNITS AFTER ENTRY INTO STATE B_1B_2
$c_3(t)$	$\int_0^t [f_1(t-\tau)\beta_2(\tau)b_1(\tau)$ + $f_2(t-\tau)\,\beta_1(\tau)b_2(\tau)]d\tau$	INTENSITY OF RE-ENTRY INTO STATE B_1B_2 t TIME UNITS AFTER A PREVIOUS ENTRY

Table 2b (Continued)

Double-Fault Model Equations

FUNCTION	MATHEMATICAL EXPRESSION	DEFINITION
$p_3(t)$	$f_1(t) + \int_0^t c_3(t-\tau)p_3(\tau)d\tau$	INTENSITY OF ENTRY INTO STATE B_1B_2 t TIME UNITS AFTER ENTRY INTO STATE A_2B_1
$p_{DF}(t)$	$c_1(t) + \int_0^t c_4(t-\tau)p_3(\tau)d\tau$	INTENSITY OF ENTRY INTO STATE F t TIME UNITS AFTER ENTRY INTO STATE A_2B_1

3.0 CARE III Program Modules

CARE III consists of three interdependent software modules written in FORTRAN EXTENDED Version 4:

1) CAREIN - input processor,
2) COVRGE - coverage model, and
3) CARE3 - reliability model (see Figures 3.0.1 and 3.0.2).

The following sections contain a description sheet for each of the main programs, subroutines, and functions which comprise each of the modules listed above. Each routine description sheet consists of the files, parameters, arrays, common areas, externals, and inline functions required by the routine. Where appropriate, the mathematical functions are included with the routine that solves them. A functional flowchart is included for each module, and a description of the files, enumerated in the program card of each main program, is also included.

The following are the execution field lengths for each program comprising CARE III.

MODULE	PROGRAM	FIELD LENGTH (Octal)
CAREIN	CAREIN	154000
COVRGE	COVRGE	163700
	CVGPLT	127600
CARE3	CARE3	157200
	RELPLT	076000

CARE III STRUCTURE

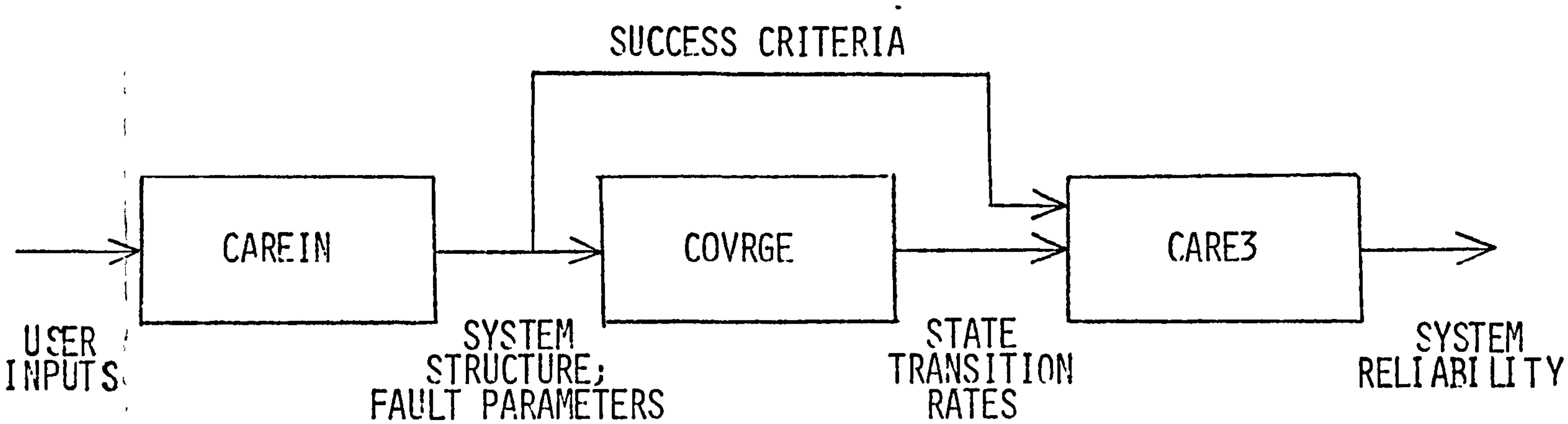


Figure 3.0.1

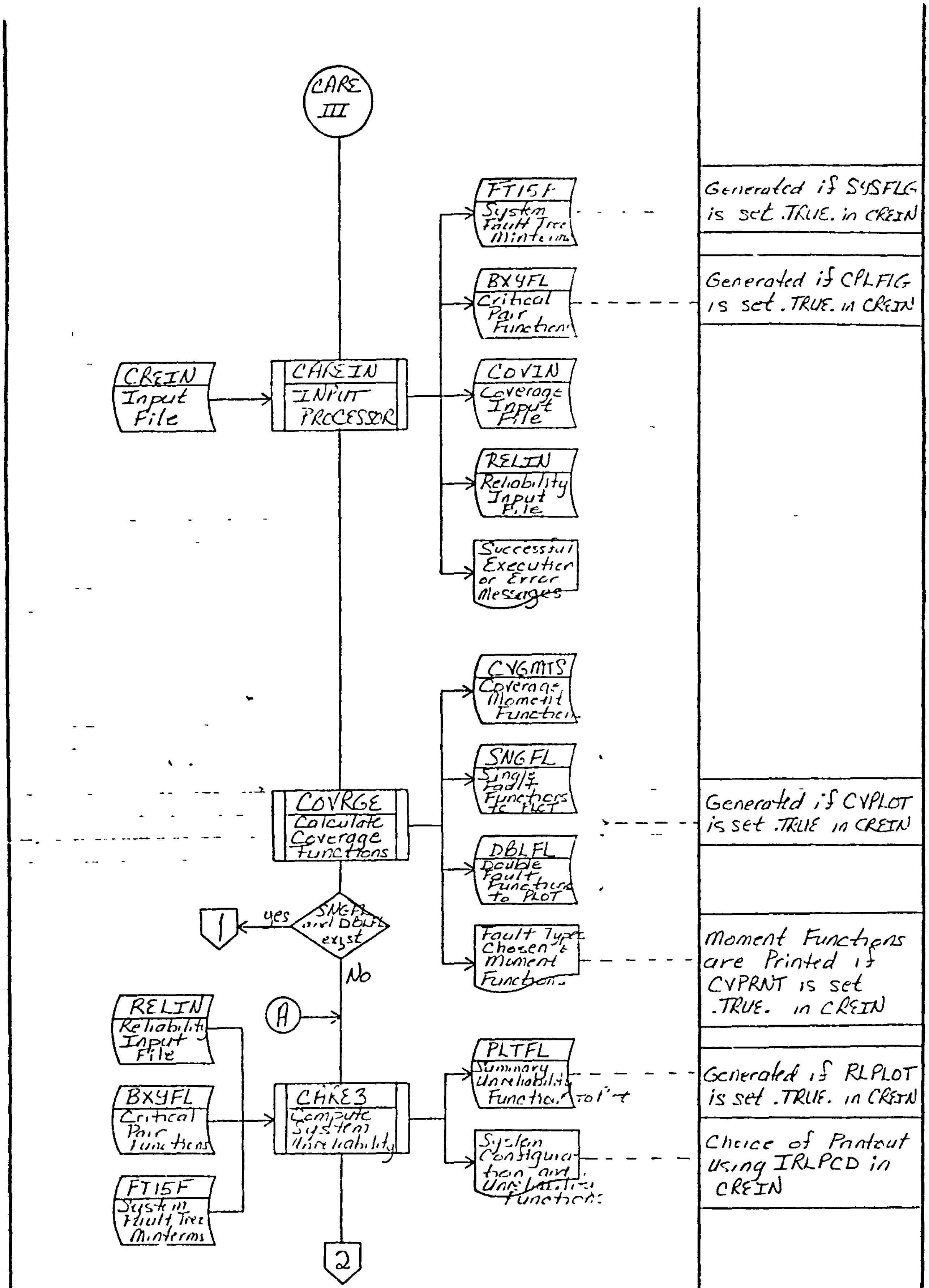
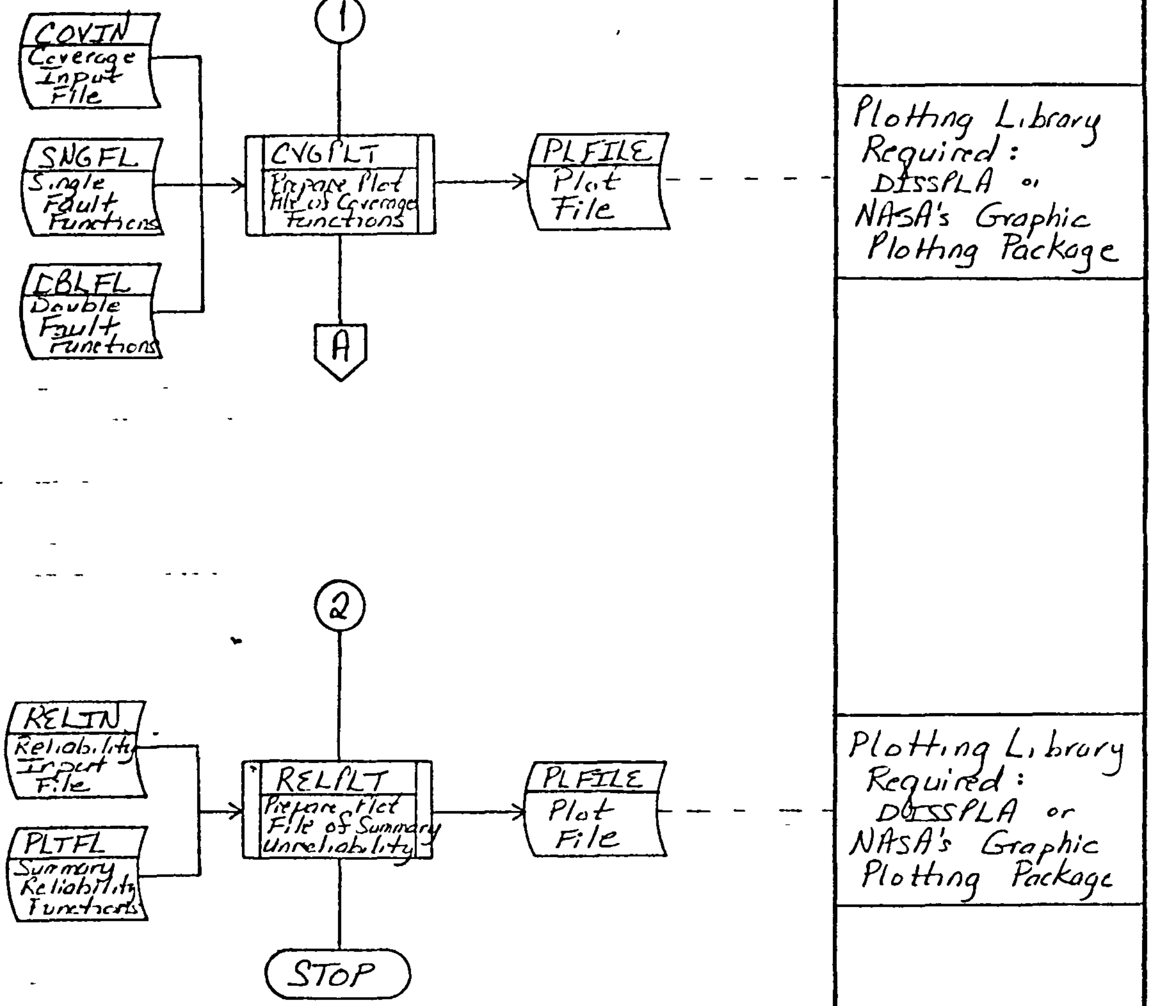


Figure 3.0.2

CARE III Functional Flowchart

RAYTHEON

RAYTHEON COMPANY
LEXINGTON MASS 02173



RAYTHEON

RAYTHEON COMPANY
LEXINGTON MASS 02173

CARE III Functional Flow

49956 Lynn A Bryant 4/20/81

SHEET 2 OF 2

3.1 CAREIN Module

The CAREIN software module consists of the main program CAREIN and several subroutines and functions. A major portion of this module consists of subroutine FTREE and its related routines developed by Boeing Aircraft Company and described in the CARE III PHASE II Report, User's Manual.*

The CAREIN module processes user input contained in file CREIN and then generates files for modules COVRGE and CARE3. (See the CARE III User's Manual for a description of the user input to CAREIN.)

---

*L. A. Bryant and J. J. Stiffler, CARE III PHASE II Report, User's Manual, NASA CR-165864, September 1982.

3.1.1 CAREIN Main Program

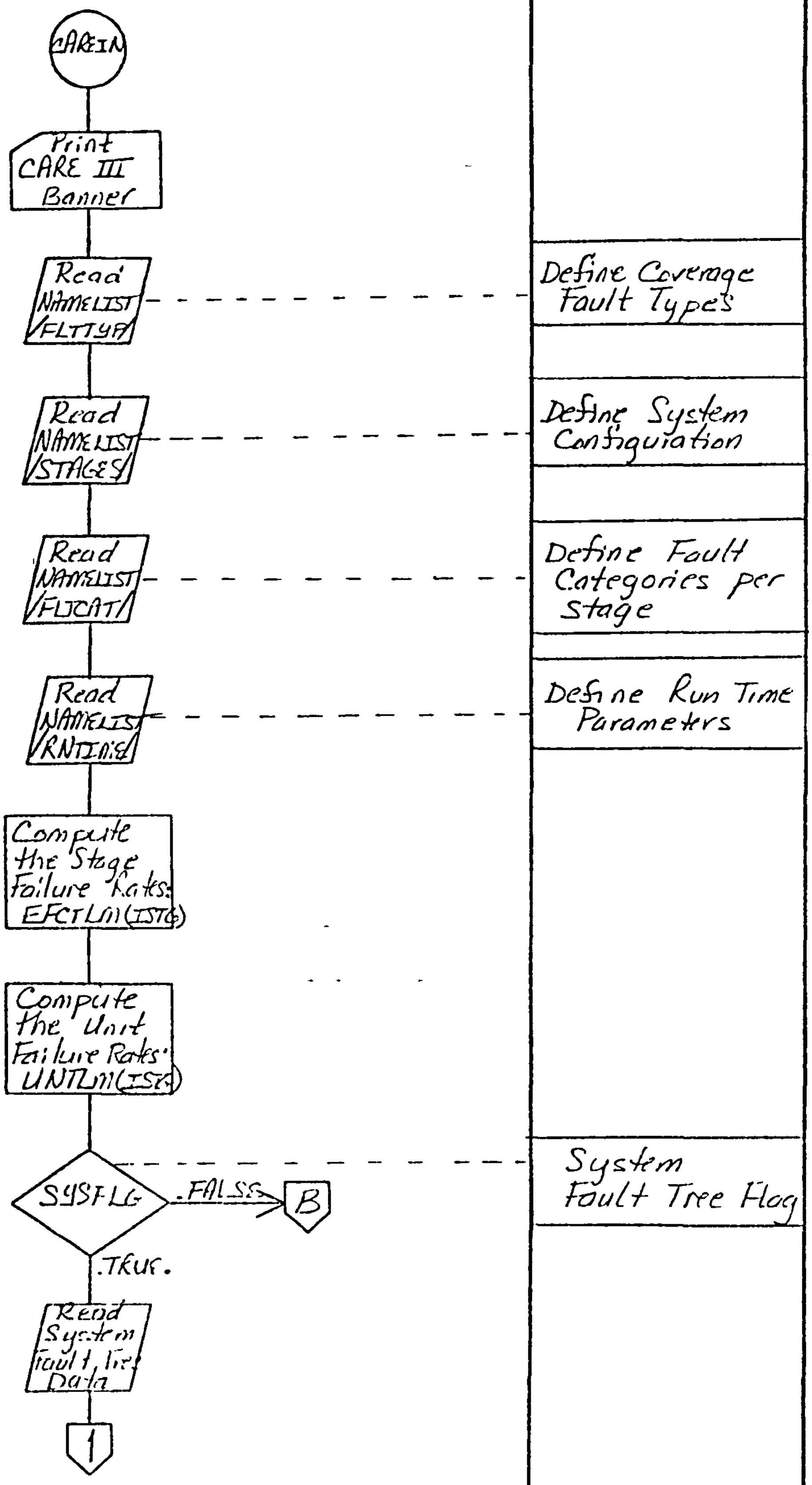
```
      PROGRAM CAREIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,CREIN,
     1 TAPE7=CREIN,CREOUT,TAPE8=CREOUT,TAPE10=82,FT11F,TAPE11=FT11F,
     2 BXYFL=0,TAPE14=BXYFL,FT15F,TAPE15=FT15F,FT16F,TAPE16=FT16F,
     3 FT25F,TAPE25=FT25F,FT26F,TAPE26=FT26F,COVIN=0,TAPE3=COVIN,
     4 RELIN=0,TAPE4=RELIN)
C
C***********************************************************************
C*                                                                     *
C*       TITLE:  COMPUTER-AIDED RELIABILITY ESTIMATION, VERSION THREE *
C*       ACRONYM: CARE III                                             *
C*       MODEL DESIGN: DR. J.J.STIFFLER                                *
C*       SOFTWARE DESIGN AND IMPLEMENTATION: LYNN A. BRYANT            *
C*       DATE: MARCH 23,1981                                           *
C*       DATE EDITED: 4-22-81                                          *
C*       EDITED BY: LYNN A. BRYANT                                     *
C*                                                                     *
C*       PURPOSE:  CARE III IS A GENERAL-PURPOSE RELIABILITY           *
C*                 ESTIMATION TOOL FOR FAULT-TOLERANT AVIONICS         *
C*                 SYSTEMS.                                            *
C*                                                                     *
C***********************************************************************
C
C  FT15F IS THE SYSTEM MINTERM FILE.
C  FT25F IS THE CRITICAL PAIR MINTERM FILE.
C
      COMMON// P(2000),T(50),SUM(2000),PR(70),LS,LTT,SAV,MSENS,ID,
     1 QP(70),PQ(70),BOG,BIG,DELTA,IC(2000),IN(2000),IP(10000),IR(2000),
     2 IDEP(700),IDUM(4300),CINDBG
      LOGICAL CINDBG,FATAL,EOFFLG
C* NOTE: 'IDUM' IS A PLACE-HOLDER FOR USE IN SUBROUTINE CTRLPRS WHEN
C* 'IC' IS EQUIVALENCED TO 'KNTMTS' DIMENSIONED TO (10,10,210).
C
```

```
EXTERNALS            TYPE    ARGS
       BUFBLK                 6

       CRTLPRS                4
       EOF         REAL       1
       FNCK        REAL       2
       FTREE                  1
       SPLIT                  6
       SUBRUN                 6

NAMELISTS
       FLTCAT
       FLTTYP
       RNTIME
       STAGES
```

3.1.2 CAREIN Files

	File Name	Unit Name	Description
1.†	COVIN=Ø,	TAPE3 = COVIN,	coverage definition input file generated by this program for COVRGE program.
2.†	RELIN=Ø,	TAPE4 = RELIN,	system definition input file generated by this program for CARE3 program.
3.	INPUT,	TAPE5 = INPUT	
4.	OUTPUT,	TAPE6 = OUTPUT	
5.	CREIN,	TAPE7 = CREIN	
6.	CREOUT,	TAPE8 = CREOUT	
7.		TAPE10,	TAPE11 converted for READ*.
8.	FT11F,	TAPE11 = FT11F,	FTREE input.
9.†	BXYFL=Ø,	TAPE14 = BXYFL,	binary critical pair results file passed to CARE3 program.
10.†	FT15F,	TAPE15 = FT15F,	system minterm file passed to CARE3 program.
11.	FT16F,	TAPE16 = FT16F,	fault tree output from system fault tree definition.
12.	FT25F,	TAPE25 = FT25F,	critical pair minterm file.
13.	FT26F,	TAPE26 = FT26F,	fault tree output from critical pair fault tree definition(s).

†Must be saved and passed to other programs.

