

LEVEL

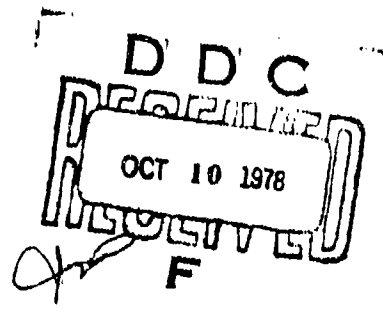
12

STRUCTURAL AREA INSPECTION FREQUENCY EVALUATION (SAIFE)

Volume IV. Software Documentation and User's Manual

Book 2. Modified Program

James Gillespie



DDC FILE COPY AD A059539

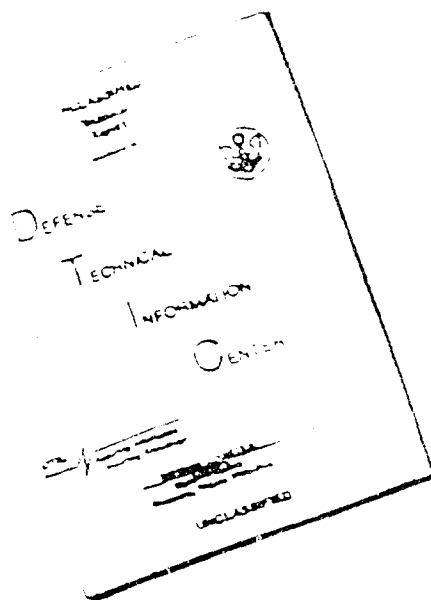
APRIL 1978
FINAL REPORT

Document is available to the U.S. public through
the National Technical Information Service,
Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20590

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST
QUALITY AVAILABLE. THE COPY
FURNISHED TO DTIC CONTAINED
A SIGNIFICANT NUMBER OF
PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

REPRODUCED FROM
BEST AVAILABLE COPY

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

1. Report No. FAA-RD-78-29, IV, Book 2	2. Government Accession No. (10111)	3. Recipient's Catalog No. (11)
4. Title and Subtitle STRUCTURAL AREA INSPECTION FREQUENCY EVALUATION Volume IV. Software Documentation and User's Manual. Book 2 ~~MODIFIED PROGRAM~~		5. Report Date April 1978
6. Author(s) James Gillespie		6. Performing Organization Code AFS-510
9. Performing Organization Name and Address FAA, AFS-512 Aeronautical Center, P. O. Box 25082 Oklahoma City, Oklahoma 73125		8. Performing Organization Report No.
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, D.C. 20590		10. Work Unit No. (TRAIS)
13. Supplementary Notes 4/29/78		11. Contract or Grant No. DOT-FA74WA-3493
16. Abstract To assist in the evaluation of proposed structural inspection programs for commercial jet transport aircraft, a logic was developed to simulate structural defects, failures, and inspections. This logic was incorporated in a computer program entitled Structural Area Inspection Frequency Evaluation (SAIFE). With the objective of quantifying the evaluation process currently used to establish and modify inspection intervals, SAIFE accounts for the following factors: (1) aircraft design analysis; (2) fatigue testing; (3) production, service, and corrosion defects; (4) probability of crack or corrosion detection; and (5) aircraft modification economics. As a five-volume document, this report covers the initial contract effort plus a subsequent parametric analysis as follows: Volume I (~~entitled Executive Summary~~) presents the SAIFE logic and documents the methodology for the decision-making processes in the simulation logic. Volume II (~~entitled Description of Simulation Logic~~) details the SAIFE simulation logic, presents the background data for the analytical functions and decision-making processes, and includes data for a typical simulation problem. Volume III (~~entitled Demonstration Input, Inspection Survey, and MRR Data~~) presents data tabulations derived from historical trends and design input data for a SAIFE demonstration problem. As the user's manual for the SAIFE computer program, Volume IV contains detailed computer logic flow diagrams and a complete listing of the program which is written in SIMSCRIPT II.5. Volume V (~~entitled Results of Model Demonstration~~) presents the results of the program application to a hypothetical aircraft and compares these results with the service experience of operational aircraft.		13. Type of Report and Period Covered Final Report
17. Key Words Aircraft experience simulation; aircraft inspection intervals; aircraft hazard rate and reliability; production, service, and corrosion defects; crack or corrosion detection probability; crack growth.		14. Sponsoring Agency Code
18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.		15. Sponsoring Agency Code
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages 280
22. Price		

344 734

y/B

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
in	inches	2.5	Centimeters	cm
ft	feet	30	Centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
AREA				
in ²	square inches	6.5	square centimeters	cm ²
ft ²	square feet	0.09	square meters	m ²
yd ²	square yards	0.8	square meters	m ²
mi ²	square miles	2.6	square kilometers	km ²
	acres	0.4	hectares	ha
MASS (weight)				
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons	0.9	tonnes	t
	(2000 lb)			
VOLUME				
cup	teaspoons	5	milliliters	ml
fl oz	tablespoons	15	milliliters	ml
c	fluid ounces	30	milliliters	ml
pt	cups	0.24	liters	l
qt	pints	0.47	liters	l
gal	quarts	0.95	liters	l
cu ft	gallons	3.8	liters	l
	cubic feet	0.03	cubic meters	m ³
	cubic yards	0.76	cubic meters	m ³
TEMPERATURE (exact)				
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

* 1 m = 2.54 (exact). For other exact conversions and more detailed tables, see NBS Unc. Publ. 286, Units of Weight and Measures, Price \$2.25, SC Catalog No. C12.102266.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
km	kilometers	1.1	miles	mi
		0.6	miles	mi
AREA				
cm ²	square centimeters	0.16	square inches	in ²
m ²	square meters	1.2	square yards	yd ²
km ²	square kilometers	0.4	square miles	mi ²
ha	hectares (10,000 m ²)	2.5	acres	ac
MASS (weight)				
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	ton
VOLUME				
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m ³	cubic meters	35	cubic feet	ft ³
m ³	cubic meters	1.3	cubic yards	yd ³
TEMPERATURE (exact)				
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



TABLE OF CONTENTS

INTRODUCTION	1
SYSTEM DESCRIPTION	1
GENERAL PROGRAM DESCRIPTION	3
PROGRAM INPUT	3
AIRCRAFT DATA	4
LONG LIST DATA	7
ELEMENT DATA	7
FORMAT SPECIFICATIONS	11
SAMPLE INPUT	11
PROGRAM OUTPUT	14
STANDARD OUTPUT ELEMENT DATA	14
STANDARD OUTPUT SUMMARY DATA	28
LONG LIST DATA	28
APPENDIX A - DETAILED PROGRAM DESCRIPTION	34
APPENDIX B - MODIFIED PROGRAM SOURCE LISTING	92

ACCESSION for	
NTIS	Code Section ☒
DIC	Code Section ☐
UNANNOUNCED	☐
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
IN	SPECIAL
A	

INTRODUCTION

The Structural Area Inspection Frequency Evaluation (SAIFE) computer simulation was developed for the FAA under contract DOT-FA74WA-3493 by Technology Inc., Dayton, Ohio. The computer program simulates the structural inspection program of an airline fleet. SAIFE accounts for the following factors:

1. Aircraft design analysis
2. Component and full scale fatigue testing
3. Production, service and corrosion defects
4. Probability of crack or corrosion detection
5. Aircraft component repair or modification

The original computer model has been extensively modified by the Engineering and Manufacturing Branch, Flight Standards National Field Office. Modifications have occurred in program input, simulation logic, and program output. Major modifications were:

1. Replacement of the two crack growth rate model by a four crack growth rate model.
2. Introduction of sampling at the D-level inspection in that certain aircraft receive an internal inspection based upon program input and program logic.
3. Additional logic to calculate the probability of aircraft failure based upon crack distribution.
4. Changes to inspection reliability curves which determine probability of defect detection at a given inspection level.
5. Changes to logic for reducing inspection intervals and scheduling special inspections based on crack detection.

During the program modification period Technology Inc. provided support under an additional FAA contract.

SYSTEM DESCRIPTION

The eight blocks in Figure 1 represent the major aspects of the SAIFE simulation logic. Block 1 accepts input data for the aircraft fleet and for each structural element in the aircraft. After determining whether element modifications are required because of the fatigue test results in Block 2, Block 1 assigns a fatigue life to each element in each aircraft. Block 3 determines whether production, service, or corrosion defects will occur; if it is determined that such defects will occur, Block 3 predicts the times when they will occur. After comparing the flight loads with the strength of each element, Block 4 predicts the time to failure for each element. Block 5 conducts the periodic inspections of each element. If defects are detected, Block 6 repairs the element and assigns it a new fatigue life. However, if an existing defect is allowed to grow until element failure, Block 5 deletes the aircraft from the fleet. Depending on the magnitude of the detected defects, special inspections and increased inspection frequencies may be

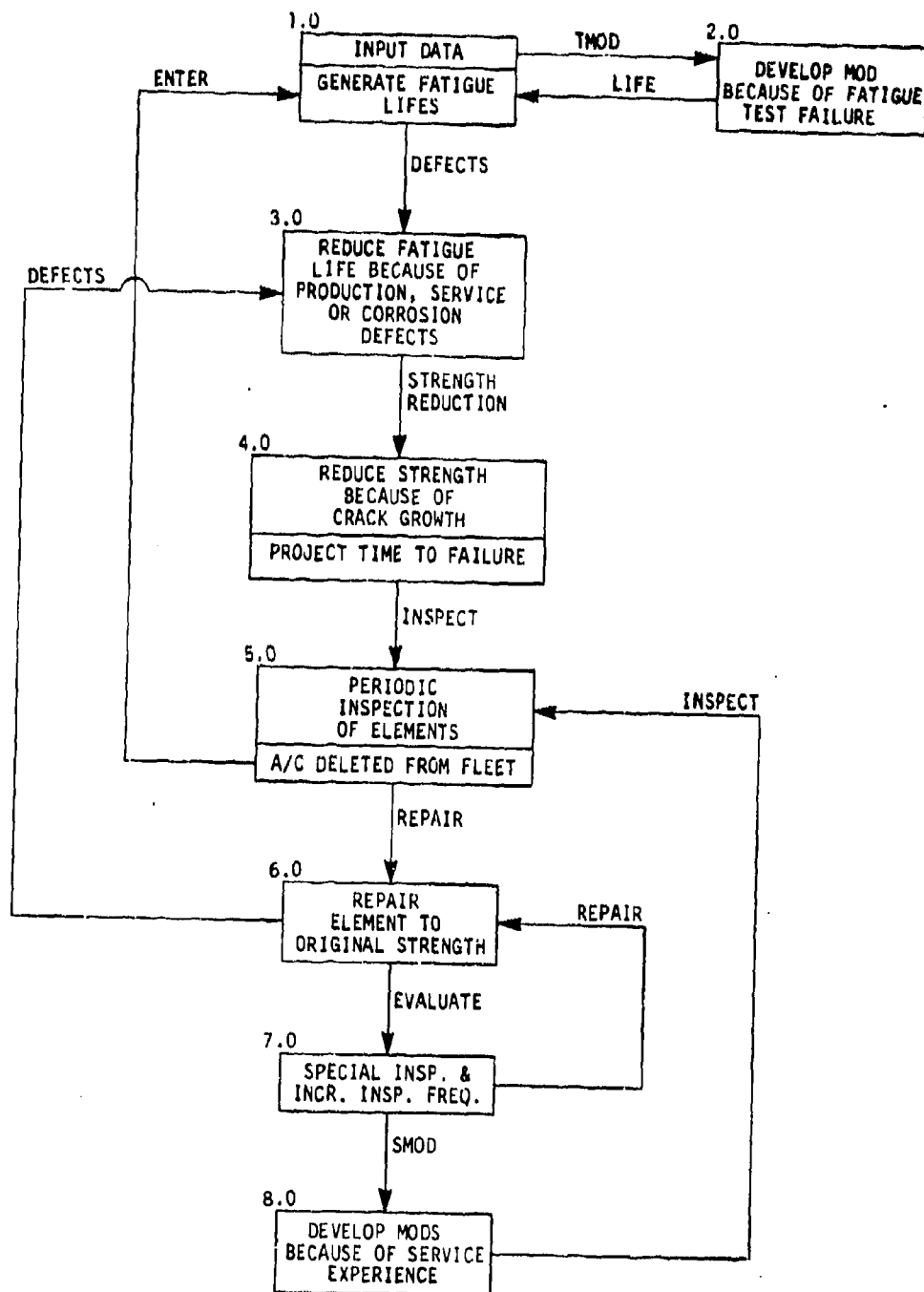


Figure 1. Major Aspects of SAIPE Logic

called for in Block 7 and modifications may be instituted in Block 8. When all the aircraft have been deleted from the fleet or retired from service, the simulation is complete.

GENERAL PROGRAM DESCRIPTION

The SAIFE program is a large, complex math model designed to simulate the structural performance of an aircraft fleet and the effectiveness of the inspection program for the aircraft fleet. The aircraft model in the computer simulation is divided into structurally significant elements and the inspection program for each element is defined. Structural defects are classified as follows: fatigue and corrosion which are wear-out and aging phenomena; production or design defects; and, operational or maintenance damage. These defects and the inspection program are treated as probabilistic phenomena interacting over time. SIMSCRIPT II.5 (trademark, Consolidated Analysis Centers, Inc., Los Angeles, California), a computer language designed for discrete-event simulation applications, was chosen for the SAIFE simulation.

SAIFE uses ten random number streams to generate the probabilistic phenomena in the simulation. Because random numbers generation is computer word length dependent, different computer systems will generate different random numbers and thus different output for a given set of input data. This has been noticed since the computer program has been run on both IBM and CDC computers. Results may be duplicated on a given computer since the compiler initializes the seeds for the random number streams to the same value before each run. An option has been incorporated in the computer program so that the user can input seeds to evaluate independent trials for a given set of input.

Because of the extensive detail, the program events and routines are described in Appendix A.

PROGRAM INPUT

The program input consists of three parts. The first part consists of input variables which pertain to the aircraft type under consideration. These variables are input only once per simulation run and are constant from element to element. If the user desires to input random number seeds, the ten seeds are input after the aircraft input. The second part of the input consists of long list data if standard output is not used. The third part of the input consists of input variables whose values are unique to each element. These variables must be input in their entirety for each element being simulated.

AIRCRAFT DATA. The input variables which pertain to the aircraft type are listed and described below in the order in which they are read in by SAIFE.

MODEL (*) - This one-dimensional alpha array of size two identifies the aircraft type under consideration. The total length of this identification cannot exceed eight characters.

SIZE.OF.FLEET - This integer variable is the number of aircraft in the fleet being simulated. The output format requires that this variable does not exceed 99999.

USAGE.LIFE - This real variable is the service life in flight hours of the aircraft being simulated. All aircraft in the fleet must have the same service life. The output format requires that this variable does not exceed 9999999.

BEGIN.PRODUCTION - This real variable is the time in flight hours relative to the start of the simulation when the first aircraft enters service. This variable in conjunction with the input variable START.TEST enables the user to start the fatigue test of the element before, after, or at the same time the first aircraft enters service.

PRODUCTION.TIME - This real variable defines the initial aircraft production rate. It is the time in flight hours between aircraft entering service.

2.PRODUCTION.TIME - This real variable defines the second aircraft production rate. It is the time in flight hours between aircraft entering service.

PRCHG - This real variable is the simulation time when the second aircraft production rate takes effect. Note that this time is measured from the time that the first aircraft enters service and not from the start of the simulation.

START.TEST - This real variable is the time in flight hours relative to the start of the simulation when the fatigue test of an element is begun. If no fatigue test is to be conducted, this variable is set to the machine upper limit.

TEST.ACCEL.FACT - This real variable is the fatigue test acceleration factor, that is, the quotient of the equivalent flight hours divided by the fatigue test hours.

C.GROWTH.RATE - This real variable is the corrosion area growth rate in square inches per hour for the aircraft being considered. The growth rate for each element in the aircraft is modified by its associated CRR (corrosion resistance rating).

C7 - If a modification is developed because of a fatigue test failure, this real variable is the percentage (expressed as a decimal fraction) of the test life when the inspection frequency is increased.

C28 - This real variable is the percentage (expressed as a decimal fraction) reduction in the remaining fatigue life of an element when corrosion occurs in a stress concentration.

C29 - This real variable is the percentage (expressed as a decimal fraction) reduction in the remaining fatigue life of an element when corrosion occurs outside a stress concentration.

MU.R - This real variable is the mean of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life.

SIG.R - This real variable is the standard deviation of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life.

DLL - This real variable is the design limit load in g's above the 1-g level.

1ABCD(1) - This real variable is the inspection interval in flight hours of the A-level inspection. It remains constant throughout the simulation.

1ABCD(2) - This real variable is the inspection interval in flight hours of the B-level inspection. It remains constant throughout the simulation.

CABCD(*) - This one-dimensional real array of size four contains the inspection cost at each level of inspection. CABCD(1) corresponds to the A-level cost; CABCD(2) corresponds to the B-level cost; CABCD(3) corresponds to the C-level cost; and CABCD(4) corresponds to the D-level cost.

S.OPT - This alpha variable is "YES" if the random number seeds are to be input; it is "NO" if seeds are not input.

LONG.LIST - This alpha variable is "NO" if standard output and "YES" for detailed output on particular aircraft. (LONG LIST DATA).

FAIL.OPT - This integer variable is "1" for output of probability of failure that is based on averaging individual element failure rates to obtain element type failure rates. This integer variable is "2" for output of probability of failure that is based on using a log-normal crack distribution and a curve fit of probability of failure versus crack length to obtain element type failure rates. This integer variable is "3" if both options are desired.

FAT.TEST.FACTOR - This real variable is the probability of a fatigue test being done on the structural element. This real variable is compared with a random number to determine if the fatigue test is done. If the fatigue test is not done, the fatigue test life is set to 9999999.

ACTUAL.AVG.FAT.LIFE - This real variable is the actual average fatigue life in flight hours determined by fatigue test. If this value is not known, input zero and SAIPE will determine it statistically.

LEAD.TIME - This real variable is the time in flight hours between when a decision is made to develop a structural modification and the time the modification is available for installation.

T.FREQ.CHG - This real variable is the percentage expressed as a decimal fraction that the D-level inspection interval is reduced due to a fatigue test failure.

S.FREQ.CHG - This real variable is the percentage expressed as a decimal fraction that the D-level inspection interval is reduced due to service experience.

FREQ.DECREASE - This real variable is the percentage expressed as a decimal fraction that the C and D-level inspection intervals are increased due to favorable service experience.

A.REPAIR.COST - This real variable is the repair cost at the A-level inspection.

B.REPAIR.COST - This real variable is the repair cost at the B-level inspection.

C.REPAIR.COST - This real variable is the repair cost at the C-level inspection.

D.REPAIR.COST - This real variable is the repair cost at the D-level inspection.

1ST.TOOLING - This real variable is the tooling cost of the first structural modification.

AD.TOOLING - This real is the tooling cost in the development of any additional structural modifications.

1ST.MD.COST - This real variable is the installation cost of the first structural modification.

AD.MD.COST - This real variable is the installation cost of any additional structural modification.

S.REPAIR.COST - This real variable is the repair cost of a defect detected during a special inspection.

SU - This real variable is the element ultimate strength in g's above the 1-g level.

SF - This real variable is the element fail safe strength in g's above the 1-g level.

S1 - S10 - These integer variables are the ten random number seeds and are input only if S.OPT = "YES". Any integer value may be used as input.

LONG LIST DATA. Occasionally in the standard output, elements will appear with unusually long fatigue cracks or early element failures. It is desirable to have a more complete service history of aircraft with these early element failures than that offered by the standard output. This service history is available through what is called the long list option. This output option is accessed by reading in alpha characters "YES" for the aircraft input variable LONG.LIST. After this input, the element description and identification numbers of the aircraft to be tracked are read in. The input variables for the long list option are listed and described below in the order in which they are read in by SAIFE.

NOE - This integer variable is the number of elements to be processed under the long list option.

ELID(*,*) - This two-dimensional alpha array of size four by NOE identifies each element to be processed. This identification must appear in the first sixteen columns of the data card and must be identical to the description read into the variable ELEMENT(*) described in ELEMENT DATA.

NOAC(*) - This one-dimensional integer array of size NOE is the number of aircraft to be tracked for each corresponding element.

TLID(*,*) - This two-dimensional integer array of size NOE by NOAC(*) contains the identification numbers of the aircraft to be tracked for a particular element.

ELEMENT DATA. The input variables which are unique to each element and must be read in for each element are listed and described below in the order in which they are read in by SAIFE.

ELEMENT(*) - This one-dimensional alpha array of size four identifies the element being simulated. The total length of this identification cannot exceed sixteen characters. The SAIPE program distinguishes between elements that are pressure loaded and flight loaded. Elements that are pressure loaded must start with characters FUS- and have MFR- as characters five through eight or FRM- as characters nine through twelve. Any characters can be used for the flight loaded elements.

PREDICTED.LIFE - This real variable is the average element fatigue life in flight hours predicted by analysis. If the actual average fatigue life is known, this variable can be entered as zero.

M1.MEAN - This real variable is the average first external crack growth rate in inches per flight hour.

M2.MEAN - This real variable is the average second external crack growth rate in inches per flight hour.

LGHT.TO.FAILURE - This real variable is the length in inches at which the crack reaches failure under a 1-g load

CONE - This real variable is the crack length in inches at which the first external crack growth rate changes to the second external crack growth rate. (First external critical crack length).

FSAF.LGT - This real variable is the length in inches at which the crack reaches the fail-safe length.

BIRTH.DEFECT.PROBABILITY - This real variable is the probability of a production defect.

CRR - This integer variable is the corrosion resistance rating.

SDM.OCCURRENCE.RATE - This real variable is the occurrence rate of service damage per element per aircraft per flight hour.

I.PROB - This real variable is the probability of cracks originating internally.

C.PROB - This real variable is the probability of corrosion originating internally.

INT.LVL.INSPECTION - This alpha variable is the letter identifying the lowest internal level inspection.

EXT.LVL.INSP - This alpha variable is the letter identifying the lowest external level inspection.

MOD.TEST - This alpha variable is input as "YES" if a structural modification is to be fatigue tested. Otherwise it is input as "NO".

LOCATED.IN.STRESS.CON - This real variable is the probability that there is corrosion in a stress concentration.

1.CDM.OCCURRENCE.RATE - This real variable is the initial corrosion occurrence rate in occurrences per element per aircraft per flight hour.

2.CDM.OCCURRENCE.RATE - This real variable is the second corrosion occurrence rate in occurrences per element per aircraft per flight hour.

CDM.RATE.CHANGE - This real variable is the aircraft service time in flight hours when the second corrosion occurrence rate takes effect.

L.EXT - This real variable is the length in inches at which a crack originating internally becomes external.

M3.MEAN - This real variable is the average third external crack growth rate in inches per flight hour.

M4.MEAN - This real variable is the average fourth external crack growth rate in inches per flight hour.

CTWO - This real variable is the crack length in inches at which the second external crack growth rate changes to the third external crack growth rate. (Second external critical crack length).

CTHREE - This real variable is the length in inches at which the third external crack growth rate changes to the fourth external crack growth rate. (Third external critical crack length).

INT.CONE - This real variable is the length in inches at which the first internal crack growth rate changes to the second internal crack growth rate. (First internal critical crack length).

INT.CTWO - This real variable is the length in inches at which the second internal crack growth rate changes to the third internal crack growth rate. (Second internal critical crack length).

IN.CTHREE - This real variable is the length in inches at which the third internal crack growth rate changes to the fourth internal crack growth rate. (Third internal critical crack length).

IABCD(3) - This real variable is the initial inspection interval in flight hours of the C-level inspection.

IABCD(4) - This real variable is the initial inspection interval in flight hours of the D-level inspection.

POP.SIZE - This integer variable is the number of elements of the same type on the aircraft. It is not necessary to input all elements of the same type. A sampling may be used and SAIFE will extrapolate the probability of failure calculation to the total number of elements actually in the aircraft.

AMEAN - This real variable is the result of fitting an exponential curve to flight or pressure load exceedance data. $AMEAN \cdot \exp(BL)$ is the number of loads per hour which exceed the load level (L).

B - This real variable is the result of fitting an exponential curve to flight or pressure load exceedance data. $AMEAN \cdot \exp(BL)$ is the number of loads per hour which exceed the load level (L).

FORMAT SPECIFICATIONS. Most of the input data are entered into SAIFE by the free-form read statement. The program has only three formatted read statements.

The aircraft type identification, the alpha array MODEL, is entered under the format specifications 2A4. This identification must be contained in the first eight columns of the first card of the Aircraft Input Data. All subsequent data in this section can appear in any columns and on as many cards as desired. All input values must be separated from one another by at least one blank column and a value cannot be continued on the next card.

The element identification alpha array ELID is entered under the format specification 4A4. This identification must be contained in the first sixteen columns of the long list element data card. Subsequent data can appear in any column and on as many cards as desired. When a second element is to be identified, its description must again appear in the first sixteen columns of the data card.

The element identification alpha array ELEMENT is entered under the format specification 4A4. This identification must be contained in the first sixteen columns of the first card of each set of Element Input Data. As in the Aircraft Input Data, all subsequent data in this section can appear in any columns and on as many cards as desired.

SAMPLE INPUT. Sample input data consisting of aircraft input data and four sets of element input data are illustrated in Figure 2. The aircraft type identification is HYBRID. The four cards immediately following the aircraft type contain the aircraft input data. The element input data begins immediately after the element identification on the same card and terminates on the last card before the next element identification. The card following the last set of element input data must contain EOD in the first three columns. Note this input is for standard output since LONG.LIST = "NO".

Sample input data to obtain long list output on the last element (FUS-MRF-SID-1740) of the sample case are shown in Figure 3. Long list data are requested on aircrafts 148 and 162. In this case we are duplicating the output of the sample case in Figure 2 by inputting the random number seeds appropriate to element FUS-MRF-SID-1740. Note that S.OPT = "YES" and LONG.LIST = "YES" in the aircraft data in Figure 3. The ten random number seeds follow the aircraft data. Next comes the long list data and finally the element data. Element output can be duplicated on a given element without rerunning all the elements ahead of the desired element by using the random number seed option.

```

HYBRID
500 60000 150 50 100 5000 0 100 .001 .8 .8 .4 1.0 .695 1.5
25 375 .172 .868 5.97 17.34 NO NO 3 .2
0 1476 .8 .65 .25 4371 4371 +117 3101 1000 1000
1311 1311 4321 2.75 1.25
FUS-MFR-SID-0540 422910 3.87E-05 1.57E-03 72. 9.68 46.25 6.72E-05 4
9.160E-09 .66 .909 D C
NO .130 1.003E-08 1.793E-08 12000 3.25
7.86E-04 3.15E-03 38. 41.
9.68 41.25 44.25
1000. 1600. 174 1.841 -14.83
FUS-MFR-SID-0560 434010 3.77E-05 7.66E-04 72. 9.68 43.25 6.72E-05 4
8.160E-09 .1 .909 D C
NO .130 1.003E-08 1.793E-08 12000 46.25
3.85E-04 1.53E-03 18. 22.
9.68 21.25 25.25
1000. 1600. 174 1.794 -14.83
FUS-MFR-SID-1720 158730 1.03E-04 2.09E-03 72. 9.68 43.25 6.72E-05 3
8.160E-09 .10 .909 D C
NO .130 1.003E-08 1.793E-08 12000 46.25
1.05E-03 4.19E-03 18. 22.
9.68 21.25 25.25
1000. 3200. 174 4.905 -14.83
FUS-MFR-SID-1740 157620 1.04E-04 4.22E-03 72. 9.68 46.25 6.72E-05 3
8.160E-09 .66 .909 D C
NO .130 1.003E-08 1.793E-08 12000 3.25
2.11E-03 8.44E-03 38. 41.
9.68 41.25 44.25
1000. 3200. 174 4.940 -14.83
END

```

Figure 2. Sample Input Data for
Standard Output

```

HYBRID
500 60000 150 50 100 5000 0 100 .001 .8 .8 .4 1.0 .695 1.5
25 375 .172 .868 5.97 17.34 YES YES 3 .2
0 1476 .8 .65 .25 4371 4371 4117 3101 1000 1000
1311 1311 4321 2.75 1.25
128995411966065 185749178106180 173106806853940 47727199222148 68702620744893
183801640538867 93399560140797 146472757558685 146971947128576 168690610584217
1
FUS-MFR-SID-1740 2 148 162
FUS-MFR-SID-1740 157620 1.04E-04 4.22E-03 72. 9.68 46.25 6.72E-05 3
8.160E-09 .66 .909 0 C
NO .130 1.003E-08 1.793E-08 12000 3.25
2.11E-03 8.44E-03 38. 41.
9.68 41.25 44.25
1000. 3200. 174 4.940 -14.83
EOD

```

Figure 3. Sample Input Data for
Long List Output

PROGRAM OUTPUT

Each element to be simulated by SAIPE is identified by three groups of alpha characters and one group of numeric characters. The alpha characters define the basic element type and general location on the aircraft, and the numeric characters define the specific location of the element by identifying the wing or fuselage station number. For example, an element identified as WNG-STR-CEN-396 would be a wing stringer located midway between the front and rear spars and centered at wing station 396.

The standard program output consists of two parts. The first part consists of the simulation results for each specific element. This part is printed for each set of element input data. The second part consists of a summary of the first parts for an element type. In the example discussed above, WNG-STR-CEN identifies the element type. Whenever the program encounters a set of element input data in which any single character of the three groups of alpha characters differs from those in the previous set of element input data, a summary is printed.

The program also has a long list form of output. This output consists of a detailed time history of individual elements and aircraft. This form of output would generally be used after a standard output has been obtained on element data. Generally, only the elements that long list output is desired on is rerun using the seed option to input the appropriate random number seeds for the desired elements.

STANDARD OUTPUT, ELEMENT DATA. Figure 4 illustrates the standard output for the input shown in Figure 2. The random number seeds and aircraft numbers for aircraft having corrosion, production defects, and service damage are printed on the first page. The random number seeds are the initial seeds for each element. Since the initial seeds for an element are dependent on the history of the preceding element, it is necessary to know the initial element seeds, if one wanted to duplicate a particular element without rerunning all the previous elements.

On the next page, the aircraft type identification, number of aircraft in the fleet, aircraft service life, structural element identification, predicted average fatigue life, actual average fatigue life, and fatigue test life are printed at the top of the page.

The number of occurrences in the fleet and the times to initiation of the four types of aircraft defects considered by SAIPE are displayed next. Whenever there is a fatigue crack initiation and there are no other cracks in the element, a first crack is said to have occurred. A single element can experience more than one first crack in its lifetime by having a crack initiation after a repair. Similarly, there can be more than one occurrence of corrosion and service damage defect to exist simultaneously in the same element.

RANDOM NUMBER SEEDS
 SEED(1) = 08251033025441
 SEED(2) = 183503209621313
 SEED(3) = 41716298375649
 SEED(4) = 98700237180545
 SEED(5) = 45258732784161
 SEED(6) = 180800743732673
 SEED(7) = 56654308844385
 SEED(8) = 17932178784449
 SEED(9) = 159014029078777
 SEED(10) = 199600214705217

SERVICE DAMAGE AIRCRAFT NO. 303

PAGES 15-26 WILL CONTAIN COMPUTER OUTPUT
 FOR THE INPUT DATA OF FIGURE 2.

FIG. 9 - SAMPLE OUTPUT STANDARD DATA FOLLOWS

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500

AIRCRAFT SERVICE LIFE: 60000 HOURS

STRUCTURAL ELEMENT: FUS-MFR-SID-0540

PREDICTED AVERAGE FATIGUE LIFE: 422910 HOURS
FATIGUE TEST LIFE: 999999 HOURS

ACTUAL AVERAGE FATIGUE LIFE: 388715 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

OCCURRENCES	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
MIN(MRS)	1	0	1	0
MAX(MRS)	19853	0	1985	---
AVG(MRS)	19853	0	1985	---

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

OCCURRENCES	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
MIN(IN)	0	0	0	0	0
MAX(IN)	0	0	0	0	0
AVG(IN)	0	0	0	0	0

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

OCCURRENCES	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
MIN(SQ.IN)	0	0	0	0	0
MAX(SQ.IN)	0	0	0	0	0
AVG(SQ.IN)	0	0	0	0	0

INSPECTION INTERVALS(MRS)

INITIAL	MOD. NO	SAMPLING	TIME
25	0	30	2200
25	0	21	4600
25	0	15	6200
25	0	11	13600
25	0	8	21700
25	0	6	33850
25	0	7	49038
25	0	9	71819
25	0	10	86865

AIRCRAFT NO. 363
CRACK LENGTHS AND CORRESPONDING CUMULATIVE PROBABILITY OF FAILURE
FLT. HOURS 2.73

PROB. OF FAILURE
3.8E-13

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 0
NUMBER OF STRUCTURAL MODIFICATIONS: 0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 388715 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0
ESTIMATED ELEMENT FAILURE RATE: 1.24E-20/HP.

STRUCTURAL FAILURE RATE
AIRCRAFT NO. 363
FLT. HOURS

RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO. 363
FLT. HOURS

RANDOM NUMBER SEEDS
 SEED(1) = 48251033025441
 SEED(2) = 143342730965628
 SEED(3) = 26721902511699
 SEED(4) = 158438712604501
 SEED(5) = 128678155263253
 SEED(6) = 73883536770321
 SEED(7) = 124221014172629
 SEED(8) = 240304445099381
 SEED(9) = 175980516536778
 SEED(10) = 31467978049536

CORROSION AIRCRAFT . 354

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500

AIRCRAFT SERVICE LIFE: 60000 HOURS

STRUCTURAL ELEMENT: FUS-MFR-SID-0560

PREDICTED AVERAGE FATIGUE LIFE: 434010 HOURS
FATIGUE TEST LIFE: 196137 HOURS

ACTUAL AVERAGE FATIGUE LIFE: 244829 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

OCURRENCES	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
MIN(HRS)	1	1	0	0
MAX(HRS)	38206	17550	0	---
AVG(HRS)	38206	17550	0	---

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

OCURRENCES	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
MIN(IN)	0	0	0	0	0
MAX(IN)	0	0	0	0	0
AVG(IN)	0	0	0	0	0

- 18 -

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

OCURRENCES	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
MIN(SQ.IN)	0	0	0	1	0
MAX(SQ.IN)	0	0	0	24.93	0
AVG(SQ.IN)	0	0	0	24.93	0

INSPECTION INTERVALS(HRS)

INITIAL	MOD NO	SAMPLING	TIME
25	375	1000	1600
25	375	1125	2400
25	375	1250	3600
25	375	1424	5400
25	375	1692	8100
25	375	1802	12150
25	375	2253	15108
25	375	2816	18984
25	375	3528	23730
25	375	4399	29663
10			0

AIRCRAFT NO. 453
CRACK LENGTHS AND CORRESPONDING CUMULATIVE PROBABILITY OF FAILURE
CRK.LGT. .93
PROB. OF FAILURE 8.5E-14

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 0
NUMBER OF STRUCTURAL MODIFICATIONS: 0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 244829 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0
ESTIMATED ELEMENT FAILURE RATE: 2.8E-21/HR.

STRUCTURAL FAILURE RATES
AIRCRAFT NO. 453
RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO. 453

RANDOM NUMBER SEEDS
 SEED(1) = 233826060175038
 SEED(2) = 1510689237728
 SEED(3) = 132198803447566
 SEED(4) = 78276774440329
 SEED(5) = 57027878446281
 SEED(6) = 142656845460439
 SEED(7) = 208345042970025
 SEED(8) = 190428986741161
 SEED(9) = 118776196986127
 SEED(10) = 275755793609760

NON-EXPLORATORY DETECTION LEVEL AT 80113 MODIFICATION 0
 ICPH = 0. MCPH = .125 RCPH = .000 TIME = 80113
 ICPH = 0. MCPH = .129 RCPH = .001 TIME = 80714
 ICPH = 0. MCPH = .143 RCPH = .001 TIME = 82713
 CRACK FOUND ON A/C NO. 436 AT 45563 HOURS DURING INTERNAL D INSPECTION
 ICPH = .003 MCPH = .156 RCPH = .003 TIME = 84213
 CRACK FOUND ON A/C NO. 469 AT 59981 HOURS DURING INTERNAL D INSPECTION
 ICPH = .003 MCPH = 25.481 RCPH = .002 TIME = 101931

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 509

STATION 00000 12417 POLARIS LATROUITY

STRUCTURAL ELEMENT: FUS-WFR-SID-1720

PREDICTED AVERAGE FATIGUE LIFE: 150730 HOURS
 ACTUAL AVERAGE FATIGUE LIFE: 128026 HOURS
 FATIGUE TEST LIFE: 999999 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

OCURRENCES	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
21				
29896				
59551				
47694				

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

OCURRENCES	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
MIN (IN)	0	0	1	0	3
MAX (IN)	0	0	1.95	1.64	.90
AVG (IN)	0	0	1.95	2.05	1.40
	0	0	1.95	2.08	1.90

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

OCCURRENCES				INSPECTION INTERVALS (HRS)				TIME			
A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	NOO NO	SAMPLING						
INITIAL	25	1980	375	0	15	3300					
1	25	1125	375	0	11	0000					
2	25	1200	375	0	4	0000					
3	25	1455	375	0	6	1500					
4	25	1602	375	0	4	2500					
5	25	1802	375	0	6	4200					
6	25	2002	375	0	7	4200					
7	25	2503	375	0	6	67150					
8	25	2503	375	0	1	84215					

[illegible]

AIRCRAFT NO.	FLY. HOURS	CHG. LOST.	POSS. OF FAILURE
3	60000	3.45	3.75-13
16	60000	1.40	6.45-14
64	60000	1.64	7.45-14
64000	200	4.40	9.45-13
200	60000	.87	9.05-00
257	60000	1.51	1.55-13
216	60000	1.49	1.15-13
305	92563	2.49	3.15-13
291	54564	3.45	5.05-13
275	60000	3.45	9.05-14
421	62563	3.45	2.75-13
426	92563	1.45	1.05-13
342	54563	.90	2.45-14
366	52563	2.40	9.05-13
443	52563	1.41	7.15-15
297	60000	.18	7.15-15
229	60000	.30	7.15-15
376	60000	.21	2.15-16
375	54000	2.17	1.05-13
405	60000	.92	5.05-14
449	55000	2.43	1.45-13

449 59001
NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 1
NUMBER OF STRUCTURAL MODIFICATIONS: 0
NUMBER OF ACTUAL AVERAGE REPORTED FAILURES: 120024
NUMBER OF AIRCRAFT REPORTED IN SERVICE: 1
ESTIMATED ELEMENT FAILURE RATE: 1.37E-10000

**NATIONAL STRENGTH EQUALS FALL-SAFE STRENGTH
AIRCRAFT NO. FLY, HOURS**

RANDOM NUMBER SEEDS
 SEED(1) = 128995411966065
 SEED(2) = 185749178106180
 SEED(3) = 173106806853940
 SEED(4) = 47727199222148
 SEED(5) = 66702620744893
 SEED(6) = 183801640538857
 SEED(7) = 93399566148797
 SEED(8) = 14647275758685
 SEED(9) = 146971947128576
 SEED(10) = 168690610584217

NON-EXPLORATORY DETECTION LEVEL AT 37792 MODIFICATION 0
 ICPH = 0. MCPH = -.030 RCPH = .007 TIME = 37792
 ICPH = 0. MCPH = -.032 RCPH = .010 TIME = 39650
 ICPH = 0. MCPH = -.034 RCPH = .013 TIME = 40900
 ICPH = 0. MCPH = -.049 RCPH = .008 TIME = 51463
 ICPH = 0. MCPH = -.050 RCPH = .010 TIME = 51801
 ICPH = 0. MCPH = -.060 RCPH = .009 TIME = 56100
 ICPH = 0. MCPH = -.061 RCPH = .010 TIME = 56500
 ICPH = 0. MCPH = -.062 RCPH = .011 TIME = 56700
 ICPH = -.000 MCPH = -.069 RCPH = .048 TIME = 57100
 ICPH = -.000 MCPH = -.071 RCPH = .044 TIME = 59288
 ICPH = -.000 MCPH = -.072 RCPH = .041 TIME = 60853
 ICPH = -.075 MCPH = -.072 RCPH = .035 TIME = 64188
 CRACK FOUND ON A/C NO. 140 AT 55138 HOURS DURING INTERNAL D INSPECTION
 ICPH = .075 MCPH = .072 RCPH = .056 TIME = 64188
 CRACK FOUND ON A/C NO. 165 AT 53506 HOURS DURING INTERNAL D INSPECTION
 CRACK FOUND ON A/C NO. 189 AT 51196 HOURS DURING INTERNAL D INSPECTION
 CRACK FOUND ON A/C NO. 241 AT 45906 HOURS DURING INTERNAL D INSPECTION
 CRACK FOUND ON A/C NO. 272 AT 42806 HOURS DURING INTERNAL D INSPECTION
 CRACK FOUND ON A/C NO. 298 AT 40206 HOURS DURING INTERNAL C INSPECTION
 CRACK FOUND ON A/C NO. 466 AT 29406 HOURS DURING INTERNAL D INSPECTION
 CRACK FOUND ON A/C NO. 435 AT 26506 HOURS DURING INTERNAL D INSPECTION
 NON-EXPLORATORY DETECTION LEVEL AT 86389 MODIFICATION 1
 ICPH = 0. MCPH = .179 RCPH = .002 TIME = 86389

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 60000 HOURS
 STRUCTURAL ELEMENT: FUS-MFR-SID-1740
 PREDICTED AVERAGE FATIGUE LIFE: 157620 HOURS ACTUAL AVERAGE FATIGUE LIFE: 66618 HOURS
 FATIGUE TEST LIFE: 999999 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
OCCURRENCES	97	0	0	0
MIN(HRS)	9396	0	0	---
MAX(HRS)	59608	0	0	---
AVG(HRS)	41413	0	0	---

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	0	5	17	46
MIN(IN)	0.	0.	1.69	.23	.36
MAX(IN)	0.	0.	3.07	6.81	8.97
AVG(IN)	0.	0.	2.16	2.08	2.43

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	0	0	0	0
MIN(SQ.IN)	0.	0.	0.	0.	0.
MAX(SQ.IN)	0.	0.	0.	0.	0.
AVG(SQ.IN)	0.	0.	0.	0.	0.

INSPECTION INTERVALS(HRS)

INITIAL	25	375	1000	3200	0	MOD NO	SAMPLING	TIME
2	25	375	1125	4800	0	15	15	3800
3	25	375	1266	7200	0	8	8	8600
4	25	375	1424	10800	0	6	6	15800
5	25	375	1602	16200	0	5	5	26800
6	25	375	2002	20250	0	6	6	42800
7	25	375	2002	7088	0	17	17	57100
8	25	375	2002	2481	0	1	1	64188
9	25	375	2002	868	0	1	1	64188
10	25	375	2062	2481	1	1	1	65664
11	25	375	2253	3721	1	2	2	68805
12	25	375	2534	5581	1	2	2	69645

CRACK LENGTHS AND CORRESPONDING CUMULATIVE PROBABILITY OF FAILURE

AIRCRAFT NO.	FLY. HOURS	CRK.LGT.	PROB. OF FAILURE
17	36842	5.21	5.3E-13
104	34200	3.57	3.7E-13
36	39000	.23	7.1E-15
110	45413	3.07	3.3E-13
119	4851	1.93	1.3E-13
138	47250	4.07	6.2E-13
142	47250	4.06	6.9E-13
144	47250	2.54	1.9E-13
148	47250	18.73	7.3E-12
7	56650	3.08	2.4E-13
12	56400	8.45	2.5E-12
19	56050	1.91	1.5E-13
20	56000	4.29	3.7E-13
22	55900	4.76	7.4E-13
24	55800	.41	2.1E-14
39	55050	2.03	9.9E-14
48	54600	2.92	3.1E-13
52	54400	3.19	4.3E-13
70	53500	1.18	8.5E-14
75	53250	3.68	4.7E-13
88	52600	1.86	1.2E-13
89	52550	3.61	3.9E-13
106	51450	4.29	7.0E-13
118	50250	1.53	8.5E-14
128	49250	2.35	1.7E-13
133	48750	.36	2.1E-14
171	44950	.77	4.3E-14
173	44750	2.05	1.4E-13
214	40650	1.32	7.8E-14
229	39150	2.71	2.5E-13
234	38650	4.31	6.9E-13
255	36550	.45	2.1E-14
302	34032	1.69	1.4E-13
2	60000	1.16	1.1E-13
13	60000	3.49	2.5E-13
94	56063	1.08	1.6E-13
26	60000	1.90	7.8E-14
29	60000	.32	2.1E-14
31	60000	2.39	1.9E-13
32	60000	1.37	1.2E-13
37	60000	.93	5.0E-14
38	60000	4.53	3.3E-13
42	60000	4.66	7.7E-13
46	60000	.73	6.4E-14
50	60000	.42	2.1E-14
53	60000	2.58	2.1E-13
54	60000	.37	7.1E-15
55	60000	1.78	1.2E-13
57	60000	4.08	5.3E-13
60	60000	.09	0.0E+00
73	60000	.67	4.3E-14
125	56638	.28	2.1E-14
140	55138	2.12	1.4E-13

83
96
105
116
117
146
151
162
166
172
172
208
233
282
301
303
325
352
358
377
377
165
189
241
272
298
406
435
103
107
112
115
131
139
163
168
179
190
191
192
194
211
240
264
479
389

59938
59288
58638
57538
57438
54538
54038
52938
52538
51938
48338
45838
40938
39038
38838
36638
33938
33338
31438
53506
51106
45906
42806
40206
29406
26506
60000
60000
59674
59374
57774
56974
54574
54074
52974
51874
51774
51674
51474
49774
46874
44474
43439
60000

1.69
2.22
3.54
2.99
3.92
6.15
1.29
16.70
.82
3.44
1.93
2.60
2.21
1.92
1.21
1.91
3.28
.66
1.98
1.66
3.11
1.99
1.54
2.02
.74
.49
1.46
.15
.53
.14
.01
3.23
1.13
.21
.21
.15
2.03
.52
2.09
.25
1.06
.40
1.82
.67

1.1E-13
1.7E-13
4.1E-13
2.7E-13
8.5E-13
7.8E-14
7.7E-12
4.3E-14
4.1E-13
1.7E-13
2.3E-13
1.6E-13
1.8E-13
8.5E-14
3.1E-13
2.8E-14
1.7E-13
5.0E-14
2.0E-13
8.5E-14
9.9E-14
1.5E-13
4.3E-14
2.1E-14
8.5E-14
7.1E-15
2.8E-14
7.1E-15
0.0E+00
3.3E-13
7.8E-14
7.1E-15
7.1E-15
7.1E-15
1.6E-13
2.8E-14
2.1E-13
7.1E-15
6.4E-14
2.8E-14
1.4E-13
4.3E-14

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 2
NUMBER OF STRUCTURAL MODIFICATIONS: 1
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 97716 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 392
ESTIMATED ELEMENT FAILURE RATE: 1.18E-18/HR.

STRUCTURAL FAILURES
AIRCRAFT NO. -----
FLY. HOURS -----

RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO. -----
FLY. HOURS -----

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 68000 HOURS

SUMMARY OF STRUCTURAL ELEMENT: FUS-MFR-SID

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
OCCURRENCES	120	1	1	0
MIN(HRS)	9396	17550	19853	---
MAX(HRS)	59600	17550	19853	---
AVG(HRS)	42306	17550	19853	---

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	0	6	21	49
MIN(IN)	0.	0.	1.69	.23	.36
MAX(IN)	0.	0.	3.95	6.81	8.97
AVG(IN)	0.	0.	2.46	2.16	2.40

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	0	0	1	0
MIN(SQ. IN)	0.	0.	0.	24.93	0.
MAX(SQ. IN)	0.	0.	0.	24.93	0.
AVG(SQ. IN)	0.	0.	0.	24.93	0.

INSPECTION INTERVALS (HRS)

INITIAL	25
SHORTEST	25
LONGEST	25

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 3

NUMBER OF STRUCTURAL MODIFICATIONS: 1

ESTIMATED ELEMENT TYPE FAILURE RATE USING AVG: 5.78E-17/HR

SAMPLE CRK. LGT. MEAN(IN) 2.15 SAMPLE STD. DEV. 1.056

CRK. LGT. VS PROBABILITY CURVE FIT CONST: A = -13.59065629553 B = .300332229694

STRUCTURAL FAILURES

AIRCRAFT NO. STA. NO.

RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO. STA. NO.

EST. AIRCRAFT FAILURE RATE OF FLIGHT LOADED STRUCTURE USING AVG: 0.00E+00/HR.

EST. AIRCRAFT FAILURE RATE OF PRESSURE LOADED STRUCTURE USING AVG: 5.78E-17/HR.

EST. AIRCRAFT FAILURE RATE OF FLIGHT LOADED STRUCTURE: 0.00E+00/HR.

EST. AIRCRAFT FAILURE RATE OF PRESSURE LOADED STRUCTURE: 5.14E-15/HR.

AVERAGE FLIGHT CRACKS	0.	0.	0.	0.	0.
AVERAGE PRESSURE CRACKS	.232	0.	.232	.232	.232
FLIGHT CRACKS	0.	0.	0.	0.	0.
PRESSURE CRACKS	.477	.426	.400	.349	.344

FAILURE SUMMARY
NONE

END OF SIMULATION

Production defects are one-time occurrences unless there is a structural modification installed. These, too, can have production defects. The times to defect initiation are measured from the time when the aircraft enters service for the initial defects and from the time when the aircraft was last repaired for subsequent defects.

Next, the number and lengths of cracks detected in the fleet at each level of inspection are printed. These numbers include second crack detections. Following the crack detection output are the number and areas of corrosion defects detected in the fleet at each level in inspection. A history of the inspection interval changes is printed next. Each time that the aircraft service experience indicates that either an interval increase or an interval decrease is needed, the new interval values are printed. The history of the inspection intervals includes the modification in effect at a given interval, the D-level internal sampling change, and the time the interval changed. A modification number of 0 indicates the element was not modified. Although the number of interval changes allowed in the simulation is unlimited, the output array size limits the number printed to 30.

A table of crack lengths versus cumulative probability of failure is printed next. This table includes aircraft number, flight hours, crack length, and probability of failure. Flight hours equal to the service time indicate that the crack was never detected or repaired. Times other than this indicate when the crack was repaired or a modification was installed.

The number of fleet-wide special inspections performed is printed next. Next, the number of structural modifications developed is printed. This number includes modifications because of fatigue test failures or aircraft service experience. The final actual average fatigue life is printed next. If there have been no modifications, this number will be the same as that at the top of the page. If there have been modifications, this number is the actual average fatigue life of the most recently developed modification.

Shown next is the number of aircraft modified in service. If the only modification developed was due to a fatigue test failure, this number can be zero if the test life was such that retrofits were not required. If there were more than one modification requiring retrofits, this number can be greater than the size of the fleet. Printed next is the element failure rate. Finally, each time an aircraft experiences structural failure or its residual strength reaches its fail-safe strength, the aircraft number and the number of accrued flight hours are printed. The aircraft number is assigned by its relative time of entry into service. Aircraft No. 1 is the first aircraft to enter service.

STANDARD OUTPUT SUMMARY DATA. A summary is shown in Figure 4 after the element output. Number and time to initiation of aircraft defects, number and length of cracks detected at each level of inspection, number and area of corrosion defects detected at each level of inspection, initial, minimum and maximum inspection intervals, number of special inspection conducted, number of structural modifications, number of aircraft modified in service, structural failures, and fail safe lengths are summaries of the element data output. Also included is the element type failure rate. Element type failure rate can be calculated by two methods. One is by simply taking the average of the individual element failure rates and multiplying by the total element type POP.SIZE. The other method uses a log normal distribution to generate additional cracks to account for the POP.SIZE. The probability of failure from the additional cracks is obtained from a semi-log curve fit of crack length versus probability of failure for the existing input elements of the element type. The constants of the curve fit and the mean and standard deviation of the crack lengths are shown. The output of the type of probability of failure calculation is dependent on the input quantity FAIL.OPT.

After the last element summary, the aircraft failure rate is printed. The aircraft failure rate is based on pressure and flight loaded structure. Two types of failure rates can be printed, depending on the input quantity FAIL.OPT. Next the five maximum crack lengths divided by the fail-safe length are printed for flight and pressure loaded structure and the two failure options. The cracks based on averaging are weighted by the POP.SIZE divided by the total number of element types that are input. Thus, the same crack may appear more than once for the averaging option. Cracks based on the curve fit option include both the cracks that occur in the structural element and additional cracks that are generated to account for POP.SIZE. Finally, a failure summary is printed.

LONG LIST DATA. If the long list option is used, a detailed time history of elements and aircraft are printed. A long list printout for the input in Figure 3 is shown in Figure 5. Long list print information is contained in the following events and routines:

1. Event	ENTER.SERVICE
2. Routine	INSTALL.MODIFICATION
3. Event	IN.SERVICE.DAMAGE
4. Event	CORROSION
5. Event	1.STRENGTH.REDUCTION
6. Event	2.STRENGTH.REDUCTION
7. Event	1.ITE
8. Event	2.ITE
9. Event	D.LEVEL.INSPECTION
10. Routine	EXAMINE
11. Event	REACH.FAIL.SAFE.LGT
12. Event	FAILURE
13. Event	RETIRE.FROM.SERVICE
14. Event	REPAIR
15. Event	T.INSPECTION.INCREASE
16. Event	INCREASE.INSPECTION.FREQUENCY
17. Event	IMMEDIATE.FLEET.INSPECTION
18. Event	DECISION.ON.MOD
19. Event	IMPLEMENT.MODIFICATION

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500

AIRCRAFT SERVICE LIFE: 60000 HOURS

STRUCTURAL ELEMENT: FUS-MFR-SID-1740

PREDICTED AVERAGE FATIGUE LIFE: 157620 HOURS

ACTUAL AVERAGE FATIGUE LIFE: 66618 HOURS

INITIAL INSPECTION INTERVALS

A-LEVEL 25 HOURS
B-LEVEL 375 HOURS
C-LEVEL 1000 HOURS
D-LEVEL 3200 HOURS

NOTE: Pages 30-33 WILL
CONTAIN COMPUTER
OUTPUT FOR INPUT
DATA OF FIGURE 3.

INSPECTION INTERVAL INCREASE IMPLEMENTED 3800 HOURS FROM START OF SIMULATION
C-LEVEL INTERVAL NOW 1125 HOURS
D-LEVEL INTERVAL NOW 4800 HOURS
SAMPLING NOW 11

INSPECTION INTERVAL INCREASE IMPLEMENTED 8600 HOURS FROM START OF SIMULATION
C-LEVEL INTERVAL NOW 1266 HOURS
D-LEVEL INTERVAL NOW 7200 HOURS
SAMPLING NOW 8

A/C NO. 148 ENTERS SERVICE 9850 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 22042 FLIGHT HOURS
2ND CRACK INITIATION PROJECTED AT 32715 FLIGHT HOURS
SLOW CRACK GROWTH RATE1 = .000270 INCHES/HOUR
FAST CRACK GROWTH RATE1 = .010958 INCHES/HOUR
SLOW CRACK GROWTH RATE2 = .005479 INCHES/HOUR
FAST CRACK GROWTH RATE2 = .021916 INCHES/HOUR
INTERNAL SLOW CRACK GROWTH RATE1 = .000270 INCHES/HOUR
INTERNAL FAST CRACK GROWTH RATE1 = .010958 INCHES/HOUR
INTERNAL SLOW CRACK GROWTH RATE2 = .005479 INCHES/HOUR
INTERNAL FAST CRACK GROWTH RATE2 = .021916 INCHES/HOUR
CRR = 2.80 A = 1.1115

A/C NO. 162 ENTERS SERVICE 11250 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 22119 FLIGHT HOURS
2ND CRACK INITIATION PROJECTED AT 46986 FLIGHT HOURS
SLOW CRACK GROWTH RATE1 = .000291 INCHES/HOUR
FAST CRACK GROWTH RATE1 = .011810 INCHES/HOUR
SLOW CRACK GROWTH RATE2 = .005905 INCHES/HOUR
FAST CRACK GROWTH RATE2 = .023620 INCHES/HOUR
INTERNAL SLOW CRACK GROWTH RATE1 = .000291 INCHES/HOUR
INTERNAL FAST CRACK GROWTH RATE1 = .011810 INCHES/HOUR
INTERNAL SLOW CRACK GROWTH RATE2 = .005905 INCHES/HOUR
INTERNAL FAST CRACK GROWTH RATE2 = .023620 INCHES/HOUR
CRR = 2.80 A = 1.1032

INSPECTION INTERVAL INCREASE IMPLEMENTED 15800 HOURS FROM START OF SIMULATION
C-LEVEL INTERVAL NOW 1424 HOURS

FIG. 5- SAMPLE OUTPUT- LONG LIST DATA

D-LEVEL INTERVAL NOW 10800 HOURS
SAMPLING NOW 6

EXTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 148 AT 10800 HOURS

EXTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 162 AT 10800 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED 26600 HOURS FROM START OF SIMULATION

C-LEVEL INTERVAL NOW 1602 HOURS

D-LEVEL INTERVAL NOW 16200 HOURS

SAMPLING NOW 5

A/C NO. 148 EXPERIENCES 1ST CRACK INITIATION AT 22042 HOURS

CRACK INITIATES INTERNALLY

ELEMENT FAILURE PROJECTED AT 62160 FLIGHT HOURS

A/C NO. 162 EXPERIENCES 1ST CRACK INITIATION AT 22119 HOURS

ELEMENT FAILURE PROJECTED AT 59396 FLIGHT HOURS

EXTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 148 AT 27000 HOURS

ICPH = 0. MCPH = .030 RCPH = .007 TIME = 37792

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 162 AT 27000 HOURS

ICPH = 0. MCPH = .032 RCPH = .010 TIME = 39650

ICPH = 0. MCPH = .034 RCPH = .013 TIME = 40900

A/C NO. 148 EXPERIENCES 2ND CRACK INITIATION AT 32715 HOURS

CRACK INITIATES INTERNALLY

ELEMENT FAILURE PROJECTED AT 63985 FLIGHT HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED 42800 HOURS FROM START OF SIMULATION

C-LEVEL INTERVAL NOW 2002 HOURS

D-LEVEL INTERVAL NOW 20250 HOURS

SAMPLING NOW 6

A/C NO. 148 HAS INTERNAL FIRST CRACK BECOME EXTERNAL AT 3.25 INCHES AND 34077 FLIGHT HOURS

ICPH = 0. MCPH = .049 RCPH = .008 TIME = 51463

ICPH = 0. MCPH = .050 RCPH = .015 TIME = 51801

A/C NO. 148 HAS INTERNAL SECOND CRACK BECOME EXTERNAL AT 3.25 INCHES AND 44750 FLIGHT HOURS

ICPH = 0. MCPH = .060 RCPH = .009 TIME = 56100

ICPH = 0. MCPH = .061 RCPH = .010 TIME = 56500

ICPH = 0. MCPH = .061 RCPH = .011 TIME = 56700

CRACK OF LENGTH 6.81 INCHES DETECTED DURING D-LEVEL INSPECTION OF A/C NO. 148 AT 47250 FLIGHT HOURS

EXTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 148 AT 47250 HOURS

INTERVAL REDUCTION AT 57100 HRS. DUE TO 313.0 IN. POTENTIAL CRACK ON AIRCRAFT 148 EXCEEDING LARGE CRACK CRITERIA

A/C NO. 148 HAS ALL DEFECTS REPAIRED AT 47250 FLIGHT HOURS

1ST CRACK INITIATION PROJECTED AT 50819 FLIGHT HOURS

2ND CRACK INITIATION PROJECTED AT 72256 FLIGHT HOURS

FLEET WIDE SPECIAL INSPECTION PERFORMED 57100 HOURS FROM START OF SIMULATION

Fig. 5. (continued)

INSPECTION INTERVAL DECREASE IMPLEMENTED 57100 HOURS FROM START OF SIMULATION
 C-LEVEL INTERVAL NOW 2002 HOURS
 D-LEVEL INTERVAL NOW 7088 HOURS
 SAMPLING NOW 17
 ICPH = .000 MCPH = .062 RCPH = .048 TIME = 57100

A/C NO. 162 EXPERIENCES 2ND CRACK INITIATION AT 46986 HOURS
 ELEMENT FAILURE PROJECTED AT 62049 FLIGHT HOURS
 ICPH = .000 MCPH = .069 RCPH = .044 TIME = 59288
 ICPH = .000 MCPH = .071 RCPH = .041 TIME = 60863
 ICPH = .000 MCPH = .072 RCPH = .035 TIME = 64188
 INTERVAL REDUCTION AT 64188 HRS. DUE TO 1389.6 IN. POTENTIAL CRACK ON AIRCRAFT 140 EXCEEDING LARGE CRACK CRITERIA

FLEET WIDE SPECIAL INSPECTION PERFORMED 64188 HOURS FROM START OF SIMULATION

CRACK OF LENGTH 8.97 INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. 162 AT 52938 FLIGHT HOURS

CRACK OF LENGTH 1.73 INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. 162 AT 52938 FLIGHT HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED 64188 HOURS FROM START OF SIMULATION
 C-LEVEL INTERVAL NOW 2002 HOURS
 D-LEVEL INTERVAL NOW 2481 HOURS
 SAMPLING NOW 1

INTERVAL REDUCTION AT 64188 HRS. DUE TO 31.6 IN. POTENTIAL CRACK ON AIRCRAFT 162 EXCEEDING LARGE CRACK CRITERIA

A/C NO. 162 HAS ALL DEFECTS REPAIRED AT 52938 FLIGHT HOURS
 1ST CRACK INITIATION PROJECTED AT 31198 FLIGHT HOURS
 2ND CRACK INITIATION PROJECTED AT 57816 FLIGHT HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED 64188 HOURS FROM START OF SIMULATION
 C-LEVEL INTERVAL NOW 2002 HOURS
 D-LEVEL INTERVAL NOW 868 HOURS
 SAMPLING NOW 1
 ICPH = .075 MCPH = .072 RCPH = .056 TIME = 64188

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 148 AT 55206 HOURS

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 162 AT 53806 HOURS

65664 HOURS FROM START OF SIMULATION
 INSPECTION INTERVAL RETURN TO PRIOR INTERVAL DUE TO SERVICE MOD
 C-LEVEL INTERVAL NOW 2002 HOURS
 D-LEVEL INTERVAL NOW 2481 HOURS
 SAMPLING NOW 1
 AVG.FAT.LIFE = 97716

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 148 AT 56674 HOURS

MODIFICATION INSTALLED ON A/C NO. 148 AT 56074 FLIGHT HOURS
 1ST CRACK INITIATION PROJECTED AT 92087 FLIGHT HOURS
 2ND CRACK INITIATION PROJECTED AT 98740 FLIGHT HOURS
 SLOW CRACK GROWTH RATE1 = .000266 INCHES/HOUR
 FAST CRACK GROWTH RATE1 = .010773 INCHES/HOUR
 SLOW CRACK GROWTH RATE2 = .005387 INCHES/HOUR
 FAST CRACK GROWTH RATE2 = .021547 INCHES/HOUR

FIG. 5 (CONTINUED)

INTERNAL SLOW CRACK GROWTH RATE1 = .000266 INCHES/HOUR
 INTERNAL FAST CRACK GROWTH RATE1 = .010773 INCHES/HOUR
 INTERNAL SLOW CRACK GROWTH RATE2 = .005387 INCHES/HOUR
 INTERNAL FAST CRACK GROWTH RATE2 = .021547 INCHES/HOUR

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 162 AT 54674 HOURS

MODIFICATION INSTALLED ON A/C NO. 162 AT 54674 FLIGHT HOURS
 1ST CRACK INITIATION PROJECTED AT 81036 FLIGHT HOURS
 2ND CRACK INITIATION PROJECTED AT 124085 FLIGHT HOURS
 SLOW CRACK GROWTH RATE1 = .000194 INCHES/HOUR
 FAST CRACK GROWTH RATE1 = .007873 INCHES/HOUR
 SLOW CRACK GROWTH RATE2 = .003936 INCHES/HOUR
 FAST CRACK GROWTH RATE2 = .015746 INCHES/HOUR
 INTERNAL SLOW CRACK GROWTH RATE1 = .000194 INCHES/HOUR
 INTERNAL FAST CRACK GROWTH RATE1 = .007873 INCHES/HOUR
 INTERNAL SLOW CRACK GROWTH RATE2 = .003936 INCHES/HOUR
 INTERNAL FAST CRACK GROWTH RATE2 = .015746 INCHES/HOUR

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 148 AT 58555 HOURS

INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. 162 AT 57155 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED 68405 HOURS FROM START OF SIMULATION
 C-LEVEL INTERVAL NOW 2253 HOURS
 D-LEVEL INTERVAL NOW 3721 HOURS
 SAMPLING NOW 2

INSPECTION INTERVAL INCREASE IMPLEMENTED 69645 HOURS FROM START OF SIMULATION
 C-LEVEL INTERVAL NOW 2534 HOURS
 D-LEVEL INTERVAL NOW 5581 HOURS
 SAMPLING NOW 2

A/C NO. 148 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

A/C NO. 162 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

ICPH = 0. MCPH = .179 RCPH = .002 TIME = 86389

FIG. 5 - (CONTINUED)

APPENDIX A

DETAILED PROGRAM DESCRIPTION

In the following detailed description of the SAIFE program, each event and routine in the program is presented separately. Each presentation consists of a description, the definition of the local variables, if any, and major logic steps including subroutine calls and event scheduling as necessary. All events, subroutines, and variables are capitalized.

APPENDIX A

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1.	PREAMBLE	36
2.	MAIN	46
3.	INITIALIZATION	47
4.	SUM. INITIALIZE	48
5.	COMP. RISK	49
6.	ESTIMATE. FAILURE. RATE	51
7.	REAL. LIFE	52
8.	ENTER. SERVICE	53
9.	FATIGUE. LIFE. SCATTER	55
10.	INSTALL. MODIFICATION	56
11.	IN. SERVICE. DAMAGE	57
12.	T. IMPLEMENT. MOD	58
13.	PREEDICT. SERVICE. DAMAGE	59
14.	PREDICT. CORROSION	60
15.	CORROSION	61
16.	RATE	62
17.	1. STRENGTH. REDUCTION	63
18.	2. STRENGTH. REDUCTION	65
19.	1. ITE	67
20.	2. ITE	68
21.	INSPECTION. SCHEDULER	69
22.	A. LEVEL. INSPECTION	70
23.	B. LEVEL. INSPECTION	71
24.	C. LEVEL. INSPECTION	72
25.	D. LEVEL. INSPECTION	73
26.	EXAMINE	75
27.	PODD	77
28.	CANCEL. SCHEDULED. INSPECTIONS	78
29.	REACH. FAIL. SAFE. LGT	79
30.	FAILURE	80
31.	RETIRE. FROM. SERVICE	81
32.	REPAIR	82
33.	T. INSPECTION. INCREASE	84
34.	INCREASE. INSPECTION. FREQUENCY	85
35.	IMMEDIATE. FLEET. INSPECTION	86
36.	DECISION. ON. MOD	88
37.	IMPLEMENT. MODIFICATION	89
38.	DISPLAY. OUTPUT	90
39.	SUMMARY	91

PREAMBLE

DESCRIPTION - The PREAMBLE is the definition section of a SIMSCRIPT program. All global variables and global arrays are defined. Temporary entities are defined and tally statistics are identified. Event notices and functions are defined and an event priority order is set. Global variables which are input variables are not included here but can be found in the input section.

Global Real Variables

FLEET.STR.RED - This variable is the sum of crack lengths found in the fleet since the last inspection frequency change.

1AAFL - This variable is the actual average fatigue life of the element design determined in the MAIN program.

CRRF - Assigned a value in routine INITIALIZATION according to the element corrosion resistance rating, this variable is multiplied by the aircraft corrosion growth rate to give the element corrosion growth rate.

COST.OF.REPAIRS - This variable is the sum of repair costs for the fleet since the last modification.

FIXIT.COST - This variable is the cost of repairing a defect found at a particular inspection level. Its value is set in the inspection events.

CHG.FREQ.TIME - This variable is set equal to TIME.V whenever an inspection interval change is scheduled in the event REPAIR.

1CRKT - Each time a first crack occurs, this variable is set equal to the service time on the aircraft.

1CORT - Each time corrosion occurs, this variable is set equal to the service time on the aircraft.

1SDT - Each time service damage occurs, this variable is set equal to the service time on the aircraft.

ACRKL, BCRKL, CCRKL, DCRKL, SCRKL - Each time a crack is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the crack length.

ACA, BCA, CCA, DCA, SCA - Each time corrosion is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the corrosion area.

AIRFRAME.TIME - This variable is the number of flight hours accumulated since the last modification for aircraft no longer in service.

GICRK - Each time a first crack occurs, this variable is set equal to the service time on the aircraft.

GICOR - Each time corrosion occurs, this variable is set equal to the service time on the aircraft.

GLSD - Each time service damage occurs, this variable is set equal to the service time on the aircraft.

GACRK, GBCRK, GCCRK, GDCRK, GSCRK - Each time a crack is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the crack length.

GACA, GBCA, GCCA, GDCA, GSCA - Each time corrosion is found during an A-level, B-level, C-level, D-level, or special inspection, the corresponding variable is set equal to the corrosion area.

CINSL, DINSL - Each time there is an inspection interval change, these variables are set equal to the C-level and D-level intervals, respectively.

ABC.OLD - This variable is used to store the value of the C-level inspection interval prior to a interval reduction.

ABCD.OLD.D - This variable is used to store the value of the D-level inspection interval prior to an interval reduction.

SAMP.SIZE - This variable is the number of elements of an element type that is input.

LAMDA - This variable is used to store intermediate data in calculating the aircraft failure rate.

ELTYP.FAIL.RATE - This variable is the element type failure rate.

AC.FAIL.RATE - This variable is the aircraft failure rate due to flight loads.

AP.FAIL.RATE - This variable is the aircraft failure rate due to pressure loads.

SDIF - This variable is used to store element residual strength.

RLGT - This variable is used to store crack lengths in calculating element residual strength reduction rates.

AEXP - This variable is the result of fitting a curve $AEXP * EXP(BA * CL)$ to crack length versus probability of failure.

BA - This variable is the result of fitting a curve $AEXP * EXP(BA * CL)$ to the crack length versus probability of failure.

AVGL - This variable is the average of element type crack lengths.

STD - This variable is the standard deviation of element type crack lengths.

The global integer variables are listed next. Again, input variables are not included in this list.

Global Integer Variables

ID - In each event and routine, this variable is the identification number of the aircraft being processed.

IDCK - This variable is initialized to zero and incremented by one each time an aircraft enters service.

I - This variable is used as a local index or array subscript in different locations in the program.

COUNT.ELEMENT - Each time new element data is read in, this variable is incremented by one.

NICHG - This variable is the number of times that the inspection intervals have changed.

LHTA - This variable is the identification number of the aircraft among the ten high-time aircraft with the fewest flight hours.

1.NUM.OF.RETIRE - This variable is the number of aircraft that have been retired from service.

2.NUM.OF.CRASH - This variable is the number of aircraft which have been removed from service because of structural failure.

ITRNL - This variable is the numeric identification of the lowest internal level of inspection.

EXT.INSPECTION.LEVEL - This variable is the numeric identification of the lowest external level of inspection.

LIL, LEL - These variables are the numeric identifications of the lowest internal and external levels of inspection, respectively. If either of these variables is less than three, it is set equal to three.

TO.BE.MODIFIED - This variable is the number of aircraft with a pending retrofit modification.

BEEN.MODIFIED - This variable is the number of aircraft that have had a current retrofit modification installed.

FDCK - This variable is the number of aircraft in service when a modification is implemented because of a fatigue test failure.

OICR, OCOR, OSDM, OPD - These variables are the number of occurrences of first cracks, corrosion, service damage, and production defects, respectively, for a particular element.

OSCR, OSCO - These variables are the number of cracks and corrosion defects, respectively, detected during a special inspection for a particular element.

NSIC - This variable is the number of special inspections conducted for a particular element.

NSMD - This variable is the number of aircraft modified in service for a particular element.

NSFL - This variable is the number of aircraft experiencing structural failure for a particular element.

NMD - This variable is the number of structural modifications made on a particular element.

NRFS - This variable is the number of aircraft with the residual strength for a particular element reaching the fail-safe strength.

SNRFS - This variable is the number of aircraft with the residual strength for a particular element type reaching the fail-safe strength.

J - This variable is used as a local index or array subscript in different locations in the program.

LDX - If the long list option is in effect, this variable is the ascending sequential position of the element being processed among those elements read in under the long list option.

GOICR, GOCR, GOSDM - These variables are the number of occurrences of first cracks, corrosion damage, and service damage, respectively, for a particular element type.

GOSCR, GOSCO - These variables are the number of cracks and corrosion defects, respectively, detected during a special inspection for a particular element type.

GOPD - This variable is the number of occurrences of production defects for a particular element type.

SNSIC - This variable is the number of special inspections conducted for a particular element type.

SNMD - This variable is the number of structural modifications made on a particular element type.

SNSMD - This variable is the number of aircraft modified in service for a particular element type.

SNSFL - This variable is the number of aircraft experiencing structural failure for a particular element type.

CR.CTR - This variable counts the number of detected cracks.

OLD.SAMP - This variable stores the value of D-level internal sampling prior to an interval reduction.

MOD.NO - This variable counts the number of structural modifications for each element.

NCZ - This variable is used as a counter in calculating the probability of failure.

NCZO - This variable is used as a counter in calculating the probability of failure.

NTIME - This variable is the number of times an inspection interval change occurs.

SAMPLING - This variable is the sampling at the internal D-level inspection. For a sampling of 4, aircrafts 1, 5, 9 ... receive internal inspections on their first D-level inspection. Aircrafts 2, 6, 10... receive internal inspections on their second D-level inspection. Sampling is initialized to 48000 divided by 1ABCD(4).

Alpha Variables

YES.FAILURE - This variable is set to YES if there is a fatigue test failure.

PREVIOUSLY.MODIFIED - This variable is set to YES if an element has been modified.

IMP.MOD.SCH - This variable is set to YES if the event IMPLEMENT.MODIFICATION is scheduled.

LTHO - This variable is set to YES if LONG.LIST is input as YES.

DEC.ON.MOD.SCH - This variable is set to YES if event DECISION.ON.MOD is scheduled.

SEL1, SEL2, SEL3 - These variables contain parts of the element name and are used to distinguish between pressure loaded and flight loaded structures.

SM.CRK - This variable is set equal to NO if a large crack occurs and YES if a small crack occurs.

PRIOR.CR - This variable is set equal to YES when more than one crack occurs in an element.

DEC.INT - This variable is set equal to YES when an interval decrease is scheduled.

D.INT.FIND - This variable is set equal to YES if a crack is found in an interval D-level inspection.

DONE - This variable is set equal to YES when the inspection interval is first reduced.

IFLAG - This variable is set equal to YES when the inspection interval is reduced due to a fatigue test failure.

Real Arrays

C.INTERVAL, D.INTERVAL - The elements of these arrays are the current C-level and D-level inspection intervals for each aircraft in the fleet.

ABCD - This array is of size four and contains the most recent intervals for each of the four levels of inspection.

CKREP.TIME - This array is the simulation time of the most recent crack repair for each aircraft.

COREP.TIME - This array is the simulation time of the most recent corrosion repair for each aircraft.

LAST.SD - This array is the simulation time of the most recent occurrence of service damage for each aircraft.

OCCUR.MOD - This array is the simulation time when the most recent modification was installed for each aircraft.

SC, SD - These arrays contain each of the inspection interval changes for the C-level and D-level, respectively.

XRN - This array is the random number selected to calculate the time until structural failure for each aircraft.

MRDD - This array is the simulation time of the most recently detected defect at all inspection levels for each of the ten-high aircraft.

CGRI - This array is the corrosion multiplying factor for each aircraft.

MSR1, MFR1, MSR2, MFR2, MSR1.INT, MFR1.INT, MSR2.INT, MFR2.INT - These variables are the individual element crack growth rates. There are four external rates and four internal rates.

INT.LGT - Elements of this array are set equal to L.EXT for aircraft whose cracks originate internally. They remain zero for aircraft whose cracks originate externally.

SRRATE - This array contains the aircraft strength reduction rates.

T.LAST.D - This array contains the last D-level inspection interval simulation time.

TRCHG - This array contains times used in the calculation of probability of failure.

CLGT - This array contains the length of all element type cracks.

AFACT - This array contains a loading spectrum value drawn from a log normal distribution of AMEAN and divided by AMEAN for each aircraft.

CHG.TIME - This array records the times changes are made in the inspection intervals.

FSAVE - This array is used to save the five largest cracks on flight loaded structure based on failure option 2.

PSAVE - This array is used to save the five largest cracks on pressure loaded structure based on failure option 2.

AVFSA - This array is used to save the five largest cracks on flight loaded structure based on failure option 1.

AVPSA - This array is used to save the five largest cracks on pressure loaded structure based on failure option 1.

The following are the integer arrays. Again, unless otherwise noted, all arrays are 1-dimensional and input arrays are not included.

Integer Arrays

A1SR, A2SR - These arrays contain the event notice identification numbers for each aircraft for the events 1.STRENGTH.REDUCTION, and 2.STRENGTH.REDUCTION, respectively.

AF - This array is the event notice identification number for each aircraft for event FAILURE.

AIRPLANE - This array is the temporary entity identification number for each aircraft.

AAL, ABL, ACL, ADL - These arrays are the event notice identification numbers for each aircraft for the events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION, respectively.

AC, ATII - These arrays are the event notice identification numbers for each aircraft for events COROSION and T.INSPECTION.INCREASE, respectively.

ACID - This array contains the identification numbers of those aircraft experiencing structural failure for a particular element.

OICR, OICO - These arrays are the number of cracks and corrosion defects, respectively, detected at each of the four levels of inspection for a particular element.

SACID - This array contains the identification numbers of those aircraft experiencing structural failure for a particular element type.

GOICR, GOICO - These arrays are the number of cracks and corrosion defects, respectively, detected at each of the four levels of inspection for a particular element type.

HI.TIME.ACRFT - This array contains the identification numbers of the ten high-time aircraft.

APID - This array contains the identification numbers of those aircraft with a particular element whose residual strength has reached the fail-safe strength.

SAPID - This array contains the identification numbers of those aircraft with a particular element type whose residual strength has reached the fail-safe strength.

STIM - This array contains the flight hours on each aircraft when the residual strength for a particular element reaches the fail-safe strength.

SSTIM - This array contains the flight hours on each aircraft when the residual strength for a particular element type reaches the fail-safe strength.

FLTHR - This array contains the flight hours on each aircraft when structural failure occurs for a particular element.

SFLTHR - This array contains the flight hours on each aircraft when structural failure occurs for a particular element type.

ARFSL - This array is the event notice identification number for each aircraft for event REACH.FAIL.SAFE.LGT.

A1E, A2E - These arrays are the event notice identification numbers for each aircraft for events 1.ITE, 2.ITE, respectively.

D.IN - This array is used in determining which aircraft receive an internal D-level inspection.

SAMP - This array contains the sampling used at different inspection intervals.

MOD.CTR - This array counts the number of modifications for each aircraft.

MOD.SAVE - This array contains the modification at each different inspection interval.

PFID - This array contains the aircraft number for aircraft that have cracks.

PFTIM - This array contains the time on the aircraft for cracks used to calculate probability of failure.

The global alpha arrays are listed next. As before, input arrays are not included in this list.

Global Alpha Arrays

1.CR.EXISTS, 2.CR.EXISTS - The elements of these arrays are set equal to "YES" for each aircraft whenever there is a first and second initiation, respectively.

CO.EXISTS - This array is set equal to "YES" for each aircraft when it has corrosion initiation.

SD.SCH - This array is set equal to "YES" for each aircraft that has event IN.SERVICE.DAMAGE scheduled.

SSTAN - This array is the station number which identifies each aircraft experiencing the failure of a particular element type.

SELNB - This array is the station number which identifies each aircraft with a particular element type whose residual strength has reached the fail-safe strength.

AIL, FSH - This array is set equal to "YES" for each aircraft when events REACH.FAIL.SAFE.LGT and FAILURE, respectively, are scheduled.

IE1, IE2 - This array is set equal to "YES" for each aircraft that has events 1.ITE and 2.ITE, respectively, scheduled.

TMOD.PENDING - This array is set equal to "YES" for each aircraft that has a modification pending because of a fatigue test failure.

SMOD.PENDING - This array is set equal to "YES" for each aircraft that has a modification pending because of service experience.

INSP.SCH - This array is set equal to "YES" for each aircraft that has inspections below the overhaul level scheduled.

1.INT, 2.INT - These arrays are set equal to "YES" for each aircraft that has a first crack or second crack, respectively, initiated internally.

C.INT - This array is set equal to "YES" for each aircraft that has corrosion initiated internally.

D.EXT - The appropriate element of this array is set equal to YES when an aircraft receives a D-level external inspection.

E1, E2, E3, E4 - These arrays are used to store the names of structural elements that have aircraft failures.

The temporary entity definitions and tally statements are self-explanatory. The events, functions, and routines are described in detail in the following sections.

MAIN

DESCRIPTION - In the MAIN program, space is reserved for all global arrays. The following operations are performed in the order given: all input data are read in; the actual average fatigue life of the element is calculated; the fatigue test life is calculated; the necessity of a structural modification because of a fatigue test failure is determined; the element type failure rate is calculated; the first event ENTER.SERVICE is scheduled; and the simulation is initiated.

Local Variables

FIRST.LIFE - This real variable is the fatigue test life in flight hours.

NFTS - This real variable is the time in flight hours from which the second production rate goes into effect to when the last aircraft enters service.

PE - This real variable is used to store intermediate values to calculate probability of failure.

PROB - This real variable is used to store intermediate values to calculate probability of failure.

SAIL - This real variable is the earliest simulation time at which a structural modification because of a fatigue test failure is ready for installation.

MAJOR LOGIC STEPS

1. Reserve aircraft arrays.
2. Read input data.
3. Call SUM.INITIALIZE.
4. Call INITIALIZATION.
5. If new element type data:
 - a. Call ESTIMATE.FAILURE.RATE.
 - b. Call SUMMARY.
6. If actual average fatigue life is 0: (Call REAL.LIFE).
7. Calculate fatigue test life (Call FATIGUE.LIFE.SCATTER).
8. If fatigue test failure (Schedule T.IMPLEMENT.MOD).
9. Start simulation.
10. Call DISPLAY.OUTPUT.
11. Read new element data.
12. If EOI STOP.

ROUTINE INITIALIZATION

DESCRIPTION - This routine is called immediately after reading each new set of element input by the MAIN program. This routine changes the inspection level codes to numeric values, sets the corrosion growth multiplying factor based on the corrosion resistance rating, and resets the tally counters. It also initializes all the element global variables which are not part of the input.

ROUTINE SUM.INITIALIZE

DESCRIPTION - This routine is called from the MAIN program each time a new element type is read in. The element type is identified by the first twelve characters of the element identification. This routine initializes the global variables and resets the tally counters.

ROUTINE COMP.RISK

DESCRIPTION - This routine determines the probability of failure due to a first or second crack. Failure due to a first crack is based on a five part strength reduction curve and failure due to a second crack is based on a nine part strength reduction curve. If a second crack exists the length relationships of the first and second crack must be determined. These are dependent on the three critical crack lengths and the four crack growth rates. The logic for testing all the combinations of first crack critical lengths to second crack critical lengths is lengthy. This routine is called from routine INSTALL.MODIFICATION and events CORROSION, 2.STRENGTH.REDUCTION, REACH.FAIL.SAFE.LGT, FAILURE, RETIRE.FROM.SERVICE and REPAIR.

Local Variables

A - This real variable is obtained for each aircraft from a log-normal distribution defined by AMEAN.

AZRD - This real variable is used to store intermediate values in calculating the probability of failure.

CCL1, CCL2, CCL3 - These real variables are the first, second, and third critical crack lengths.

CL - This real variable is the sum of the first and second crack lengths.

CL1, CL2 - These real variables are the first and second crack lengths.

DLC1, DL1, DL2 - These real variables are used to store intermediate values in calculating first and second crack length.

K14, K15, K16, K17, K18, K19, K20, K21, K22, K23, K24, K25, K26, K27 - These real variables are used to store intermediate results in calculating the probability of failure.

RS - This real variable is the residual strength of an element.

SR, SR1, SR2, SR3, SR4, SR5, SR6, SR7 - These real variables are strength reduction rates.

S1, S2, S3, S4, S5, S6 - These real variables are strength reductions at the three critical crack lengths for the two cracks.

TAC - This real variable is the simulation time to corrosion initiation.

TA1 - This real variable is the simulation time to the first crack initiation.

TA2 - This real variable is the simulation time to the second crack initiation.

T, T1, T11, T12, T2, T21, T22, T23, T3, T4, T5, T6 - These real variables are intermediate time variables used in calculating the probability of failure.

ROUTINE ESTIMATE.FAILURE.RATE

DESCRIPTION - This routine determines the failure rate of an element type. This routine fits a semi-log curve to crack length versus probability of failure for all cracks that occur in the individual elements of the element type. If a sample of elements is used, the routine determines the average number of cracks per element and extrapolates the total number of cracks to the actual element population. The additional cracks are generated with a log-normal distribution using the mean and standard deviation of the cracks in the element sample. The probability of failure of the additional cracks is determined from the curve fit of crack length versus probability of failure. The element type failure rate is then determined and returned to the calling program. In addition, the five largest cracks in terms of fail-safe length are determined and saved each time the routine is called. Cracks are saved in four categories as described in the discussion on program output. This routine is called from the MAIN program.

Local Variables

C - This real variable is used to count the number of cracks with a probability of failure equal to zero.

CL - This real variable is the crack length.

LYS, XLYS - These real variables are used as intermediate storage in the curve fit procedure.

NUM.CRKS - This real variable is the additional cracks generated in the routine to account for the POP.SIZE of the element type.

PE,PF,PROB,DET - These real variables are used as intermediate storage in calculating the element type failure rate.

SSQL - This real variable is the sum of the squares of the crack lengths.

SUML - This real variable is the sum of the crack lengths.

I - This real variable is the fleet service life.

I, IK, J, K, KI, KK, NN, TEMP - These variables are used in determining the five largest cracks to save.

ROUTINE REAL.LIFE

DESCRIPTION - This routine accepts (1) the predicted average fatigue life of an element (2) the mean and standard deviation of the ratio of the actual average fatigue life to the predicted average fatigue life. A random selection is made from a log-normal distribution and multiplied by the predicted average fatigue life. The ratio of the actual element fatigue life to the predicted fatigue life is limited from .1 to 10.0. The resulting actual average fatigue life is returned to the calling program. REAL.LIFE can be called from the MAIN program and events IMPLEMENT.MODIFICATION and T.IMPLEMENT.MOD.

Local Variables

MEAN - This real variable, whose value is passed from the calling routine, is the mean of the ratio distribution.

RATIO - This real variable, determined to be log-normally distributed, is the ratio of the actual fatigue life of an element design to its predicted fatigue life.

STD.DEV - This real variable, whose value is passed to the calling routine, is the standard deviation of the ratio distribution.

PDL - This real variable is the design predicted average fatigue life passed from the calling routine.

RFL - This real variable is the element actual average fatigue life which is returned to the calling routine.

EVENT ENTER.SERVICE

DESCRIPTION - This event represents the entry into service of a new aircraft. The temporary entity AIRCRAFT is created and identified by the variable AIRPLANE (ID). The entity attributes TAIL.ID and ENTRY.TIME are defined and the AIRPLANE is filed in the set ACTIVE.FLEET. AFACT is determined for each aircraft from a log-normal distribution using input AMEAN. The routine FATIGUE.LIFE.SCATTER is called to determine the times to first and second crack initiation. Element external and internal crack growth rates are calculated. The times to corrosion initiation and service damage are calculated. If either of these times is less than the service life of the aircraft, the corresponding defect is scheduled. The probability of a production defect is determined. If there is a production defect, the time to first crack initiation is replaced by a time drawn from a distribution of times to crack initiation of aircraft with production defects. Crack initiations, D-level inspection, and retirement from service are also scheduled. If the present aircraft is not the last aircraft in the fleet, another ENTER.SERVICE is scheduled. This event is scheduled in the MAIN program and within itself.

Local Variables

ASD - This real variable is the standard deviation equal to $.15 * \text{AMEAN}$ used to determine AFACT.

DEFECT.LIFE - This real variable is the time to first crack initiation when the aircraft has a production defect.

FIRST.LIFE - This real variable is the time to first crack initiation when the aircraft has no production defect.

HOURS.TO.CORROSION - This real variable is the time to corrosion initiation.

INT.FAST1 - This real variable is the standard deviation of the second internal crack growth rate distribution.

INT.FAST2 - This real variable is the standard deviation of the fourth internal crack growth rate distribution.

INT.SLOW1 - This real variable is the standard deviation of the first internal crack growth rate distribution.

INT.SLOW2 - This real variable is the standard deviation of the third internal crack growth rate distribution.

OURS.TO.SERVICE.DAMAGE - This real variable is the time to service damage occurrence.

RN - This real variable is a uniformly distributed random number between zero and one.

STD.FAST1 - This real variable is the standard deviation of the second external crack growth rate distribution.

STD.FAST2 - This real variable is the standard deviation of the fourth external crack growth rate distribution.

STD.SLOW1 - This real variable is the standard deviation of the first external crack growth rate distribution.

STD.SLOW2 - This real variable is the standard deviation of the third external crack growth rate distribution.

MAJOR LOGIC STEPS

1. Create AIRCRAFT
2. Calculate times to first and second crack initiation (Call FATIGUE.LIFE.SCATTER)
3. Calculate external and internal crack growth rates (Call RATE eight times).
4. Calculate corrosion time (Call PREEDICT.CORROSION).
5. Calculate service damage time (Call PREEDICT.SERVICE.DAMAGE).
6. If corrosion occurs in aircraft service life (Schedule CORROSION).
7. If service change occurs in aircraft service life (Schedule IN.SERVICE.DAMAGE).
8. Determine if production defect occurs.
9. If first crack occurs in aircraft service life (Schedule 1.STRENGTH.REDUCTION).
10. If second crack occurs in aircraft service life (Schedule 2.STRENGTH.REDUCTION).
11. Schedule D.LEVEL.INSPECTION.
12. Schedule RETIRE.FROM.SERVICE.
13. Schedule ENTER.SERVICE for next aircraft if not last aircraft.

ROUTINE FATIGUE.LIFE.SCATTER

DESCRIPTION - This routine receives the actual average fatigue life of the element design and calculates the fatigue test life and the time to crack initiation of the two fatigue cracks. These times are random selections from a two-parameter Weibull distribution. If the routine is not being used to calculate fatigue test life, then crack initiation times are adjusted by dividing them by AFACT. This routine can be called from the MAIN program, INSTALL.MODIFICATION, ENTER.SERVICE, and REPAIR.

Local Variables

ALPHA - This real variable is the shape parameter of the fatigue life distribution.

BETA - This real variable is the scale parameter of the fatigue life distribution.

FIRST.LIFE - This real variable is the fatigue test life or time to first crack initiation.

LIFE - This real array of length two is used to temporarily store the times to crack initiation.

N - This integer variable, passed from the calling routine, identifies the random number stream to be used.

RFL - This real variable is the element actual average fatigue life passed from the calling routine.

RN - This real variable is a uniformly distributed random number.

SECOND LIFE - This real variable is the time to second crack initiation.

ROUTINE. INSTALL. MODIFICATION

DESCRIPTION - This routine represents the installation of a structural modification caused by a fatigue test failure or by aircraft service experience. The modification is installed during a repair or a D-level inspection. All previously scheduled defect initiations are cancelled, and new times to defect initiations are calculated for each aircraft when it is modified. This routine can be called from the event REPAIR and D.LEVEL.INSPECTION.

Local Variables (Same as ENTER.SERVICE except no variable ASD).

MAJOR LOGIC STEPS

1. Call COMP.RISK if crack exists.
2. If inspection scheduled (Call CANCEL.SCHEDULED.INSPECTIONS).
3. Cancel FAILURE, CORROSION, 1.STRENGTH.REDUCTION, 2.STRENGTH.REDUCTION, 1.ITE, 2.ITE, and REACH.FAIL.SAFE.LGT if scheduled.
4. Do steps 2-10 in ENTER.SERVICE.

EVENT IN.SERVICE.DAMAGE

DESCRIPTION - This event represents the occurrence of a service damage defect. This occurrence results in the immediate initiation of the next scheduled crack. A new time to service damage is determined. If the new time is less than the remaining aircraft service time, this event is scheduled once again. This event can be scheduled from within itself or in event ENTER.SERVICE.

Local Variables

IDSDM - This integer variable is the identification number of the aircraft from which the event was scheduled.

OURS.TO.SERVICE.DAMAGE - This real variable is the time to the next occurrence of service damage.

RST - This real variable is the remaining service time to the retirement of the aircraft being considered.

EVENT T.IMPLEMENT.MOD

DESCRIPTION - This event represents the development of a structural modification because of a fatigue test failure. A new predicted life for the modification is determined by using the old predicted life or twice the actual average fatigue life, whichever is greater. The actual average fatigue life of the modification is determined by calling REAL.LIFE. If the modification is fatigue tested and the actual average fatigue life is less than the predicted life, the actual average fatigue life is set equal to the predicted life. If the fatigue test failure occurs before one service life, a retrofit modification is indicated and an inspection increase is scheduled for active aircraft. This event is scheduled in the MAIN program.

Local Variables

NEW.LIFE - This real variable is the predicted life of the modification.

NMU - This real variable is $MU.R + .15 (1.-MU.R)$ and is based on the assumption that a modification usually improves the actual average fatigue life of a particular design.

NSIG - This real variable is $.85(SIG.R)$ and is based on the same assumption as for NMU.

ROUTINE PREEDICT.SERVICE.DAMAGE

DESCRIPTION - This routine generates the time to service damage occurrence for a given aircraft from a constant service damage occurrence rate. If the service damage occurrence rate is zero in the input, the routine sets the time to service damage to twice the aircraft service life. This routine can be called from events ENTER.SERVICE and IN.SERVICE.DAMAGE.

Local Variables

OURS.TO.SERVICE.DAMAGE - This real variable is the time to service damage in flight hours.

RN - This real variable is a uniformly distributed random number between zero and one.

ROUTINE PREDICT.CORROSION

DESCRIPTION - This routine generates time to corrosion initiation for a given aircraft from a time-dependent occurrence rate approximated by two constant rates. The first constant occurrence rate, the second constant occurrence rate, and the service time on the aircraft when the second rate goes into effect are all input variables. This routine can be called from the routine INSTALL.MODIFICATION and events ENTER.SERVICE and REPAIR.

Local Variables

CRCT - This real variable is the remaining time in flight hours until the second corrosion occurrence rate goes into effect. This variable can be negative indicating that the second rate is already in effect.

HOURS.TO.CORROSION - This real variable is the flight time until corrosion initiation.

LP - This real variable is used to hold an intermediate value during the calculation of time to corrosion initiation.

RN - This real variable is a uniformly distributed random number between zero and one.

EVENT CORROSION

DESCRIPTION - This event represents the initiation of a corrosion defect. A probabilistic determination is made of whether corrosion occurs externally or internally and if it occurs at a stress concentration. A corrosion damage factor is calculated depending on whether corrosion occurs in a stress concentration. A multiplying factor for crack growth rates is set equal to the numeric value for the corrosion resistance rating. If either of the events FAILURE or REACH.FAIL.SAFE.LGT is scheduled, its remaining time to occurrence is reduced by the corrosion resistance rating. The remaining time to crack initiation is reduced by the corrosion damage factor. This event can be scheduled in the routine INSTALL.MODIFICATION and in events ENTER.SERVICE and REPAIR.

Local Variables

CDM.MULTIPLYING FACTOR - This real variable is the factor which when multiplied by the remaining time to crack initiation accounts for the shortening effect of corrosion on fatigue life.

IDCO - This integer variable contains the identification number of the aircraft for which the event CORROSION was scheduled.

NFTM - If a FAILURE has been scheduled, this real variable is the remaining time until its occurrence.

REDUCED.REMAINING.LIFE - This real variable is the REMAINING.LIFE multiplied by the corrosion damage factors.

REMAINING.LIFE - This real variable is the remaining time until a scheduled crack initiation.

RST - This real variable is the remaining service time of the aircraft under consideration.

TRT - If a REACH.FAIL.SAFE.LGT has been scheduled, this real variable is the remaining time until its occurrence multiplied by the corrosion damage factor.

MAJOR LOGIC STEPS

1. Call COMP.RISK if crack exists.
2. Determine if corrosion is internal or external.
3. If inspections not scheduled (Call INSPECTION.SCHEDULER).
4. Determine if corrosion in stress concentration factor.
5. Calculate corrosion damage factor.
6. Set crack growth rate multiplying factor to corrosion resistance rating.
7. Cancel 1.ITE, 2.ITE, FAILURE, REACH.FAIL.SAFE.LGT, 1STRENGTH.REDUCTION, and 2.STRENGTH.REDUCTION if scheduled.
8. Schedule 1.ITE, 2.ITE, FAILURE, REACH.FAIL.SAFE.LGT, 1.STRENGTH.REDUCTION, and 2.STRENGTH.REDUCTION based on time reduction by corrosion resistance rating or corrosion damage factor.

ROUTINE RATE

DESCRIPTION - This routine statistically generates element crack growth rates which reflect variation in material properties and load environment. The growth rates are randomly drawn from a normal distribution which is defined by a mean growth rate and a standard deviation passed from the calling routine. If a random draw yields a negative growth rate, the rate is set equal to the mean growth rate minus four standard deviations. Thus, the user must be sure that the standard deviation is always less than one-fourth of the means. This routine is defined as a function in the PREAMBLE and is used in event ENTER.SERVICE and INSTALL.MODIFICATION.

Local Variables

G1,G2 - These real variables are used to hold intermediate values in the calculation of the crack growth rate.

M - This real variable is the mean crack growth rate passed from the calling routine.

RN - This real variable is a uniformly distributed random number between zero and one.

RNF - This real variable is $1 - RN$.

S - This real variable is the crack growth rate standard deviation passed from the calling routine.

W - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

Z - This real variable is the element crack growth rate returned to the calling routine.

EVENT 1. STRENGTH. REDUCTION

DESCRIPTION - This event represents the initiation of the first crack. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 1.ITE is scheduled. The time to structural failure is calculated using a five-part residual strength curve. If this time is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail safe length is calculated. If this time is less than the remaining service life of the aircraft, the event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

Local Variables

A - This real variable is obtained for each aircraft from a log-normal distribution defined by AMEAN.

ARG, K1, K2, K4, K8, K9, K10, K11, K12, K13, K14, K15, K16, K17, K18, K19, K20, K21, K22, K23, K24, LG, LOK5 - These real variables are used as intermediate values in the calculation of time until structural failure.

GR1 - This real variable is the element first crack growth rate.

GR2 - This real variable is the element second crack growth rate.

GR3 - This real variable is the element third crack growth rate.

GR4 - This real variable is the element fourth crack growth rate.

IDISR - This integer variable is the aircraft identification number.

R1, R2, R3, R4, R5 - These real variables represent the strength degradation on the five-part strength degradation curve.

S1, S2, S3 - These real variables represent the residual strengths at CONE, CTWO, and CTHREE.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

T1, T2, T3, T4 - These real variables are the time required for the crack to grow to CONE, CTWO, CTHREE, and FSAF.LGT.

TTF - This real variable is the time in flight hours until element failure.

MAJOR LOGIC STEPS

1. Determine if crack initiates internally.
2. Calculate time for internal crack to become external.
3. If crack internal and becomes external within aircraft service life (Schedule 1.ITE).
4. If inspections not scheduled (Call INSPECTION.SCHEDULER).
5. Determine time to reach fail safe length.
6. If crack reaches fail safe length within aircraft service life (Schedule REACH.FAIL.SAFE.LGT).
7. Determine time to failure.
8. If time to failure within aircraft service life (Schedule FAILURE).

EVENT 2. STRENGTH REDUCTION

DESCRIPTION - This event represents the second crack initiation. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 2.ITE is scheduled if it occurs within the service life of the airplane. The time to structural failure is calculated using a nine-part residual strength curve, which is dependent on the crack growth rate history of both the first and second crack. If the time until failure is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches fail safe strength is calculated. If this time is less than the remaining service life of the aircraft, the event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

Local Variables

A - This real variable is obtained for each aircraft from a log-normal distribution defined by AMEAN.

ARG, K1, K2, K4, K8, K9, K10, K11, K12, K13, K14, K15, K16, K17, K18, K19, K20, K21, K22, K23, K24, K25, K26, K27, K28, K29, K30, K31, K32, K33, K34, K35, K36, K37, K38, K39, K40, K41, K42, K43, K44, LGK5, LG - These local variables are used as intermediate values in the calculation of time to element failure.

CCL1, CCL2, CCL3 - These real variables equal CONE, CTWO, and CTHREE respectively.

DL - This real variable is the length of the first crack at corrosion initiation.

GR1, GR2, GR3, GR4, GR5, GR6, GR7, GR8, GR9 - These real variables are the combined growth rates of the first and second cracks.

ID2SR - This integer variable is the aircraft identification number.

R1, R2, R3, R4, R5, R6, R7, R8, R9, These real variables represent the strength degradation on the nine-part strength degradation curve.

S1, S2, S3, S4, S5, S6, S7 - These real variables represent the residual strengths at the actual crack length and fail safe length of the first and second cracks.

T - This real variable is the time in flight hours until a crack initiated Internally becomes external.

TAC - This real variable is the time of corrosion initiation.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

TA1 - This real variable is the time of the first crack initiation.

TCL1 - This real variable is the time for the crack to reach CONE.

TCLL - This real variable is the time for the crack to grow between CONE and CTWO.

TCL3 - This real variable is the time for the crack to grow between CTWO and CTHREE.

T1, T2, T3, T4, T5, T6, T7, T8 - These real variables are the times required for the two cracks to reach critical crack lengths and fail safe lengths.

MAJOR LOGIC STEPS

1. Call COMP.RISK.
2. Determine if crack initiates internally.
3. Calculate time for internal crack to become external.
4. If crack internal and becomes external within aircraft service life (Schedule 2.ITE).
5. Determine time for two cracks to reach fail safe length.
6. If cracks reach fail safe length within aircraft service life (Schedule REACH.FAIL.SAFE.LGT).
7. Determine time to failure.
8. If time to failure within aircraft service life (Schedule FAILURE).

EVENT 1.ITE

DESCRIPTION - This event represents the time when a first crack which initiated internally becomes external. This time is defined by L.EXT, an input quantity. At this time the appropriate element of the alpha array 1.INT is changed from "YES" to "NO". This event is scheduled in event 1.STRENGTH.REDUCTION.

Local Variables

IDIE - This integer variable is the aircraft identification number.

EVENT 2.ITE

DESCRIPTION - This event represents the time when a second crack which initiated internally becomes external. This time is defined by L.EXT, an input quantity. At this time the appropriate element of the alpha array 2.INT is changed from "YES" to "NO". This event is scheduled in event 2.STRENGTH REDUCTION.

Local Variables

ID2E - This integer variable is the aircraft identification number.

ROUTINE INSPECTION.SCHEDULER

DESCRIPTION - This routine schedules all inspections below the D-level on a given aircraft. To conserve execution time, the inspections are scheduled so that the aircraft is not inspected before the defect reaches its minimum detectable size at the A, B, and C inspection levels. This routine is called from events 1.STRENGTH.REDUCTION and COROSION.

Local Variables

C1 - This real variable is the corrosion growth rate used to calculate the time to the minimum detectable corrosion area.

M1 - This real variable is the crack growth rate used to calculate the time to the minimum detectable crack length.

N - This integer variable indicates whether a crack initiation or a corrosion initiation caused this routine to be called.

S.INSPECTION.AT - This real variable is the simulation time at which the first inspection at each level is scheduled.

TML - This real variable is the time to the minimum detectable defect size calculated for each level of inspection.

EVENT A.LEVEL.INSPECTION

DESCRIPTION - This event represents the performance of an A-level inspection. The constants which define the probability of detection are passed to the routine EXAMINE which handles the actual inspection. If a defect is detected on one of the ten high time aircraft, the time is stored in array MRDD. This event can be scheduled in the routine INSPECTION.SCHEDULER and the events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

Local Variables

FOUND - This real variable is used to indicate if a defect is found and the type of defect.

IDA - This integer variable is the identification number of the aircraft being inspected.

EVENT B.LEVEL.INSPECTION

DESCRIPTION - This event represents the performance of a B-level inspection. The inspection is not done, if the B-level inspection is within one-half a B-level interval from the next C-level or D-level inspection. The constants which define the probability of detection equation at the B-level are passed to the routine EXAMINE, which determines if the defect is found. The inspection interval for each inspection level is not necessarily an even multiple of all lower level intervals. If an A-level inspection is scheduled, it is cancelled and rescheduled at present time, plus one A-level interval later. If a defect is detected on one of the ten high time aircraft, the time is stored in array MRDD. This event can be scheduled in the routine INSPECTION.SCHEDULER and events B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

Local Variables

FOUND - This real variable is used to indicate if a defect is found and the type of defect.

IDB - This integer variable is the identification number of the aircraft being inspected.

EVENT C.LEVEL.INSPECTION

DESCRIPTION - This event represents the performance of a C-level inspection. The inspection is not done, if the C-level inspection is within one-half a C-level interval of the next D-level inspection. If there is either an A-level or a B-level inspection currently scheduled, it is cancelled and rescheduled at one A-level or B-level interval, respectively, later. The constants which define the probability of defect detection equation at the C-level are passed to the routine EXAMINE, which determines if the defect is found. The C-level inspection uses both exploratory and non-exploratory probability of detection curves. The non-exploratory curve goes into effect after one crack has been detected. If a defect is detected on one of the ten high time aircraft, the time is stored in array MRDD. This event can be scheduled in routine INSPECTION.SCHEDULER and events C.LEVEL.INSPECTION and D.LEVEL.INSPECTION.

Local Variables

FOUND - This real variable is used to indicate if a defect is found and the type of defect.

IDC - This integer variable is the identification number of the aircraft being inspected.

EVENT D.LEVEL.INSPECTION

DESCRIPTION - This event represents the performance of a D-level inspection. If there are any lower level inspections currently scheduled, they are cancelled and rescheduled at one inspection interval later. A determination is made whether an aircraft receives an external or internal inspection dependent on the current value of the variable SAMPLING. The constants which define the probability of defect detection equation at the D internal and external levels are passed to the routine EXAMINE, which determines if a defect is found. Explanatory and non-explanatory probability of crack detection curves are used. Non-explanatory crack detection curves go into effect after a crack has been detected. The defect histories of the ten-high time aircraft are now examined. If all of the ten-high time aircraft time goes one D-level interval without a defect being detected at any inspection level, the C-level and D-level intervals are increased and the value of SAMPLING is changed.

The changes in the C and D intervals and SAMPLING are dependent on the current value of the D-level interval. If the current D-level interval is less than or equal to 12,000 hours, the following relations are used:

$$\frac{ABCD(3)}{NEW} = \frac{ABCD(3)}{OLD} * FREQ.DECREASE / 2 + 1$$

$$\frac{ABCD(4)}{NEW} = \frac{ABCD(4)}{OLD} * FREQ.DECREASE * 2 + 1$$

$$\frac{SAMPLING}{NEW} = \frac{SAMPLING}{OLD} * (FREQ.DECREASE * 2 + 1) + 1$$

If the current D-level interval is greater than 12,000 hours, the following relations are used:

$$\frac{ABCD(3)}{NEW} = \frac{ABCD(3)}{OLD} * (FREQ.DECREASE + 1)$$

$$\frac{ABCD(4)}{NEW} = \frac{ABCD(4)}{OLD} * (FREQ.DECREASE + 1)$$

$$\frac{SAMPLING}{NEW} = \frac{SAMPLING}{OLD} + 1$$

The D-level inspection interval is limited to 32,000 hours and the C-Level interval is limited to 32,000 times the ratio of the initial C-interval to the initial D-interval. The D-level inspection interval for all aircraft is rescheduled at the new value. No changes are permitted if a modification is pending. If a service modification or retrofit modification is pending, it is done at the D-level. If the D-level interval has been decreased prior to a modification, then it is restored to its value prior to the decrease after the modification.

A check is done in the routine to replace the ten high time aircraft after they retire. The ten high time aircraft are replaced by ten aircraft whose next D-level inspection occurs closest to one D-level interval from the last increase. After this point the last ten aircraft in the fleet are taken as the high time aircraft.

If a defect is found and a modification is not done, a repair is scheduled. This event can be scheduled in the events ENTER.SERVICE, D.LEVEL.INSPECTION, T.INSPECTION.INCREASE and INCREASE.INSPECTION.FREQUENCY.

Local Variables

DECC - This real variable is the factor the C-interval is multiplied by for an interval increase when old D-interval is less than 12,000 hours.

DECD - This real variable is the factor the D-interval is multiplied by for an interval increase when old D-interval is less than 12,000 hours.

FOUND - This real variable is used to indicate if a defect is found and the type of defect.

IDD - This integer variable is the identification number of the aircraft being inspected.

N - This integer variable contains the identification numbers of aircraft that receive internal D-level inspections.

SMALL - This real variable is used in determining which ten aircraft replace the ten high time aircraft after they retire.

MAJOR LOGIC STEPS

1. Schedule the next D.LEVEL.INSPECTION.
2. Cancel and reschedule lower level inspections one interval from the present D-level inspection.
3. Determine if internal or external inspection.
4. Call EXAMINE.
5. Check high time aircraft for defect.
6. Increase inspection intervals if no modification pending and no defect on high time aircraft.
7. Reschedule D.LEVEL.INSPECTION if step 6 is true.
8. Determine aircraft that will replace ten high time aircraft after they retire.
9. If modification pending:
 - a. Call INSTALL.MODIFICATION.
 - b. Reschedule D.LEVEL.INSPECTION.
 - c. Return.
10. If defect found, schedule REPAIR.

ROUTINE EXAMINE

DESCRIPTION - This routine performs the numerical comparison which determines whether a defect is detected at each level of inspection. This routine receives constants from calling routines which determine the probability of defect detection. The calculation of the probability of corrosion detection and the probability of crack detection that is exponential is done by routine PDD. For crack lengths that are not on the exponential curve the probability of detection is determined in routine EXAMINE for all inspection levels. Internal defects can only be detected by internal inspection. Cracks that originate internally and become external can be detected by external inspection, but use only the external length of the crack. If a defect is detected, a repair is scheduled if the inspection level is A, B, or C. This routine can be called from events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

Local Variables

AREA - This real variable is the calculated area of the corrosion defect.

CCL1 - This real variable is the first critical crack length.

CCL2 - This real variable is the second critical crack length.

CCL3 - This real variable is the third critical crack length.

CL - This real variable is the calculated length of the fatigue crack.

FOUND - This real variable is set equal to two whenever corrosion is detected and to one whenever a crack is detected.

M1, M2, M3, M4 - These real variables are the individual element crack growth rates.

N - This integer variable identifies the inspection level and is passed by the calling event.

PL - This real variable is the probability of defect detection.

TAC - This real variable is the simulation time of corrosion initiation.

TA1 - This real variable is the simulation time of the first crack initiation.

TA2 - This real variable is the simulation time of the second crack initiation.

XA, XL, YA, ZA, ZL - These real variables are the probability of defect detection constants for the exponential curve and are passed by the calling routines.

Z - This alpha variable is the level of inspection being performed.

MAJOR LOGIC STEPS

1. If corrosion exists, and inspection is internal, or corrosion is external:
 - a. Calculate corrosion area.
 - b. Call PODD to determine probability of detection.
 - c. Draw random number and compare to probability of detection. If random number less than or equal to probability of detection, then corrosion is found.
2. If first crack exists, and inspection is internal or crack is external:
 - a. Calculate crack length.
 - b. If crack originated internally and inspection is external, subtract internal portion of crack from crack length.
 - c. Determine probability of detection for crack, if crack length is on non-exponential part of detection curve. Probability of crack detection is dependent on inspection level and crack length.
 - d. If crack length on exponential part of detection curve, call PODD for probability of detection.
 - e. Draw random number and compare to probability of detection. If random number less than or equal to probability of detection, the first crack is found.
 - f. If crack found and crack originated internally and external inspection, add internal portion of crack back to length for recording of total crack length.
3. If second crack exists, and inspection is internal or crack is external (Same as steps 2A - 2F).
4. If defect found, schedule REPAIR if inspection level A, B, or C.

ROUTINE PODD

DESCRIPTION - This routine computes the probability of detection for corrosion and for cracks that are on the exponential portion of the crack detection curve. This probability is returned to the calling routine. PODD is called from routine EXAMINE and event IMMEDIATE, FLEET.INSPECTION.

Local Variables

L - This real variable is the size of the defect under consideration.

RL - This real variable is the probability of detection of the defect.

X - This real variable is the maximum probability of detection at a given inspection level.

Y - This real variable is an empirically determined equation constant for each level of inspection.

Z - This real variable is the minimum defect size detectable at a given inspection level.

ROUTINE CANCEL.SCHEDULED.INSPECTIONS

DESCRIPTION - This routine cancels all scheduled inspections below the D-level on a given aircraft. Whenever a defect is detected and repaired, it is assumed that all other defects existing on that particular element are also repaired. This routine is called at this time to cancel all subsequent inspections. Also, if an element fails or an aircraft with existing defects is retired, this routine is called to cancel all scheduled inspections. This routine can be called from events FAILURE, RETIRE.FROM.SERVICE, and REPAIR.

EVENT REACH.FAIL.SAFE.LGT

DESCRIPTION - This event represents the time when the residual strength of the element has been reduced to the fail-safe strength. The time and aircraft identification numbers are saved as output. The calculations of the strength reduction is based on the sum of the two crack lengths, if the second crack is present in the element. This event can be scheduled in events 1.STRENGTH.REDUCTION and 2.STRENGTH.REDUCTION.

Local Variables

IDRFS - This integer variable is the identification number of the aircraft being processed.

EVENT FAILURE

DESCRIPTION - This event represents structural failure. When this event occurs, the aircraft is removed from the active fleet. If this aircraft was one of the ten high-time aircraft being monitored for the purpose of increasing inspection intervals, it is replaced by the next high-time aircraft in active service. Any remaining scheduled events are cancelled and their event routines destroyed. The events INCREASE.INSPECTION.FREQUENCY and routine COMP.RISK is called. This event can be scheduled in events 1.STRENGTH.REDUCTION and 2.STRENGTH.REDUCTION.

Local Variables

HOLD This integer variable serves as an intermediate storage for aircraft identification when replacing one of the ten high-time aircrafts.

IDFA - This integer variable is the identification number of the aircraft under consideration.

S - This real variable is the residual strength at failure.

EVENT RETIRE.FROM.SERVICE

DESCRIPTION - This event represents the retirement of an aircraft from active service. If a crack exists the routine COMP.RISK is called. The aircraft being retired is removed from the set ACTIVE.FLEET and filed in the set FLEET.RETIRED. All remaining scheduled events for this aircraft are cancelled and the event notices destroyed. This event is scheduled in the event ENTER.SERVICE.

Local Variables

IDRET - This integer variable is the identification number of the aircraft being retired from service.

EVENT REPAIR

DESCRIPTION - This event represents the structural repair of an element. If the events FAILURE and REACH.FAIL.SAFE.LGT are scheduled, they are cancelled. It is assumed that all existing defects are repaired and that new times to defect occurrences are determined in the same manner as when the aircraft entered service. The size and number of all existing cracks are compared with the inspection interval decrease criteria.

The inspection interval decrease criteria differentiates between large and small cracks. Large crack criteria is a crack greater than the fail-safe length or if the sum of the crack lengths plus crack growth rate times inspection interval is greater than $(SU-1.2*(DLL+1)-1)*FSAF.LGT/(SU-SF)$. The inspection interval is the current D-level interval if the crack is found in an external inspection. Otherwise, the inspection interval is the current D-level interval times the current SAMPLING. If a large crack is found, an IMMEDIATE.FLEET.INSPECTION and INCREASE.INSPECTION.FREQUENCY is scheduled.

If the special fleet inspection detects a crack, the inspection interval can be reduced again if the crack detected during the special inspection meets the criteria just discussed. Small cracks are those not determined by the criteria to be large. If the sum of the cracks found in the entire fleet times the fail-safe crack length is greater than one-fifth of the fleet size, the inspection interval is reduced.

If a modification is pending on the aircraft being repaired, the modification is installed at this time. A decision on modification, due to the current defect, is scheduled. The routine COMP.RISK is called. This event can be scheduled in routine EXAMINE and events D.LEVEL.INSPECTION and IMMEDIATE.FLEET.INSPECTION.

Local Variables

AAFL - This real variable is the actual average fatigue life of the element.

CCL1, CCL2, CCL3 - These real variables are the critical crack lengths.

CL - This real variable is the crack length.

FIRST.LIFE - This real variable is the time in flight hours to first crack initiation.

HOURS.TO.CORROSION - This real variable is the time in flight hours to corrosion initiation.

IDREP - This integer variable is the aircraft identification number.

MAX.CRK - This real variable is the sum of the crack lengths in the element.

POT.CRK - This real variable is the sum of the maximum crack length and the length that the crack will gain in the next D-level inspection interval.

RST - This real variable is the remaining service time of the aircraft.

SDL - This real variable is $1.2 * DLL - 1$.

SECOND.LIFE - This real variable is the time in flight hours to the second crack initiation.

STR.RED - This real variable is the element strength reduction because of all existing cracks.

TAC - This real variable is simulation time to corrosion initiation.

TA1, TA2 - These real variables are simulation time to the first and second crack initiation.

MAJOR LOGIC STEPS

1. Cancel REACH.FAIL.SAFE.LGT and FAILURE, if scheduled.
2. Cancel CORROSION if corrosion exists.
3. Call PREDICT.CORROSION.
4. Schedule CORROSION if it occurs within remaining service life.
5. Calculate first and second crack lengths.
6. Calculate fleet strength reduction and potential crack growth.
7. If crack large, schedule an IMMEDIATE.FLEET.INSPECTION and an INCREASE.INSPECTION.FREQUENCY.
8. If crack small and fleet strength reduction criteria met, schedule an INCREASE.INSPECTION.FREQUENCY.
9. Install modification if pending, call INSTALL.MODIFICATION.
10. For fatigue test failure modification cancel T.INSPECTION. INCREASE if scheduled.
11. If crack exists, call COMP.RISK.
12. Schedule DECISION.ON.MOD.
13. Cancel and reschedule fatigue crack initiation.
14. If inspection scheduled, call CANCEL.SCHEDULED.INSPECTIONS.

EVENT T.INSPECTION.INCREASE

DESCRIPTION - This represents an inspection interval decrease for a particular aircraft pending a structural modification because of a fatigue test failure. The D-interval is reduced by the input quantity T.FREQ.CHG. The D-level inspection is rescheduled for aircraft with modification pending. This event is scheduled in event T.IMPLEMENT.MOD.

Local Variables

IDTI - This integer variable is the identification number of the aircraft under consideration.

EVENT INCREASE.INSPECTION.FREQUENCY

DESCRIPTION - This event represents a fleet wide decrease in the D-level inspection interval and a change in SAMPLING. The D-level inspection interval is reduced by the input quantity S.FREQ.CHG. SAMPLING is set equal to one, if the crack is detected in a D-level internal inspection. For SAMPLING equal to one, every aircraft is inspected internally at the D-level. If the crack is not found in an internal D-level inspection SAMPLING is changed by dividing its current value by S.FREQ.CHG. The D-level inspection interval for the entire fleet is then rescheduled. This event can be scheduled in events REPAIR and IMMEDIATE.FLEET.INSPECTION.

EVENT IMMEDIATE.FLEET.INSPECTION

DESCRIPTION - This event represents an immediate fleet-wide inspection caused by finding a large crack. Existing crack lengths and corrosion areas are calculated along with the associated probabilities of detection. As in the scheduled inspections, these probabilities are compared with a random number to determine whether or not the defect is detected. If any cracks are found and the inspection intervals had not previously been reduced an INCREASE.INSPECTION.FREQUENCY is scheduled. If any defects are found a REPAIR is scheduled. These defects will not cause any additional fleet-wide inspections. This event is scheduled in event REPAIR.

Local Variables

AREA - This real variable is the area of an existing corrosion defect.

CCL1, CCL2, CCL3 - These real variables are the critical crack lengths.

CK.FIND - This real variable is set equal to one if a crack is found in the special inspection.

CL - This real variable is the length of an existing crack.

FOUND - This real variable indicates if a defect is found and the type of defect.

M1, M2, M3, M4 - These real variables are the four crack growth rates.

PL - This real variable is the probability of detection of an existing defect.

TAC - This real variable is the time of initiation of an existing corrosion defect.

TA1, TA2 - These real variables are the times to initiation of the first and second cracks.

MAJOR LOGIC STEPS

1. If corrosion exists:
 - a. Calculate area.
 - b. Determine probability of detection, call PODD.
 - c. Compare with random number to determine if detected.
2. If first crack exists:
 - a. Calculate length.
 - b. Determine probability of detection, call PODD if on exponential part of detection curve.
 - c. Compare with random number to determine if detected.
3. If second crack exists:
(Same as 2A - 2C).

4. If high time aircraft have defects, record time.
5. If defects found, schedule REPAIR.
6. If crack found and inspection interval not previously reduced, schedule an INCREASE.INSPECTION.FREQUENCY.

EVENT DECISION.ON.MOD

This event makes the decision on whether or not to develop a structural modification because of service experience. The decision to develop a modification is made by comparing the cost per flight hours of the modification with the repair cost per flight hours plus the increased inspection cost per flight hours. The modification cost per flight hours is found by dividing the total fleet modification cost by the remaining service life of the fleet. The repair cost per flight hours is found by dividing the total fleet repair costs since the last modification by the fleet flight time since the last modification. The increased inspection cost per flight hour is found by dividing the projected increased inspection costs by the remaining service life of the fleet. SAMPLING is included in the increased inspection costs calculation since cracks found in internal D-level inspections set SAMPLING equal to one, while SAMPLING is changed by S.FREQ.CHG if the crack is not found at the internal D-level inspection. If a decision is made to modify, the event IMPLEMENT.MODIFICATION is scheduled. This event is scheduled in the event REPAIR.

LOCAL VARIABLES

ACCUMULATED.HRS - This real variable is the total fleet time since the last modification.

ICPH - This real variable is the increased inspection cost per flight hour.

MCPH - This real variable is the modification cost per flight hour.

MD.COST - This real variable is the cost of installing a modification on a single aircraft.

MRFH - This real variable is the service time remaining on a particular aircraft after its modification.

NFTS - This real variable is the total production time of all aircraft entering service after the second production rate goes into effect.

NPDL - This real variable is the number of aircraft which have entered service.

POST.MOD.HRS - This real variable is the total fleet service time remaining after the modification.

RCPH - This real variable is the repair cost per flight hour.

TOOLING - This real variable is the tooling cost in the development of a modification.

EVENT IMPLEMENT.MODIFICATION

DESCRIPTION - This event represents the development of a modification because of service experience. If the modification is to be fatigue tested, the actual average fatigue life is set equal to the original predicted life of the element design. Otherwise, the actual average fatigue life is determined by calling routine REAL.LIFE. The modification will be installed on each aircraft at the next D-level inspection or defect repair. The D-level inspection interval and D-level internal sampling are set to their values prior to the interval reduction. This event can be scheduled in event DECISION.ON.MOD.

Local Variables

NSIG - This real variable is the standard deviation of the ratio distribution passed to routine REAL.LIFE.

NMU - This real variable is the mean of the ratio distribution passed to routine REAL.LIFE.

ROUTINE DISPLAY.OUTPUT

DESCRIPTION - This routine prints the output for each element. This output is suppressed if the long list option is in effect. This routine is called from the MAIN program.

Local Variables

All the local variables in this routine are used to temporarily store output values.

ROUTINE SUMMARY

DESCRIPTION - This routine prints the summary output for each element type. This output is suppressed if the long list option is in effect. This routine is called from the MAIN program.

Local Variables

All the local variables in this routine are used to temporarily store output values.

ROUTINE SUMMARY

DESCRIPTION - This routine prints the summary output for each element type. This output is suppressed if the long list option is in effect. This routine is called from the MAIN program.

Local Variables

All the local variables in this routine are used to temporarily store output values.