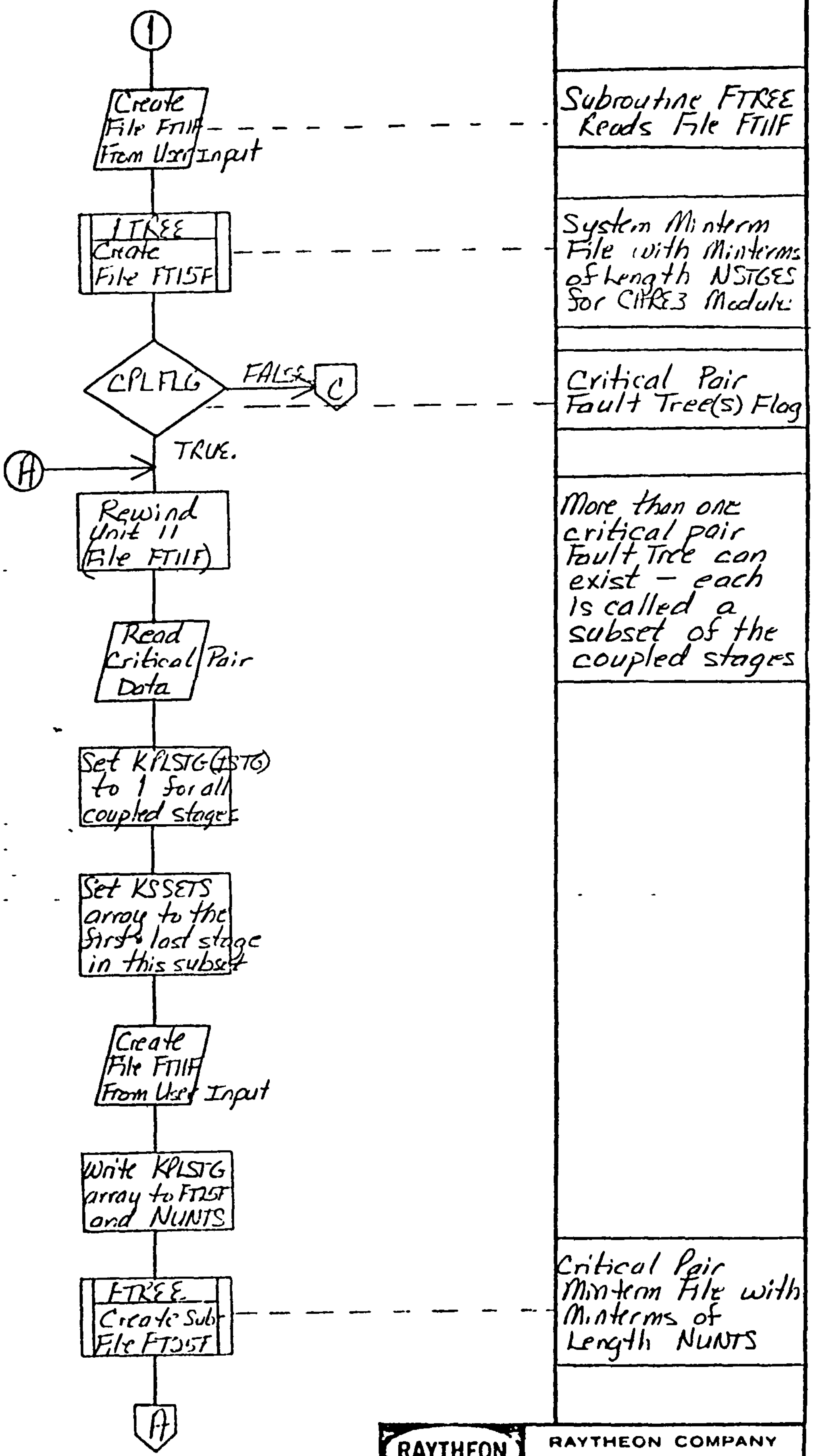


3.1.3 CAREIN Functional Flow Chart

RAYTHEON | RAYTHEON COMPANY LEXINGTON MASS 02173

CAREIN

49956

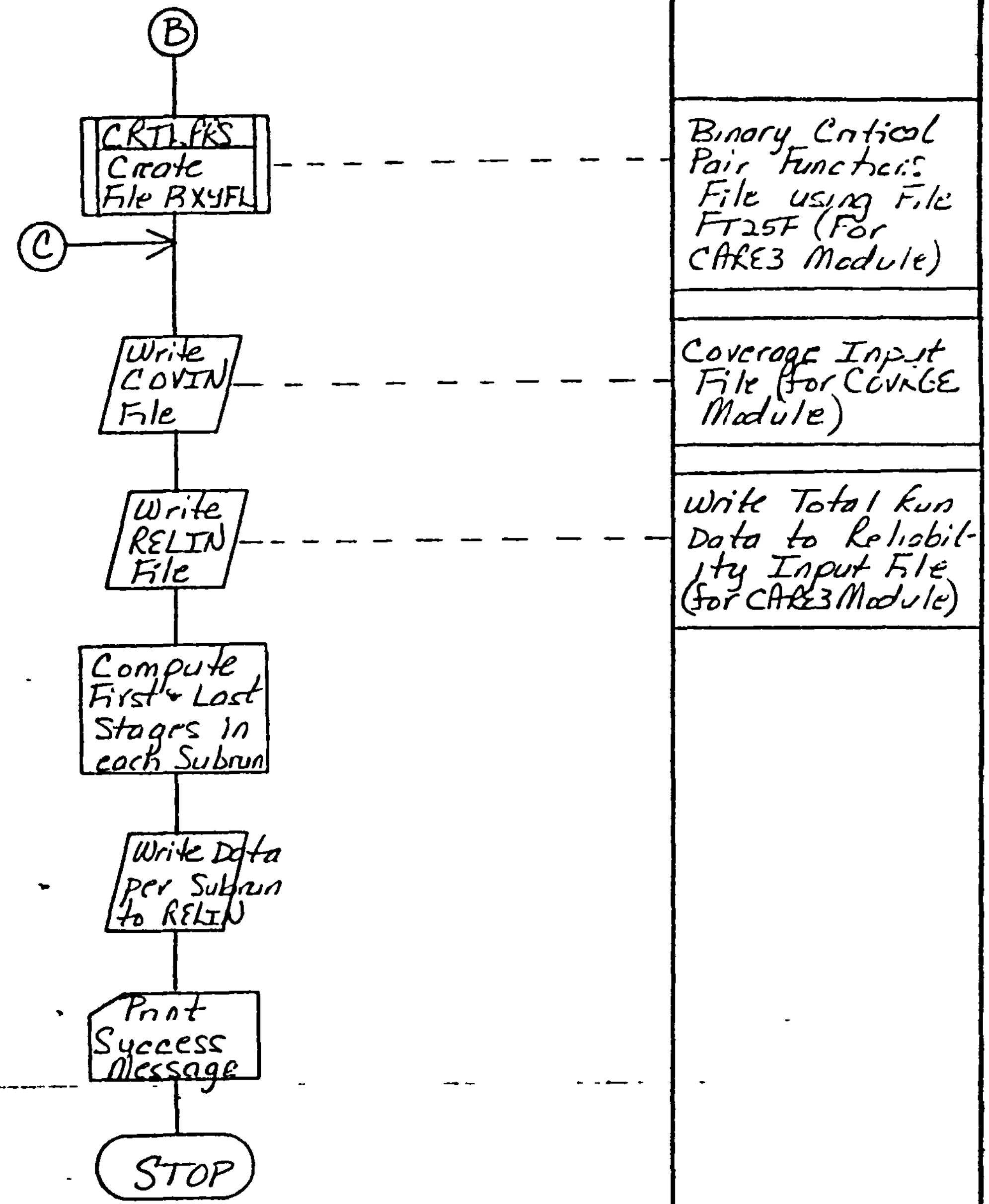


RAYTHEON | RAYTHEON COMPANY LEXINGTON MASS 02173

CARE IN

49956 | Lynn H. [illegible] 6/3/81

SHEET 2 OF 3



RAYTHEON

RAYTHEON COMPANY
LEXINGTON MASS 02173

CAREIN

49956

SHEET 3 OF 3

3.1.4 CAREIN Subroutines

```
C*****************************************************************
C
C*     SUBROUTINE BINOM
C
C*     THIS SECTION CALCULATES THE PROBABILITY OF
C*     M AT LEAST N.  M HAS A MAXIMUM VALUE OF 20.
C*     PROGRAM WILL HANDLE UP TO 10 N'S AND THEN
C*     DOES A FLIPFLOP TO CALCULATE AN N > 10.
C
C****************************************************************
C
C*          VARIABLE DEFINITIONS
C
C*          Q1   20  PROBABILITY
C*          Q2   20  COMPLEMENT OF PROBABILITY
C*          ST   1   CALCULATED PROBABILITY FOR EACH MINTERM
C*          C    20  COUNTER AND CONTROLLER
C*          NT   1   NUMBER OF TERMS(M)
C
C*          JJ   1   MINIMUM NUMBER OF SUCCESS(N)
C
C***************************************************************
      SUBROUTINE BINOM (NT,JJ,Q,ST)
      REAL Q(20),Q1(20),Q2(20),ST
```

```
EXTERNALS            TYPE   ARGS
      CALC                   5
```

```
C************************************************************
C
C*      SUBROUTINE CALC
C*      CALLED BY BINOM
C
C*      DOES ACTUAL CALCULATION FOR 'AT LEAST' CONDITION.
C
C************************************************************
      SUBROUTINE CALC (Q1,Q2,ST,C,NT)
      REAL Q1(20),Q2(20),SS,S,ST
```

```
      SUBROUTINE CRTLPRS(N,M,NOP,LC)
C
      COMMON// P(2000),T(50),SUM(2000),PR(70),LS,LTT,SAV,MSENS,ID,
     1 QP(70),PQ(70),BOG,BIG,DELTA,IC(2000),IN(2000),IP(10000),IR(2000),
     2 IDEP(700),IDUM(4300),CINDBG
      LOGICAL CINDBG,FATAL,EOFFLG
      DIMENSION KNTMTS(10,10,210),MNTRMV(70),MTSTGS(70)
      EQUIVALENCE (IC(1),KNTMTS(1,1,1))
C
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
      DIMENSION BCOMAR(5802)
      EQUIVALENCE (BXYAR(1,1,1),BCOMAR(1))
      DIMENSION N(70),M(70),NOP(5,70),LC(70)
```

```
       EXTERNALS            TYPE    ARGS
              BUFBLK                  6
              EOF         REAL        1
              FNCK        REAL        2

       INLINE FUNCTIONS     TYPE    ARGS
              FLOAT       REAL        1   INTRIN
              MINO        INTEGER     0   INTRIN
```

```
      SUBROUTINE DFIELD(ARRAY,IDIM,ITAPE,LN)
C*   SUBROUTINE DFIELD READS FREE FIELD FORMAT
C
      DIMENSION ARRAY(50),GATE(7)
```

```
    EXTERNALS            TYPE    ARGS
          EOF         REAL         1
```

```
      SUBROUTINE FTREE(IUNIT)
C*********************************************************************
C*            TITLE:  SIMPLIFIED COMPUTER EVALUATION OF FAULT TREES
C*            ACRONYM: SCEFT
C*            AUTHOR: C. R. STANDER
C*            DATE:  OCT. 25, 1976
C*            DATE EDITED:  4-22-81
C*            EDITED BY:  LYNN BRYANT
C
C*            PURPOSE:  SCEFT PROVIDES THE ANALYST REQUIRED TO WORK WITH
C*                      FAULT TREES A CAPABILITY TO QUICKLY AND SIMPLY
C*                    EVALUATE THE PROBABILITIES OF ALL EVENTS ON THE
C*                    TREE.
C
C*            THE TYPES OF DEPENDENCIES THE PROGRAM WILL HANDLE ARE:
C*              1. INPUT EVENTS THAT OCCUR IN MULTIPLE PLACES IN THE TREE
C*              2. COMMON TIME-SEQUENTIAL REDUNDANT PROBABILITIES
C*                 (CONDITIONAL-AND)
C*              3. COMMON ACTIVE AND PASSIVE REDUNDANCY WITH OR WITHOUT
C*                 SWITCHING DEVICES, AND
C*              4. EXCLUSIONS (EXCLUSIVE-OR).
C
```

```
EXTERNALS            TYPE    ARGS
       ALOG       REAL         1 LIBRARY
       DFIELD                  4
       EXP        REAL         1 LIBRARY
       MINTRMS                 4
       PROCIND                 0
       SORT                    2
       TTREE                   0

INLINE FUNCTIONS     TYPE    ARGS
       ABS        REAL         1  INTRIN
```

```
      SUBROUTINE GNMNFL(ZSUM,IUNIT,FRSTCM)
C
C  THIS SUBROUTINE WRITES VALID SYSTEM MINTERMS TO FILE 'FT15F'(IUNIT=
C  15) OR VALID CRITICAL PAIR MINTERMS TO FILE 'FT25F'(IUNIT=25).
C
      DIMENSION IIPP(70)
      LOGICAL FRSTCM,CINDBG
      COMMON// P(2000),T(50),SUM(2000),PR(70),LS,LTT,SAV,MSENS,ID,
     1 QP(70),PQ(70),BOG,BIG,DELTA,IC(2000),IN(2000),IP(10000),IR(2000),
     2 IDEP(700),IDUM(4300),CINDBG
C*  NOTE: 'IDUM' IS A PLACE-HOLDER FOR USE IN SUBROUTINE CTRLPRS WHEN
C*  'IC' IS EQUIVALENCED TO 'NNTMTS' DIMENSIONED TO (10,10,210).
C
```

```
      SUBROUTINE MINTRMS(NN,TSUM,IUNIT,FRSTCM)
C
      LOGICAL FRSTCM,CINDBG
      COMMON// P(2000),T(50),SUM(2000),PR(70),LS,LTT,SAV,MSENS,ID,
     1 QP(70),PQ(70),BOG,BIG,DELTA,IC(2000),IN(2000),IP(10000),IR(2000),
     2 IDEP(700),IDUM(4300),CINDBG
C*  NOTE: 'IDUM' IS A PLACE-HOLDER FOR USE IN SUBROUTINE CTRLPRS WHEN
C*  'IC' IS EQUIVALENCED TO 'KNTMTS' DIMENSIONED TO (10,10,210).
C


C*          THIS SECTION CALCULATES EACH MINTERM AS EACH
C*          MINTERM FAILS.  MCOMB DETERMINES THE NUMBER OF MINTERMS
C*          THAT WILL BE EVALUATED BY THE SYSTEM.  THE MAXIMUM NUMBER
C*          FOR MCOMB IS 13.
C
```

```
       EXTERNALS           TYPE   ARGS
              GNMNFL                 3

              TTREE                  0
```

```
      SUBROUTINE PROCIND
C*  THIS ROUTINE PREPROCESSES INPUT DATA SO THAT SUBROUTINE DFIELD CAN
C*  READ THE INFORMATION USING FORTRAN READ*.  NOTE: NON-NUMERIC DATA
C*  IS CONVERTED TO A NEGATIVE INTEGER REPRESENTING THE HOLLERITH
C*  CHARACTER, INSTEAD OF DELIMITING THE CHARACTER BY 'NOT EQUAL SIGNS'.
C
```

```
        EXTERNALS            TYPE    ARGS
              EOF          REAL        1
```

```
C***********************************************************
C
C*      SUBROUTINE SORT
C*      PUTS GATES IN ASCENDING ORDER
C
C***********************************************************
C*        VARIABLE DEFINITION
C
C*        N 1  NUMBER OF TERMS TO BE SORTED
C*        C 50 TERMS TO BE SORTED
C
C***********************************************************
      SUBROUTINE SORT (N,C)
      INTEGER C(50),D
```

```
      SUBROUTINE SPLIT(NKSTGF,NKSTGL,MXCPLS,MXSBRN,ISBRN,ISBRAR)
C
C  THIS SUBROUTINE SPLITS A RANGE OF NON-COUPLED STAGES INTO 'MXCPLS' OR
C  LESS SETS OF APPROXIMATELY EQUAL SIZE.  THE FIRST AND LAST STAGES OF
C  EACH SET FOR EACH SUBRUN ARE THEN STORED IN 'ISBRAR'(I,ISBRN), WHERE
C  'ISBRAR'(1,ISBRN) = FIRST STAGE IN THE SUBSET AND
C  'ISBRAR'(2,ISBRN) = LAST STAGE IN THE SUBSET; 'ISBRN' IS THEN
C  INCREMENTED.
C
      DIMENSION ISBRAR(2,MXSBRN)
```

```
      SUBROUTINE SUBRUN(KSSETS,NSTGES,MXCPLS,MXSBRN,ISBRN,ISBRAR)
C
C  THIS SUBROUTINE DIVIDES 'NSTGES', SOME OF WHICH ARE COUPLED, I.E.
C  THEY DEFINE CRITICAL-PAIRS OF FAULTS, INTO SUBRUNS.
C  'KSSETS'(I,ISBSET) DEFINES ALL CRITICAL-PAIR FAULT SUBSETS.
C  ** EACH SUCH SUBSET MUST START AND END A NEW SUBRUN. **
C  SUBROUTINE SPLIT IS USED TO DIVIDE THE RANGE OF NON-COUPLED STAGES
C  INTO SUBRUNS.
C
      DIMENSION KSSETS(2,MXSBRN),ISBRAR(2,MXSBRN)
```

EXTERNALS	TYPE	ARGS
SPLIT		6

```
C*************************************************************
C
C*       SUBROUTINE TTREE
C*       CALCULATES PROBABILITIES DEPENDING ON
C*       TYPE OF GATE(IR)            DCBB78
C
C************************************************
C
C*          Q          20  CONTAINS PROBABILITY FOR EACH INPUT GATE
C*                         THAT FEEDS INTO 'AT LEAST' GATE
C*          SUM      2000  HOLDS PROBABILITY OF EACH LOGIC GATE
C*          ST          1  RETURNS PROBABILITY FROM 'BINOM'
C*          SS          1  TEMP STORAGE FOR PROBABILITIES
C***************************************************************
      SUBROUTINE TTREE
      DIMENSION Q(20)
      COMMON// P(2000),T(50),SUM(2000),PR(70),LS,LTT,SAV,MSENS,ID,
     1 QP(70),PQ(70),BOG,BIG,DELTA,IC(2000),IN(2000),IP(10000),IR(2000),
     2 IDEP(700),IDUM(4300),CINDBG
      LOGICAL CINDBG
C*  NOTE: 'IDUM' IS A PLACE-HOLDER FOR USE IN SUBROUTINE CTRLPRS WHEN
C*  'IC' IS EQUIVALENCED TO 'KNTMTS' DIMENSIONED TO (10,10,210).
C
```

```
      EXTERNALS            TYPE    ARGS
             BINOM                   4
```

3.2 COVRGE Module

The COVRGE module consists of two main programs, COVRGE and CVGPLT, and several subroutines and functions. Program COVRGE computes the state transition rates required by program CARE3. (See the CARE III PHASE II Report, Mathematical Description* for a description of the mathematical model used in the coverage model implementation.) Program CVGPLT creates the plot file of the single- and double-fault functions generated in this module.

For further information regarding the mathematical equations used in the coverage model see the Appendix entitled "Computational Complexity in Solving the 'Stiff' Volterra Equation."

---

*J. J. Stiffler and L. A. Bryant, CARE III PHASE II Report, Mathematical Description, NASA CR-3566, 1982.

3.2.1 COVRGE and CVGPLT Main Programs

```
      PROGRAM COVRGE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,DBUG,
     1  CVGMTS=0,TAPE7=CVGMTS,TAPE8=DBUG,SNGFL=0,TAPE9=SNGFL,
     2  DBLFL=0,TAPE10=DBLFL,COVIN=0,TAPE3=COVIN)
C
C***********************************************************************
C*                                                                     *
C*        TITLE:  COMPUTER-AIDED RELIABILITY ESTIMATION, VERSION THREE *
C*        ACRONYM: CARE III                                            *
C*        MODEL DESIGN: DR. J.J.STIFFLER                               *
C*        SOFTWARE DESIGN AND IMPLEMENTATION: LYNN A. BRYANT           *
C*        DATE: MARCH 23,1981                                          *
C*        DATE EDITED: 5-02-81                                         *
C*        EDITED BY: LYNN A. BRYANT                                    *
C*                                                                     *
C*        PURPOSE:  CARE III IS A GENERAL-PURPOSE RELIABILITY          *
C*                  ESTIMATION TOOL FOR FAULT-TOLERANT AVIONICS        *
C*                  SYSTEMS.                                           *
C*                                                                     *
C***********************************************************************
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      DIMENSION REC1(4),REC2(3)
      EQUIVALENCE (ITSTPS,REC1(1)),(NTYPS,REC2(1))
C
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      DIMENSION REC3(3)
      EQUIVALENCE (CVPRNT,REC3(1))
C
      LOGICAL DELCDN,RHOCDN,EPSCDN
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      DIMENSION REC4(55)
      EQUIVALENCE (ALPHA(1),REC4(1))
C
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
?? /P20
C  FOR OPTIMIZATION PURPOSES DURING THE BUFFER OUT OF /CVRGCOM/,
C  DEFINE AND EQUIVALENCE 'CVRGAR' TO ENCOMPASS THE ENTIRE COMMON AREA.
      DIMENSION CVRGAR(9805)
      EQUIVALENCE (CMST0(1,1,1),CVRGAR(1))
      LOGICAL TRANS,CVPRNT,CVPLOT,FATAL,EOFFLG


        EXTERNALS          TYPE    ARGS
               ALOG        REAL      1 LIBRARY
               BUFBLK                6
               DBLFLT                0
               PRNTCVG               0
               SNGFLT                0
```

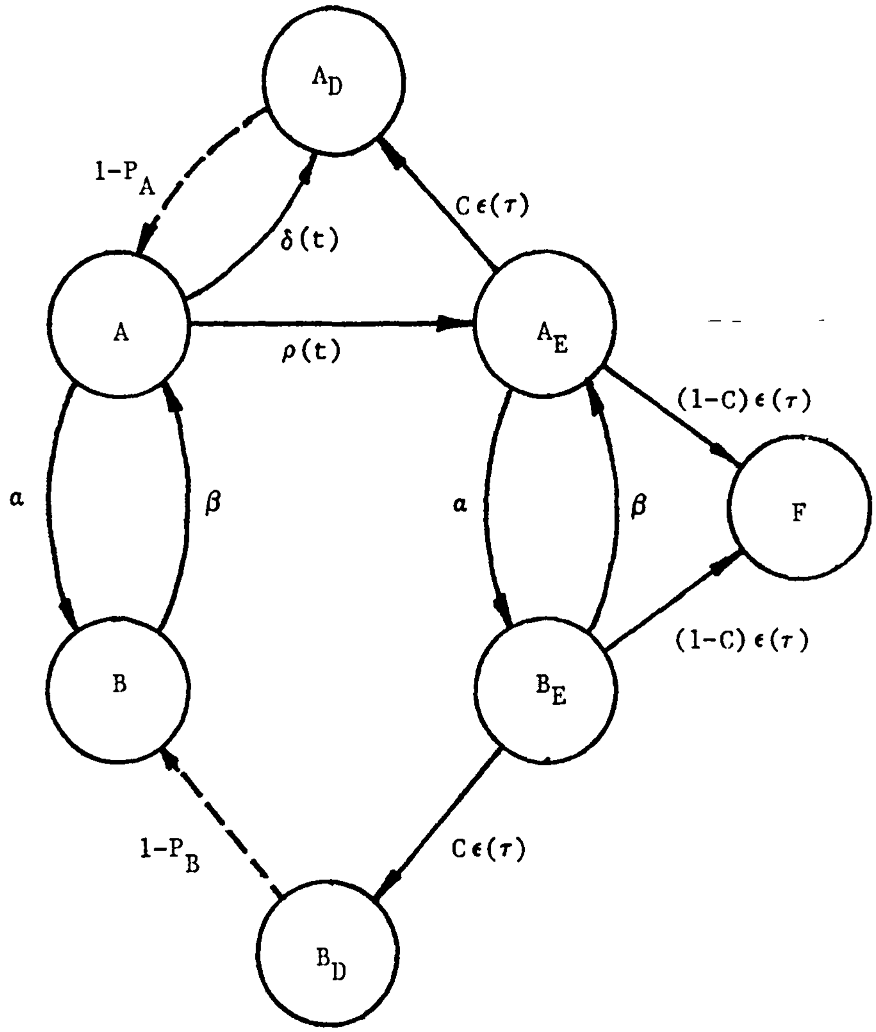


Figure 3.2.1.1
CARE III SINGLE FAULT MODEL

α = active to benign transition rate

β = benign to active transition rate

$\rho(t)$ = error generation rate

$\epsilon(t)$ = error propagation rate

$\delta(t)$ = fault detection rate

t = time from entry into active state

τ = time from entry into error state

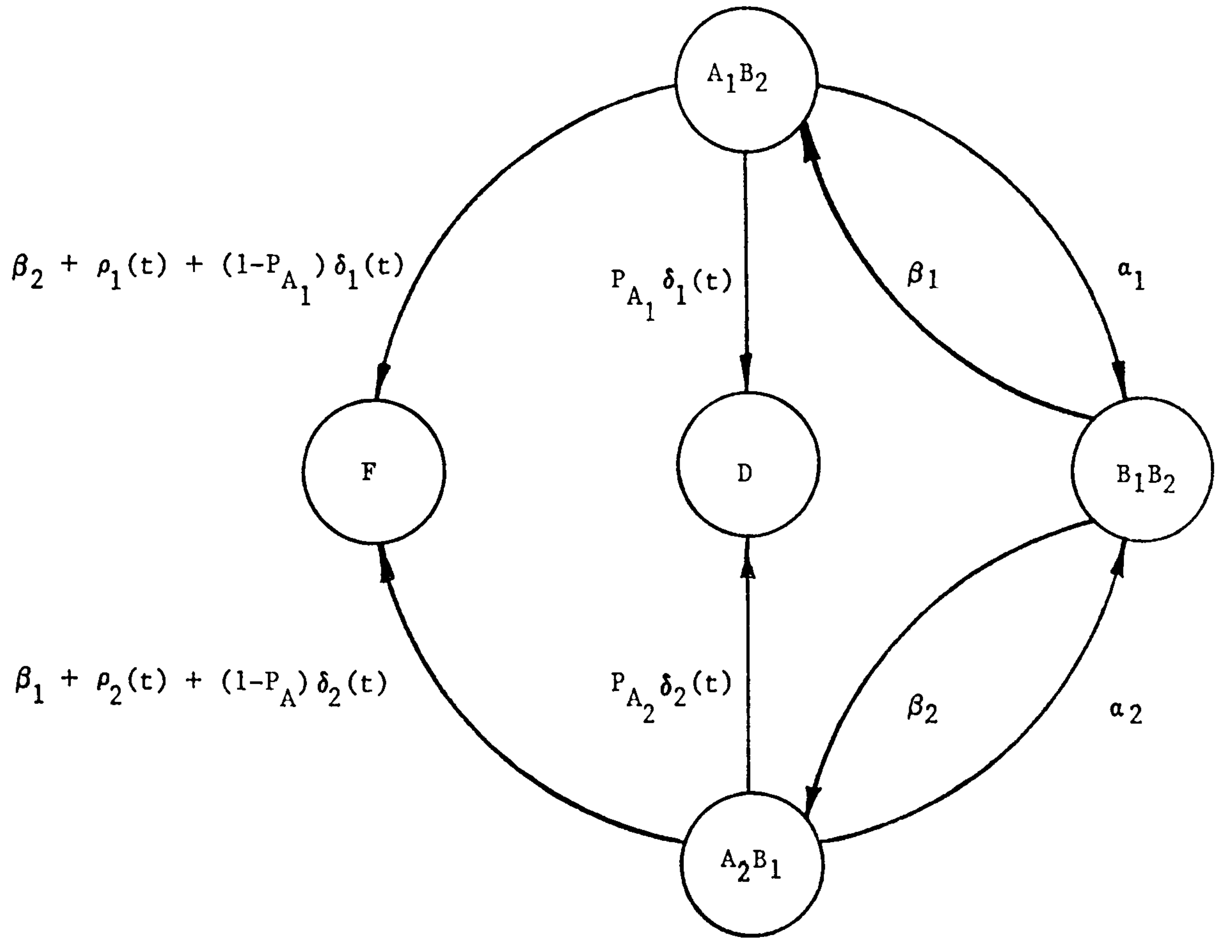


Figure 3.2.1.2

CARE III DOUBLE FAULT MODEL

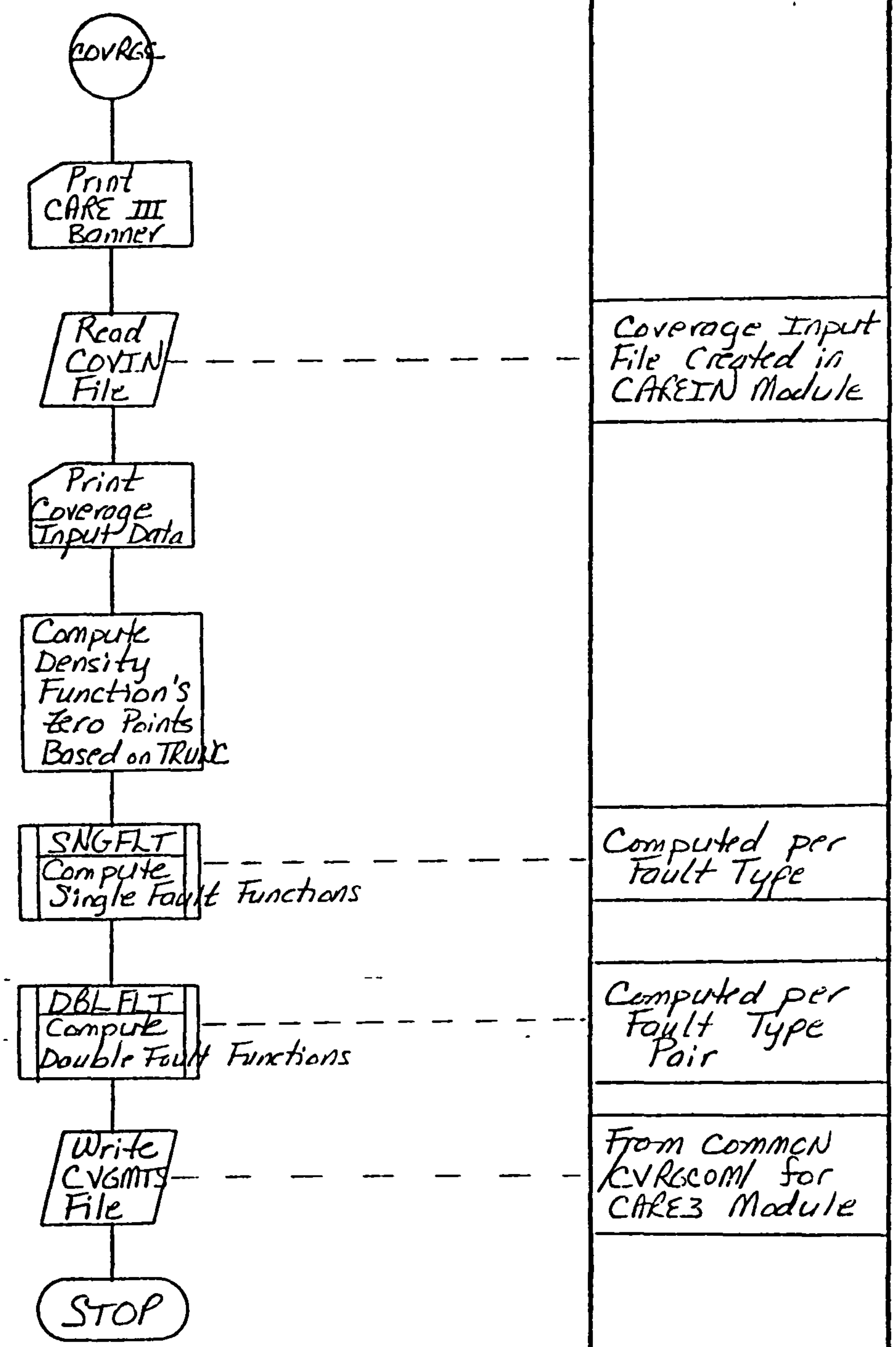
```
      PROGRAM CVGPLT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PLFILE,
     1  TAPE4=PLFILE,SNGFL=0,TAPE9=SNGFL,DBLFL=0,TAPE10=DBLFL,COVIN=0,
     2  TAPE3=COVIN)
C
C***********************************************************************
C*                                                                     *
C*        TITLE:  COMPUTER-AIDED RELIABILITY ESTIMATION, VERSION THREE *
C*        ACRONYM: CARE III                                            *
C*        MODEL DESIGN: DR. J.J.STIFFLER                               *
C*        SOFTWARE DESIGN AND IMPLEMENTATION: LYNN A. BRYANT           *
C*        DATE: MARCH 23,1981                                          *
C*        DATE EDITED: 4-30-81                                         *
C*        EDITED BY: LYNN A. BRYANT                                    *
C*                                                                     *
C*        PURPOSE:  CARE III IS A GENERAL-PURPOSE RELIABILITY          *
C*                  ESTIMATION TOOL FOR FAULT-TOLERANT AVIONICS        *
C*                  SYSTEMS.                                           *
C*                                                                     *
C***********************************************************************
C
C  THIS PROGRAM PLOTS THE COVERAGE SINGLE AND DOUBLE-FAULT FUNCTIONS.
C  IT BUFFERS IN FROM FILE 'SNGFL' THE TWENTY-FIVE POSSIBLE SINGLE-
C  FAULT FUNCTIONS (FIVE PER FAULT CATEGORY).  IT THEN BUFFERS IN
C  FROM FILE 'DBLFL' THE TWENTY-FIVE POSSIBLE DOUBLE-FAULT FUNCTIONS
C  (ALL COMBINATIONS OF PAIRED FAULT CATEGORIES).
C  THE SINGLE-FAULT FUNCTIONS ARE BUFFERED IN AND PLOTTED PER FAULT
C  CATEGORY.  THE DOUBLE-FAULT FUNCTIONS ARE BUFFERED IN AND PLOTTED
C  PER FAULT CATEGORY PAIR.
C  NOTE - TAPE4=PLFILE IS USED BY THE DISSPLA ROUTINES.
C
      COMMON/SNGFUNC/ PBNG(513),PBGSTEP,NPBGSTP(20),
     1                PNBNG(513),PNBSTEP,NPNBSTP(20),
     2                PFLD(513),PFSTEP,NPFSTP(20),
     3                PLAT(513),PLTSTEP,NPLTSTP(20),
     4                PDP(513),PDPSTEP,NPDPSTP(20),LNORLG
C  FOR OPTIMIZATION PURPOSES DURING THE BUFFER IN OF /SNGFUNC/,
C  DEFINE 'PLTSNG' TO ENCOMPASS THE ENTIRE COMMON AREA.
      DIMENSION PLTSNG(2671)
      EQUIVALENCE (PBNG(1),PLTSNG(1))
```

```
   EXTERNALS           TYPE    ARGS
          BUFBLK                 6
          CPLOT                  8
          DONEPL                 0
          GENTMAR                4
          STPINDX                5
```

3.2.2 COVRGE and CVGPLT Files

File Name	Unit Name	Description
COVRGE Files:		
1. COVIN=Ø,	TAPE3 = COVIN,	coverage definition input file generated by CAREIN program.
2. INPUT,	TAPE5 = INPUT	
3. OUTPUT,	TAPE6 = OUTPUT	
4.† CVGMTS=Ø,	TAPE7 = CVGMTS,	binary coverage moment functions passed to CARE3 program.
5. DBUG,	TAPE8 = DBUG	
6.† SNGFL=Ø,	TAPE9 = SNGFL,	plot file of single-fault probability functions, passed to CVGPLT program.
7.† DBLFL=Ø,	TAPE10 = DBLFL,	plot file of double-fault probability functions, passed to CVGPLT program.
CVGPLT Files:		
1. COVIN=Ø,	TAPE3 = COVIN,	coverage input file required in CVGPLT also.
2.* PLFILE,	TAPE4 = PLFILE,	DISSPLA plot file.
3. INPUT,	TAPE5 = INPUT	
4. OUTPUT,	TAPE6 = OUTPUT	
5. SNGFL=Ø,	TAPE9 = SNGFL,	single-fault binary file from COVRGE program.
6. DBLFL=Ø,	TAPE10 = DBLFL,	double-fault binary file from COVRGE program.

†Must be saved and passed to other programs.
*Must be saved and passed to DISSPLA plot program.



3.2.3 COVRGE Functional Flow Chart

RAYTHEON | RAYTHEON COMPANY LEXINGTON MASS 02173
COVRGE
49956 Flynn H. Bryant 5/3/81
SHEET 1 OF 1

3.2.4 COVRGE Subroutines

```
      SUBROUTINE DBLFLT
C
C  THIS SUBROUTINE COMPUTES THE DOUBLE-FAULT COVERAGE FUNCTION FOR
C  COVERAGE FAULT TYPE PAIR ('ITYP','JTYP').
C
      COMMON/WRKING/ B1AR(513),B1STEP,NB1STP(20),B2AR(513),B2STEP,
     1  NB2STP(20),DUM(513),NDUM(20),STPINIT(9)
      COMMON/FUNCCOM/ C1AR(513),C1STEP,NC1STP(20),
     1                C2AR(513),C2STEP,NC2STP(20),
     2                F1AR(513),F1STEP,NF1STP(20),
     3                F2AR(513),F2STEP,NF2STP(20),
     4                C3AR(513),C3STEP,NC3STP(20),
     5                C4AR(513),C4STEP,NC4STP(20),
     6                P3AR(513),P3STEP,NP3STP(20),
C  *********** STORAGE AREA FOR FINAL DOUBLE-FAULT FUNCTION ************
     7                PDFAR(513),PDFSTEP,NPDFSTP(20),LNORLG,
C  *********************************************************************
     8                XB1INTG(513),XB1STEP,NXB1STP(20),
     9                XB2INTG(513),XB2STEP,NXB2STP(20)
```

EXTERNALS	TYPE	ARGS
BUFBLK		6
COMPFUN		6
FBDBL		0
FCDBL		0
FFDBL		0
GENMNTS		4
PREVNRC		13
SUMARS		11
TMAXDBL		7
VOLTERA		10

See the next page for the Double-fault Model Equations.

DOUBLE-FAULT MODEL EQUATIONS

$$\left.\begin{aligned}
\alpha_{\substack{1\\2}}(t) &= \alpha_{\substack{1\\2}}\, e^{-\alpha_{\substack{1\\2}} t} \\
a_{\substack{1\\2}}(t) &= e^{-\alpha_{\substack{1\\2}} t} \\
\beta_{\substack{1\\2}}(t) &= \beta_{\substack{1\\2}}\, e^{-\beta_{\substack{1\\2}} t} \\
b_{\substack{1\\2}}(t) &= e^{-\beta_{\substack{1\\2}} t}
\end{aligned}\right\}\ \text{FUNCTION CNSRTDN}$$

$$\left.\begin{aligned}
d_{\substack{1\\2}}(t) &= 1 - \int_0^t \delta_{\substack{1\\2}}(\tau)\,d\tau \\
r_{\substack{1\\2}}(t) &= 1 - \int_0^t \rho_{\substack{1\\2}}(\tau)\,d\tau
\end{aligned}\right\}\ \text{FUNCTION RTDNINT}$$

VARIABLE

$$c_{\substack{1\\2}}(t) = \beta_{\substack{1\\2}}(t)\,d_{\substack{2\\1}}(t)\,r_{\substack{2\\1}}(t)\,a_{\substack{2\\1}}(t) + (1-P_{A_{\substack{2\\1}}})\,b_{\substack{1\\2}}(t)\,\delta_{\substack{2\\1}}(t)\,r_{\substack{2\\1}}(t)\,a_{\substack{2\\1}}(t) + b_{\substack{1\\2}}(t)\,d_{\substack{2\\1}}(t)\,\rho_{\substack{2\\1}}(t)\,a_{\substack{2\\1}}(t)$$

C1AR(IT)
C2AR(IT)

$$f_{\substack{2\\1}}(t) = \alpha_{\substack{1\\2}}(t)\,b_{\substack{2\\1}}(t)\,d_{\substack{1\\2}}(t)\,r_{\substack{1\\2}}(t)$$

F2AR(IT)
F1AR(IT)

$$c_4(t) = \int_0^t [c_1(t-\tau)\,\beta_2(\tau)\,b_1(\tau) + c_2(t-\tau)\,\beta_1(\tau)\,b_2(\tau)]\,d\tau$$

C4AR(IT)

$$c_3(t) = \int_0^t [f_1(t-\tau)\,\beta_2(\tau)\,b_1(\tau) + f_2(t-\tau)\,\beta_1(\tau)\,b_2(\tau)]\,d\tau$$

C3AR(IT)

$$p_3(t) = f_1(t) + \int_0^t c_3(t-\tau)\,p_3(\tau)\,d\tau$$

P3AR(IT)

$$p_{Df}(t) = c_1(t) + \int_0^t c_4(t-\tau)\,p_3(\tau)\,d\tau$$

PDFAR(IT)

```
      SUBROUTINE COMPFUN(STEPST,EXTFUN,ITH,TFMAX,NSTEPAR,FUNCAR)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      DIMENSION FUNCAR(1),NSTEPAR(1)
      LOGICAL THRASH,CVPRNT,CVPLOT,CHIDBG
      DATA DIVDIF/0.1/,MAXDIV/4/,REALMIN/1.0E-293/
C
C  THIS SUBROUTINE COMPUTES A FUNCTION USING INCREASING STEP SIZES,
C  WHERE THE STEP SIZE IS DOUBLED EACH TIME IT INCREASES.  THE
C  NUMBER OF STEPS COMPUTED FOR EACH STEP SIZE IS STORED IN NSTEPAR
C  ARRAY.
C
```

```
     EXTERNALS            TYPE    ARGS
            EXTFUN     REAL         2    F.P.

     INLINE FUNCTIONS      TYPE    ARGS
            ABS        REAL         1   INTRIN
            AMAX1      REAL         0   INTRIN
```

```
      SUBROUTINE CVLTAR(SFAR,SFSTPST,NSFSTP)
C
C   COMPUTE VOLTERRA EQUATION USING SINGLE-FAULT ARRAY 'SFAR', ITS
C   INITIAL STEP SIZE 'SFSTPST', AND ITS 'NSFSTP' ARRAY DEFINING HOW
C   MANY OF EACH "POWER OF TWO" TIMES 'SFSTPST' STEPS WERE GENERATED.
C   THESE ARRAYS WILL BE USED TO STORE THE FINAL RESULT OF THE CALL
C   TO SUBROUTINE 'VOLTERA' UPON RETURN.  'FAR','FSTPST' AND 'NFSTPAR'
C   WILL BE USED AS TEMPORARY STORAGE FOR USE IN THE VOLTERA SUBROUTINE.
C
C   'FAR(TIME)' = 'SFAR(TIME)' + INTEGRATION FROM 0.0 TO TIME OF
C   'FEEAR(TIME-TAU)' * 'FAR(TAU)'.
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/WRKING/ FAR(513),FSTPST,NFSTPAR(20),FEEAR(513),FEESTEP,
     1  NFEESTP(20),GAR(513),NGSTPAR(20),GSTPST(9)
      DIMENSION SFAR(513),NSFSTP(20)
```

EXTERNALS
        VOLTERA

```
      SUBROUTINE FILLDBL
C
C  THIS SUBROUTINE FILLS THE DOUBLE-FAULT FUNCTIONS IN COMMON/CVRGCOM/.
C
      COMMON/WRKING/ DFMOAR(513),DFMSTEP,NDFMSTP(20),DFM1AR(513),
     1       DFM1DUM,NDFM1DM(20),DFM2AR(513),NDFM2DM(20),DFM2DUM(9)
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      LOGICAL TRANS,CVPRNT,CVPLOT
      COMMON/TMARCOM/ DFTMAR(513),TMAR2(513),TMAR3(513),TMAR4(513)
C
```

EXTERNALS	TYPE	ARGS
FLINTP	REAL	5
GENTMAR		4
STPINDX		5

```
      SUBROUTINE FILLSNG(MCHI)
C
C  THIS SUBROUTINE FILLS THE SINGLE-FAULT FUNCTIONS IN COMMON/CVRGCOM/.
C
      COMMON/WRKING/ SFM0AR(513),SFMSTEP,NSFMSTP(20),SFM1AR(513),
     1       SFM1DUM,NSFM1DM(20),SFM2AR(513),NSFM2DM(20),SFM2DUM(9)
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      LOGICAL TRANS,CVPRNT,CVPLOT
      COMMON/TMARCOM/ SFTMAR(513),TMAR2(513),TMAR3(513),TMAR4(513)
C
```

EXTERNALS	TYPE	ARGS
FLINTP	REAL	5
GENTMAR		4
STPINDX		5

```
      SUBROUTINE GENMNTS(FLTAR,FLTSTEP,NFLTSTP,MCHI)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/WRKING/ FMOAR(513),FMSTEP,NFMSTP(20),FM1AR(513),FM1DUM,
     1    NFM1DM(20),FM2AR(513),NFM2DM(20),FM2DUM(9)
      LOGICAL CVPRNT,CVPLOT
      DIMENSION FLTAR(1),NFLTSTP(1)
      DATA CNSONE/1.0/
C
C  IF THE FAULT FUNCTION IS CONSTANTLY ZERO, THERE IS NO NEED TO COM-
C  PUTE THE MOMENT FUNCTIONS, SINCE THE MOMENT ARRAYS ARE INITIALIZED
C  TO ZERO.
```

```
       EXTERNALS          TYPE   ARGS
              FILLDBL               0
              FILLSNG               1
              VSTPINT               6
```

```
      SUBROUTINE GENTMAR(STEPST,NSTEPAR,TMAR,TMX)
C
      DIMENSION NSTEPAR(1),TMAR(1)
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      LOGICAL CHIDBG
C
C  THIS SUBROUTINE GENERATES AN ARRAY OF TIMES FOR WHICH A FUNCTION
C  WAS PREVIOUSLY COMPUTED.  'STEPST' IS THE INITIAL STEP SIZE USED
C  TO COMPUTE THE FUNCTION.  'NSTEPAR' IS THE ARRAY DEFINING HOW
C  MANY OF EACH 'POWER OF 2' TIMES 'STEPST' WERE USED.
C  I.E. NSTEPAR(NSTPIN) = THE NUMBER OF STEPS GENERATED HAVING
C  A STEP SIZE OF '2**(NSTPIN-1) * STEPST'.
C
```

```
      SUBROUTINE PREVNRC(F,FSTEP,NFSTEP,FEE,FEESTEP,NFEESTP,
     1   P,PSTEP,NPSTEP,CNSINTG,PAR,PARSTEP,NPARSTP)
C
      COMMON/TMARCOM/ FTMAR(513),FEETMAR(513),PTMAR(513),PARTMAR(513)
C
C  THIS SUBROUTINE'S MAIN FUNCTION IS AS A PREPROCESSOR FOR CALLING
C  SUBROUTINE 'VLTNREC'.  THE TWO ARRAYS INVOLVED WITHIN THE CONVOLUTED
C  INTEGRATION: 'FEE' AND 'P' MUST BE TESTED FOR THEIR TIME MAXIMUMS
C  (MAXIMUM X VALUE) AND PASSED TO SUBROUTINE 'VLTNREC' SO THAT THE
C  FUNCTION WITH THE LARGEST TIME MAXIMUM IS PASSED AS THE FUNCTION
C  THAT IS INTEGRATED FORWARD IN TIME.
C  EXAMPLE:
C  'PAR'(T) = INTEGRATION FROM 0.0 TO 'T' OF 'FEE'(T-TAU) * 'P'(TAU).
C  IF 'P' FUNCTION HAS THE LARGEST TIME MAXIMUM (X VALUE), THE FUNCTIONS
C  SHOULD BE PASSED IN THE SAME ORDER AS ABOVE.  IF 'FEE' FUNCTION HAS
C  THE LARGEST TIME MAXIMUM, THE EQUATION SHOULD BE AS FOLLOWS:
C  'PAR'(T) = INTEGRATION FROM 0.0 TO 'T' OF 'P'(T-TAU) * 'FEE'(TAU).
```

```
     EXTERNALS            TYPE   ARGS
            GENTMAR                 4
            VLTNREC                18
```

```
      SUBROUTINE PRNTCVG
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      LOGICAL TRANS
C
C  PRINT CVRGCOM COMMON AREA.
C
```

```
      SUBROUTINE SNGFLT
C
C  THIS SUBROUTINE COMPUTES THE SINGLE-FAULT COVERAGE FUNCTIONS FOR
C  COVERAGE FAULT TYPE 'ITYP'.
C
      COMMON/WRKING/ FAR(513),FSTPST,NFSTPAR(20),FEEAR(513),FEESTEP,
     1  NFEESTP(20),GAR(513),NGSTPAR(20),GSTPST(9)
      COMMON/FUNCCOM/ PA(513),PASTEP,NPASTP(20),
     1                PB1(513),PB1STEP,NPB1STP(20),
     2                PERR(513),PERSTEP,NPERSTP(20),
C  ********** FINAL AREA FOR STORAGE OF SINGLE-FAULT FUNCTIONS *********
     3                PEAR(513),PESTEP,NPESTP(20),
     4                PNEAR(513),PNESTEP,NPNESTP(20),
     5                PFLD(513),PFSTEP,NPFSTP(20),
     6                PSIA(513),PSASTEP,NPSASTP(20),
     7                PDP(513),PDPSTEP,NPDPSTP(20),LNORLG,
C  *******************************************************************
     8                PSIB(513),PSBSTEP,NPSBSTP(20),
     9                PB2(513),PB2STEP,NPB2STP(20)
```

EXTERNALS	TYPE	ARGS
BUFBLK		6
COMPFUN		6
CVLTAR		3
FGSNGL		0
GENMNTS		4
PREVNRC		13
SUMARS		11
TMAXSNG		2
VOLTERA		10
VSTPINT		6

See the next two pages for the Single-fault Model Equations.