APPENDIX B
MODIFIED PROGRAM SOURCE LISTING

```

1 PREAMBLE
2
3 THE SYSTEM OWNS AN ACTIVE.FLEET, A FLEET.RETIRED AND A CRASHED.FLEET
4
5 ## GLOBAL REAL VARIABLES
6
7 DEFINE PREDICTED.LIFE, ACTUAL.AVG.FAT.LIFE, START.TEST.TEST.LIFE,
8 TEST.ACCEL.FACT, BEGIN.PRODUCTION, PRODUCTION.TIME, USAGE.LIFE
9 AS REAL VARIABLES
10 DEFINE LGHT.TO.FAILURE, FLEET.STR.RED, CONE, LEAD.TIME, IAAFL,
11 CTVO, CTHREE, INT.CONE, INT.CTMO, IN.CTHREE
12 AS REAL VARIABLES
13 DEFINE RIRTH.DEFECT.PROBABILITY, SDM.OCCURRENCE.RATE, C.GROWTH.RATE,
14 CRPF, LOCATED.IN.STRESS.CON,
15 MU.R, SIG.R, FSAF.LGT,
16 MI.MEAN, M2.MEAN, AMEAN, B, C7, C28, M3.MEAN, M4.MEAN,
17 INT.M1, INT.M2, INT.M3, INT.M4
18 AS REAL VARIABLES
19 DEFINE A.REPAIR.COST, B.REPAIR.COST,
20 C.REPAIR.COST, D.REPAIR.COST, S.REPAIR.COST,
21 COST.OF.REPAIRS, FIXIT.COST, 1ST.MD.COST, AD.MD.COST, 1ST.TOOLING,
22 AD.TOOLING, T.FREQ.CHG, S.FREQ.CHG, CHG.FREQ.TIME
23 AS REAL VARIABLES
24 DEFINE L.PROB, C.PROB, ICRKT, ICRKT, 1SDT, ACRKL, BCRKL,
25 CCRKL, DCRKL, SCRKL, ACA, BCA, CCA, DCA, SCA
26 AS REAL VARIABLES
27 DEFINE 1.CDM.OCCURRENCE.RATE, 2.CDM.OCCURRENCE.RATE, CDM.RATE.CHANGE,
28 FREQ.DECREASE, 2.PRODUCTION.TIME, PRCHG, AIRFRAME.TIME
29 AS REAL VARIABLES
30 DEFINE GICRK, GICOR, G1SD, GACRK, GBCRK, GCCRK, GSCRK, GACA, GBACA,
31 GSCA, GBACA, GSCA, CINS, DINS, SU, SF, L.FIT,
32 ABC.OLD.WARD.OLD.D.POP.SIZE.SAMP.SIZE, LAMDA, ELTYP.FAIL.RATE,
33 AC.FAIL.RATE, AP.FAIL.RATE, AEXP, BA, AVGL, STD,
34 AV.AP.FAIL, AV.AC.FAIL, AV.EL.FAIL, DLL
35 AS REAL VARIABLES
36
37 ## GLOBAL INTEGER VARIABLES
38
39 DEFINE ID, IDCK, SIZE.OF.FLEET, COUNT.ELEMENT, NICHG, LHTA,
40 1.NUM.OF.RETIRE, 2.NUM.OF.CRASH, ITHML, EXT.INSPL, LEVEL,
41 LIL, LEL, TO.BE.MODIFIED, REEN.MODIFIED, FDCK, CR
42 AS INTEGER VARIABLES
43 DEFINE DICR, OCOR, OSDM, OPD, OSCR, NSIC, NSMD, NSFL, NMD,
44 NRFS, SNRFS, J, NOE, LD, IMOD.NO.FAIL,
45 GICR, GICOR, GOSDM, GOSCR, GOSCO, GORD, SNSIC, SNMD, SNSMD, SNSFL,
46 CR.CTR, SAMPLING, OLD.SAMP, MOD.NO, NCZ.NCZO, NTIN, FAIL.OPT, NN
47 AS INTEGER VARIABLES
48
49 ## GLOBAL ALPHA VARIABLES
50
51 DEFINE TES.FAILURE, MOD.TESTED, PREVIOUSLY.MODIFIED, IMP.MOD.SCH,
52 LONG.LIST, IFAIL, LTHO,
53 DEC.ON.MOD.SCH, SEL1, SEL2, SEL3, INT.LVL.INSPL, EXT.LVL.INSPL,
54 SB.CRK, BEC.INT, D.INT.FIND, IFLAG.FLAG.RETRO

```

```

54      AS ALPHA VARIABLES
55      REAL ARRAYS
56      #B
57      #E
58      DEFINE C-INTERVAL, D-INTERVAL, ABCD, CABCD, IABCD, CKREP.TIME, COREP.TIME,
59      LAST.SD, OCCUR.MOD, MSR1, MFR1, SC, SD, XRM, MRDD, CGRI, INT.LGT,
60      T-LAST.D, MSR2, MFR2, MSR1.INT, MFR1.INT, MSR2.INT, MFR2.INT,
61      SRRATE, TRCHG, SKES, CLGT, AFACI, CHG.TIME,
62      FSAVE, PSAVE, AVPSA, AVESA, SDIF, RLGT
63      AS REAL, 1-DIMENSIONAL ARRAYS
64      #B
65      #E
66      INTEGER ARRAYS
67      #B
68      #E
69      DEFINE AIRR, A2SP, AF, AIRPLANE, AAL, ABL, ACL, ADL, AC, ATII, ACID,
70      OICR, OICU, SACID, GOICR, GOICO, HI.TIME, ACRFT, APID, SAPID, STIM,
71      SSTIM, FLTHR, SPLTHR, ARPSL, NGAC, AIE, AZE, D.IN, SAMP,
72      MOD.SAVE, PFID, PFTIM
73      AS INTEGER, 1-DIMENSIONAL ARRAYS
74      DEFINE TLID AS AN INTEGER, 2-DIMENSIONAL ARRAY
75      #B
76      #E
77      ALPHA ARRAYS
78      #B
79      #E
80      DEFINE MODEL, ELEMENT, 1.CR.EXISTS, 2.CR.EXISTS, CO.EXISTS,
81      SO.SCH, SSTAN, SELNB, AIL, FSH, IE1, IE2, D.FXT,
82      TMOD.PENDING, SMOD.PENDING, INSP.SCH, 1.INT, 2.INT, C.INT,
83      EL.E2, E3, E4
84      AS ALPHA, 1-DIMENSIONAL ARRAYS
85      DEFINE ELID AS AN ALPHA, 2-DIMENSIONAL ARRAY
86      DEFINE HZRD, HZ AS DOUBLE, 1-DIMENSIONAL ARRAYS
87      #B
88      #E
89      TEMPORARY ENTITIES
90      #B
91      #E
92      EVERY AIRCRAFT HAS AN ENTRY.TIME, A TAIL.ID
93      AND MAY BELONG TO AN ACTIVE.FLEET, A FLEET.RETIPED AND A CRASHED.FLEET
94      DEFINE TAIL.ID AS AN INTEGER VARIABLE
95      #B
96      #E
97      TALLY NICO AS THE MINIMUM, XICO AS THE MAXIMUM, MICO AS THE MEAN OF ICORRT
98      TALLY NISO AS THE MINIMUM, XISO AS THE MAXIMUM, MISO AS THE MEAN OF ICORT
99      TALLY NICO AS THE MINIMUM, XICO AS THE MAXIMUM, MICO AS THE MEAN OF ICORT
100      TALLY NISO AS THE MINIMUM, XISO AS THE MAXIMUM, MISO AS THE MEAN OF ICORT
101      TALLY NICO AS THE MINIMUM, XICO AS THE MAXIMUM, MICO AS THE MEAN OF ICORT
102      TALLY NISO AS THE MINIMUM, XISO AS THE MAXIMUM, MISO AS THE MEAN OF ICORT
103      TALLY NICO AS THE MINIMUM, XICO AS THE MAXIMUM, MICO AS THE MEAN OF ICORT
104      TALLY NISO AS THE MINIMUM, XISO AS THE MAXIMUM, MISO AS THE MEAN OF ICORT
105      TALLY NICO AS THE MINIMUM, XICO AS THE MAXIMUM, MICO AS THE MEAN OF ICORT
106      TALLY NISO AS THE MINIMUM, XISO AS THE MAXIMUM, MISO AS THE MEAN OF ICORT

```

107 TALLY SBCR AS THE MINIMUM, LBCR AS THE MAXIMUM, ABCR AS THE MEAN OF GBCRK
 108 TALLY SCCP AS THE MINIMUM, LCCR AS THE MAXIMUM, ACCR AS THE MEAN OF GBCRK
 109 TALLY SDGP AS THE MINIMUM, LDCR AS THE MAXIMUM, ADCR AS THE MEAN OF GBCRK
 110 TALLY SSCP AS THE MINIMUM, LDCR AS THE MAXIMUM, ADCR AS THE MEAN OF GBCRK
 111 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 112 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 113 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 114 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 115 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 116 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 117 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 118 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 119 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 120 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 121 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 122 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 123 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 124 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 125 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 126 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 127 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 128 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 129 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 130 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 131 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 132 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 133 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 134 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 135 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 136 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 137 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 138 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 139 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 140 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 141 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 142 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 143 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 144 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 145 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 146 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 147 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 148 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 149 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 150 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 151 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 152 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 153 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 154 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 155 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 156 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 157 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 158 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK
 159 TALLY SBGA AS THE MINIMUM, LACA AS THE MAXIMUM, AACR AS THE MEAN OF GBCRK

EVENT NOTICES INCLUDE ENTER-SERVICE, T-IMPLEMENT,MOD, IMPLEMENT,MODIFICATION,
 INCREASE,INSPECTION,FREQUENCY, IMMEDIATE,FLEET,INSPECTION,
 DECISION,ON,MOD
 EVERY IN-SERVICE,DAMAGE HAS AN IDSON
 DEFINE IDSON AS AN INTEGER VARIABLE
 EVERY CORROSION HAS AN IDCO
 DEFINE IDCO AS AN INTEGER VARIABLE
 EVERY 1-STRENGTH,REDUCTION HAS AN ID1SP
 DEFINE ID1SP AS AN INTEGER VARIABLE
 EVERY 2-STRENGTH,REDUCTION HAS AN ID2SR
 DEFINE ID2SR AS AN INTEGER VARIABLE
 EVERY 1-ITE HAS AN IDIE
 DEFINE IDIE AS AN INTEGER VARIABLE
 EVERY 2-ITE HAS AN ID2E
 DEFINE ID2E AS AN INTEGER VARIABLE
 EVERY A-LEVEL,INSPECTION HAS AN IDA
 DEFINE IDA AS AN INTEGER VARIABLE
 EVERY B-LEVEL,INSPECTION HAS AN IDB
 DEFINE IDB AS AN INTEGER VARIABLE
 EVERY C-LEVEL,INSPECTION HAS AN IDC
 DEFINE IDC AS AN INTEGER VARIABLE
 EVERY D-LEVEL,INSPECTION HAS AN IDD
 DEFINE IDD AS AN INTEGER VARIABLE
 EVERY T-INSPECTION,INCREASE HAS AN IDTI
 DEFINE IDTI AS AN INTEGER VARIABLE
 EVERY REPAIR HAS AN IDREP AND AN NI
 DEFINE IDREP AS AN INTEGER VARIABLE
 DEFINE NI AS AN INTEGER VARIABLE
 EVERY FAILURE HAS AN IDFA
 DEFINE IDFA AS AN INTEGER VARIABLE
 EVERY REACH-FAIL,SAFE-LGT HAS AN IDRFS
 DEFINE IDRFS AS AN INTEGER VARIABLE
 EVERY RETIRE,FROM-SERVICE HAS AN IDRET
 DEFINE IDRET AS AN INTEGER VARIABLE

PRIORITY ORDER IS IMMEDIATE,FLEET,INSPECTION,
 INCREASE,INSPECTION,FREQUENCY, REPAIR,
 DECISION,ON,MOD, D-LEVEL,INSPECTION,
 C-LEVEL,INSPECTION, B-LEVEL,INSPECTION

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ ..KRONOS 2.1.2 .03/02/78. 18.36.02. PAGE 4

160
161
162
**
END

SYMBOLIC REFERENCE MAP (R = 1) PREAMBLE

AACA	PROCEDURE	REAL	1 REFS
AACR	PROCEDURE	REAL	1 REFS
AAL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ABCA	PROCEDURE	REAL	1 REFS
ABCD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
ABCD.OLD.D	PROCEDURE	REAL	1 REFS
ABCR	PROCEDURE	REAL	1 REFS
ABC.OLD	GLOBAL VARIABLE	REAL	1 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACA	PROCEDURE	REAL	2 REFS
ACCA	PROCEDURE	REAL	1 REFS
ACCR	PROCEDURE	REAL	1 REFS
ACID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACRKL	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
ACTIVE.FLEET	SET	REAL	1 REFS
ACTUAL.AVG.FAT.LIFE	GLOBAL VARIABLE	REAL	1 REFS
AC.FAIL.RATE	GLOBAL VARIABLE	REAL	1 REFS
ADCA	PROCEDURE	REAL	1 REFS
ADCR	PROCEDURE	REAL	1 REFS
ADL	PROCEDURE	(1-D) INTEGER	1 REFS
AD.MD.COST	GLOBAL VARIABLE	REAL	1 REFS
AD.TOOLING	GLOBAL VARIABLE	REAL	1 REFS
AEXP	GLOBAL VARIABLE	REAL	1 REFS
AF	GLOBAL VARIABLE	REAL	1 REFS
AFAC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AIL	GLOBAL VARIABLE	(1-D) REAL	1 REFS
AIRCRAFT	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
AIRFRAME.TIME	TEMPORARY ENTITY	REAL	1 REFS
AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AMEAN	GLOBAL VARIABLE	REAL	1 REFS
APID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AP.FAIL.RATE	GLOBAL VARIABLE	REAL	1 REFS
APSEL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ASCA	PROCEDURE	REAL	1 REFS
ASCR	PROCEDURE	REAL	1 REFS
ATII	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AVFSA	GLOBAL VARIABLE	(1-D) REAL	1 REFS
AVEL	GLOBAL VARIABLE	REAL	1 REFS
AVPSA	GLOBAL VARIABLE	(1-D) REAL	1 REFS
AV.AC.FAIL	GLOBAL VARIABLE	REAL	1 REFS
AV.AP.FAIL	GLOBAL VARIABLE	REAL	1 REFS
AV.EL.FAIL	GLOBAL VARIABLE	REAL	1 REFS
AICO	PROCEDURE	REAL	1 REFS
AICR	PROCEDURE	REAL	1 REFS
AIE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AISO	PROCEDURE	REAL	1 REFS
AISP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZSP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS

A-LEVEL-INSPECTION	EVENT NOTICE	1 REFS
A-REPAIR-COST	GLOBAL VARIABLE	1 REFS
B	GLOBAL VARIABLE	1 REFS
BA	GLOBAL VARIABLE	1 REFS
BCA	GLOBAL VARIABLE	2 REFS
BCRKL	GLOBAL VARIABLE	2 REFS
BEEN-MODIFIED	GLOBAL VARIABLE	1 REFS
BEGIN-PRODUCTION	GLOBAL VARIABLE	1 REFS
BIRTH-DEFECT-PRORABT	GLOBAL VARIABLE	1 REFS
B-LEVEL-INSPECTION	GLOBAL VARIABLE	1 REFS
B-REPAIR-COST	EVENT NOTICE	2 REFS
CACBD	GLOBAL VARIABLE	1 REFS
CCA	GLOBAL VARIABLE	1 REFS
CCRKL	GLOBAL VARIABLE	2 REFS
CDM-RATE-CHANGE	GLOBAL VARIABLE	1 REFS
CERI	GLOBAL VARIABLE	1 REFS
CHG-FREQ-TIME	GLOBAL VARIABLE	1 REFS
CHG-TIME	GLOBAL VARIABLE	1 REFS
CINSL	GLOBAL VARIABLE	2 REFS
CKREP-TIME	GLOBAL VARIABLE	1 REFS
CLST	GLOBAL VARIABLE	1 REFS
CONE	GLOBAL VARIABLE	1 REFS
COREP-TIME	GLOBAL VARIABLE	1 REFS
COROSTON	EVENT NOTICE	1 REFS
COST-OF-REPAIRS	GLOBAL VARIABLE	1 REFS
COUNT-ELEMENT	GLOBAL VARIABLE	1 REFS
CO-EXISTS	GLOBAL VARIABLE	1 REFS
CRASHED-FLEET	SET	2 REFS
CRR	GLOBAL VARIABLE	1 REFS
CRPF	GLOBAL VARIABLE	1 REFS
CR-CTR	GLOBAL VARIABLE	1 REFS
CTHREE	GLOBAL VARIABLE	1 REFS
CTWO	GLOBAL VARIABLE	1 REFS
C28	GLOBAL VARIABLE	1 REFS
C29	GLOBAL VARIABLE	1 REFS
C7	GLOBAL VARIABLE	1 REFS
C-GROWTH-RATE	GLOBAL VARIABLE	1 REFS
C-INT	GLOBAL VARIABLE	1 REFS
C-INTERVAL	GLOBAL VARIABLE	1 REFS
C-LEVEL-INSPECTION	EVENT NOTICE	2 REFS
C-PROB	GLOBAL VARIABLE	1 REFS
C-REPAIR-COST	GLOBAL VARIABLE	1 REFS
DCA	GLOBAL VARIABLE	2 REFS
DCHKL	GLOBAL VARIABLE	2 REFS
DECISION-ON-MOD	EVENT NOTICE	1 REFS
DEC-INT	GLOBAL VARIABLE	1 REFS
DEC-ON-MOD-SCH	GLOBAL VARIABLE	1 REFS
DINSL	GLOBAL VARIABLE	2 REFS
DLL	GLOBAL VARIABLE	1 REFS
D-EXT	GLOBAL VARIABLE	1 REFS
D-IN	GLOBAL VARIABLE	1 REFS
D-INTERVAL	GLOBAL VARIABLE	1 REFS
D-INT-FIND	GLOBAL VARIABLE	1 REFS

O-LEVEL-INSPECTION	EVENT NOTICE	REAL	2 REFS
D-REPAIR-COST	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
ELEMENT	GLOBAL VARIABLE	(2-D) ALPHA	1 REFS
ELID	GLOBAL VARIABLE	REAL	1 REFS
ELIYP-FAIL-RATE	EVENT NOTICE	REAL	1 REFS
ENTER-SERVICE	TEMPORARY ATTRIBUTE	REAL	1 REFS
ENTRY-TIME	GLOBAL VARIABLE	ALPHA	1 REFS
EXT-INSPECTION-LEVEL	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
EXT-LVL-INSPECTION	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
E1	GLOBAL VARIABLE	ALPHA	1 REFS
E2	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
E3	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
E4	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
FAILURE	EVENT NOTICE	INTEGER	1 REFS
FAIL-OPT	GLOBAL VARIABLE	INTEGER	1 REFS
FOCK	GLOBAL VARIABLE	INTEGER	1 REFS
FIXIT-COST	GLOBAL VARIABLE	REAL	2 REFS
FLEET-RETIRED	SET	REAL	1 REFS
FLEET-STR-REV	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
FLTHR	GLOBAL VARIABLE	REAL	1 REFS
FREQ-DECREASE	GLOBAL VARIABLE	REAL	1 REFS
FSAF-ALGT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
FSH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
GACA	GLOBAL VARIABLE	REAL	2 REFS
GACRK	GLOBAL VARIABLE	REAL	2 REFS
GSCA	GLOBAL VARIABLE	REAL	2 REFS
GSCRK	GLOBAL VARIABLE	REAL	2 REFS
GCCA	GLOBAL VARIABLE	REAL	2 REFS
GCCPK	GLOBAL VARIABLE	REAL	2 REFS
GSCA	GLOBAL VARIABLE	REAL	2 REFS
GSCRK	GLOBAL VARIABLE	REAL	2 REFS
GSCOR	GLOBAL VARIABLE	REAL	2 REFS
GOICO	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
GOICR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
GOPD	GLOBAL VARIABLE	INTEGER	1 REFS
GOSCO	GLOBAL VARIABLE	INTEGER	1 REFS
GOSCR	GLOBAL VARIABLE	INTEGER	1 REFS
GOSDM	GLOBAL VARIABLE	INTEGER	1 REFS
GOICR	GLOBAL VARIABLE	REAL	2 REFS
GSCA	GLOBAL VARIABLE	REAL	2 REFS
GSCRK	GLOBAL VARIABLE	REAL	2 REFS
GICOR	GLOBAL VARIABLE	REAL	2 REFS
GICR	GLOBAL VARIABLE	REAL	2 REFS
GISD	GLOBAL VARIABLE	REAL	2 REFS
MI-TIME-ACRFT	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
HZ	GLOBAL VARIABLE	(1-D) REAL	1 REFS
HZRD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
ID	TEMPORARY ATTRIBUTE	INTEGER	1 REFS
IDA	TEMPORARY ATTRIBUTE	INTEGER	2 REFS
IDB	TEMPORARY ATTRIBUTE	INTEGER	2 REFS
IDC	TEMPORARY ATTRIBUTE	INTEGER	2 REFS
IDCK	GLOBAL VARIABLE	INTEGER	1 REFS

LSCR	GLOBAL VARIABLE	REAL	1 REFS
LTHO	GLOBAL VARIABLE	ALPHA	1 REFS
LICO	GLOBAL VARIABLE	REAL	1 REFS
LICR	GLOBAL VARIABLE	REAL	1 REFS
LISD	GLOBAL VARIABLE	REAL	1 REFS
L-EXT	GLOBAL VARIABLE	REAL	1 REFS
MACA	PROCEDURE	REAL	1 REFS
MCCR	PROCEDURE	REAL	1 REFS
MBCA	PROCEDURE	REAL	1 REFS
MBCP	PROCEDURE	REAL	1 REFS
MCCA	PROCEDURE	REAL	1 REFS
MCCP	PROCEDURE	REAL	1 REFS
MCCA	PROCEDURE	REAL	1 REFS
MDCP	PROCEDURE	REAL	1 REFS
MFI1	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MFR1.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MFR2	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MFR2.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MODEL	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
MOD.NO	GLOBAL VARIABLE	INTEGER	1 REFS
MOD.SAVE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
MOD.TESTED	GLOBAL VARIABLE	ALPHA	1 REFS
MRDO	GLOBAL VARIABLE	REAL	1 REFS
MSCA	PROCEDURE	REAL	1 REFS
MSCP	PROCEDURE	REAL	1 REFS
MSPI	PROCEDURE	REAL	1 REFS
MSPI1.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSR2	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSR2.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MU.P	GLOBAL VARIABLE	REAL	1 REFS
MICO	PROCEDURE	REAL	1 REFS
MICR	PROCEDURE	REAL	1 REFS
MISD	PROCEDURE	REAL	1 REFS
M1.MEAN	GLOBAL VARIABLE	REAL	1 REFS
M2.MEAN	GLOBAL VARIABLE	REAL	1 REFS
M3.MEAN	GLOBAL VARIABLE	REAL	1 REFS
M4.MEAN	GLOBAL VARIABLE	REAL	1 REFS
NACA	GLOBAL VARIABLE	REAL	1 REFS
NACP	GLOBAL VARIABLE	REAL	1 REFS
NBCA	GLOBAL VARIABLE	REAL	1 REFS
NBCP	GLOBAL VARIABLE	REAL	1 REFS
NCCA	GLOBAL VARIABLE	REAL	1 REFS
NCCR	GLOBAL VARIABLE	REAL	1 REFS
NCINS	GLOBAL VARIABLE	REAL	1 REFS
NCZ	GLOBAL VARIABLE	REAL	1 REFS
NCZO	GLOBAL VARIABLE	REAL	1 REFS
NCCA	GLOBAL VARIABLE	REAL	1 REFS
NCCR	GLOBAL VARIABLE	REAL	1 REFS
NDINS	GLOBAL VARIABLE	REAL	1 REFS
NI	TEMPORARY ATTRIBUTE	REAL	1 REFS
NICMG	GLOBAL VARIABLE	INTEGER	2 REFS
NMD	GLOBAL VARIABLE	INTEGER	1 REFS

MN	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
MOAC	GLOBAL VARIABLE		INTEGER	1 REFS
NOE	GLOBAL VARIABLE		INTEGER	1 REFS
NO-FAIL	GLOBAL VARIABLE		INTEGER	1 REFS
NRFS	GLOBAL VARIABLE		REAL	1 REFS
NSCA	GLOBAL VARIABLE		REAL	1 REFS
NSCR	GLOBAL VARIABLE		INTEGER	1 REFS
NSFL	GLOBAL VARIABLE		INTEGER	1 REFS
NSIC	GLOBAL VARIABLE		INTEGER	1 REFS
NSMD	GLOBAL VARIABLE		INTEGER	1 REFS
NTIM	GLOBAL VARIABLE		INTEGER	1 REFS
NICO	GLOBAL VARIABLE		REAL	1 REFS
MICR	GLOBAL VARIABLE		REAL	1 REFS
MISD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
OCCUR-MOD	GLOBAL VARIABLE		INTEGER	1 REFS
OCOR	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
OICO	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
OICR	GLOBAL VARIABLE		INTEGER	1 REFS
OLD-SAMP	GLOBAL VARIABLE		INTEGER	1 REFS
OPD	GLOBAL VARIABLE		INTEGER	1 REFS
CSCO	GLOBAL VARIABLE		INTEGER	1 REFS
OSCR	GLOBAL VARIABLE		INTEGER	1 REFS
OSDM	GLOBAL VARIABLE		INTEGER	1 REFS
OICP	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
PFID	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
PFTH	GLOBAL VARIABLE		REAL	1 REFS
POP SIZE	GLOBAL VARIABLE		REAL	1 REFS
PRCMG	GLOBAL VARIABLE		REAL	1 REFS
PREDICTED-LIFE	GLOBAL VARIABLE		ALPHA	1 REFS
PREVIOUSLY-MODIFIED	GLOBAL VARIABLE		REAL	1 REFS
PRODUCTION-TIME	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
PSAVE	PROCEDURE		REAL	1 REFS
RATE	EVENT NOTICE		REAL	1 REFS
REACH-FAIL-SAFE-LGT	EVENT NOTICE		ALPHA	1 REFS
REPAIR	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
RETIRE-FROM-SERVICE	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
RETRO	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
RLGT	GLOBAL VARIABLE		REAL	1 REFS
SACA	GLOBAL VARIABLE		REAL	1 REFS
SACID	GLOBAL VARIABLE		REAL	1 REFS
SACR	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
SAMP	GLOBAL VARIABLE		INTEGER	1 REFS
SAMPLING	GLOBAL VARIABLE		REAL	1 REFS
SAMP-SIZE	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
SAPID	GLOBAL VARIABLE		REAL	1 REFS
SRCA	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
SBCR	GLOBAL VARIABLE		REAL	1 REFS
SC	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
SCA	GLOBAL VARIABLE		REAL	2 REFS
SCCA	GLOBAL VARIABLE		REAL	1 REFS
SCCR	GLOBAL VARIABLE		REAL	1 REFS
SCRRL	GLOBAL VARIABLE		REAL	2 REFS
SD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS

SDCA	GLOBAL VARIABLE	REAL	1 REFS
SDCP	GLOBAL VARIABLE	REAL	1 REFS
SDIF	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SDM.OCCURRENCE.RATE	GLOBAL VARIABLE	REAL	1 REFS
SD.SCH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
SELN6	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
SEL1	GLOBAL VARIABLE	ALPHA	1 REFS
SEL2	GLOBAL VARIABLE	ALPHA	1 REFS
SEL3	GLOBAL VARIABLE	ALPHA	1 REFS
SF	GLOBAL VARIABLE	REAL	1 REFS
SFLTHR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SIG.R	GLOBAL VARIABLE	REAL	1 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE	INTEGER	1 REFS
SMOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
SM.CMK	GLOBAL VARIABLE	ALPHA	1 REFS
SNMD	GLOBAL VARIABLE	INTEGER	1 REFS
SNRFS	GLOBAL VARIABLE	INTEGER	1 REFS
SNSFL	GLOBAL VARIABLE	INTEGER	1 REFS
SNSIC	GLOBAL VARIABLE	INTEGER	1 REFS
SNSMO	GLOBAL VARIABLE	INTEGER	1 REFS
SRES	GLOBAL VARIABLE	INTEGER	1 REFS
SERATE	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SSCA	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SSCP	GLOBAL VARIABLE	REAL	1 REFS
SSTAN	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
SSTIM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
START.TEST	GLOBAL VARIABLE	REAL	1 REFS
STDL	GLOBAL VARIABLE	REAL	1 REFS
STIM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SU	GLOBAL VARIABLE	REAL	1 REFS
SICO	GLOBAL VARIABLE	REAL	1 REFS
SICR	GLOBAL VARIABLE	REAL	1 REFS
SISD	GLOBAL VARIABLE	REAL	1 REFS
S.FREQ.CHG	GLOBAL VARIABLE	REAL	1 REFS
S.REPAIR.COST	GLOBAL VARIABLE	REAL	1 REFS
TAIL.TO	TEMPORARY ATTRIBUTE	INTEGER	2 REFS
TEST.ACCEL.FACT	GLOBAL VARIABLE	REAL	1 REFS
TEST.LIFF	GLOBAL VARIABLE	REAL	1 REFS
TES.FAILURE	GLOBAL VARIABLE	ALPHA	1 REFS
TLID	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
THOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
TO.BE.MODIFIED	GLOBAL VARIABLE	INTEGER	1 REFS
TRCMG	GLOBAL VARIABLE	(1-D) REAL	1 REFS
T.FREQ.CHG	GLOBAL VARIABLE	REAL	1 REFS
T.IMPLEMENT.MOD	EVENT NOTICE	REAL	1 REFS
T.INSPECTION.INCREAS	EVENT NOTICE	REAL	1 REFS
T.LAST.O	GLOBAL VARIABLE	(1-D) REAL	1 REFS
USAGE.LIFE	GLOBAL VARIABLE	REAL	1 REFS
XACA	GLOBAL VARIABLE	REAL	1 REFS
XACR	GLOBAL VARIABLE	REAL	1 REFS
XSCA	GLOBAL VARIABLE	REAL	1 REFS
XBCR	GLOBAL VARIABLE	REAL	1 REFS
XCCA	GLOBAL VARIABLE	REAL	1 REFS

XCCR	GLOBAL VARIABLE	REAL	1 REFS
XCINS	GLOBAL VARIABLE	REAL	1 REFS
XDCA	GLOBAL VARIABLE	REAL	1 REFS
XDCR	GLOBAL VARIABLE	REAL	1 REFS
XDINS	GLOBAL VARIABLE	REAL	1 REFS
XPN	(1-D) GLOBAL VARIABLE	REAL	1 REFS
XSCA	GLOBAL VARIABLE	REAL	1 REFS
XSCR	GLOBAL VARIABLE	REAL	1 REFS
XICR	GLOBAL VARIABLE	REAL	1 REFS
XISD	GLOBAL VARIABLE	REAL	1 REFS
IAAFL	GLOBAL VARIABLE	REAL	1 REFS
IABCD	GLOBAL VARIABLE	REAL	1 REFS
ICORT	GLOBAL VARIABLE	REAL	2 REFS
ICRKT	GLOBAL VARIABLE	REAL	2 REFS
ISDT	GLOBAL VARIABLE	REAL	2 REFS
1ST.MD.COST	GLOBAL VARIABLE	REAL	1 REFS
1ST.TOOLING	GLOBAL VARIABLE	REAL	1 REFS
1.CDM.OCCURRENCE.PAT	GLOBAL VARIABLE	REAL	1 REFS
1.CR.EXISTS	GLOBAL VARIABLE	REAL	1 REFS
1.INT	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
1.ITE	EVENT NOTICE	(1-D) ALPHA	1 REFS
1.NUM.OF.RETIRE	GLOBAL VARIABLE	INTEGER	1 REFS
1.PROB	GLOBAL VARIABLE	REAL	1 REFS
1.STRENGTH.REDUCTION	EVENT NOTICE	REAL	1 REFS
2.COM.OCCURRENCE.PAT	GLOBAL VARIABLE	REAL	1 REFS
2.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2.INT	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2.ITE	EVENT NOTICE	INTEGER	1 REFS
2.NUM.OF.CRASH	GLOBAL VARIABLE	REAL	1 REFS
2.PRODUCTION.TIME	GLOBAL VARIABLE	REAL	1 REFS
2.STRENGTH.REDUCTION	EVENT NOTICE	REAL	1 REFS

```

1  MAIN
2
3  DEFINE S.OPT AS AN ALPHA VARIABLE
4  DEFINE S1.S2.S3.S4.S5.S6.S7.S8.S9.S10.S11.S12
5  AS INTEGER VARIABLES
6  LET COUNT.ELEMENT = 0
7
8  ** RESERVE AIRCRAFT ARRAYS
9
10 RESERVE HI.TIME.ACPT(*) , MRDD(*) AS 10
11 RESERVE SC(*) , SD(*) , SAMP(*) , MOD.SAVE(*) , CHG.TIME(*) AS 30
12 RESERVE APID(*) , FLTHR(*) AS 10
13 RESERVE APID(*) , STIM(*) AS 20
14 RESERVE SAPID(*) , SFLTHR(*) , SSTAN(*) AS 100
15 RESERVE SAPID(*) , SSTIM(*) , SELNH(*) AS 200
16 RESERVE MODEL(*) AS 2 AND ELEMENT(*) AS 4
17 RESERVE ABC(*) , LABCD(*) , CABCD(*) AS 4
18 RESERVE E1(*) , E2(*) , E3(*) , E4(*) AS 25
19 RESERVE FSAVE(*) , PSAVE(*) , AVPSA(*) , AVFSA(*) AS 5
20 FOR I = 1 TO 5
21 DO
22 LET FSAVE(I)=0.0
23 LET PSAVE(I)=0.0
24 LET AVPSA(I)=0.0
25 LET AVFSA(I)=0.0
26 LOOP
27 LET NO.FAIL = 0
28
29 ** READ AIRCRAFT INPUT
30
31 READ MODEL(1) , MODEL(2) AS 2 4 4
32 READ SIZE.OF.FLEET , USAGE.LIFE
33 READ BEGIN.PRODUCTION , PRODUCTION.TIME , 2.PRODUCTION.TIME , PRCHG
34 READ START.TEST , 1ST.ACCEL.FACT , C.GROWTH.RATE
35 READ C7 , C28 , C29 , MU.R , SIG.R , DLL
36 READ LABCD(1) , LABCD(2)
37 READ CABCD(1) , CABCD(2) , CABCD(3) , CABCD(4)
38 READ LONG.LIST.S.OPT , FAIL.OPT , FAT.TEST.FACTOR
39 READ ACTUAL.AVG.FAT.LIFE , LEAD.TIME
40 READ T.FREQ.CHG , S.FREQ.CHG , FREQ.DECREASE
41 READ A.REPAIR.COST , R.REPAIR.COST , C.REPAIR.COST , D.REPAIR.COST
42 READ 1ST.TOOLING.AD , TOOLING.1ST.MD.COST , AD.MD.COST
43 READ S.REPAIR.COST , SU.SF
44 LET A.REPAIR.COST = 2.0*A.REPAIR.COST
45 LET B.REPAIR.COST = 2.0*B.REPAIR.COST
46 LET C.REPAIR.COST = 2.0*C.REPAIR.COST
47 LET D.REPAIR.COST = 2.0*D.REPAIR.COST
48 LET S.REPAIR.COST = 2.0*S.REPAIR.COST
49 IF S.OPT="YES"
50 READ S1.S2.S3.S4.S5.S6.S7.S8.S9.S10
51 LET SEED.V(1)=S1
52 LET SEED.V(2)=S2
53 LET SEED.V(3)=S3
54 LET SEED.V(4)=S4

```

```

54 LET SEED.V(5)=S5
55 LET SEED.V(6)=S6
56 LET SEED.V(7)=S7
57 LET SEED.V(8)=S8
58 LET SEED.V(9)=S9
59 LET SEED.V(10)=S10
60 ALWAYS
61 LET DLT=OLL*1
62 LET T.FREQ.CHG = 1.0 - T.FREQ.CHG
63 LET S.FREQ.CHG = 1.0 - S.FREQ.CHG
64 LET FREQ.DECREASE = 1.0 + FREQ.DECREASE
65 IF LONG.LIST = "YES"
66 READ NOE
67 RESERVE ELID(1) AS 4 BY NOE AND NOAC(1) AS NOE
68 RESERVE TLID(1) AS NOE BY *
69 FOR I = 1 TO NOE
70 DO
71 START NEW CARD
72 READ ELID(1), ELID(2), ELID(3), ELID(4), ELID(5) AS 4 4
73 READ NOAC(1)
74 RESERVE TLID(1) AS NOAC(1)
75 ALSO FOR J = 1 TO NOAC(1)
76 DO
77 READ TLID(1,J)
78 LOOP
79 ALWAYS
80
81 ## RESERVE ELEMENT ARRAYS
82
83 RESERVE OICR(1), OICR(2), OICR(3), OICR(4), OICR(5) AS 4
84 RESERVE AIRPLANE(1), TMOD.PENDING(1), SHOD.PENDING(1) AS SIZE.OF.FLEET
85 RESERVE C.INTERVAL(1), AIL(1), AMPL(1), FSH(1) AS SIZE.OF.FLEET
86 RESERVE D.INTERVAL(1), OCCUR.MOD(1), CREP.TIME(1) AS SIZE.OF.FLEET
87 RESERVE COREP.TIME(1), I.CR.EXISTS(1), P.CR.EXISTS(1) AS SIZE.OF.FLEET
88 RESERVE CO.EXISTS(1), INSP.SCH(1), SD.SCH(1) AS SIZE.OF.FLEET
89 RESERVE AISR(1), AZSR(1), AF(1), AC(1), ATII(1) AS SIZE.OF.FLEET
90 RESERVE AAL(1), ABL(1), ACL(1), ADL(1) AS SIZE.OF.FLEET
91 RESERVE I.INT(1), Z.INT(1), C.INT(1) AS SIZE.OF.FLEET
92 RESERVE MSR1(1), MFR1(1), MSR2(1), MFR2(1), LAST.SD(1) AS SIZE.OF.FLEET
93 RESERVE MSR1.INT(1), MFR1.INT(1), MSR2.INT(1), MFR2.INT(1)
94 AS SIZE.OF.FLEET
95 RESERVE ARM(1), CGRI(1) AS SIZE.OF.FLEET
96 RESERVE IE1(1), IE2(1) AS SIZE.OF.FLEET
97 RESERVE ALE, AZE AS SIZE.OF.FLEET
98 RESERVE INT.LGT(1) AS SIZE.OF.FLEET
99 RESERVE T.LAST.D(1) AS SIZE.OF.FLEET
100 RESERVE D.INT(1) AS SIZE.OF.FLEET
101 RESERVE D.EXT(1) AS SIZE.OF.FLEET
102 RESERVE SRATE(1), TRCHG(1), SRES(1), NZRO(1), RLGT(1), SDIF(1) AS SIZE.OF.FLEET
103 RESERVE MZ(1), CLGT(1), PFID(1), PFTIM(1) AS 2000
104 RESERVE AFAC(1) AS SIZE.OF.FLEET
105
106 ## READ ELEMENT INPUT

```

```

107 READP ADD 1 TO COUNT.ELEMENT
108 START NEW CARD
109 READ ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4) AS 4 A 4
110
111 ** INITIALIZE GLOBAL VARIABLES
112 **
113 IF SEL1 NE ELEMENT(1) OR SEL2 NE ELEMENT(2) OR SEL3 NE ELEMENT(3)
114 IF COUNT.ELEMENT GT 1
115 LET ESUN = ELAMB
116 LET ELAMB = 0.0
117 IF FAIL.OPT = 2 OR FAIL.OPT = 3
118 CALL ESTIMATE.FAILURE.RATE YIELDING ELTP.FAIL.RATE
119 ALWAYS
120 IF FAIL.OPT = 1 OR FAIL.OPT = 3
121 LET AV.EL.FAIL = FSUM*POP.SIZE/SAMP.SIZE
122 ALWAYS
123 IF SEL1 = "FUS-"
124 IF SEL2 = "MFR-" OR SEL3 = "FMW-"
125 IF FAIL.OPT = 2 OR FAIL.OPT = 3
126 ADD ELTP.FAIL.RATE TO AP.FAIL.RATE
127 ALWAYS
128 IF FAIL.OPT = 1 OR FAIL.OPT = 3
129 ADD AV.EL.FAIL TO AV.AP.FAIL
130 ALWAYS
131 JUMP AHEAD
132 ELSE
133 ALWAYS
134 IF FAIL.OPT = 2 OR FAIL.OPT = 3
135 ADD ELTP.FAIL.RATE TO AC.FAIL.RATE
136 ALWAYS
137 IF FAIL.OPT = 1 OR FAIL.OPT = 3
138 ADD AV.EL.FAIL TO AV.AC.FAIL
139 ALWAYS
140 HERE
141 IF LONG.LIST = "NU"
142 CALL SUMMARY
143 ALWAYS
144 REGARDLESS
145 IF ELEMENT(1) = "E00 "
146 GO TO EOJ
147 OTHERWISE
148 LET SEL1 = ELEMENT(1)
149 LET SEL2 = ELEMENT(2)
150 LET SEL3 = ELEMENT(3)
151 CALL SUM.INITIALIZE
152 REGARDLESS
153 READ PREDICTED.LIFE, M1.MEAN, M2.MEAN
154 READ LGT.TO.FAILURE, CONE, FSAF.LGT, BIRTH.DEFECT.PROBABILITY
155 READ CRR, SUM.OCCURRENCE.RATE
156 READ 1.PROB, C.PROB, INT.LVL.INSP, EXT.LVL.INSP
157 READ MOD.TESTED
158 READ LOCATED.IN.STRESS.CON
159

```

READ 1.CDM.OCCURRENCE.RATE, 2.CDM.OCCURRENCE.WATE, CDM.RATE.CHANGE

READ L.EXI.M3.MEAN, M4.MEAN, CDMO.CTHREE

READ INT.CONE, INT.CTWO, INT.CTHREE

READ IABCD(3), IABCD(4), POP.SIZE

READ AHEAD, B

LET ACTUAL.AVG.FAT.LIFE = 0.0

LET SAMPLING = 48000 / IABCD(4)

LET INT.M1 = M1.MEAN

LET INT.M2 = M2.MEAN

LET INT.M3 = M3.MEAN

LET INT.M4 = M4.MEAN

IF LONG.LIST = "NO"

START NEW PAGE

PRINT 1 LINE AS FOLLOWS

FOR I = 1 TO 10

DO

PRINT 1 LINE WITH I, SEED.V(I) AS FOLLOWS

SEED(**) = *****

LOOP

SKIP 2 OUTPUT LINES

ALWAYS

LET SAMPLING = SAMPLING

CALL INITIALIZATION

LET M1.MEAN = M1.MEAN*CRHF

LET M2.MEAN = M2.MEAN*CRHF

LET M3.MEAN = M3.MEAN * CRHF

LET M4.MEAN = M4.MEAN * CRHF

LET INT.M1 = INT.M1 * CRHF

LET INT.M2 = INT.M2 * CRHF

LET INT.M3 = INT.M3 * CRHF

LET INT.M4 = INT.M4 * CRHF

LET MOD.SAVE(I) = 0

IF LONG.LIST = "YES"

LET LTHO="YES"

FOR I = 1 TO MOE

DO

IF ELEMENT(1) = ELID(1,1) AND ELEMENT(2) = ELID(2,1) AND ELEMENT(3) =

ELID(3,1) AND ELEMENT(4) = ELID(4,1)

LET LDX = I

LEAVE

ELSE

LOOP

ALWAYS

IF ACTUAL.AVG.FAT.LIFE = 0.0

CALL REAL.LIFE(MU.R.SIG.R,PREDICTED.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE

REGARDLESS

LET IARFL = ACTUAL.AVG.FAT.LIFE

IF RANDOM.F(3) LE FAT.TEST.FACTOR

LET TEST.LIFE = 9999999.

JUMP AHEAD

CDC 6600 CACI SIMSCRIPT II.5 VERSION /A.0-00/ KRONOS 2.1-2 03/02/78. 18.36.02. PAGE 17

```

211 ALWAYS
212 CALL FATIGUE.LIFE,SCATTER(ACTUAL.AVG,FAT.LIFE,3.3.97)YIELDING FIRST.LIFE
213 AND SECOND.LIFE
214 LET TEST.LIFE = FIRST.LIFE
215 LET NPTS = (SIZE-OF-FLEET + 1 - TRUNC.F(PRCHG/PRODUCTION.TIME)) *
216 2-PRODUCTION.TIME
217 LET SAIL = START.TEST + FIRST.LIFE/TEST.ACCEL.FACT + LEAD.TIME
218 IF SAIL LT BEGIN.PRODUCTION + PRCHG + NPTS + USAGE.LIFE
219 IF PREDICTED.LIFE LT 2.0 * USAGE.LIFE AND FIRST.LIFE LT PREDICTED.LIFE OR
220 PREDICTED.LIFE GE 2.0 * USAGE.LIFE AND FIRST.LIFE LT 2.0 * USAGE.LIFE
221 SCHEDULE A T.IMPLEMENT.MOD AT SAIL
222 LET TES.FAILURE = "YES"
223 REGARDLESS
224 HERE
225 IF LTHO = "YES"
226 START NEW PAGE
227 WRITE MODEL(1), MODEL(2) AS S 50, "AIRCRAFT TYPE: ", 2 A 4
228 SKIP 2 OUTPUT LINES
229 PRINT 1 DOUBLE LINE WITH SIZE-OF-FLEET, USAGE.LIFE AS FOLLOWS
230 NUMBER OF AIRCRAFT IN FLEET: ***** AIRCRAFT SERVICE LIFE: ***** HOURS
231 SKIP 1 OUTPUT LINE
232 WRITE ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4)
233 AS S 45, "STRUCTURAL ELEMENT: ", 4 A 4
234 SKIP 2 OUTPUT LINES
235 IF PREDICTED.LIFE = 0.0
236 PRINT 1 DOUBLE LINE WITH 1AFL AS FOLLOWS
237 ELEMENT DESIGN FATIGUE TESTED PRIOR TO PRODUCTION
238 JUMP AHEAD
239 ELSE
240 PRINT 1 DOUBLE LINE WITH PREDICTED.LIFE, 1AFL AS FOLLOWS
241 PREDICTED AVERAGE FATIGUE LIFE: ***** HOURS
242 HERE
243 PRINT 1 LINE AS FOLLOWS
244 INITIAL INSPECTION INTERVALS
245 SKIP 1 OUTPUT LINE
246 PRINT 4 LINES WITH 1ABCD(1), 1ABCD(2), 1ABCD(3), 1ABCD(4) AS FOLLOWS
247 A-LEVEL ***** HOURS
248 B-LEVEL ***** HOURS
249 C-LEVEL ***** HOURS
250 D-LEVEL ***** HOURS
251 ALWAYS
252 SCHEDULE AN ENTER.SERVICE AT BEGIN.PRODUCTION
253 #
254 START SIMULATION
255 START NEW PAGE
256 LET PROB = 0.0
257 LET SAMP.SIZE = SAMP.SIZE + 1.0
258 FOR I = NCZO TO NCZ
259 DO
260 LET PF = 1.0 - HZ(1)
261 LET PROB = PROB + PF
262

```

ACTUAL AVERAGE FATIGUE LIFE: ***** HOURS

ACTUAL AVERAGE FATIGUE LIFE: ***** HOURS

```

256 LOOP
257 LET T = SIZE-OF-FLEET * USAGE-LIFE
258 LET LAMBDA = PROB/T
259 LET ELAMB = ELAMB*LAMBDA
260 IF LONG-LIST = "NO"
261 CALL DISPLAY-OUTPUT
262 ALWAYS
263 LET NCZO = NCZ +1
264 FOR EVERY AIRCRAFT IN CRASHED-FLEET
265 DO
266 REMOVE AIRPLANE(TAIL-ID) FROM CRASHED-FLEET
267 DESTROY THE AIRCRAFT CALLED AIRPLANE(TAIL-ID)
268 LOOP
269 FOR EVERY AIRCRAFT IN FLEET-RETIRED
270 DO
271 REMOVE AIRPLANE(TAIL-ID) FROM FLEET-RETIRED
272 DESTROY THE AIRCRAFT CALLED AIRPLANE(TAIL-ID)
273 LOOP
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302

```

GO READ
 REQJ# START NEW PAGE
 IF FAIL-OPT = 1 OR FAIL-OPT = 3
 WRITE AV.AC-FAIL AS
 "EST.AIRCRAFT FAILURE RATE OF FLIGHT LOADED STRUCTURE USING AVG:"
 *E(10,2)."/HR."
 SKIP 2 OUTPUT LINES
 WRITE AV.AP-FAIL AS
 "EST.AIRCRAFT FAILURE RATE OF PRESSURE LOADED STRUCTURE USING AVG:"
 *E(10,2)."/HR."
 SKIP 2 OUTPUT LINES
 ALWAYS
 IF FAIL-OPT = 2 OR FAIL-OPT = 3
 WRITE AC-FAIL-RATE AS "EST. AIRCRAFT FAILURE RATE OF FLIGHT LOADED STRUCTURE:"
 *E(10,2)."/HR."
 SKIP 2 OUTPUT LINES
 WRITE AP-FAIL-RATE AS "EST. AIRCRAFT FAILURE RATE OF PRESSURE LOADED STRUCTURE:"
 *E(10,2)."/HR."
 SKIP 2 OUTPUT LINES
 ALWAYS
 PRINT 1 LINE WITH AVFSA(1),AVFSA(2),AVFSA(3),AVFSA(4),AVFSA(5) AS FOLLOWS
 AVERAGE FLIGHT CRACKS ***
 PRINT 1 LINE WITH AVPSA(1),AVPSA(2),AVPSA(3),AVPSA(4),AVPSA(5) AS FOLLOWS
 AVERAGE PRESSURE CRACKS ***
 PRINT 1 LINE WITH FSAVE(1),FSAVE(2),FSAVE(3),FSAVE(4),FSAVE(5) AS FOLLOWS
 FLIGHT CRACKS ***
 PRINT 1 LINE WITH PSAVE(1),PSAVE(2),PSAVE(3),PSAVE(4),PSAVE(5) AS FOLLOWS
 PRESSURE CRACKS ***
 SKIP 2 OUTPUT LINES
 PRINT 1 LINE AS FOLLOWS
 FAILURE SUMMARY
 IF NO-FAIL = 0
 PRINT 1 LINE AS FOLLOWS
 NONE

```

303 JUMP AHEAD
304 ALWAYS
305 FOR I = 1 TO NO.FAIL
306 DO
307 WRITE E1(I),E2(I),E3(I),E4(I) AS A A A
308 LOOP
309 HERE
310 SKIP 2 OUTPUT LINES
311 PRINT 1 LINE AS FOLLOWS
END OF SIMULATION
312 STOP
313 END

```

SYMBOLIC REFERENCE MAP (R = 1) MAIN

112

AAL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ABCD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACTUAL-AVG-FAT-LIFE	GLOBAL VARIABLE	REAL	6 REFS
AC-FAIL-RATE	GLOBAL VARIABLE	REAL	2 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AD-MD-COST	GLOBAL VARIABLE	REAL	1 REFS
AD-TOOLING	GLOBAL VARIABLE	REAL	1 REFS
AF	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AFACT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
AIL	GLOBAL VARIABLE	(1-D) ALPHA	4 REFS
AIRCRAFT	TEMPORARY ENTITY		
AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	5 REFS
AMEAN	GLOBAL VARIABLE	REAL	1 REFS
APID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AP-FAIL-RATE	GLOBAL VARIABLE	REAL	2 REFS
ARFSL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ATII	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AVFSA	GLOBAL VARIABLE	(1-D) REAL	7 REFS
AVPSA	GLOBAL VARIABLE	(1-D) REAL	7 REFS
AV-AC-FAIL	GLOBAL VARIABLE	REAL	2 REFS
AV-AP-FAIL	GLOBAL VARIABLE	REAL	2 REFS
AV-EL-FAIL	GLOBAL VARIABLE	REAL	3 REFS
AIE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AISR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZSR	GLOBAL VARIABLE	REAL	3 REFS
A-REPAIR-COST	GLOBAL VARIABLE	REAL	1 REFS
B	GLOBAL VARIABLE	REAL	3 REFS
BEGIN-PRODUCTION	GLOBAL VARIABLE	REAL	3 REFS
BIRTH-DEFECT-PROBAB	GLOBAL VARIABLE	REAL	3 REFS
S-REPAIR-COST	GLOBAL VARIABLE	REAL	3 REFS
CABCD	GLOBAL VARIABLE	(1-D) REAL	5 REFS
CDP-RATE-CHANGE	GLOBAL VARIABLE	REAL	1 REFS
CGRI	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CHG-TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CKREP-TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CLGT	GLOBAL VARIABLE	REAL	1 REFS
CONF	GLOBAL VARIABLE	REAL	1 REFS
COREP-TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
COUNT-ELEMENT	GLOBAL VARIABLE	INTEGER	3 REFS
CO-EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
CRASHED-FLEET	GLOBAL VARIABLE	INTEGER	2 REFS
CRR	GLOBAL VARIABLE	INTEGER	1 REFS
CRUF	GLOBAL VARIABLE	REAL	4 REFS
CTHREE	GLOBAL VARIABLE	REAL	1 REFS
CTWO	GLOBAL VARIABLE	REAL	1 REFS
CZB	GLOBAL VARIABLE	REAL	1 REFS

```

C29      GLOBAL VARIABLE      REAL      1 REFS
CZ       GLOBAL VARIABLE      REAL      1 REFS
C.GROWTH.RATE      GLOBAL VARIABLE      REAL      1 REFS
C.INT      GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
C.INTERVAL      GLOBAL VARIABLE      ( 1-D) REAL      1 REFS
C.PROB      GLOBAL VARIABLE      REAL      1 REFS
C.REPAIR.COST      GLOBAL VARIABLE      REAL      3 REFS
DISPLAY.OUTPUT      PROCEDURE      INTEGER      1 REFS
DLL       GLOBAL VARIABLE      REAL      3 REFS
D.EXT     GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
D.IN      GLOBAL VARIABLE      ( 1-D) INTEGER      1 REFS
D.INTERVAL      GLOBAL VARIABLE      ( 1-D) REAL      1 REFS
D.REPAIR.COST      GLOBAL VARIABLE      REAL      3 REFS
ELEMEN     GLOBAL RECURSIVE      REAL      4 REFS
ELID      GLOBAL VARIABLE      ( 1-D) ALPHA      20 REFS
SLIYP.FAIL.RATE      GLOBAL VARIABLE      ( 2-D) ALPHA      9 REFS
ENTER.SERVICE      EVENT NOTICE      REAL      3 REFS
EOJ       UNSUBSCRIPTED LABEL      2 REFS
ESTIMATE.FAULTURE.RAT      PROCEDURE      INTEGER      1 REFS
ESUM      LOCAL RECURSIVE      REAL      2 REFS
EXT.LVL.INSR      GLOBAL VARIABLE      REAL      2 REFS
E1        GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
E2        GLOBAL VARIABLE      ( 1-D) ALPHA      2 REFS
E3        GLOBAL VARIABLE      ( 1-D) ALPHA      2 REFS
E4        GLOBAL VARIABLE      ( 1-D) ALPHA      2 REFS
FAIL.OPT   GLOBAL VARIABLE      INTEGER      17 REFS
FATIGUE.LIFE.SCATTER      PROCEDURE      1 REFS
FATTEST.FACTOR      LOCAL RECURSIVE      REAL      2 REFS
FIRST.LIFE      LOCAL RECURSIVE      REAL      5 REFS
FLEET.RETIRED      SET      2 REFS
FLTHR      GLOBAL VARIABLE      ( 1-D) INTEGER      1 REFS
FREQ.DECREASE      GLOBAL VARIABLE      REAL      3 REFS
FSAF.LGT   GLOBAL VARIABLE      REAL      1 REFS
FSAVE      GLOBAL VARIABLE      ( 1-D) REAL      7 REFS
FSH        GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
GOICO      GLOBAL VARIABLE      ( 1-D) INTEGER      1 REFS
GOICR      GLOBAL VARIABLE      ( 1-D) INTEGER      1 REFS
HI.TIME.ACRFT      GLOBAL VARIABLE      ( 1-D) INTEGER      1 REFS
HZ         GLOBAL VARIABLE      ( 1-D) REAL      2 REFS
HZRD       GLOBAL VARIABLE      ( 1-D) REAL      1 REFS
I          GLOBAL RECURSIVE      REAL      31 REFS
IE1        GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
IE2        GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
INITIALIZATION      PROCEDURE      INTEGER      1 REFS
INSP.SCH   GLOBAL VARIABLE      ( 1-D) ALPHA      1 REFS
INT.COME   GLOBAL VARIABLE      REAL      1 REFS
INT.CTWC   GLOBAL VARIABLE      REAL      1 REFS
INT.LGT    GLOBAL VARIABLE      REAL      1 REFS
INI.LVL.INSR      GLOBAL VARIABLE      ( 1-D) REAL      1 REFS
INT.M1     GLOBAL VARIABLE      ALPHA      1 REFS
INT.M2     GLOBAL VARIABLE      REAL      3 REFS
INI.M3     GLOBAL VARIABLE      REAL      3 REFS

```

INT.M4	GLOBAL VARIABLE	REAL	3 REFS
IN.CTHREE	GLOBAL VARIABLE	REAL	1 REFS
J	GLOBAL VARIABLE	INTEGER	2 REFS
LAMDA	GLOBAL VARIABLE	REAL	2 REFS
LAST.SD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
LDX	GLOBAL VARIABLE	REAL	1 REFS
LEAD.TIME	GLOBAL VARIABLE	INTEGER	2 REFS
LGHT.TO.FAILURE	GLOBAL VARIABLE	REAL	2 REFS
LOCATED.IN.STRESS.CO	GLOBAL VARIABLE	REAL	1 REFS
LONG.LIST	GLOBAL VARIABLE	REAL	1 REFS
LTHO	GLOBAL VARIABLE	ALPHA	6 REFS
L.EXT	GLOBAL VARIABLE	ALPHA	2 REFS
MFRI	GLOBAL VARIABLE	REAL	1 REFS
MFRI.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MFRI2	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MFRI2.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MODEL	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MOD.SAVE	GLOBAL VARIABLE	(1-D) ALPHA	5 REFS
MOD.TESTED	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
MRDD	GLOBAL VARIABLE	ALPHA	1 REFS
MSRI	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSRI.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSR2	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSR2.INT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MU.R	GLOBAL VARIABLE	(1-D) REAL	1 REFS
M1.MEAN	GLOBAL VARIABLE	(1-D) REAL	1 REFS
M2.MEAN	GLOBAL VARIABLE	REAL	2 REFS
M3.MEAN	GLOBAL VARIABLE	REAL	4 REFS
M4.MEAN	GLOBAL VARIABLE	REAL	4 REFS
NCZ	GLOBAL VARIABLE	REAL	4 REFS
NCZO	GLOBAL VARIABLE	REAL	4 REFS
NFTS	GLOBAL VARIABLE	INTEGER	2 REFS
NOAC	GLOBAL VARIABLE	INTEGER	2 REFS
NOE	GLOBAL VARIABLE	REAL	4 REFS
NO.FAIL	GLOBAL VARIABLE	INTEGER	6 REFS
OCCUP.MOD	GLOBAL VARIABLE	INTEGER	3 REFS
OICO	GLOBAL VARIABLE	(1-D) REAL	1 REFS
OICR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
PF	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
PFID	LOCAL RECURSIVE	REAL	2 REFS
PFTIM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
POP.SIZE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
PRONG	GLOBAL VARIABLE	REAL	2 REFS
PREDICTED.LIFE	GLOBAL VARIABLE	REAL	3 REFS
PROB	GLOBAL VARIABLE	REAL	7 REFS
PRODUCTION.TIME	LOCAL RECURSIVE	REAL	4 REFS
PSAVE	GLOBAL VARIABLE	REAL	2 REFS
RANDOM.F	GLOBAL VARIABLE	REAL	7 REFS
READ	PROCEDURE	REAL	1 REFS
REAL.LIFE	UNSUBSCRIPTED LABEL	REAL	2 REFS
RLGT	PROCEDURE	INTEGER	1 REFS
SACID	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SAMP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS

SAMPLING	GLOBAL VARIABLE	INTEGER	2 REFS	
SAMP.SIZE	GLOBAL VARIABLE	REAL	3 REFS	
SAPID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
SATL	LOCAL RECURSIVE	REAL	3 REFS	
SC	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
SD	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
SDIF	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
SDM.OCCURRENCE.RATE	GLOBAL VARIABLE	REAL	1 REFS	
SD.SCH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS	
SECOND.LIFE	LOCAL RECURSIVE	REAL	1 REFS	
SEED.V	SYSTEM ATTRIBUTE	(1-D) INTEGER	11 REFS	
SELNB	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS	
SEL1	GLOBAL VARIABLE	ALPHA	3 REFS	
SEL2	GLOBAL VARIABLE	ALPHA	3 REFS	
SEL3	GLOBAL VARIABLE	ALPHA	3 REFS	
SF	GLOBAL VARIABLE	REAL	1 REFS	
SFLTHR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
SIG.R	GLOBAL VARIABLE	REAL	2 REFS	
SIZE.OF.FLEET	GLOBAL VARIABLE	INTEGER	23 REFS	
SMOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS	
SRES	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
SRATE	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
SSIAN	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS	
SSTIM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
START.TEST	GLOBAL VARIABLE	PEAL	2 REFS	
STIM	GLOBAL VARIABLE	PEAL	1 REFS	
SU	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
SUMMARY	PROCEDURE	REAL	1 REFS	
SUM.INITIALIZE	PROCEDURE	INTEGER	1 REFS	
S1	LOCAL RECURSIVE	INTEGER	3 REFS	
S10	LOCAL RECURSIVE	INTEGER	3 REFS	
S2	LOCAL RECURSIVE	INTEGER	3 REFS	
S3	LOCAL RECURSIVE	INTEGER	3 REFS	
S4	LOCAL RECURSIVE	INTEGER	3 REFS	
S5	LOCAL RECURSIVE	INTEGER	3 REFS	
S6	LOCAL RECURSIVE	INTEGER	3 REFS	
S7	LOCAL RECURSIVE	INTEGER	3 REFS	
S8	LOCAL RECURSIVE	INTEGER	3 REFS	
S9	LOCAL RECURSIVE	INTEGER	3 REFS	
S.FREQ.CHG	GLOBAL VARIABLE	REAL	3 REFS	
S.OPT	LOCAL RECURSIVE	ALPHA	3 REFS	
S.REPAIR.COST	GLOBAL VARIABLE	REAL	3 REFS	
T	LOCAL RECURSIVE	REAL	2 REFS	
TAIL.ID	TEMPORARY ATTRIBUTE	INTEGER	4 REFS	
TEST.ACCEL.FACT	GLOBAL VARIABLE	REAL	2 REFS	
TEST.LIFE	GLOBAL VARIABLE	REAL	2 REFS	
TES.FAILURE	GLOBAL VARIABLE	ALPHA	1 REFS	
TLID	GLOBAL VARIABLE	(2-D) INTEGER	3 REFS	
TMOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS	
TRCNE	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
TRUNC.F	PROCEDURE	INTEGER	1 REFS	
T.FREQ.CHG	GLOBAL VARIABLE	REAL	3 REFS	
T.IMPLEMENT.MOD	EVENT NOTICE		1 REFS	

STRAY NAME

-----CDC 6600 CACI SIMSCRIPT II-3 VERSION /4-0-00/ KRONOS 2.1.2 03/02/78. 18.36.02. PAGE 24

T.LAST.D	GLOBAL VARIABLE	(1-D) REAL	1 REFS
USAGE.LIFE	GLOBAL VARIABLE	REAL	7 REFS
XRN	GLOBAL VARIABLE	(1-D) REAL	1 REFS
1*AFI	GLOBAL VARIABLE	REAL	3 REFS
1*BCD	GLOBAL VARIABLE	(1-D) REAL	10 REFS
1*MD.COST	GLOBAL VARIABLE	REAL	1 REFS
1*TOOLING	GLOBAL VARIABLE	REAL	1 REFS
1*CDM.OCCURRENCE.RAT	GLOBAL VARIABLE	REAL	1 REFS
1*CR.EXISTS	GLOBAL VARIABLE	REAL	1 REFS
1*INT	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
1*PROB	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2*CDM.OCCURRENCE.RAT	GLOBAL VARIABLE	REAL	1 REFS
2*CR.EXISTS	GLOBAL VARIABLE	REAL	1 REFS
2*INT	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2*PRODUCTION.TIME	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
	GLOBAL VARIABLE	REAL	2 REFS

```

1 ROUTINE INITIALIZATION
2 **
3 ** CHANGE INSPECTION LEVEL CODE TO NUMERIC VALUE
4 **
5 IF INT.LVL.INSPE = "A" LET ITRNL = 1 JUMP AHEAD
6 OTHERWISE
7 IF INT.LVL.INSPE = "B" LET ITRNL = 2 JUMP AHEAD
8 OTHERWISE
9 IF INT.LVL.INSPE = "C" LET ITRNL = 3 JUMP AHEAD
10 OTHERWISE
11 LET ITRNL = 4
12
13 **
14 IF EXT.LVL.INSPE = "A" LET EXT.INSPE.LEVEL = 1 JUMP AHEAD
15 OTHERWISE
16 IF EXT.LVL.INSPE = "B" LET EXT.INSPE.LEVEL = 2 JUMP AHEAD
17 OTHERWISE
18 IF EXT.LVL.INSPE = "C" LET EXT.INSPE.LEVEL = 3 JUMP AHEAD
19 OTHERWISE
20 LET EXT.INSPE.LEVEL = 4
21
22 **
23 ** CHANGE CORROSION RESISTANCE RATING TO MULTIPLYING FACTOR
24 **
25 IF CRR = 1 LET CRRF = 4.0 JUMP AHEAD
26 OTHERWISE
27 IF CRR = 2 LET CRRF = 3.0 JUMP AHEAD
28 OTHERWISE
29 IF CRR = 3 LET CRRF = 2.0 JUMP AHEAD
30 OTHERWISE
31 LET CRRF = 1.5
32
33 **
34 FOR I = 1 TO 4
35 DO
36 LET ABCD(I) = 1/ABCD(I)
37 LET OICR(I) = 0
38 LET OICR(I) = 0
39 LOOP
40
41 LET HI.TIME.ACRFT(I) = 1
42 LET MADD(I) = 0.0
43 LOOP
44
45 **
46 LET LMTA = 0
47 LET MOD.NO = 0
48 LET CR.CTR = 0
49 LET FECK = 0
50 LET NICHG = 1
51 LET OICR = 0
52 LET OCOR = 0
53 LET OSDM = 0
54 LET OSCR = 0

```

```

54 LET OSCG = 0
55 LET OPD = 0
56 LET NSFL = 0
57 LET NRFS = 0
58 LET NMO = 0
59 LET NSMD = 0
60 LET NSIC = 0
61 LET NN = 0
62 LET IMOD = 0
63 LET 1.NUM.OF.REPAIRS = 0
64 LET 2.NUM.OF.CRASH = 0
65 LET TO.BE.MODIFIED = 0
66 LET BEEN.MODIFIED = 0
67 LET IDCK = 0
68 LET TIME.V = 0.0
69 LET AIRFRAME.TIME = 0.0
70 LET COST.OF.REPAIRS = 0.0
71 LET CHG.FREQ.TIME = 0.0
72 LET CINSL = IABCD(3)
73 LET DINSL = IABCD(4)
74 LET LIL = ITRNL
75 LET LEL = EAT.INSPL.LEVEL
76 IF LEL = 1 OR LEL = 2
77 IF LIL = 1 OR LIL = 2
78 LET LIL = 3
79 ALWAYS
80 LET LEL = 3
81 REGARDLESS
82 LET IFAIL = "NO"
83 LET LTHO = "NO"
84 LET RETRO = "NO"
85 LET YES.FAILURE = "NO"
86 LET MOD.TESTED = "NO"
87 LET DEC.ON.MOD.SCH = "NO"
88 LET IMP.MOD.SCH = "NO"
89 LET PREVIOUSLY.MODIFIED = "NO"
90 LET DEC.INT = "NO"
91 LET IFLAG = "NO"
92 LET JFLAG = "NO"
93
94 FOR I = 1 TO SIZE.OF.FLEET
95 DO
96 LET IEL(I) = "NO"
97 LET IEZ(I) = "NO"
98 LET GRI(I) = 1.0
99 LET CCUR.MOD(I) = 0.0
100 LET AIL(I) = "NO"
101 LET FSH(I) = "NO"
102 LET INSP.SCH(I) = "NO"
103 LET TMOO.PENDING(I) = "NO"
104 LET C.INTERVAL(I) = IABCD(3)
105 LET O.INTERVAL(I) = IABCD(4)
106 LET SMOD.PENDING(I) = "NO"

```

```

107 LET I.CR-EXISTS(I) = "NN"
108 LET 2.CR-EXISTS(I) = "NN"
109 LET CO-EXISTS(I) = "NN"
110 LET SO-SCH(I) = "NO"
111 LET O.IN(I) = 0
112 LET INT-LGT(I) = 0.0
113 LOOP
114
115 ## RESET TALLY COUNTERS
116 ##
117
118 RESET TOTALS OF LCWK
119 RESET TOTALS OF LCORT
120 RESET TOTALS OF LSDT
121 RESET TOTALS OF ACRKL
122 RESET TOTALS OF BCRKL
123 RESET TOTALS OF CCRKL
124 RESET TOTALS OF DCRKL
125 RESET TOTALS OF SCRKL
126 RESET TOTALS OF ACA
127 RESET TOTALS OF BCA
128 RESET TOTALS OF CCA
129 RESET TOTALS OF DCA
130 RESET TOTALS OF SCA
131 RETURN
    END

```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE INITIALIZATION

ABCD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
ACA	GLOBAL VARIABLE		REAL	1 REFS
ACRKL	GLOBAL VARIABLE		REAL	1 REFS
AIL	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
AIRFRAME.TIME	GLOBAL VARIABLE		REAL	1 REFS
BCA	GLOBAL VARIABLE		REAL	1 REFS
BCRKL	GLOBAL VARIABLE		REAL	1 REFS
BEEN.MODIFIED	GLOBAL VARIABLE		REAL	1 REFS
CCA	GLOBAL VARIABLE		INTEGER	1 REFS
CCRKL	GLOBAL VARIABLE		REAL	1 REFS
CGI	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
CHG.FREQ.TIME	GLOBAL VARIABLE		REAL	1 REFS
CINSL	GLOBAL VARIABLE		REAL	1 REFS
COST.OF.REPAIRS	GLOBAL VARIABLE		REAL	1 REFS
CO.EXISTS	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
CRF	GLOBAL VARIABLE		INTEGER	3 REFS
CRF	GLOBAL VARIABLE		REAL	4 REFS
CR.CTR	GLOBAL VARIABLE		INTEGER	1 REFS
C.INTERVAL	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
DCA	GLOBAL VARIABLE		REAL	1 REFS
DEC.INT	GLOBAL VARIABLE		REAL	1 REFS
DEC.ON.MOD.SCH	GLOBAL VARIABLE		ALPHA	1 REFS
DINSL	GLOBAL VARIABLE		ALPHA	1 REFS
D.IN	GLOBAL VARIABLE		REAL	1 REFS
D.INTERVAL	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
EXT.INSPEL	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
EXT.LVL.INSPEL	GLOBAL VARIABLE		INTEGER	5 REFS
FOCK	GLOBAL VARIABLE		ALPHA	3 REFS
FSH	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
HI.TIME.ACRFT	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
I	LOCAL RECURSIVE		INTEGER	27 REFS
IDCK	GLOBAL VARIABLE		REAL	1 REFS
IE1	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
IE2	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
IFAIL	GLOBAL VARIABLE		ALPHA	1 REFS
IFLAG	GLOBAL VARIABLE		ALPHA	1 REFS
IMOD	GLOBAL VARIABLE		INTEGER	1 REFS
IMP.MOD.SCH	GLOBAL VARIABLE		INTEGER	1 REFS
INITIALIZATION	PROCEDURE		INTEGER	1 REFS
INSP.SCH	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
INT.LGT	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
INT.LVL.INSPEL	GLOBAL VARIABLE		ALPHA	1 REFS
ITRNL	GLOBAL VARIABLE		INTEGER	1 REFS
JFLAG	GLOBAL VARIABLE		INTEGER	1 REFS
LFL	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
LMTA	GLOBAL VARIABLE		ALPHA	3 REFS
LTL	GLOBAL VARIABLE		INTEGER	4 REFS
LTHO	GLOBAL VARIABLE		INTEGER	1 REFS
MOD.NO	GLOBAL VARIABLE		ALPHA	1 REFS
MOD.TESTED	GLOBAL VARIABLE		INTEGER	1 REFS

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4-0-00/ KRONOS 2.1.2

MRDD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
NICMG	GLOBAL VARIABLE	INTEGER	1 REFS
NMD	GLOBAL VARIABLE	INTEGER	1 REFS
NN	GLOBAL VARIABLE	INTEGER	1 REFS
NRFS	GLOBAL VARIABLE	INTEGER	1 REFS
NSFL	GLOBAL VARIABLE	INTEGER	1 REFS
NSIC	GLOBAL VARIABLE	INTEGER	1 REFS
NSHD	GLOBAL VARIABLE	INTEGER	1 REFS
UCCUR.MOD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
OCOP	GLOBAL VARIABLE	INTEGER	1 REFS
OICO	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
OICR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
OPD	GLOBAL VARIABLE	INTEGER	1 REFS
OSCO	GLOBAL VARIABLE	INTEGER	1 REFS
OSCR	GLOBAL VARIABLE	INTEGER	1 REFS
OSDM	GLOBAL VARIABLE	INTEGER	1 REFS
OICR	GLOBAL VARIABLE	INTEGER	1 REFS
PREVIOUSLY.MODIFIED	GLOBAL VARIABLE	ALPHA	1 REFS
RETRO	GLOBAL VARIABLE	ALPHA	1 REFS
SCA	GLOBAL VARIABLE	REAL	1 REFS
SCPKL	GLOBAL VARIABLE	REAL	1 REFS
SD.SCM	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SHOD.PENDING	GLOBAL VARIABLE	ALPHA	1 REFS
YES.FAILURE	GLOBAL VARIABLE	ALPHA	1 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	1 REFS
TMOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
TO.RE.MODIFIED	GLOBAL VARIABLE	INTEGER	1 REFS
IABCD	GLOBAL VARIABLE	(1-D) REAL	5 REFS
ICORT	GLOBAL VARIABLE	REAL	1 REFS
ICRFT	GLOBAL VARIABLE	REAL	1 REFS
ISOT	GLOBAL VARIABLE	REAL	1 REFS
1.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
1.NUM.OF.RETIRE	GLOBAL VARIABLE	INTEGER	1 REFS
2.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2.NUM.OF.CRASH	GLOBAL VARIABLE	INTEGER	1 REFS

```
1 ROUTINE SUM-INITIALIZE
2 FOR I = 1 TO 4
3 DO
4 LET GOICR(I) = 0
5 LET GOICR(I) = 0
6 LOOP
7 LET SAMP.SIZE = 0.0
8 LET SNSIC = 0
9 LET SNMO = 0
10 LET SNSMD = 0
11 LET SNSFL = 0
12 LET SNRFS = 0
13 LET GOICR = 0
14 LET GQPD = 0
15 LET GOCOR = 0
16 LET GOSOM = 0
17 LET GOSCM = 0
18 LET GOSCO = 0
19 LET NCZO = 1
20 LET NCZ = 0
21 ##
22 RESET TOTALS OF GICRK
23 RESET TOTALS OF GICOR
24 RESET TOTALS OF GISO
25 RESET TOTALS OF SACRK
26 RESET TOTALS OF 68CRK
27 RESET TOTALS OF 6CCRK
28 RESET TOTALS OF 6DCRK
29 RESET TOTALS OF 6SCRK
30 RESET TOTALS OF 6ACA
31 RESET TOTALS OF 6CCA
32 RESET TOTALS OF 6DCA
33 RESET TOTALS OF 6GCA
34 RESET TOTALS OF 6CNSL
35 RESET TOTALS OF 6CNSL
36 RESET TOTALS OF 6CNSL
37 RESET TOTALS OF 6CNSL
38 ## RETURN
39 END
```



```

1 ROUTINE COMP.RISK YIELDING CL
2 LET A = AVEAN*AFACI(ID)
3 LET T = TIME-V - TRCHG(ID)
4 LET SR1 = SRPATE(ID)
5 LET SR = SR1
6 LET RS = SRES(ID) - T*SR1
7 LET K14 = (A*EXP.F(8*SRES(ID)))/(8*SR1)
8 LET K15 = EXP.F(-8*SR1*T)-1.0
9 LET CGR = 1.0
10 LET CCL1 = CONE
11 LET CCL2 = CTWO
12 LET CCL3 = CTHREE
13 LET L-STRENGTH-REDUCTION = ALSR(ID)
14 LET TAL = TIME-ALL-STRENGTH-REDUCTION
15 IF CO-EXISTS(ID) = "YES"
16 LET COROSION = AC(ID)
17 LET TAC = TIME-A-COROSION
18 ALWAYS
19 IF TAC LE TAL OR TAC GE TRCHG(ID)
20 IF TAC GT 0.0 AND TAC LE TAL
21 LET CGR = CGR1(ID)
22 ALWAYS
23 LET DL1 = (TRCHG(ID)-TAL)*MSR1(ID)*CGR
24 IF DL1 GT CCL1
25 LET DL1 = CCL1 + (DL1-CCL1)*(MFR1(ID)/MSR1(ID))
26 ALWAYS
27 IF DL1 GT CCL2
28 LET DL1 = CCL2 + (DL1-CCL2)*(MSR2(ID)/MFR1(ID))
29 ALWAYS
30 IF DL1 GT CCL3
31 LET DL1 = CCL3 + (DL1-CCL3)*(MFR2(ID)/MSR2(ID))
32 ALWAYS
33 JUMP AHEAD
34 ELSE
35 LET DLC1 = (TAC-TAL)*MSR1(ID)
36 IF DLC1 GT CCL1
37 LET DLC1 = CCL1 + (DLC1-CCL1)*(MFR1(ID)/MSR1(ID))
38 ALWAYS
39 IF DLC1 GT CCL2
40 LET DLC1 = CCL2 + (DLC1-CCL2)*(MSR2(ID)/MFR1(ID))
41 ALWAYS
42 IF DLC1 GT CCL3
43 LET DLC1 = CCL3 + (DLC1-CCL3)*(MFR2(ID)/MSR2(ID))
44 ALWAYS
45 IF DLC1 GT CCL3
46 LET DL1 = DLC1 + (TRCHG(ID)-TAC)*MFR2(ID)*CGR1(ID)
47 JUMP AHEAD
48 ELSE
49 IF DLC1 GT CCL2
50 LET DL1 = DLC1 + (TRCHG(ID)-TAC)*MSR2(ID)*CGR1(ID)
51 IF DL1 GT CCL3
52 LET DL1 = CCL3 + (DL1-CCL3)*(MFR2(ID)/MSR2(ID))
53 ALWAYS

```

03/02/78. 18.36.02-

CDC 6600 CACI SINSCTRI-II.5 VERSION /4.0-00/ KRONOS 2.1.2

```

54      JUMP AHEAD
55      ELSE
56      IF DL1 GT CCL1
57      LET DL1 = DL1 + (TRCHG(ID)-TAC)*MFR1(ID)*CGRI(ID)
58      IF DL1 GT CCL2
59      LET DL1 = CCL2 + (DL1-CCL2)*(MSR2(ID)/MFR1(ID))
60      ALWAYS
61      IF DL1 GT CCL3
62      LET DL1 = CCL3 + (DL1-CCL3)*(MFR2(ID)/MSR2(ID))
63      ALWAYS
64      JUMP AHEAD
65      ELSE
66      LET DL1 = DL1 + (TRCHG(ID)-TAC)*MSR1(ID)*CGRI(ID)
67      IF DL1 GT CCL1
68      LET DL1 = CCL1 + (DL1-CCL1)*(MFR1(ID)/MSR1(ID))
69      ALWAYS
70      IF DL1 GT CCL2
71      LET DL1 = CCL2 + (DL1-CCL2)*(MSR2(ID)/MFR1(ID))
72      ALWAYS
73      IF DL1 GT CCL3
74      LET DL1 = CCL3 + (DL1-CCL3)*(MFR2(ID)/MSR2(ID))
75      ALWAYS
76      HERE
77      IF DL1 GT CCL3
78      LET CL1 = DL1 + T*MFR2(ID)*CGRI(ID)
79      JUMP AHEAD
80      ELSE
81      IF DL1 GT CCL2
82      LET CL1 = DL1 + T*MSR2(ID)*CGRI(ID)
83      IF CL1 GT CCL3
84      LET CL1 = CCL3 + (CL1-CCL3)*(MFR2(ID)/MSR2(ID))
85      ALWAYS
86      JUMP AHEAD
87      ELSE
88      IF DL1 GT CCL1
89      LET CL1 = DL1 + T*MFR1(ID)*CGRI(ID)
90      IF CL1 GT CCL2
91      LET CL1 = CCL2 + (CL1-CCL2)*(MSP2(ID)/MFR1(ID))
92      ALWAYS
93      IF CL1 GT CCL3
94      LET CL1 = CCL3 + (CL1-CCL3)*(MFR2(ID)/MSR2(ID))
95      ALWAYS
96      JUMP AHEAD
97      ELSE
98      LET CL1 = DL1 + T*MSR1(ID)*CGRI(ID)
99      IF CL1 GT CCL1
100     LET CL1 = CCL1 + (CL1-CCL1)*(MFR1(ID)/MSR1(ID))
101     ALWAYS
102     IF CL1 GT CCL2
103     LET CL1 = CCL2 + (CL1-CCL2)*(MSR2(ID)/MFR1(ID))
104     ALWAYS
105     IF CL1 GT CCL3
106     LET CL1 = CCL3 + (CL1-CCL3)*(MFR2(ID)/MSR2(ID))

```

```

107 ALWAYS
108 HERE
109 IF 2.CR.EXISTS(ID) = "YES"
110 LET 2.STRENGTH-REDUCTION = A2SR(ID)
111 LET TAZ = TIME-A(2-STRENGTH-REDUCTION)
112 IF TAZ LT TRNGG(ID)
113 LET DL2 = (TRNGG(ID)-TAZ)*MSR1(ID)
114 IF DL2 GT CCL1
115 LET DL2 = CCL1 + (DL2-CCL1)*(MFR1(ID)/MSR1(ID))
116 ALWAYS
117 IF DL2 GT CCL2
118 LET DL2 = CCL2 + (DL2-CCL2)*(MSR2(ID)/MFR1(ID))
119 ALWAYS
120 IF DL2 GT CCL3
121 LET DL2 = CCL3 + (DL2-CCL3)*(MFR2(ID)/MSR2(ID))
122 ALWAYS
123 ALWAYS
124 IF DL2 GT CCL3
125 LET CL2 = DL2 + T*MFR2(ID)*CGRI(ID)
126 JUMP AHEAD
127 ELSE
128 IF DL2 GT CCL2
129 LET CL2 = DL2 + T*MSR2(ID)*CGRI(ID)
130 IF CL2 GT CCL3
131 LET CL2 = CCL3 + (CL2-CCL3)*(MFR2(ID)/MSR2(ID))
132 ALWAYS
133 JUMP AHEAD
134 ELSE
135 IF DL2 GT CCL1
136 LET CL2 = DL2 + T*MFR1(ID)*CGRI(ID)
137 IF CL2 GT CCL2
138 LET CL2 = CCL2 + (CL2-CCL2)*(MSR2(ID)/MFR1(ID))
139 ALWAYS
140 IF CL2 GT CCL3
141 LET CL2 = CCL3 + (CL2-CCL3)*(MFR2(ID)/MSR2(ID))
142 ALWAYS
143 JUMP AHEAD
144 ELSE
145 LET CL2 = DL2 + T*MSR1(ID)*CGRI(ID)
146 IF CL2 GT CCL1
147 LET CL2 = CCL1 + (CL2-CCL1)*(MFR1(ID)/MSR1(ID))
148 ALWAYS
149 IF CL2 GT CCL2
150 LET CL2 = CCL2 + (CL2-CCL2)*(MSR2(ID)/MFR1(ID))
151 ALWAYS
152 IF CL2 GT CCL3
153 LET CL2 = CCL3 + (CL2-CCL3)*(MFR2(ID)/MSR2(ID))
154 ALWAYS
155 HERE
156 ALWAYS
157 LET CL = CCL1 + CL2
158 IF 2.CR.EXISTS(ID) NE "YES"
159 IF DL1 GT CCL3

```

```

160 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15)
161 JUMP AHEAD
162 ELSE
163 IF DL1 GT CCL2
164 IF CL1 GT CCL3
165 LET T1 = (CCL3-DL1)/(MSR2(ID)*CGRI(ID))
166 LET SR2 = (MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
167 LET K15 = EXP.F(-B*SR1*T1)-1.0
168 LET S1 = SRES(ID) - SR1*T1
169 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
170 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
171 LET ARG1 = K14*K15*K16*K17
172 IF ARG1 LT -675.
173 LET MZRD(ID) = MZRD(ID)*.0001
174 PRINT 2 LINES WITH ARG1, TIME, V, TRCHG(ID), SR1, DL1, DL2, CL1, CL2, SRES(ID),
175 RLGT(ID), SDIF(ID) AS FOLLOWS
176 ARG1=*****. TIME=***** TRCHG=***** SR1=*****
177 DL1=***** CL1=***** CL2=***** SRES=***** R=***** SD=*****
178 GO TO TWO
179 ALWAYS
180 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15)+(K16*K17)
181 N1=0
182 LET SR = SR2
183 LET RS = S1 - (T-T1)*SR2
184 JUMP AHEAD
185 ELSE
186 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15)
187 JUMP AHEAD
188 ELSE
189 IF DL1 GT CCL1
190 LET AZRD = EXP.F(K14*K15)
191 IF CL1 GT CCL2
192 LET T1 = (CCL2-DL1)/(MFR1(ID)*CGRI(ID))
193 LET SR2 = (MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
194 LET S1 = SRES(ID) - T1*SR1
195 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
196 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
197 LET AZRD = EXP.F(K14*K15)*(K16*K17)
198 LET SR = SR2
199 LET RS = S1 - (T-T1)*SR2
200 IF CL1 GT CCL3
201 LET T2 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
202 LET SR3 = (MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
203 LET S2 = S1 - T2*SR2
204 LET K17 = EXP.F(-B*SR2*T2)-1.0
205 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
206 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
207 LET AZRD = EXP.F(K14*K15)*(K16*K17)*(K18*K19)
208 LET SR = SR3
209 LET RS = S2 - (T-T1-T2)*SR3
210 ALWAYS
211 ALWAYS
212 LET MZRD(ID) = MZRD(ID)*AZRD
213 JUMP AHEAD

```

```

211 ELSE
212   LET AZRD = EXP.F(K14*K15)
213   IF CL1 GT CCL1
214     LET T1 = (CCL1-DL1)/(MSR1(ID)*CGRI(ID))
215     LET S1 = SRES(ID) - T1*SR1
216     LET SR2 = (MFR1(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
217     LET K15 = EXP.F(-8*SR1**2)-1.0
218     LET K16 = (A*EXP.F(B*S1))/(B*SR2)
219     LET K17 = EXP.F(-8*S2*(T-T1))-1.0
220     LET ARG = K14 * K15 * K16 * K17
221     IF ARG LT -675.
222       PRINT 1 LINE WITH ARG AS FOLLOWS
223       ARG=*****
224       LET AZRD=.0001
225       GO TO ONE
226       ALWAYS
227       #ONE# LET SR.= SR2
228       LET RS = S1 - (T-T1)*S*2
229       IF CL1 GT CCL2
230         LET T2 = (CCL2-CCL1)/(MFR1(ID)*CGRI(ID))
231         LET S2 = S1 - T2*SR2
232         LET SR3 = (MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
233         LET K17 = EXP.F(-8*SR2**2)-1.0
234         LET K18 = (A*EXP.F(B*S2))/(B*SR3)
235         LET K19 = EXP.F(-8*SR3*(T-T1-T2))-1.0
236         LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19))
237         LET S* = SR3
238         LET RS = S2 - (T-T1-T2)*SR3
239         IF CL1 GT CCL3
240           LET T3 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
241           LET S3 = S2 - T3*SR3
242           LET SR4 = (MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
243           LET K19 = EXP.F(-8*SR3**2)-1.0
244           LET K20 = (A*EXP.F(B*S3))/(B*SR4)
245           LET K21 = EXP.F(-8*SR4*(T-T1-T2-T3))-1.0
246           LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21))
247           LET SR = SR4
248           LET RS = S3 - (T-T1-T2-T3)*SR4
249           ALWAYS
250           ALWAYS
251           ALWAYS
252           LET HZRD(ID) = HZRD(ID)*AZRD
253           JUMP AHEAD
254         ELSE
255           IF DL1 GT CCL3
256             IF DL2 GT CCL3 OR DL2 GT CCL2 AND CL2 LT CCL3 OR DL2 GT CCL1 AND CL2 LT CCL2
257               OR CL2 LT CCL1
258               LET HZRD(ID) = HZRD(ID)*EXP.F(K14*K15)
259               JUMP AHEAD
260             ELSE
261               IF DL2 GT CCL2
262                 LET T1 = (CCL3-DL2)/(MSR2(ID)*CGRI(ID))

```

```

263 LET S1 = SRES(ID) - SR1*T1
264 LET SR2 = (2.*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
265 LET K15 = EXP.F(-B*SR1*T1)-1.0
266 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
267 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
268 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17))
269 LET SR = SR2
270 LET RS = S1 - (T-T1)*SR2
271 JUMP AHEAD
272 ELSE
273 IF DL2 GT CCL1
274 LET T1 = (CCL2-DL2)/(MFR1(ID)*CGRI(ID))
275 LET S1 = SRES(ID) - SR1*T1
276 LET SR2 = (MSR2(ID)*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
277 LET K15 = EXP.F(-B*SR1*T1)-1.0
278 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
279 IF CL2 GT CCL3
280 LET T2 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
281 LET S2 = S1 - SR2*T2
282 LET SR3 = (2.*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
283 LET K17 = EXP.F(-B*SR2*T2)-1.0
284 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
285 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
286 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19))
287 LET SR = SR3
288 LET K5 = S2 - (T-T1-T2)*SR3
289 JUMP AHEAD
290 ELSE
291 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
292 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17))
293 LET SR = SR2
294 LET RS = S1 - (T-T1)*SR2
295 JUMP AHEAD
296 ELSE
297 LET T1 = (CCL1-DL2)/(MFR1(ID)*CGRI(ID))
298 LET S1 = SRES(ID) - SR1*T1
299 LET SR2 = (MFR1(ID)*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
300 LET K15 = EXP.F(-B*SR1*T1)-1.0
301 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
302 IF CL2 GT CCL2
303 LET T2 = (CCL2-CCL1)/(MFR1(ID)*CGRI(ID))
304 LET S2 = S1 - SR2*T2
305 LET SR3 = (MFR2(ID)*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
306 LET K17 = EXP.F(-B*SR2*T2)-1.0
307 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
308 IF CL2 GT CCL3
309 LET T3 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
310 LET S3 = S2 - SR3*T3
311 LET SR4 = (2.*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
312 LET K19 = EXP.F(-B*SR3*T3)-1.0
313 LET K20 = (A*EXP.F(B*S3))/(B*SR4)
314 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
315 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21))

```

```

316 LET SR = SR4
317 LET RS = S3 - (T-T1-T2-T3)*SR4
318 JUMP AHEAD
319 ELSE
320 LET K19 = EXP.F(-8*SR3*(T-T1-T2))-1.0
321 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15+K16*K17+K18*K19)
322 LET SR = SR3
323 LET RS = S2 - (T-T1-T2)*SR3
324 JUMP AHEAD
325 ELSE
326 LET K17 = EXP.F(-8*SR2*(T-T1))-1.0
327 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15+(K16*K17))
328 LET SR = SR2
329 LET RS = S1 - (T-T1)*SR2
330 JUMP AHEAD
331 ELSE
332 IF DL1 GT CCL2
333 IF CCL1 GT CCL3
334 LET T11 = (CCL3-DL1)/(MSR2(ID)*CGRI(ID))
335 IF DL2 GT CCL2
336 LET T1 = T11
337 LET S1 = SRES(ID) - SW1*T1
338 LET SR2 = ((MFR2(ID)+MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
339 LET K15 = EXP.F(-8*SW1*T1)-1.0
340 LET K16 = (A*EXP.F(8*S1))/(8*SR2)
341 IF CL2 GT CCL3
342 LET T2 = (CCL3-DL2)/(MSR2(ID)*CGRI(ID))
343 LET S2 = S1 - SR2*T2
344 LET SR3 = (2*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
345 LET K17 = EXP.F(-8*SR2*T2)-1.0
346 LET K18 = (A*EXP.F(8*S2))/(8*SR3)
347 LET K19 = EXP.F(-8*SR3*(T-T1-T2))-1.0
348 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15+(K16*K17)+(K18*K19))
349 LET SR = SR3
350 LET RS = S2 - (T-T1-T2)*SR3
351 JUMP AHEAD
352 ELSE
353 LET K17 = EXP.F(-8*SR2*(T-T1))-1.0
354 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15+(K16*K17))
355 LET SR = SR2
356 LET RS = S1 - (T-T1)*SR2
357 JUMP AHEAD
358 ELSE
359 IF DL2 GT CCL1
360 IF CL2 GT CCL2
361 LET T21 = (CCL2-DL2)/(MFR1(ID)*CGRI(ID))
362 LET T1 = T11
363 LET SR2 = ((MFR1(ID)+MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
364 LET T2 = T21-T1
365 LET SR3 = ((MSR2(ID)+MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
366 IF T2 LT T11
367 LET T1 = T21
368 LET SR2 = (2*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)

```

```

369 LET T2 = T11-T21
370 ALWAYS
371 LET S1 = SRES(ID) - SR1*T1
372 LET S2 = S1 - SR2*T2
373 LET K15 = EXP.F(-B*SR1*T1)-1.0
374 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
375 LET K17 = EXP.F(-B*SR2*T2)-1.0
376 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
377 IF CL2 GT CCL3
378 LET T3 = (CCL3-CCL2)/(MSR2(ID)*CGR1(ID)) + T21 - (T1*T2)
379 LET S3 = S2 - SR3*T3
380 LET SR4 = T2.*HFR2(ID)*CGR1(ID)*SDIF(ID)/RLGT(ID)
381 LET K19 = EXP.F(-B*SR3*T3)-1.0
382 LET K20 = (A*EXP.F(B*S3))/(B*SR4)
383 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
384 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21))
385 LET SR = SR4
386 LET RS = S3 - (T-T1-T2-T3)*SR4
387 JUMP AHEAD
388 ELSE
389 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
390 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19))
391 LET SR = SR3
392 LET RS = S2 - (T-T1-T2)*SR3
393 JUMP AHEAD
394 ELSE
395 LET T1 = T11
396 LET S1 = SRES(ID) - SR1*T1
397 LET SR2 = ((HFR1(ID)*HFR2(ID))*CGR1(ID)*SDIF(ID))/RLGT(ID)
398 LET K15 = EXP.F(-B*SR1*T1)-1.0
399 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
400 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
401 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)*(K16*K17))
402 LET SR = SR2
403 LET RS = S1 - (T-T1)*SR2
404 JUMP AHEAD
405 ELSE
406 LET T1 = T11
407 LET S1 = SRES(ID) - SR1*T1
408 LET SR2 = ((MSR1(ID)*HFR2(ID))*CGR1(ID)*SDIF(ID))/RLGT(ID)
409 LET K15 = EXP.F(-B*SR1*T1)-1.0
410 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
411 IF CL2 GT CCL1
412 LET T21 = (CCL1-CL2)/(MSR1(ID)*CGR1(ID))
413 LET T2 = T21-T11
414 LET SR3 = ((HFR1(ID)*HFR2(ID))*CGR1(ID)*SDIF(ID))/RLGT(ID)
415 IF T21 LT T11
416 LET T1 = T21
417 LET T2 = T11-T21
418 LET SR2 = ((HFR1(ID)*MSR2(ID))*CGR1(ID)*SDIF(ID))/RLGT(ID)
419 LET S1 = SRES(ID) - SR1*T1
420 ALWAYS
421 LET S2 = S1 - SR2*T2

```

```

422 LET K15 = EXP.F(-8*SR1*T1)-1.0
423 LET K16 = (A*EXP.F(8*SR1))/(8*SR1)
424 IF CL2 GT CCL2
425 LET T22 = (CCL2-CCL1)/(MFR1(ID)*CGRI(ID))
426 LET T3 = T22
427 LET SR4 = ((MFR2(ID)+MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
428 IF T21+T22 LT T11
429 LET T2 = T22
430 LET T3 = T11 - T22 - T21
431 LET SR3 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
432 ALWAYS
433 LET S3 = S2 - SR3*T3
434 LET K17 = EXP.F(-8*SR2*T2)-1.0
435 LET K18 = (A*EXP.F(8*SR2))/(8*SR2)
436 LET K19 = EXP.F(-8*SR3*T3)-1.0
437 LET K20 = (A*EXP.F(8*SR3))/(8*SR4)
438 IF CL2 GT CCL3
439 LET T23 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
440 LET T4 = (T21+T22+T23)-T11
441 LET S4 = S3 - SR4*T4
442 LET SK5 = (2.*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
443 LET K21 = EXP.F(-8*SR4*T4)-1.0
444 LET K22 = (A*EXP.F(8*SR4))/(8*SR5)
445 LET K23 = EXP.F(-8*SR5*(T-T1-T2-T3-T4))-1.0
446 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17)+(K18*K19)+(K20*K21)+
447 (K22*K23))
448 LET SK = SK5
449 LET MS = S4 - (T-T1-T2-T3)*SR5
450 JUMP AHEAD
451 ELSE
452 LET K21 = EXP.F(-8*SK4*(T-T1-T2-T3))-1.0
453 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17)+(K18*K19)+(K20*K21))
454 LET SR = SR4
455 LET MS = S3 - (T-T1-T2-T3)*SR4
456 JUMP AHEAD
457 ELSE
458 LET K17 = EXP.F(-8*SR2*T2)-1.0
459 LET K18 = (A*EXP.F(8*SR2))/(8*SR3)
460 LET K19 = EXP.F(-8*SR3*(T-T1-T2))-1.0
461 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17)+(K18*K19))
462 LET SR = SR3
463 LET RS = S2 - (T-T1-T2)*SR3
464 JUMP AHEAD
465 ELSE
466 LET K17 = EXP.F(-8*SK2*(T-T1))-1.0
467 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17))
468 LET SR = SR2
469 LET RS = S1 - (T-T1)*SR2
470 JUMP AHEAD
471 ELSE
472 IF DL2 GT CCL2
473 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15)
474 JUMP AHEAD

```

```

475 ELSE
476 IF DL2 GT CCL1
477 IF CL2 GT CCL2
478 LET T1 = (CCL2-DL2)/(MF1(ID)*CGRI(ID))
479 LET S1 = SRES(ID) - T1*SK1
480 LET SR2 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
481 LET K15 = EXP.F(-8*SR1*T1)-1.0
482 LET K16 = (A*EXP.F(B*S1))/B*SR2
483 LET K17 = EXP.F(-8*SR2*(T-T1))
484 LET HZRD(ID)=HZRD(ID)*EXP.F((K14*K15)*K16*K17)
485 LET SR = SR2
486 LET RS = S1 - (T-T1)*SR2
487 JUMP AHEAD
488 ELSE
489 LET HZRD(ID)=HZRD(ID)*EXP.F(K14*K15)
490 JUMP AHEAD
491 ELSE
492 IF CL2 GT CCL1
493 LET T1 = (CCL1-DL2)/(MSR1(ID)*CGRI(ID))
494 LET S1 = SRES(ID) - T1*SR1
495 LET SR2 = (MF1(ID)*SR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
496 LET K15 = EXP.F(-8*SK1*T1)-1.0
497 LET K16 = (A*EXP.F(B*S1))/B*SR2
498 IF CL2 GT CCL2
499 LET T2 = (CCL2-CCL1)/(MF1(ID)*CGRI(ID))
500 LET S2 = S1 - T2*SR2
501 LET SR3 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
502 LET K17 = EXP.F(-8*SR2*T2)-1.0
503 LET K18 = (A*EXP.F(B*S2))/B*SR3
504 LET K19 = EXP.F(-8*SR3*(T-T2-T1))-1.0
505 LET HZRD(ID)=HZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19))
506 LET SR = SR3
507 LET RS = S2 - (T-T1-T2)*SR3
508 JUMP AHEAD
509 ELSE
510 LET K17 = EXP.F(-8*SR2*(T-T1))-1.0
511 LET HZRD(ID)=HZRD(ID)*EXP.F((K14*K15)*(K16*K17))
512 LET SR = SR2
513 LET RS = S1 - (T-T1)*SR2
514 JUMP AHEAD
515 ELSE
516 LET HZRD(ID)=HZRD(ID)*EXP.F(K14*K15)
517 JUMP AHEAD
518 ELSE
519 IF DL1 GT CCL1
520 IF CL1 GT CCL3
521 LET T11 = (CCL2-DL1)/(MF1(ID)*CGRI(ID))
522 LET T12 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
523 IF DL2 GT CCL1
524 IF CL2 GT CCL3
525 LET T21 = (CCL2-DL2)/(MF1(ID)*CGRI(ID))
526 LET T22 = T12
527 LET T1 = T11

```

```

528 LET S1 = SRES(ID) - SRI*TI
529 LET SR2 = ((MFRI(ID)*MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
530 LET T2 = T21-T11
531 LET SR3 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
532 LET T3 = T12 - T2
533 LET SR4 = ((MSR2(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
534 LET T4 = T22 - T3
535 LET SR5 = (2.*MFR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
536 IF T11+T12 LT T21
537 LET T2 = T12
538 LET SR3 = ((MFRI(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
539 LET T3 = T21-T11-T12
540 LET T4 = T22
541 ALWAYS
542 LET S2 = S1 - T2*SR2
543 LET S3 = S2 - T3*SR3
544 LET S4 = S3 - T4*SR4
545 LET K15 = EXP.F((-B*SR1+T1))-1.0
546 LET K16 = (A*EXP.F(B*S1))/((B*SR2)
547 LET K17 = EXP.F((-B*SR2+T2))-1.0
548 LET K18 = (A*EXP.F(B*S2))/((B*SR3)
549 LET K19 = EXP.F((-B*SP3+T3))-1.0
550 LET K20 = (A*EXP.F(B*S3))/((B*SR4)
551 LET K21 = EXP.F((-B*SP4+T4))-1.0
552 LET K22 = (A*EXP.F(B*S4))/((B*SR5)
553 LET K23 = EXP.F((-B*SR5*(T1-T2-T3-T4)))-1.0
554 LET MZMD(ID)=HZMD(ID)*EXP.F((K14*K15)+(K16*K17)+(K18*K19)+(K20*K21)*
555 (K22*K23))
556 LET SR = S45
557 LET R5 = S4 - (T-T1-T2-T3)*SR5
558 JUMP AHEAD
559 ELSE
560 IF CL2 GT CCL2
561 LET T21 = (CCL2-DL2)/(MFR1(ID)*CGRI(ID))
562 LET T22 = (CCL2-CCL2)/(MSR2(ID)*CGRI(ID))
563 LET T1 = T11
564 LET T2 = T21-T11
565 LET SR2 = ((MFRI(ID)*MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
566 LET T3 = T12 - T2
567 LET SR3 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
568 LET SR4 = ((MSR2(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
569 IF T11+T12 LT T21
570 LET T2 = T12
571 LET T3 = T21-T11-T12
572 LET SR3 = ((MFRI(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
573 ALWAYS
574 LET S1 = SRES(ID) - T1*S41
575 LET S2 = S1 - T2*SR2
576 LET S3 = S2 - T3*SR3
577 LET K15 = EXP.F((-B*SR1+T1))-1.0
578 LET K16 = (A*EXP.F(B*S1))/((B*SR2)
579 LET K17 = EXP.F((-B*SR2+T2))-1.0
580 LET K18 = (A*EXP.F(B*S2))/((B*SR3)

```

```

581 LET K19 = EXP.F(-R*SR3*T3)-1.0
582 LET K20 = (A*EXP.F(B*S311/19*SR4)
583 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
584 LET M2PD(ID)=M2RD(ID)*EXP.F((K14*K15)+(K16*K17)+(K18*K19)*(K20*K21))
585 LET SR = SR4
586 LET RS = S3 - (T-T1-T2-T3)*SR4
587 JUMP AHEAD
588 ELSE
589 LET T1 = T11
590 LET T2 = T12
591 LET SR2 = ((MFRI(ID)+MSR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
592 LET SR3 = ((MFRI(ID)+MFR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
593 LET S1 = SRES(ID) - T1*SR1
594 LET S2 = S1 - T2*SR2
595 LET K15 = EXP.F(-R*SK1*T1)-1.0
596 LET K16 = (A*EXP.F(R*S1))/((R*SR2)
597 LET K17 = EXP.F(-B*SR2*T2)-1.0
598 LET K18 = (A*EXP.F(R*S2))/((R*SR3)
599 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
600 LET M2PD(ID)=M2RD(ID)*EXP.F((K14*K15)+(K16*K17)+(K18*K19)
601 LET SR = SR5
602 LET RS = S2 - (T-T1-T2)*SR3
603 JUMP AHEAD
604 ELSE
605 LET T1 = T11
606 LET T2 = T12
607 LET SR2 = ((MFRI(ID)+MSR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
608 LET SR3 = ((MFRI(ID)+MFR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
609 IF CL2 GT CCL1
610 LET T21 = (CCL1-DL2)/(MSR1(ID)+CGRI(ID))
611 LET T3 = T21
612 LET SR4 = ((MFRI(ID)+MFR2(ID))*CGRI(ID))
613 IF T21 LT T11+T12
614 LET T2 = T21-T11
615 LET T3 = T12-T2
616 LET SR3 = ((MFRI(ID)+MSR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
617 IF T21 LT T11
618 LET T1 = T21
619 LET T2 = T11-T21
620 LET T3 = T12
621 LET SR2 = (2.*MFR1(ID)*CGRI(ID)*SDIF(ID))/PLGT(ID)
622 ALWAYS
623 IF CL2 GT CCL2
624 LET T22 = (CCL2-CCL1)/(MFRI(ID)+CGRI(ID))
625 LET T4 = T22
626 LET SR5 = ((MSR2(ID)+MFR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
627 IF T21 LT T11+T12
628 LET T3 = T12-T22
629 LET T4 = T22-T3
630 IF T21+T22 LT T11+T12
631 LET T3 = T22
632 LET T4 = T12-T21+T22
633

```

```

634 LET SR4 = (2*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
635 ALWAYS
636 IF T21 LT T11
637 LET T3 = T22-T2
638 LET T4 = T12-T3
639 IF T11+T12 LT T21+T22
640 LET T3 = T12
641 LET T4 = T22-T2-T3
642 LET SR4 = ((MFW1(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
643 ALWAYS
644 ALWAYS
645 ALWAYS
646 IF CL2 GT CCL3
647 LET SR6 = (2*MFW2(ID)*CGRI(ID)*SDIF(ID))/HLGT(ID)
648 LET T23 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
649 LET T5 = T23
650 IF T21+T22 LT T11+T12
651 LET T5 = T23-T4
652 ALWAYS
653 ALWAYS
654 ALWAYS
655 ALWAYS
656 LET S1 = SRES(ID) - T1*SR1
657 LET S2 = S1 - T2*SR2
658 LET K15 = EXP.F(R*SR1*T1)-1.0
659 LET K16 = (A*EXP.F(R*S1))/(B*SR2)
660 LET K17 = EXP.F(B*SR2*T2)-1.0
661 LET K18 = (A*EXP.F(R*S2))/(B*SR3)
662 IF CL2 GT CCL1
663 LET S3 = S2 - T3*SR3
664 LET K19 = EXP.F(B*SR3*T3)-1.0
665 LET K20 = (A*EXP.F(B*S3))/(B*SR4)
666 IF CL2 GT CCL2
667 LET S4 = S3 - T4*SR4
668 LET K21 = EXP.F(B*SR4*T4)-1.0
669 LET K22 = (A*EXP.F(B*S4))/(B*SR5)
670 IF CL2 GT CCL3
671 LET S5 = S4 - T5*SR5
672 LET K23 = EXP.F(B*SR5*T5)-1.0
673 LET K24 = (A*EXP.F(B*S5))/(B*SR6)
674 LET K25 = EXP.F(B*SR6*(T1-T2-T3-T4-T5))-1.0
675 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21)*
676 (K22*K23)*(K24*K25))
677 LET SR = SR6
678 LET S5 = S5 - (T1-T2-T3-T4-T5)*SP6
679 JUMP AHEAD
680 ELSE
681 LET K23 = EXP.F(B*SR5*(T1-T2-T3-T4))-1.0
682 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21)*
683 (K22*K23))
684 LET SR = SR5
685 LET S5 = S4 - (T1-T2-T3)*SR5
686 JUMP AHEAD

```

```

687 ELSE
688 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
689 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)+(K16*K17)+(K18*K19)+(K20*K21))
690 LET SR = SR4
691 LET RS = S3 - (T-T1-T2-T3)*SR4
692 JUMP AHEAD
693 ELSE
694 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
695 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)+(K16*K17)+(K18*K19))
696 LET SR = SR3
697 LET RS = S2 - (T-T1-T2)*SR3
698 JUMP AHEAD
699 ELSE
700 IF CL1 GT CCL2
701 LET T11 = (CCL2-DL1)/(MFRI(ID)*CGRI(ID))
702 IF DL2 GT CCL1
703 LET T1 = T11
704 LET S1 = SRES(ID) - T1*SR1
705 LET SR2 = ((MFRI(ID)*MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
706 LET K15 = EXP.F(-B*SW1*(T1))-1.0
707 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
708 IF CL2 GT CCL2
709 LET T21 = (CCL2-DL1)/(MFRI(ID)*CGRI(ID))
710 LET T2 = T21-T1
711 LET S2 = S1 - T2*SR2
712 LET SR3 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
713 LET K17 = EXP.F(-B*SR2*(T2))-1.0
714 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
715 LET K19 = EXP.F(-B*SW3*(T-T1-T2))-1.0
716 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)+(K16*K17)+(K18*K19))
717 LET SR = SR3
718 LET RS = S2 - (T-T1-T2)*SR3
719 JUMP AHEAD
720 ELSE
721 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
722 LET HZRD(ID) = HZRD(ID)*EXP.F((K14*K15)+(K16*K17))
723 LET SR = SR2
724 LET RS = S1 - (T-T1)*SR2
725 JUMP AHEAD
726 ELSE
727 LET T1 = T11
728 IF CL2 GT CCL1
729 LET T21 = (CCL1-DL2)/(MSW1(ID)*CGRI(ID))
730 LET T2 = T21-T1
731 LET SR2 = ((MSW1(ID)*SR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
732 IF T2 LT T11
733 LET T1 = T21
734 LET T2 = T11-T21
735 LET SR2 = (2.*MFRI(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
736 ALWAYS
737 LET S1 = SRES(ID) - T1*SW1
738 LET S2 = S1 - T2*SR2
739 LET SR3 = ((MFRI(ID)*MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)

```

```

740 LET K15 = EXP.F(-B*SP1*T1)-1.0
741 LET K16 = (A-EXP.F(8*S1))/ (8*SR2)
742 LET K17 = EXP.F(-B*SR2*T2)-1.0
743 LET K18 = (A*EXP.F(8*S2))/ (8*SR3)
744 IF CL2 GT CCL2
745 LET T22 = (CCL2-CCL1)/(MF*K1(ID)*CGRI(ID))
746 LET T3 = T21+T22-T1-T2
747 LET S3 = S2 - T3*SR3
748 LET SR4 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
749 LET K19 = EXP.F(-B*SR3*T3)-1.0
750 LET K20 = (A*EXP.F(8*SR3))/ (8*SR4)
751 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
752 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K18*K17)*(K18*K19)*(K20*K21))
753 LET SR = SR4
754 LET RS = S3 - (T-T1-T2-T3)*SR4
755 JUMP AHEAD
756 ELSE
757 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
758 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17)*(K18*K19))
759 LET SR = SR3
760 LET RS = S2 - (T-T1-T2)*SR3
761 JUMP AHEAD
762 ELSE
763 LET SR2 = ((MSR1(ID)*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
764 LET S1 = SRES(ID) - T1*SR1
765 LET K15 = EXP.F(-B*SR1*T1)-1.0
766 LET K16 = (A*EXP.F(8*S1))/ (8*SR2)
767 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
768 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17))
769 LET SR = SR2
770 LET RS = S1 - (T-T1)*SR2
771 JUMP AHEAD
772 ELSE
773 IF CL2 GT CCL1
774 LET T1 = (CCL1-OL2)/(MSR1(ID)*CGRI(ID))
775 LET S1 = SRES(ID) - T1*SR1
776 LET SR2 = (2.*MF*K1(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
777 LET K15 = EXP.F(-B*SR1*T1)-1.0
778 LET K16 = (A*EXP.F(8*S1))/ (8*SR2)
779 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
780 LET MZRD(ID) = MZRD(ID)*EXP.F((K14*K15)*(K16*K17))
781 LET SR = SR2
782 LET RS = S1 - (T-T1)*SR2
783 JUMP AHEAD
784 ELSE
785 LET MZRD(ID) = MZRD(ID)*EXP.F(K14*K15)
786 JUMP AHEAD
787 ELSE
788 LET MZRD = EXP.F(K14*K15)
789 IF CL1 GT CCL1
790 LET T1 = (CCL1-OL1)/(MSR1(ID)*CGRI(ID))
791 LET T1 = T1
792 LET S1 = SRES(ID) - T1*SR1

```

```

793 LET SR2 = (MSR1(ID)*MFRI(ID))*CGR1(ID)*SDIF(ID)/RLGT(ID)
794 LET K15 = EXP.F(-B*SR1*T1)-1.0
795 LET K16 = (A*EXP.F(B*S1))/(B*SR2)
796 LET K17 = EXP.F(-B*SR2*(T-T1))-1.0
797 LET AZRD = EXP.F(K14*K15)*(K16*K17))
798 LET SR = SR2
799 LET RS = S1 - (T-T1)*SR2
800 IF CL2 GT CCL1
801 LET T21 = (CCL1-DL2)/(MSR1(ID)*CGR1(ID))
802 LET T2 = T21-T11
803 LET SR3 = (2.*MFRI(ID)*CGR1(ID)*SDIF(ID))/RLGT(ID)
804 LET S2 = S1 - T2*SR2
805 LET K17 = EXP.F(-B*SR2*T2)-1.0
806 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
807 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
808 LET AZRD = EXP.F(K14*K15)*(K16*K17)*(K18*K19))
809 LET SR = SR3
810 LET RS = S2 - (T-T1-T2)*SR3
811 ALWAYS
812 IF CL1 GT CCL2
813 LET T12 = (CCL2-CCL1)/(MFRI(ID)*CGR1(ID))
814 LET T2 = T12
815 LET SR3 = (MSR1(ID)*MSR2(ID))*CGR1(ID)*SDIF(ID)/RLGT(ID)
816 LET S2 = S1 - T2*SR2
817 LET K17 = EXP.F(-B*SR2*T2)-1.0
818 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
819 LET K19 = EXP.F(-B*SR3*(T-T1-T2))-1.0
820 LET AZRD = EXP.F(K14*K15)*(K16*K17)*(K18*K19))
821 LET SR = SR3
822 LET RS = S2 - (T-T1-T2)*SR3
823 IF CL2 GT CCL1
824 LET T3 = T21-T11-T12
825 LET SR4 = (MFRI(ID)*MSR2(ID))*CGR1(ID)*SDIF(ID)/RLGT(ID)
826 LET S3 = S2 - T3*SR3
827 LET K19 = EXP.F(-B*SR3*T3)-1.0
828 LET K20 = (A*EXP.F(B*S3))/(B*SR4)
829 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
830 LET AZRD = EXP.F(K14*K15)*(K16*K17)*(K18*K19)*(K20*K21))
831 LET SR = SR4
832 LET RS = S3 - (T-T1-T2-T3)*SR4
833 IF T21 LT T11+T12
834 LET T2 = T21-T11
835 LET T3 = T12-T2
836 LET SR3 = (2.*MFRI(ID)*CGR1(ID)*SDIF(ID))/RLGT(ID)
837 LET S2 = S1 - T2*SR2
838 LET S3 = S2 - T3*SR3
839 LET K17 = EXP.F(-B*SR2*T2)-1.0
840 LET K18 = (A*EXP.F(B*S2))/(B*SR3)
841 LET K19 = EXP.F(-B*SR3*T3)-1.0
842 LET K20 = (A*EXP.F(B*S3))/(B*SR4)
843 LET K21 = EXP.F(-B*SR4*(T-T1-T2-T3))-1.0
844 LET AZRD = EXP.F(K14*K15)*(K16*K17)*(K18*K19)*(K20*K21))
845 LET SR = SR4

```

```

846 LET RS = S3 - (T-T1-T2-T3)*SR4
847 IF CL2 GT CCL2
848   ALWAYS
849   IF CL2 GT CCL2
850     LET T2 = (CCL2-CCL1)/(MFR1(ID)*CGRI(ID))
851     LET T4 = T2*(T2-T1-T2-T3)
852     LET S4 = S3 - T4*SR4
853     LET SR5 = (2.*MSR2(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
854     LET K21 = EXP.F((-8*SR4*T4)-1.0)
855     LET K22 = (A*EXP.F(B*SR4))/((B*SR4)
856     LET K23 = EXP.F((-8*SR5*(T-T1-T2-T3-T4))-1.0)
857     LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21)*(K22*K23))
858     LET SR = SR5
859     LET RS = S4 - (T-T1-T2-T3)*SR5
860   ALWAYS
861   IF CL1 GT CCL1
862     LET T13 = (CCL3-CCL2)/(MSK2(ID)*CGRI(ID))
863     LET T3 = T13
864     LET SR4 = ((MSR1(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
865     LET S3 = S2 - T3*SR3
866     LET K19 = EXP.F((-8*SK3*T3)-1.0)
867     LET K20 = (A*EXP.F(B*SR3))/((B*SR3)
868     LET K21 = EXP.F((-8*SR4*(T-T1-T2-T3))-1.0)
869     LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21))
870     LET SR = SR4
871     LET RS = S3 - (T-T1-T2-T3)*SR4
872   IF CL2 GT CCL1
873     LET T4 = T21-T11-T12-T13
874     LET SR5 = ((MFR1(ID)*MFR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
875     IF T21 LT T11-T12-T13
876       LET T3 = T21-T11-T12
877       LET T4 = T13-T3
878       LET SR4 = ((MFR1(ID)*MSR2(ID))*CGRI(ID)*SDIF(ID))/RLGT(ID)
879       ALWAYS
880       IF T21 LT T11-T12
881         LET T2 = T21-T11
882         LET T3 = T12-T2
883         LET T4 = T13
884         LET SR3 = (2.*MFR1(ID)*CGRI(ID)*SDIF(ID))/RLGT(ID)
885       ALWAYS
886       LET S2 = S1 - T2*SR2
887       LET S3 = S2 - T3*SR3
888       LET S4 = S3 - T4*SR4
889       LET K17 = EXP.F((-8*SR2*T2)-1.0)
890       LET K18 = (A*EXP.F(B*SR2))/((B*SR2)
891       LET K19 = EXP.F((-8*SR3*T3)-1.0)
892       LET K20 = (A*EXP.F(B*SR3))/((B*SR3)
893       LET K21 = EXP.F((-8*SR4*T4)-1.0)
894       LET K22 = (A*EXP.F(B*SR4))/((B*SR4)
895       LET K23 = EXP.F((-8*SR5*(T-T1-T2-T3-T4))-1.0)
896       LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21)*(K22*K23))
897       LET SR = SR5
898       LET RS = S4 - (T-T1-T2-T3)*SR5

```

```

899  ALWAYS
900  IF CL2 GT CCL2
901      LET T4 = T13
902      LET T5 = T21-T22-T11-T12-T13
903      LET SR6 = ((MSR2(ID)+MR2(ID))*CGRI(ID)*SDIF(ID))/PLGT(ID)
904      IF T21-T22 LT T11-T12-T13
905          LET T4 = T21-T22-T11-T12-T13
906          LET T5 = T13-T4
907          LET SR5 = T2-MSK2(ID)*CGRI(ID)*SDIF(ID)/PLGT(ID)
908      ALWAYS
909      LET S4 = S3 - T4*SR4
910      LET S5 = S4 - T5*SR5
911      LET K21 = EXP.F(-B*SR4*T4)-1.0
912      LET K22 = (A*EXP.F(B*S4))/(B*SR5)
913      LET K23 = EXP.F(-B*SR5*T5)-1.0
914      LET K24 = (A*EXP.F(B*S5))/(B*SR6)
915      LET K25 = EXP.F(-B*SR6*(T1-T1-T2-T3-T4-T5))-1.0
916      LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21)*(K22*K23)*
917          (K24*K25))
918      LET SR = SR6
919      LET RS = S5 - (T1-T1-T2-T3-T4)*SR6
920      ALWAYS
921      IF CL2 GT CCL3
922          LET T6 = T21-T11
923          LET SR7 = (2-MSR2(ID)*CGRI(ID)*SDIF(ID))/PLGT(ID)
924          LET K25 = EXP.F(-B*SR6*T6)-1.0
925          LET S6 = S5 - T6*SR6
926          LET K26 = (A*EXP.F(B*S6))/(B*SR7)
927          LET K27 = EXP.F(-B*SR7*(T1-T1-T2-T3-T4-T5-T6))-1.0
928          LET AZRD = EXP.F((K14*K15)*(K16*K17)*(K18*K19)*(K20*K21)*(K22*K23)*
929              (K24*K25)*(K26*K27))
930          LET SR = SR7
931          LET RS = S6 - (T1-T1-T2-T3-T4-T5)*SR7
932      ALWAYS
933      ALWAYS
934      ALWAYS
935      ALWAYS
936      LET HZRD(ID) = HZRD(ID)+AZRD
937      HERE
938      LET SRRATE(ID) = SP
939      LET SRES(ID) = RS
940      RETURN
941      END
    
```

1-41

SYMBOLIC REFERENCE MAP (R = ?) - ROUTINE COMP.RISK

A	LOCAL RECURSIVE	< 2	REAL	64 REFS
AC	GLOBAL VARIABLE	(1-D) INTEGER	(1-D) REAL	1 REFS
AFAC7	GLOBAL VARIABLE	(1-D) REAL	REAL	1 REFS
AREAN	GLOBAL VARIABLE	< 43	REAL	3 REFS
ARG	LOCAL RECURSIVE	< 26	REAL	3 REFS
ARG1	LOCAL RECURSIVE	< 37	REAL	22 REFS
AZRD	LOCAL RECURSIVE	(1-D) INTEGER	REAL	1 REFS
A1SR	GLOBAL VARIABLE	(1-D) INTEGER	REAL	235 REFS
AZSR	GLOBAL VARIABLE	(1-D) INTEGER	REAL	59 REFS
B	LOCAL RECURSIVE	< 10	REAL	81 REFS
CCL1	LOCAL RECURSIVE	< 11	REAL	72 REFS
CCL2	LOCAL RECURSIVE	< 12	REAL	3 REFS
CCL3	LOCAL RECURSIVE	< 4	REAL	122 REFS
CGR	LOCAL RECURSIVE	< 1	REAL	7 REFS
CGR1	GLOBAL VARIABLE	< 17	REAL	36 REFS
CL	YIELDED ARGUMENT	< 20	REAL	58 REFS
CL1	LOCAL RECURSIVE	(1-D) INTEGER	REAL	1 REFS
CL2	LOCAL RECURSIVE	REAL	REAL	2 REFS
COMP.RISK	PROCEDURE	(1-D) ALPHA	REAL	1 REFS
COME	GLOBAL VARIABLE	REAL	REAL	17 REFS
CONGSION	EVENT NOTICE	REAL	REAL	54 REFS
CO.FIXITS	GLOBAL VARIABLE	REAL	REAL	43 REFS
CTHREE	GLOBAL VARIABLE	REAL	REAL	229 REFS
CTMO	GLOBAL VARIABLE	(1-D) REAL	REAL	86 REFS
DLC1	LOCAL RECURSIVE	< 16	REAL	806 REFS
DL1	LOCAL RECURSIVE	< 15	REAL	60 REFS
DL2	LOCAL RECURSIVE	< 14	REAL	71 REFS
EXP.F	PROCEDURE	(1-D) REAL	REAL	73 REFS
HZRD	GLOBAL VARIABLE	< 41	REAL	51 REFS
ID	GLOBAL VARIABLE	< 42	REAL	48 REFS
K14	LOCAL RECURSIVE	< 47	REAL	30 REFS
K15	LOCAL RECURSIVE	< 48	REAL	34 REFS
K16	LOCAL RECURSIVE	< 56	REAL	15 REFS
K17	LOCAL RECURSIVE	< 57	REAL	5 REFS
K18	LOCAL RECURSIVE	< 63	REAL	6 REFS
K19	LOCAL RECURSIVE	< 68	REAL	2 REFS
K20	LOCAL RECURSIVE	< 69	REAL	53 REFS
K21	LOCAL RECURSIVE	(1-D) REAL	REAL	49 REFS
K22	LOCAL RECURSIVE	(1-D) REAL	REAL	29 REFS
K23	LOCAL RECURSIVE	(1-D) REAL	REAL	74 REFS
K24	LOCAL RECURSIVE	(1-D) REAL	REAL	2 REFS
K25	LOCAL RECURSIVE	(1-D) REAL	REAL	69 REFS
K26	LOCAL RECURSIVE	(1-D) REAL	REAL	49 REFS
K27	LOCAL RECURSIVE	(1-D) REAL	REAL	49 REFS
MFR1	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MFR2	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR1	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR2	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR3	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR4	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR5	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR6	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR7	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR8	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR9	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR10	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR11	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR12	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR13	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR14	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR15	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR16	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR17	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR18	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR19	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR20	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR21	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR22	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR23	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR24	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR25	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR26	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR27	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR28	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR29	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR30	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR31	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR32	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR33	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR34	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR35	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR36	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR37	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR38	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR39	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR40	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR41	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR42	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR43	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR44	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR45	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR46	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR47	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR48	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR49	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR50	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR51	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR52	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR53	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR54	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR55	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR56	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR57	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR58	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR59	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR60	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR61	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR62	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR63	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR64	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR65	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR66	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR67	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR68	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR69	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR70	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR71	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR72	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR73	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR74	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR75	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR76	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR77	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR78	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR79	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR80	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR81	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR82	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR83	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR84	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR85	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR86	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR87	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR88	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR89	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR90	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR91	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR92	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR93	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR94	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR95	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR96	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR97	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR98	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR99	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS
MSR100	GLOBAL VARIABLE	(1-D) REAL	REAL	49 REFS

```

SDIF          GLOBAL VARIABLE          < 5          ( 1-D) REAL          69 REFS
SR            LOCAL RECURSIVE          < 5          ( 1-D) REAL          49 REFS
SRES          GLOBAL VARIABLE          < 5          ( 1-D) REAL          26 REFS
SRRATE        GLOBAL VARIABLE          < 5          ( 1-D) REAL          2 REFS
SR1           LOCAL RECURSIVE          < 4          ( 1-D) REAL          49 REFS
SR2           LOCAL RECURSIVE          < 22         REAL          129 REFS
SR3           LOCAL RECURSIVE          < 39         REAL          107 REFS
SR4           LOCAL RECURSIVE          < 46         REAL          67 REFS
SR5           LOCAL RECURSIVE          < 55         REAL          31 REFS
SR6           LOCAL RECURSIVE          < 59         REAL          12 REFS
SR7           LOCAL RECURSIVE          < 66         REAL          5 REFS
S1            LOCAL RECURSIVE          < 23         REAL          77 REFS
S2            LOCAL RECURSIVE          < 40         REAL          63 REFS
S3            LOCAL RECURSIVE          < 45         REAL          40 REFS
S4            LOCAL RECURSIVE          < 54         REAL          19 REFS
S5            LOCAL RECURSIVE          < 61         REAL          7 REFS
S6            LOCAL RECURSIVE          < 67         REAL          3 REFS
T            LOCAL RECURSIVE          < 3           REAL          105 REFS
TAC           LOCAL RECURSIVE          < 14         REAL          10 REFS
TAL           LOCAL RECURSIVE          < 13         REAL          5 REFS
TA2           LOCAL RECURSIVE          < 18         REAL          3 REFS
TIME.A        TEMPORARY ATTRIBUTE
TIME.V        SYSTEM ATTRIBUTE
TRCHG         GLOBAL VARIABLE          ( 1-D) REAL          2 REFS
TWO          UNSUBSCRIPTED LABEL          2 REFS
T1            LOCAL RECURSIVE          < 21         REAL          156 REFS
T11           LOCAL RECURSIVE          < 49         REAL          54 REFS
T12           LOCAL RECURSIVE          < 58         REAL          35 REFS
T13           LOCAL RECURSIVE          < 64         REAL          10 REFS
T2            LOCAL RECURSIVE          < 38         REAL          178 REFS
T21           LOCAL RECURSIVE          < 50         REAL          58 REFS
T22           LOCAL RECURSIVE          < 51         REAL          28 REFS
T23           LOCAL RECURSIVE          < 52         REAL          5 REFS
T3            LOCAL RECURSIVE          < 44         REAL          49 REFS
T4            LOCAL RECURSIVE          < 53         REAL          39 REFS
T5            LOCAL RECURSIVE          < 60         REAL          12 REFS
T6            LOCAL RECURSIVE          < 65         REAL          4 REFS
1-STRENGTH-REDUCTION EVENT NOTICE
2-CR-EXISTS          GLOBAL VARIABLE          ( 1-D) ALPHA          2 REFS
2-STRENGTH-REDUCTION EVENT NOTICE

```

```

1 ROUTINE ESTIMATE-FAILURE-RATE YIELDING ELTYP-FAIL-RATE
2 DEFINE NUM.CRKS=NN+1-J+K+KI+K+KK AS INTEGER VARIABLES
3 DEFINE TEMP AS A REAL+1-DIMENSIONAL ARRAY
4 RESERVE TEMP(*) AS 5
5 IF NCZ = 0
6 LET ELTYP-FAIL-RATE = 0.0
7 RETURN
8 ALWAYS
9 IF NCZ = 1
10 LET PF = (1 - HZ(1)) * POP.SIZE / SAMP.SIZE
11 GO TO FBI
12 ALWAYS
13 LET SUMLPF = 0.0
14 LET SUMLPCL = 0.0
15 LET C = 0.0
16 LET PROB = 0.0
17 FOR I = 1 TO NCZ
18 DO
19 LET PF = 1.0 - HZ(I)
20 LET PROB = PROB + PF
21 COMPUTE SUML AS THE SUM. AVGCL AS THE MEAN. STDL AS THE STD.DEV AND
22 SSCL AS THE SUM.OF.SQUARES OF CLGT(I)
23 IF HZ(I) GE 1.0
24 LET C = C + 1.0
25 JUMP AHEAD
26 ALWAYS
27 LET LPF = LOG.12.F(PF)
28 LET SUMLPF = SUMLPF + LPF
29 LET SUMLPCL = SUMLPCL + LPF * CLGT(I)
30 HERE
31 LOOP
32 IF C = NCZ OR C-1 = NCZ
33 LET ELTYP-FAIL-RATE = 0.0
34 RETURN
35 ALWAYS
36 LET DET = (NCZ-C) * SSCL - SUMLPCL * SUMLPCL
37 LET AEXP = (SUMLPF*SSCL - SUMLPCL * SUMLPCL)/DET
38 LET BA = ((NCZ-C) * SUMLPCL - SUMLPCL * SUMLPF)/DET
39 LET NUM.CRKS = NCZ*(POP.SIZE/SAMP.SIZE) - NCZ
40 LET NN = POP.SIZE/SAMP.SIZE
41 IF SEL1 = "FUS-"
42 IF SEL2 = "MFR-" OR SEL3 = "FRN-"
43 FOR I = 1 TO NCZ
44 DO
45 FOR J = 1 TO 5
46 DO
47 IF CLGT(I)/SSCL LET GT AVUSC(J)
48 LET KI = J-1+MM
49 IF KI GE 5
50 LET KI = 5
51 GO TO A1
52 ALWAYS
53 FOR K = KI TO 4

```

```

54 DO
55 LET TEMP(K+1) = AVPSA(K+1)-NA,
56 LOOP
57 LET IK = KI+1
58 FOR K = IK TO 5
59 DO
60 LET AVPSA(K) = TEMP(K)
61 LOOP
62 FOR K = J TO KI
63 DO
64 LET AVPSA(K) = CLGT(I)/FSAF.LGT
65 LOOP
66 JUMP AHEAD
67 ALWAYS
68 LOOP
69 HERE
70 LOOP
71 GO TO A2
72 OTHERWISE
73 ALWAYS
74 FOR I = 1 TO NCZ
75 DO
76 FOR J = 1 TO 5
77 DO
78 IF CLGT(I)/FSAF.LGT GT AVPSA(J)
79 LET KI = J-1+NN
80 IF KI GE 5
81 LET KI = 5
82 GO TO A3
83 ALWAYS
84 FOR K = KI TO 4
85 DO
86 LET TEMP(K+1) = AVPSA(K+1)-NN
87 LOOP
88 LET IK = KI+1
89 FOR K = IK TO 5
90 DO
91 LET AVPSA(K) = TEMP(K)
92 LOOP
93 FOR K = J TO KI
94 DO
95 LET AVPSA(K) = CLGT(I)/FSAF.LGT
96 LOOP
97 JUMP AHEAD
98 ALWAYS
99 LOOP
100 HERE
101 LOOP
102 NA2# IF SEL1 = "FUS-"
103 IF SEL2 = "MFR-" OR SEL3 = "FAM-"
104 FOR I = 1 TO NCZ
105 DO
106 FOR J = 1 TO 5

```

```

187 DO
188 IF CLGT(I)/FSAF.LGT GT PSAVE(J)
189 FOR K = J TO 4
190 DO
191 LET TEMP(K+1) = PSAVE(K)
192 LOOP
193 LET KK = J+1
194 FOR K = KK TO 5
195 DO
196 LET PSAVE(K) = TEMP(K)
197 LOOP
198 LET PSAVE(J) = CLGT(I)/FSAF.LGT
199 JUMP AHEAD
200 ALWAYS
201 LOOP
202 HERE
203 LOOP
204 GO TO A4
205 OTHERWISE
206 ALWAYS
207 FOR I = 1 TO NCZ
208 DO
209 FOR J = 1 TO 5
210 DO
211 IF CLGT(I)/FSAF.LGT GT FSAVE(J)
212 FOR K = J TO 4
213 DO
214 LET TEMP(K+1) = FSAVE(K)
215 LOOP
216 LET KK = J+1
217 FOR K = KK TO 5
218 DO
219 LET FSAVE(K) = TEMP(K)
220 LOOP
221 LET FSAVE(J) = CLGT(I)/FSAF.LGT
222 JUMP AHEAD
223 ALWAYS
224 LOOP
225 HERE
226 LOOP
227 FOR I = 1 TO NUM.CNKS
228 DO
229 LET CL = LOG.NORMAL.F(AVGL.STDL.1)
230 IF SEL1 = "FUS-"
231 IF SEL2 = "MFR-" OR SEL3 = "FRN-"
232 FOR J = 1 TO 5
233 DO
234 IF CL/FSAF.LGT GT PSAVE(J)
235 FOR K = J TO 4
236 DO
237 LET TEMP(K+1) = PSAVE(K)
238 LOOP
239 LET KK = J+1
240 FOR K = KK TO 5
241 DO
242 LET PSAVE(K) = TEMP(K)
243 LOOP
244 JUMP AHEAD
245 ALWAYS
246 LOOP
247 HERE
248 LOOP
249 FOR I = 1 TO NUM.CNKS
250 DO
251 LET CL = LOG.NORMAL.F(AVGL.STDL.1)
252 IF SEL1 = "FUS-"
253 IF SEL2 = "MFR-" OR SEL3 = "FRN-"
254 FOR J = 1 TO 5
255 DO
256 IF CL/FSAF.LGT GT PSAVE(J)
257 FOR K = J TO 4
258 DO
259 LET TEMP(K+1) = PSAVE(K)
260 LOOP
261 LET KK = J+1
262 FOR K = KK TO 5
263 DO
264 LET PSAVE(K) = TEMP(K)
265 LOOP
266 JUMP AHEAD
267 ALWAYS
268 LOOP
269 HERE
270 LOOP

```

```

160 FOR K = KK TO 5
161 DO
162 LET PSAVE(K) = TEMP(K)
163 LOOP
164 LET PSAVE(J) = CL/FSAF.LGT
165 JUMP AHEAD
166 ALWAYS
167 LGOP
168 HERE
169 GO TO A5
170 OTHERWISE
171 ALWAYS
172 FOR J = 1 TO 5
173 DO
174 IF CL/FSAF.LGT GT FSAVE(J)
175 FOR K = J TO 4
176 DO
177 LET TEMP(K+1) = FSAVE(K)
178 LGOP
179 LET KK = J+1
180 FOR K = KK TO 5
181 DO
182 LET FSAVE(K) = TEMP(K)
183 LOOP
184 LET FSAVE(J) = CL /FSAF.LGT
185 JUMP AHEAD
186 ALWAYS
187 LOOP
188 HERE
189 BASP LET PFL = AEXP + B4 * CL
190 LET PF = 10.0 ** PFL
191 IF PF LT 0.0
192 LET PF = 0.0
193 ALWAYS
194 LET PROB = PROB + PF
195 LOOP
196 #FAIL LET I = SIZE.OF.FLEET * USAGE.LIFE
197 LET ELTYP*FAIL.RATE =PROB/I
198 RELEASE TEMP(*)
199 RETURN
200 END

```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE ESTIMATE.FAILURE.RAT

AEPS	GLOBAL VARIABLE	REAL	2 REFS
AVPSA	GLOBAL VARIABLE	(1-D) REAL	4 REFS
AVBL	GLOBAL VARIABLE	REAL	2 REFS
AVPSA	GLOBAL VARIABLE	(1-D) REAL	4 REFS
A1	UNSCRIPTED LABEL		2 REFS
A2	UNSCRIPTED LABEL		2 REFS
A3	UNSCRIPTED LABEL		2 REFS
A4	UNSCRIPTED LABEL		2 REFS
A5	UNSCRIPTED LABEL		2 REFS
BA	GLOBAL VARIABLE	REAL	2 REFS
C	LOCAL RECURSIVE	REAL	7 REFS
CL	LOCAL RECURSIVE	REAL	6 REFS
CLGT	LOCAL VARIABLE	(1-D) REAL	10 REFS
DET	LOCAL RECURSIVE	REAL	3 REFS
ELTYP.FAIL.RATE	YIELDED ARGUMENT	REAL	4 REFS
	GLOBAL VARIABLE	INTEGER	1 REFS
ESTIMATE.FAILURE.RAT	PROCEDURE		12 REFS
FAI	UNSCRIPTED LABEL		4 REFS
FSAP.LGT	GLOBAL VARIABLE	REAL	2 REFS
FSAVE	GLOBAL VARIABLE	(1-D) REAL	12 REFS
HZ	GLOBAL VARIABLE	REAL	4 REFS
I	LOCAL RECURSIVE	(1-D) REAL	3 REFS
IK	LOCAL RECURSIVE	INTEGER	19 REFS
J	LOCAL RECURSIVE	INTEGER	5 REFS
K	LOCAL RECURSIVE	INTEGER	29 REFS
KI	LOCAL RECURSIVE	INTEGER	41 REFS
KK	LOCAL RECURSIVE	INTEGER	13 REFS
LOG.NORMAL.F	LOCAL RECURSIVE	INTEGER	9 REFS
LOG.18.F	LOCAL RECURSIVE	REAL	1 REFS
LPF	PROCEDURE	REAL	1 REFS
MCZ	PROCEDURE	REAL	3 REFS
NN	LOCAL RECURSIVE	INTEGER	13 REFS
NUM.CPRS	LOCAL RECURSIVE	INTEGER	6 REFS
-PF	LOCAL RECURSIVE	INTEGER	3 REFS
PFL	LOCAL RECURSIVE	REAL	8 REFS
POP.SIZE	LOCAL RECURSIVE	REAL	2 REFS
PROB	LOCAL RECURSIVE	REAL	3 REFS
PSAVE	LOCAL RECURSIVE	REAL	6 REFS
SAMP.SIZE	LOCAL RECURSIVE	REAL	8 REFS
SEL1	GLOBAL VARIABLE	REAL	3 REFS
SEL2	GLOBAL VARIABLE	ALPHA	3 REFS
SEL3	GLOBAL VARIABLE	ALPHA	3 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE	INTEGER	1 REFS
SQL	LOCAL RECURSIVE	REAL	3 REFS
STD	LOCAL RECURSIVE	REAL	2 REFS
SUM	LOCAL RECURSIVE	REAL	5 REFS
SUM.PCL	LOCAL RECURSIVE	REAL	5 REFS
SUM.PF	LOCAL RECURSIVE	REAL	5 REFS
T	LOCAL RECURSIVE	REAL	2 REFS

PAGE 57

03/02/78. 18.36.02.