SINGLE-FAULT MODEL EQUATIONS

	VARIABLE
$\phi(t) = \alpha e^{-\beta t} \int_0^t e^{-(\alpha-\beta)\tau} r(\tau) d(\tau) d\tau$	FEEAR(IT)
$P_a(t) = e^{-\alpha t} r(t) d(t) + \beta \int_0^t \phi(t-\tau) P_a(\tau) d\tau$	PA(IT)
$P_b(t) = \phi(t) + \beta \int_0^t \phi(t-\tau) P_b(\tau) d\tau$	PB1(IT)
$P_e(t) = \int_0^t e^{-\alpha\tau} \rho(\tau) d(\tau) e(t-\tau) d\tau + \beta \int_0^t \phi(t-\tau) P_e(\tau) d\tau$	PERR(IT)
$p_e(t) = e^{-\alpha t} \rho(t) d(t) + \beta \int_0^t \phi(t-\tau) p_e(\tau) d\tau$	PEAR(IT)
$p_{\bar{e}}(t) = e^{-\alpha t} \delta(t) r(t) + \beta \int_0^t \phi(t-\tau) p_{\bar{e}}(\tau) d\tau$	PNEAR(IT)
$p_f(t) = (1-C) \int_0^t p_e(\tau) \varepsilon(t-\tau) d\tau$	PFLD(IT)
$\Psi_A(t) = C \int_0^t p_e(\tau) \varepsilon(t-\tau) \left(\frac{\beta + \alpha e^{-(\alpha+\beta)(t-\tau)}}{\alpha+\beta} \right) d\tau + p_{\bar{e}}(t)$	PSIA(IT)
$\Psi_B(t) = \frac{\alpha C}{\alpha+\beta} \int_0^t p_e(\tau)(1 - e^{-(\alpha+\beta)(t-\tau)}) \varepsilon(t-\tau) d\tau$	PSIB(IT)
$\chi_B(t) = \int_0^t \Psi_B(\tau) e^{-\beta(t-\tau)} d\tau$	PB2(IT)
$P_{dp}(t) = P_A \int_0^t \Psi_A(\tau) d\tau + P_B \int_0^t \Psi_B(\tau) d\tau$	PDP(IT)

SINGLE FAULT MODEL EQUATIONS (CONT.)

VARIABLE

LET

$$F_X(t) = F_x(t) + \int_0^t [(1-P_A)\Psi_A(t-\tau) + (1-P_B)\beta\chi_B(t-\tau)]F_X(\tau)d\tau$$

WITH

$F_X(t)$	$P_{DP}(t)$	$P_A(t)$	$P_{B1}(t)$	$P_{B2}(t)$	$P_E(t)$	$p_F(t)$
$F_x(t)$	$P_{dp}(t)$	$P_a(t)$	$P_b(t)$	$\chi_B(t)$	$P_e(t)$	$p_f(t)$

$$P_B(t) = P_{B1}(t) + P_{B2}(t)$$ PBNG(IT)

$$P_{\overline{B}}(t) = P_A(t) + P_E(t)$$ PNBNG(IT)

$$P_L(t) = \begin{cases} P_B(t) + P_{\overline{B}}(t) & \text{PERMANENT FAULTS} \\ P_B(t) & \text{TRANSIENT FAULTS} \end{cases}$$ PLAT(IT)

```
      SUBROUTINE STPINDX(T,STEPST,NSTEPAR,STPSZE,INDX)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      DIMENSION NSTEPAR(1)
C
C  THIS SUBROUTINE COMPUTES THE STEP SIZE AND INDEX GIVEN A TIME
C  FOR A FUNCTION WITH ORIGINAL STEP SIZE OF 'STEPST'. 'NSTEPAR'
C  DEFINES HOW MANY OF EACH 'POWER OF TWO' TIMES 'STEPST' STEPS
C  WERE COMPUTED FOR THE FUNCTION.
C
```

```
      SUBROUTINE SUMARS(AR1,STEP1ST,NSTPAR1,CN1,AR2,STEP2ST,NSTPAR2,CN2,
     1           ARSUM,STEPST,NSTEPAR)
C
C   THIS SUBROUTINE SUMS TWO ARRAYS 'AR1' AND 'AR2' YIELDING 'ARSUM'.
C   'STEP1ST' AND 'STEP2ST' ARE ARRAYS' 'AR1' AND 'AR2' INITIAL STEP
C   SIZES RESPECTIVELY.  'NSTEPAR', WHICH WILL CONTAIN THE NUMBER OF
C   STEPS OF EACH "POWER OF TWO" TIMES 'STEPST' FOR 'ARSUM', IS
C   CREATED IN THIS SUBROUTINE ALSO.
C   'CN1' AND 'CN2' ARE CONSTANTS TO MULTIPLY EACH FUNCTION BY BEFORE
C   THE SUM.
C
      DIMENSION AR1(1),AR2(1),ARSUM(1),NSTPAR1(1),NSTPAR2(1),NSTEPAR(1)
      COMMON/TMARCOM/ A1TMAR(513),A2TMAR(513),TMAR3(513),TMAR4(513)
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      LOGICAL A1FNSH,A2FNSH,THRASH,CHIDBG
```

```
        EXTERNALS            TYPE    ARGS
               FLINTP     REAL         5
               GENTMAR                 4
               STPINDX                 5

        INLINE FUNCTIONS     TYPE    ARGS
               ABS        REAL         1   INTRIN
               AMAX1      REAL         0   INTRIN
               AMIN1      REAL         0   INTRIN
```

```
      SUBROUTINE TMAXDBL(TC1MX,TC2MX,TF1MX,TF2MX,TB1MX,TB2MX,STPINIT)
C
C  COMPUTE TIME MAXIMUMS FOR THE SIX INITIAL DOUBLE-FAULT FUNCTIONS,
C  TWO C FUNCTIONS, TWO F FUNCTIONS AND TWO B FUNCTIONS, BASED ON THE
C  TIME MAXIMUMS FOR THE VARIOUS RATE FUNCTIONS.
C
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      LOGICAL CVPRNT,CVPLOT,DELCDN,RHOCDN,EPSCDN
      DIMENSION STPINIT(9)
```

```
INLINE FUNCTIONS    TYPE    ARGS
       AMAX1       REAL       0  INTRIN
       AMIN1       REAL       0  INTRIN
```

```
      SUBROUTINE TMAXSNG(TSNGMX,STPINIT)
C
C  COMPUTE TIME MAXIMUMS FOR THE NINE INITIAL SINGLE-FAULT G FUNCTIONS
C  BASED ON THE TIME MAXIMUMS FOR THE VARIOUS RATE FUNCTIONS.
C
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      DIMENSION TSNGMX(9),STPINIT(9)
      LOGICAL CVPRNT,CVPLOT,DELCDN,RHOCDN,EPSCDN
```

```
     INLINE FUNCTIONS   TYPE   ARGS
            AMAX1      REAL       0  INTRIN
            AMIN1      REAL       0  INTRIN
```

```
      SUBROUTINE VLTNREC(F,FSTEP,NFSTEP,FTMAR,FEE,FEESTEP,NFEESTP,
     1   FEETMAR,P,PSTEP,NPSTEP,PTMAR,PTMX,CNSINTG,PAR,PARSTEP,NPARSTP,
     2   PARTMAR)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/TMPNTRS/ T,IFEEPNT,IPPNT,FEEFLG,PFLG
      DIMENSION F(1),FEE(1),P(1),PAR(1),NFSTEP(1),NFEESTP(1),NPSTEP(1),
     1  NPARSTP(1),FTMAR(1),FEETMAR(1),PTMAR(1),PARTMAR(1)
      LOGICAL THRASH,RECRSVF,FEEFLG,PFLG,CHIDBG
      DATA REALMIN/1.0E-293/
C
C  THIS SUBROUTINE IS VERY SIMILAR TO THE VOLTERA SUBROUTINE.  THE
C  ONLY DIFFERENCE IS THAT THIS SOLVES A NON-RECURSIVE CONVOLUTED
C  INTEGRAL WHERE ALL FUNCTIONS ARE KNOWN.
C  'PAR'(T) = 'F'(T) + THE INTEGRATION FROM ZERO TO 'T' OF
C  'FEE'(T-TAU) * 'P'(TAU).
C
```

```
        EXTERNALS            TYPE    ARGS
               CNVLINT   REAL      12
               FLINTP    REAL       5
               STPINDX              5

        INLINE FUNCTIONS     TYPE    ARGS
               ABS       REAL       1  INTRIN
               AMAX1     REAL       0  INTRIN
               AMIN1     REAL       0  INTRIN
```

```
      SUBROUTINE VOLTERA(F,FSTEP,NFSTEP,FEE,FEESTEP,NFEESTP,CNSINTG,
     1                   P,PSTEP,NPSTEP)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/TMPNTRS/ T,IFEEPNT,IPPNT,FEEFLG,PFLG
      DIMENSION F(1),FEE(1),P(1),NPSTEP(1),NFEESTP(1),NFSTEP(1)
      COMMON/TMARCOM/ FTMAR(513),FEETMAR(513),PTMAR(513),TMAR4(513)
      LOGICAL THRASH,STDYFLG,RECRSVF,FEEFLG,PFLG,CHIDBG
      DATA REALMIN/1.0E-293/
C
C   VOLTERA SOLVES ANY LINEAR NON-SINGULAR VOLTERRA INTEGRAL EQUATION
C   OF THE SECOND KIND WITH CONVOLUTION KERNEL, WHERE 'F','FEE'(KERNEL)
C   ARE GIVEN ARRAYS WITH INITIAL STEP SIZES 'FSTEP' AND 'FEESTEP'.
C   'P' IS THE COMPUTED ARRAY.  THE LOWER LIMIT OF INTEGRATION IS
C   ASSUMED TO BE 0.0.
```

```
EXTERNALS              TYPE    ARGS
       CNVLINT       REAL      12
       FLINTP        REAL       5
       GENTMAR                  4
       STPINDX                  5

INLINE FUNCTIONS       TYPE    ARGS
       ABS           REAL       1   INTRIN
       AMAX1         REAL       0   INTRIN
       AMIN1         REAL       0   INTRIN
```

```
      SUBROUTINE VSTPINT(AR,STEPST,NSTEPAR,CNSINTG,MOMNT,ARINT)
C
C  THIS SUBROUTINE INTEGRATES A FUNCTION STORED IN 'AR' WHICH HAS A
C  GIVEN INITIAL STEP SIZE - 'STEPST' AND A GIVEN SET OF STEPS WITH
C  INCREASING STEP SIZE.  'NSTEPAR' CONTAINS THE NUMBER OF STEPS GENER-
C  ATED FOR EACH 'POWER OF TWO' TIMES 'STEPST'.  'ARINT' WILL HAVE THE
C  SAME 'STEPST' AND 'NSTEPAR' AS 'AR'.
C  INTEGRATE THE ARRAY (Y VALUE) MULTIPLIED BY THE TIME (X VALUE)
C  RAISED TO THE 'MOMNT' POWER.
C  IF 'MOMNT' = 0, INTEGRATE THE ARRAY WITH NO MULTIPLIER.
C  IF 'MOMNT' = 1, INTEGRATE THE ARRAY MULTIPLIED BY THE TIME.
C  IF 'MOMNT' = 2, INTEGRATE THE ARRAY MULTIPLIED BY THE TIME SQUARED.
C
      DIMENSION AR(1),NSTEPAR(1),ARINT(1),REGINTG(20)
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      LOGICAL CHIDBG
      COMMON/TMARCOM/ TMAR(513),XFNATTM(513),TMAR3(513),TMAR4(513)
C
```

EXTERNALS	TYPE	ARGS
GENTMAR		4
SIMPINT	REAL	6
STPINDX		5

3.2.5 COVRGE Functions

```
      FUNCTION ARMOMNT(INDX,MOMNT,TMAR,AR)
C
      DIMENSION AR(1),TMAR(1)
C
C   THIS FUNCTION MULTIPLIES '(TMAR(INDX)**MOMNT)*AR(INDX)' FOR INTEGRA-
C   TION ROUTINE SIMPINT.
C   THE TEST FOR A VALID MOMNT IS HANDLED IN SIMPINT.
C
```

```
      FUNCTION CNSRTDN(RATE,TIME,DNFLG)
      LOGICAL DNFLG
C
C  THIS FUNCTION RETURNS A CONSTANT RATE VALUE GIVE 'RATE' AND 'TIME',
C  IF DNFLG IS .FALSE.

C
C  THIS FUNCTION RETURNS A CONSTANT DENSITY VALUE GIVEN 'RATE' AND 'TIME',
C  IF DNFLG IS .TRUE.
```

```
        EXTERNALS          TYPE    ARGS
              PREEXP    REAL        1
```

```
      FUNCTION CNVLINT(FVAL,FEE,FEETMAR,FEESTEP,NFEESTP,INDEXP,P,PTMAR,
     1                 PSTEP,NPSTEP,RECRSVF,CNSINTG)
C
C  THIS FUNCTION INTEGRATES FROM 0 TO 'T' (STORED IN COMMON/TMPNTRS/)
C  CONVOLVING TWO FUNCTIONS STORED IN ARRAYS FEE AND P.
C  IF RECRSVF, RECURSIVE FLAG, IS .TRUE., THE P FUNCTION USED
C  IN THE INTEGRATION IS THE FUNCTION BEING COMPUTED.
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      COMMON/TMPNTRS/ T,IFEEPNT,IPPNT,FEEFLG,PFLG
      LOGICAL PINTFLG,FEEINTF,FEEFLG,PFLG,RECRSVF
      DIMENSION FEE(1),P(1),FEETMAR(1),PTMAR(1),NFEESTP(1),NPSTEP(1)
C
```

```
          EXTERNALS           TYPE    ARGS
                 FCNVLTM   REAL       2
                 FLINTP    REAL       5
                 FTCHSTP   REAL       3
                 STPINDX              5
```

```
      FUNCTION FBDBL(ITH,T)
C
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      LOGICAL DELCDN,RHOCDN,EPSCDN,DNSFLG,CVPRNT,CVPLOT
      DATA DNSFLG/.FALSE./
C
C   COMPUTE DOUBLE-FAULT B FUNCTION 1 OR 2.  THE TWO FUNCTIONS, IN-
C   VOLVING 'BETA', USED TO DEFINE THIS FUNCTION ARE CONSTANT RATE
C   FUNCTIONS.
C
```

```
EXTERNALS          TYPE    ARGS
       CNSRTDN    REAL       3
       RTDNINT    REAL       3
```

```
      FUNCTION FCDBL(ITH,T)
C
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      LOGICAL DELCDN,RHOCDN,EPSCDN,DNSFLG,CVPRNT,CVPLOT
C
C  COMPUTE DOUBLE-FAULT C FUNCTION 1 OR 2.  ALL FUNCTIONS USED TO COM-
C  PUTE THIS C FUNCTION ARE CONSTANT RATE FUNCTIONS UNLESS FLAGS
C  'RHOCDN','DELCDN','EPSCDN' SPECIFY CONSTANT DENSITY FOR FUNCTIONS
C  INVOLVING 'RHO','DELTA' OR 'EPSLN'.
C
```

EXTERNALS	TYPE	ARGS
CNSRTDN	REAL	3
RTDNINT	REAL	3

```
      FUNCTION FCNVLTM(FEETMAR,PTMAR)
C
      DIMENSION FEETMAR(1),PTMAR(1)
      COMMON/TMPNTRS/ T,IFEEPNT,IPPNT,FEEFLG,PFLG
      LOGICAL FEEFLG,PFLG
C
C   FCNVLTM - FETCH CONVOLUTED TIME.
C   THIS FUNCTION RETURNS THE MAXIMUM TIME NOT YET FETCHED FROM EITHER
C   'T-FEETMAR' OR PTMAR.  IFEEPNT IS THE POINTER INTO FEETMAR; IPPNT IS
C   THE POINTER INTO PTMAR.  FEEFLG AND PFLG ARE INTERPOLATION FLAGS;
C   IF BOTH ARE SET .TRUE., THEY ARE USED AS FLAGS SPECIFYING THE
C   COMPLETION OF TIMES RETRIEVED FROM ONE OF THE ARRAYS.
C
```

```
      FUNCTION FFDBL(ITH,T)
C
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      LOGICAL DELCDN,RHOCDN,EPSCDN,DNSFLG,CVPRNT,CVPLOT
C
C  COMPUTE DOUBLE-FAULT F FUNCTION 1 OR 2.  ALL FUNCTIONS USED TO
C  COMPUTE THIS F FUNCTION ARE CONSTANT RATE FUNCTIONS UNLESS FLAGS
C  'RHOCDN','DELCDN','EPSCDN' SPECIFY CONSTANT DENSITY FOR FUNCTIONS
C  INVOLVING 'RHO','DELTA' OR 'EPSLN'.
C
```

```
    EXTERNALS          TYPE   ARGS
          CNSRTDN    REAL      3
          RTDNINT    REAL      3
```

```
      FUNCTION FGSNGL(ITH,T)
C
C  COMPUTE SINGLE-FAULT G FUNCTIONS 1 THROUGH 9.
C
      COMMON/FLTYPCM/ ITYP,JTYP,CVPRNT,CVPLOT,IAXSTYP
      COMMON/RATECOM/ ALPHA(5),BETA(5),DELTA(5),RHO(5),EPSLN(5),
     1                DELCDN(5),RHOCDN(5),EPSCDN(5),
     2                PACNS(5),PBCNS(5),CCNS(5),
     3                TALPMX(5),TBETMX(5),TDELMX(5),TRHOMX(5),TEPSMX(5)
      LOGICAL DELCDN,RHOCDN,EPSCDN,DNSFLG,CVPRNT,CVPLOT
      DATA DNSFLG/.FALSE./
C
C  ALL FUNCTIONS ARE CONSTANT RATE FUNCTIONS UNLESS FLAGS DELCDN,RHOCDN,
C  EPSCDN SPECIFY CONSTANT DENSITY FOR FUNCTIONS INVOLVING RHO,DELTA
C  OR EPSLN.
C
```

```
EXTERNALS            TYPE    ARGS
      CNSRTDN     REAL      3
      PREEXP      REAL      1
      RTDNINT     REAL      3
```

```
      FUNCTION FLINTP(T,STEP,ARTOINT,J,TIMEJ)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      DIMENSION ARTOINT(1)
C
C  THIS FUNCTION LINEARLY INTERPOLATES THE ARTOINT ARRAY WITH
C  NON-UNIFORM STEP SIZES.
C
```

```
      FUNCTION FTCHSTP(STEPST,NSTEPAR,INDX)
C
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
      DIMENSION NSTEPAR(1)
C
C   GIVEN AN INDEX INTO AN ARRAY WITH NON-UNIFORM STEP SIZES
C   (NOTE: THE STEP SIZES ARE 'POWERS OF 2' TIMES THE INITIAL
C   STEP SIZE - 'STEPST'),RETURN THE STEP SIZE ASSOCIATED WITH
C   THAT POINT.
C   NSTEPAR CONTAINS THE NUMBER OF STEPS GENERATED FOR EACH
C   'POWER OF 2' TIMES 'STEPST'.
C
```

```
      FUNCTION RTDNINT(RATE,TIME,DNFLG)
      LOGICAL DNFLG
C
C  THIS FUNCTION COMPUTES '1-THE INTEGRATION FROM 0 TO 'TIME' OF THE
C  CONSTANT RATE FUNCTION' EXPLICITLY, IF DNFLG IS .FALSE.

C
C  THIS FUNCTION COMPUTES '1-THE INTEGRATION FROM 0 TO 'TIME' OF THE
C  CONSTANT DENSITY FUNCTION' EXPLICITLY, IF DNFLG IS .TRUE.
```

```
     EXTERNALS          TYPE   ARGS
           PREEXP    REAL        1
```

```
      FUNCTION SIMPINT(ITFROM,ITTO,STPSZE,MOMNT,TMAR,AR)
C
      DIMENSION AR(1),TMAR(1)
      COMMON/CVGSTEP/ ITSTPS,MAXSTP,RELSTEP,TBASE,NTYPS,DBLDF,TRUNC,
     1                INMAX,NSTPMX,DIFCHNG
C
C  THIS FUNCTION INTEGRATES THE FUNCTION IN 'AR' MULTIPLIED BY
C  (TIME**MOMNT) FROM 'ITFROM' TO 'ITTO' WITH UNIFORM STEP SIZE
C  'STPSZE' OVER THIS REGION.
C
```

```
EXTERNALS            TYPE    ARGS
       ARMOMNT    REAL        4

INLINE FUNCTIONS     TYPE    ARGS
       MOD        INTEGER     2   INTRIN
```

3.3 CARE3 Module

The CARE3 module consists of two main programs, CARE3 and RELPLT, and several subroutines and functions. Program CARE3 calculates the system reliability by calculating the failed state probabilities $Q_{\ell}(t)$. (See the CARE III Final Report* for a description of the mathematical model used in the reliability model implementation.) Program RELPLT creates the plot file of the summary reliability functions.

---

*J. J. Stiffler and L. A. Bryant, CARE III Final Report, Phase II, Preliminary, Raytheon Company, May 1981.

3.3.1 CARE3 and RELPLT Main Programs

```
      PROGRAM CARE3(RELIN=0,TAPE4=RELIN,INPUT,OUTPUT,TAPE5=INPUT,
     1 TAPE6=OUTPUT,CVGMTS=0,TAPE7=CVGMTS,DBUG,TAPE8=DBUG,TAPE9=0,
     2 TAPE10=0,TAPE11=0,TAPE12=0,GHFNCS,TAPE13=GHFNCS,BXYFL=0,
     3 TAPE14=BXYFL,FT15F,TAPE15=FT15F,PLTFL=0,TAPE16=PLTFL,TAPE17=0)
C
C
C***********************************************************************
C*                                                                     *
C*       TITLE:  COMPUTER-AIDED RELIABILITY ESTIMATION, VERSION THREE *
C*       ACRONYM: CARE III                                             *
C*       MODEL DESIGN: DR. J.J.STIFFLER                                *
C*       SOFTWARE DESIGN AND IMPLEMENTATION: LYNN A. BRYANT            *
C*       DATE: MARCH 23,1981                                           *
C*       DATE EDITED: 5-03-81                                          *
C*       EDITED BY: LYNN A. BRYANT                                     *
C*                                                                     *
C*       PURPOSE:  CARE III IS A GENERAL-PURPOSE RELIABILITY           *
C*                 ESTIMATION TOOL FOR FAULT-TOLERANT AVIONICS         *
C*                 SYSTEMS.                                            *
C*                                                                     *
C***********************************************************************
C
```

EXTERNALS	TYPE	ARGS
BUFBLK		6
EOF	REAL	1
FNCK	REAL	2
RLSBRN		0

```
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
C  NOTE:  THIS BLANK COMMON AREA IS ALSO USED TO HOLD THE LOADER,
C         IN ORDER TO KEEP THE LOAD FIELD LENGTH REQUIREMENT THE
C         SAME SIZE AS THE EXECUTION FIELD LENGTH.
C
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      DIMENSION REC1(4),REC2(6)
      EQUIVALENCE (ITSTPS,REC1(1)),(NSTGRN,REC2(1))
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      DIMENSION REC3(163)
      EQUIVALENCE (NCONVEC(1),REC3(1))
C
      COMMON/RATES/ OMGA(5,20),RLAM(5,20)
      DIMENSION REC4(200)
      EQUIVALENCE (OMGA(1,1),REC4(1))
C
      COMMON/CNFGVAR/ LFLTVEC(20),QLTSUM(65),PSTSUM(65),QPSTSM(65),
     1      QSUMWT(14),QNOEFCT,PNOEFCT,PSTCOM,STOPARM,SUMK(3),DBGFLG
C  'PLTREL' WILL ENCOMPASS THE THREE ARRAYS PASSED TO PROGRAM RELPLT
C  FOR RELIABILITY PLOTS.
      DIMENSION PLTREL(195)
      EQUIVALENCE (QLTSUM(1),PLTREL(1))
C
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      DIMENSION SRPSTF(65,20)
C  AFTER EACH SUBRUN, 'SRPSTF' IS COMPUTED AND BUFFERED OUT TO TAPE17.
C  'AXIAR' IS NO LONGER NEEDED AT THAT POINT, UNTIL THE NEXT SUBRUN -
C  WHERE IT IS CREATED USING THAT SUBRUN DATA.
      EQUIVALENCE (AXIAR(1,1,1),SRPSTF(1,1))
C
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
C  FOR OPTIMIZATION PURPOSES DURING THE BUFFER IN OF /CVRGCOM/,
C  DEFINE AND EQUIVALENCE 'CVRGAR' TO ENCOMPASS THE ENTIRE COMMON AREA.
      DIMENSION CVRGAR(9805)
      EQUIVALENCE (CMST0(1,1,1),CVRGAR(1))
C
      DIMENSION PSTFAR(65,70),MNTRMV(70),PSTFPD(65),PSTSMW(14),TITLE(80)
      DIMENSION NSTGAR(35)
C  COVERAGE ARRAYS ARE NO LONGER NEEDED WHEN 'PSTFAR' IS BUFFERED IN
C  FROM TAPE17.
      EQUIVALENCE (CMST0(1,1,1),PSTFAR(1,1))
      DIMENSION PSTBUF(4550)
      EQUIVALENCE (PSTFAR(1,1),PSTBUF(1))
C
      LOGICAL TRANS,TRNSFC,DBGFLG,CHIDBG,RLPLOT
```

```
      PROGRAM RELPLT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PLFILE,
     1      TAPE4=PLFILE,RELIN=0,TAPE2=RELIN,PLTFL=0,TAPE16=PLTFL)
C
C***********************************************************************
C*                                                                     *
C*        TITLE:  COMPUTER-AIDED RELIABILITY ESTIMATION, VERSION THREE *
C*        ACRONYM: CARE III                                            *
C*        MODEL DESIGN: DR. J.J.STIFFLER                               *
C*        SOFTWARE DESIGN AND IMPLEMENTATION: LYNN A. BRYANT           *
C*        DATE: MARCH 23,1981                                          *
C*        DATE EDITED: 4-30-81                                         *
C*        EDITED BY: LYNN A. BRYANT                                    *
C*                                                                     *
C*        PURPOSE:  CARE III IS A GENERAL-PURPOSE RELIABILITY          *
C*                  ESTIMATION TOOL FOR FAULT-TOLERANT AVIONICS        *
C*                  SYSTEMS.                                           *
C*                                                                     *
C***********************************************************************
C
C  THIS PROGRAM PLOTS THE TOTAL SYSTEM UNRELIABILITY - 'QLTSUM'; THE
C  TOTAL PERFECT COVERAGE UNRELIABILITY - 'PSTSUM'; AND THE SUM OF
C  THE TWO - 'QPSTSM', USING FILE PLTFL GENERATED IN PROGRAM CARE3.
C  NOTE - TAPE4=PLFILE IS USED BY THE DISSPLA ROUTINES.
C
      COMMON/PLTCOM/ QLTSUM(65),PSTSUM(65),QPSTSM(65)
      DIMENSION PLTREL(195)
      EQUIVALENCE (QLTSUM(1),PLTREL(1))
C
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      DIMENSION REC1(4),REC2(6)
      EQUIVALENCE (ITSTPS,REC1(1)),(NSTGRN,REC2(1))
C
```

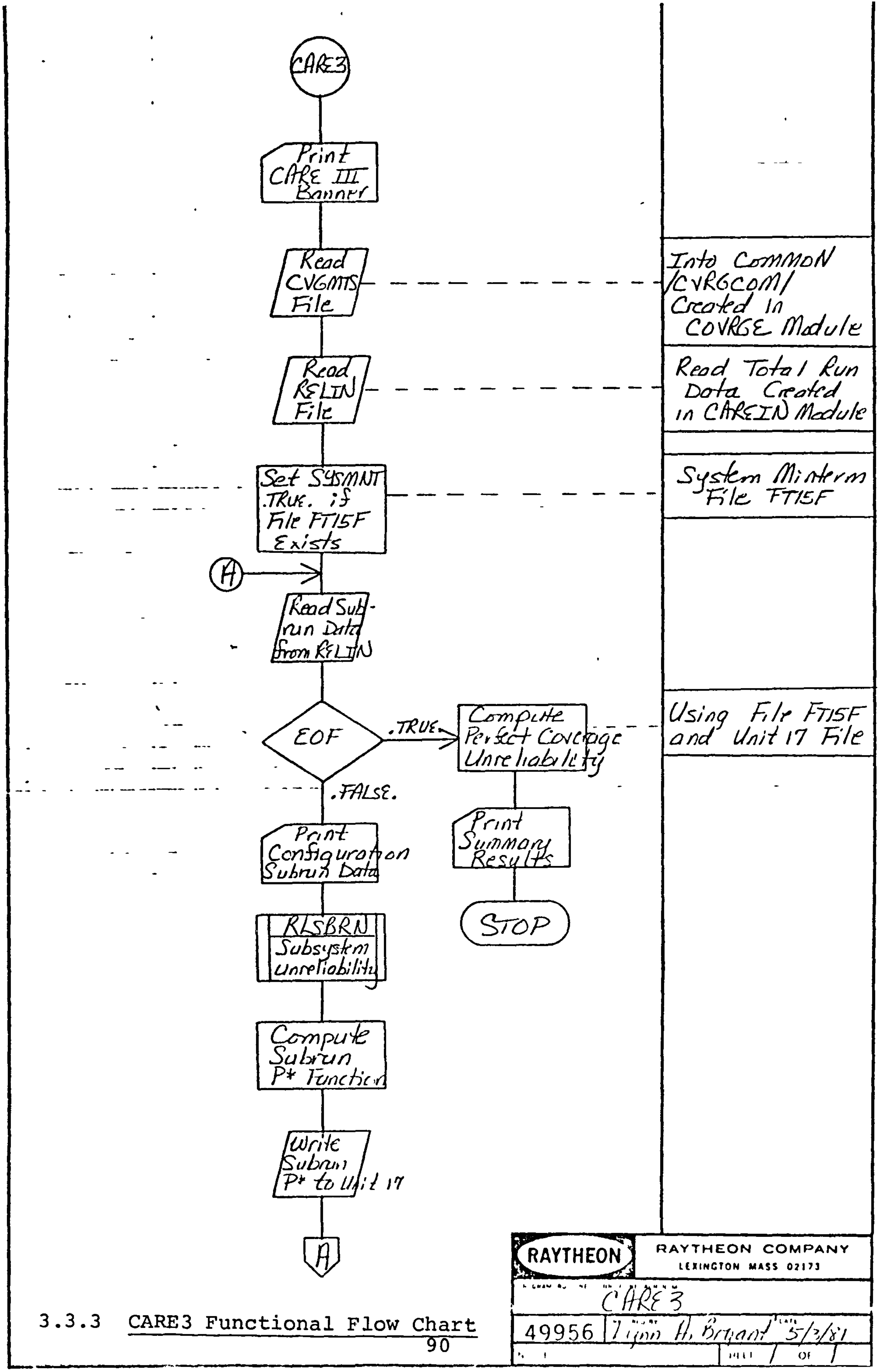
EXTERNALS	TYPE	ARGS
BUFBLK		6
CPLOT		8
DONEPL		0

3.3.2 CARE3 and RELPLT Files

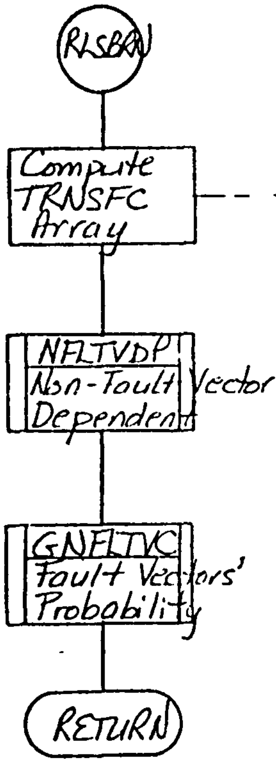
	File Name	Unit Name	Description
CARE3 Files:			
1.	RELIN=Ø,	TAPE4 = RELIN,	system definition input file generated by CAREIN program.
2.	INPUT,	TAPE5 = INPUT	
3.	OUTPUT,	TAPE6 = OUTPUT	
4.	CVGMTS=Ø,	TAPE7 = CVGMTS,	binary coverage moments file from COVRGE program.
5.	DBUG,	TAPE8 = DBUG	
6.		TAPE9 = Ø,	HLAT internal binary file.
7.		TAPE1Ø = Ø,	GFLD internal binary file.
8.		TAPE11 = Ø,	GNBNG internal binary file.
9.		TAPE12 = Ø,	HDF internal binary file.
10.	GHFNCS,	TAPE13 = GHFNCS,	debug file, under variable CHIDBG control, contains G and H functions computed using CVGMTS file.
11.	BXYFL=Ø,	TAPE14 = BXYFL,	binary critical pair results file created in program CAREIN.
12.	FT15F,	TAPE15 = FT15F,	system minterm file from CAREIN program.
13.†	PLTFL=Ø,	TAPE16 = PLTFL,	plot file of Q SUM, P* SUM, and Q+P* SUM passed to program RELPLT.
14.		TAPE17 = Ø,	function used to compute perfect coverage function (computed per subrun).

†Must be saved and passed to program RELPLT.

File Name	Unit Name	Description
RELPLT Files:		
1. RELIN=Ø,	TAPE2 = RELIN,	system definition input file.
2. PLFILE,	TAPE4 = PLFILE,	DISSPLA plot file.
3. INPUT,	TAPE5 = INPUT	
4. OUTPUT,	TAPE6 = OUTPUT	
5. PLTFL=Ø,	TAPE16 = PLTFL,	plot file of Q SUM, P* SUM, and Q+P* SUM.



3.3.3 CARE3 Functional Flow Chart



Specifies Transient Fault Categories

RAYTHEON

RAYTHEON COMPANY
LEXINGTON MASS 02173

RLSBRN

49956 Lynn A. Bryant 6/3/71

SHEET 1 OF 1

3.3.4 CARE3 Subroutines

```
      SUBROUTINE ABCST(MCHI,IT,ISTG,ICAT,JSTG,JCAT,N,FUNC,
     1                 ACOEF,BCOEF,CCOEF)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      LOGICAL TRNSFC,CHIDBG,SYSMNT,RLPLOT
C
C  THIS SUBROUTINE COMPUTES THE A,B AND C COEFFICIENTS THAT ARE
C  MULTIPLIED BY THE CMST FUNCTION ARRAYS, GOTTEN FROM THE COVERAGE
C  MODEL, TO YIELD THE FHSFST AND FHDFST FUNCTIONS.
C
```

```
EXTERNALS            TYPE    ARGS
       FUNC        REAL        6    F.P.

INLINE FUNCTIONS     TYPE    ARGS
       MOD         INTEGER     2   INTRIN
```

```
      SUBROUTINE ARZERO(AR,N,ALLO)
C
C  THIS SUBROUTINE TESTS 'AR' FOR A NON-ZERO VALUE.  IF ALL VALUES IN
C  THE ARRAY EQUAL ZERO, 'ALLO' IS SET .TRUE.
C
      LOGICAL ALLO
      DIMENSION AR(1)
```

```
      SUBROUTINE BUFFIN(MCHI,IT,ITB,IUNIT)
C
C  THIS SUBROUTINE FILLS EITHER GORHSF(G OR H SINGLE FAULT FUNCTION
C  FOR 'ITBLKSZ' NUMBER OF TIME STEP RECORDS) OR HDFPTS(H DOUBLE FAULT
C  FUNCTION PER TIME STEP) ARRAY DEPENDING UPON MCHI.  IUNIT SPECIFIES
C  WHICH LOCAL FILE TO READ FROM.  EACH BLOCK OF RECORDS ON 'TAPE
C  (IUNIT)' WAS CREATED FOR 'ITBLKSZ' NUMBER OF TIME STEPS
C  FOR ALL FAULT CATEGORIES (OVER ALL STAGES).  THE DOUBLE FAULT FILE
C  CONTAINS ONLY ONE TIME STEP RECORD PER BLOCK.
C
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC,CHIDBG,SYSMNT
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                 PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
C
```

```
 EXTERNALS          TYPE   ARGS
       UNIT       REAL       1
```

```
      SUBROUTINE BUFFOUT(MCHI,ITB,IUNIT)
C
C  THIS SUBROUTINE CREATES A BLOCK OF RECORDS ON LOCAL FILE
C  'TAPE(IUNIT)' USING BUFFER OUT, FROM EITHER GORHSF OR HDFPTS ARRAY
C  DEPENDING UPON MCHI.  EACH BLOCK OF RECORDS CONTAINS 'ITBLKSE'
C  NUMBER OF TIME STEPS FOR ALL FAULT CATEGORIES (OVER ALL STAGES)
C  FROM GORHSF ARRAY.  WHEN CREATING THE DOUBLE FAULT FILE USING
C  HDFPTS ARRAY, EACH BLOCK CONTAINS JUST ONE TIME STEP RECORD.
C
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC,CHIDBG,SYSMNT,RLPLOT
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
C
```

EXTERNALS	TYPE	ARGS
UNIT	REAL	1

```
      SUBROUTINE BXYC(ISTG,JSTG,INBXY,LXC,LYC)
C
C   FOR A 20 STAGE SUBRUN, 210 UNIQUE BXY FUNCTION DEFINITIONS ARE
C   POSSIBLE, BUT ONLY 55 DEFINITIONS CAN BE IN MEMORY AT ONE TIME.
C   'IBREC' SPECIFIES WHICH RECORD IS CURRENTLY IN MEMORY.  (A MAXIMUM
C   OF FOUR RECORDS CAN EXIST PER 20 STAGE BLOCK OF DEFINITIONS.)
C
C   INDEX INTO 'BXYAR': BXYAR(LX-MUX+1,LY-MUY+1,INBXY), WHERE LX AND LY
C   ARE THE CURRENT NUMBER OF FAULTS IN 'ISTG' AND 'JSTG' RESPECTIVELY
C   AND MUX=0..LX AND MUY=0..LY.
C
C   THIS SUBROUTINE COMPUTES 'INBXY' GIVEN THE 'ISTG,JSTG' PAIR.
C   'IJSTGIN' CONTAINS THE INDEX 'INBXY' BASED ON THE CONVERSION OF
C   'ISTG,JSTG' INTO 'IJSTG'.  NOTE: THE BXY FUNCTION IS THE SAME FOR
C   THE PAIRS 'ISTG,JSTG' AND 'JSTG,ISTG', I.E. STAGE PAIR ORDER IS NOT
C   IMPORTANT.
C   IF 'INBXY' EQUALS ZERO, NO BXY FUNCTION EXISTS FOR THAT STAGE PAIR.
C   IF 'INBXY' IS GREATER THAN 'IBREC'*'MXBDEF', BUFFER IN THE NEXT
C   RECORD, WHICH CONTAINS THE NEXT 55 DEFINITIONS OF THE BXY FUNCTIONS.
C   IF 'INBXY' IS .LE. TO 'IBREC'*'MXBDEF', BACKSPACE TWO RECORDS AND
C    BUFFER IN THE DATA.  THIS IS POSSIBLE BECAUSE BXY FUNCTIONS ARE
C   USED IN BOTH ROUTINES 'FBCRTL' AND 'FDSCRTL' PER FAULT VECTOR.
C   NOTE: IN MOST CASES, ONLY ONE RECORD IS REQUIRED TO DEFINE THE NON-
C   ZERO BXY FUNCTIONS FOR 'NSTGS' COUPLED STAGES SUBRUN.  I.E. THERE
C   WILL ONLY EXIST 'MXBDEF' OR LESS NON-ZERO BXY FUNCTIONS.  THEREFORE
C   ONLY THE INITIAL BUFFER IN WILL BE REQUIRED PER SUBRUN, WHICH IS DONE
C   IN SUBROUTINE NFLTVDP.
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC,FATAL,EOFFLG
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
C   FOR OPTIMIZATION PURPOSES DURING THE BUFFER IN OF /BXYCOM/, DEFINE
C   AND EQUIVALENCE 'BCOMAR' TO ENCOMPASS THE ENTIRE DATA TO BE BUFFERED.
      DIMENSION BCOMAR(5802)
      EQUIVALENCE (BXYAR(1,1,1),BCOMAR(1))
```

```
 EXTERNALS             TYPE    ARGS
        BUFBLK                   6

 INLINE FUNCTIONS     TYPE    ARGS
        MOD        INTEGER      2   INTRIN
```

```
      SUBROUTINE CAXLAT
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON/RATES/ OMGA(5,20),RLAM(5,20)
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,NWT,PSTRNC,
     1               PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      LOGICAL TRNSFC,CHIDBG,SYSMNT,RLPLOT
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      DIMENSION HLAT(5,20,65)
      EQUIVALENCE (HLAT(1,1,1),GORHSF(1,1,1))
C
C  THIS SUBROUTINE COMPUTES ARRAY AXIAR(ICAT,ISTG,IT) WHICH IS THE
C  PROBABILITY THAT A GIVEN STAGE X MODULE (ISTG) HAS A CATEGORY
C  XI (ICAT) LATENT FAULT AT TIME 'T', REPRESENTED BY 'IT',
C  GIVEN THAT IT HAS EXPERIENCED SOME FAULT BY TIME 'T'.
C
```

$$a_{x_i}(t) = \begin{cases} \dfrac{H_L(t|x_i)}{1-r_x(t)} & \text{PERMANENT} \\ H_L(t|x_i) & \text{TRANSIENT} \end{cases}$$

AXIAR(ICAT, ISTG, IT) using FHSFST(MLAT, IT, ISTG, ICAT) and RXAR(ISTG, IT)

EXTERNALS	TYPE	ARGS
BUFFOUT		3
FHSFST	REAL	4
PRNTGH		3

INLINE FUNCTIONS	TYPE	ARGS	
MOD	INTEGER	2	INTRIN

```
      SUBROUTINE CRXFF
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,NWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      LOGICAL TRNSFC,CHIDBG,TRNSFLT,SYSMNT,RLPLOT
C
C  THIS SUBROUTINE COMPUTES THE PROBABILITY THAT A SINGLE STAGE X MODULE
C  IS FAULT-FREE AT TIME T, REPRESENTED BY IT, FOR ALL STAGES.
C
```