KRONOS 2.1.2

CACI SIMSCRIPT (1.5. VERSION /4.9-00/

CDC 6600

15 REFS
1 REFS

REAL
REAL

LOCAL RECURSIVE
GLOBAL VARIABLE

TEMP
USAGE.LIFE

```

1 ROUTINE REAL.LIFE(MEAN,STD.DEV,PDL) YIELDING RFL
2
3
4 PDL = PREDICTED AVERAGE FATIGUE LIFE OF ELEMENT DESIGN
5 RFL = ACTUAL AVERAGE FATIGUE LIFE OF ELEMENT DESIGN
6 RATIO = RATIO OF ACTUAL LIFE / PREDICTED LIFE
7 IF RATIO > 10.0 RATIO IS SET TO 10
8 IF RATIO < .1 RATIO IS SET TO .1
9 DISTRIBUTION OF RATIOS IS REPRESENTED AS LOG NORMAL
10 MEAN = MEAN OF DISTRIBUTION OF RATIOS
11 STD.DEV = STANDARD DEVIATION OF DISTRIBUTION OF RATIOS
12
13 LET RATIO = LOG.NORMAL.F(MEAN,STD.DEV,9)
14 IF RATIO > 10.0 LET RATIO = 10.0 REGARDLESS
15 IF RATIO < .1 LET RATIO = .1 REGARDLESS
16 LET RFL = RATIO * PDL
17 RETURN
18 END

```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE REAL.LIFE

LOG.NORMAL.F	PROCEDURE	REAL	1 REFS
MEAN	GIVEN ARGUMENT	REAL	2 REFS
PDL	GIVEN ARGUMENT	REAL	2 REFS
RATIO	LOCAL RECURSIVE	REAL	6 REFS
REAL.LIFE	PROCEDURE	INTEGER	1 REFS
RFL	YIELDED ARGUMENT	REAL	2 REFS
STD.DFV	GIVEN ARGUMENT	REAL	2 REFS

```

1  EVENT ENTER.SERVICE
2  LET IDCK = IDCK + 1
3  LET ID = IDCK
4  CREATE AN AIRCRAFT CALLED AIRPLANE(ID)
5  LET TAIL.ID(AIRPLANE(ID)) = ID
6  LET ENTRY.TIME(AIRPLANE(ID)) = TIME.V
7  FILE AIRPLANE(ID) IN ACTIVE.FLEET
8  LET ASD = .15 * AMEAN
9  LET ASD = .15 * AMEAN
10 CALL FATIGUE.LIFE(SCATTER:ACTUAL.AVG.FAT.LIFE*8.3.7127) YIELDING FIRST.LIFE
11 AND SECOND.LIFE
12 LET STD.FAST1 = M2.MEAN * .15
13 LET STD.SLOW1 = M1.MEAN * .15
14 LET STD.FAST2 = M4.MEAN * .15
15 LET STD.SLOW2 = M3.MEAN * .15
16 LET INT.SLOW1 = INT.M1 * .15
17 LET INT.FAST1 = INT.M2 * .15
18 LET INT.SLOW2 = INT.M3 * .15
19 LET INT.FAST2 = INT.M4 * .15
20 LET PNI = 1.0 - RANDOM.F(5)
21 LET MSR1.INT(ID) = RATE(INT.M1.SLOW1.PNI)*AFACT(ID)
22 LET MSR1.INT(ID) = RATE(INT.M2.FAST1.PNI)*AFACT(ID)
23 LET MSR2.INT(ID) = RATE(INT.M3.SLOW2.PNI)*AFACT(ID)
24 LET MSR2.INT(ID) = RATE(INT.M4.FAST2.PNI)*AFACT(ID)
25 LET MSR1(ID) = RATE(M1.MEAN.STD.SLOW1.PNI)*AFACT(ID)
26 LET MSR1(ID) = RATE(M2.MEAN.STD.FAST1.PNI)*AFACT(ID)
27 LET MSR2(ID) = RATE(M3.MEAN.STD.SLOW2.PNI)*AFACT(ID)
28 LET MSR2(ID) = RATE(M4.MEAN.STD.FAST2.PNI)*AFACT(ID)
29 CALL PREDICT.CORROSION YIELDING HOURS.TC.CORROSION
30 CALL PREDICT.SERVICE.DAMAGE YIELDING HOURS.TC.SERVICE.DAMAGE
31 IF HOURS.TC.CORROSION LT USAGE.LIFE
32 SCHEDULE A CORROSION(ID) AT TIME.V + HOURS.TC.CORROSION
33 LET AC(ID) = CORROSION
34 LET CORP.TIME(ID) = TIME.V
35 LET CO.EXISTS(ID) = "NS"
36 REGARDLESS
37 **
38 IF OURS.TC.SERVICE.DAMAGE LT USAGE.LIFE
39 SCHEDULE AN IN.SERVICE.DAMAGE(ID) AT TIME.V + OURS.TC.SERVICE.DAMAGE
40 LET LAST.SO(ID) = TIME.V
41 LET SO.SCH(ID) = "YES"
42 REGARDLESS
43 IF RANDOM.F(7) LE BIRTH.DEFECT.PROBABILITY
44 IF LTNG = "NO"
45 PRINT 1 LINE WITH ID AS FOLLOWS
46 PRODUCTION DEFECT AIRCRAFT NO. ***
47 ALWAYS
48 LET RN = RANDOM.F(3)
49 LET DEFECT.LIFE = (-4042.6 * LOG.E.F(RN))*1.0132
50 ADD 1 TO OPD
51 ADD 1 TO GOPD
52 IF DEFECT.LIFE LT FIRST.LIFE
53 LET FIRST.LIFE = DEFECT.LIFE

```

```

53  REGARDLESS
54  REGARDLESS
55  IF LTMO = "YES"
56  FOR I = 1 TO NOAC(LDX)
57  DO
58  IF ID = TLID(LDX,I)
59  SKIP 1 OUTPUT LINE
60  PRINT 1 LINE WITH ID, TIME.V AS FOLLOWS
61  A/C NO. *** ENTERS SERVICE ***** HOURS FROM START OF SIMULATION
62  IF DEFECT.LIFE GT 0.0
63  PRINT 1 LINE AS FOLLOWS
64  PRODUCTION DEFECT PRESENT
65  ALWAYS
66  SKIP 1 OUTPUT LINE
67  PRINT 2 LINES WITH FIRST.LIFE, SECOND.LIFE AS FOLLOWS
68  1ST CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
69  2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
70  PRINT 8 LINES WITH MSR1(ID), MFR1(ID), MSR2(ID), MFR2(ID),
71  MSR1.INT(ID), MFR1.INT(ID), MSR2.INT(ID), MFR2.INT(ID) AS FOLLOWS
72  SLOW CRACK GROWTH RATE1 = ***** INCHES/HOUR
73  FAST CRACK GROWTH RATE1 = ***** INCHES/HOUR
74  SLOW CRACK GROWTH RATE2 = ***** INCHES/HOUR
75  FAST CRACK GROWTH RATE2 = ***** INCHES/HOUR
76  INTERNAL SLOW CRACK GROWTH RATE1 = ***** INCHES/HOUR
77  INTERNAL FAST CRACK GROWTH RATE1 = ***** INCHES/HOUR
78  INTERNAL SLOW CRACK GROWTH RATE2 = ***** INCHES/HOUR
79  INTERNAL FAST CRACK GROWTH RATE2 = ***** INCHES/HOUR
80  PRINT 1 LINE WITH CRF, AFACT(ID) AS FOLLOWS
81  CRF = *** A = ***
82  LEAVE
83  ELSE
84  LOOP
85  ALWAYS
86  IF FIRST.LIFE LT USAGE.LIFE OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
87  SCHEDULE A 1-STRENGTH-REDUCTION(ID) AT TIME.V + FIRST.LIFE
88  LET AISR(ID) = 1-STRENGTH-REDUCTION
89  LET CKRP.TIME(ID) = TIME.V
90  LET 1.CR.FAIST(ID) = "NS"
91  IF SECOND.LIFE LT USAGE.LIFE OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
92  SCHEDULE A 2-STRENGTH-REDUCTION(ID) AT TIME.V + SECOND.LIFE
93  LET A2SR(ID) = 2-STRENGTH-REDUCTION
94  LET 2.CR.EXISTS(ID) = "NS"
95  REGARDLESS
96  SCHEDULE A D.LEVEL-INSPECTION(ID) AT TIME.V + ARCD(4)
97  LET 1.LAST.D(ID) = TIME.V
98  LET ADL(ID) = D.LEVEL-INSPECTION
99  SCHEDULE A RETIRE-FROM-SERVICE(ID) AT TIME.V + USAGE.LIFE
100  IF IDCK EQ SIZE.OF.FLEET
101  RETURN
102  OTHERWISE
103  IF TIME.V = BEGIN.PRODUCTION GE PRCHG

```

CDC 6600 CACI SIMSCRIPT II.5 VERSION 74-0-00/ 03/02/78. 18.36.02. PAGE 62

03/02/78. 18.36.02.

```
93 SCHEDULE AN ENTER.SERVICE AT TIME.V + 2.PRODUCTION.TIME
94 RETURN
95 OTHERWISE
96 SCHEDULE AN ENTER.SERVICE AT TIME.V + PRODUCTION.TIME
97 RETURN
98 END
```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT ENTER.SERVICE

ABCD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
AC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACTIVE.FLEET	SET		1 REFS
ACTIVE.AVG.FAT.LIFE	GLOBAL VARIABLE	REAL	1 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AFAC	GLOBAL VARIABLE	(1-D) REAL	10 REFS
AIRCRAFT	TEMPORARY ENTITY		1 REFS
AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	4 REFS
AMEAN	GLOBAL VARIABLE	REAL	3 REFS
ASD	LOCAL RECURSIVE	REAL	2 REFS
ALSP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZSR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
REGIN.PRODUCTION	GLOBAL VARIABLE	REAL	1 REFS
BIRTH.DEFECT.PROBAB	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CKPEP.TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CORP.TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
COROSION	EVENT NOTICE		2 REFS
CO.FAISTS	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS
CMF	GLOBAL VARIABLE	REAL	1 REFS
DEFECT.LIFE	LOCAL RECURSIVE	REAL	4 REFS
D.LEVEL.INSPECTION	EVENT NOTICE		2 REFS
ENTER.SERVICE	EVENT NOTICE		3 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	REAL	1 REFS
FATIGUE.LIFE.SCATTER	PROCEDURE	INTEGER	1 REFS
FIRST.LIFE	LOCAL RECURSIVE	REAL	6 REFS
GOPD	GLOBAL VARIABLE	INTEGER	1 REFS
HOURS.TO.COROSION	LOCAL RECURSIVE	REAL	3 REFS
I	LOCAL RECURSIVE	REAL	2 REFS
ID	GLOBAL VARIABLE	INTEGER	57 REFS
IDCK	GLOBAL VARIABLE	INTEGER	4 REFS
INT.FAST1	LOCAL RECURSIVE	REAL	2 REFS
INT.FAST2	LOCAL RECURSIVE	REAL	2 REFS
INT.M1	GLOBAL VARIABLE	REAL	2 REFS
INT.M2	GLOBAL VARIABLE	REAL	2 REFS
INT.M3	GLOBAL VARIABLE	REAL	2 REFS
INT.M4	GLOBAL VARIABLE	REAL	2 REFS
INT.SLOW1	LOCAL RECURSIVE	REAL	2 REFS
INT.SLOW2	LOCAL RECURSIVE	REAL	2 REFS
IN.SERVICE.DAMAGE	EVENT NOTICE	(1-D) REAL	1 REFS
LAST.SD	GLOBAL VARIABLE	INTEGER	2 REFS
LDX	GLOBAL VARIABLE	REAL	1 REFS
LOG.E.F	PROCEDURE	REAL	1 REFS
LOG.NORMAL.F	PROCEDURE	REAL	2 REFS
LTHO	GLOBAL VARIABLE	ALPHA	2 REFS
MFR1	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MFR1.INT	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MFR2	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MFR2.INT	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MSR1	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MSR1.INT	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MSR2	GLOBAL VARIABLE	(1-D) REAL	2 REFS

MSP2.INT	GLOBAL VARIABLE	(1-0)	REAL	2 REFS
M1.MEAN	GLOBAL VARIABLE		REAL	2 REFS
M2.MEAN	GLOBAL VARIABLE		REAL	2 REFS
M3.MEAN	GLOBAL VARIABLE		REAL	2 REFS
M4.MEAN	GLOBAL VARIABLE		REAL	2 REFS
NOAC	GLOBAL VARIABLE		REAL	1 REFS
OPD	GLOBAL VARIABLE	(1-0)	INTEGER	1 REFS
OURS.TO.SERVICE.DAM	GLOBAL RECURSIVE		INTEGER	1 REFS
PRCHG	GLOBAL RECURSIVE	< 14	REAL	3 REFS
PREDICT.CONPROSION	GLOBAL VARIABLE		REAL	1 REFS
PREDICT.SERVICE.DAM	PROCEDURE		INTEGER	1 REFS
PRODUCTION.TIME	PROCEDURE		INTEGER	1 REFS
RANDOM.F	GLOBAL VARIABLE		REAL	1 REFS
RATE	PROCEDURE		REAL	3 REFS
RETIRE.FROM.SERVICE	PROCEDURE		REAL	4 REFS
RN	EVENT NOTICE		REAL	1 REFS
RNI	LOCAL RECURSIVE	< 25	REAL	2 REFS
SD.SCH	LOCAL RECURSIVE	< 12	REAL	9 REFS
SECOND.LIFE	LOCAL VARIABLE		REAL	3 REFS
SIZE.OF.FLEET	LOCAL RECURSIVE	(1-0)	ALPHA	4 REFS
STD.FAST1	GLOBAL VARIABLE	< 3	REAL	1 REFS
STD.FAST2	LOCAL RECURSIVE	< 4	REAL	2 REFS
STD.SLOW1	LOCAL RECURSIVE	< 6	REAL	2 REFS
STD.SLOW2	LOCAL RECURSIVE	< 5	REAL	2 REFS
TAIL.ID	LOCAL RECURSIVE	< 7	REAL	2 REFS
TIME.V	TEMPORARY ATTRIBUTE		INTEGER	1 REFS
TLID	SYSTEM ATTRIBUTE		REAL	15 REFS
T.LAST.D	GLOBAL VARIABLE	(2-0)	INTEGER	1 REFS
USAGE.LIFE	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
1.CP.EXISTS	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
1.STRENGTH.REDUCTION	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
2.CR.EXISTS	EVENT NOTICE		REAL	2 REFS
2.PRODUCTION.TIME	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
2.STRENGTH.REDUCTION	GLOBAL VARIABLE		REAL	1 REFS

```

1 ROUTINE FATIGUE.LIFE-SCATTER(RFL,N,ALPHA) YIELDING FIRST.LIFE AND SECOND.LIFE
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

```

00
 01 GENERATES ELEMENT FATIGUE LIVES REFLECTING BASIC FATIGUE SCATTER AND LOAD
 02 ENVIRONMENT VARIATION
 03
 04
 05 DISTRIBUTION OF FATIGUE LIVES IS REPRESENTED AS TWO PARAMETER WEIBULL
 06 ALPHA = SHAPE PARAMETER
 07 SEITA = SCALE PARAMETER (CHARACTERISTIC VALUE)
 08 N = RANDOM NUMBER STREAM
 09
 10
 11 DEFINE N AS AN INTEGER VARIABLE
 12 DEFINE LIFE AS A REAL, 1-DIMENSIONAL ARRAY
 13 RESERVE LIFE(*) AS 2
 14 LET BETA = RFL/.89018
 15 LET RN = RANDOM.F(N)
 16 FOR I = 1 TO 2, DO
 17 LET LIFE(I) = BETA * (LOG.E.F(1 / RN))**(1 / ALPHA)
 18 LET RN = UNIFORM.F(0.0, MN, 3)
 19 LOOP
 20 LET FIRST.LIFE = LIFE(1)
 21 LET SECOND.LIFE = LIFE(2)
 22 RELEASE LIFE(*)
 23 IF ALPHA LT 3.8
 24 LET FIRST.LIFE = FIRST.LIFE/AFACT(ID)
 25 LET SECOND.LIFE = SECOND.LIFE/AFACT(ID)
 26 ALWAYS
 27 RETURN
 28 END

SYMBOLIC REFERENCE MAP (P = 1) - ROUTINE FATIGUE.LIFE-SCATTER

AFACT	GLOBAL VARIABLE	< 3	(1-0)	REAL	2 REFS
ALPHA	GIVEN ARGUMENT	< 3		REAL	3 REFS
BETA	LOCAL RECURSIVE	< 7		REAL	2 REFS
FATIGUE.LIFE-SCATTER	PROCEDURE	< 4		INTEGER	1 REFS
FIRST.LIFE	YIELDED ARGUMENT	< 9		REAL	4 REFS
I	LOCAL RECURSIVE	< 9		REAL	2 REFS
ID	GLOBAL VARIABLE	< 6	(1-0)	INTEGER	2 REFS
LIFE	LOCAL RECURSIVE	< 6	(1-0)	REAL	6 REFS
LOG.E.F	PROCEDURE	< 2		REAL	1 REFS
N	GIVEN ARGUMENT	< 2		INTEGER	3 REFS
RANDOM.F	PROCEDURE	< 1		REAL	1 REFS
RFL	GIVEN ARGUMENT	< 4		REAL	2 REFS
RN	LOCAL RECURSIVE	< 4		REAL	4 REFS
SECOND.LIFE	YIELDED ARGUMENT	< 5		REAL	4 REFS
UNIFORM.F	PROCEDURE	< 5		REAL	1 REFS

```

1  ROUTINE INSTALL.MODIFICATION
2
3  ** THIS ROUTINE REPRESENTS THE INSTALLATION OF A STRUCTURAL MODIFICATION CAUSED
4  ** BY A FATIGUE TEST FAILURE OR BY A SERVICE DEFECT FOUND IN THE ELEMENT
5
6  LET CR.CTR = 0
7  LET JFLAG = "NO"
8  IF 1.CR.EXISTS(ID) = "YES"
9    CALL COMP-RISK YIELDING CL
10   LET NCZ = NCZ + 1
11   LET CLGT(NCZ) = CL
12   LET HZ(NCZ) = HZ00(ID)
13   LET PFID(NCZ) = ID
14   LET PFTIM(NCZ) = TIME.V-ENTRY.TIME(AIRPLANE(ID))
15   ALWAYS
16   LET NSMD = NSMD + 1
17   LET BEEN.MODIFIED = BEEN.MODIFIED + 1
18   LET OCCUR.MOD(ID) = TIME.V
19
20  ** CANCEL SCHEDULED INSPECTIONS
21
22  IF INSP.SCH(ID) = "YES"
23    CALL CANCEL.SCHEDULED.INSPECTIONS
24    REGARDLESS
25
26  ** IF FSH(ID) = "YES"
27  LET FSH(ID) = "NO"
28  LET FAILURE = AF(ID)
29  CANCEL THE FAILURE
30  DESTROY THE FAILURE
31  ALWAYS
32  CALL FATIGUE.LIFE-SCATTER(ACTUAL.AVG.FAT.LIFE*8.37127) YIELDING FIRST.LIFE
33  AND SECOND.LIFE
34  LET STD.SLOW1 = M1.MEAN * .15
35  LET STD.FAST1 = M2.MEAN * .15
36  LET STD.SLOW2 = M3.MEAN * .15
37  LET STD.FAST2 = M4.MEAN * .15
38  LET INT.SLOW1 = INT.M1 * .15
39  LET INT.FAST1 = INT.M2 * .15
40  LET INT.SLOW2 = INT.M3 * .15
41  LET INT.FAST2 = INT.M4 * .15
42  LET RN1 = 1.0 - RANDOM.F(5)
43  LET MSR1.INT(ID) = RATE(INT.M1,INT.SLOW1,RN1)*AFACT(ID)
44  LET MFR1.INT(ID) = RATE(INT.M2,INT.FAST1,RN1)*AFACT(ID)
45  LET MSR2.INT(ID) = RATE(INT.M3,INT.SLOW2,RN1)*AFACT(ID)
46  LET MFR2.INT(ID) = RATE(INT.M4,INT.FAST2,RN1)*AFACT(ID)
47  LET MSR1(ID) = RATE(M1.MEAN,STD.SLOW1,RN1)*AFACT(ID)
48  LET MFR1(ID) = RATE(M2.MEAN,STD.FAST1,RN1)*AFACT(ID)
49  LET MSR2(ID) = RATE(M3.MEAN,STD.SLOW2,RN1)*AFACT(ID)
50  LET MFR2(ID) = RATE(M4.MEAN,STD.FAST2,RN1)*AFACT(ID)
51  LET RST = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
52
53  ** CANCEL PREVIOUSLY SCHEDULED CRACK AND CORROSION INITIATIONS, RESCHEDULE

```

-159-

CDC 6600 CACI SIMSCRIPT II.5 VERSION /A.0-00/ PRO VS 2.1.2 03/02/78. 18.36.02. THESE EVENTS IF THEY OCCUR WITHIN THE REMAINING SERVICE LIFE (RST) OF THE AIRCRAFT

```

54 IF CO-EXISTS(ID) NE "NN"
55 LET COROSION = AC(ID)
56 IF CO-EXISTS(ID) = "NS"
57 CANCEL THE COROSION
58 ALWAYS
59 DESTROY THE COROSION
60 LET CO-EXISTS(ID) = "NN"
61 LET CORR(ID) = 1.0
62 REGARDLESS
63 CALL PREDICT-COROSION YIELDING HOURS.TO-COROSION
64 IF HOURS.TO-COROSION LT RST
65 SCHEDULE A COROSION(ID) AT TIME.V + HOURS.TO-COROSION
66 LET AC(ID) = COROSION
67 LET CORP-TIME(ID) = TIME.V
68 LET CO-EXISTS(ID) = "NS"
69 REGARDLESS
70 IF RANDOM.F(7) LE NIRTH.DEFECT.PROBABILITY
71 LET OPD = OPD + 1
72 LET RN = RANDOM.F(3)
73 LET DEFECT-LIFE = (-0.042.6 * LOG.E.F(RN))**1.0132
74 IF DEFECT-LIFE LT FIRST-LIFE
75 LET FIRST-LIFE = DEFECT-LIFE
76 REGARDLESS
77 REGARDLESS
78 IF 1.CR-EXISTS(ID) NE "NN"
79 LET 1-STRENGTH-REDUCTION = AISR(ID)
80 IF 1.CR-EXISTS(ID) = "NS"
81 CANCEL THE 1-STRENGTH-REDUCTION
82 REGARDLESS
83 DESTROY THE 1-STRENGTH-REDUCTION
84 LET 1.CR-EXISTS(ID) = "NN"
85 LET 2-STRENGTH-REDUCTION = A2SR(ID)
86 IF 2.CR-EXISTS(ID) = "NS"
87 CANCEL THE 2-STRENGTH-REDUCTION
88 REGARDLESS
89 DESTROY THE 2-STRENGTH-REDUCTION
90 LET 2.CR-EXISTS(ID) = "NN"
91 REGARDLESS
92 IF 1E1(ID) = "YES"
93 LET 1.IYE = AIE(ID)
94 LET 1E1(ID) = "NO"
95 CANCEL THE 1.IYE
96 DESTROY THE 1.IYE
97 IF 1E2(ID) = "YES"
98 LET 2.IYE = AIE(ID)
99 LET 1E2(ID) = "NO"

```

```

107 CANCEL THE 2.ITE
108 DESTROY THE 2.ITE
109 ALWAYS
110 ALWAYS
111 IF AIL(ID) = "YES"
112 LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
113 CANCEL THE REACH.FAIL.SAFE.LGT
114 DESTROY THE REACH.FAIL.SAFE.LGT
115 LET AIL(ID) = "NO"
116 REGARDLESS
117
118 IF FIRST.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SL.SCM(ID) = "YES"
119 SCHEDULE A 1-STRENGTH-REDUCTION(ID) AT TIME.V + FIRST.LIFE
120 LET AISR(ID) = 1-STRENGTH-REDUCTION
121 LET CAREP.TIME(ID) = TIME.V
122 LET 1-CR.EXISTS(ID) = "NS"
123 IF SECOND.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCM(ID) = "YES"
124 SCHEDULE A 2-STRENGTH-REDUCTION(ID) AT TIME.V + SECOND.LIFE
125 LET A2SR(ID) = 2-STRENGTH-REDUCTION
126 LET 2-CR.EXISTS(ID) = "NS"
127 REGARDLESS
128 REGARDLESS
129
130 IF LTHO = "YES"
131 FOR I = 1 TO NOAC(LDX)
132 DO
133 IF ID = TLID(LDX,I)
134 SKIP 1 OUTPUT LINE
135 PRINT 11 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)), FIRST.LIFE,
136 SECOND.LIFE, MSR1(ID), MSR2(ID), MFR1(INT(ID), MSR2(INT(ID), MFR2(INT(ID) AS FOLLOWS
137 MODIFICATION INSTALLED ON A/C NO. *** AT ***** FLIGHT HOURS
138 1ST CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
139 2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
140 SLOW CRACK GROWTH RATE1 = .***** INCHES/HOUR
141 FAST CRACK GROWTH RATE1 = .***** INCHES/HOUR
142 SLOW CRACK GROWTH RATE2 = .***** INCHES/HOUR
143 FAST CRACK GROWTH RATE2 = .***** INCHES/HOUR
144 INTERNAL SLOW CRACK GROWTH RATE1 = .***** INCHES/HOUR
145 INTERNAL FAST CRACK GROWTH RATE1 = .***** INCHES/HOUR
146 INTERNAL SLOW CRACK GROWTH RATE2 = .***** INCHES/HOUR
147 INTERNAL FAST CRACK GROWTH RATE2 = .***** INCHES/HOUR
148 LEAVE
149 ELSE
150 LOOP
151 ALWAYS
152 RETURN
153 END

```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE INSTALL-MODIFICATION

AC	GLOBAL VARIABLE	(1-D)	INTEGER	2 REFS
ACTUAL-AVG-FAT-LIFE	GLOBAL VARIABLE	REAL	1 REFS	
AF	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
AFACT	GLOBAL VARIABLE	(1-D) REAL	4 REFS	
AIRPLANE	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS	
AREFL	GLOBAL VARIABLE	(1-D) INTEGER	3 REFS	
AIE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
AISR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
AZE	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS	
AZSR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS	
BIRTH-DEFECT--ORABI	GLOBAL VARIABLE	INTEGER	2 REFS	
CANCEL-SCHEDULED-INS	PROCEDURE	REAL	1 REFS	
CGRI	GLOBAL VARIABLE	INTEGER	1 REFS	
CRREP-TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
CL	LOCAL RECURSIVE	(1-D) REAL	2 REFS	
CLGT	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
COMP-RISK	PROCEDURE	INTEGER	1 REFS	
COREP-TIME	GLOBAL VARIABLE	(1-D) REAL	5 REFS	
COROSION	GLOBAL VARIABLE	(1-D) ALPHA	6 REFS	
CO-EXISTS	EVENT NOTICE	INTEGER	1 REFS	
CRCTR	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS	
DEFECT-LIFE	LOCAL RECURSIVE	INTEGER	1 REFS	
ENTRY-TIME	TEMPORARY ATTRIBUTE	REAL	3 REFS	
FAILURE	EVENT NOTICE	REAL	3 REFS	
FATIGUE-LIFE-SCATTER	PROCEDURE	INTEGER	1 REFS	
FIRST-LIFE	LOCAL RECURSIVE	REAL	6 REFS	
FSH	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS	
HOURS-TO-CORROSION	LOCAL RECURSIVE	REAL	3 REFS	
HZ	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
HZRD	GLOBAL VARIABLE	(1-D) REAL	1 REFS	
I	LOCAL RECURSIVE	REAL	2 REFS	
ID	GLOBAL VARIABLE	INTEGER	74 REFS	
IE1	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS	
IE2	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS	
INSP-SCH	PROCEDURE	ALPHA	1 REFS	
INSTALL-MODIFICATION	PROCEDURE	INTEGER	1 REFS	
INT-FAST1	LOCAL RECURSIVE	REAL	2 REFS	
INT-FAST2	LOCAL RECURSIVE	REAL	2 REFS	
INT-M1	GLOBAL VARIABLE	REAL	2 REFS	
INT-M2	GLOBAL VARIABLE	REAL	2 REFS	
INT-M3	GLOBAL VARIABLE	REAL	2 REFS	
INT-M4	GLOBAL VARIABLE	REAL	2 REFS	
INT-SLOW1	LOCAL RECURSIVE	REAL	2 REFS	
INT-SLOW2	LOCAL RECURSIVE	REAL	2 REFS	
JFLAG	GLOBAL VARIABLE	ALPHA	1 REFS	
LDX	GLOBAL VARIABLE	INTEGER	2 REFS	
LOG-E-F	PROCEDURE	REAL	1 REFS	
LTHD	GLOBAL VARIABLE	ALPHA	1 REFS	
MFRI	GLOBAL VARIABLE	(1-D) REAL	2 REFS	

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2
 MFR1.INT GLOBAL VARIABLE (1-D) REAL 2 REFS
 MFR2.INT GLOBAL VARIABLE (1-D) REAL 2 REFS
 MSR1 INT GLOBAL VARIABLE (1-D) REAL 2 REFS
 MSR2 INT GLOBAL VARIABLE (1-D) REAL 2 REFS
 M1.MEAN GLOBAL VARIABLE (1-D) REAL 2 REFS
 M2.MEAN GLOBAL VARIABLE REAL 2 REFS
 M3.MEAN GLOBAL VARIABLE REAL 2 REFS
 M4.MEAN GLOBAL VARIABLE REAL 2 REFS
 NCZ GLOBAL VARIABLE REAL 2 REFS
 NOAC GLOBAL VARIABLE INTEGER 1 REFS
 NSMD GLOBAL VARIABLE INTEGER 2 REFS
 OCCUR.MOD GLOBAL VARIABLE REAL 2 REFS
 OPD GLOBAL VARIABLE INTEGER 2 REFS
 PFID GLOBAL VARIABLE INTEGER 1 REFS
 PFTIM GLOBAL VARIABLE INTEGER 1 REFS
 PREDICT.CORROSION PROCEDURE PFAL 3 REFS
 RANDOM.F PROCEDURE PFAL 3 REFS
 RATE EVENT NOTICE 3 REFS
 REACH.FAIL.SAFE.LGT LOCAL RECURSIVE < 15 RFAL 2 REFS
 RN LOCAL RECURSIVE < 12 REAL 9 REFS
 RST LOCAL RECURSIVE < 13 REAL 4 REFS
 SD.SCH SECOND.LIFE (1-D) ALPHA 2 REFS
 STD.FAST: LOCAL RECURSIVE < 3 PFAL 4 REFS
 STD.FAST2 LOCAL RECURSIVE < 5 REAL 2 REFS
 STD.SLOW1 LOCAL RECURSIVE < 7 REAL 2 REFS
 STD.SLOW2 LOCAL RECURSIVE < 4 REAL 2 REFS
 TIME.V SYSTEM ATTRIBUTE REAL 9 REFS
 TLID GLOBAL VARIABLE (2-D) INTEGER 1 REFS
 USAGE.LIFE GLOBAL VARIABLE REAL 1 REFS
 1.CR.EXISTS GLOBAL VARIABLE (1-D) ALPHA 5 REFS
 1.ITE EVENT NOTICE 3 REFS
 1.STRENGTH.REDUCTION GLOBAL VARIABLE (1-D) ALPHA 5 REFS
 2.CR.EXISTS GLOBAL VARIABLE (1-D) ALPHA 4 REFS
 2.ITE EVENT NOTICE 3 REFS
 2.STRENGTH.REDUCTION EVENT NOTICE 5 REFS

```

1  EVENT IN-SERVICE-DAMAGE(IDSDM)
2  **
3  ** REPRESENTS THE OCCURRENCE OF A SERVICE DAMAGE DEFECT? RESULTS IN IMMEDIATE
4  ** INITIATION OF NEXT SCHEDULED CRACK
5  **
6  DEFINE IDSDM AS AN INTEGER VARIABLE
7  LET ID = IDSDM
8  IF LTHO = "NO"
9  PRINT 1 LINE WITH ID AS FOLLOWS
10 SERVICE DAMAGE AIRCRAFT NO. ***
11 ALWAYS
12 IF LTHO = "YES"
13 FOR I = 1 TO NOAC(LDX)
14 DO
15 IF ID = TLID(LDX,I)
16 SKIP 1 OUTPUT LINE
17 PRINT 2 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
18 A/C NO. *** EXPERIENCES SERVICE DAMAGE AT ***** FLIGHT HOURS
19 CAUSES IMMEDIATE INITIATION OF NEXT SCHEDULED CRACK
20 LEAVE
21 ELSE
22 LOOP
23 ALWAYS
24 LET OSDM = OSDM + 1
25 LET GOSDM = GOSDM + 1
26 LET ISDT = TIME.V - LAST.SD(ID)
27 LET GISD = TIME.V - LAST.SD(ID)
28 CALL PREDICT-SERVICE-DAMAGE YIELDING OUMS.TO-SERVICE-DAMAGE
29 LET PST = USAGE-LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
30 LET SD.SCH(ID) = "NO"
31 IF OUMS.TO-SERVICE-DAMAGE LT PST
32 SCHEDULE AN IN-SERVICE-DAMAGE(ID) AT TIME.V + OUMS.TO-SERVICE-DAMAGE
33 LET LAST.SD(ID) = TIME.V
34 LET SD.SCH(ID) = "YES"
35 REGARDLESS
36 **
37 ** TEST FOR SCHEDULED FIRST CRACK INITIATION
38 **
39 LET 1-STRENGTH-REDUCTION = AISR(ID)
40 LET 2-STRENGTH-REDUCTION = A2SR(ID)
41 IF 1-CR-EXISTS(ID) = "NS"
42 CANCEL THE 1-STRENGTH-REDUCTION
43 RESCHEDULE THE 1-STRENGTH-REDUCTION(ID) NOW
44 IF SD.SCH(ID) = "NO" AND TIME.A(2-STRENGTH-REDUCTION) GE PST + TIME.V
45 AND CO-EXISTS(ID) NE "NS"
46 CANCEL THE 2-STRENGTH-REDUCTION
47 DESTROY THE 2-STRENGTH-REDUCTION
48 LET 2-CR-EXISTS(ID) = "NN"
49 ALWAYS
50 RETURN
51 OTHERWISE
52 **
53 ** TEST FOR SCHEDULED SECOND CRACK INITIATION
54 **

```

164

CDC 6600 CACI SIMSCRIPT II-5 VERSION /4.0-00/ KRONOS 2.1.2 03/02/78. 18.36.02. PAGE 73

```
51 00
52 IF 2.CR.EXISTS(ID1) & "MS"
53 CANCEL THE 2.STRENGTH.REDUCTION
54 RESCHEDULE THE 2.STRENGTH.REDUCTION(ID) NOW
55 RETURN
56 OTHERWISE
57 RETURN
58 END
```

SYMBOLIC REFERENCE MAP (2 = 1) - EVENT IN SERVICE-DAMAGE

AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
AISP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
A2SR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
CO-EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	REAL	2 REFS
GCSDM	GLOBAL VARIABLE	INTEGER	2 REFS
GLSD	GLOBAL VARIABLE	REAL	1 REFS
I	LOCAL RECURSIVE	REAL	2 REFS
ID	GLOBAL VARIABLE	INTEGER	21 REFS
IDSDM	GIVEN ARGUMENT	INTEGER	
	TEMPORARY ATTRIBUTE	INTEGER	
IN-SERVICE-DAMAGE	EVENT NOTICE		3 REFS
LAST.SD	GLOBAL VARIABLE	(1-D) REAL	2 REFS
LDX	GLOBAL VARIABLE	INTEGER	3 REFS
LTHO	GLOBAL VARIABLE	ALPHA	2 REFS
NOAC	GLOBAL VARIABLE	ALPHA	2 REFS
OSDM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
OURS.TD-SERVICE-DAMA	GLOBAL VARIABLE	INTEGER	2 REFS
PREDICT-SERVICE-DAM	LOCAL RECURSIVE	REAL	3 REFS
RST	PROCEDURE	INTEGER	1 REFS
SD-SCH	LOCAL RECURSIVE	REAL	3 REFS
TIME-A	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS
TIME-V	TEMPORARY ATTRIBUTE	REAL	1 REFS
TLID	SYSTEM ATTRIBUTE	REAL	7 REFS
USAGE.LIFE	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
ISDT	GLOBAL VARIABLE	REAL	1 REFS
1-CR-EXISTS	GLOBAL VARIABLE	REAL	1 REFS
1-STRENGTH-REDUCTION	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2-CR-EXISTS	EVENT NOTICE		3 REFS
2-STRENGTH-REDUCTION	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
	EVENT NOTICE		6 REFS

1-61

```

1  EVENT T.IMPLEMENT.MCO
2
3  ** REPRESENTS DEVELOPMENT OF MODIFICATION BECAUSE OF FATIGUE TEST FAILURE
4  **
5  LET MOD.NO = MOD.NO + 1
6  LET NMD = NMD + 1
7  LET FOCK = IDCK
8  LET BEEN.MODIFIED = 0
9  LET IMOD = IDCK
10 LET COST.OF.REPAIRS = 0.0
11 LET AIRFRAME.TIME = 0.0
12 IF 2.0*ACTUAL.AVG.FAT.LIFE GT PREDICTED.LIFE
13 LET NEW.LIFE = 2.0*ACTUAL.AVG.FAT.LIFE
14 JUMP AHEAD
15 OTHERWISE
16 LET NEW.LIFE = PREDICTED.LIFE
17 HERE
18 LET NMU = MU.R + ((1.0-MU.R)*.15)
19 LET NSIG = SIG.P * .85
20 CALL REAL.LIFE(NMU,NSIG,NEW.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE
21 IF MOD.TESTED = "YES" AND ACTUAL.AVG.FAT.LIFE LT NEW.LIFE
22 LET ACTUAL.AVG.FAT.LIFE = NEW.LIFE
23 ALWAYS
24 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
25 DO
26 LET ID = TAIL.ID
27 IF TEST.LIFE LT USAGE.LIFE
28 LET RETRO = "YES"
29 LET TO.RE.MODIFIED = IDCK
30 LET MOD.PENDING(ID) = "YES"
31 IF ENTRY.TIME(AIRPLANE(ID)) + C7*TEST.LIFE GT TIME.V
32 SCHEDULE A T.INSPECTION.INCREASE(ID) AT ENTRY.TIME(AIRPLANE(ID)) + C7 *
33 TEST.LIFE
34 LET ATIII(ID) = T.INSPECTION.INCREASE
35 ALWAYS
36 REGARDLESS
37 LOOP
38 RETURN
39 END

```

SYMBOLIC REFERENCE MAP (P = 1) - EVENT T.IMPLEMENT.MOD

ACTIVE.FLEET	SET	1 REFS
ACTUAL.AVG.FAT.LIFE	GLOBAL VARIABLE	5 REFS
AIRCRAFT	TEMPORARY ENTITY	1 REFS
AIRFRAME.TIME	GLOBAL VARIABLE	1 REFS
AIRPLANE	GLOBAL VARIABLE	2 REFS
ATII	GLOBAL VARIABLE	1 REFS
BEEN.MODIFIED	GLOBAL VARIABLE	1 REFS
COST.OF.REPAIRS	GLOBAL VARIABLE	1 REFS
CT	GLOBAL VARIABLE	2 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	2 REFS
FDCK	GLOBAL VARIABLE	1 REFS
ID	GLOBAL VARIABLE	4 REFS
IDCK	GLOBAL VARIABLE	3 REFS
IMOD	GLOBAL VARIABLE	1 REFS
MOD.NO	GLOBAL VARIABLE	2 REFS
MOD.TESTED	GLOBAL VARIABLE	1 REFS
MU.P	GLOBAL VARIABLE	2 REFS
NEW.LIFE	LOCAL RECURSIVE	5 REFS
NMD	GLOBAL VARIABLE	2 REFS
NSIG	LOCAL RECURSIVE	2 REFS
PREDICTED.LIFE	LOCAL RECURSIVE	2 REFS
REAL.LIFE	GLOBAL VARIABLE	2 REFS
RETRO	PROCEDURE	1 REFS
SIG.R	GLOBAL VARIABLE	1 REFS
TAIL.ID	TEMPORARY ATTRIBUTE	1 REFS
TEST.LIFE	GLOBAL VARIABLE	3 REFS
TIME.V	SYSTEM ATTRIBUTE	1 REFS
IMOD.PENDING	GLOBAL VARIABLE	1 REFS
TO.BE.MODIFIED	GLOBAL VARIABLE	1 REFS
T.IMPLEMENT.MOD	EVENT NOTICE	1 REFS
T.INSPECTION.INCREAS	EVENT NOTICE	2 REFS
USAGE.LIFE	GLOBAL VARIABLE	1 REFS

CDC #600 CADI SIMSCRIPT II-5 VERSION /4.0-00/ KRONOS 2.1.2 03/02/78. 18.36.02. PAGE 77

```
1 ROUTINE PREEDICT.SERVICE.DAMAGE YIELDING OURS.TO.SERVICE.DAMAGE
2 **
3 ** GENERATES TIME TO SERVICE DAMAGE OCCURRENCE FOR A GIVEN AIRCRAFT FROM A
4 ** CONSTANT SERVICE DAMAGE OCCURRENCE RATE
5 **
6 IF SDM.OCCURRENCE.RATE LE 0.0
7   LET OURS.TO.SERVICE.DAMAGE = 2.0 * USAGE.LIFE
8   RETURN
9 ELSE
10  LET RN = RANDOM.F(2)
11  LET OURS.TO.SERVICE.DAMAGE = -LOG.E.F(RN) / SDM.OCCURRENCE.RATE
12  RETURN
13 FWD
```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE PREEDICT.SERVICE.GAM

LOG.E.F	PROCEDURE	REAL	1 REFS
OURS.TO.SERVICE.DAMA	YIELDED ARGUMENT	REAL	3 REFS
PREEDICT.SERVICE.DAM	PROCEDURE	INTEGER	1 REFS
RANDOM.F	PROCEDURE	REAL	1 REFS
RN	LOCAL RECURSIVE	REAL	2 REFS
SDM.OCCURRENCE.KATE	GLOBAL VARIABLE	REAL	2 REFS
USAGE.LIFE	GLOBAL VARIABLE	REAL	1 REFS

```

1 ROUTINE PREDICT.CORROSION YIELDING HOURS.TO.CORROSION
2 **
3 ** GENERATES TIME TO CORROSION INITIATION FROM ELEMENT HAZARD RATE WHICH IS
4 ** APPROXIMATED BY TWO CONSTANT OCCURRENCE RATES
5 **
6 ** 1.CDM.OCCURRENCE.RATE = FIRST CONSTANT OCCURRENCE RATE
7 ** 2.CDM.OCCURRENCE.RATE = SECOND CONSTANT OCCURRENCE RATE
8 ** CDM.RATE.CHANGE = TIME ON AIRFRAME AT WHICH SECOND RATE IS USED
9 **
10 LET RN = RANDOM.F(4)
11 LET CRCT = CDM.RATE.CHANGE - TIME.V + ENTHY.TIME(AIRPLANE(ID))
12 IF CRCT LE 0.0
13 IF 2.CDM.OCCURRENCE.RATE LE 0.0
14 LET HOURS.TO.CORROSION = 2.0 * USAGE.LIFE
15 RETURN
16 ELSE
17 LET HOURS.TO.CORROSION = LOG.E.F(RN) / (-2.CDM.OCCURRENCE.RATE)
18 RETURN
19 OTHERWISE
20 **
21 IF RN GE EXP.F(-CRCT * 1.CDM.OCCURRENCE.RATE)
22 IF 1.CDM.OCCURRENCE.RATE LE 0.0
23 LET HOURS.TO.CORROSION = 2.0 * USAGE.LIFE
24 RETURN
25 ELSE
26 LET HOURS.TO.CORROSION = LOG.E.F(RN) / (-1.CDM.OCCURRENCE.RATE)
27 RETURN
28 OTHERWISE
29 IF 2.CDM.OCCURRENCE.RATE LE 0.0
30 LET HOURS.TO.CORROSION = 2.0 * USAGE.LIFE
31 RETURN
32 ELSE
33 LET LD = (2.CDM.OCCURRENCE.RATE - 1.CDM.OCCURRENCE.RATE) * CRCT
34 LET HOURS.TO.CORROSION = LOG.F.F(RN/EXP.F(LD)) / (-2.CDM.OCCURRENCE.RATE)
35 RETURN
36 END

```

- 71 -

SYMBOLIC REFERENCE MAP (P = 1) - ROUTINE PREDICT.CORROSION

AIRPLANE	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
CDM-RATE.CHANGE	GLOBAL VARIABLE		REAL	1 REFS
CRCI	LOCAL RECURSIVE	< 3	REAL	4 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE		REAL	1 REFS
EXP.F	PROCEDURE		REAL	2 REFS
HOURS.TO.CORROSION	YIELDED ARGUMENT	< 1	REAL	7 REFS
ID	GLOBAL VARIABLE		INTEGER	1 REFS
LD	LOCAL RECURSIVE	< 4	REAL	2 REFS
LOG.E.F	PROCEDURE		REAL	3 REFS
PREDICT.CORROSION	PROCEDURE		INTEGER	1 REFS
RANDOM.F	PROCEDURE		REAL	1 REFS
RN	LOCAL RECURSIVE	< 2	REAL	5 REFS
TIME.V	SYSTEM ATTRIBUTE		REAL	1 REFS
USAGE.LIFE	GLOBAL VARIABLE		REAL	3 REFS
1.CDM.OCCURRENCE.RAT	GLOBAL VARIABLE		REAL	4 REFS
2.CDM.OCCURRENCE.RAT	GLOBAL VARIABLE		REAL	5 REFS

```

1  EVENT CORROSION(IDCO) SAVING THE EVENT NOTICE
2  **
3  ** REPRESENTS THE INITIATION OF CORROSION? #F# REMAINING TIME TO CRACK INITIATION OF
4  ** ALL SCHEDULED CRACKS IS REDUCED BY CDM* MULTIPLYING FACTOR
5  **
6  DEFINE C-STR AS AN ALPHA VARIABLE
7  DEFINE IDCO AS AN INTEGER VARIABLE
8  LET ID = IDCO
9  IF I.CR.EXISTS(ID) = "YES"
10 (CALL COMP.RISK YIELDING CL
11 ALWAYS
12 IF LTHO = "NO"
13 PRINT 1 LINE WITH ID AS FOLLOWS
14 CORROSION AIRCRAFT NO. ***
15 ALWAYS
16 LET RST = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
17 LET ICOR = TIME.V - CORP.TIME(ID)
18 LET GICOR = TIME.V - CORP.TIME(ID)
19 LET OCOR = OCOR + 1
20 LET GOCOR = GOCOR + 1
21 LET C.INT(ID) = "NO"
22 IF RANDOM.F(10) LE C.PPOB
23 LET C.INT(ID) = "YES"
24 REGARDLESS
25 LET CO.EXISTS(ID) = "YES"
26 IF INSP.SCH(ID) = "NO"
27 CALL INSPECTION.SCHEDULEM(2)
28 REGARDLESS
29 LET CDM.MULTIPLYING.FACTOR = 1.0 - C29
30 IF RANDOM.F(1) LE LOCATED.IN.STRESS.CON
31 LET CDM.MULTIPLYING.FACTOR = 1.0 - C24
32 LET C-STR = "YES"
33 REGARDLESS
34 LET CGRI(ID) = CHRF
35 IF LTHO = "YES"
36 FOR I = 1 TO NOAC(LDX)
37 DO
38 IF ID = TLID(LDX,I)
39 LET LIST = 1.0
40 SKIP 1 OUTPUT LINE
41 PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
42 A/C NO. *** EXPERIENCES CORROSION INITIATION AT ***** HOURS
43 SKIP 1 OUTPUT LINE
44 PRINT 1 LINE WITH CGRI(ID) AS FOLLOWS
45 CRACK GROWTH RATE CORROSION MULTIPLYING FACTOR = ***
46 IF C.INT(ID) = "YES"
47 PRINT 1 LINE AS FOLLOWS
48 CORROSION IS INTERNAL
49 JUMP AHEAD
50 OTHERWISE
51 PRINT 1 LINE AS FOLLOWS
52 CORROSION IS EXTERNAL

```

-173-

```

49      HERE
50      IF C-STR = "YES"
51      PRINT 1 LINE AS FOLLOWS
52      COPROSION IN STRESS CONCENTRATION
53      JUMP AHEAD
54      OTHERWISE
55      PRINT 1 LINE AS FOLLOWS
56      COPROSION NOT IN STRESS CONCENTRATION
57      HERE
58      LEAVE
59      ELSE
60      LOOP
61      ALWAYS
62      IF IE1(ID) = "YES"
63      LET 1.ITE = AIE(ID)
64      LET TRT = (TIME-A(1.ITE)-TIME.V) / CGP1(ID)
65      CANCEL THE 1.ITE
66      IF TRT LT NST
67      RESCHEDULE THE 1.ITE(ID) AT TIME.V + TRT
68      JUMP AHEAD
69      ELSE
70      DESTROY THE 1.ITE
71      LET IE1(ID) = "NO"
72      HERE
73      IF IE2(ID) = "YES"
74      LET 2.ITE = AIE(ID)
75      LET TRT = (TIME-A(2.ITE)-TIME.V) / CGP2(ID)
76      CANCEL THE 2.ITE
77      IF TRT LT NST
78      RESCHEDULE THE 2.ITE(ID) AT TIME.V + TRT
79      JUMP AHEAD
80      ELSE
81      DESTROY THE 2.ITE
82      LET IE2(ID) = "NO"
83      HERE
84      ALWAYS
85      IF ALL(ID) = "YES"
86      LET REACH.FAIL.SAFF.LST = AMPL(ID)
87      LET TRT = (TIME-A(NEACH.FAIL.SAFF.LST) - TIME.V) / CGP1(ID)
88      CANCEL THE REACH.FAIL.SAFF.LST
89      IF TRT LT NST
90      RESCHEDULE THE REACH.FAIL.SAFF.LST(ID) AT TIME.V + TRT
91      JUMP AHEAD
92      ELSE
93      DESTROY THE REACH.FAIL.SAFF.LST
94      LET ALL(ID) = "NO"
95      HERE
96      REGARDLESS
97      IF FSM(ID) = "YES"
98      LET FAILURE = AF(ID)
99      LET MFTN = TIME-A(FAILURE) - TIME.V
100     IF LIST = 1.0

```

```

100 PRINT 1 LINE WITH TIME.V * NFM * CDM * MULTIPLYING.FACTOR -
101 ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
102 ELEMENT FAILURE PROJECTED AT ***** FLIGHT HOURS
103 ALWAYS
104 CANCEL THE FAILURE
105 IF NFM LT RST
106 RESCHEDULE THE FAILURE(ID) AT TIME.V * NFM / COR1(ID)
107 JUMP AHEAD
108 ELSE
109 DESTROY THE FAILURE
110 LET FSH(ID) = "NO"
111 HERE
112 REGARDLESS
113
114 TEST FOR SCHEDULED FIRST CRACK INITIATION
115
116 IF 1.CR.EXISTS(ID) = "NS"
117 LET 1.STRENGTH.REDUCTION = AISR(ID)
118 LET REMAINING.LIFE = TIME.A(1.STRENGTH.REDUCTION) - TIME.V
119 LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * CDM * MULTIPLYING.FACTOR
120 IF LIST = 1.0
121 PRINT 1 LINE WITH TIME.V * REDUCED.REMAINING.LIFE - ENTRY.TIME(AIRPLANE(ID))
122 AS FOLLOWS
123 1ST CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
124 ALWAYS
125 CANCEL THE 1.STRENGTH.REDUCTION
126 DESTROY THE 1.STRENGTH.REDUCTION
127 LET 1.CR.EXISTS(ID) = "NN"
128 IF REDUCED.REMAINING.LIFE LT RST OR SD.SCH(ID) = "YES"
129 SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V * REDUCED.REMAINING.LIFE
130 LET AISR(ID) = 1.STRENGTH.REDUCTION
131 LET 1.CR.EXISTS(ID) = "NS"
132 ALWAYS
133 REGARDLESS
134
135 TEST FOR SCHEDULED SECOND CRACK INITIATION
136
137 IF 2.CR.EXISTS(ID) = "NS"
138 LET 2.STRENGTH.REDUCTION = A2SR(ID)
139 LET REMAINING.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
140 LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * CDM * MULTIPLYING.FACTOR
141 IF LIST = 1.0
142 PRINT 1 LINE WITH TIME.V * REDUCED.REMAINING.LIFE - ENTRY.TIME(AIRPLANE(ID))
143 AS FOLLOWS
144 2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
145 ALWAYS
146 CANCEL THE 2.STRENGTH.REDUCTION
147 DESTROY THE 2.STRENGTH.REDUCTION
148 LET 2.CR.EXISTS(ID) = "NN"
149 IF REDUCED.REMAINING.LIFE LT RST OR SD.SCH(ID) = "YES"
150 SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V * REDUCED.REMAINING.LIFE
151 LET A2SR(ID) = 2.STRENGTH.REDUCTION
152 LET 2.CR.EXISTS(ID) = "NS"

```

- 175 -

```

150 ALWAYS
151 REGARDLESS
152 IF I-CR-EXISTS(ID) = "YES"
153 LET SRRTF(ID) = SRRTF(ID)+CGRF(ID)
154 LET TCHMS(ID) = TIME,V
155 ALWAYS
156 RETURN
157 END

```


03/02/70. 18.36.02.

KRONOS 2.1.2

/4.0-00/

CACI SIMSCRIPT-II-5 VERSION

CDC 6600

TRCHG	GLOBAL VARIABLE	(1-D) REAL	1 REFS
TRY	LOCAL RECURSIVE	REAL	9 REFS
USAGE.LIFE	GLOBAL VARIABLE	REAL	1 REFS
ICORT	GLOBAL VARIABLE	REAL	1 REFS
1-CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	5 REFS
1-ITE	EVENT NOTICE		5 REFS
1-STRENGTH.REDUCTION	EVENT NOTICE	(1-D) ALPHA	6 REFS
2-CR.EXISTS	GLOBAL VARIABLE		3 REFS
2-ITE	EVENT NOTICE		5 REFS
2-STRENGTH.REDUCTION	EVENT NOTICE		4 REFS

```

1 ROUTINE RATE (M,S,RN)
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

    GENERATES ELEMENT CRACK PROPAGATION RATES REFLECTING VARIATION IN MATERIAL
    PROPERTIES AND LOAD ENVIRONMENT VARIATION

    DISTRIBUTION OF CRACK PROPAGATION RATES IS REPRESENTED AS NORMAL
    M = MEAN PROPAGATION RATE
    S = STANDARD DEVIATION
    RN = RANDOM NUMBER

    METHOD BASED ON APPROXIMATIONS IN C. HASTINGS. APPROXIMATIONS FOR DIGITAL
    COMPUTERS

    IF PROCEDURE YIELDS NEGATIVE RATE, RATE IS SET EQUAL TO MEAN MINUS FOUR
    STANDARD DEVIATIONS

    LET MNI = RN
    IF RN GT 0.5
    LET RN = 1.0 - RN
    REGARDLESS
    LET W = SORT.F(LOG.E.F(1.0/RN**2))
    LET G1 = 2.515517 + .802853*M + .010396*W**2
    LET G2 = 1.0 + 1.432788*M + .189259*W**2 + .001308*W**3
    LET Z = W - G1/G2
    IF MNI GT 0.5
    LET Z = -Z
    REGARDLESS
    LET Z = Z * S * M
    IF Z LE 0
    LET Z = M - (4 * S)
    REGARDLESS
    RETURN WITH Z
    END

```

SYMBOLIC REFERENCE MAP (R = I) - ROUTINE RATE

61	LOCAL RECURSIVE	< 6	REAL	2 REFS
62	LOCAL RECURSIVE	< 7	REAL	2 REFS
LOG.E.F	PROCEDURE	< 1	REAL	1 REFS
M	GIVEN ARGUMENT	< 1	REAL	3 REFS
RATE	PROCEDURE	< 3	REAL	1 REFS
RNI	GIVEN ARGUMENT	< 4	REAL	4 REFS
S	LOCAL RECURSIVE	< 2	REAL	2 REFS
SORT.F	GIVEN ARGUMENT	< 5	REAL	3 REFS
W	PROCEDURE	< 5	REAL	1 REFS
Z	LOCAL RECURSIVE	< 8	REAL	7 REFS
				8 REFS

EVENT 1: STRENGTH REDUCTION (IDISR) SAYING THE EVENT NOTICE

```

1  **
2  ** REPRESENTS FIRST CRACK INITIATION
3
4  DEFINE IDISR AS AN INTEGER VARIABLE
5  LET ID = IDISR
6  LET SDIF(ID) = SU-SF
7  LET PLGT(ID) = FSAF.LGT
8  LET A = AMEAN * AFACI(ID)
9
10 LET TAR = ENTRY.TIME(AIRPLANE(ID)) + USAGE.LIFE
11 IF LTWO = "YES"
12   FOR I = 1 TO NOAC(LDX)
13   DO
14     IF ID = TLID(LDX,I)
15     LET LIST = 1.0
16     SKIP 1 OUTPUT LINE
17     PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
18     A/C NO. *** EXPERIENCES 1ST CRACK INITIATION AT ***** HOURS
19     LEAVE
20   ELSE
21     LOOP
22   ALWAYS
23   LET LCRTK = TIME.V - CKREP.TIME(ID)
24   LET GICRK = TIME.V - CKREP.TIME(ID)
25   LET OICR = OICR + 1
26   LET GOICR = GOICR + 1
27   LET L.INT(ID) = "NO"
28   IF RANDOM.F(10) LE 1.0*PHOM
29   LET L.INT(ID) = "YES"
30   LET MSRI(ID) = MSRI.INT(ID)
31   LET MFR1(ID) = MFR1.INT(ID)
32   LET MSR2(ID) = MSR2.INT(ID)
33   LET MFR2(ID) = MFR2.INT(ID)
34   LET CONE = INT.CONE
35   LET CTWO = INT.CTWO
36   LET CTMREF = IN.CTMREF
37   IF LIST = 1.0
38   PRINT 1 LINE AS FOLLOWS
39   CRACK INITIATES INTERNALLY
40   ALWAYS
41   LET T = L.EXT/(MSRI(ID)*CGRI(ID))
42   IF L.EXT GT COME
43   LET T = COME / (MSRI(ID) * CGRI(ID))
44   * (L.EXT - COME) / (MFR1(ID) * CGRI(ID))
45   ALWAYS
46   IF L.EXT GT CTWO
47   LET T = COME/(MSRI(ID) * CGRI(ID))
48   * (CTWO - COME)/(MFR1(ID)*CGRI(ID))
49   * (L.EXT-CTWO)/(MSR2(ID)*CGRI(ID))
50   ALWAYS
51   IF L.EXT GT CTMREF
52   LET T = COME / (MSRI(ID)*CGRI(ID))
53   * (CTWO - COME)/(MFR1(ID)*CGRI(ID))

```

```

52 * (CTHREE - CTWG) / (MSR2(ID) * CGRI(ID))
53 * (L.EXT - CTHREE) / (MSR2(ID) * CGRI(ID))
54 ALWAYS
55 IF TIME.V * T LT TAR OR CO.EXISTS(ID) = "NS"
56 SCHEDULE A I.ITE(ID) AT TIME.V + T
57 LET AIE(ID) = I.ITE
58 LET IEI(ID) = "YES"
59 ALWAYS
60 REGARDLESS
61 LET I.CR.EXISTS(ID) = "YES"
62 IF INSP.SCH(ID) = "NO"
63 CALL INSPECTION.SCHEDULEK(1)
64 REGARDLESS
65
66 PREDICT TIME TO FAILURE FROM FIRST CRACK INITIATION
67
68 LET XRN(ID) = RANDOM.F(3)
69 LET LG = LOG.E.F(XRN(ID))
70 LET GR1 = MCR1(ID) * CGRI(ID)
71 LET GR2 = MFR1(ID) * CGRI(ID)
72 LET GR3 = MSR2(ID) * CGRI(ID)
73 LET GR4 = MFR2(ID) * CGRI(ID)
74 LET T1 = COME / GR1
75 LET T2 = T1 * (CTWO - COME) / GR2
76 LET T3 = T2 * (CTHREE - CTWO) / GR3
77 LET T4 = T3 * (FSAF.LGT - CTHREE) / GR4
78 LET R1 = ((SU-SF)*GR1) / FSAF.LGT
79 LET R2 = ((SU-SF)*GR2) / FSAF.LGT
80 LET R3 = ((SU-SF)*GR3) / FSAF.LGT
81 LET R4 = ((SU-SF)*GR4) / FSAF.LGT
82 LET R5 = GR4 / (LGHT.10.FAILURE - FSAF.LGT)*SF
83
84 IF TIME.V * T4 LT TAR OR CO.EXISTS(ID) = "NS"
85 SCHEDULE = OECM.FAIL.SAFE.LGT(ID) AT TIME.V + T4
86 LET AMFSL(ID) = HEACH.FAIL.SAFE.LGT
87 LET AIL(ID) = "YES"
88 REGARDLESS
89 LET R1 = A * EXP.F(SU * R)
90 LET R2 = R * R1
91 LET TTF = -LOG.E.F((R2/R1)*LG + 1.0)/R2
92 IF TTF LE T1
93 JUMP AHEAD
94 ELSE
95 LET S1 = SU - R1 * T1
96 LET R4 = R * R2
97 LET LGK5 = LOG.E.F(R) * (R * S1 + 8 * R2 * T1)
98 LET R4 = (A*EXP.F(R*52))/K4
99 LET R4 = (K1/K2)*(EXP.F(-8*R1*T1))-1.01
100 LET TTF = LGK5 - LOG.E.F(ABS.F(R4)) - LOG.E.F(ABS.F(LG*K8-K9)))/K4
101 IF TTF LE T2
102 JUMP AHEAD
103 ELSE
104 LET S2 = S1 - R2*(T2-T1)

```

```

105 LET K10 = B*R3
106 LET ARG = B*S2*B*R3*T2
107 LET K11 = -LOG.E.F(-K10) + LOG.E.F(A) + ARG
108 LET K12 = A*EXP.F(R*S2)/K10
109 LET K13 = EXP.F(B*R2*T1 - B*R2*T2) - 1.0
110 LET K14 = -LOG.E.F(-(LG*K12-(K8*K13)-K9))
111 LET TTF = (K11+K14)/K10
112 IF TTF LE T3
113 JUMP AHEAD
114 ELSE
115 LET S3 = S2-R3*(T3-T2)
116 LET K15 = B*R4
117 LET K16 = -LOG.E.F(-K15)+LOG.E.F(A) + B*S3 + B*R4*T3
118 LET K17 = A*EXP.F(B*S3)/K15
119 LET K18 = EXP.F(H*R3*T2-H*R3*T3) - 1.0
120 LET K19 = -LOG.E.F(-(LG*K17-(K12*K18)-(K8*K13)-K9))
121 LET TTF = (K16+K19)/K15
122 IF TTF LE T4
123 JUMP AHEAD
124 ELSE
125 LET K20 = B*R5
126 LET K21 = -LOG.E.F(-K20) + LOG.E.F(A) + B*SF + B*R5*T4
127 LET K22 = A*EXP.F(B*SF)/K20
128 LET K23 = EXP.F(H*R4*T3-H*R4*T4) - 1.0
129 LET K24 = -LOG.E.F(-(LG*K22-(K12*K23)-(K12*K18)-(K8*K13)-K9))
130 LET TTF = (K21+K24)/K20
131 HERE
132 IF LIST = 1.0
133 PRINT 1 LINE WITH TIME.V+TTF-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
    ELEMENT FAILURE PROJECTED AT ***** FLIGHT HOURS
    ALWAYS
134 IF TIME.V + TTF LT TAP OR CO-EXISTS(ID) = "NS"
135 SCHEDULE A FAILURE(ID) AT TIME.V + TTF
136 LET AFID) = FAILURE
137 LET FSH(ID) = "YES"
138 ALWAYS
139 LET SRES(ID) = SU
140 LET SRATE(ID) = R1
141 LET TRCHG(ID) = TIME.V
142 LET HZRD(ID) = 1.0
143 RETURN
144 END
145

```

SYMBOLIC REFERENCE MAP (P = 1) - EVENT 1-STREAMING-REDUCTION

A	LOCAL RECURSIVE	< 2	REAL	10 REFS
ABS.F	PROCEDURE		INTEGER	2 REFS
AF	GLOBAL VARIABLE		(1-D) REAL	1 REFS
AFACT	GLOBAL VARIABLE		(1-D) REAL	1 REFS
AIL	GLOBAL VARIABLE		(1-D) ALPHA	1 REFS
AIRPLANE	GLOBAL VARIABLE		(1-D) INTEGER	3 REFS
AREAN	GLOBAL VARIABLE		REAL	1 REFS
ARFSL	GLOBAL VARIABLE		(1-D) INTEGER	1 REFS
ASB	LOCAL RECURSIVE	< 42	REAL	2 REFS
AIE	GLOBAL VARIABLE		(1-D) REAL	1 REFS
B	GLOBAL VARIABLE		REAL	25 REFS
CARI	GLOBAL VARIABLE		(1-D) REAL	14 REFS
CKREF.TIME	GLOBAL VARIABLE		(1-D) REAL	2 REFS
CONF	GLOBAL VARIABLE		REAL	10 REFS
CO-EXISTS	GLOBAL VARIABLE		(1-D) ALPHA	3 REFS
CTHREE	GLOBAL VARIABLE		REAL	6 REFS
CTWO	GLOBAL VARIABLE		REAL	8 REFS
ENTRL.TIME	TEMPORARY ATTRIBUTE		REAL	3 REFS
EXP.F	PROCEDURE		REAL	9 REFS
FAILURE	EVENT NOTICE		REAL	2 REFS
FSAF.L67	GLOBAL VARIABLE		REAL	7 REFS
FSH	GLOBAL VARIABLE		(1-D) ALPHA	1 REFS
GOICR	GLOBAL VARIABLE		INTEGER	2 REFS
601CR	LOCAL RECURSIVE	< 19	REAL	3 REFS
601	LOCAL RECURSIVE	< 20	REAL	3 REFS
602	LOCAL RECURSIVE	< 21	REAL	3 REFS
603	LOCAL RECURSIVE	< 22	REAL	4 REFS
604	LOCAL RECURSIVE		REAL	1 REFS
61CRK	GLOBAL VARIABLE		REAL	1 REFS
61CRD	GLOBAL VARIABLE	< 6	REAL	2 REFS
I	GLOBAL VARIABLE		REAL	49 REFS
ID	GIVEN ARGUMENT	< 1	INTEGER	3 REFS
IDISR	TEMPORARY ATTRIBUTE		INTEGER	1 REFS
IEI	GLOBAL VARIABLE		(1-D) ALPHA	1 REFS
INSPECTION.SCHEDULER	PROCEDURE		INTEGER	1 REFS
INSP.SCH	GLOBAL VARIABLE		(1-D) ALPHA	1 REFS
INT.COME	GLOBAL VARIABLE		REAL	1 REFS
INT.CTWO	GLOBAL VARIABLE		REAL	1 REFS
IN.CTHREE	GLOBAL VARIABLE		REAL	1 REFS
K1	LOCAL RECURSIVE	< 32	REAL	3 REFS
K10	LOCAL RECURSIVE	< 41	REAL	4 REFS
K11	LOCAL RECURSIVE	< 43	REAL	2 REFS
K12	LOCAL RECURSIVE	< 44	REAL	4 REFS
K13	LOCAL RECURSIVE	< 45	REAL	4 REFS
K14	LOCAL RECURSIVE	< 46	REAL	2 REFS
K15	LOCAL RECURSIVE	< 48	REAL	4 REFS
K16	LOCAL RECURSIVE	< 49	REAL	2 REFS
F17	LOCAL RECURSIVE	< 50	REAL	3 REFS
K18	LOCAL RECURSIVE	< 51	REAL	3 REFS
K19	LOCAL RECURSIVE	< 52	REAL	2 REFS
K2	LOCAL RECURSIVE	< 33	REAL	4 REFS

```

M20 LOCAL RECURSIVE < 53 REAL 4 REFS
M21 LOCAL RECURSIVE < 54 REAL 2 REFS
M22 LOCAL RECURSIVE < 55 REAL 2 REFS
M23 LOCAL RECURSIVE < 56 REAL 2 REFS
M24 LOCAL RECURSIVE < 57 REAL 2 REFS
M4 LOCAL RECURSIVE < 36 REAL 4 REFS
M8 LOCAL RECURSIVE < 38 REAL 5 REFS
K9 LOCAL RECURSIVE < 39 REAL 5 REFS
LDX GLOBAL VARIABLE INTEGER 7 REFS
L6 GLOBAL RECURSIVE < 18 REAL 4 REFS
LGHT.TO.FAILURE REAL 1 REFS
LGK5 GLOBAL VARIABLE REAL 1 REFS
LIST LOCAL RECURSIVE < 37 REAL 2 REFS
LOG.E.F LOCAL RECURSIVE < 5 REAL 3 REFS
LTHO PROCEDURE REAL 14 REFS
LTEXT GLOBAL VARIABLE ALPHA 1 REFS
MFR1 GLOBAL VARIABLE REAL 7 REFS
MFR2 GLOBAL VARIABLE ( 1-D) REAL 5 REFS
MFR2-INT GLOBAL VARIABLE ( 1-D) REAL 1 REFS
MSR1 GLOBAL VARIABLE ( 1-D) REAL 3 REFS
MSR1-INT GLOBAL VARIABLE ( 1-D) REAL 1 REFS
MSR2 GLOBAL VARIABLE ( 1-D) REAL 6 REFS
MSR2-INT GLOBAL VARIABLE ( 1-D) REAL 1 REFS
MOAC GLOBAL VARIABLE ( 1-D) REAL 4 REFS
OICR GLOBAL VARIABLE ( 1-D) REAL 1 REFS
RANDOM.F GLOBAL VARIABLE INTEGER 1 REFS
REACH.FAIL.SAFE.I67 REAL 2 REFS
PLGT EVENT NOTICE 2 REFS
R1 GLOBAL VARIABLE ( 1-D) REAL 1 REFS
R2 LOCAL RECURSIVE < 27 REAL 5 REFS
R3 LOCAL RECURSIVE < 28 REAL 6 REFS
R4 LOCAL RECURSIVE < 29 REAL 6 REFS
R5 LOCAL RECURSIVE < 30 REAL 5 REFS
SDIF LOCAL RECURSIVE < 31 REAL 3 REFS
SF GLOBAL VARIABLE REAL 1 REFS
SNES GLOBAL VARIABLE ( 1-D) REAL 8 REFS
SRRATE GLOBAL VARIABLE ( 1-D) REAL 1 REFS
SU GLOBAL VARIABLE REAL 1 REFS
S1 GLOBAL RECURSIVE < 35 REAL 4 REFS
S2 LOCAL RECURSIVE < 40 REAL 4 REFS
S3 LOCAL RECURSIVE < 47 REAL 3 REFS
T LOCAL RECURSIVE < 17 REAL 6 REFS
TAR LOCAL RECURSIVE < 3 REAL 4 REFS
TIME.V SYSTEM ATTRIBUTE REAL 11 REFS
TLID GLOBAL VARIABLE ( 2-D) INTEGER 1 REFS
TRCHG GLOBAL VARIABLE ( 1-D) REAL 1 REFS
TTF LOCAL RECURSIVE < 34 REAL 12 REFS
T1 LOCAL RECURSIVE < 23 REAL 8 REFS
T2 LOCAL RECURSIVE < 24 REAL 4 REFS
T3 LOCAL RECURSIVE < 25 REAL 7 REFS
T4 LOCAL RECURSIVE < 26 REAL 6 REFS
USAGE.LIFE GLOBAL VARIABLE REAL 1 REFS

```

ARM	GLOBAL VARIABLE	(1-D) REAL	2 REFS
ICBKT	GLOBAL VARIABLE	REAL	1 REFS
1.CP-EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
1.INT	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
1.ITE	EVENT NOTICE		2 REFS
1.PROB	GLOBAL VARIABLE	REAL	1 REFS
1.STRENGTHM.REDUCTION	EVENT NOTICE		1 REFS

```

1  EVENT 2.STRENGTH-REDUCTION(ID2SP) SAVING THE EVENT NOTICE
2  **
3  ** REPRESENTS SECOND CRACK INITIATION
4  **
5  DEFINE ID2SP AS AN INTEGER VARIABLE
6  LET ID = ID2SP
7  CALL COMP.RISK YIELDING CL
8  LET SRRATE(ID) = SRRATE(ID) + (MSR1(ID)*CGRI(ID)*SOIF(ID))/HLGT(ID)
9  LET TRCHG(ID) = TIME.V
10 LET A = AMEAN * AFACCT(ID)
11 LET TAR = ENTRY.TIME(AIRPLANE(ID)) + USAGE.LIFE
12 IF LTMO = "YES"
13   FOR I = 1 TO NOAC(LDX)
14     DO
15       IF ID = TLIP(LDX,I)
16         LET LIST = 1.0
17         SKIP 1 OUTPUT LINE
18         PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
19         A/C NO. *** EXPERIENCES 2ND CRACK INITIATION AT ***** HOURS
20         LEAVE
21       ELSE
22         LOOP
23     ALWAYS
24     LET 2.INT(ID) = "NO"
25     IF RANDOM.F(10) LE 1.PROB
26       LET 2.INT(ID) = "YES"
27       LET MSR1(ID) = MSR1.INT(ID)
28       LET MFR1(ID) = MFR1.INT(ID)
29       LET MSR2(ID) = MSR2.INT(ID)
30       LET MFR2(ID) = MFR2.INT(ID)
31       LET CONE = INT.CONE
32       LET CTWO = INT.CTWO
33       LET CTHREE = IN.CTHREE
34       IF LIST = 1.0
35         PRINT 1 LINE AS FOLLOWS
36         CRACK INITIATES INTERNALLY
37         ALWAYS
38         LET T = L.EXT/MSR1(ID)*CGRI(ID)
39         IF L.EXT GT COME
40           LET T = COME / (MSR1(ID) * CGRI(ID))
41           * (L.EXT - COME) / (MFR1(ID) * CGRI(ID))
42         ALWAYS
43         IF L.EXT GT CTWO
44           LET T = COME/(MSR1(ID)*CGRI(ID))
45           * (CTWO-CONE)/(MFR1(ID)*CGRI(ID))
46           * (L.EXT-CTWO)/(MSR2(ID)*CGRI(ID))
47         ALWAYS
48         IF L.EXT GT CTHREE
49           LET T=CONF/MSR1(ID)*CGRI(ID)+(CTWO-CONE)/
50             (MFR1(ID)*CGRI(ID))+(CTHREE-CTWO)/(MSR2(ID)*
51             CGRI(ID))+(L.EXT-CTHREE)/(MFR2(ID)*CGRI(ID))
52         ALWAYS
53         IF TIME.V + T LT TAR OR CO.EXISTS(ID) = "NS"

```

```

52 SCHEDULE A 2.ITE(ID) AT TIME.V * T
53 LET A2E(ID) = 2.ITE
54 LET IER(ID) = "YES"
55 ALWAYS
56 REGARDLESS.
57 LET 2.CR.EXISTS(ID) = "YES"
58 LET 1.STRENGTH.REDUCTION = A1SW(ID)
59 LET CCL1 = COME
60 LET CCL2 = CTWO
61 LET CCL3 = CTHREE
62 LET TAL = TIME.A(1).STRENGTH.REDUCTION
63 LET TCL1 = CCL1/(MSR1(ID)*CGRI(ID))
64 LET TCL2 = (CCL2-CCL1)/(MSR1(ID)*CGRI(ID))*TCL1
65 LET TCL3 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))*TCL2
66
67 DETERMINE RESIDUAL STRENGTH REDUCTION BECAUSE OF FIRST CRACK
68
69 IF CO-EXISTS(ID) = "YES"
70 LET COROSION = AC(ID)
71 LET TAC = TIME.A(COROSION)
72 IF TAC LE TAL
73 LET CL = (TIME.V-TAL) * MSR1(ID) * CGRI(ID)
74 IF CL GT CCL1
75 LET CL = CCL1 * ((CL-CCL1)/MSR1(ID))*MSR1(ID)
76 ALWAYS
77 IF CL GT CCL2
78 LET CL = CCL2 * ((CL-CCL2)/MSR1(ID))*MSR2(ID)
79 ALWAYS
80 IF CL GT CCL3
81 LET CL = CCL3 * ((CL-CCL3)/MSR2(ID))*MSR2(ID)
82 ALWAYS
83 JUMP AHEAD
84 OTHERWISE
85 LET DL = (TAC-TAL)*MSR1(ID)
86 IF CCL1 LT DL
87 LET CL = CCL1 * ((DL-CCL1)/MSR1(ID))*MSR1(ID) * (TIME.V-TAC)*MSR1(ID) *
      CGRI(ID)
88 JUMP AHEAD
89 OTHERWISE
90 IF CCL2 LT DL
91 LET CL = CCL2 * ((DL-CCL2)/MSR1(ID))*MSR2(ID) * (TIME.V-TAC)*MSR2(ID) *
      CGRI(ID)
92 JUMP AHEAD
93 OTHERWISE
94 IF CCL3 LT DL
95 LET CL = CCL3 * ((DL-CCL3)/MSR2(ID))*MSR2(ID) * (TIME.V-TAC)*MSR2(ID) *
      CGRI(ID)
96 JUMP AHEAD
97 OTHERWISE
98 IF CCL1 LT DL
99 LET CL = DL * (TIME.V-TAC)*MSR1(ID)*CGRI(ID)
100 IF CCL2 LT DL
101 LET CL = CCL1 * ((CL-CCL1)/MSR1(ID))*MSR1(ID)
102 REGARDLESS
103
104

```

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.9-00/ RRGNDOS 2.1.2

```

105 IF CCL2 LT CL
106 LET CL = CCL2 + ((CL-CCL2)/MFR1(ID))*MSR2(ID)
107 REGARDLESS
108 IF CCL3 LT CL
109 LET CL = CCL3 + ((CL-CCL3)/MSR2(ID))*MFR2(ID)
110 REGARDLESS
111 JUMP AHEAD
112 OTHERWISE
113 LET CL = (TIME.V-TA1)*MSR1(ID)
114 IF CL GT CCL1
115 LET CL = CCL1 + ((CL-CCL1)/MSR1(ID))*MFR1(ID)
116 REGARDLESS
117 IF CL GT CCL2
118 LET CL = CCL2 + ((CL-CCL2)/MFR1(ID))*MSR2(ID)
119 REGARDLESS
120 IF CL GT CCL3
121 LET CL = CCL3 + ((CL-CCL3)/MSR2(ID))*MFR2(ID)
122 REGARDLESS
123 HERE
124
125 ** PREDICT TIME TO FAILURE FROM SECOND CRACK INITIATION
126 **
127 LET TT12 = (CCL2-CCL1)/(MFR1(ID)*CGRI(ID))
128 LET TT23 = (CCL3-CCL2)/(MSR2(ID)*CGRI(ID))
129 IF CL LT CCL1
130 LET T1 = TIME.V-TA1
131 LET GR1 = MSR1(ID)*CGRI(ID)
132 LET T2 = (CCL1-CL)/GR1 + T1
133 LET GR2 = 2.0 * MSR1(ID)*CGRI(ID)
134 LET GR3 = (MSR1(ID)*MFR1(ID))*CGRI(ID)
135 IF TCL1+T1 GT T2 + TT12
136 LET T3 = TT12 + T2
137 LET T4 = TCL1+T1
138 LET GR4 = (MSR1(ID) + MSR2(ID))*CGRI(ID)
139 JUMP AHEAD
140 OTHERWISE
141 LET T3 = TCL1+T1
142 LET GR4 = 2.0*MFR1(ID)*CGRI(ID)
143 LET T4 = TT12 + T2
144 HERE
145 IF TCL2+T1 GT T2 + TT12 + TT23
146 LET T5 = T2 + TT12 + TT23
147 LET GR5 = (MFR1(ID) + MSR2(ID))*CGRI(ID)
148 LET T6 = TCL2+T1
149 LET GR6 = (MFR1(ID)*MFR2(ID))*CGRI(ID)
150 JUMP AHEAD
151 OTHERWISE
152 LET T5 = TCL2+T1
153 LET T6 = T2 + TT12 + TT23
154 LET GR6 = 2.0*(MSR2(ID)*CGRI(ID))
155 LET GR5 = (MFR1(ID)*MSR2(ID))*CGRI(ID)
156 HERE
157 LET T7 = TCL3+T1

```

```

158 LET GR7 = (MSR2(ID)*MSR2(ID))*CGRI(ID)
159 LET GR8 = 2.0*MSR2(ID)*CGRI(ID)
160 LET T8 = T7 + (FSAP*LG1 - CCL3)/GR8
161 ALWAYS
162 IF CL LT CCL2 AND CL GE CCL1
163 LET T1 = TIME.V - (CL-CCL1)/(MFR1(ID)*CGRI(ID))-TA1
164 LET GR1 = MSPI(ID)*CGRI(ID)
165 LET T2 = TIME.V-TA1
166 LET GR2 = MFR1(ID)*CGRI(ID)
167 LET T3 = TIME.V + (CCL2-CL)/(MFR1(ID)*CGRI(ID))-TA1
168 LET GR3 = (MFR1(ID)*MSK1(ID))*CGRI(ID)
169 IF TCL1-T2 GT T3 + TT23
170 LET T4 = T3 + TT23
171 LET GR4 = (MSR1(ID) + MSR2(ID))*CGRI(ID)
172 LET T5 = TCL1-T2
173 LET GR5 = (MSH1(ID) + MFM/(ID))*CGRI(ID)
174 JUMP AHEAD
175 OTHERWISE
176 LET T4 = TCL1-T2
177 LET T5 = T3 + TT23
178 LET GR5 = (MFR1(ID) + MSK2(ID))*CGRI(ID)
179 LET GR4 = (MSPI(ID)*MSR2(ID))*CGRI(ID)
180 MERGE
181 LET T6 = TCL2-T2
182 LET T7 = TCL3-T2
183 LET GR6 = (MFR1(ID) + MFM2(ID))*CGRI(ID)
184 LET GR7 = (MSR2(ID) + MFM2(ID))*CGRI(ID)
185 LET GR8 = 2.0*MSR2(ID)*CGRI(ID)
186 LET T8 = T7 + (FSAP*LG1 - CCL3)/GR8
187 ALWAYS
188 IF CL LT CCL3 AND CL GE CCL2
189 LET T1 = TIME.V - (CL-CCL2)/(MSR2(ID)*CGRI(ID))-TT12-TA1
190 LET GR1 = MSR1(ID) + CGRI(ID)
191 LET T2 = T1 + TT12
192 LET GR2 = MFR1(ID) + CGRI(ID)
193 LET T3 = TIME.V-TA1
194 LET GR3 = MSR2(ID) + CGRI(ID)
195 IF TCL1-T3 GT T2 + TT12
196 LET T4 = T2+TT12
197 LET T5 = TCL1-T3
198 LET GR4 = (MSR2(ID) + MSK1(ID))*CGRI(ID)
199 LET GR5 = (MFR2(ID) + MSK1(ID))*CGRI(ID)
200 JUMP AHEAD
201 OTHERWISE
202 LET T4 = TCL1-T3
203 LET T5 = T2 + TT12
204 LET GR5 = (MFR1(ID) + MSR2(ID))*CGRI(ID)
205 LET GR4 = (MSR2(ID)*MSR1(ID))*CGRI(ID)
206 MERGE
207 LET T6 = TCL2-T3
208 LET T7 = TCL3-T3
209 LET GR6 = (MFR1(ID) + MFM2(ID))*CGRI(ID)
210 LET GR7 = (MSR2(ID) + MFM2(ID))*CGRI(ID)

```

```

211 LET GR8 = 2.0*MF2(ID)*CGRI(ID)
212 LET T8 = T7 + (FSAF.LGT - CCL3)/GR8
213 ALWAYS
214 IF CL GE CCL3
215 LET T1 = TIME.V - (CL-CCL3)/(MF2(ID)*CGRI(ID))-TT23 - TT12-TA1
216 LET GR1 = MSRI(ID)*CGRI(ID)
217 LET T2 = T1 + TT12
218 LET GR2 = MF2(ID)*CGRI(ID)
219 LET T3 = T2 + TT23
220 LET GR3 = MSRI(ID)*CGRI(ID)
221 LET T4 = TIME.V-TA1
222 LET GR4 = MF2(ID)*CGRI(ID)
223 LET T5 = T4 + T4
224 LET GR5 = MSRI(ID) + MF2(ID)*CGRI(ID)
225 LET T6 = T5 + T4
226 LET GR6 = (MF1(ID) + MF2(ID))*CGRI(ID)
227 LET T7 = T6 + T4
228 LET GR7 = (MSRI(ID) + MF2(ID))*CGRI(ID)
229 LET GR8 = 2.0*MF2(ID)*CGRI(ID)
230 LET T8 = T7 + (FSAF.LGT - CCL3)/GR8
231 ALWAYS
232 LET C2 = (LGMT.T0-FAILURE-FSAF.LGT)
233 LET C1 = (SU-SF)/FSAF.LGT
234 LET R1 = C1*GR1
235 LET R2 = C1*GR2
236 LET R3 = C1*GR3
237 LET R4 = C1*GR4
238 LET R5 = C1*GR5
239 LET R6 = C1*GR6
240 LET R7 = C1*GR7
241 LET R8 = C1*GR8
242 LET R9 = GR8/C2*SF
243 IF CL GE FSAF.LGT
244 LET R1 = GR1/C2
245 LET R2 = GR2/C2
246 LET R3 = GR3/C2
247 LET R4 = GR4/C2
248 LET R5 = GR5/C2
249 LET R6 = GR6/C2
250 LET R7 = GR7/C2
251 LET R8 = GR8/C2
252 LET R9 = R8
253 LET T8 = 0
254 REBARDLESS
255 IF (TA1 + T8 LT TAP OR CO.EXISTS(ID) = "MS") AND T8 GT 0.0
256 IF ATL(ID) = "YES"
257 LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
258 CANCEL THE REACH.FAIL.SAFE.LGT
259 RESCHEDULE THE REACH.FAIL.SAFE.LGT(ID) AT TA1 + T8
260 JUMP AHEAD
261 ELSE
262 SCHEDULE A REACH.FAIL.SAFE.LGT(ID) AT TA1 + T8
263 LET ARFSL(ID) = REACH.FAIL.SAFE.LGT

```

```

264 LET AIL(ID) = "YES"
265 MEBE
266 REGARDOLESS
267 LET LG = LOG.E.F(XRN(ID))
268 LET K1 = A * EXP.F(SU * 8)
269 LET K2 = 8 * K1
270 LET TTF = -LOG.E.F((K2/K1)*LG * 1.0)/K2
271 IF TTF LE T1
272 JUMP AHEAD
273 ELSE
274 LET S1 = SU - R1 * T1
275 LET K4 = R * K2
276 LET LK5 = LOG.E.F(A) * (8 * S1 + 8 * K2 * T1)
277 LET K8 = (A*EXP.F(8*S1))/K4
278 LET K9 = (K1/K2)*(EXP.F(-8*R1*T1))-1.0
279 LET TTF = (LK5 - LOG.E.F(ABS.F(K9)) - LOG.E.F(ABS.F(LG*K8-K9)))/K4
280 IF TTF LE T2
281 JUMP AHEAD
282 ELSE
283 LET S2 = S1 - R2*(T2-T1)
284 LET K10 = R*K3
285 LET ARG = 8*S2*R*R3*T2
286 LET K11 = -LOG.E.F(-K10) * LOG.E.F(A) * ARG
287 LET K12 = A*EXP.F(R*S2)/K10
288 LET K13 = EXP.F(8*R2*T1 - R*R2*T2) - 1.0
289 LET K16 = -LOG.E.F(-(LG*K12-(K8*K13)-K9))
290 LET TTF = (K11*K16)/K10
291 IF TTF LE T3
292 JUMP AHEAD
293 ELSE
294 LET S3 = S2 - R3*(T3-T2)
295 LET K15 = R*R4
296 LET K16 = -LOG.E.F(-K15)*LOG.E.F(A) * 8*S3 * 8*R4*T3
297 LET K17 = A*EXP.F(R*S3)/K15
298 LET K18 = EXP.F(8*R3*T2-R*R3*T3) - 1.0
299 LET K19 = -LOG.E.F(-(LG*K17-(K12*K18)-(K8*K13)-K9))
300 LET TTF = (K15*K19)/K15
301 IF TTF LE T4
302 JUMP AHEAD
303 ELSE
304 LET S4 = S3 - R4*(T4-T3)
305 LET K20 = R*R5
306 LET K21 = -LOG.E.F(-K20) * LOG.E.F(A) * 8*S4 * 8*R5*T4
307 LET K22 = A*EXP.F(R*S4)/K20
308 LET K23 = EXP.F(8*R4*T3-R*R4*T4) - 1.0
309 LET K24 = -LOG.E.F(-(LG*K22-(K17*K23)-(K12*K18)-(K8*K13)-K9))
310 LET TTF = (K21*K24)/K24
311 IF TTF LE T5
312 JUMP AHEAD
313 ELSE
314 LET S5 = S4 - R5*(T5-T4)
315 LET K25 = R*R6
316 LET K26 = -LOG.E.F(-K25) * LOG.E.F(A) * 8*S5 * 8*R6*T5

```

```

317 LET K27 = A*EXP.F(B*SS)/K25
318 LET K28 = EXP.F(B*RS*14-B*RS*15) - 1.0
319 LET K29 = -LOG.E.F(-(LG*K27-(K22*K28)-(K17*K23)-(K12*K18)-(K8*K13)-K9))
320 LET TTF = (K26*K29)/K25
321 IF TTF LE T6
322 JUMP AHEAD
323 ELSE
324 LET S6=S5 - R6/(T6-T5)
325 LET K30 = B*P7
326 LET K31 = -LOG.E.F(-(K30)*LOG.E.F(A)+B*G*8*B*7*T6
327 LET K32 = A*EXP.F(B*S6)/K30
328 LET K33 = EXP.F(B*RG*TS-B*RG*TB)-1.0
329 LET K34 = -LOG.E.F(-(LG*K32-K27*K33-K22*K28-K17*K23-K12*K18-K8*K13-K9))
330 LET TTF = (K31*K34)/K30
331 IF TTF LE T7
332 JUMP AHEAD
333 ELSE
334 LET S7 = S6 - R7/(T7-T6)
335 LET K35 = B*P8
336 LET K36 = -LOG.E.F(-(K35) * LOG.E.F(A) + B*ST* B*8*T7
337 LET K37 = A*EXP.F(B*S7)/K35
338 LET K38 = EXP.F(B*ST*TB-B*ST*TT)-1.0
339 LET K39 = -LOG.E.F(-(LG*K37-K32*K38-K27*K33-K22*K28-K17*K23-K12*K18
340 -K8*K13-K9))
341 LET TTF = (K36*K39)/K35
342 IF TTF LE T8
343 JUMP AHEAD
344 ELSE
345 LET K40 = B*P9
346 LET K41 = -LOG.E.F(-(K40) * LOG.E.F(A) + B*SF + B*P9*T8
347 LET K42 = A*EXP.F(B*SF)/K40
348 LET K43 = EXP.F(B*P9*TB-B*P9*TT)-1.0
349 LET K44 = -LOG.E.F(-(LG*K42-K37*K43-K32*K38-K27*K33-K22*K28-K17*K23
350 -K12*K18-K8*K13-K9))
351 LET TTF = (K41*K44)/K40
352 HERE
353 IF LIST = 1.0
354 PRINT 1 LINE WITH TAI*TTF-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
    ELEMENT FAILURE PROJECTED AT ***** FLIGHT HOURS
    ALWAYS
    IF TAI * TTF LT IAK OR CO.EXISTS(ID) = "NSH"
    IF FSK(ID) = "YES"
    LET FAILURE = AF(ID)
    CANCEL THE FAILURE
    IF TAI * TTF LT TIME.V
    RESCHEDULE THE FAILURE(ID) NOW
    LET AF(ID) = FAILURE
    JUMP AHEAD
    OTHERWISE
    RESCHEDULE THE FAILURE(ID) AT TAI * TTF
    LET AF(ID) = FAILURE
    JUMP AHEAD
    ELSE
355
356
357
358
359
360
361
362
363
364
365
366
367
368

```

```

369      LET FSHIID) = "YES"
370      IF TAI * TTF LT TIME.Y
371      SCHEDULE A FAILURE(ID) NOW
372      LET AF(ID) = FAILURE
373      JUMP AHEAD
374      OTHERWISE
375      SCHEDULE A FAILURE(ID) AT TAI * TTF
376      LET AF(ID) = FAILURE
377      HERE
378      ALWAYS
379      RETURN
380      END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT 2-STRENGTH-REDUCTION

A	LOCAL RECURSIVE	< 3	REAL	18 REFS
ABS.F	PROCEDURE		INTEGER	2 REFS
AC	GLOBAL VARIABLE		(1-D) INTEGER	1 REFS
AF	GLOBAL VARIABLE		(1-D) INTEGER	5 REFS
AFACT	GLOBAL VARIABLE		(1-D) REAL	1 REFS
AEL	GLOBAL VARIABLE		(1-D) ALPHA	2 REFS
AIRPLANE	GLOBAL VARIABLE		(1-D) INTEGER	3 REFS
ANEAN	GLOBAL VARIABLE		REAL	1 REFS
ARFSL	GLOBAL VARIABLE		(1-D) INTEGER	2 REFS
AR6	GLOBAL VARIABLE	< 68	REAL	2 REFS
ASR	GLOBAL VARIABLE		(1-D) INTEGER	1 REFS
AZE	GLOBAL VARIABLE		(1-D) INTEGER	1 REFS
B	GLOBAL VARIABLE		REAL	49 REFS
CCL1	LOCAL RECURSIVE	< 19	REAL	20 REFS
CCL2	LOCAL RECURSIVE	< 20	REAL	21 REFS
CCL3	LOCAL RECURSIVE	< 21	REAL	22 REFS
CGPI	GLOBAL VARIABLE		(1-D) REAL	44 REFS
CL	LOCAL RECURSIVE	< 2	REAL	45 REFS
COMP.RISK	PROCEDURE		INTEGER	1 REFS
CONE	GLOBAL VARIABLE		REAL	9 REFS
COROSION	EVENT NOTICE		PEAL	2 REFS
CO-EXISTS	GLOBAL VARIABLE		(1-D) ALPHA	4 REFS
CTHREE	GLOBAL VARIABLE		REAL	5 REFS
CTWO	GLOBAL VARIABLE		REAL	7 REFS
C1	LOCAL RECURSIVE	< 7	REAL	9 REFS
C2	LOCAL RECURSIVE	< 46	REAL	10 REFS
DL	LOCAL RECURSIVE	< 27	REAL	8 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE		REAL	3 REFS
EXP.F	PROCEDURE		REAL	17 REFS
FAILURE	EVENT NOTICE		REAL	10 REFS
FSAF.LGT	GLOBAL VARIABLE		REAL	7 REFS
FSH	GLOBAL VARIABLE		(1-D) ALPHA	2 REFS
GR1	LOCAL RECURSIVE	< 31	REAL	7 REFS
GR2	LOCAL RECURSIVE	< 33	REAL	6 REFS
GR3	LOCAL RECURSIVE	< 34	REAL	6 REFS
GR4	LOCAL RECURSIVE	< 37	REAL	9 REFS
GR5	LOCAL RECURSIVE	< 39	REAL	9 REFS
GR6	LOCAL RECURSIVE	< 41	REAL	7 REFS
GR7	LOCAL RECURSIVE	< 43	REAL	6 REFS
GR8	LOCAL RECURSIVE	< 44	REAL	11 REFS
I	LOCAL RECURSIVE	< 5	REAL	2 REFS
ID	GLOBAL VARIABLE		INTEGER	226 REFS
ID2SP	GIVEN ARGUMENT	< 1	INTEGER	3 REFS
IC2	TEMPORARY ATTRIBUTE		INTEGER	1 REFS
INT.COME	GLOBAL VARIABLE		(1-D) ALPHA	1 REFS
INT.CTWO	GLOBAL VARIABLE		REAL	1 REFS
IN-CTHREE	GLOBAL VARIABLE		REAL	1 REFS
K1	GLOBAL VARIABLE	< 58	REAL	3 REFS
K10	LOCAL RECURSIVE	< 67	REAL	4 REFS
K11	LOCAL RECURSIVE	< 69	REAL	2 REFS

CDC 6600	LOCAL RECURSIVE	< 70	REAL	8 REFS
K12	LOCAL RECURSIVE	< 71	REAL	6 REFS
K13	LOCAL RECURSIVE	< 72	REAL	2 REFS
K14	LOCAL RECURSIVE	< 74	REAL	4 REFS
K15	LOCAL RECURSIVE	< 75	REAL	7 REFS
K16	LOCAL RECURSIVE	< 76	REAL	7 REFS
K17	LOCAL RECURSIVE	< 77	REAL	7 REFS
K18	LOCAL RECURSIVE	< 78	REAL	2 REFS
K19	LOCAL RECURSIVE	< 59	REAL	4 REFS
K20	LOCAL RECURSIVE	< 80	REAL	4 REFS
K21	LOCAL RECURSIVE	< 81	REAL	2 REFS
K22	LOCAL RECURSIVE	< 82	REAL	6 REFS
K23	LOCAL RECURSIVE	< 83	REAL	6 REFS
K24	LOCAL RECURSIVE	< 84	REAL	2 REFS
K25	LOCAL RECURSIVE	< 86	REAL	4 REFS
K26	LOCAL RECURSIVE	< 87	REAL	2 REFS
K27	LOCAL RECURSIVE	< 88	REAL	5 REFS
K28	LOCAL RECURSIVE	< 89	REAL	5 REFS
K29	LOCAL RECURSIVE	< 90	REAL	2 REFS
K30	LOCAL RECURSIVE	< 92	REAL	4 REFS
K31	LOCAL RECURSIVE	< 93	REAL	7 REFS
K32	LOCAL RECURSIVE	< 94	REAL	4 REFS
K33	LOCAL RECURSIVE	< 96	REAL	4 REFS
K34	LOCAL RECURSIVE	< 96	REAL	2 REFS
K35	LOCAL RECURSIVE	< 98	REAL	4 REFS
K36	LOCAL RECURSIVE	< 99	REAL	2 REFS
K37	LOCAL RECURSIVE	< 100	REAL	3 REFS
K38	LOCAL RECURSIVE	< 101	REAL	1 REFS
K39	LOCAL RECURSIVE	< 102	REAL	2 REFS
K40	LOCAL RECURSIVE	< 103	REAL	4 REFS
K41	LOCAL RECURSIVE	< 104	REAL	2 REFS
K42	LOCAL RECURSIVE	< 105	REAL	2 REFS
K43	LOCAL RECURSIVE	< 106	REAL	2 REFS
K44	LOCAL RECURSIVE	< 107	REAL	2 REFS
K8	LOCAL RECURSIVE	< 64	REAL	9 REFS
K9	LOCAL RECURSIVE	< 65	REAL	2 REFS
L0K	GLOBAL VARIABLE	< 65	INTEGER	2 REFS
L6	LOCAL RECURSIVE	< 7	REAL	10 REFS
L6NT.10.FAILURE	GLOBAL VARIABLE	< 7	REAL	1 REFS
L6K5	LOCAL RECURSIVE	< 63	REAL	2 REFS
L1ST	LOCAL RECURSIVE	< 6	REAL	3 REFS
L0G.E.F	PROCEDURE		REAL	26 REFS
L7H0	GLOBAL VARIABLE		ALPHA	1 REFS
L.FEY	GLOBAL VARIABLE		REAL	7 REFS
WFR1	GLOBAL VARIABLE		REAL	31 REFS
WFR1.INT	GLOBAL VARIABLE		REAL	1 REFS
WFR2	GLOBAL VARIABLE		REAL	24 REFS
WFR2.INT	GLOBAL VARIABLE		REAL	1 REFS
WFR1	GLOBAL VARIABLE		REAL	30 REFS
WFR1.INT	GLOBAL VARIABLE		REAL	1 REFS
WFR2	GLOBAL VARIABLE		REAL	31 REFS
WFR2.INT	GLOBAL VARIABLE		REAL	1 REFS

NOAC	GLOBAL VARIABLE	(1-0)	INTEGER	1 REFS
RANDOM.F	PROCEDURE		REAL	1 REFS
REACH.FAIL.SAFE.LGT	EVENT NOTICE			5 REFS
RLGT	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
R1	LOCAL RECURSIVE	< 48		5 REFS
R2	LOCAL RECURSIVE	< 49		7 REFS
R3	LOCAL RECURSIVE	< 50		7 REFS
R4	LOCAL RECURSIVE	< 51		7 REFS
R5	LOCAL RECURSIVE	< 52		7 REFS
R6	LOCAL RECURSIVE	< 53		7 REFS
R7	LOCAL RECURSIVE	< 54		7 REFS
R8	LOCAL RECURSIVE	< 55		7 REFS
R9	LOCAL RECURSIVE	< 56		4 REFS
SDIF	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
SF	GLOBAL VARIABLE		REAL	4 REFS
SRHATF	GLOBAL VARIABLE	(1-0)	REAL	2 REFS
SU	GLOBAL VARIABLE		REAL	3 REFS
S1	LOCAL RECURSIVE	< 61		4 REFS
S2	LOCAL RECURSIVE	< 66		4 REFS
S3	LOCAL RECURSIVE	< 73		4 REFS
S4	LOCAL RECURSIVE	< 79		4 REFS
S5	LOCAL RECURSIVE	< 85		4 REFS
S6	LOCAL RECURSIVE	< 91		4 REFS
S7	LOCAL RECURSIVE	< 97		3 REFS
T	LOCAL RECURSIVE	< 18		6 REFS
TAC	LOCAL RECURSIVE	< 26		7 REFS
TAR	LOCAL RECURSIVE	< 4		4 REFS
TA1	LOCAL RECURSIVE	< 22		22 REFS
TCL1	LOCAL RECURSIVE	< 23		12 REFS
TCL2	LOCAL RECURSIVE	< 24		8 REFS
TCL3	LOCAL RECURSIVE	< 25		5 REFS
TIME.A	TEMPORARY ATTRIBUTE		REAL	2 REFS
TIME.V	SYSTEM ATTRIBUTE	(2-0)	INTEGER	20 REFS
TL.D	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
TRCHG	GLOBAL VARIABLE		REAL	1 REFS
TTF	LOCAL RECURSIVE	< 60		23 REFS
TI12	LOCAL RECURSIVE	< 28		14 REFS
TI23	LOCAL RECURSIVE	< 29		9 REFS
T1	LOCAL RECURSIVE	< 30		20 REFS
T2	LOCAL RECURSIVE	< 32		25 REFS
T3	LOCAL RECURSIVE	< 35		19 REFS
T4	LOCAL RECURSIVE	< 36		16 REFS
T5	LOCAL RECURSIVE	< 38		13 REFS
T6	LOCAL RECURSIVE	< 40		11 REFS
T7	LOCAL RECURSIVE	< 42		13 REFS
T8	LOCAL RECURSIVE	< 45		12 REFS
USAGE.LIFE	GLOBAL VARIABLE		REAL	1 REFS
XRN	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
1.PROB	GLOBAL VARIABLE		REAL	1 REFS
1.STRENGTH	REDUCTION	(1-0)	ALPHA	2 REFS
2.OP.EXISTS	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
2.INT	GLOBAL VARIABLE			2 REFS
2.IIE	EVENT NOTICE			2 REFS

1971

CDC 6600 CACI SIMSCRIPT II.5 VERSION /A-0-00/ KPMODS 2.1.2
2.STRENGTH.REDUCTION EVENT NOTICE

03/02/78. 18.36.02. PAGE 106

1 REFS

```

1  EVENT 1.ITE(IDIE)
2  DEFINE IDIE AS AN INTEGER VARIABLE
3  LET ID = IDIE
4  LET 1.INT(ID) = "NO"
5  LET 1.EI(ID) = "NO"
6  LET 1.LGT(ID) = 1.EXT
7  IF LTMO = "YES"
8  FOR I = 1 TO MOAC(LDX)
9  DO
10 IF ID = TLID(LDX,I)
11 SKIP 1 OUTPUT LINE
12 PRINT 1 DOUBLE LINE WITH ID, L.EXT, TIME.V-
13 ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
14 A/C NO. *** HAS INTERNAL FIRST CRACK BECOME EXTERNAL AT *** INCHES AND ***** FLIGHT HOURS
15 LEAVE
16 ELSE
17 LOOP
18 ALWAYS
19 RETURN
20 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT 1.ITE

AIRPLANE	GLOBAL VARIABLE	(1-0)	INTEGER	1 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE		REAL	1 REFS
I	LOCAL RECURSIVE	< 2	REAL	2 REFS
ID	GLOBAL VARIABLE		INTEGER	7 REFS
ID1E	GIVEN ARGUMENT	< 1	INTEGER	
	TEMPORARY ATTRIBUTE		INTEGER	3 REFS
IE1	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
INT.LGT	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
LDX	GLOBAL VARIABLE		INTEGER	2 REFS
LYMO	GLOBAL VARIABLE		ALPHA	1 REFS
L-EXT	GLOBAL VARIABLE		REAL	2 REFS
NOAC	GLOBAL VARIABLE	(1-0)	INTEGER	1 REFS
TIME.V	SYSTEM ATTRIBUTE		REAL	1 REFS
TLID	GLOBAL VARIABLE	(2-0)	INTEGER	1 REFS
1-INT	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
1.ITE	EVENT NOTICE			1 REFS

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2 03/02/78. 18.36.02. PAGE 109
 1 EVENT 2.ITE(102E)
 2 DEFINE_ID2E AS AN INTEGER VARIABLE
 3 LET ID = 102E
 4 LET 2.INT(1D) = "NO"
 5 LET IER(1D) = "NO"
 6 LET INT.1GT(1D) = L.EXT
 7 IF LTHO = "YES"
 8 FOR I = 1 TO NOAC(1DX)
 9 DO
 10 IF ID = 11D(1DX,1)
 11 SKIP 1 OUTPUT LINE
 12 PRINT 1 DOUBLE LINE WITH ID, L.EXT, TIME.V-
 13 ENTRY.TIME(1AIRPLANE(1D)) AS FOLLOWS
 14 A/C NO. *** HAS INTERNAL SECOND CRACK BECOME EXTERNAL AT **.** INCHES AND ***** FLIGHT HOURS
 15 LEAVE
 16 ELSE
 17 LOOP
 18 ALWAYS
 19 RETURN
 20 END

SYMBOLIC REFERENCE MAP (R = 1) - EVENT 2.ITE

AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	REAL	1 REFS
I	LOCAL RECURSIVE	REAL	2 REFS
ID	GLOBAL VARIABLE	INTEGER	7 REFS
IDZE	GIVEN ARGUMENT	INTEGER	3 REFS
	+TEMPORARY ATTRIBUTE	INTEGER	3 REFS
IE2	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
INT.LGT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
LOX	GLOBAL VARIABLE	INTEGER	2 REFS
LTHO	GLOBAL VARIABLE	ALPHA	1 REFS
LSEXT	GLOBAL VARIABLE	REAL	2 REFS
NOAC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	1 REFS
TLID	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
2.INT	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2.ITE	EVENT NOTICE		1 REFS

```

1  ROUTINE INSPECTION.SCHEDULER(N)
2  **
3  ** SCHEDULES INSPECTIONS BELOW OVERHAUL LEVEL ON A GIVEN AIRCRAFT SUCH THAT THE
4  ** AIRCRAFT IS NOT INSPECTED PRIOR TO CRACK REACHING MINIMUM DETECTABLE LGT
5  **
6  DEFINE N AS AN INTEGER VARIABLE
7  LET M1 = MSRI(ID) * CGRI(ID)
8  LET C1 = C.GROWTH.RATE * CRRF
9  IF EXT.INSPE.LEVEL LE 1
10 LET TML = TIME.V * .551/M1
11 IF N = 2
12 LET TML = TIME.V * .996/C1
13 ALWAYS
14 LET S.INSPE.AT = ENTRY.TIME(AIRPLANE(ID)) + IANCD(1)
15 + TRUNC.F((TML-ENTRY.TIME(AIRPLANE(ID)))/IABCD(1)) * 1.0)
16 SCHEDULE AN A.LEVEL.INSPECTION(ID) AT S.INSPE.AT
17 LET AAL(ID) = A.LEVEL.INSPECTION
18 REGARDLESS
19
20 IF EXT.INSPE.LEVEL LE 2
21 LET TML = TIME.V * .41/M1
22 IF N = 2
23 LET TML = TIME.V * .635/C1
24 ALWAYS
25 LET S.INSPE.AT = ENTRY.TIME(AIRPLANE(ID)) + IANCD(2)
26 + TRUNC.F((TML-ENTRY.TIME(AIRPLANE(ID)))/IABCD(2)) * 1.0)
27 SCHEDULE A B.LEVEL.INSPECTION(ID) AT S.INSPE.AT
28 LET ABL(ID) = B.LEVEL.INSPECTION
29 REGARDLESS
30
31 IF EXT.INSPE.LEVEL LE 3
32 LET TML = TIME.V * .266/M1
33 IF N = 2
34 LET TML = TIME.V * .563/C1
35 ALWAYS
36 LET S.INSPE.AT = ENTRY.TIME(AIRPLANE(ID)) + C.INTERVAL(ID)
37 + TRUNC.F((TML-ENTRY.TIME(AIRPLANE(ID)))/C.INTERVAL(ID)) * 1.0)
38 SCHEDULE A C.LEVEL.INSPECTION(ID) AT S.INSPE.AT
39 LET ACL(ID) = C.LEVEL.INSPECTION
40 REGARDLESS
41
42 LET INSP.SCH(ID) = "YES"
43 RETURN
44 END

```

SYMBOLIC REFERENCE MAP (R = I) - ROUTINE INSPECTION, SCHEDULER

AAL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ATPLANE	GLOBAL VARIABLE	(1-D) INTEGER	6 REFS
A.LEVEL.INSPECTION	EVENT NOTICE	(1-D) INTEGER	2 REFS
B.LEVEL.INSPECTION	EVENT NOTICE	(1-D) REAL	2 REFS
CBRI	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CRRF	GLOBAL VARIABLE	WFAL	1 REFS
C1	LOCAL RECURSIVE	WFAL	4 REFS
C.GROWTH.RATE	GLOBAL VARIABLE	REAL	1 REFS
C.INTERVAL	GLOBAL VARIABLE	(1-D) REAL	2 REFS
C.LEVEL.INSPECTION	EVENT NOTICE	REAL	7 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	REAL	4 REFS
EXT.INSPECTION	GLOBAL VARIABLE	INTEGER	3 REFS
ID	GLOBAL VARIABLE	INTEGER	17 REFS
INSPECTION.SCHEDULER	PROCEDURE	INTEGER	1 REFS
INSPECTION	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
INSPI	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MI	LOCAL RECURSIVE	REAL	4 REFS
N	GIVEN ARGUMENT	INTEGER	5 REFS
S.INSPECTION	LOCAL RECURSIVE	REAL	6 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	6 REFS
TML	LOCAL RECURSIVE	WFAL	9 REFS
TRUNC.F	PROCEDURE	INTEGER	3 REFS
LANCD	GLOBAL VARIABLE	(1-D) REAL	4 REFS

```

1  EVENT A-LEVEL-INSPECTION(IDA)
2  DEFINE IDA AS AN INTEGER VARIABLE
3  LET ID = IDA
4  LET FIXIT.COST = A.REPAIR.COST
5  *#
6  CALL EXAMINE(1.012,4.,996.012,4.,551) YIELDING FOUND
7  IF FOUND = 1
8  LET CR.CTR = CR.CTR + 1
9  IF LTHO = "NO"
10 IF CP.CTR = 1 AND JFLAG="NO"
11 PRINT 1 LINE WITH TIME.V, MOD.NO AS FOLLOWS
12 NON-EXPLORATORY DETECTION LEVEL AT ***** MODIFICATION *
13 LET JFLAG="YES"
14 ALWAYS
15 ALWAYS
16 *#
17 IF FOUND = 1 OR FOUND = 2
18 FOR I=1 TO 10
19 DO
20 IF MI.TIME.ACHFT(I)=10
21 LET MRDD(I) = TIME.V
22 LEAVE
23 ELSE
24 LOOP
25 REGARDLESS
26 SCHEDULE AN A-LEVEL-INSPECTION(ID) AT TIME.V + 1ABCO(1)
27 LET AAL(ID) = A.LEVEL-INSPECTION
28 LET D.LMT.FIND = "NO"
29 RETURN
30 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT A.LEVEL.INSPECTION

AAL	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
A.LEVEL.INSPECTION	EVENT NOTICE			3 REFS
A.REPAIR.COST	GLOBAL VARIABLE		REAL	1 REFS
CR.CTR	GLOBAL VARIABLE		INTEGER	3 REFS
D.INT.FIND	GLOBAL VARIABLE		ALPHA	1 REFS
EXAMINE	PROCEDURE		INTEGER	1 REFS
FIXIT.COST	GLOBAL VARIABLE		REAL	1 REFS
FOUND	LOCAL RECURSIVE	< 2	REAL	4 REFS
MI.TIME.ACRFT	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
I	LOCAL RECURSIVE	< 13	REAL	3 REFS
ID	GLOBAL VARIABLE		INTEGER	4 REFS
IDA	GIVEN ARGUMENT	< 1	INTEGER	3 REFS
	*TEMPORARY ATTRIBUTE		INTEGER	2 REFS
JFLAG	GLOBAL VARIABLE		ALPHA	1 REFS
LTHO	GLOBAL VARIABLE		ALPHA	1 REFS
MOD.NO	GLOBAL VARIABLE		INTEGER	1 REFS
NRDD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
TIME.V	SYSTEM ATTRIBUTE		REAL	3 REFS
LABCD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS

```

1  EVENT B.LEVEL.INSPECTION(IDB)
2  DEFINE IDB AS AN INTEGER VARIABLE
3  LET ID = IDB
4  SCHEDULE A B.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(2)
5  LET ABL(ID) = B.LEVEL.INSPECTION
6  LET C.LEVEL.INSPECTION = ACL(ID)
7  LET D.LEVEL.INSPECTION = ADL(ID)
8  IF TIME.AIG.LEVEL.INSPECTION - TIME.V LE ABCD(2) / 2
9  OR TIME.AID.LEVEL.INSPECTION - TIME.V LE ABCD(2) / 2
10 RETURN
11 OTHERWISE
12
13 IF EXT.INSPECTION LE 1
14 LET A.LEVEL.INSPECTION = AAL(ID)
15 CANCEL THE A.LEVEL.INSPECTION
16 RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(1)
17 LET AAL(ID) = A.LEVEL.INSPECTION
18 ALWAYS
19 LET FIXIT.COST = B.REPAIR.COST
20
21 CALL EXAMINE(2,.045,.4,.635,.045,.4,.41) YIELDING FOUND
22 IF FOUND = 1
23 LET CR.CTR = CR.CTR + 1
24 IF LTHO = "NO"
25 IF CR.CTR = 1 AND JFLAG="NO"
26 PRINT 1 LINE WITH TIME.V, MOD.NO AS FOLLOWS
27   NON-EXPLORATORY DETECTION LEVEL AT ***** MODIFICATION. *
28   LET JFLAG="YES"
29   ALWAYS
30   ALWAYS
31
32 IF FOUND = 1 OR FOUND = 2
33   FOR I=1 TO 10
34   DO
35   IF HI.TIME.ACFT(I) = ID
36   LET MRDD(I) = TIME.V
37   LEAVE
38   ELSE
39   LOOP
40 REGARDLESS
41 LET D.INT.FIND = "NO"
42 RETURN
43 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT B.LEVEL.INSPECTION

AAL	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
ARCD	GLOBAL VARIABLE	(1-D) REAL	2 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
A.LEVEL.INSPECTION	EVENT NOTICE		4 REFS
B.LEVEL.INSPECTION	EVENT NOTICE		3 REFS
B.REPAIR.COST	GLOBAL VARIABLE	REAL	1 REFS
CR.CTR	GLOBAL VARIABLE	INTEGER	3 REFS
C.LEVEL.INSPECTION	EVENT NOTICE		2 REFS
D.INT.FIND	GLOBAL VARIABLE	ALPHA	1 REFS
D.LEVEL.INSPECTION	EVENT NOTICE		2 REFS
EXAMINE	PROCEDURE	INTEGER	1 REFS
EXT.IMP.LVL	GLOBAL VARIABLE	INTEGER	1 REFS
FIXT.COST	LOCAL VARIABLE	REAL	1 REFS
FOUND	LOCAL RECURSIVE	REAL	4 REFS
HI.TIME.ACRFT	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
I	LOCAL RECURSIVE	REAL	3 REFS
ID	GLOBAL VARIABLE	INTEGER	9 REFS
IDB	GIVEN ARGUMENT	INTEGER	
	*TEMPORARY ATTRIBUTE	INTEGER	3 REFS
JFLAG	GLOBAL VARIABLE	ALPHA	2 REFS
LTHO	GLOBAL VARIABLE	ALPHA	1 REFS
MOD.NO	GLOBAL VARIABLE	INTEGER	1 REFS
MRDD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
TIME.A	TEMPORARY ATTRIBUTE	REAL	2 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	6 REFS
LABCD	GLOBAL VARIABLE	(1-D) REAL	2 REFS

1 208 1

```

1  EVENT C.LEVEL.INSPECTION(IDC)
2  DEFINE IDC AS AN INTEGER VARIABLE
3  LET ID = IDC
4  SCHEDULE A C.LEVEL.INSPECTION(ID) AT TIME.V + C.INTERVAL(ID)
5  LET ACL(ID) = C.LEVEL.INSPECTION
6  LET D.LEVEL.INSPECTION = ADL(ID)
7  IF TIME.AID.LEVEL.INSPECTION - TIME.V LE ABCD(3) / 2
8  RETURN
9  OTHERWISE
10
11  IF EXT.INSPECTION LE 2
12  LET B.LEVEL.INSPECTION = ABL(ID)
13  CANCEL THE B.LEVEL.INSPECTION
14  RESCHEDULE THE B.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(2)
15
16  IF EXT.INSPECTION LE 1
17  LET A.LEVEL.INSPECTION = AAL(ID)
18  CANCEL THE A.LEVEL.INSPECTION
19  RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(1)
20  LET AAL(ID) = A.LEVEL.INSPECTION
21  ALWAYS
22  ALWAYS
23  LET FIXIT.COST = C.REPAIR.COST
24
25  IF CR.CTR=6
26  CALL EXAMINE(3.14.35.563.08.35.255. YIELDING FOUND
27  JUMP AHEAD
28  OTHERWISE
29  CALL EXAMINE(3.14.35.563.14.35.266) YIELDING FOUND
30  HERE
31  IF FOUND = 1
32  LET CR.CTR = CR.CTR + 1
33  IF LINO = "NO"
34  IF CR.CTR = 1 AND JFLAG="NO" AND SMOD.PENDING(ID) = "NO"
35  PRINT 1 LINE WITH TIME.V, MOD.NO AS FOLLOWS
36  NON-EXPLORATORY DETECTION LEVEL AT ***** MODIFICATION *
37  LET JFLAG="YES"
38  ALWAYS
39  ALWAYS
40
41  IF FOUND = 1 OR FOUND = 2
42  FOR I = 1 TO 10
43  DO
44  IF HI.TIME.ACRFT(I) = ID
45  LET MRDD(I) = TIME.V
46  LEAVE
47  ELSE
48  LOOP
49  REGARDLESS
50  LET D.INT.FIND = "NO"
51  RETURN
52  END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT C-LEVEL-INSPECTION

AAL	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
ABCD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
A-LEVEL-INSPECTION	EVENT NOTICE	(1-D) INTEGER	4 REFS
B-LEVEL-INSPECTION	EVENT NOTICE	(1-D) INTEGER	3 REFS
CR-CTR	GLOBAL VARIABLE	(1-D) REAL	4 REFS
C-INTERVAL	GLOBAL VARIABLE	(1-D) REAL	1 REFS
C-LEVEL-INSPECTION	EVENT NOTICE	(1-D) REAL	3 REFS
C-REPAIR-COST	GLOBAL VARIABLE	REAL	1 REFS
D-INT-FIND	GLOBAL VARIABLE	ALPHA	1 REFS
D-LEVEL-INSPECTION	EVENT NOTICE	ALPHA	2 REFS
EXAMINE	PROCEDURE	INTEGER	2 REFS
EXT-INSPECTION	GLOBAL VARIABLE	INTEGER	2 REFS
FIXIT-COST	GLOBAL VARIABLE	REAL	1 REFS
FOUND	LOCAL RECURSIVE	REAL	5 REFS
H2-TIME-ACRFT	LOCAL RECURSIVE	REAL	1 REFS
I	LOCAL RECURSIVE	REAL	3 REFS
ID	GLOBAL VARIABLE	INTEGER	12 REFS
IDC	GIVEN ARGUMENT	INTEGER	3 REFS
JFLAG	*TEMPORARY ATTRIBUTE	INTEGER	2 REFS
LTHO	GLOBAL VARIABLE	ALPHA	1 REFS
MOD-NO	GLOBAL VARIABLE	ALPHA	1 REFS
MRDN	GLOBAL VARIABLE	INTEGER	1 REFS
SMOD-PENDING	GLOBAL VARIABLE	(1-D) REAL	1 REFS
TIME-A	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
TIME-V	TEMPORARY ATTRIBUTE	REAL	6 REFS
LABCD	SYSTEM ATTRIBUTE	REAL	2 REFS

```

1  EVENT D.LEVEL.INSPECTION(IDD)
2  DEFINE IDD AS AN INTEGER VARIABLE
3  DEFINE J AS AN INTEGER VARIABLE
4  LET D.INT.FIND = "NO"
5  LET ID = IDD
6  SCHEDULE A D.LEVEL.INSPECTION(ID) AT TIME.V + D.INTERVAL(ID)
7  LET ADL(ID) = D.LEVEL.INSPECTION
8  LET T.LAST.D(ID) = TIME.V
9  IF INSP.SCH(ID) = "YES"
10 IF PAT.INSP.LEVEL LE 3
11 LET C.LEVEL.INSPECTION = ACL(ID)
12 CANCEL THE C.LEVEL.INSPECTION
13 RESCHEDULE THE C.LEVEL.INSPECTION(ID) AT TIME.V + C.INTERVAL(ID)
14
15
16 IF EXT.INSP.LEVEL LE 2
17 LET B.LEVEL.INSPECTION = ABL(ID)
18 CANCEL THE B.LEVEL.INSPECTION
19 RESCHEDULE THE B.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(2)
20
21 IF EXT.INSP.LEVEL LE 1
22 LET A.LEVEL.INSPECTION = AAL(ID)
23 CANCEL THE A.LEVEL.INSPECTION
24 RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(1)
25 ALWAYS
26 ALWAYS
27 REGARDLESS
28 LET FIXIT.COST = D.REPAIR.COST
29 IF SAMPLING = 1
30 GO TO INT
31 ALWAYS
32 LET D.IN(ID) = D.IN(ID) + 1
33 IF D.IN(ID) GT SAMPLING
34 LET D.IN(ID) = D.IN(ID) - SAMPLING
35 ALWAYS
36 LET N = D.IN(ID)
37 #CHK# IF IC = N
38 GO TO INT
39 ALWAYS
40 IF N GT SIZE.OF.FLEET
41 JUMP AHEAD
42 OTHERWISE
43 LET N = N + SAMPLING
44 GO TO CHK
45 HERE
46 LET D.EXT(ID) = "YES"
47
48 IF CR.CTR=0 AND IFAIL="NO"
49 CALL EXAMINE(A..858..7..351..58..6..144) YIELDING FOUND
50 JUMP AHEAD
51 OTHERWISE
52 CALL EXAMINE(A..858..7..351..858..7..144) YIELDING FOUND
53 HERE

```

```

54 IF LTHO = "YES"
55 FOR I = 1 TO NOAC(LDX)
56 DO
57 IF ID = TLID(LDX,I)
58 SKIP 1 OUTPUT LINE
59 PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
EXTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. *** AT ***** HOURS
60 LEAVE
61 ELSE
62 LOOP
63 ALWAYS
64 GO TO EXT
65 PRINT* LET D.EXT(ID) = "NO"
66 IF CR.CTR=0 AND IFAIL="NO"
67 CALL EXAMINE(4..858..7..351..58..6..144) YIELDING FOUND
68 JUMP AHEAD
69 OTHERWISE
70 CALL EXAMINE(4..858..7..351..858..7..144) YIELDING FOUND
71 HERE
72 IF FOUND = 1
73 LET D.INT.FIND = "YES"
74 IF LTHO = "NO"
75 PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
CRACK FOUND ON A/C NO. *** AT ***** HOURS DURING INTERNAL D INSPECTION
76 ALWAYS
77 IF LTHO = "YES"
78 FOR I = 1 TO NOAC(LDX)
79 DO
80 IF ID = TLID(LDX,I)
81 SKIP 1 OUTPUT LINE
82 PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
INTERNAL D-LEVEL INSPECTION PERFORMED ON A/C NO. *** AT ***** HOURS
83 LEAVE
84 ELSE
85 LOOP
86 ALWAYS
87 LET CR.CTR = CR.CTR + 1
88 IF CR.CTR = "NO"
89 IF LTHO = "NO"
90 PRINT 1 LINE WITH TIME.V. MOD.NO AS FOLLOWS
NON-EXPLORATORY DETECTION LEVEL AT ***** MODIFICATION *
91 LET JFLAG="YES"
92 ALWAYS
93 ALWAYS
94 ALWAYS
95 ALWAYS
96 ALWAYS
97 ALWAYS
98 IF MN = 2
99 GO TO MOCHG
100 ELSE
101 LET I = HI.TIME.ACFT(10)
102 IF I GE ID AND TIME.V - ENTRY.TIME(AIRPLANE(I)) + 10 GE ABCD(4)

```

```

103 FOR I = 1 TO 10
104 DO
105 IF MI.TIME.ACRFT(I) = ID
106 IF FOUND = 1 OR FOUND = 2
107 LET MRGD(I) = TIME.V
108 ALWAYS
109 FOR J = 1 TO 10
110 DO
111 IF TIME.V - MPDD(J) LT ABCD(4)
112 GO TO NOCHG
113 ELSE
114 LOOP
115 IF IMP.MOD.SCH = "YES" OR TMOD.PENDING(ID) = "YES"
116 OR SMOD.PENDING(ID) = "YES" OR DEC.INT = "YES"
117 GO TO NOCHG
118 ALWAYS
119 IF ABCD(4) LE 12000.
120 LET DECC = (FREQ.DECREASE -1.) / 2. + 1.
121 LET DECD = (FREQ.DECREASE -1.) * 2. + 1.
122 LET ABCD(3) = ABCD(3) * DECC
123 LET ABCD(4) = ABCD(4) * DECD
124 LET SAMPLING = SAMPLING/DECD+1
125 JUMP AHEAD
126 ALWAYS
127 LET ABCD(3) = ABCD(3) * FREQ.DECREASE
128 LET ABCD(4) = ABCD(4) * FREQ.DECREASE
129 LET SAMPLING = SAMPLING + 1
130 IF ABCD(4) GT 32000.
131 LET ABCD(4) = 32000.
132 LET ABCD(3) = 32000. * 1ABCD(3) / 1ABCD(4)
133 ALWAYS
134 HERE
135 IF LTHO = "YES"
136 SKIP 1 OUTPUT LINE
137 PRINT 4 LINES WITH TIME.V, ABCD(3), ABCD(4), SAMPLING AS FOLLOWS
INSPECTION INTERVAL INCREASE IMPLEMENTED ***** HOURS FROM START OF SIMULATION
C-LEVEL INTERVAL NOW ***** HOURS
D-LEVEL INTERVAL NOW ***** HOURS
SAMPLING NOW **
138 ALWAYS
139 LET NICHG = NICHG + 1
140 IF NICHG GT 30
141 JUMP AHEAD
142 ELSE
143 LET CHG.TIME(NICHG) = TIME.V
144 LET SC(NICHG) = ABCD(3)
145 LET SD(NICHG) = ABCD(4)
146 LET SAMP.NICHG = SAMPLING
147 LET MOD.SAVE(NICHG) = MOD.NO
148 HERE
149 LET CINSL = ABCD(3)
150 LET DINSL = ABCD(4)
151 FOR EVERY AIRCRAFT IN ACTIVE.FLEET

```