RELIABILITY OF A STAGE x MODULE

$$r_x(t) = \prod_i r_{x_i}(t)$$

$$RXAR(IT) = \prod_{ICAT} FRXIFF(IT, ISTG, ICAT, TRNSFLT)$$

```
EXTERNALS          TYPE   ARGS
      FRXIFF       REAL     4
      PRNTGH                3
```

```
      SUBROUTINE GNFLTVC(ISTINIT,ISTSTOP,MAXLAST)
C
C  THIS SUBROUTINE GENERATES FAULT VECTORS IN SETS STARTING AT
C  SET ISTINIT AND STOPPING AT SET ISTSTOP.  IF ISTINIT.EQ.1,
C  THE UNRELIABILITY OF THE SYSTEM IS BEING COMPUTED.  IF ISTINIT
C  .GT.1, PERFECT COVERAGE PROBABILITIES WILL BE COMPUTED.
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CNFGVAR/ LFLTVEC(20),QLTSUM(65),PSTSUM(65),QPSTSM(65),
     1        QSUMWT(14),QNOEFCT,PNOEFCT,PSTCOM,STOPARM,SUMK(3),DBGFLG
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/GNVCCOM/ QLT(65),PSTLAR(65),PLT(65),QINTGRL(3),
     1                ISETVC(20),ISETVCS(20),LVLAST(20),QSUMSF,PSUMSF
      LOGICAL TRNSFC,DBGFLG,COMPQ,COMPPST,RLPLOT
      LOGICAL QNOEFCT,PNOEFCT,PSTCOM,ALLO,SYSMNT
```

```
       EXTERNALS           TYPE   ARGS
              ARZERO                3
              FPSTAR    REAL        2
              UNRELQ                2

       INLINE FUNCTIONS    TYPE   ARGS
              AMAX1     REAL        0   INTRIN
              MINO      INTEGER     0   INTRIN
```

```
      SUBROUTINE NFLTVDP
C
C  THIS SUBROUTINE AND SUBROUTINES CALLED WITHIN COMPUTE ALL FUNCTIONS
C  THAT ARE NOT DEPENDENT UPON THE FAULT VECTORS.
C  FUNCTIONS THAT ARE USED CONTINUALLY THROUGHOUT THE ROUTINES ARE STORED
C  IN COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65).
C  IN ORDER TO SAVE MEMORY, CERTAIN G AND H FUNCTIONS, NAMELY HLAT(LATENT
C  FAULT), GFLD(FAILED), GNBNG(NOT BENIGN) AND HDF(DOUBLE FAULT), ARE
C  COMPUTED IN THIS SUBROUTINE AND BUFFERED OUT TO A LOCAL FILE WHICH
C  WILL BE BUFFERED IN WHEN REQUIRED.  LOCAL FILES TAPE9,TAPE10,TAPE11,
C  TAPE12 CONTAIN THESE G AND H FUNCTIONS.
C  TAPE13 NAMED GHFNCS IS A DEBUG PRINTOUT OF ALL G AND H FUNCTIONS
C  COMPUTED - CREATED IN SUBROUTINE PRNTGH IF CHIDBG IS .TRUE.
C
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC,CHIDBG,SYSMNT,NONFTL,EOFFLG,RLPLOT
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      DIMENSION HBNG(5,20,65)
      EQUIVALENCE (HBNG(1,1,1),GORHSF(1,1,1))
C  FOR OPTIMIZATION PURPOSES DURING THE BUFFER IN OF /BXYCOM/, DEFINE
C  AND EQUIVALENCE 'BCOMAR' TO ENCOMPASS THE ENTIRE DATA TO BE BUFFERED.
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
      DIMENSION BCOMAR(5802)
      EQUIVALENCE (BXYAR(1,1,1),BCOMAR(1))



      EXTERNALS            TYPE    ARGS
             BUFBLK                  6
             BUFFOUT                 3
             CAXLAT                  0
             CRXFF                   0
             FGST        REAL        4
             FHDFST      REAL        6
             FHSFST      REAL        4
             FNCK        REAL        2
             PRNTGH                  3

      INLINE FUNCTIONS     TYPE    ARGS
             MOD         INTEGER     2   INTRIN
```

```
      SUBROUTINE PRNTGH(MCHI,IUNIT,GORH)
C
C  THIS SUBROUTINE IS USED TO PRINT G AND H FUNCTIONS CURRENTLY STORED
C  IN COMMON//. MCHI DETERMINES WHETHER THE ARRAY TO BE PRINTED
C  IS A SINGLE OR DOUBLE FAULT ARRAY.  WHEN MCHI.EQ.MDF, IUNIT SPECIFIES
C  WHICH FILE CONTAINS THE DOUBLE FAULT FUNCTION.  SINCE THE DOUBLE
C  FAULT FUNCTION IS TOO LARGE TO STORE IN CENTRAL MEMORY, IT MUST BE
C  BUFFERED IN FROM DISK IN ORDER TO BE PRINTED.
C
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC,CHIDBG,SYSMNT,RLPLOT
      DIMENSION GHSFPTS(5,20)
      EQUIVALENCE (GHSFPTS(1,1),GORHSF(1,1,1))
      DIMENSION MCHICD(6)
      DATA MCHICD/4HDP  ,4HLAT ,4HFLD ,4HBNG ,4HNBNG,4HDF  /
```

```
  EXTERNALS          TYPE   ARGS
       BUFFIN                4
```

```
      SUBROUTINE RLSBRN
C
C  THIS SUBROUTINE COMPUTES THE SYSTEM UNRELIABILITY PER FAULT VECTOR
C  FOR A SUBSET OF THE TOTAL NUMBER OF STAGES.  A MAXIMUM OF 20 STAGES
C  PER SUBRUN ARE USED.
C
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/RATES/ OMGA(5,20),RLAM(5,20)
      COMMON/CNFGVAR/ LFLTVEC(20),QLTSUM(65),PSTSUM(65),QPSTSM(65),
     1        QSUMWT(14),QNOEFCT,PNOEFCT,PSTCOM,STOPARM,SUMK(3),DBGFLG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      LOGICAL TRANS,TRNSFC,DBGFLG,RLPLOT
      LOGICAL QNOEFCT,PNOEFCT,PSTCOM,SYSMNT
```

EXTERNALS		TYPE	ARGS	
	GNFLTVC		3	
	NFLTVDP		0	

INLINE	FUNCTIONS	TYPE	ARGS	
	MINO	INTEGER	0	INTRIN

```
      SUBROUTINE SUMMAT(IT,NFLTS)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CNFGVAR/ LFLTVEC(20),QLTSUM(65),PSTSUM(65),QPSTSM(65),
     1        QSUMWT(14),QNOEFCT,PNOEFCT,PSTCOM,STOPARM,SUMK(3),DBGFLG
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/GNVCCOM/ QLT(65),PSTLAR(65),PLT(65),QINTGRL(3),
     1                ISETVC(20),ISETVCS(20),LVLAST(20),QSUMSF,PSUMSF
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
      LOGICAL TRNSFC,DBGFLG,CHIDBG,QNOEFCT,PNOEFCT,PSTCOM,SYSMNT,RLPLOT
      DIMENSION LMUNTVC(20)
C
C  THIS SUBROUTINE COMPUTES THE SUMMATION, OVER ALL FAULT CATEGORIES,
C  OF ALL THE FUNCTIONS INVOLVED IN COMPUTING THE SUBSYSTEM RELIABILITY
C  MODEL.  THIS SUMMATION IS COMPUTED OVER ALL TIME STEPS FOR THE
C  CURRENT LFLTVEC AND STORED IN SUMK ARRAY.  SUMK IS INTEGRATED USING
C  THE SIMPSON'S 1/3 RULE IN ORDER TO GENERATE THE QLT ARRAY FOR THE
C  CURRENT LFLTVEC PER TIME STEP.
C
```

$$K_{\underline{\ell}}(\tau) = \sum_{y_j}\left[C_{y_j}(\tau|\underline{\ell}-\varepsilon_y)P^*_{\underline{\ell}-\varepsilon_y}(\tau)(n_y-\ell_y+1)\lambda_{y_j}(\tau)\right]$$

$$+ \quad A'(\tau|\underline{\ell})P^*_{\underline{\ell}}(\tau) + a'(\tau|\underline{\ell})P^*_{\underline{\ell}}(\tau)$$

```
       EXTERNALS            TYPE    ARGS
              FAPC       REAL        2
              FCYJ       REAL        4
              FLAM       REAL        3
              FPSTAR     REAL        2
              FPSTREC    REAL        4

       INLINE FUNCTIONS      TYPE    ARGS
              MOD        INTEGER     2   INTRIN
```

```
      SUBROUTINE UNRELQ(NFLTS,NFLSTG)
      COMMON/GNVCCOM/ QLT(65),PSTLAR(65),PLT(65),QINTGRL(3),
     1                ISETVC(20),ISETVCS(20),LVLAST(20),QSUMSF,PSUMSF
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CNFGVAR/ LFLTVEC(20),QLTSUM(65),PSTSUM(65),QPSTSM(65),
     1         QSUMWT(14),QNOEFCT,PNOEFCT,PSTCOM,STOPARM,SUMK(3),DBGFLG
      LOGICAL QNOEFCT,PNOEFCT,PSTCOM,DBGFLG,SYSMNT,RLPLOT
C
C  THIS SUBROUTINE COMPUTES THE SYSTEM UNRELIABILITY OVER ALL TIME
C  STEPS PER FAULT VECTOR AS THEY ARE GENERATED IN SUBROUTINE
C  GNFLTVC.
C
```

MODIFIED RELIABILITY EQUATION

$$Q_{\underline{\ell}}(t) = \int_0^t K_{\underline{\ell}}(\tau)\,d\tau$$

Variables	Components of Equation
QLT(65)	$Q_{\underline{\ell}}(t)$
SUMK(3)	$K_{\underline{\ell}}(\tau)$

```
 EXTERNALS           TYPE    ARGS
        FINTGRT    REAL        2
        SUMMAT                 2
```

3.3.5 CARE3 Functions

```
      FUNCTION FAC(IT,LVECTOR)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC,CHIDBG
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1              KFSTG,KPLSTG(70)
      DIMENSION LVECTOR(20)
C  THE THIRD DIMENSION IN THE FOLLOWING THREE ARRAYS MUST EQUAL
C  'ITBLKSZ'.
      DIMENSION HLAT(5,20,17),GFLD(5,20,17),GNBNG(5,20,17)
      EQUIVALENCE (HLAT(1,1,1),GORHSF(1,1,1)),(GFLD(1,1,1),GORHSF(1,1,
     1  18)),(GNBNG(1,1,1),GORHSF(1,1,35))
C
C  USING MDF(DOUBLE FAULT DID CAUSE FAILURE), THIS FUNCTION
C  COMPUTES THE RATE AT WHICH THE SYSTEM FAILS BY TIME T DUE
C  A CRITICAL FAULT CONDITION.  NOTE: THIS FUNCTION IS NOT CALLED
C  UNLESS LVECTOR CONTAINS AT LEAST 2 FAULTS.
C
C  FAC = 0.0 AT TIME 0.
      FAC = 0.0
C  IF 'IBREC' EQUALS 0 OR THIS SUBRUN'S FIRST STAGE DOES NOT EQUAL THE
C  CURRENT COUPLED PAIR SUBSET'S FIRST STAGE, NO CRITICAL PAIR FAULTS
C  EXIST AND THIS FUNCTION EQUALS ZERO.
```

RATE WHICH SYSTEMS HAVING $\underline{\ell}$
FAULTS FAIL AT TIME t DUE TO
CRITICAL FAULT CONDITIONS

$$A'(t|\underline{\ell}) = \sum_{x_i,y_j} \frac{h_{DF}(t|x_i,y_j)}{H_L(t|x_i)H_L(t|y_j)} B_{x_i,y_j}(t|\underline{\ell})$$

```
    EXTERNALS             TYPE    ARGS
          BUFFIN                   4
          FBCRTL       REAL        6

    INLINE FUNCTIONS      TYPE    ARGS
          MOD          INTEGER     2   INTRIN
```

```
      FUNCTION FAPC(IT,LVECTOR)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC,CHIDBG
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      DIMENSION LVECTOR(20)
C  THE THIRD DIMENSION IN THE FOLLOWING THREE ARRAYS MUST EQUAL
C  'ITBLKSZ'.
      DIMENSION HLAT(5,20,17),GFLD(5,20,17),GNBNG(5,20,17)
      EQUIVALENCE (HLAT(1,1,1),GORHSF(1,1,1)),(GFLD(1,1,1),GORHSF(1,1,
     1  18)),(GNBNG(1,1,1),GORHSF(1,1,35))
C
C  THIS FUNCTION COMPUTES THE RATE AT WHICH A SYSTEM HAVING LVECTOR
C  FAULTS FAILS AT TIME T DUE TO A NONCRITICAL FAULT.
C
```

RATE AT WHICH SYSTEMS HAVING
$\underline{\ell}$ FAULTS FAIL AT TIME t DUE TO
ERROR PROPAGATION

$$a'(t|\underline{\ell}) = \underbrace{\sum_{x_i} \frac{\ell_x h_F(t|x_i)}{1-r_x(t)}}_{\text{PERMANENT}} + \underbrace{\sum_{x_i} (n_x-\ell_x) h_F(t|x_i)}_{\text{TRANSIENT}}$$

```
INLINE FUNCTIONS   TYPE    ARGS
      MOD        INTEGER    2  INTRIN
```

```
      FUNCTION FBCRTL(IT,ISTG,ICAT,JSTG,JCAT,LVECTOR)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC,TRNSXI,TRNSYJ
      DIMENSION LVECTOR(20)
      DATA MAXFLT/10/
C
C  THIS FUNCTION COMPUTES THE PROBABILITY THAT A SYSTEM CONTAINING
C  LVECTOR FAULTS WOULD ENTER AN XI,YJ CRITICAL STATE, REPRESENTED
C  BY 'ISTG,ICAT;JSTG,JCAT', WERE A CATEGORY YJ FAULT TO OCCUR AT
C  TIME T, REPRESENTED BY 'IT'.
C
```

EXPECTED NUMBER OF x_i, y_j-
CRITICAL FAULTS AT TIME t
GIVEN $\underline{\ell}$ PERMANENT FAULTS

$$B_{x_i,y_j}(t|\underline{\ell}) = \sum_{\mu_x,\mu_y} b_{x,y}(\ell_x-\mu_x,\ell_y-\mu_y)P(\mu_x,t|\ell_x) P(\mu_y,t|\ell_y)C(x_i,y_j)a_{x_i}(t)a_{y_j}(t)$$

$b_{x,y}$ stored in BXYAR(10,10,55)

```
EXTERNALS         TYPE   ARGS
      BXYC                 5
      FPMUX      REAL      4
```

See next page for $C(x_i,y_j)$ definition.

FUNCTION	MATHEMATICAL EXPRESSION	
$C(x_i, y_j)$	$\frac{\mu_x \mu_y}{a_x(t) a_y(t)}$	x_i, y_j = PERMANENT $x \neq y$
	$\frac{\mu_x(\mu_x - 1)}{a_x^2(t)}$	x_i, y_j = PERMANENT $x = y$
	$\frac{\mu_x(n_y - \ell_y)}{a_x(t)}$	x_i = PERMANENT y_j = TRANSIENT
	$\frac{(n_x - \ell_x)\mu_y}{a_y(t)}$	x_i = TRANSIENT y_j = PERMANENT
	$(n_x - \ell_x)(\mu_y - \ell_y)$	x_i, y_j = TRANSIENT $x \neq y$
	$(n_x - \ell_x)(n_x - \ell_x - 1)$	x_i, y_j = TRANSIENT $x = y$

```
      FUNCTION FCLAM(IT,ITAU,ISTG,ICAT)
C
      COMMON/RATES/ OMGA(5,20),RLAM(5,20)
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      LOGICAL SYSMNT,RLPLOT
C
```

$$\Lambda_{x_i}(t) = \lambda_{x_i} t^{\omega_{x_i}}$$

Variables	Components of Equation
RLAM(ICAT,ISTG)	λ_{x_i}
OMGA(ICAT,ISTG)	ω_{x_i}

```
      FUNCTION FCYJ(IT,JSTG,JCAT,LVECTOR)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDF,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      LOGICAL TRNSFC,CHIDBG
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
      DIMENSION LVECTOR(20)
C  THE THIRD DIMENTION IN THE FOLLOWING THREE ARRAYS MUST EQUAL
C  'ITBLKSZ'.
      DIMENSION HLAT(5,20,17),GFLD(5,20,17),GNBNG(5,20,17)
      EQUIVALENCE (HLAT(1,1,1),GORHSF(1,1,1)),(GFLD(1,1,1),GORHSF(1,1,
     1  18)),(GNBNG(1,1,1),GORHSF(1,1,35))
C
```

PROBABILITY THAT A CATEGORY y_j
FAULT WOULD PRODUCE A SYSTEM
FAILURE AT TIME t GIVEN $\underline{\ell}$
FAULTS AT TIME t^-

$$C_{y_j}(t|\underline{\ell}) = \sum_{x_i} \frac{H_{\bar{B}}(t|x_i)}{H_L(t|x_i)} D_{x_i,y_j}(t|\underline{\ell})$$

```
EXTERNALS              TYPE    ARGS
       FDSCRTL      REAL        6

INLINE FUNCTIONS       TYPE    ARGS
       MOD          INTEGER     2   INTRIN
```

```
      FUNCTION FDSCRTL(IT,ISTG,ICAT,JSTG,JCAT,LVECTOR)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON/BXYCOM/ BXYAR(10,10,55),IJSTGIN(210),LCNDVEC(20),IBREC,
     1               KFSTG,KPLSTG(70)
      DIMENSION LVECTOR(20)
      DATA MAXFLT/10/
C
C  THIS FUNCTION COMPUTES THE PROBABILITY THAT A SYSTEM CONTAINING
C  LVECTOR FAULTS WOULD BE IN A SUPERCRITICAL STATE WERE A CATEGORY
C  YJ FAULT, REPRESENTED BY 'JSTG,JCAT', TO OCCUR AT TIME T,
C  REPRESENTED BY 'IT'.
C
```

EXPECTED NUMBER OF $x_i y$-CRITICAL FAULTS, GIVEN $\underline{\ell}$ PERMANENT FAULTS, THAT WOULD BE CREATED AS THE RESULT OF A STAGE y FAULT AT TIME t

$$D_{x_i,y}(t|\underline{\ell}) = \sum_{\mu_x,\mu_y} b_{x,y}(\ell_x-\mu_x,\ell_y-\mu_y)P(\mu_x,t|\ell_x) P(\mu_y,t|\ell_y) \begin{cases} \mu_x \dfrac{a_{x_i}(t)}{a_x(t)} & x_i = \text{PERMANENT} \\ (n_x-\ell_x)a_{x_i}(t) & x_i = \text{TRANSIENT} \end{cases}$$

EXTERNALS	TYPE	ARGS
BXYC		5
FPMUX	REAL	4

```
      FUNCTION FFDFST(MCHI,KINDX,ISTG,ICAT,JSTG,JCAT)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC,CHIDBG
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      DIMENSION HBNG(5,20,65)
      EQUIVALENCE (HBNG(1,1,1),GORHSF(1,1,1))
C
C  THIS FUNCTION IN COMBINATION WITH SUBROUTINE ABCST REPRESENTS
C  SLOW VARYING RELIABILITY FUNCTIONS BY A THIRD-ORDER POLYNOMIAL OVER
C  THE RANGE OF INTEREST FOR DOUBLE FAULTS.
C
```

```
      EXTERNALS          TYPE   ARGS
             FFSFST    REAL       6
```

```
      FUNCTION FFSFST(MCHI,KINDX,ISTG,ICAT,JDUMS,JDUMC)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC,CHIDBG
C
C  THIS FUNCTION IN COMBINATION WITH SUBROUTINE ABCST REPRESENTS
C  SLOW VARYING RELIABILITY FUNCTIONS BY A THIRD-ORDER POLYNOMIAL OVER
C  THE RANGE OF INTEREST FOR SINGLE FAULTS.
C
```

```
         EXTERNALS          TYPE   ARGS
              FLAM       REAL        3
```

```
      FUNCTION FGST(MCHI,IT,ISTG,ICAT)
C
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      LOGICAL TRNSFC,CHIDBG
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
C
C  THIS FUNCTION COMPUTES THE PROBABILITY THAT A CATEGORY XI FAULT IS
C  IN STATE MCHI AT TIME T, REPRESENTED BY IT, GIVEN THAT IT WAS LATENT
C  PREVIOUSLY.
```

```
      FUNCTION FHDFST(MCHI,IT,ISTG,ICAT,JSTG,JCAT)
C
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      LOGICAL TRANS,TRNSFC,CHIDBG,SYSMNT,RLPLOT
      EXTERNAL FFDFST
C
C   THIS ROUTINE COMBINES A POLYNOMIAL REPRESENTATION OF RELIABILITY
C   FUNCTIONS(SLOW VARYING) AND MOMENTS OF COVERAGE PROBABILITY FUNCTIONS
C   (FAST VARYING) TO YIELD A CONVOLUTED INTEGRATION OF THE TWO TYPES OF
C   DOUBLE FAULT FUNCTIONS.
C
```

RATE AT WHICH AN $x_i y_j$-CRITICAL
FAULT CAUSES SYSTEM FAILURE

$$h_{DF}(t|x_i,y_j) = a_{DF}(t|x_i,y_j)m^0_{DF}(t|x_i,y_j) + b_{DF}(t|x_i,y_j)m^1_{DF}(t|x_i,y_j) + c_{DF}(t|x_i,y_j)m^2_{DF}(t|x_i,y_j)$$

```
 EXTERNALS          TYPE    ARGS
       ABCST                 11
       FFDFST                 0

 INLINE FUNCTIONS    TYPE    ARGS
       MOD        INTEGER     2   INTRIN
```

```
      FUNCTION FHSFST(MCHI,IT,ISTG,ICAT)
C
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      LOGICAL TRANS,TRNSFC,CHIDBG,SYSMNT,RLPLOT
      EXTERNAL FFSFST
      DATA JDUM/0/
C
C   THIS ROUTINE COMBINES A POLYNOMIAL REPRESENTATION OF RELIABILITY
C   FUNCTIONS(SLOW VARYING) AND MOMENTS OF COVERAGE PROBABILITY FUNCTIONS
C   (FAST VARYING) TO YIELD A CONVOLUTED INTEGRATION OF THE TWO TYPES OF
C   SINGLE FAULT FUNCTIONS.
C
C
```

```
       EXTERNALS            TYPE   ARGS
              ABCST                  11
              FFSFST                  0

       INLINE FUNCTIONS     TYPE   ARGS
              MOD        INTEGER      2  INTRIN
```

See the next page for the Single-fault H functions computed using this function.