```

152 DO
153 LET C.INTERVAL(TAIL.ID) = ABCD(3)
154 LET D.INTERVAL(TAIL.ID) = ABCD(4)
155 LET D.LEVEL.INSPECTION = ADL(TAIL.ID)
156 CANCEL THE D.LEVEL.INSPECTION
157 RESCHEDULE THE D.LEVEL.INSPECTION(TAIL.ID) AT T.LAST.D(TAIL.ID) +
158 D.INTERVAL(TAIL.ID)
159 LOOP
160 IF TIME.V LT USAGE.LIFE + BEGIN.PRODUCTION
161 FOR J = 1 TO 10
162 DO
163 LET MRD(J) = TIME.V
164 LOOP
165 ALWAYS
166 IF TIME.V + ABCD(4) GT USAGE.LIFE + BEGIN.PRODUCTION
167 LET SMALL = 100000.
168 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
169 DO
170 LET D.LEVEL.INSPECTION = ADL(TAIL.ID)
171 LET DIFF = TIME.AID.LEVEL.INSPECTION - TIME.V
172 LET DIFFER = TIME - A(D.LEVEL.INSPECTION) - ABCD(4) - ENTRY.TIME(AIRCRAFT
173 TAIL.ID)) - USAGE.LIFE
174 IF DIFF LT SMALL AND DIFFER LT 0.
175 LET SMALL = DIFF
176 LET LMTA = TAIL.ID - 1
177 ALWAYS
178 LOOP
179 IF TIME.V GT USAGE.LIFE + BEGIN.PRODUCTION
180 LET LMTA = SIZE.OF.FLEET - 10
181 LET MN = MN+1
182 ALWAYS
183 FOR I = 1 TO 10
184 DO
185 LET MI.TIME.ACRFT(I) = LMTA + I
186 LOOP
187 ALWAYS
188 ENDOF LEAVE
189 ELSE
190 LOOP
191 ALWAYS
192 IF THOD.PENDING(ID) = "YES" OR SMOD.PENDING(ID) = "YES"
193 LET C.INTERVAL(ID) = ABCD(3)
194 LET D.INTERVAL(ID) = ABCD(4)
195 LET D.LEVEL.INSPECTION = ADL(ID)
196 CANCEL THE D.LEVEL.INSPECTION
197 RESCHEDULE THE D.LEVEL.INSPECTION AT TIME.V + ABCD(4)
198 IF THOD.PENDING(ID) = "YES"
199 CALL INSTALL.MODIFICATION
200 LET THOD.PENDING(ID) = "NO"
201 IF TIME.V LT ENTRY.TIME(AIRCRAFT(ID)) + C7 + 1AAFL
202 LET T.INSPECTION.INCREASE = ATT(IID)
203 CANCEL THE T.INSPECTION.INCREASE
204

```

```

205 DESTROY THE T.INSPECTION.INCREASE
206 REGARDLESS
207 RETURN
208 OTHERWISE
209 CALL INSTALL.MODIFICATION
210 LET SMOD.PENDING(ID) = "NO"
211 RETURN
212 OTHERWISE
213 END
214 IF FOUND GT 0
215 SCHEDULE A REPAIR(.D.4) NOW
216 REGARDLESS
217 RETURN
218 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT D.LEVEL.INSPECTION

AAL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ABCD	GLOBAL VARIABLE	(1-D) REAL	27 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACTIVE.FLEET	SET		2 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	4 REFS
AIRCRAFT	TEMPORARY ENTITY		2 REFS
AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	6 REFS
ATII	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
A.LEVEL.INSPECTION	EVENT NOTICE	REAL	3 REFS
BEGIN.PRODUCTION	GLOBAL VARIABLE	(1-D) REAL	3 REFS
A.LEVEL.INSPECTION	EVENT NOTICE	(1-D) REAL	1 REFS
CHG.TIME	GLOBAL VARIABLE	(1-D) REAL	2 REFS
CHK	UNSCRIPTED LABEL		7 REFS
CINSL	GLOBAL VARIABLE	REAL	1 REFS
CR.CTR	GLOBAL VARIABLE	INTEGER	5 REFS
C7	GLOBAL VARIABLE	REAL	1 REFS
C.INTERVAL	GLOBAL VARIABLE	(1-D) REAL	3 REFS
C.LEVEL.INSPECTION	EVENT NOTICE		3 REFS
DECC	LOCAL RECURSIVE	REAL	2 REFS
DECD	LOCAL RECURSIVE	REAL	3 REFS
DEC.INT	GLOBAL VARIABLE	ALPHA	1 REFS
DIFER	LOCAL RECURSIVE	REAL	2 REFS
DIFF	LOCAL RECURSIVE	REAL	3 REFS
DINSL	GLOBAL VARIABLE	REAL	1 REFS
D.EXT	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
D.IN	GLOBAL VARIABLE	(1-D) INTEGER	6 REFS
D.INT.FIND	GLOBAL VARIABLE	(1-D) REAL	4 REFS
D.LEVEL.INSPECTION	GLOBAL VARIABLE	ALPHA	2 REFS
D.REPAIR.COST	EVENT NOTICE		12 REFS
ENTRY.TIME	GLOBAL VARIABLE	REAL	1 REFS
EXAMINE	TEMPORARY ATTRIBUTE	REAL	6 REFS
EXT	PROCEDURE	INTEGER	4 REFS
EXT.INSPECTION	UNSCRIPTED LABEL		2 REFS
FIXIT.COST	GLOBAL VARIABLE	INTEGER	3 REFS
FOUND	GLOBAL VARIABLE	REAL	1 REFS
FREQ.DECREASE	LOCAL RECURSIVE	REAL	9 REFS
HI.TIME.ACHFT	GLOBAL VARIABLE	REAL	4 REFS
I	GLOBAL VARIABLE	(1-D) INTEGER	3 REFS
ID	LOCAL RECURSIVE	REAL	13 REFS
IDO	GLOBAL VARIABLE	INTEGER	46 REFS
IFAIL	GIVEN ARGUMENT	INTEGER	3 REFS
IMP.MOD.SCH	*TEMPORARY ATTRIBUTE	INTEGER	2 REFS
INSPECTION	GLOBAL VARIABLE	ALPHA	1 REFS
INSPECTION	GLOBAL VARIABLE	ALPHA	1 REFS
INSTALL.MCDIFICATION	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
INT	PROCEDURE	INTEGER	2 REFS
J	UNSCRIPTED LABEL		3 REFS
JFLAG	LOCAL RECURSIVE	INTEGER	5 REFS
	*GLOBAL VARIABLE	ALPHA	2 REFS

LDX	GLOBAL VARIABLE	INTEGER	4 REFS
LHTA	GLOBAL VARIABLE	INTEGER	3 REFS
LTHD	GLOBAL VARIABLE	ALPHA	5 REFS
MOD.NO	GLOBAL VARIABLE	INTEGER	2 REFS
MOD.SAVE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
MRDD	GLOBAL VARIABLE	(1-D) REAL	3 REFS
N	LOCAL RECURSIVE	REAL	5 REFS
NICHG	GLOBAL VARIABLE	INTEGER	8 REFS
NN	GLOBAL VARIABLE	INTEGER	3 REFS
NOAC	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
NOCHG	UNSCRIPTED LABEL		4 REFS
REPAIR	EVENT NOTICE		1 REFS
SAMP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SAMPLING	GLOBAL VARIABLE	INTEGER	10 REFS
SC	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE	INTEGER	2 REFS
SMALL	LOCAL RECURSIVE	REAL	3 REFS
SMOD.PENDING	GLOBAL VARIABLE	ALPHA	4 REFS
TAIL.ID	TEMPORARY ATTRIBUTE	INTEGER	9 REFS
TIME.A	TEMPORARY ATTRIBUTE	REAL	2 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	21 REFS
TLID	GLOBAL VARIABLE	(2-D) INTEGER	2 REFS
TMOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	4 REFS
T.INSPECTION.INCREAS	EVENT NOTICE		3 REFS
T.LAST.D	GLOBAL VARIABLE	(1-D) REAL	2 REFS
USAGE.LIFE	GLOBAL VARIABLE	REAL	4 REFS
IAAFL	GLOBAL VARIABLE	REAL	1 REFS
IABCD	GLOBAL VARIABLE	(1-D) REAL	4 REFS

```

1 ROUTINE EXAMINE(M,XA,YA,ZA,XL,YL,ZL) YIELDING FOUND
2 DEFINE N AS AN INTEGER VARIABLE
3 DEFINE Z AS AN ALPHA VARIABLE
4 LET TAC = 0.0
5 IF LTHO = "YES"
6   FOR I = 1 TO NOAC(LDX)
7     DO
8   IF IO = TLIO(LDX,I)
9     LET LIST = 1.0
10    LEAVE
11  ELSE
12    LOOP
13  ALWAYS
14    LET M1 = MSRI(ID)
15    LET M2 = MFR1(ID)
16    LET M3 = MSK2(ID)
17    LET M4 = MFR2(ID)
18    LET CCL1 = CONE
19    LET CCL2 = CTNO
20    LET CCL3 = CTHREL
21  IF CO.EXISTS(ID) = "YES"
22    IF C.INT(ID) = "NO" OR C.INT(ID) = "YES" AND N EQ 4 AND O.EXT(ID) = "NO"
23      LET COROSION = AC(ID)
24      LET TAC = TIME.A(COROSION)
25      LET AREA = C.GROWTH.PATE * CRRF * (TIME.V - TIME.A(COROSION))
26      CALL PGOO(XA,YA,ZA,AREA) YIELDING PL
27      IF RANDOM.F(1) LE PL
28        LET OI(O(N)) = OI(O(N)) + 1
29        LET GOICO(N) = GOICO(N) + 1
30        IF N = 1
31          LET ACA = AREA
32          LET GACA = AREA
33          LET Z = "A"
34          JUMP AHEAD
35          IF N = 2
36            LET BCA = AREA
37            LET GBCA = AREA
38            LET Z = "B"
39            JUMP AHEAD
40            IF N = 3
41              LET CCA = AREA
42              LET GCCA = AREA
43              LET Z = "C"
44              JUMP AHEAD
45              LET DCA = AREA
46              LET GDCA = AREA
47              LET Z = "D"
48              JUMP AHEAD
49              IF LIST = 1.0
50                SKIP 1 OUTPUT LINE
51                PRINT 1 DOUBLE LINE WITH AREA, Z, IO, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
52                FOLLOWS
53                CORROSION OF AREA **.* SQ. INCHES DETECTED DURING **-LEVEL INSPECTION OF A/C NO. *** AT ***** FLIGHT HOURS
54                ALWAYS
55                LET FOUND = Z
56                REGARDLESS

```

```

53 REGARDLESS
54 REGARDLESS
55 **
56 IF 1.CP.EXISTS(ID) = "YES"
57 IF 1.INT(ID) = "YES" AND D.EXT(ID) = "YES" AND N = 4
58 GO TO SKIP1
59 ALWAYS
60 IF 1.INT(ID) = "NO" OR 1.INT(ID) = "YES" AND ITRNL LE N
61 LET 1.STRENGTH.REDUCTION = AISR(ID)
62 LET TAI = TIME.A(1).STRENGTH.REDUCTION)
63 LET CL = (TIME.V-TAI) * M1 * CGRI(ID)
64 IF TAC GT TAI
65 LET CL = (TAC-TAI)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
66 REGARDLESS
67 IF CL GT CCL1
68 LET CL = CCL1 + ((CL-CCL1)/M1)*M2
69 REGARDLESS
70 IF CL GT CCL2
71 LET CL = CCL2 + ((CL-CCL2)/M2)*M3
72 REGARDLESS
73 IF CL GT CCL3
74 LET CL = CCL3 + ((CL-CCL3)/M3)*M4
75 REGARDLESS
76
77 IF ITRNL GT N
78 LET CL = CL - INT.LGT(ID)
79 ALWAYS
80 IF 1.INT(ID) = "NO" AND D.EXT(ID) = "YES" AND N = 4
81 LET CL = CL - INT.LGT(ID)
82 ALWAYS
83 IF N = 1
84 IF CL LE 4.0
85 JUMP AHEAD
86 ALWAYS
87 IF CL GT 4.0 AND CL LE 36.0
88 LET PL = .00502 + .00097 * CL
89 GO TO CHK1
90 OTHERWISE
91 LET PL = .04
92 GO TO CHK1
93 ALWAYS
94 IF N = 2
95 IF CL LE 8.0
96 JUMP AHEAD
97 ALWAYS
98 IF CL GT 8.0 AND CL LE 36.0
99 LET PL = .03504 + .00097 * CL
100 GO TO CHK1
101 OTHERWISE
102 LET PL = .07
103 GO TO CHK1
104 ALWAYS
105 IF N = 3

```

```

106 IF CL LE 5.0
107 JUMP AHEAD
108 ALWAYS
109 IF CL GT 5.0 AND CL LE 36.0
110 IF CR.CTR=0
111 LET PL = .0267 + .0076 * CL
112 GO TO CHK1
113 OTHERWISE
114 LET PL = .06055 + .01055 * CL
115 GO TO CHK1
116 OTHERWISE
117 IF CR.CTR=0
118 LET PL = .3
119 GO TO CHK1
120 OTHERWISE
121 LET PL = .44
122 GO TO CHK1
123 ALWAYS
124 IF N = 4
125 IF CR.CTR=0 AND IFAIL="NO"
126 IF CL LE 6.0
127 JUMP AHEAD
128 OTHERWISE
129 IF CL GT 6.0 AND CL LE 32.0
130 LET PL = .4964 + .01105 * CL
131 GO TO CHK1
132 OTHERWISE
133 LET PL = .85
134 GO TO CHK1
135 ELSE
136 IF CL LE 5.0
137 JUMP AHEAD
138 OTHERWISE
139 IF CL GT 5.0 AND CL LE 12.5
140 LET PL = .7172 + .0223 * CL
141 GO TO CHK1
142 OTHERWISE
143 LET PL = .999
144 GO TO CHK1
145 ALWAYS
146 HERE
147 CALL PDD(XL, YL, ZL, CL) YIELDING PL
148 WCHK1 IF RANDOM.F(1) LE PL
149 IF 1.INT(ID) = "NO" AND D.EXT(ID) = "YES" AND N = 4
150 LET CL = CL + INT.LGT(ID)
151 ALWAYS
152 IF ITNHL GT N
153 LET CL = CL + INT.LGT(ID)
154 ALWAYS
155 LET OICR(N) = OICR(N) + 1
156 LET GOICR(N) = GOICR(N) + 1
157 IF N = 1
158 LET ACRKL = CL
159 LET GACRKL = CL

```

```

159      LET Z = "A"
160      JUMP_AHEAD
161      IF N = 2
162      LET BCRKL = CL
163      LET Z = "B"
164      JUMP_AHEAD
165      LET CCRKL = CL
166      LET Z = "C"
167      JUMP_AHEAD
168      LET DCRKL = CL
169      LET Z = "D"
170      JUMP_AHEAD
171      LET Z = "D"
172      HERE
173      IF LIST = 1.0
174      SKIP 1 OUTPUT LINE
175      PRINT 1 DOUBLE LINE WITH CL, Z, ID, TIME, V-ENTRY, TIME(AIRPLANE(ID)) AS
176      FOLLOWS
177      CRACK OF LENGTH ** INCHES DETECTED DURING ** LEVEL INSPECTION OF A/C NO. *** AT ***** FLIGHT HOURS
178      ALWAYS
179      LET FOUND = 1
180      REGARDLESS
181      ALWAYS
182      00
183      00SKIP IF 2.CP.EXISTS(ID) = "YES"
184      IF 2.INT(ID) = "YES" AND D.EXT(ID) = "YES" AND N = 4
185      GO TO SKIP2
186      ALWAYS
187      IF 2.INT(ID) = "NO" OR 2.INT(ID) = "YES" AND [THNL LE N
188      LET 2.STRENGTH-REDUCTION = ASR(ID)
189      LET TAZ = TIME.A(2.STRENGTH-REDUCTION)
190      LET CL = (TIME.V-TAZ) * M1 * CGR1(ID)
191      IF TAC GT TAZ
192      LET CL = (TAC-TAZ)*M1 + (TIME.V-TAC)*M1*CGR1(ID)
193      REGARDLESS
194      IF CL GT CCL1
195      LET CL = CCL1 + ((CL-CCL1)/M1)*M2
196      REGARDLESS
197      IF CL GT CCL2
198      LET CL = CCL2 + ((CL-CCL2)/M2)*M3
199      REGARDLESS
200      IF CL GT CCL3
201      LET CL = CCL3 + ((CL-CCL3)/M3)*M4
202      REGARDLESS
203      IF ITRNL GT N
204      LET CL = CL - INT.LGT(ID)
205      ALWAYS
206      00
207      IF 2.INT(ID) = "NO" AND D.EXT(ID) = "YES" AND N = 4
208      LET CL = CL - INT.LGT(ID)
209      ALWAYS
210      IF N = 1
211      IF CL LE 4.9

```

```

211 JUMP AHEAD
212 ALWAYS
213 IF CL GT 4.0 AND CL LE 36.0
214 LET PL = .00502 + .00097 * CL
215 GO TO CHK2
216 OTHERWISE
217 LET PL = .04
218 GO TO CHK2
219 ALWAYS
220 IF N = 2
221 IF CL LE 8.0
222 JUMP AHEAD
223 ALWAYS
224 IF CL GT 8.0 AND CL LE 36.0
225 LET PL = .03504 + .00097 * CL
226 GO TO CHK2
227 OTHERWISE
228 LET PL = .07
229 GO TO CHK2
230 ALWAYS
231 IF N = 3
232 IF CL LE 5.0
233 JUMP AHEAD
234 ALWAYS
235 IF CL GT 5.0 AND CL LE 36.0
236 IF CR.CTR=0
237 LET PL = .0267 + .0076 * CL
238 GO TO CHK2
239 OTHERWISE
240 LET PL = .06055 + .01055 * CL
241 GO TO CHK2
242 OTHERWISE
243 IF CR.CTR=0
244 LET PL = .3
245 GO TO CHK2
246 OTHERWISE
247 LET PL = .44
248 GO TO CHK2
249 ALWAYS
250 IF N = 4
251 IF CR.CTR=0 AND IFAIL="NO"
252 IF CL LE 6.0
253 JUMP AHEAD
254 OTHERWISE
255 IF CL GT 6.0 AND CL LE 32.0
256 LET PL = .4984 + .01105 * CL
257 GO TO CHK2
258 OTHERWISE
259 LET PL = .85
260 GO TO CHK2
261 ELSE
262 IF CL LE 5.0
263 JUMP AHEAD

```

```

264 OTHERWISE
265 IF CL GT 5.0 AND CL LE 12.5
266 LET PL = .7172 + .0223 * CL
267 GO TO CHK2
268 OTHERWISE
269 LET PL = .999
270 GO TO CHK2
271 ALWAYS
272 HERE
273 CALL PDDO(XL, YL, ZL, CL) YIELDING PL
274 IF RANDOM.F(1) LE PL
275 IF 2.INT(ID) = "NO" AND D.EXT(ID) = "YES" AND N = 4
276 LET CL = CL + INT.LGT(ID)
277 ALWAYS
278 IF ITRNL GT N
279 LET CL = CL + INT.LGT(ID)
280 ALWAYS
281 LET OICR(N) = OICR(N) + 1
282 LET GOICR(N) = GOICR(N) + 1
283 IF N = 1
284 LET ACCKL = CL
285 LET GACCKL = CL
286 LET Z = "N"
287 JUMP AHEAD
288 LET RCKL = CL
289 LET GRCKL = CL
290 LET Z = "B"
291 JUMP AHEAD
292 LET CCKL = CL
293 LET GCCKL = CL
294 LET Z = "C"
295 JUMP AHEAD
296 LET DCKL = CL
297 LET GDCKL = CL
298 LET Z = "D"
299 HERE
300 IF LIST = 1.0
301 SKIP 1 OUTPUT LINE
302 PRINT 1 DOUBLE LINE WITH CL, Z, ID, TIME, V-ENTRY, TIME(AIRPLANE(ID)) AS
FOLLOWS
303 CPACK OF LENGTH ** ** INCHES DETECTED DURING **-LEVEL INSPECTION OF A/C NO. *** AT ***** FLIGHT HOURS
304 ALWAYS
305 LET FOUND = 1
306 REGARDLESS
307 ALWAYS
308 ALWAYS
309 **
310 BSKIP2B IF N = 4 RETURN OTHERWISE
311 **
312 IF FOUND GT 0
313 SCHEDULE A REPAIR(ID,N) NOW
314 REGARDLESS
315 RETURN

```

—CDC 6600 .CACI-SIMSRIPT II.5-VERSION /4.0-00/ KRONOS 2.1.2
316 END

03/02/78. 18.36.02. - PAGE 132

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE EXAMINE

AC	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
ACA	GLOBAL VARIABLE		REAL	1 REFS
ACRKL	GLOBAL VARIABLE		REAL	2 REFS
AIRPLANE	GLOBAL VARIABLE		INTEGER	3 REFS
AREA	LOCAL RECURSIVE	< 20	REAL	11 REFS
AISP	GLOBAL VARIABLE		(1-D) INTEGER	1 REFS
AZSR	GLOBAL VARIABLE		(1-D) INTEGER	1 REFS
BCA	GLOBAL VARIABLE		REAL	1 REFS
BCKKL	GLOBAL VARIABLE		REAL	2 REFS
CCA	GLOBAL VARIABLE		REAL	1 REFS
CCL1	LOCAL RECURSIVE	< 17	REAL	7 REFS
CCL2	LOCAL RECURSIVE	< 18	REAL	7 REFS
CCL3	LOCAL RECURSIVE	< 19	REAL	7 REFS
CCRKL	GLOBAL VARIABLE		REAL	2 REFS
CGRI	GLOBAL VARIABLE		(1-D) REAL	4 REFS
CHK1	UNSUBSCRIPTED LABEL			13 REFS
CHK2	UNSUBSCRIPTED LABEL			13 REFS
CL	LOCAL RECURSIVE	< 34	REAL	100 REFS
CONE	GLOBAL VARIABLE		REAL	1 REFS
CORUSION	EVENT NOTICE		REAL	3 REFS
ED.EXISTS	GLOBAL VARIABLE		(1-D) ALPHA	1 REFS
CRF	GLOBAL VARIABLE		REAL	1 REFS
CR.CTR	GLOBAL VARIABLE		INTEGER	6 REFS
CTHREE	GLOBAL VARIABLE		REAL	1 REFS
CTWO	GLOBAL VARIABLE		REAL	1 REFS
C.GROWTH.RATE	GLOBAL VARIABLE		REAL	1 REFS
C.INT	GLOBAL VARIABLE		(1-D) ALPHA	2 REFS
DCA	GLOBAL VARIABLE		REAL	1 REFS
DCRKL	GLOBAL VARIABLE		REAL	2 REFS
D.EXT	GLOBAL VARIABLE		(1-D) ALPHA	7 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE		REAL	3 REFS
EXAMINE	PROCEDURE		INTEGER	1 REFS
FOUND	YIELDED ARGUMENT	< 8	REAL	5 REFS
GACA	GLOBAL VARIABLE		REAL	1 REFS
GACRK	GLOBAL VARIABLE		REAL	2 REFS
GBCA	GLOBAL VARIABLE		REAL	1 REFS
GRCKK	GLOBAL VARIABLE		REAL	2 REFS
GCCA	GLOBAL VARIABLE		REAL	1 REFS
SCCRK	GLOBAL VARIABLE		REAL	2 REFS
SDCA	GLOBAL VARIABLE		REAL	1 REFS
SDCRK	GLOBAL VARIABLE		REAL	2 REFS
SGICO	GLOBAL VARIABLE		REAL	2 REFS
SGICR	GLOBAL VARIABLE		(1-D) INTEGER	2 REFS
I	LOCAL RECURSIVE	< 11	(1-D) INTEGER	4 REFS
ID	GLOBAL VARIABLE		REAL	2 REFS
IFAIL	GLOBAL VARIABLE		INTEGER	49 REFS
INT.LGT	GLOBAL VARIABLE		ALPHA	2 REFS
IXPNL	GLOBAL VARIABLE		(1-D) REAL	8 REFS
LDA	GLOBAL VARIABLE		INTEGER	6 REFS
LIST	LOCAL RECURSIVE	< 12	INTEGER	2 REFS
LTHQ	GLOBAL VARIABLE		REAL	4 REFS
			ALPHA	1 REFS

1225

```

NEP1  GLOBAL VARIABLE ( 1-D) REAL 1 REFS
NEP2  GLOBAL VARIABLE ( 1-D) REAL 1 REFS
MSR1  GLOBAL VARIABLE ( 1-D) REAL 1 REFS
MSR2  GLOBAL VARIABLE ( 1-D) REAL 1 REFS
M1    LOCAL RECURSIVE < 13 REAL 9 REFS
M2    LOCAL RECURSIVE < 14 REAL 5 REFS
M3    LOCAL RECURSIVE < 15 REAL 5 REFS
M4    LOCAL RECURSIVE < 16 REAL 3 REFS
N      GIVEN ARGUMENT < 1 INTEGER 46 REFS
NOAC   GLOBAL VARIABLE ( 1-D) INTEGER 1 REFS
OICO   GLOBAL VARIABLE ( 1-D) INTEGER 2 REFS
OICR   GLOBAL VARIABLE ( 1-D) INTEGER 4 REFS
PL      LOCAL RECURSIVE < 21 REAL 30 REFS
PODD    PROCEDURE INTEGER 3 REFS
RANDOM-F REPAIR 1 REFS
REPAIR SKIP1 2 REFS
SKIP2  UNSUBSCRIPTED LABEL 1 REFS
TAC     LOCAL RECURSIVE < 10 REAL 2 REFS
TAL     LOCAL RECURSIVE < 33 REAL 8 REFS
TAR     LOCAL RECURSIVE < 36 REAL 4 REFS
TIME-A  TEMPORARY ATTRIBUTE REAL 4 REFS
TIME-V  SYSTEM ATTRIBUTE REAL 4 REFS
TLIN    GLOBAL VARIABLE ( 2-D) INTEGER 8 REFS
XA      GIVEN ARGUMENT < 2 REAL 1 REFS
XL      GIVEN ARGUMENT < 5 REAL 2 REFS
YA      GIVEN ARGUMENT < 3 REAL 3 REFS
YL      GIVEN ARGUMENT < 6 REAL 2 REFS
Z       LOCAL RECURSIVE < 9 REAL 16 REFS
ZA      GIVEN ARGUMENT < 4 REAL 2 REFS
ZL      GIVEN ARGUMENT < 7 REAL 2 REFS
1-CR-EXISTS 1-CR-EXISTS 1 REFS
1-INT 1-INT 1 REFS
1-STRENGTH-REDUCTION 1-STRENGTH-REDUCTION 5 REFS
2-CR-EXISTS 2-CR-EXISTS 2 REFS
2-INT 2-INT 1 REFS
2-STRENGTH-REDUCTION 2-STRENGTH-REDUCTION 5 REFS
EVENT NOTICE GLOBAL VARIABLE ( 1-D) ALPHA 2 REFS
EVENT NOTICE GLOBAL VARIABLE ( 1-D) ALPHA 2 REFS

```

```

1 ROUTINE P00DIX,Y,Z,L) YIELDING PL
2 **
3 ** COMPUTES PROBABILITY OF DETECTING A SERVICE DEFECT OF A GIVEN SIZE
4 ** X = MAXIMUM PROBABILITY OF DETECTION AT A GIVEN INSPECTION LEVEL
5 ** Y = EQUATION PARAMETER - CONSTANT FOR EACH LEVEL OF INSPECTION
6 ** Z = MINIMUM SIZE OF DEFECT DETECTABLE AT EACH INSPECTION LEVEL
7 ** L = SIZE OF DEFECT BEING TESTED:
8 **     LENGTH FOR CRACK DEFECTS
9 **     AREA FOR CORROSION DEFECTS
10 **
11 LET PL = X * (1.0 - EXP.F(-Y * (L - Z)))
12 RETURN
13 END

```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE P0DD

EXP.F	PROCEDURE					
L	GIVEN ARGUMENT	< 4	REAL	1 REFS		
PL	YIELDED ARGUMENT	< 5	REAL	2 REFS		
P0DD	PROCEDURE		INTEGER	1 REFS		
X	GIVEN ARGUMENT	< 1	REAL	2 REFS		
Y	GIVEN ARGUMENT	< 2	REAL	2 REFS		
Z	GIVEN ARGUMENT	< 3	REAL	2 REFS		

```

1 ROUTINE CANCEL.SCHEDULED.INSPECTIONS
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```

00 CANCELS ALL SCHEDULED INSPECTIONS BELOW OVERHAUL LEVEL ON A GIVEN AIRCRAFT
 IF EXT. INSP. LEVEL LE 3
 LET C. LEVEL. INSPECTION = AQL (ID)
 CANCEL THE C. LEVEL. INSPECTION
 DESTROY THE C. LEVEL. INSPECTION
 00
 IF EXT. INSP. LEVEL LE 2
 LET B. LEVEL. INSPECTION = ABL (ID)
 CANCEL THE B. LEVEL. INSPECTION
 DESTROY THE B. LEVEL. INSPECTION
 00
 IF EXT. INSP. LEVEL LE 1
 LET A. LEVEL. INSPECTION = AAL (ID)
 CANCEL THE A. LEVEL. INSPECTION
 DESTROY THE A. LEVEL. INSPECTION
 ALWAYS
 ALWAYS
 00
 LET INSP. SCH (ID) = "NO"
 RETURN
 END

SYMBOLIC REFERENCE MAP (= 1) - ROUTINE CANCEL.SCHEDULED.INS

AAL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ABL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
A.LEVEL.INSPECTION	EVENT NOTICE		3 REFS
B.LEVEL.INSPECTION	EVENT NOTICE		3 REFS
CANCEL.SCHEDULED.INS	PROCEDURE	INTEGER	1 REFS
EXT.INSPECTION	EVENT NOTICE		3 REFS
EXT.INSPECTION	GLOBAL VARIABLE	INTEGER	3 REFS
ID	GLOBAL VARIABLE	INTEGER	4 REFS
INSP.SCH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS

```

1  EVENT REACH.FAIL.SAFE.LGT(IDRFS)
2  DEFINE IDRFS AS AN INTEGER VARIABLE
3  LET ID = IDRFS
4  CALL COMP.RISK YIELDING CL
5  LET SDIF(ID) = SF
6  LET RLGT(ID) = LGHT.TO.FAILURE-FSAF.LGT
7  LET SRRATE(ID) = SRRATE(ID)*(FSAF.LGT/(SU-SF))*(SDIF(ID)/R_GT(ID))
8  LET TRCHG(ID) = TIME.V
9  IF LTIMO = "YES"
10  FOR I = 1 TO NOAC(LDX)
11  DO
12  IF ID = TLID(LDX,I)
13  SKIP 1 OUTPUT LINE
14  PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
15  A/C NO. *** REACHES FAIL-SAFE STRENGTH AT ***** FLIGHT HOURS
16  LEAVE
17  ELSE
18  LOOP
19  ALWAYS
20  IF NRFS LT 20
21  LET NRFS = NRFS + 1
22  LET APID(NRFS) = ID
23  LET STIM(NRFS) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
24  REGARDLESS
25  IF SNRFS LT 200
26  LET SNRFS = SNRFS + 1
27  LET SAPID(SNRFS) = ID
28  LET SSTIM(SNRFS) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
29  LET SELMB(SNRFS) = ELEMENT(4)
30  REGARDLESS
31  LET AIL(ID) = "NO"
32  RETURN
33  END

```

- 231 -

SYMBOLIC REFERENCE MAP (R = 1) - EVENT REACH.FAIL.SAFE.LGT

					STRAY NAME
AIL	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS	
AIRPLANE	GLOBAL VARIABLE	(1-D)	INTEGER	3 REFS	
APID	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS	
CL	LOCAL RECURSIVE	< 2	REAL	1 REFS	
COMP-RISK	PROCEDURE		INTEGER	1 REFS	
ELEMENT	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS	
ENTRY.TIME	TEMPORARY ATTRIBUTE		MEAL	3 REFS	
FSAF.LG1	GLOBAL VARIABLE		REAL	2 REFS	
I	LOCAL RECURSIVE	< 3	REAL	2 REFS	
ID	GLOBAL VARIABLE		INTEGER	1 REFS	
IDRFS	GIVEN ARGUMENT	< 1	INTEGER	16 REFS	
	+TEMPORARY ATTRIBUTE		INTEGER	3 REFS	
LIX	GLOBAL VARIABLE		INTEGER	2 REFS	
LIGHT.TO-FAILURE	GLOBAL VARIABLE		DEAL	1 REFS	
LTHO	GLOBAL VARIABLE		ALPHA	1 REFS	
MOC	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS	
NRRFS	GLOBAL VARIABLE		INTEGER	5 REFS	
REACH.FAIL-SAFE.LG1	EVENT NOTICE		INTEGER	1 REFS	
RGLT	GLOBAL VARIABLE	(1-D)	REAL	2 REFS	
SAPID	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS	
SOIF	GLOBAL VARIABLE	(1-D)	PEAL	2 REFS	
SELMB	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS	
SF	GLOBAL VARIABLE		REAL	2 REFS	
SNRFS	GLOBAL VARIABLE		INTEGER	6 REFS	
SRARTE	GLOBAL VARIABLE	(1-D)	REAL	2 REFS	
SSTIM	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS	
STIM	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS	
SU	GLOBAL VARIABLE		HEAL	1 REFS	
TIME.V	SYSTEM ATTRIBUTE		REAL	4 REFS	
TLLD	GLOBAL VARIABLE	(2-D)	INTEGER	1 REFS	
TRCMG	GLOBAL VARIABLE	(1-D)	REAL	1 REFS	

```

1  EVENT FAILURE(IDFA)
2  DEFINE IDEA AS AN INTEGER VARIABLE
3  DEFINE HOLD AS AN INTEGER VARIABLE
4  LET ID = IDFA
5  LET IFAIL = "YES"
6  LET NO_FAIL = NO_FAIL+1
7  LET E1(NO_FAIL) = ELEMENT(1)
8  LET E2(NO_FAIL) = ELEMENT(2)
9  LET E3(NO_FAIL) = ELEMENT(3)
10 LET E4(NO_FAIL) = ELEMENT(4)
11 IF 1.CR.EXISTS(ID) = "YES"
12 CALL COMP-RISK YIELDING CL
13 LET NCZ = NCZ+1
14 LET CL6T(NCZ) = CL
15 LET HZ(NCZ) = HZRD(ID)
16 LET PFID(NCZ) = ID
17 LET PFTIM(NCZ) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
18 ALWAYS
19 IF LTHO = "YES"
20 FOR I = 1 TO NOAC(LDX)
21 DO
22 IF ID = TLID(LDX,I)
23 SKIP 1 OUTPUT LINE
24 PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
25 A/C NO. *** EXPERIENCES ELEMENT FAILURE AT ***** FLIGHT HOURS
26 LET 1.STRENGTH.REDUCTION = AISR(ID)
27 LET TAI = TIME.A(1.STRENGTH.REDUCTION)
28 LET CCL1 = CONE
29 LET CCL2 = CTWO
30 LET CCL3 = CTHREE
31 IF CO.EXISTS(ID) = "YES"
32 LET COROSION = AC(ID)
33 LET TAC = TIME.A(COROSION)
34 ALWAYS
35 LET 1CL = (TIME.V-TAI)*MSR1(ID)*CBRI(ID)
36 LET 1CL = (TAC-TAI)*MSR1(ID) + (TIME.V-TAC)*MSR1(ID)*CBRI(ID)
37 ALWAYS
38 IF 1CL GT CCL1
39 LET 1CL = CCL1 + (1CL-CCL1)*(MFR1(ID)/MSR1(ID))
40 ALWAYS
41 IF 1CL GT CCL2
42 LET 1CL = CCL2 + (1CL-CCL2)*(MSR2(ID)/MFR1(ID))
43 ALWAYS
44 IF 1CL GT CCL3
45 LET 1CL = CCL3 + (1CL-CCL3)*(MFR2(ID)/MSR2(ID))
46 ALWAYS
47 IF 2.CR.EXISTS(ID) = "YES"
48 LET 2.STRENGTH.REDUCTION = AZSR(ID)
49 LET TAZ = TIME.A(2.STRENGTH.REDUCTION)
50 LET 2CL = (TIME.V-TAZ)*MSR1(ID)*CBRI(ID)
51 IF TAC GT TAZ
52 LET 2CL = (TAC-TAZ)*MSR1(ID) + (TIME.V-TAC)*MSR1(ID)*CBRI(ID)

```

```

53 ALWAYS
54 IF ZCL GT CCL1
55 LET ZCL = CCL1 + (ZCL-CCL1)*(WFR1(ID)/WFR1(ID))
56 ALWAYS
57 IF ZCL GT CCL2
58 LET ZCL = CCL2 + (ZCL-CCL2)*(WFR2(ID)/WFR1(ID))
59 ALWAYS
60 IF ZCL GT CCL3
61 LET ZCL = CCL3 + (ZCL - CCL3)*(WFR2(ID)/WFR2(ID))
62 ALWAYS
63 ALWAYS
64 LET CL = ICL*ZCL
65 LET S = S0 - CL*(SU-SF/FSAF.LGT)
66 IF CL GT FSAF.LGT
67 LET S = (LGT.TO.FAILURE-CL)/(LGT.TO.FAILURE-FSAF.LGT)
68 ALWAYS
69 PRINT 2 LINES WITH CL, S/SU AS FOLLOWS
    SUM OF CRACK LENGTHS AT FAILURE = **.** INCHES
    RESIDUAL STRENGTH AT FAILURE = **.** ULTIMATE
70 LEAVE
71 ELSE
72 LOOP
73 ALWAYS
74 SCHEDULE AN INCREASE INSPECTION.FREQUENCY NO*
75 SCHEDULE AN IMMEDIATE.FLEET.INSPECTION NO*
76 LET FSH(ID) = "NO"
77 IF WSP LT 10
78 LET WSP = WSP + 1
79 LET ACID(WSP) = ID
80 LET FLTHR(WSP) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
81 REGARDLESS
82 IF WSP LT 100
83 LET WSP = WSP + 1
84 LET SACID(WSP) = ID
85 LET SFLTHR(WSP) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
86 LET STANT(WSP) = ELEMENT(4)
87 REGARDLESS
88 FOR I = 1 TO 10
89 DO
90 IF WSP.ACFT(I) = ID
91 ADD 1 TO LHTA
92 HEPE
93 IF LHTA GT SIZE.OF.FLEET
94 JUMP AHEAD
95 ELSE
96 FOR EVERY AIRCRAFT IN CRASHED.FLEET
97 DO
98 IF LHTA = TAIL.ID
99 ADD 1 TO LHTA
100 JUMP BACK
101 OTHERWISE
102 LOOP
103

```

```

104 HERE
105 LET HOLD = HI.TIME.ACRFT(10)
106 LET HI.TIME.ACRFT(10) = LMTA
107 IF I NE 10
108 LET HI.TIME.ACRFT(1) = HOLD
109 REGARDLESS
110 LEAVE
111 OTHERWISE
112 LOOP
113 IF SMOD.PENDING(ID) = "YES"
114 LET BEEN.MODIFIED = BEEN.MODIFIED - 1
115 ALWAYS
116 IF SMOD.PENDING(ID) = "NO"
117 IF OCCUR.MOD(ID) GT ENTRY.TIME(AIRPLANE(ID))
118 LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - OCCUR.MOD(ID)
119 JUMP AHEAD
120 OTHERWISE
121 LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - ENTRY.TIME(AIRPLANE(ID))
122 HERE
123 REGARDLESS
124 REMOVE AIRPLANE(ID) FROM ACTIVE.FLEET
125 FILE AIRPLANE(ID) IN CRASHED.FLEET
126 ADD 1 TO 2.NUM.OF.CRASH
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

```

IF IE1(ID) = "YES"
 LET 1.ITE = AIE(ID)
 CANCEL THE 1.ITE
 DESTROY THE 1.ITE
 IF IE2(ID) = "YES"
 LET 2.ITE = AIE(ID)
 CANCEL THE 2.ITE
 DESTROY THE 2.ITE
 ALWAYS
 ALWAYS
 IF CO.EXISTS(ID) NE "NN"
 LET CORSION = AC(ID)
 IF CO.EXISTS(ID) = "NS"
 CANCEL THE CORSION
 ALWAYS
 DESTROY THE CORSION
 LET CO.EXISTS(ID) = "NN"
 REGARDLESS
 IF 1.CR.EXISTS(ID) NE "NN"
 LET 1.STRENGTH.REDUCTION = AISR(ID)
 IF 1.CR.EXISTS(ID) = "NS"
 CANCEL THE 1.STRENGTH.REDUCTION
 ALWAYS
 DESTROY THE 1.STRENGTH.REDUCTION
 LET 1.CR.EXISTS(ID) = "NN"
 IF 2.CR.EXISTS(ID) NE "NN"
 LET 2.STRENGTH.REDUCTION = AISR(ID)

```

157 IF 2.CR.EXISTS(ID) = "NS"
158 CANCEL THE 2.STRENGTH.REDUCTION
159 ALWAYS
160 DESTROY THE 2.STRENGTH.REDUCTION
161 LET 2.CR.EXISTS(ID) = "NN"
162 REGARDLESS
163 REGARDLESS
164 IF AIL(ID) = "YES"
165 LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
166 CANCEL THE REACH.FAIL.SAFE.LGT
167 DESTROY THE REACH.FAIL.SAFE.LGT
168 LET AIL(ID) = "NO"
169 REGARDLESS
170 LET D.LEVEL.INSPECTION = ADL(ID)
171 CANCEL THE D.LEVEL.INSPECTION
172 DESTROY THE D.LEVEL.INSPECTION
173 IF INSP.SCH(ID) = "YES"
174 CALL CANCEL.SCHEDULED.INSPECTIONS
175 REGARDLESS
176
177 FOR EACH RETIRE.FROM.SERVICE IN EV.S(I).RETIRE.FROM.SERVICE)
178 DO
179 IF IDRET = ID CANCEL THE RETIRE.FROM.SERVICE
180 DESTROY THE RETIRE.FROM.SERVICE
181 LEAVE ELSE
182 LOOP
183 FOR EACH IN.SERVICE.DAMAGE IN EV.S(I).IN.SERVICE.DAMAGE)
184 DO
185 IF IDSDM = ID CANCEL THE IN.SERVICE.DAMAGE
186 DESTROY THE IN.SERVICE.DAMAGE
187 LEAVE ELSE
188 LOOP
189 FOR EACH T.INSPECTION.INCREASE IN EV.S(I).T.INSPECTION.INCREASE)
190 DO
191 IF IDTI = ID CANCEL THE T.INSPECTION.INCREASE
192 DESTROY THE T.INSPECTION.INCREASE
193 LEAVE ELSE
194 LOOP
195 RETURN
196 END

```

SYMBOLIC REFERENCE MAP (R = I) - EVENT FAILURE

AC	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
ACID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACTIVE.FLEET	SET	(1-D) INTEGER	1 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AIL	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
AIRCRAFT	TEMPORARY ENTITY	REAL	1 REFS
AIRFRAME.TIME	GLOBAL VARIABLE	(1-D) INTEGER	4 REFS
AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	8 REFS
ARFSL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AIE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AISR	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
AZE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZSR	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
REEM.MODIFIED	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
CANCEL.SCHEDULED.INS	PROCEDURE	INTEGER	1 REFS
CCL1	LOCAL RECURSIVE	< 17	7 REFS
CCL2	LOCAL RECURSIVE	< 18	7 REFS
CCL3	LOCAL RECURSIVE	< 19	7 REFS
CGRT	GLOBAL VARIABLE	(1-D) REAL	4 REFS
CL	LOCAL RECURSIVE	< 3	7 REFS
CLGT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
COMP.RISK	PROCEDURE	INTEGER	1 REFS
CONE	GLOBAL VARIABLE	REAL	1 REFS
CORROSION	EVENT NOTICE	REAL	1 REFS
CG.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	5 REFS
CRASHED.FLEET	SET	(1-D) ALPHA	4 REFS
CTHREE	GLOBAL VARIABLE	REAL	2 REFS
CTWO	GLOBAL VARIABLE	REAL	1 REFS
D.LEVEL.INSPECTION	EVENT NOTICE	REAL	1 REFS
ELEMENT	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	REAL	5 REFS
EV.5	SET	REAL	6 REFS
E1	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS
E2	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
E3	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
E4	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
FAILURE	EVENT NOTICE	(1-D) INTEGER	1 REFS
FLTHP	GLOBAL VARIABLE	REAL	1 REFS
FSAF.LGT	GLOBAL VARIABLE	REAL	3 REFS
FSH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
HI.TIME.ACRAFT	GLOBAL VARIABLE	(1-D) INTEGER	4 REFS
HOLD	LOCAL RECURSIVE	INTEGER	3 REFS
HZ	GLOBAL VARIABLE	(1-D) REAL	1 REFS
HZRD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
I	GLOBAL VARIABLE	REAL	6 REFS
ID	LOCAL RECURSIVE	REAL	73 REFS
IDFA	GIVEN ARGUMENT	INTEGER	3 REFS
IDRET	*TEMPORARY ATTRIBUTE	INTEGER	1 REFS
IDSDM	TEMPORARY ATTRIBUTE	INTEGER	1 REFS
IDTI	TEMPORARY ATTRIBUTE	INTEGER	1 REFS

IE1	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
IE2	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
IFAIL	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
IMMEDIATE.FLEET.INS	EVENT NOTICE	(1-D) ALPHA	1 REFS
INCREASE.INSPECTION.	EVENT NOTICE	(1-D) ALPHA	1 REFS
INSP.SCH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
IN.SERVICE.DAMAGE	EVENT NOTICE	(1-D) ALPHA	3 REFS
I.IN.SERVICE.DAMAGE	GLOBAL VARIABLE	INTEGER	1 REFS
I.RETIRE.FROM.SERVIC	GLOBAL VARIABLE	INTEGER	1 REFS
I.I.INSPECTION.INCRE	GLOBAL VARIABLE	INTEGER	1 REFS
LDX	GLOBAL VARIABLE	INTEGER	2 REFS
LGHT.TO.FAILURE	GLOBAL VARIABLE	REAL	2 REFS
LMTA	GLOBAL VARIABLE	INTEGER	5 REFS
LTHO	GLOBAL VARIABLE	ALPHA	1 REFS
MFR1	GLOBAL VARIABLE	(1-D) REAL	4 REFS
MFR2	GLOBAL VARIABLE	(1-D) REAL	2 REFS
MSR1	GLOBAL VARIABLE	(1-D) REAL	8 REFS
MSR2	GLOBAL VARIABLE	(1-D) REAL	4 REFS
NCZ	GLOBAL VARIABLE	INTEGER	6 REFS
NOAC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
NO.FAIL	GLOBAL VARIABLE	INTEGER	6 REFS
NSFL	GLOBAL VARIABLE	INTEGER	5 REFS
OCCUR.MOD	GLOBAL VARIABLE	(1-D) REAL	2 REFS
PFID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
PFTIM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
REACH.FAIL.SAFE.LGT	EVENT NOTICE	(1-D) INTEGER	3 REFS
RETIRE.FROM.SERVICE	EVENT NOTICE	(1-D) INTEGER	3 REFS
S	LOCAL RECURSIVE	REAL	3 REFS
SACID	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SF	GLOBAL VARIABLE	REAL	1 REFS
SFLTHR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE	INTEGER	1 REFS
SMOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
SNFL	GLOBAL VARIABLE	INTEGER	6 REFS
SSTAN	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
SU	GLOBAL VARIABLE	REAL	3 REFS
TAC	LOCAL RECURSIVE	REAL	7 REFS
TAIL.ID	TEMPORARY ATTRIBUTE	INTEGER	1 REFS
TAI	LOCAL RECURSIVE	REAL	4 REFS
TA2	LOCAL RECURSIVE	REAL	4 REFS
TIME.A	TEMPORARY ATTRIBUTE	REAL	3 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	10 REFS
TLID	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
T.INSPECTION.INCREAS	EVENT NOTICE	(2-D) INTEGER	3 REFS
1CL	LOCAL RECURSIVE	REAL	12 REFS
1.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	4 REFS
1.IFE	EVENT NOTICE	(1-D) ALPHA	3 REFS
1.STRENGTH.REDUCTION	EVENT NOTICE	(1-D) ALPHA	5 REFS
2CL	LOCAL RECURSIVE	REAL	12 REFS
2.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	4 REFS
2.IFE	EVENT NOTICE	(1-D) ALPHA	3 REFS
2.NUM.OF.CRASH	GLOBAL VARIABLE	INTEGER	1 REFS
2.STRENGTH.REDUCTION	EVENT NOTICE	INTEGER	5 REFS

```

1  EVENT RETIRE.FROM.SERVICE(IDPET)
2  **
3  ** REPRESENTS RETIREMENT OF THE AIRCRAFT FROM SERVICE
4  **
5  DEFINE IDPET AS AN INTEGER VARIABLE
6  LET ID = IDRET
7  IF 1.CR.EXISTS(ID) = "YES"
8  CALL COMP.RISK.YIELDING CL
9  LET NCZ = NCZ+1
10 LET CLGT(NCZ) = CL
11 LET HZ(NCZ) = HZRD(ID)
12 LET PFID(NCZ) = ID
13 LET PFTIM(NCZ) = TIME.V-ENTRY.TIME(AIRPLANE(ID))
14 ALWAYS
15 IF (YHO = "YES"
16 FOR I = 1 TO NOAC(IDX)
17 DO
18 IF ID = TLID(LDX,I)
19 SKIP 1 OUTPUT LINE
20 PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
21 A/C NO. *** RETIRED FROM SERVICE AT ***** FLIGHT HOURS
22 LEAVE
23 ELSE
24 LOOP
25 ALWAYS
26 IF SMOD.PENDING(ID) = "YES"
27 LET REEN.MODIFIED = BEEN.MODIFIED -1
28 ALWAYS
29 IF SMOD.PENDING(ID) = "NO"
30 IF OCCUP.MOD(ID) GT ENTRY.TIME(AIRPLANE(ID))
31 LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - OCCUR.MOD(ID)
32 JUMP AHEAD
33 OTHERWISE
34 LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - ENTRY.TIME(AIRPLANE(ID))
35 HERE
36 REGARDLESS
37 REMOVE AIRPLANE(ID) FROM ACTIVE.FLEET
38 FILE AIRPLANE(ID) IN FLEET.RETIRED
39 ADD 1 TO 1.NUM.OF.RETIRE
40 **
41 ** CANCEL SCHEDULED EVENTS AND DESTROY EVENT NOTICES
42 **
43 IF IE1(ID) = "YES"
44 LET 1.ITE = AIE(ID)
45 CANCEL THE 1.ITE
46 DESTROY THE 1.ITE
47 IF IE2(ID) = "YES"
48 LET 2.ITE = A2E(ID)
49 CANCEL THE 2.ITE
50 DESTROY THE 2.ITE
51 ALWAYS
52 IF CO.EXISTS(ID) NE "NN"

```

```

53 LET COROSION = AC(ID)
54 IF CO.EXISTS(ID) = "NS"
55 CANCEL THE COROSION
56 ALWAYS
57 DESTROY THE COROSION
58 LET CO.EXISTS(ID) = "NN"
59 REGARDLESS
60 IF 1.CR.EXISTS(ID) = "NN"
61 LET 1.STRENGTH.REDUCTION = AISR(ID)
62 IF 1.CR.EXISTS(ID) = "NS"
63 CANCEL THE 1.STRENGTH.REDUCTION
64 ALWAYS
65 DESTROY THE 1.STRENGTH.REDUCTION
66 LET 1.CR.EXISTS(ID) = "NN"
67 IF 2.CR.EXISTS(ID) = "NN"
68 LET 2.STRENGTH.REDUCTION = A2SR(ID)
69 IF 2.CR.EXISTS(ID) = "NS"
70 CANCEL THE 2.STRENGTH.REDUCTION
71 ALWAYS
72 DESTROY THE 2.STRENGTH.REDUCTION
73 LET 2.CR.EXISTS(ID) = "NN"
74 REGARDLESS
75 DESTROY THE 2.STRENGTH.REDUCTION
76 REGARDLESS
77 IF INSP.SCH(ID) = "YES"
78 CALL CANCEL-SCHEDULED-INSPECTIONS
79 REGARDLESS
80 LET D.LEVEL.INSPECTION = ADL(ID)
81 CANCEL THE D.LEVEL.INSPECTION
82 DESTROY THE D.LEVEL.INSPECTION
83 IF AIL(ID) = "YES"
84 LET RECH.FAIL.SAFE.LGT = ARSL(ID)
85 CANCEL THE RECH.FAIL.SAFE.LGT
86 DESTROY THE RECH.FAIL.SAFE.LGT
87 LET AIL(ID) = "NO"
88 REGARDLESS
89 IF FSH(ID) = "YES"
90 LET FAILURE = AF(ID)
91 CANCEL THE FAILURE
92 DESTROY THE FAILURE
93 LET FSH(ID) = "NO"
94 ALWAYS
95 RETURN
96 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT RETIRE.FROM.SERVICE

AC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACTIVE.FLEET	SET	(1-D) INTEGER	1 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AF	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ATL	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
AIRFRAME.TIME	GLOBAL VARIABLE	REAL	4 REFS
AIRPLANE	GLOBAL VARIABLE	(1-D) INTEGER	6 REFS
ARFSL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AIE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ATSP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AZSR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ARMED.MODIFIED	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
CANCEL.SCHEDULED.INS	PROCEDURE	INTEGER	1 REFS
CL	LOCAL RECURSIVE	(1-D) REAL	2 REFS
CLGT	GLOBAL VARIABLE	(1-D) REAL	1 REFS
COMP.RISK	PROCEDURE	INTEGER	1 REFS
CORROSION	EVENT NOTICE	INTEGER	3 REFS
CG.FXISTS	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS
D.LEVEL.INSPECTION	EVENT NOTICE	REAL	4 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE	REAL	3 REFS
FAILURE	EVENT NOTICE	REAL	3 REFS
FLEET.RETIRED	SET	REAL	3 REFS
FSH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
MZ	GLOBAL VARIABLE	(1-D) REAL	2 REFS
HZRD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
I	LOCAL RECURSIVE	REAL	1 REFS
ID	GLOBAL VARIABLE	INTEGER	2 REFS
IDRET	GIVEN ARGUMENT	INTEGER	40 REFS
IE1	*TEMPORARY ATTRIBUTE	INTEGER	3 REFS
IE2	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
INSP.SCH	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
LXI	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
LTHO	GLOBAL VARIABLE	INTEGER	1 REFS
NCZ	GLOBAL VARIABLE	ALPHA	6 REFS
NOAC	GLOBAL VARIABLE	INTEGER	1 REFS
OCCUR.MOD	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
PFID	GLOBAL VARIABLE	(1-D) REAL	1 REFS
PFTTM	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
PEACH.FAIL.SAFE.LGT	GLOBAL VARIABLE	(1-D) INTEGER	3 REFS
RETIRE.FROM.SERVICE	EVENT NOTICE	INTEGER	1 REFS
SMOD.PENDING	EVENT NOTICE	INTEGER	2 REFS
TIME.V	GLOBAL VARIABLE	(1-D) ALPHA	2 REFS
TLID	SYSTEM ATTRIBUTE	REAL	4 REFS
1.CK.EXISTS	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
1.IYE	GLOBAL VARIABLE	(1-D) ALPHA	4 REFS
1.NUM.OF.RETIRE	EVENT NOTICE	INTEGER	3 REFS
1.STRENGTH.REDUCTION	GLOBAL VARIABLE	INTEGER	1 REFS
2.CK.EXISTS	EVENT NOTICE	(1-D) ALPHA	3 REFS
2.IYE	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS

-241-

--- CDC- 6600 - CACI-SIMSSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2
2.STRENGTH.REDUCTION EVENT NOTICE

02/02/78. 18.36.02.