DEFINITION

FUNCTION	MATHEMATICAL EXPRESSION

PROBABILITY THAT A CATEGORY x_i TRANSIENT FAULT IS DETECTED AS PERMANENT

$$H_{DPT}(t|x_i) = a_{DP}(t|x_i)m_{DP}^{0}(t|x_i)+b_{DP}(t|x_i)m_{DP}^{1}(t|x_i) + c_{DP}(t|x_i)m_{DP}^{2}(t|x_i)$$

PROBABILITY OF A LATENT CATEGORY x_i FAULT AT TIME t

$$H_L(t|x_i) = a_L(t|x_i)M_L^0(t|x_i)+b_L(t|x_i)M_L^1(t|x_i) + c_L(t|x_i)M_L^2(t|x_i)$$

RATE OF ERROR PROPAGATION FAILURE DUE TO A CATEGORY x_i FAULT

$$h_F(t|x_i) = a_F(t|x_i)m_F^0(t|x_i)+b_F(t|x_i)m_F^1(t|x_i) + c_F(t|x_i)m_F^2(t|x_i)$$

PROBABILITY OF A BENIGN LATENT FAULT AT TIME t

$$H_B(t|x_i) = a_B(t|x_i)M_B^0(t|x_i)+b_B(t|x_i)M_B^1(t|x_i) + c_B(t|x_i)M_B^2(t|x_i)$$

PROBABILITY OF A NON-BENIGN LATENT FAULT AT TIME t

$$H_{\overline{B}}(t|x_i) = a_{\overline{B}}(t|x_i)M_{\overline{B}}^0(t|x_i)+b_{\overline{B}}(t|x_i)M_{\overline{B}}^1(t|x_i) + c_{\overline{B}}(t|x_i)M_{\overline{B}}^2(t|x_i)$$

```
      FUNCTION FINTGRT(IT,ARTOINT)
C
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      DIMENSION ARTOINT(3)
      LOGICAL SYSMNT,RLPLOT
C
C  THIS FUNCTION USES THE 1/3 SIMPSON'S INTEGRATION METHOD ON THE
C  VALUES STORED IN ARTOINT WHEN 'IT'.GT.2.  'IT' REPRESENTS
C  WHICH TIME STEP IS BEING COMPUTED.
C  WHEN 'IT' EQUALS 2, THE TRAPEZOIDAL INTEGRATION METHOD IS USED.
C
```

```
      FUNCTION FLAM(IT,ISTG,ICAT)
C
      COMMON/RATES/ OMGA(5,20),RLAM(5,20)
      COMMON/STEPCOM/ ITSTPS,MAXSTP,RELSTEP,TBASE,NSTGRN,KWT,PSTRNC,
     1                PRCODE,RLPLOT,IAXSRL,SYSMNT,TBCF
      LOGICAL SYSMNT,RLPLOT
C
```

$$\lambda_{x_i}(t) = \omega_{x_i}\lambda_{x_i}t^{\omega_{x_i}-1} .$$

Variables	Components of Equation
RLAM(ICAT, ISTG)	λ_{x_i}
OMGA(ICAT, ISTG)	ω_{x_i}

```
      FUNCTION FNFACT(N)
C
C
C  THIS FUNCTION COMPUTES N FACTORIAL
C
```

```
      FUNCTION FPMUX(IT,ISTG,MUX,LX)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/NONLDEF/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC
C
C  THIS FUNCTION COMPUTES THE PROBABILITY THAT A SUBSYSTEM CONTAINS
C  MUX STAGE X LATENT FAULTS GIVEN THAT IT HAS LX FAULTY STAGE X
C  MODULES.
C
```

PROBABILITY THAT A SUBSYSTEM CONTAINS μ_x STAGE x LATENT PERMANENT FAULTS GIVEN THAT IT HAS ℓ_x STAGE x PERMANENT FAULTS

$$P(\mu_x, t \mid \ell_x) = \binom{\ell_x}{\mu_x} (1-a_x(t))^{\ell_x - \mu_x} \, a_x^{\mu_x}(t) .$$

```
     EXTERNALS          TYPE    ARGS
           FNCK         REAL      2
```

```
      FUNCTION FPSTAR(IT,LVECTOR)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC
C
C  THIS FUNCTION COMPUTES THE PROBABILITY THAT A SYSTEM HAS SUSTAINED
C  EXACTLY LVECTOR FAILURES BY TIME T, REPRESENTED BY IT.
C
      DIMENSION LVECTOR(20)
```

```
 EXTERNALS           TYPE   ARGS
        FNCK        REAL      2
```

```
      FUNCTION FPSTREC(IT,JSTG,LVECTOR,PSTPRV)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      LOGICAL TRNSFC
C
C  THIS FUNCTION COMPUTES PSTAR FOR LVECTOR RECURSIVELY - GIVEN
C  PSTAR FOR LVECTOR MINUS THE UNIT VECTOR REPRESENTED BY JSTG
C  PASSED AS PSTPRV.
C
      DIMENSION LVECTOR(20)
```

```
      FUNCTION FRXIFF(IT,ISTG,ICAT,TRNSFLT)
C
      COMMON/CONFIG/ NCONVEC(20),MSRVVEC(20),NFLTCAT(20),MFLTYPE(5,20),
     1  NSTGS,IFSTG,NTYPS,TRNSFC(5,20),MAXSTG,MAXCAT,MAXTYP,MXSBRN
      COMMON/CHICOM/ MDP,MLAT,MFLD,MBNG,MNBNG,MDF,CHIDBG
      COMMON/CVRGCOM/ CMST0(5,65,5),CMST1(5,65,5),CMST2(5,65,5),
     1                CMDF0(5,5,65),CMDF1(5,5,65),CMDF2(5,5,65),
     2                TZEROST(5,5),TODF(5,5),TRANS(5)
      COMMON/NONLDEP/ AXIAR(5,20,65),RXAR(20,65)
      COMMON// GORHSF(5,20,65),HDFPTS(100,100),ITBLKSZ
      LOGICAL TRANS,TRNSFC,CHIDBG,TRNSFLT
      DIMENSION HDPFC(5,20,3),HINT(5,20,3),HDP(3)
C  USE AXIAR AS WORKING STORAGE FOR THIS ROUTINE BECAUSE AXIAR IS NOT
C  COMPUTED UNTIL RXAR IS COMPLETELY DEFINED.
      EQUIVALENCE (AXIAR(1,1,1),HDPFC(1,1,1)),(AXIAR(1,1,4),HINT(1,1,1))
C
C  THIS FUNCTION COMPUTES THE PROBABILITY THAT A GIVEN STAGE X
C  MODULE HAS EITHER NOT EXPERIENCED, OR HAS RECOVERED FROM, ANY
C  CATEGORY XI FAULTS BY TIME T.
C
```

PROBABILITY THAT A GIVEN STAGE
x MODULE HAS NOT EXPERIENCED A
CATEGORY-x_i FAULT BY TIME t

$$r_{x_i}(t) \quad \begin{cases} e^{-\Lambda_{x_i}(t)} & \text{PERMANENT} \\ e^{-H_{DPT}(\tau | x_i) d\tau} & \text{TRANSIENT} \end{cases}$$

EXTERNALS	TYPE	ARGS
FCLAM	REAL	4
FHSFST	REAL	4
FINTGRT	REAL	2
PREEXP	REAL	1

3.4 General Purpose Routines

The following routines are used in at least two of the three program modules described previously. NOTE: Subroutine CAREPLT interfaces to NASA's Graphic Plotting Package, and subroutine CPLOT interfaces to the DISSPLA Plotting Package. Currently, programs CVGPLT and RELPLT call CPLOT. Slight modifications must be made to these programs in order to use CAREPLT.

3.4.1 General Purpose Subroutines

```
      SUBROUTINE BUFBLK(INOUT,IUNIT,BLOCK,ILST,FATAL,EOFFLG)
C
C  THIS SUBROUTINE BUFFERS IN A BLOCK OF DATA INTO 'BLOCK'(1)..'BLOCK'
C  (ILST), IF 'INOUT'.EQ.1, FROM UNIT 'IUNIT'.  IT BUFFERS OUT A BLOCK
C  OF DATA FROM 'BLOCK'(1)..'BLOCK'(ILST), IF 'INOUT'.EQ.2, TO UNIT
C  'IUNIT'.
C
      DIMENSION BLOCK(1)
      LOGICAL FATAL,EOFFLG
```

```
        EXTERNALS           TYPE   ARGS
               UNIT      REAL       1
```

```
      SUBROUTINE CAREPLT(ARPASD,XPASD,STEP,GENXPTS,NPTS,LNORLG)
C **********************************************************************
C  THIS ROUTINE WORKS WITH NASA'S GRAPHIC PLOTTING PACKAGE.
C **********************************************************************
      DIMENSION ARTOPLT(515),X(515),ARPASD(1),XPASD(1)
C*****NOTE THAT TWO ADDITIONAL ELEMENTS ARE ASSIGNED AS
C*****REQUIRED BY ASCALE,LINPLT,SCALOG
      LOGICAL GENXPTS
      DATA XAXISMX/8.0/,YAXISMX/10.0/,XDV/2.0/,XTIC/.5/,
     1    YDV/1.0/,YTIC/1.0/
C
C*********************************************************************
C
C
C     ARTOPLT
C           THIS IS THE ARRAY TO BE PLOTTED WITH 'STEP' AS STEP SIZE
C     NPTS
C          THIS IS THE NUMBER OF POINTS TO BE PLOTTED
C          CAN BE PLOTTED.
C     IAXSTYP
C            THIS IS THE Y-AXIS TYPE DESIRED(LINEAR,SEMI-LOG OR BOTH)
C     XAXISMX
C          THIS IS THE LENGTH IN INCHES OVER WHICH THE RANGE
C          OF X VALUES IS TO BE PLOTTED.
C     XDV
C          THE NUMBER OF DIVISIONS PER INCH OF PAPER TO BE
C          USED FOR THE X AXIS.
C     XTIC
C          THIS IS THE LENTGTH IN INCHES BETWEEN ANNOTATED
C          TIC MARKS ON THE X AXIS.
C     YAXISMX
C          THIS IS THE LENGTH IN INCHES OVER WHICH THE RANGE
C          OF Y VALUES IS TO BE PLOTTED.
C     YDV
C         THE NUMBER OF DIVISIONS PER INCH OF PAPER TO BE
C         USED FOR THE Y AXIS.
C     YTIC
C          THIS IS THE LENGTH IN INCHES BETWEEN ANNOTATED
C          TIC MARKS ON THE Y AXIS
```

EXTERNALS	TYPE	ARGS
ASCALE		5
AXES		11
AXESLOG		10
CALPLT		3
LGLIN		8
LINPLT		8
NFRAME		0
PSEUDO		0
SCALOG		5

```
      SUBROUTINE CPLOT(ARTOPLT,X,STEP,GENXPTS,NPTS,LNORLG,ITITLE,
     1     JTITLE)
C ***********************************************************************
C  THIS ROUTINE WORKS WITH THE DISSPLA PLOTTING PACKAGE.
C ***********************************************************************
C  NOTE - ARRAYS 'ARTOPLT' AND 'X' MUST BE OF THE SAME DIMENSION
C  IN THE CALLING ROUTINE.
      DIMENSION ARTOPLT(1),X(1),ITITLE(1),JTITLE(1)
      LOGICAL GENXPTS
```

EXTERNALS	TYPE	ARGS
ALGPLT		5
AXSPLT		6
BGNPL		1
BLNK1		5
COMPRS		0
CURVE		4
ENDPL		1
FRAME		0
GRAPH		4
GRID		2
MESSAG		4
NOCHEK		0
RESET		1
TITLE		8
YLOG		4

3.4.2 General Purpose Functions

```
      FUNCTION FNCK(NFAC,KFAC)
C
C  THIS FUNCTION COMPUTES BINOMIAL COEFFICIENTS.
C  GIVEN TWO PARAMETERS N,K WILL COMPUTE NCK I.E. NFACT/(N-K)FACT
C  * KFACT.  THIS IS EQUIVALENT TO N(N-1)...(K+1)/(N-K)FACT.
C
```

```
      FUNCTION PREEXP(X)
C
      DATA REALMAX/1.0E+322/,REALMIN/1.0E-293/,EXPMAX/741.67/,
     1     EXPMIN/-675.82/
C
      IF(X.GT.EXPMIN .AND. X.LT.EXPMAX) GO TO 100
C  SET FUNCTION TO A VALUE VERY CLOSE TO 0.0 BUT NOT EQUAL TO 0.0
      IF(X.LE.EXPMIN) PREEXP = REALMIN
C  SET FUNCTION TO THE MAXIMUM NUMBER THE CDC CAN HANDLE.
      IF(X.GE.EXPMAX) PREEXP = REALMAX
      GO TO 200
  100 PREEXP = EXP(X)
  200 RETURN
      END
```

```
       EXTERNALS          TYPE    ARGS
              EXP       REAL        1 LIBRARY
```

4.0 Error Messages

The following sections contain the error and warning messages generated by modules CAREIN, COVRGE, and CARE3. The utility routine error messages can occur in any of the three modules. The error messages are fatal to the computer run; therefore, the run will halt after one occurs. The warning messages are nonfatal and are used to alert the user to any out-of-the-ordinary processing.

The variables enclosed in single quote marks are used to show which variables will have their contents printed. The following variables, used in the error messages, have values which do not change during the course of the computer run:

Variable Name	Value	Description
MAXTYP	5	Maximum Number of Fault Types
MXSTGS	70	Maximum Number of Stages
MAXCAT	5	Maximum Number of Fault Categories
MXUNTS	70	Maximum Number of Coupled Units Per Fault Tree
MXCPLS	20	Maximum Number of Coupled Stages Per Fault Tree
MXSBRN	35	Maximum Number of Subruns
INMAX	513	Maximum Index Into Coverage Functions
MXSTGF	13	Maximum Number of Stage Failures

Variable Name	Value	Description
IJMAX	210	Maximum Number of Unique Fault Pair Functions Per BXYFL Subfile
MAXSTG	20	Maximum Number of Stages Per Subrun

Beside each message is a code used to differentiate the user errors from the possible internal program errors for which tests are made.

4.1 CAREIN ERROR MESSAGES (FATAL)

** ERROR - NFTYPS = 'NFTYPS' MUST BE LESS THAN OR EQUAL TO 'MAXTYP' AND UE
NSTGES = 'NSTGES' MUST BE LESS THAN OR EQUAL TO 'MXSTGS'.

** ERROR - NFCATS('ISTG') = 'NFCATS(ISTG)' MUST BE LESS THAN OR EQUAL TO 'MAXCAT'. UE

** ERROR - EOF ENCOUNTERED ON FILE CREIN WHEN MORE DATA WAS REQUIRED. UE

** ERROR - STAGE NUMBERS MUST RANGE FROM 1 TO 'MXSTGS' MAXIMUM UE
AND THE LAST STAGE MUST BE LESS THAN OR EQUAL TO 'NSTGES' DEFINED AS 'NSTGES'
- NOT 'IFSTG' TO 'ILSTG'.

** ERROR - EXCEEDED MAXIMUM NUMBER OF COUPLED UNITS UE
PER CRITICAL PAIR FAULT-TREE OF 'MXUNTS'.

** ERROR - ILLEGAL STAGE SPECIFIED IN CRITICAL PAIR DATA = 'ISTG'. UE

** ERROR - NUMBER OF UNITS SPECIFIED FOR STAGE 'ISTG', IN THE CRITICAL PAIR DATA, UE
DOES NOT EQUAL THE NUMBER OF IDENTICAL UNITS SPECIFIED FOR THIS STAGE.
IF ALL UNITS ARE NOT INVOLVED IN THE CRITICAL PAIR FAULT-TREE, SET THE NUMBER
OF OPERATING STATES TO THE NUMBER OF INVOLVED UNITS FOR THIS STAGE, AND
INCLUDE EACH UNIT IN THE FAULT-TREE.

** ERROR - EXCEEDED MAXIMUM NUMBER OF COUPLED STAGES UE
PER CRITICAL PAIR FAULT-TREE OF 'MXCPLS'.

** ERROR - EXCEEDED MAXIMUM RANGE OF COUPLED STAGES UE
PER CRITICAL PAIR FAULT-TREE OF 'MXCPLS',
IN THE TREE HAVING FIRST STAGE = 'KSTGF' AND THE LAST STAGE = 'KSTGL'

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

CAREIN ERROR MESSAGES (FATAL)

** ERROR - COUPLED STAGES, DEFINED IN CRITICAL PAIR FAULT-TREES, MUST BE IN ASCENDING ORDER (FROM ONE TREE TO THE NEXT ALSO). UE

** ERROR - OCCURRED WHILE PROCESSING MINTERM FILE FOR CRITICAL PAIR FAULT DEFINITIONS. IE
'PRBMT' '(MNTRMV(I),I=1,NUNTS)'

** ERROR - SUBRUN ARRAY INDEX GREATER THAN MAXIMUM NUMBER OF SUBRUNS = 'MXSBRN'. IE

** ERROR - ILLEGAL FIRST STAGE IN CRITICAL-PAIR SUBSET ARRAY: KSSETS(1,'ISBSET'). IE

CAREIN WARNING MESSAGES (NON-FATAL)

** WARNING - INPUT FILE 'CREIN' EMPTY - DEFAULTS WILL BE USED.

** WARNING - NUMBER OF INTEGRATION STEPS, PLUS 1 TO INCLUDE TIME ZERO, REDUCED FROM 'NSTEPS' TO THE MAXIMUM OF 'MAXSTP'.

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

4.2 COVRGE ERROR MESSAGES (FATAL)

** ERROR - NUMBER OF FAULT TYPES REQUESTED = 'NTYPS' IE
BUT MUST BE LESS THAN OR EQUAL TO 'MAXTYP'.

** ERROR - INCORRECT RELIABILITY TIME BASE = 'TBASE'. IE

** ERROR - THRASHING OCCURRED IN SUBROUTINE COMPFUN.
RERUN PROGRAM WITH A DIFFERENT 'DBLDF' VALUE.

** ERROR - ITH G SINGLE-FAULT FUNCTION = 'ITH' WHILE ONLY G FUNCTIONS 1 THROUGH 9 IE
MAY BE COMPUTED WITH FUNCTION FGSNGL.

** ERROR - 'DELTA'('ITYP') EQUALS ZERO, WHILE THE CONSTANT DENSITY FUNCTION WAS UE
CHOSEN FOR 'DELTA'.
EITHER ASSIGN 'DELTA' A VALUE OR SPECIFY CONSTANT RATE FUNCTION FOR THIS FAULT TYPE.

** ERROR - 'RHO'('ITYP') EQUALS ZERO, WHILE THE CONSTANT DENSITY FUNCTION WAS CHOSEN UE
FOR 'RHO'.
EITHER ASSIGN 'RHO' A VALUE OR SPECIFY CONSTANT RATE FUNCTION FOR THIS FAULT TYPE.

** ERROR - BOTH 'RHO' AND 'DELTA' ('ITYP') EQUAL ZERO, WHILE THE CONSTANT DENSITY UE
FUNCTION WAS CHOSEN FOR BOTH FAULT RATES.
EITHER ASSIGN THEM A VALUE OR SPECIFY CONSTANT RATE FUNCTIONS FOR THIS FAULT TYPE.

** ERROR - ITH C DOUBLE-FAULT FUNCTION = 'ITH' WHILE ONLY C FUNCTIONS 1 AND 2 MAY BE IE
COMPUTED WITH FUNCTION FCDBL.

** ERROR - ITH F DOUBLE-FAULT FUNCTION = 'ITH' WHILE ONLY F FUNCTIONS 1 AND 2 MAY BE IE
COMPUTED WITH FUNCTION FFDBL.

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

COVRGE ERROR MESSAGES (FATAL)

** ERROR - ITH B DOUBLE-FAULT FUNCTION = 'ITH' WHILE ONLY B FUNCTIONS 1 AND 2 MAY BE IE
COMPUTED WITH FUNCTION FBDBL.

** ERROR - THRASHING OCCURRED IN SUBROUTINE SUMARS. RERUN PROGRAM WITH A DIFFERENT UE
'DBLDF' VALUE.

** ERROR - THRASHING OCCURRED IN SUBROUTINE VOLTERA. RERUN PROGRAM WITH A DIFFERENT UE
'DBLDF' VALUE.

** ERROR - THRASHING OCCURRED IN SUBROUTINE VLTNREC. RERUN PROGRAM WITH A DIFFERENT UE
'DBLDF' VALUE.

** ERROR - INDEX PASSED TO FUNCTION FTCHSTP = 'INDX' IS LARGER THAN THE TOTAL NUMBER IE
OF POINTS COMPUTED.

** ERROR - ILLEGAL MCHI OF 'MCHI' WHICH SPECIFIES WHICH SINGLE OR DOUBLE-FAULT IE
FUNCTION TO STORE INTO COMMON /CVRGCOM/,
MUST BE IN THE RANGE OF 1 THROUGH 6 ONLY.

** ERROR WITH MOMNT = 'MOMNT' IN FUNCTION SIMPINT. IE
ONLY MONENTS 0,1, AND 2 ARE VALID.

** ERROR WITH INDICES REPRESENTING INTEGRATION LIMITS IN FUNCTION SIMPINT: 'ITFROM' = IE
'ITFROM' 'ITTO' = 'ITTO' WHERE 'INMAX' = 'INMAX'.

** ERROR - INDEX INTO 'TMAR' = 'IT' IS GREATER THAN MAXIMUM ALLOWABLE INDEX OF 'INMAX'. IE

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

COVRGE WARNING MESSAGES (NON-FATAL)

** WARNING - NSTPIN GREATER THAN 'NSTPMX' IN SUBROUTINE COMPFUN. ZERODF DECREASED TO 'ZERODF'.

** WARNING - NUMBER OF POINTS REQUIRED TO DEFINE A FUNCTION ARRAY IS LARGER THAN 'INMAX'. ZERODF INCREASED TO 'ZERODF'.

** WARNING - NSTPIN GREATER THAN 'NSTPMX' IN SUBROUTINE SUMARS. ZERODF DECREASED TO 'ZERODF'.

** WARNING - NPSTPIN GREATER THAN 'NSTPMX'. ZERODF DECREASED TO 'ZERODF'.

** WARNING - NUMBER OF POINTS REQUIRED TO DEFINE P ARRAY IS LARGER THAN 'INMAX'. ZERODF INCREASED TO 'ZERODF'.

4.3 CARE3 ERROR MESSAGES (FATAL)

** ERROR - INCORRECT TIME BASE = 'TBASE'. IE

** ERROR - 'KWT' = 'KWT'. 'KWT' MUST BE INPUT WHEN THE SYSTEM MINTERM FILE 'FT15F' UE
IS MISSING.
THE NUMBER OF STAGE FAILURES CANNOT EXCEED 'MXSTGF'.

** ERROR - SUBRUN NUMBER EXCEEDED MAXIMUM OF 'MXSBRN'. IE

** ERROR - EOF ENCOUNTERED ON UNIT 4 DURING BUFFER IN. IE

** ERROR - WITH SYSTEM MINTERM FILE 'FT15F' - IE
'PRBMT' (MNTRMV(ISTG),ISTG=1,NSTGRN)

** ERROR - NUMBER OF UNIQUE FAULT PAIR FUNCTIONS = 'NBXY' FOR 'NSTGS' COUPLED STAGES, IE
WHICH EXCEEDS THE MAXIMUM OF 'IJMAX' FOR THE MAXIMUM NUMBER OF COUPLED STAGES =
'MAXSTG'.

** ERROR - INVALID BXYAR INDEX OF 'INBXY' EXISTS IN INDEX ARRAY IJSTGIN('IJSTG'). IE

** ERROR - INVALID CONDITIONAL FAULT VALUE OF 'LC' EXISTS IN CONDITIONAL FAULT UE
VECTOR LCNDVEC('I').

** ERROR - INVALID SPECIFICATION OF WHICH BXY FUNCTION DEFINITION RECORD IS IE
CURRENTLY IN MEMORY - IBREC = 'IBREC'.

** ERROR - EOF ENCOUNTERED ON PREVIOUS BUFFER IN ON UNIT 'IUNIT' WHILE TRYING TO IE
READ RECORD 'IT'.

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

CARE3 ERROR MESSAGES (FATAL)

** ERROR - PARITY ERROR ENCOUNTERED DURING BUFFER IN ON UNIT 'IUNIT' WHILE TRYING TO READ RECORD 'IT'. IE

** ERROR - EOF ENCOUNTERED ON PREVIOUS BUFFER OUT ON UNIT 'IUNIT' WHILE TRYING TO WRITE RECORD 'ITB'. IE

** ERROR - PARITY ERROR ENCOUNTERED DURING BUFFER OUT ON UNIT 'IUNIT' WHILE TRYING TO WRITE RECORD 'ITB'. IE

** ERROR - INVALID SPECIFICATION OF WHICH BXY FUNCTION DEFINITION RECORD IS CURRENTLY IN MEMORY - IBREC = 'IBREC'. IE

** ERROR - CURRENT BLOCK OF CRITICAL PAIR DATA IN MEMORY, STARTING AT STAGE 'KFSTG' DOES NOT MATCH THE CURRENT SUBRUN STARTING AT STAGE 'IFSTG'. IE

** ERROR IN FPMUX -'MUX' LATENT FAULTS IN STAGE 'ISTG' IS LARGER THAN THE TOTAL NUMBER OF FAULTS = 'LX'. IE

** ERROR - FGST CALLED WITH AN INVALID MCHI OF 'MCHI'. IE

** ERROR - INCORRECT FAILED STATE REQUESTED FOR USE IN FUNCTION FHSFST: MCHI = 'MCHI'. IE

** ERROR - ILLEGAL FAULT TYPE = 'ITYP' WHILE TRYING TO READ COVERAGE FUNCTIONS IN ROUTINE FHSFST. IE

** ERROR - INCORRECT FAILED STATE REQUESTED FOR USE IN FUNCTION FHDFST: MCHI = 'MCHI'. IE

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

CARE3 ERROR MESSAGES (FATAL)

** ERROR - ILLEGAL FAULT TYPE(S) = 'ITYP' AND/OR 'JTYP' WHILE TRYING TO READ COVERAGE FUNCTIONS IN ROUTINE FHDFST. IE

** ERROR - INCORRECT FAILED STATE REQUESTED FOR USE IN FUNCTION FFSFST: MCHI = 'MCHI'. IE

** ERROR - INCORRECT FAILED STATE REQUESTED FOR USE IN FUNCTION FFDFST: MCHI = 'MCHI'. IE

** ERROR - ILLEGAL NUMBER PASSED TO N FACTORIAL FUNCTION = 'N'. IE

CARE3 WARNING MESSAGES (NON-FATAL)

** WARNING - SYSTEM MINTERM FILE 'FT15F', GENERATED BY PROGRAM CAREIN, IS EMPTY.
CARE3 WILL GENERATE P*'S WITHOUT USING THE SYSTEM FAULT-TREE DEFINITION,
SINCE A 'KWT' VALUE OF 'KWT' WAS SPECIFIED.

** WARNING - CRITICAL FAULT PAIR FILE 'BXYFL' IS EMPTY.
THIS RUN ASSUMES THAT NO CRITICAL PAIRS OF FAULTS EXIST.

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

4.4 UTILITY ROUTINE ERROR MESSAGES (FATAL)

** ERROR - ILLEGAL COMBINATORIAL; NFAC = 'NFAC' KFAC = 'KFAC'.	IE
** ERROR - ILLEGAL VALUE FOR 'INOUT' IN SUBROUTINE BUFBLK.	IE
** ERROR - EOF ENCOUNTERED ON UNIT 'IUNIT' DURING BUFFER 'BTYPE'.	IE
** ERROR - PARITY ERROR ENCOUNTERED ON UNIT 'IUNIT' DURING BUFFER 'BTYPE'.	IE

NOTE: UE = USER ERROR; IE = INTERNAL ERROR.

This Page Intentionally Left Blank

APPENDIX

COMPUTATIONAL COMPLEXITY IN SOLVING THE "STIFF" VOLTERRA INTEGRAL EQUATION

by
Lynn A. Bryant

This Page Intentionally Left Blank

INTRODUCTION

The Volterra integral equation has received much attention in the numerical analysis literature. Many algorithms exist for solving the Volterra equation due to the fact that "there is a strong link between Volterra equations and initial value ordinary differential equation problems...(but) the major difference in procedure, compared with differential equations, lies in the error (and step size control) techniques needed...(while) no algorithms exist for "stiff" volterra problems."[1]

The purpose of this paper is not to describe existing algorithms for Volterra integral equations but rather to explore one type of Volterra equation which plays a significant role in reliability theory - the renewal equation. The renewal equation is a linear Volterra equation with convolution kernal, defined as the expected number of renewals by time t. In particular, the "stiff" renewal equation will be explored. This is an equation where the functions involved contain vastly different time constants. The computational complexities involved in defining an algorithm to solve this equation will be explored; and a solution will be proposed based on the generalized trapezoidal formula.

VOLTERRA INTEGRAL EQUATIONS

An integral equation of the Volterra type is an equation which involves integrals of an unknown function $x(t)$, which is a real or complex valued function of a single real variable t. The following is a nonlinear integral equation of Volterra type:

---

[1] Delves, L. M., "Numerical Software for Integral Equations," *Numerical Software Needs and Availability*, Academic Press, pg. 305, [1978].

$$x(t) = f(t) + \int_a^t g(t,s,x(s))ds, \quad \text{where } t \geq a. \tag{1}$$

Functions f and g are known, and the lower limit of integration a denotes a fixed real number while the unknown in this equation is the function $x(t)$. Notice that the solution at any point t depends upon the values of $x(s)$ on the interval $a \leq s \leq t$ only.

Another form of the Volterra equation is the nonlinear Volterra integral equation of convolution type:

$$x(t) = f(t) + \int_0^t h(t-s,x(s))ds. \tag{2}$$

The linear form of this equation

$$x(t) = f(t) + \int_0^t k(t,s)x(s)ds \tag{3}$$

contains a function k called the kernal of the equation.

The equation, which is both linear and of convolution type,

$$x(t) = f(t) + \int_0^t a(t-s)x(s)ds \tag{4}$$

will be the focus of this paper. Function $f(t)$ is called the forcing function.

This linear Volterra equation of convolution type, which arises in a variety of mathematical, engineering and scientific applications, is commonly referred to as the classic renewal equation. The forcing function f is continuous, and the kernal $a(t)$ is of class L^1 on each finite subinterval of $[0,\infty)$. The renewal equation is defined as the expected number of renewals by time t. It can be used, for example, to determine the expected number of failures in a given time, or the probability distribution of the time at which all failures occur.

NUMERICAL METHODS AND VOLTERRA INTEGRAL EQUATIONS

Due to the strong link between Volterra equations and initial value ordinary differential equation problems, the numerical methods used to solve both are also strongly linked. Therefore equation (1) can be rewritten as

$$x(t+t_0) = f(t+t_0) + \int_a^t g(t+t_0, s, x(s))ds + \int_t^{t+t_0} g(t+t_0, s, x(s))ds \qquad (5)$$

which involves a step-by-step solution. This recursive relationship contains implicitly the initial condition $x(a) = f(a)$.

Several numerical methods algorithms exist for the solution of (1). The methods are often based on the Runge-Kutta or the multi-step (predictor-corrector) methods. For an excellent comparison of existing algorithms see Delves in Jacobs [1978].

STIFF VOLTERRA INTEGRAL EQUATION WITH CONVOLUTION KERNAL

The type of Volterra equations for which no algorithms exist are termed "stiff" Volterra problems. "Stiff equations contain time constants with greatly different values. Stability requires that the step size be not much greater than the smallest time constant present when conventional methods are used. The terms due to the small time constants decay rapidly, but the step size cannot be increased because of stability."[1]

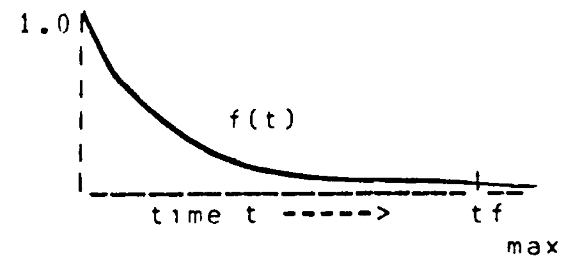
---

[1] Gear, C.W., "The Automatic Integration of Stiff Ordinary Differential Equations", Information Processing 68, pg. 187, [1968].

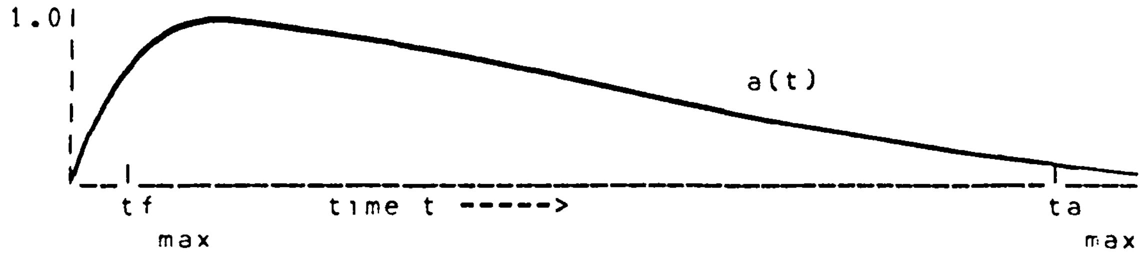
As a very straightforward example of the stiff Volterra equation of (4), let function f be of exponential type of the form

$$f(t) = e^{-at}, \quad \text{where } a > 0$$

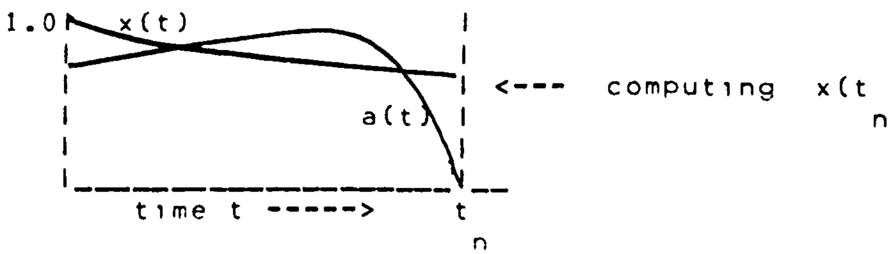
with the following graphical representation:

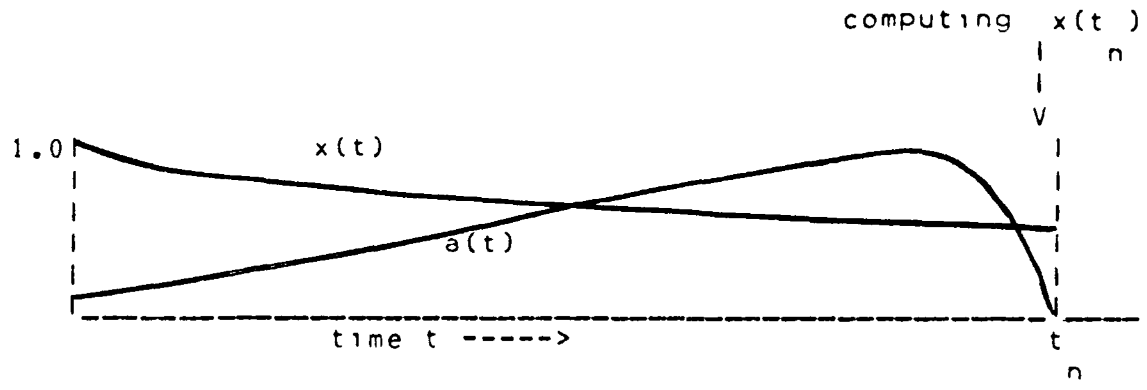


and let a(t) be a continuous function graphically represented by



where ta_{max} is many magnitudes larger than tf_{max}. Determine an effective zero point for each function by choosing a truncation value $t_r \leq 10^{-3}$. Function f is defined over the interval $[0, tf_{max}]$, and function a is defined over the interval $[0, ta_{max}]$. In order to illustrate the processing involved in solving x(t), given f(t) and a(t) as defined above, the following graphs are presented.





The following five points are illustrated by the previous graphs:

1) Initially $x(0) = f(0) = 1.0$.

2) $a(0) = 0$, therefore it is not necessary to know the value of $x(t_n)$ when solving for $x(t_n)$.

3) For each successive computation of $x(t_n)$, where $0 \leq t_n \leq tx_{max}$, convoluted kernal a moves farther out on the time axis.

4) tx_{max} is determined when either the function x equals the specified truncation value t_r or function x reaches a steady state value.

5) tx_{max} may be magnitudes larger than either tf_{max} or ta_{max}.

In order to solve this "stiff" Volterra equation with convolution kernal, the step size must be carefully chosen to ensure stability. Therefore the smallest step size required by the functions involved in the integration determines the step size. But if this is used throughout the computation of $x(t)$, the time required to compute the number of points necessary to define function x, from time zero to time tx_{max}, would not be computationally feasible due to the extremes of the time maximums of each function involved in the computation. Storage of function $x(t)$ could also cause a problem due to the recursive nature of the solution.

To illustrate the enormous number of calculations required to compute $x(t)$ under these stiff conditions, let tf_{max} = 1 minute and ta_{max} = 1000 hours. Let the step size of function f equal 1 second, yielding 60 computed values for $f(t)$ over the interval $[0, tf_{max}]$. If the 1 second step size were also used for function a, which would not be necessary, 3,600,000 points would be computed for function a. Instead, let the step size for function a equal 10 hours, yielding 100 computed values for $a(t)$ over the interval $[0, ta_{max}]$. In order to preserve stability, function x's step size must initially equal 1 second. If this step size were used to compute $x(t)$ over the interval $[0, tx_{max}]$, the number of computed $x(t)$ values might exceed 3,600,000 values, due to the convoluted kernal. This method is not computationally feasible because each new computed x value requires the multiplication of all previous computed x values times the shifted function a. This would result in a minimum of 3,600,000 factorial multiplications plus 3,600,000 additions.

One possible solution to the step size problem would allow for the step size of function $x(t)$ to increase as the rate of change of $x(t)$ decreases by some specified amount. Doubling the step size each time an increase is warranted would minimize the number of time values that must be stored with the computed function values.

TRAPEZOIDAL SOLUTION

To obtain the approximate solution to equation (4), using the generalized trapezoidal formula, divide the interval [0,t] into n parts to yield the solution

$$x(t_n) \approx f(t_n) + h/2\,[x(0)a(t_n) + 2(x(t_1)a(t_n - t_1) + \dots + x(t_{n-1})a(t_1)) + x(t_n)a(0)],$$

where $t_0 = 0$, $t_1 = h$, ..., $t_n = nh = t$.

$x(t_n)$ must be eliminated from the right hand side of the equation because this is the value currently being computed. Subtract the final product from both sides of the equation to yield

$$x(t_n) - h/2\,[x(t_n)a(0)] \approx f(t_n) + h/2\,[x(0)a(t_n) + 2(x(t_1)a(t_n - t_1) + \ldots + x(t_{n-1})a(t_1))].$$

Factor out $x(t_n)$ to yield

$$x(t_n)(1 - (h/2)a(0)) \approx f(t_n) + h/2\,[x(0)a(t_n) + 2(x(t_1)a(t_n - t_1) + \ldots + x(t_{n-1})a(t_1))].$$

Finally divide by $(1 - (h/2)a(0))$ to yield

$$x(t_n) \approx [1/(1 - (h/2)a(0))]\,[f(t_n) + h/2\,[x(0)a(t_n) + 2(x(t_1)a(t_n - t_1) + \ldots + x(t_{n-1})a(t_1))].$$

This equation can be written in a more general form as

$$x(t_n) \approx [1/(1 - (h/2)a(0))]\,[f(t_n) + (h/2)x(0)a(t_n) + h \cdot \sum_{i=1}^{n-1} x(t_i)a(t_n - t_i)] \qquad (6)$$

The following solution, based on the trapezoidal method described above, is a proposed solution to the stiff Volterra integral equation with convolution kernal:

$$x(t_n) \approx [1/(1 - ((t_n - t_{n-1})/2)a(0))]\ [f(t_n) + (t_1/2)x(0)a(t_n) + \sum_{i=1}^{n-1} ((t_{i+1} - t_{i-1})/2)x(t_i) \cdot a(t_n - t_i)] \qquad (7)$$

Notice that this solution does not require equal step sizes for functions x and a; this is mandatory due to the extremes in the times for which the functions exist. The following is a graphical representation of the times involved in the previous solution for which functions a and x are known:

```
                                                    t                t
                                                     k+1              n-1|
|________________|__|__|__|__|__|__|__|__|__|__|__|__|__|__|__| a
|                                                                |
|                                                                |
|                                                                |
|                                                                |
|__|__|____|________|________|________|__________________| x
0  t                                    t                  t
    1                                    k                  n
                  time t ----->
```

Function x has been computed up to time t_k and is currently being solved for time t_n. In solution (7) $a(t_n - t_{n-1})$ must be multiplied by $x(t_{n-1})$ in order to preserve stability and likewise $a(t_n - t_{n-2})$ must be multiplied by $x(t_{n-2})$ etc. It should be evident that a problem exists because $x(t_k)$ is the last computed value for function x, where $t_k < t_{k+1} < \ldots < t_{n-2} < t_{n-1} < t_n$. Function x could be extrapolated but, due to the recursive nature of the problem, this would introduce an inordinate amount of error, and instability would be introduced.

The following solution eliminates the extrapolation problem:

$$x(t_n) \approx \frac{[f(t_n) + (t_1/2)x(0)a(t_n) + \sum_{i=1}^{k} ((t_{i+1} - t_{i-1})/2)x(t_i)a(t_n - t_i) + x(t_k) \cdot \sum_{i=k+1}^{n-1} ((t_{i+1} - t_{i-1})/2)[(t_n - t_i)/(t_n - t_k)]a(t_n - t_i)]}{1 - \sum_{i=k+1}^{n-1} ((t_{i+1} - t_{i-1})/2)[t_i - t_k)/t_n - t_k)]a(t_n - t_i) - ((t_n - t_{n-1})/2)a(0)} \quad (8)$$

Solution (8) is an expanded version of (7); the integration has been split into two sections:

1) The integration over the interval $[0,t_k]$, where function x has been previously computed and can be interpolated if necessary;

2) the integration over the interval $[t_k,t_n]$, where function x is unknown, except for point $x(t_k)$.

This solution uses linear interpolation between the last known value $x(t_k)$ and unknown value $x(t_n)$ using the weights $[(t_n - t_i)/(t_n - t_k)]$ and $[(t_i - t_k)/(t_n - t_k)]$. Then $x(t_n)$ is eliminated from the right hand side of the equation in the exact same manner as described above for generating solution (6).

This solution minimizes the problems that make the stiff Volterra equation infeasible to compute. It allows for the step size of function x to be initially as small as necessary to ensure stability, and then enables the step size to increase as the function changes less dramatically. Each successive computed $x(t)$ value uses all of the points of functions a and f, and possibly other interpolated points of these functions, no matter how small the step sizes happen to be in relation to the current step size of function x. This method eliminates unnecessary $x(t)$ computations but incorporates all necessary points of functions a and f in each $x(t)$ calculation.

CONCLUSION

No general purpose algorithms exist for solving the stiff Volterra integral equation with convoluted kernal due partly to the problems of determining the correct step size to ensure stability. Also, the convoluted kernal requires that for each successive $x(t)$ calculation all previous x values must be multiplied by the kernal function shifted, so that, in general, $a(t_n - t_{n-m})$ is multiplied by $x(t_{n-m})$, where $t_{n-m} < t_n$.

The solution based on the trapezoidal method, which allows for adaptive step sizes, eliminates these problems and works well in solving the renewal equation with continuous, "well behaved" functions that are piece-wise linear.

ACKNOWLEDGEMENT

Special thanks must be extended to Dr. J.J. Stiffler for his help with the final trapezoidal solution.

BIBLIOGRAPHY

1. Berezin, I.S., and N.P. Zhidkov [1965]. Computing Methods, Addison - Wesley Publishing Co., Reading, MA.

2. Delves, L.M. [1978]. "Numerical Software for Integral Equations," in Jacobs [1978] pp. 303 - 323.

3. Fox, L. [1962]. Numerical Solution of Ordinary and Partial Differential Equations, Addison - Wesley Publishing Co., Reading, MA.

4. Gear, C. W. [1968]. "The Automatic Integration of Stiff Ordinary Differential Equations," Information Processing 68, pp. 187 - 193.

5. Hartmanis J. [1968]. Feasible Computations and Provable Complexity Properties, Society for Industrial and Applied Mathematics, Philadelphia, PA.

6. Hennell, M. A. and L. M. Delves [1980]. Production and Assessment of Numerical Software, Academic Press, New York, NY.

7. Jacobs, D. [1978]. Numerical Software Needs and Availability, Academic Press, New York, NY.

8. Miller, R.K. [1971]. Nonlinear Volterra Integral Equations, W.A. Benjamin, Inc., Menlo Park, CA.

9. Riddell, I. and L. M. Delves [1980]. "The Numerical Testing of Integral Equation Routines," in Hennell and Delves [1980], pp. 41 - 56.

1 Report No NASA CR-165863	2 Government Accession No	3 Recipient's Catalog No

4 Title and Subtitle CARE III PHASE II Report, Maintenance Manual	5 Report Date September 1982 6 Performing Organization Code
7 Author(s) L. A. Bryant and J. J. Stiffler	8 Performing Organization Report No
9 Performing Organization Name and Address Raytheon Company Sudbury, MA 01776	10 Work Unit No 11 Contract or Grant No NAS1-15072
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546 and Wright-Patterson AFB, Air Force Avionics Lab., Dayton, OH	13 Type of Report and Period Covered Contractor Report 14 Sponsoring Agency Code 505-34-43-05

15 Supplementary Notes

NASA Project Engineer, Salvatore J. Bavuso
WPAFB Technical Monitor, Dennis Miller (AFWAL)

16 Abstract

CARE III (Computer-Aided Reliability Estimation, version three) is a computer program designed to help estimate the reliability of complex, redundant systems. Although the program can model a wide variety of redundant structures, it was developed specifically for fault-tolerant avionics systems--systems distinguished by the need for extremely reliable performance since a system failure could well result in the loss of human life.

The first CARE program, developed at the Jet Propulsion Laboratory in 1971, provided an aid for estimating the reliability of systems consisting of a combination of any of several standard configurations (e.g., standby-replacement configurations, triple-modular redundant configurations, etc.). CARE II was subsequently developed by Raytheon, under contract to the NASA Langley Research Center, in 1974. It substantially generalized the class of redundant configurations that could be accommodated, and included a coverage model to determine the various coverage probabilities as a function of the applicable fault recovery mechanisms (detection delay, diagnostic scheduling interval, isolation and recovery delay, etc.).

CARE III further generalizes the class of system structures that can be modeled and greatly expands the coverage model to take into account such effects as intermittent and transient faults, latent faults, error propagation, etc.

17 Key Words (Suggested by Author(s)) Reliability modeling Fault coverage Fault models Fault-tolerant avionics	18 Distribution Statement Unclassified - Unlimited Subject Category 59

19 Security Classif (of this report) Unclassified	20 Security Classif (of this page) Unclassified	21 No of Pages 159	22 Price* A08

N-305

*For sale by the National Technical Information Service, Springfield Virginia 22161

End of Document