PAGE 150

3 REFS

-242-

```

1  EVENT REPAIR(IDREP*NI)
2  DEFINE IDREP AS AN INTEGER VARIABLE
3  DEFINE DONE AS AN ALPHA VARIABLE
4  IF CHG.FREQ.TIME NE TIME.V
5  LET DONE = "NO"
6  ALWAYS
7  LET ID = IDREP
8  LET TAC = 0.0
9  LET CL2 = 0.0
10 IF FSH(ID) = "YES"
11 LET FAILURE = AF(ID)
12 CANCEL THE FAILURE
13 DESTROY THE FAILURE
14 LET FSH(ID) = "NO"
15 ALWAYS
16 IF AIL(ID) = "YES"
17 LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
18 CANCEL THE REACH.FAIL.SAFE.LGT
19 DESTROY THE REACH.FAIL.SAFE.LGT
20 LET AIL(ID) = "NO"
21 REGARDLESS
22 LET RST = USAGE.LIFE - TIME.V * ENTRY.TIME(AIRPLANE(ID))
23 IF CO.EXISTS(ID) = "YES"
24 LET CGRI(ID) = 1.0
25 LET CORROSION = AC(ID)
26 LET TAC = TIME.ATCORROSION
27 DESTROY THE CORROSION
28 LET CO.EXISTS(ID) = "NN"
29 CALL PREDICT.CORROSION YIELDING HOURS.TO.CORROSION
30 IF HOURS.TO.CORROSION LT RST
31 SCHEDULE A CORROSION LT RST
32 LET AC(ID) = CORROSION
33 LET COREP.TIME(ID) = TIME.V
34 LET CO.EXISTS(ID) = "NS"
35 REGARDLESS
36 REGARDLESS
37 IF 1.CR.EXISTS(ID) = "YES"
38 LET CCL1 = COME
39 LET CCL2 = CTWO
40 LET CCL3 = CTTHREE
41 LET 1.STRENGTH.REDUCTION = AISR(ID)
42 LET TAI = TIME.A(1).STRENGTH.REDUCTION
43 LET CL = (TIME.V-TAI) * MSRI(ID) * CGRI(ID)
44 IF TAC GT TAI
45 LET CL = (TAC-TAI)*MSRI(ID) + (TIME.V-TAC)*MSRI(ID)*CGRI(ID)
46 REGARDLESS
47 IF CCL1 LT CL
48 LET CL = CCL1 + ((CL-CCL1)/MSRI(ID))*MFRI(ID)
49 REGARDLESS
50 IF CCL2 LT CL
51 LET CL = CCL2 + ((CL-CCL2)/MFRI(ID))*MSR2(ID)
52 REGARDLESS
53 IF CCL3 LT CL

```

```

54 LET CL = CCL3 + ((CL-CCL3)/MSR2(ID))*MFR2(ID)
55 REGARDLESS
56 LET CL1 = CL
57 LET STR.RED = CL
58 IF 2.CR.EXISTS(ID) = "YES"
59 LET 2.STRENGTH.REDUCTION = A2SR(ID)
60 LET TAZ = TIME.A(2.STRENGTH.REDUCTION)
61 LET CL = (TIME.V-TAZ) * MSR1(ID) * CGRI(ID)
62 IF TAC GT TAZ
63 LET CL = (TAC-TAZ)*MSR1(ID) + (TIME.V-TAC)*MSR1(ID)*CGRI(ID)
64 REGARDLESS
65 IF CCL1 LT CL
66 LET CL = CCL1 + ((CL-CCL1)/MSR1(ID))*MFR1(ID)
67 REGARDLESS
68 IF CCL2 LT CL
69 LET CL = CCL2 + ((CL-CCL2)/MFR1(ID))*MSR2(ID)
70 REGARDLESS
71 IF CCL3 LT CL
72 LET CL = CCL3 + ((CL-CCL3)/MSR2(ID))*MFR2(ID)
73 REGARDLESS
74 LET CL2 = CL
75 ADD CL TO STR.MED
76 REGARDLESS
77 LET MAX.CRK = CL1 + CL2
78 LET FLEET.STR.RED = FLEET.STR.RED + STR.RED
79 IF DONE = "YES"
80 GO TO SKP
81 ALWAYS
82 IF D.INT.FIND = "YES" AND M1 = 4
83 LET POT.CRK = MAX.CRK + M1.MEAN * ARCD(4)*SAMPLING
84 LET GROW=MAX.CRK*ARCD(4)*SAMPLING/(TIME.V-ENTRY.TIME(AIRPLANE(ID)))
85 JUMP AHEAD
86 ALWAYS
87 LET POT.CRK = MAX.CRK + M1.MEAN*ARCD(4)
88 LET GROW=MAX.CRK*ARCD(4)/(TIME.V-ENTRY.TIME(AIRPLANE(ID)))
89 HERE
90 IF POT.CRK GT CCL1
91 LET POT.CRK = CCL1 + ((POT.CRK-CCL1)/M1.MEAN) * M2.MEAN
92 REGARDLESS
93 IF POT.CRK GT CCL2
94 LET POT.CRK = CCL2 + ((POT.CRK-CCL2)/M2.MEAN)*M3.MEAN
95 REGARDLESS
96 IF POT.CRK GT CCL3
97 LET POT.CRK = CCL3 + ((POT.CRK -CCL3)/M3.MEAN)*M4.MEAN
98 REGARDLESS
99 IF MAX.CRK GT CCL1
100 IF D.INT.FIND = "YES" AND M1 = 4
101 LET POT.CRK = MAX.CRK + M2.MEAN * ARCD(4)*SAMPLING
102 JUMP AHEAD
103 ALWAYS
104 LET POT.CRK = MAX.CRK + M2.MEAN * ARCD(4)
105 HERE
106 IF POT.CRK GT CCL2

```

1 244 7

```

107 LET POT.CPK = CCL2 + ((POT.CRK - CCL2) / M2.MEAN) * M3.MEAN
108 REGARDLESS
109 IF POT.CRK GT CCL3
110 LET POT.CRK = CCL3 + ((POT.CRK - CCL3) / M3.MEAN) * M4.MEAN
111 REGARDLESS
112 IF MAX.CRK GT CCL2
113 IF D.INT.FIND = "YES" AND NI = 4
114 LET POT.CRK = MAX.CRK + M3.MEAN * ABCD(4) * SAMPLING
115 JUMP AHEAD
116 ALWAYS
117 LET POT.CRK = MAX.CRK + M3.MEAN * ABCD(4)
118 HERE
119 IF POT.CRK GT CCL3
120 LET POT.CRK = CCL3 + ((POT.CRK - CCL3) / M3.MEAN) * M4.MEAN
121 REGARDLESS
122 IF MAX.CRK GT CCL3
123 IF D.INT.FIND = "YES" AND NI = 4
124 LET POT.CRK = MAX.CRK + M4.MEAN * ABCD(4) * SAMPLING
125 JUMP AHEAD
126 ALWAYS
127 LET POT.CRK = MAX.CRK + M4.MEAN * ABCD(4)
128 HERE
129 REGARDLESS
130 IF POT.CRK LT GROW
131 LET POT.CRK = GROW
132 ALWAYS
133 IF MAX.CRK GE FSAF.LGT
134 LET SM.CRK = "NO"
135 IF CHG.FREQ.TIME NE TIME.V
136 SCHEDULE AN IMMEDIATE.FLEET.INSPECTION NOW
137 ALWAYS
138 SCHEDULE AN INCREASE.INSPECTION.FREQUENCY NOW
139 IF CHG.FREQ.TIME = TIME.V
140 LET DONE = "YES"
141 ALWAYS
142 IF LTMO = "YES"
143 PRINT 1 DOUBLE LINE WITH TIME.V, MAX.CRK, IN AS FOLLOWS
144 INTERVAL REDUCTION AT ***** HRS. DUE TO ***, IN. CRACK ON AIRCRAFT *** EXCEEDING FAIL-SAFE LENGTH
145 ALWAYS
146 LET CHG.FREQ.TIME = TIME.V
147 JUMP AHEAD
148 REGARDLESS
149 LET SDI = 1.2 * DLI - 1
150 IF POT.CRK GT (SU-SDI) / (SU-SF) * FSAF.LGT
151 LET SM.CRK = "NO"
152 IF CHG.FREQ.TIME NE TIME.V
153 SCHEDULE AN IMMEDIATE.FLEET.INSPECTION NOW
154 ALWAYS
155 SCHEDULE AN INCREASE.INSPECTION.FREQUENCY NOW
156 IF CHG.FREQ.TIME = TIME.V
157 LET DONE = "YES"
158

```

245-

```

159 ALWAYS
160 IF LTMO = "YES"
161 PRINT 1 DOUBLE LINE WITH TIME.V, POT.CRK, ID AS FOLLOWS
INTERVAL REDUCTION AT ***** HRS. DUE TO ***** IN. POTENTIAL CRACK ON AIRCRAFT *** EXCEEDING LARGE CRACK CRITERIA
162 ALWAYS
163 LET CHG.FREQ.TIME = TIME.V
164 JUMP AHEAD
165 REGARDLESS
166 IF FLEET.STR.RED GT (ESAF-LOT/5.0) * (IDCK-1.NUM.OF.RETIRE-2.NUM.OF.CRASH)
167 SCHEDULE AN INCREASE.F.INSPECTION.FREQUENCY NOW
168 IF CHG.FREQ.TIME = TIME.V
169 LET DONE = "YES"
170 ALWAYS
171 IF LTMO = "YES"
172 PRINT 1 DOUBLE LINE WITH TIME.V, FLEET.STR.UED AS FOLLOWS
INTERVAL REDUCTION AT ***** HRS. DUE TO ***** IN. SUM OF CRACK LENGTHS EXCEEDING SMALL CRACK CRITERIA
173 ALWAYS
174 LET CHG.FREQ.TIME = TIME.V
175 LET FLEET.STR.RED = 0.
176 REGARDLESS
177 LET SM.CRK = "YES"
178 HERE
179 *SKIP* IF THOD.PENDING(ID) = "YES" OR SMOD.PENDING(ID) = "YES"
180 IF THOD.PENDING(ID) = "YES"
181 CALL INSTALL.MODIFICATION
182 LET THOD.PENDING(ID) = "NO"
183 IF TIME.V LT ENTRY.TIME(AIRPLANE(ID)) * C7 * 1A2FL
184 LET T.INSPECTION.INCREASE = ATIII(ID)
185 CANCEL THE T.INSPECTION.INCREASE
186 DESTROY THE T.INSPECTION.INCREASE
187 *REGARDLESS
188 JUMP AHEAD
189 ALWAYS
190 CALL INSTALL.MODIFICATION
191 LET SMOD.PENDING(ID) = "NO"
192 HERE
193 RETURN
194 ALWAYS
195 IF 1.CP.EXISTS(ID) = "YES"
196 CALL COMP.RISK YIELDING CL
197 LET NCZ = NCZ+1
198 LET CLAT(NCZ) = CL
199 LET MZ(NCZ) = MZPD(ID)
200 LET PFI(NCZ) = ID
201 LET PFI(MCZ) = TIME.V-ENTRY.TIME(AIRPLANE(ID))
202 ALWAYS
203 IF IMOD=0 OR RETRO="YES" OR MOD.NO NE 1 OR (ID GT IMOD AND RETRO="NO")
204 LET COST.OF.REPAIRS = COST.OF.REPAIRS + FIXIT.COST
205 ALWAYS
206 IF IMP.MOD.SCH = "NO" AND DEC.ON.MOD.SCH = "NO"
207 IF TEST.FAILURE = "YES" AND TIME.V LT START.TEST + TEST.LIFE /
208 TEST.ACCEL.FACT OR TEST.FAILURE = "YES" AND TIME.V GT START.TEST +
209 TEST.LIFE / TEST.ACCEL.FACT + LEAD.TIME OR TEST.FAILURE = "NO"

```

1246.

```

210 IF TO_BE_MODIFIED = HEEN_MODIFIED - 1.MUM.OF.RETIME - 2.MUM.OF.CRASH LE 0
211 IF IMOD=0 OR RETRO="YES" OR MOD.MQ ME 1 OR IID 67 IMOD AND RETRO="NO"
212 SCHEDULE A DECISION.ON.MOD.NOW
213 LET DEC.ON.MOD.SCH = "YES"
214 ALWAYS
215 REGARDLESS
216 REGARDLESS
217 REGARDLESS
218
219 LET PAFL = ACTUAL.AVG.FAT.LIFE
220 IF ID LE FOUR
221 LET AAFL = 1AAFL
222 REGARDLESS
223 CALL FATIGUE.LIFE.SCATTER(AAFL,3.3,7127)YIELDING FIRST.LIFE AND SECOND.LIFE
224 LET 1.STRENGTH.REDUCTION = A1SR(IID)
225 LET 2.STRENGTH.REDUCTION = A2SR(IID)
226 DESTROY THE 1.STRENGTH.REDUCTION
227 IF 2.CR.EXISTS(IID) = "YES"
228 DESTROY THE 2.STRENGTH.REDUCTION
229 JUMP AHEAD
230 OTHERWISE
231 IF 2.CR.EXISTS(IID) = "NS"
232 IF TIME.A(2.STRENGTH.REDUCTION) = TIME.V LT FIRST.LIFE
233 LET SECOND.LIFE = FIRST.LIFE
234 LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION)
235 CANCEL THE 2.STRENGTH.REDUCTION
236 DESTROY THE 2.STRENGTH.REDUCTION
237 JUMP AHEAD
238 OTHERWISE
239 LET SECOND.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
240 CANCEL THE 2.STRENGTH.REDUCTION
241 DESTROY THE 2.STRENGTH.REDUCTION
242 JUMP AHEAD
243 OTHERWISE
244 LET SECOND.LIFE = USAGE.LIFE
245 HERE
246 LET 1.CR.EXISTS(IID) = "NS"
247 LET 2.CR.EXISTS(IID) = "NS"
248 IF FIRST.LIFE LT NST OR CO.EXISTS(IID) = "NS" OR SD.SCH(IID) = "YES"
249 SCHEDULE A 1.STRENGTH.REDUCTION(IID) AT TIME.V + FIRST.LIFE
250 LET A1SR(IID) = 1.STRENGTH.REDUCTION
251 LET 1.CR.EXISTS(IID) = "NS"
252 IF SECOND.LIFE LT NST OR CO.EXISTS(IID) = "NS" OR SD.SCH(IID) = "YES"
253 SCHEDULE A 2.STRENGTH.REDUCTION(IID) AT TIME.V + SECOND.LIFE
254 LET A2SR(IID) = 2.STRENGTH.REDUCTION
255 LET 2.CR.EXISTS(IID) = "NS"
256 REGARDLESS
257 REGARDLESS
258 REGARDLESS
259 IF IE1(IID) = "YES"
260 LET 1.IFE = AIE(IID)
261 LET IE1(IID) = "NO"
262 CANCEL THE 1.IFE

```

```

263 DESTROY THE 1.ITE
264 IF IE2(ID1) = "YES" -
265 LET 2.ITE = A2E(ID)
266 LET IE2(ID) = "NO"
267 CANCEL THE 2.ITE
268 DESTROY THE 2.ITE
269 ALWAYS
270 IF LTWO = "YES"
271 FOR I = 1 TO NGAC(LDX)
272 DO
273 IF ID = TLID(LDX,I)
274 SKIP 1 OUTPUT LINE
275 PRINT 3 LINES WITH ID, TIME,Y-ENTRY,TIME(AIRPLANE(ID)), FIRST-LIFE,
276 SECOND-LIFE, AS FOLLOWS
277 A/C NO. *** HAS ALL DEFECTS REPAIRED AT ***** FLIGHT HOURS
278 1ST CRACK INITIATION-PROJECTED AT ***** FLIGHT HOURS
279 2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
280 LEAVE
281 ELSE
282 LOOP
283 ALWAYS
284 CANCEL SCHEDULED INSPECTIONS ON THIS AIRCRAFT
285 IF INSP.SCH(ID) = "YES"
286 CALL CANCEL.SCHEDULED.INSPECTIONS
287 ALWAYS
288 RETURN
289 END

```

248

SYMBOLIC REFERENCE MAP (R = 1) - EVENT REPAIR

AAFL	LOCAL RECURSIVE	< 30	REAL	3 REFS
ABCD	GLOBAL VARIABLE	(1-D)	REAL	10 REFS
AC	GLOBAL VARIABLE	(1-D)	INTEGER	2 REFS
ACTUAL.AVG.FAT.LIFE	GLOBAL VARIABLE		REAL	1 REFS
AF	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
AFL	GLOBAL VARIABLE	(1-D)	ALPHA	2 REFS
AIRPLANE	GLOBAL VARIABLE	(1-D)	INTEGER	6 REFS
ARFSL	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
ATII	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
AIE	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
AISR	GLOBAL VARIABLE	(1-D)	INTEGER	3 REFS
AZE	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
AZSR	GLOBAL VARIABLE	(1-D)	INTEGER	3 REFS
BEEN.MODIFIED	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
CANCEL.SCHEDULED.INS	PROCEDURE		INTEGER	1 REFS
CCL1	LOCAL RECURSIVE	< 8	REAL	11 REFS
CCL2	LOCAL RECURSIVE	< 9	REAL	14 REFS
CCL3	LOCAL RECURSIVE	< 10	REAL	17 REFS
CGR1	GLOBAL VARIABLE	(1-D)	REAL	5 REFS
CHG.FREQ.TIME	GLOBAL VARIABLE		REAL	9 REFS
CL	LOCAL RECURSIVE	< 12	REAL	28 REFS
CL6T	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
CL1	LOCAL RECURSIVE	< 13	REAL	2 REFS
CL2	LOCAL RECURSIVE	< 5	REAL	3 REFS
COMP.RISK	PROCEDURE		INTEGER	1 REFS
CONF	GLOBAL VARIABLE		REAL	1 REFS
COREP.TIME	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
COROSION	EVENT NOTICE			5 REFS
COST.OF.REPAIRS	GLOBAL VARIABLE		REAL	2 REFS
CO.EXISTS	GLOBAL VARIABLE	(1-D)	ALPHA	5 REFS
CTHREE	GLOBAL VARIABLE		REAL	1 REFS
CTWO	GLOBAL VARIABLE		REAL	1 REFS
CT	GLOBAL VARIABLE		REAL	1 REFS
DECISION.ON.MOD	EVENT NOTICE			1 REFS
DEC.ON.MOD.SCH	GLOBAL VARIABLE		ALPHA	2 REFS
DLL	GLOBAL VARIABLE		REAL	1 REFS
DOONE	LOCAL RECURSIVE	< 3	ALPHA	6 REFS
D.INT.FIND	GLOBAL VARIABLE		ALPHA	4 REFS
ENTRY.TIME	GLOBAL VARIABLE		REAL	6 REFS
FAILURE	TEMPORARY ATTRIBUTE		REAL	3 REFS
FATIGUE.LIFE.SCATTER	EVENT NOTICE			1 REFS
FDCK	PROCEDURE		INTEGER	1 REFS
FIRST.LIFE	GLOBAL VARIABLE	< 31	INTEGER	7 REFS
FIXT.COST	LOCAL RECURSIVE		REAL	1 REFS
FLEET.STR.REU	GLOBAL VARIABLE		REAL	5 REFS
FSAF.LGT	GLOBAL VARIABLE		REAL	3 REFS
FSH	GLOBAL VARIABLE	(1-D)	ALPHA	2 REFS
GROW	LOCAL RECURSIVE	< 18	REAL	4 REFS
HOURS.TO.CORROSION	LOCAL RECURSIVE	< 7	REAL	3 REFS
HZ	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
HZRD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS

1 249

I	LOCAL RECURSIVE	< 33	REAL	2 REFS
ID	GLOBAL VARIABLE		INTEGER	16 REFS
IDCK	GLOBAL VARIABLE		INTEGER	1 REFS
IDREP	GIVEN ARGUMENT	< 1	INTEGER	
	+TEMPORARY ATTRIBUTE		INTEGER	3 REFS
IE1	GLOBAL VARIABLE	(1-D)	ALPHA	2 REFS
IE2	GLOBAL VARIABLE	(1-D)	ALPHA	2 REFS
IMOD	EVENT NOTICE		INTEGER	2 REFS
IMP.MOD.SCH	GLOBAL VARIABLE		ALPHA	4 REFS
INCREASE.INSPECTION.	EVENT NOTICE		INTEGER	1 REFS
INSP.SCH	GLOBAL VARIABLE	(1-D)	ALPHA	3 REFS
INSTALL.MODIFICATION	PROCEDURE		INTEGER	1 REFS
LDZ	GLOBAL VARIABLE		INTEGER	2 REFS
LEAD.TIME	GLOBAL VARIABLE		REAL	1 REFS
LTHO	GLOBAL VARIABLE		ALPHA	4 REFS
MAX.CRK	LOCAL RECURSIVE	< 16	REAL	16 REFS
MFR1	GLOBAL VARIABLE	(1-D)	REAL	4 REFS
MFR2	GLOBAL VARIABLE	(1-D)	REAL	4 REFS
MOD.NO	GLOBAL VARIABLE		INTEGER	2 REFS
MSR1	GLOBAL VARIABLE	(1-D)	REAL	2 REFS
MSR2	GLOBAL VARIABLE	(1-D)	REAL	8 REFS
M1.MEAN	GLOBAL VARIABLE	(1-D)	REAL	4 REFS
M2.MEAN	GLOBAL VARIABLE		REAL	3 REFS
M3.MEAN	GLOBAL VARIABLE		REAL	5 REFS
M4.MEAN	GLOBAL VARIABLE		REAL	7 REFS
NCZ	GLOBAL VARIABLE		REAL	5 REFS
NI	GIVEN ARGUMENT	< 2	INTEGER	6 REFS
	+TEMPORARY ATTRIBUTE		REAL	
NOAC	GLOBAL VARIABLE	(1-D)	INTEGER	5 REFS
PFID	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
PFTIM	GLOBAL VARIABLE	(1-D)	INTEGER	1 REFS
POT.CRK	LOCAL RECURSIVE	< 17	REAL	30 REFS
PREDICT.CORROSION	PROCEDURE		INTEGER	1 REFS
REACH.FAIL.SAFE.LGT	EVENT NOTICE		INTEGER	3 REFS
REPAIR	EVENT NOTICE		INTEGER	1 REFS
RETRO	GLOBAL VARIABLE		ALPHA	4 REFS
RST	LOCAL RECURSIVE	< 6	REAL	4 REFS
SAMPLING	GLOBAL VARIABLE		INTEGER	5 REFS
SDL	LOCAL RECURSIVE	< 29	REAL	2 REFS
SD.SCH	GLOBAL VARIABLE	(1-D)	ALPHA	2 REFS
SECOND.LIFE	LOCAL RECURSIVE	< 32	REAL	7 REFS
SF	GLOBAL VARIABLE		REAL	1 REFS
SKP	UNSUBSCRIPTED LABEL		REAL	2 REFS
SMOD.PENDING	GLOBAL VARIABLE	(1-D)	ALPHA	2 REFS
SM.CRK	GLOBAL VARIABLE		ALPHA	3 REFS
START.TEST	GLOBAL VARIABLE		REAL	2 REFS
STR.RED	LOCAL RECURSIVE	< 14	REAL	3 REFS
SU	GLOBAL VARIABLE		REAL	2 REFS
TAC	LOCAL RECURSIVE	< 4	REAL	8 REFS
TAL	LOCAL RECURSIVE	< 11	REAL	4 REFS
TAZ	LOCAL RECURSIVE	< 15	REAL	4 REFS
TEST.ACCEL.EACT	GLOBAL VARIABLE		REAL	2 REFS

TEST.LIFE	GLOBAL VARIABLE	REAL	2 REFS
TES.FAILURE	GLOBAL VARIABLE	ALPHA	3 REFS
TIME.A	TEMPORARY ATTRIBUTE	REAL	6 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	30 REFS
TLID	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
THOD.PENDING	GLOBAL VARIABLE	(1-D) ALPHA	3 REFS
TO.RE.MODIFIED	GLOBAL VARIABLE	INTEGER	1 REFS
T.INSPECTION.INCREAS	EVENT NOTICE	REAL	3 REFS
USAGE.LIFE	GLOBAL VARIABLE	REAL	2 REFS
1.APL	GLOBAL VARIABLE	REAL	2 REFS
1.CP.EXISTS	GLOBAL VARIABLE	REAL	4 REFS
1.ITE	EVENT NOTICE	(1-D) ALPHA	3 REFS
1.NUM.OF.RETIME	GLOBAL VARIABLE	INTEGER	2 REFS
1.STRENGTH.REDUCTION	EVENT NOTICE	(1-D) ALPHA	6 REFS
2.CP.EXISTS	GLOBAL VARIABLE	INTEGER	5 REFS
2.ITE	EVENT NOTICE	INTEGER	3 REFS
2.NUM.OF.CRASH	GLOBAL VARIABLE	INTEGER	2 REFS
2.STRENGTH.REDUCTION	EVENT NOTICE	INTEGER	13 REFS

```

1  EVENT T.INSPECTION,IMCHEAD(IDT1)
2  DEFINE IDT1 AS AN INTEGER VARIABLE
3
4  LET ID = IDT1
5  LET D.INTERVAL(ID) = D.INTERVAL(ID) * T.FREQ.CHG
6  IF IFLAG = "NO"
7  IF LTHO = "YES"
8  SKIP 1 OUTPUT LINE
9  PRINT 4 LINES WITH TIME.V, C.INTERVAL(ID), D.INTERVAL(ID), SAMPLING
10 AS FOLLOWS
11 INSPECTION INTERVAL DECREASE DUE TO TEST FAILURE ***** HRS FROM START
12 C-LEVEL INTERVAL ***** HOURS
13 D-LEVEL INTERVAL ***** HOURS
14 SAMPLING NOW **
15 ALWAYS
16 LET CINSL = C.INTERVAL(ID)
17 LET DINSL = D.INTERVAL(ID)
18 LET NICHG = NICHG+1
19 IF NICHG GT 30
20 JUMP AHEAD
21 OTHERWISE -
22 LET CHG.TIME(NICHG) = TIME.V
23 LET SC(NICHG) = C.INTERVAL(ID)
24 LET SC(NICHG) = D.INTERVAL(ID)
25 LET SAMP(NICHG) = SAMPLING
26 LET MOD.SAVE(NICHG) = MOD.NO
27 LET IFLAG = "YES"
28 ALWAYS
29 LET D.LEVEL.INSPECTION = ADL(ID)
30 CANCEL THE D.LEVEL.INSPECTION
31 IF T.LAST.D(ID) + D.INTERVAL(ID) LT TIME.V
32 RESCHEDULE THE D.LEVEL.INSPECTION(ID) NOW
33 JUMP AHEAD
34 OTHERWISE
35 RESCHEDULE THE D.LEVEL.INSPECTION(ID) AT T.LAST.D(ID) + D.INTERVAL(ID)
36 HERE
37 RETURN
38 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT T.INSPECTION.INCREAS

ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
CHG.TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CINSL	GLOBAL VARIABLE	REAL	1 REFS
C.INTERVAL	GLOBAL VARIABLE	(1-D) REAL	3 REFS
DINSL	GLOBAL VARIABLE	REAL	1 REFS
D.INTERVAL	GLOBAL VARIABLE	(1-D) REAL	7 REFS
D.LEVEL.INSPECTION	EVENT NOTICE	REAL	4 REFS
ID	GLOBAL VARIABLE	INTEGER	16 REFS
IDTI	GIVEN ARGUMENT	INTEGER	
	+TEMPORARY ATTRIBUTE	INTEGER	
IFLAG	GLOBAL VARIABLE	ALPHA	3 REFS
LINO	GLOBAL VARIABLE	ALPHA	2 REFS
MOD.NO	GLOBAL VARIABLE	INTEGER	1 REFS
MOD.SAVE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
NICHG	GLOBAL VARIABLE	INTEGER	8 REFS
SAMP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SAMPLING	GLOBAL VARIABLE	INTEGER	2 REFS
SC	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	3 REFS
T.FREQ.CHG	GLOBAL VARIABLE	REAL	1 REFS
T.INSPECTION.INCREAS	EVENT NOTICE		1 REFS
T-LAST.D	GLOBAL VARIABLE	(1-D) REAL	2 REFS

```

1  EVENT INCREASE.INSPECTION.FREQUENCY
2  **
3  ** INCREASES THE FREQUENCY OF THE LOWEST LEVEL INTERNAL AND EXTERNAL INSPECTIONS
4  ** THESE MAY OR MAY NOT BE THE SAME LEVEL OF INSPECTION
5  ** THE A-LEVEL AND B-LEVEL FREQUENCIES ARE NEVER CHANGED
6  **
7  IF ABCD(4) GE IABCD(4) OR ABCD(3) GE IABCD(3)
8  LET ABC.OLD = ABCD(3)
9  LET ABCD.OLD.D = ABCD(4)
10 LET OLD.SAMP = SAMPLING
11 ALWAYS
12 LET FLEET.STR.RED = 0.0
13 LET ABCD(4) = ABCD(4) * S.FREQ.CHG
14 LET DEC.INT = "YES"
15 IF D.INT.FIND = "YES"
16 LET SAMPLING = 1
17 JUMP AHEAD
18 ALWAYS
19 LET SAMPLING = SAMPLING / S.FREQ.CHG
20 HERE
21 IF ABCD(4) LT 500.
22 LET ABCD(4) = 500.
23 LET SAMPLING.OLD.SAMP
24 ALWAYS
25 IF LYHO = "YES"
26 SKIP 1 OUTPUT LINE
27 PRINT 4 LINES WITH TIME.V, ABCD(3), ABCD(4), SAMPLING AS FOLLOWS
28 INSPECTION INTERVAL DECREASE IMPLEMENTED ***** HOURS FROM START OF SIMULATION
29 C-LEVEL INTERVAL NOW ***** HOURS
30 D-LEVEL INTERVAL NOW ***** HOURS
31 SAMPLING NOW **
32 ALWAYS
33 LET CINSI = ABCD(3)
34 LET DINSI = ABCD(4)
35 LET NICHG = NICHG + 1
36 IF NICHG GT 30
37 JUMP AHEAD
38 OTHERWISE
39 LET CMG.TIME(NICHG) = TIME.V
40 LET SCINICHG = ABCD(3)
41 LET SDINICHG = ABCD(4)
42 LET SAMP(NICHG) = SAMPLING
43 LET MOD.SAVE(NICHG) = MOD.NO
44 HERE
45 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
46 DO
47 LET ID = TAIL-ID
48 LET C.INTERVAL(ID) = ABCD(3)
49 LET D.INTERVAL(ID) = ABCD(4)
50 LET D.LEVEL.INSPECTION = ADL(ID)
51 CANCEL THE D.LEVEL.INSPECTION
52 IF SP.CHR = "NO"
53 LET I-LAST.D/ID1 = TIME.V
54

```

-254-

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2 03/82/78. 18.36.02. PAGE 163
 50 RESCHEDULE THE D.LEVEL.INSPECTION(ID) AT TIME.V + ABCD(4)
 51 JUMP AHEAD
 52 OTHERWISE
 53 IF T.LAST.D(ID) + ABCD(4) LT TIME.V
 54 RESCHEDULE THE D.LEVEL.INSPECTION(ID) NOW
 55 JUMP AHEAD
 56 OTHERWISE
 57 RESCHEDULE THE D.LEVEL.INSPECTION(ID) AT T.LAST.D(ID) + ABCD(4)
 58 JUMP AHEAD
 59 MERF
 60 LOOP
 61 RETURN
 62 END

SYMBOLIC REFERENCE MAP (R = 1) - EVENT INCREASE.INSPECTION.

ABCD	GLOBAL VARIABLE	(1-D) REAL	19 REFS
ABCD.OLD.D	GLOBAL VARIABLE	REAL	1 REFS
ABC.OLD	GLOBAL VARIABLE	REAL	1 REFS
ACTIVE.FLEET	SET		1 REFS
ADL	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
AIRCRAFT	TEMPORARY ENTITY		1 REFS
CHG.TIME	GLOBAL VARIABLE	(1-D) REAL	1 REFS
CINSL	GLOBAL VARIABLE	REAL	1 REFS
C.INTERVAL	GLOBAL VARIABLE	(1-D) REAL	1 REFS
DEC.INT	GLOBAL VARIABLE	ALPHA	1 REFS
DINSL	GLOBAL VARIABLE	REAL	1 REFS
D.INTERVAL	GLOBAL VARIABLE	(1-D) REAL	1 REFS
D.INT.FIND	GLOBAL VARIABLE	ALPHA	1 REFS
D.LEVEL.INSPECTION	EVENT NOTICE		5 REFS
FLEET.SIR.RED	GLOBAL VARIABLE	REAL	1 REFS
ID	GLOBAL VARIABLE	INTEGER	10 REFS
INCREASE.INSPECTION.	EVENT NOTICE		1 REFS
LTHO	GLOBAL VARIABLE	ALPHA	1 REFS
MOD.NO	GLOBAL VARIABLE	INTEGER	1 REFS
MOD.SAVE	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
NICMG	GLOBAL VARIABLE	INTEGER	8 REFS
OLD.SAMP	GLOBAL VARIABLE	INTEGER	2 REFS
SAMP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
SAMPLING	GLOBAL VARIABLE	INTEGER	7 REFS
SC	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SD	GLOBAL VARIABLE	(1-D) REAL	1 REFS
SM.CRK	GLOBAL VARIABLE	ALPHA	1 REFS
S.FREQ.CHG	GLOBAL VARIABLE	REAL	2 REFS
TAIL.IG	TEMPORARY ATTRIBUTE	INTEGER	1 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	5 REFS
T.LAST.D	GLOBAL VARIABLE	(1-D) REAL	3 REFS
LABCD	GLOBAL VARIABLE	(1-D) REAL	2 REFS

```

1  EVENT IMMEDIATE.FLEET.INSPECTION
2  **
3  ** THIS EVENT REPRESENTS AN IMMEDIATE FLEET WIDE INSPECTION CAUSED BY FINDING A
4  ** DEFECT CONSIDERED TOO HAZARDOUS TO DEPEND ON SCHEDULED INSPECTIONS
5  **
6  ** THIS EVENT IS ALWAYS PRECEDED BY THE EVENT INCREASE.INSPECTION.FREQUENCY
7  **
8  IF LTMO = "YES"
9  SKIP 1 OUTPUT LINE
10 PRINT 1 LINE WITH TIME.V AS FOLLOWS
11 FLEET WIDE SPECIAL INSPECTION PERFORMED ***** HOURS FROM START OF SIMULATION
12 ALWAYS
13 LET CCL1 = COME
14 LET CCL2 = CTMO
15 LET CCL3 = CTMR
16 LET MSIC = MSIC * 1
17 LET FIXIT.COST = S.REPAIR.COST
18 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
19 DO
20 LET LIST = 0.0
21 LET ID = TAIL.ID.
22 IF LTMO = "YES"
23 FOR I = 1 TO NOAC(LDX)
24 DO
25 IF ID = TLID(LDX.I)
26 LET LIST = 1.0
27 LEAVE
28 ELSE
29 LOOP
30 ALWAYS
31 LET M1 = MSR1(ID)
32 LET M2 = MSR1(ID)
33 LET M3 = MSR2(ID)
34 LET M4 = MSR2(ID)
35 LET FOUND = 0
36 LET TAC = 0.0
37
38 IF CO.EXISTS(ID) = "YES"
39 LET COROSUM = AC(ID)
40 LET TAC = TIME.A(COROSUM)
41 LET AREA = C.GROWTH.RATE * CORF * (TIME.V - TAC)
42 CALL PODOI.85.7.1RD.AREA) YIELDING PI
43 IF RANDOM.F(1) LE PL
44 IF LIST = 1.0
45 SKIP 1 OUTPUT LINE
46 PRINT 1 DOUBLE LINE WITH AREA. ID. TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
47 FOLLOWS
48 CORROSION OF AREA **.. SO. INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ***** FLIGHT HOURS
49 ALWAYS
50 LET SCA = AREA
51 LET GSCA = AREA
52 LET OSCO = OSCO * 1
53 LET OSCO = OSCO * 1

```

```

52 LET FOUND = 2
53 REGARDLESS
54 REGARDLESS
55
56 IF 1.CR.EXISTS(ID) = "YES"
57 LET 1.STRENGTH.REDUCTION = AISR(ID)
58 LET TAI = TIME.A11.STRENGTH.REDUCTION)
59 LET CL = (TIME.V-TAI) * M1 * CGRI(ID)
60 IF TAC GT TAI
61 LET CL = (TAC-TAI)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
62 REGARDLESS
63 IF CL GT CCL1
64 LET CL = CCL1 + ((CL-CCL1)/M1)*M2
65 REGARDLESS
66 IF CL GT CCL2
67 LET CL = CCL2 + ((CL-CCL2)/M2)*M3
68 REGARDLESS
69 IF CL GT CCL3
70 LET CL = CCL3 + ((CL-CCL3)/M3)*M4
71 REGARDLESS
72 IF CL LE 5.0
73 CALL PDDI(.058*.7*.14*.CL) YIELDING PL
74 JUMP AHEAD
75 OTHERWISE
76 IF CL GT 5.0 AND CL LE 12.5
77 LET PL = .7172 + .0223 * CL
78 JUMP AHEAD
79 OTHERWISE
80 LET PL = .999
81 HERE
82
83 IF RANDOM.F(1) LE PL
84 IF LIST = 1.0
85 SKIP 1 OUTPUT LINE
86 PRINT 1 DOUBLE LINE WITH CL, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
87 FOLLOWS
88 CPACK OF LENGTH ** INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ***** FLIGHT HOURS
89 ALWAYS
90 LET SCHKL = CL
91 LET GSCRK = CL
92 ADD 1 TO GSCR
93 ADD 1 TO GOSCR
94 LET FOUND = 1
95 REGARDLESS
96
97 IF 2.CR.EXISTS(ID) = "YES"
98 LET 2.STRENGTH.REDUCTION = A2SR(ID)
99 LET TAZ = TIME.A12.STRENGTH.REDUCTION)
100 LET CL = (TIME.V-TAZ) * M1 * CGRI(ID)
101 IF TAC GT TAZ
102 LET CL = (TAC-TAZ)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
103 REGARDLESS
104 IF CL GT CCL1

```

258

```

104 LET CL = CCL1 + ((CL-CCL1)/M1)*M2
105 REGARDLESS
106 IF CL GT CCL2
107 LET CL = CCL2 + ((CL-CCL2)/M2)*M3
108 REGARDLESS
109 IF CL GT CCL3
110 LET CL = CCL3 + ((CL-CCL3)/M3)*M4
111 REGARDLESS
112
113 IF CL LE 5.0
114 CALL PDD((.959**/.14*CL) YIELDING PL
115 JUMP AHEAD
116 OTHERWISE
117 IF CL GT 5.0 AND CL LE 12.5
118 LET PL = .7172 + .0223 * CL
119 JUMP AHEAD
120 OTHERWISE
121 LET PL = .999
122 HERE
123
124 IF RANDOM.F(1) LE PL
125 IF LIST = 1.0
126 SKIP 1 OUTPUT LINE
127 PRINT 1 DOUBLE LINE WITH CL, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
128 FOLLOWS
129 CRACK OF LENGTH **.* INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ***** FLIGHT HOURS
130 ALWAYS
131 LET SCRKL = CL
132 LET GSCRK = CL
133 ADD 1 TO GSCRK
134 ADD 1 TO GOSCK
135 LET FOUND = 1
136 REGARDLESS
137 REGARDLESS
138
139 IF FOUND = 1 OR FOUND = 2
140 FOR I = 1 TO 10
141 DO
142 IF HI.TIME.ACRFT(1) = ID
143 LET MRDD(1) = TIME.V
144 LEAVE
145 ELSE
146 LOOP
147 REGARDLESS
148 IF FOUND GT 0
149 SCHEDULE A REPAIR(10.5) NOW
150 REGARDLESS
151 LOOP
152 RETURN
153 END

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT IMMEDIATE.FLEET.INSP

AC	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
ACTIVE.FLEET	SET		1 REFS
AIRCHAFT	TEMPORARY ENTITY		1 REFS
AIRPLANE	GLOBAL VARIABLE		3 REFS
AREA	GLOBAL RECURSIVE	< 24	5 REFS
AISP	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
A2SR	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
COL1	LOCAL RECURSIVE	< 12	7 REFS
COL2	LOCAL RECURSIVE	< 13	7 REFS
COL3	LOCAL RECURSIVE	< 14	7 REFS
CGRI	GLOBAL VARIABLE	(1-D) REAL	4 REFS
CL	GLOBAL RECURSIVE	< 28	38 REFS
CONF	GLOBAL VARIABLE	REAL	1 REFS
CORROSION	EVENT NOTICE		2 REFS
CO-EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
CRRF	GLOBAL VARIABLE	PFAL	1 REFS
CTHREE	GLOBAL VARIABLE	DEAL	1 REFS
CTWO	GLOBAL VARIABLE	REAL	1 REFS
C.GROWTH.RATE	GLOBAL VARIABLE	REAL	1 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE		3 REFS
FIXIT.COST	GLOBAL VARIABLE	< 22	1 REFS
FOUND	LOCAL RECURSIVE	REAL	7 REFS
GOSCO	GLOBAL VARIABLE	INTEGER	2 REFS
GOSCR	GLOBAL VARIABLE	INTEGER	2 REFS
GSCA	GLOBAL VARIABLE	REAL	1 REFS
GSCRK	GLOBAL VARIABLE	REAL	2 REFS
HI.TIME.ACRFT	GLOBAL VARIABLE	(1-D) INTEGER	1 REFS
I	LOCAL RECURSIVE	< 17	5 REFS
ID	GLOBAL VARIABLE	INTEGER	24 REFS
IMMEDIATE.FLEET.INSP	EVENT NOTICE		1 REFS
LXI	GLOBAL VARIABLE	INTEGER	2 REFS
LIST	LOCAL RECURSIVE	REAL	5 REFS
LITHO	GLOBAL VARIABLE	ALPHA	2 REFS
MFR1	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MFR2	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MDDO	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSR1	GLOBAL VARIABLE	(1-D) REAL	1 REFS
MSR2	GLOBAL VARIABLE	(1-D) REAL	9 REFS
M1	LOCAL RECURSIVE	REAL	5 REFS
M2	LOCAL RECURSIVE	REAL	5 REFS
M3	LOCAL RECURSIVE	REAL	3 REFS
MA	LOCAL RECURSIVE	REAL	1 REFS
NOAC	GLOBAL VARIABLE	(1-D) INTEGER	2 REFS
NSIC	GLOBAL VARIABLE	INTEGER	2 REFS
OSCO	GLOBAL VARIABLE	INTEGER	2 REFS
OSCR	GLOBAL VARIABLE	INTEGER	2 REFS
PL	GLOBAL RECURSIVE	REAL	18 REFS
POOD	PROCEDURE	INTEGER	3 REFS
RANDOM.F	PROCEDURE	REAL	3 REFS
REPAIR	EVENT NOTICE	REAL	1 REFS
SCA	GLOBAL VARIABLE	REAL	1 REFS

SCRKL	GLOBAL VARIABLE	REAL	2 REFS
S.REPAIR.COST	GLOBAL VARIABLE	REAL	1 REFS
TAC	LOCAL RECURSIVE	REAL	9 REFS
TAIL.ID	TEMPORARY ATTRIBUTE	INTEGER	1 REFS
TAIL	LOCAL RECURSIVE	REAL	4 REFS
TAZ	LOCAL RECURSIVE	REAL	4 REFS
TIME.A	TEMPORARY ATTRIBUTE	REAL	3 REFS
TIME.V	SYSTEM ATTRIBUTE	REAL	10 REFS
TLID	GLOBAL VARIABLE	(2-D) INTEGER	1 REFS
1.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
1.STRENGTH.REDUCTION	EVENT NOTICE	(1-D) ALPHA	2 REFS
2.CR.EXISTS	GLOBAL VARIABLE	(1-D) ALPHA	1 REFS
2.STRENGTH.REDUCTION	EVENT NOTICE	(1-D) ALPHA	2 REFS

```

1  EVENT DECISION.ON.MOD
2
3  98  DECISION IS MADE TO IMPLEMENT A STRUCTURAL MODIFICATION WHENEVER
4  99  PCPH(REPAIR COST PER HOUR) + ICPI(INSPECTION COST PER HOUR)
5  00  IS GREATER THAN PCPH(MODIFICATION COST PER HOUR)
6
7  DEFINE NPDL, MRPL AS INTEGER VARIABLES
8  LET DEC.ON.MOD.SCH = "NO"
9  LET TOOLING = 1ST.TOOLING
10 LET MD.COST = 1ST.MD.COST
11 IF PREVIOUSLY.MODIFIED = "YES"
12 LET TOOLING = AD.TOOLING
13 LET MD.COST = AD.MD.COST
14 REGARDLESS
15 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
16 DO
17 LET MRPH = ENTRY.TIME + USAGE.LIFE - TIME.V - LEAD.TIME - ARCD(3)/2
18 IF MRPH GT 0.0
19 LET POST.MOD.HRS = POST.MOD.HRS + MRPH
20 REGARDLESS
21
22 99  IF TMOO.PENDING(TAIL.ID) = "NO" AND SMOD.PENDING(TAIL.ID) = "NO"
23 IF IMOD=0 OR RETRO="YES" 99 MOD.NO NE 1 OR (TAIL.ID GT IMOD AND RETRO="NO")
24 IF OCCUR.MOD(TAIL.ID) GT 0.0
25 LET ACCUMULATED.MRS = ACCUMULATED.MRS + TIME.V - OCCUR.MOD(TAIL.ID)
26 JUMP AHEAD
27 OTHERWISE
28 LET ACCUMULATED.MRS = ACCUMULATED.MRS + TIME.V - ENTRY.TIME
29 ALWAYS
30 HERE
31 REGARDLESS
32 LOOP
33 LET ACCUMULATED.MRS = ACCUMULATED.MRS + AIRFRAME.TIME
34 IF IMOD GT 0 AND RETRO="NO" AND 1.NUM.OF.RETIRE GT 0 AND MOD.NO = 1
35 IF IMOD GT 1.NUM.OF.RETIRE
36 LET ACCUMULATED.MRS=ACCUMULATED.MRS-USAGE.LIFE*1.NUM.OF.RETIRE
37 JUMP AHEAD
38 OTHERWISE
39 LET ACCUMULATED.MRS=ACCUMULATED.MRS-USAGE.LIFE*IMOD
40 MFPE
41 ALWAYS
42 LET NPDL = TRUNC.F.LEAD.TIME/PRODUCTION.TIME)
43 IF NPDL GT SIZE.OF.FLEET - IDCK
44 LET NPDL = SIZE.OF.FLEET - IDCK
45 REGARDLESS
46 LET POST.MOD.HRS = POST.MOD.HRS+NPDL*(USAGE.LIFE-(LEAD.TIME+ARCD(3))/2)
47 * (SIZE.OF.FLEET - IDCK - NPDL) * USAGE.LIFE
48 IF DEC.INT = "NO"
49 LET ICPI = 0.
50 JUMP AHEAD
51 OTHERWISE
52 IF D.INT.FIND = "YES"
53 LET ICPI = CABCD(4)*(1/ABCD(4) - 1/ABCD.OLD.0)*OLD.SAMP *

```

```

54      JUMP AHEAD
55      OTHERWISE
56      LET ICPH = CAPCD(4)*(1/ABCD(4) - 1/ABCD.OLD.D) * .3
57      HERE
58      IF POST.MOD.MRS = 0.0
59      RETURN
60      OTHERWISE
61      IF 1.NUM.OF.RETIRE GT 0
62      LET NMRET = TRUNC.F(1.FAD.TIME/2.PRODUCTION.TIME)
63      JUMP AHEAD
64      OTHERWISE
65      LET NMRET = 0
66      HERE
67      LET MCPH = (MD.COST*(IDCK*MDL-1.NUM.OF.RETIRE-2.NUM.OF.CRASH-NMRET) + TOOLING) /
68      POST.MOD.MRS
69      LET RCPH = COST.OF.REPAIRS / ACCUMULATED.MRS
70      LET SIM.TIME=BEGIN.PRODUCTION+PRCHG+USAGE.LIFE
71      * (SIZE.OF.FLEET-PRCHG/PRODUCTION.TIME)*2.PRODUCTION.TIME
72      LET RCPH=RCPH+SIM.TIME/TIME.V*SIM.TIME/TIME.V
73      PRINT 1 LINE WITH ICPH, MCPH, RCPH, TIME.V AS FOLLOWS
74      ICPH = ***.*** MCPH = ***.*** TIME = *****
75      IF RCPH + ICPH GT MCPH
76      LET NMFS = (SIZE.OF.FLEET + 1 - TRUNC.F(1.PRCHG/PRODUCTION.TIME)) *
77      2.PRODUCTION.TIME
78      IF TIME.V + LEAD.TIME LT BEGIN.PRODUCTION + PRCHG + NMFS + USAGE.LIFE
79      SCHEDULE AN IMPLEMENT.MODIFICATION AT TIME.V + LEAD.TIME
80      ALWAYS
81      LET IMP.MOD.SCH = "YES"
82      LET PREVIOUSLY.MODIFIED = "YES"
83      IF TES.FAILURE = "YES" AND TIME.V LE START.TEST + ACTUAL.AVG.FAT.LIFE /
84      TEST.ACCEL.FACT
85      LET TES.FAILURE = "NO"
86      CANCEL THE T.IMPLEMENT.MOD
87      DESTROY THE T.IMPLEMENT.MOD
88      REGARDLESS
89      REGARDLESS
90      RETURN
91      FNO

```

SYMBOLIC REFERENCE MAP (R = 1) - EVENT DECISION.ON.MOD

ABCD	GLOBAL VARIABLE	(1-D)	REAL	4 REFS
ABCP.O.D.O	GLOBAL VARIABLE		REAL	3 REFS
ACCUMULATED.HRS	LOCAL RECURSIVE	< 6	REAL	11 REFS
ACTIVE.FLEET	SET			1 REFS
ACTUAL.AVG.FAT.LIFE	GLOBAL VARIABLE		REAL	1 REFS
AD.MD.COST	GLOBAL VARIABLE		REAL	1 REFS
AD.TOOLING	GLOBAL VARIABLE		REAL	1 REFS
AIRCRAFT	TEMPORARY ENTITY			1 REFS
AIRFRAME.TIME	GLOBAL VARIABLE		REAL	1 REFS
BEGIN.PRODUCTION	GLOBAL VARIABLE		REAL	2 REFS
CABCD	GLOBAL VARIABLE	(1-D)	REAL	1 REFS
COST.OF.REPAIRS	GLOBAL VARIABLE		REAL	2 REFS
DECISION.ON.MOD	EVENT NOTICE		REAL	1 REFS
DEC.INT	GLOBAL VARIABLE		ALPHA	1 REFS
DEC.ON.MOD.SCH	GLOBAL VARIABLE		ALPHA	1 REFS
D.INT.FIND	GLOBAL VARIABLE		ALPHA	1 REFS
ENTRY.TIME	TEMPORARY ATTRIBUTE		REAL	2 REFS
ICPM	LOCAL RECURSIVE	< 9	REAL	5 REFS
LOCK	GLOBAL VARIABLE		INTEGER	4 REFS
IMOD	GLOBAL VARIABLE		INTEGER	5 REFS
IMPLEMENT.MODIFICATI	EVENT NOTICE			1 REFS
IMP.MOD.SCH	GLOBAL VARIABLE		ALPHA	1 REFS
LEAD.TIME	GLOBAL VARIABLE		REAL	6 REFS
MCPH	LOCAL RECURSIVE	< 10	REAL	3 REFS
MD.COST	LOCAL RECURSIVE	< 4	REAL	3 REFS
MOD.NO	GLOBAL VARIABLE		REAL	2 REFS
MREF	LOCAL RECURSIVE	< 6	REAL	3 REFS
NFTS	LOCAL RECURSIVE	< 23	REAL	2 REFS
NNRET	LOCAL RECURSIVE	< 2	INTEGER	4 REFS
NPDL	LOCAL RECURSIVE	< 1	INTEGER	7 REFS
OCCUR.MOD	GLOBAL VARIABLE	(1-D)	REAL	2 REFS
OLD.SAMP	GLOBAL VARIABLE		INTEGER	2 REFS
POST.MOD.HRS	LOCAL RECURSIVE	< 7	REAL	6 REFS
PRCHG	GLOBAL VARIABLE		REAL	4 REFS
PREVIOUSLY.MODIFIED	GLOBAL VARIABLE		ALPHA	2 REFS
PRODUCTION.TIME	GLOBAL VARIABLE		REAL	3 REFS
RCPH	LOCAL RECURSIVE	< 11	REAL	5 REFS
RETRO	GLOBAL VARIABLE	< 12	ALPHA	3 REFS
SIM.TIME	LOCAL RECURSIVE		REAL	3 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE		INTEGER	5 REFS
SMOD.PENDING	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
START.TEST	GLOBAL VARIABLE		REAL	5 REFS
TAIL.ID	TEMPORARY ATTRIBUTE		REAL	1 REFS
TEST.ACCEL.FACT	GLOBAL VARIABLE		REAL	1 REFS
YES.FAILURE	GLOBAL VARIABLE		ALPHA	2 REFS
TIME.V	SYSTEM ATTRIBUTE		REAL	9 REFS
TMOD.PENDING	GLOBAL VARIABLE	(1-D)	ALPHA	1 REFS
TOOLING	LOCAL RECURSIVE	< 3	REAL	3 REFS
TRUNC.F	PROCEDURE		INTEGER	3 REFS
T.IMPLEMENT.MOD	EVENT NOTICE			2 REFS
USAGE.LIFE	GLOBAL VARIABLE		REAL	7 REFS

-284-

CDC 6600	CACI SIMSCRIPT II.5	VERSION /4.0-00/	KRONOS 2.1.2	03/02/78.	18.36.02.	PAGE 173
1ST.WD.COST	GLOBAL VARIABLE		REAL	1 REFS		
1ST.TOOLING	GLOBAL VARIABLE		REAL	1 REFS		
1.NUM.OF.RETIME	GLOBAL VARIABLE		INTEGER	5 REFS		
2.NUM.OF.CRASH	GLOBAL VARIABLE		INTEGER	1 REFS		
2.PRODUCTION.TIME	GLOBAL VARIABLE		REAL	3 REFS		

```

1  EVENT IMPLEMENT.MODIFICATION
2  *** REPRESENTS DEVELOPMENT OF MODIFICATION BECAUSE OF SERVICE EXPERIENCE
3  ***
4  LET MOD.NO = MOD.NO + 1
5  LET NMD = MOD + 1
6  LET FOCK = 0
7  LET TO.BE.MODIFIED = IDCK
8  LET REEN.MODIFIED = 0
9  LET IMP.MOD.SCH = "MD"
10 IF DEC.INT = "YES"
11 LET ARCD(1) = ARC.OLD
12 LET ARCD(4) = ARC.OLD.D
13 LET SAMPLING = OLD.SAMP
14 LET NICHG = NICHG + 1
15
16 IF NICHG GT 30
17 JUMP AHEAD
18 OTHERWISE
19 LET CHG.TIME(NICHG) = TIME.V
20 LET SC(NICHG) = ARC.OLD
21 LET SD(NICHG) = ARCD.OLD.D
22 LET SAMP(NICHG) = SAMPLING
23 LET MOD.SAVE(NICHG) = MOD.NO
24
25 HERE
26 LET DEC.INT = "NO"
27 ALWAYS
28 LET COST.OF.REPAIRS = 0.0
29 LET AIRFRAME.TIME = 0.0
30 IF 2.0*ACTUAL.AVG.FAT.LIFE GT PREDICTED.LIFE
31 LET NEW.LIFE = 2.0*ACTUAL.AVG.FAT.LIFE
32 JUMP AHEAD
33 OTHERWISE
34 LET NEW.LIFE = PREDICTED.LIFE
35 HERE
36 LET NMU = MU.R + ((1.0-MU.R)*.15)
37 LET NSIG = SIG.*.85
38 CALL REAL.LIFF(NMU,NSIG,NEW.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE
39 IF MOD.TESTED = "YES" AND ACTUAL.AVG.FAT.LIFE LT NEW.LIFE
40 LET ACTUAL.AVG.FAT.LIFE = NEW.LIFE
41 ALWAYS
42 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
43 DO
44 LET SMOD.PENDING(TAIL.ID) = "YES"
45 LOOP
46 IF LTMO = "YES"
47 SKIP 1 OUTPUT LINE
48 PRINT 6 LINES WITH TIME.V, ARCD(3), ARCD(4), SAMPLING,
49 ACTUAL.AVG.FAT.LIFE AS FOLLOWS
***** HOURS FROM START OF SIMULATION
INSPECTION INTERVAL RETURN TO PRIOR INTERVAL DUE TO SERVICE MOD
C-LEVEL INTERVAL NOW ***** HOURS
D-LEVEL INTERVAL NOW ***** HOURS

```

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2 03/02/78. 18.36.02. PAGE 175

SAMPLING NOW **
AVG.FAT.LIFE * *****
ALWAYS
50 RETURN
51
52 END

-267-

SYMBOLIC REFERENCE MAP (R = 1) - EVENT IMPLEMENT-MODIFICATION

ABCD	GLOBAL VARIABLE	(1-0)	REAL	4 REFS
ABCD.OLD.D	GLOBAL VARIABLE		REAL	2 REFS
ABC.OLD	GLOBAL VARIABLE		REAL	2 REFS
ACTIVE.FLEET	SET			1 REFS
ACTUAL.AVG.FAT.LIFE	GLOBAL VARIABLE		REAL	4 REFS
AIRCRAFT	TEMPORARY ENTITY			1 REFS
AIRFRAME.TIME	GLOBAL VARIABLE		REAL	1 REFS
BEEN.MODIFIED	GLOBAL VARIABLE		INTEGER	1 REFS
CHG.TIME	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
COST.OF.REPAIRS	GLOBAL VARIABLE		REAL	1 REFS
DEC.INT	GLOBAL VARIABLE		ALPHA	2 REFS
FDCK	GLOBAL VARIABLE		INTEGER	1 REFS
IDCK	GLOBAL VARIABLE		INTEGER	1 REFS
IMPLEMENT-MODIFICATION	EVENT NOTICE			1 REFS
IMP.MOD.SCH	GLOBAL VARIABLE		ALPHA	1 REFS
LTHD	GLOBAL VARIABLE		ALPHA	1 REFS
MOD.NO	GLOBAL VARIABLE		INTEGER	3 REFS
MOD.SAVE	GLOBAL VARIABLE		INTEGER	1 REFS
MOD.TESTED	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
MU.R	GLOBAL VARIABLE		REAL	2 REFS
NEW.LIFE	LOCAL RECURSIVE	< 1	REAL	5 REFS
NICMG	GLOBAL VARIABLE		INTEGER	4 REFS
NMD	GLOBAL VARIABLE		INTEGER	2 REFS
NMU	LOCAL RECURSIVE	< 2	REAL	2 REFS
NSIG	LOCAL RECURSIVE	< 3	REAL	2 REFS
OLD.SAMP	GLOBAL VARIABLE		INTEGER	1 REFS
PREDICTED.LIFE	GLOBAL VARIABLE		REAL	2 REFS
REAL.LIFE	PROCEDURE		INTEGER	1 REFS
SAMP	GLOBAL VARIABLE	(1-0)	INTEGER	1 REFS
SAMPLING	GLOBAL VARIABLE		INTEGER	1 REFS
SC	GLOBAL VARIABLE	(1-0)	REAL	3 REFS
SD	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
SIG.R	GLOBAL VARIABLE		REAL	1 REFS
SMOD.PENDING	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
TAIL.ID	TEMPORARY ATTRIBUTE		INTEGER	1 REFS
TIME.V	SYSTEM ATTRIBUTE		REAL	2 REFS
TO.RE.MODIFIED	GLOBAL VARIABLE		INTEGER	1 REFS


```

38 MAX(IN)      ***.00      ***.00      ***.00      ***.00
39 AVG(IN)      ***.00      ***.00      ***.00      ***.00
  SKIP 2 OUTPUT LINES
  PRINT 1 DOUBLE LINE AS FOLLOWS
    NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION
  SKIP 1 OUTPUT LINE
    IF OIC0(1) GT 0      LET PNACA = NACA      LET PXACA = XACA      ALWAYS
    IF OIC0(2) GT 0      LET PNBCA = NBACA     LET PXBCA = XBCA      ALWAYS
    IF OIC0(3) GT 0      LET PNCCA = NCCA      LET PXCCA = XCCA      ALWAYS
    IF OIC0(4) GT 0      LET PNDCA = NDCA      LET PXDCA = XDCA      ALWAYS
    IF OSC0 GT 0          LET PNSCA = NSCA      LET PXSCA = XSCA      ALWAYS
  PRINT 6 DOUBLE LINES WITH OIC0(1), OIC0(2), OIC0(3), OIC0(4), OSC0, PNACA,
  PNBCA, PNCCA, PNDCA, PNSCA, PXACA, PXBCA, PXCCA, PXDCA, PXSCA, MACA, MRCA,
  MCCA, MDCA, MSCA AS FOLLOWS
    A-LEVEL      H-LEVEL      C-LEVEL      D-LEVEL      SPECIAL
    -----      -----      -----      -----      -----
    OCCURRENCES      *****      *****      *****      *****
    MIN(SQ.IN)      ***.00      ***.00      ***.00      ***.00
    MAX(SQ.IN)      ***.00      ***.00      ***.00      ***.00
    AVG(SQ.IN)      ***.00      ***.00      ***.00      ***.00
  SKIP 1 OUTPUT LINE
  PRINT 1 DOUBLE LINE AS FOLLOWS
  INSPECTION INTERVAL(S)MS
  SKIP 1 OUTPUT LINE
  PRINT 1 DOUBLE LINE WITH IABCO(1), IABCO(2), IABCO(3), IABCO(4).
  MOD.SAVE(1), SAMP(1) AS FOLLOWS
    *****      *****      *****      *****
  INITIAL
  IF NICHG GT 3G      LET NICHG = 30      REGARDLESS
  FOR I = 2 TO NICHG
  DO
  PRINT 1 DOUBLE LINE WITH I, IABCO(1), IABCO(2), SC(1), SD(1),
  MOD.SAVE(1), SAMP(1), CHG.TIME(1) AS FOLLOWS
    *****      *****      *****      *****
  **
  LOOP
  SKIP 1 OUTPUT LINE
  PRINT 1 DOUBLE LINE AS FOLLOWS
  CRACK LENGTHS AND CORRESPONDING CUMULATIVE PROBABILITY OF FAILURE
  PRINT 1 DOUBLE LINE AS FOLLOWS      CPK-LGT.
  AIRCRAFT NO.      FLT. HOURS
  FOR I = MCZN TO NC7
  DO
  LET PF = 1.0 - WZ(I)
  START NEW OUTPUT LINE
  WRITE PFID(1), PFYIM(1), CLGT(1), PF AS S 21, I 3, S 24, I 3, S 21.
  D(6,2), S 23, L(1,2)
  LOOP
  SKIP 1 OUTPUT LINE
  LET SNSIC = SNSIC + NSIC
  PRINT 1 LINE WITH NSIC AS FOLLOWS
  NUMBER OF SPECIAL INSPECTIONS CONDUCTED: **
  LET SNRD = SNRD + NRD
  PRINT 1 LINE WITH NRD AS FOLLOWS
  NUMBER OF STRUCTURAL MODIFICATIONS: **

```


-273-

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2 03/02/78. 18.36.02. PAGE 182

XDCR	GLOBAL VARIABLE	PEAL	1 REFS
XSCA	GLOBAL VARIABLE	PFAL	1 REFS
XSCR	GLOBAL VARIABLE	PFAL	1 REFS
XICO	GLOBAL VARIABLE	REAL	1 REFS
XICR	GLOBAL VARIABLE	REAL	1 REFS
XISD	GLOBAL VARIABLE	REAL	1 REFS
IARFL	GLOBAL VARIABLE	PFAL	2 REFS
IARCO	GLOBAL VARIABLE	(I-D) REAL	6 REFS

```

1 ROUTINE SUMMARY
2 DEFINE RAGE AS AN INTEGER VARIABLE
3 START NEW PAGE
4 WRITE MODFL(1), MODEL(2) AS S 50. "AIRCRAFT TYPE: ", 2 A 4
5 SKIP 2 OUTPUT LINES
6 PRINT 1 DOUBLE LINE WITH SIZE OF FLEET, USAGE, LIFE AS FOLLOWS
  NUMBER OF AIRCRAFT IN FLEET: *****
  AIRCRAFT SERVICE LIFE: ***** HOURS
7 SKIP 1 OUTPUT LINE
8 WRITE SEL1, SEL2, SFL3 AS S 45. "SUMMARY OF STRUCTURAL ELEMENT: ", 2 A 4.
9 A 3
10 SKIP 2 OUTPUT LINES
11 PRINT 1 DOUBLE LINE AS FOLLOWS
  NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

```

```

12 SKIP 1 OUTPUT LINE
13 IF GOICR GT 0 LET PSICR = SICR LET PLICR = LICR ALWAYS
14 IF GOCOR GT 0 LET PSICO = SICO LET PLICO = LICO ALWAYS
15 IF GOSUM GT 0 LET PSISD = SISD LET PLISD = LISD ALWAYS
16 PRINT 6 DOUBLE LINES WITH GOICR, GOCOR, GOSUM, PSICR, PSICO, PSISD,
  PLICR, PLICO, PLISD, AICR, AICO, ALSO AS FOLLOWS
17 FIRST CRACK CORROSION SERVICE DAMAGE PRODUCTION DEFECTS
  -----
  OCCURRENCES *****
  MIN(HRS) *****
  MAX(HRS) *****
  AVG(HRS) *****

```

```

18 SKIP 2 OUTPUT LINES
19 PRINT 1 DOUBLE LINE AS FOLLOWS
  NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION
20 SKIP 1 OUTPUT LINE
21 IF GOICR(1) GT 0 LET PSACR = SACR LET PLACR = LACR ALWAYS
22 IF GOICR(2) GT 0 LET PSOCR = SOCR LET PLOCR = LOCR ALWAYS
23 IF GOICR(3) GT 0 LET PSOCR = SOCR LET PLOCR = LOCR ALWAYS
24 IF GOICR(4) GT 0 LET PSOCR = SOCR LET PLOCR = LOCR ALWAYS
25 IF GOSCR GT 0 LET PSOCR = SOCR LET PLOCR = LOCR ALWAYS
26 PRINT 6 DOUBLE LINES WITH GOICR(1), GOICR(2), GOICR(3), GOICR(4), GOSCR,
  PSACR, PSOCR, PSOCR, PSDCR, PSSCR, PLACR, PLOCR, PLDCR, PLDCR, PLDCR,
  AOCR, AOCR, AOCR, AOCR AS FOLLOWS
27 A-LEVEL R-LEVEL C-LEVEL D-LEVEL SPECIAL
28 -----
  OCCURRENCES *****
  MIN(IN) *****
  MAX(IN) *****
  AVG(IN) *****

```

```

29 SKIP 2 OUTPUT LINES
30 PRINT 1 DOUBLE LINE AS FOLLOWS
  NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION
31 SKIP 1 OUTPUT LINE
32 IF GOICR(1) GT 0 LET PSACA = SACA LET PLACA = LACA ALWAYS
33 IF GOICR(2) GT 0 LET PSACA = SACA LET PLACA = LACA ALWAYS
34 IF GOICR(3) GT 0 LET PSACA = SACA LET PLACA = LACA ALWAYS
35 IF GOICR(4) GT 0 LET PSACA = SACA LET PLACA = LACA ALWAYS
36 IF GOSCO GT 0 LET PSACA = SACA LET PLACA = LACA ALWAYS
37 PRINT 6 DOUBLE LINES WITH GOICR(1), GOICR(2), GOICR(3), GOICR(4), GOSCO,
  PSACA, PSACA, PSACA, PSACA, PSACA, PSACA, PLACA, PLACA, PLACA, PLACA,
  AACA, AACA, AACA, AACA AS FOLLOWS

```

PSACA, PSRCA, PSSCA, PSDCA, PSSCA, PLACA, PLBCA, PLCCA, PLDCA, PLSCA,
AACA, ARCA, ACCA, ADCA, ASCA AS FOLLOWS

A--LEVEL
R--LEVEL
C--LEVEL
D--LEVEL
SPECIAL

OCCURRENCES
MIN(SQ.IN)
MAX(SQ.IN)
AVG(SQ.IN)

SKIP 1 OUTPUT LINE

PRINT 1 LINE 75 FOLLOWS

INSPECTION INTERVALS(MKS)

PRINT 1 DOUBLE LINE WITH LABCD(1), LABCD(2), LABCD(3), LABCD(4) AS FOLLOWS

INITIAL PRINT 1 DOUBLE LINE WITH LABCD(1), LABCD(2), MCINS, MDINS AS FOLLOWS

SMOFTST PRINT 1 DOUBLE LINE WITH LABCD(1), LABCD(2), XCINS, XDINS AS FOLLOWS

LONGEST PRINT 1 DOUBLE LINE WITH LABCD(1), LABCD(2), XCINS, XDINS AS FOLLOWS

SKIP 1 OUTPUT LINE

PRINT 1 LINE WITH SNISIC AS FOLLOWS

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: ***

PRINT 1 LINE WITH SNMID AS FOLLOWS

NUMBER OF STRUCTURAL MODIFICATIONS: ***

PRINT 1 LINE WITH SNMID AS FOLLOWS

NUMBER OF AIRCRAFT MODIFIED IN SERVICE: *****

IF FAIL.OPT = 1 OR FAIL.OPT = 3

WRITE AV.EL.FAIL. AS S 10, "ESTIMATED ELEMENT TYPE FAILURE RATE USING AVG:M,

E(10.2),"/HR"

START NEW OUTPUT LINE

ALWAYS

IF FAIL.OPT = 2 OR FAIL.OPT = 5

WRITE ELTYP.FAIL.RATE AS S 10, "ESTIMATED ELEMENT TYPE FAILURE RATE:M,E(10.2),

"MHR."

START NEW OUTPUT LINE

PRINT 1 LINE WITH AVGL, STDL AS FOLLOWS

SAMPLE CRK. LGT. MEAN(IN) ****, SAMPLE STD. DEV. ****

PRINT 1 DOUBLE LINE WITH AEXP, RA AS FOLLOWS

CRK. LGT. VS PROBABILITY CURVE FIT CONST: A = ****, ***** B = ****

ALWAYS

SKIP 1 OUTPUT LINE

PRINT 1 DOUBLE LINE AS FOLLOWS

STRUCTURAL FAILURES

PRINT 1 DOUBLE LINE AS FOLLOWS

AIRCRAFT NO. STA. NO.

PRINT 1 DOUBLE LINE AS FOLLOWS

FLY. HOURS

LET RANGE = MAX(F(SNRF5,SMSFL))

FOR I = 1 TO RANGE

DO

IF SMSFL GE I AND SNRF5 GE I

WRITE SACTD(1), SELM(1), SSTIM(1), SSTIM(1), SELM(1) AS S 13,

I 5, S 14, I 6, S 12, A 4, S 24, I 5, S 14, I 5, S 13, A 4

JUMP AHEAD

RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH

AIRCRAFT NO. FLY. HOURS STA. NO.

```

73      ELSE
74      MP
75      IF SHRFS GE 1
76      WRITE SAPIO(I), SSTIM(I), SELNB(I) AS S 76, I 5, S 14, I 6, S 13, A 4
77      JUMP AHEAD
78      ELSE
79      WRITE SACID(I), SFLTHR(I), SSTAN(I) AS S 13, I 5, S 14, I 6, S 12, A 4
80      HERE
81      START NEW LINE
82      LOOP
83      SKIP 2 OUTPUT LINES
84      PRINT 1 LINE WITH AVFSA(1),AVFSA(2),AVFSA(3),AVFSA(4),AVFSA(5) AS FOLLOWS
      AVERAGE FLIGHT CRACKS **.* **.* **.* **.* **.*
85      PRINT 1 LINE WITH AVPSA(1),AVPSA(2),AVPSA(3),AVPSA(4),AVPSA(5) AS FOLLOWS
      AVERAGE PRESSURE CRACKS **.* **.* **.* **.* **.*
86      PRINT 1 LINE WITH FSAVE(1),FSAVE(2),FSAVE(3),FSAVE(4),FSAVE(5) AS FOLLOWS
      FLIGHT CRACKS **.* **.* **.* **.* **.*
87      PRINT 1 LINE WITH PSAVE(1),PSAVE(2),PSAVE(3),PSAVE(4),PSAVE(5) AS FOLLOWS
      PRESSURE CRACKS **.* **.* **.* **.* **.*
88      RETURN
89      END

```

SYMBOLIC REFERENCE MAP (R = 1) - ROUTINE SUMMARY

AACA	PROCEDURE	REAL	1 REFS
AACR	PROCEDURE	REAL	1 REFS
ABCA	PROCEDURE	REAL	1 REFS
ABCP	PROCEDURE	REAL	1 REFS
ABCA	PROCEDURE	REAL	1 REFS
ACCR	PROCEDURE	REAL	1 REFS
ADCA	PROCEDURE	REAL	1 REFS
ADCP	PROCEDURE	REAL	1 REFS
AEAP	GLOBAL VARIABLE	REAL	1 REFS
ASCA	PROCEDURE	REAL	1 REFS
ASCP	PROCEDURE	REAL	1 REFS
AVESA	GLOBAL VARIABLE	REAL	1 REFS
AVEL	GLOBAL VARIABLE	REAL	1 REFS
AVFA	GLOBAL VARIABLE	REAL	1 REFS
AV-EL-FAIL	GLOBAL VARIABLE	REAL	1 REFS
AICO	PROCEDURE	REAL	1 REFS
AICP	PROCEDURE	REAL	1 REFS
AISO	PROCEDURE	REAL	1 REFS
MA	GLOBAL VARIABLE	REAL	1 REFS
ELTYP-FAIL-RATE	GLOBAL VARIABLE	REAL	1 REFS
FAIL-OUT	GLOBAL VARIABLE	REAL	1 REFS
FSAVE	GLOBAL VARIABLE	REAL	1 REFS
GOCOR	GLOBAL VARIABLE	REAL	1 REFS
GOICO	GLOBAL VARIABLE	REAL	1 REFS
GOICR	GLOBAL VARIABLE	REAL	1 REFS
GOPO	GLOBAL VARIABLE	REAL	1 REFS
GOSCO	GLOBAL VARIABLE	REAL	1 REFS
GOSCR	GLOBAL VARIABLE	REAL	1 REFS
GOSOM	GLOBAL VARIABLE	REAL	1 REFS
GOICR	GLOBAL VARIABLE	REAL	1 REFS
I	LOCAL RECURSIVE	REAL	1 REFS
LACA	GLOBAL VARIABLE	REAL	1 REFS
LACR	GLOBAL VARIABLE	REAL	1 REFS
LBCA	GLOBAL VARIABLE	REAL	1 REFS
LCBR	GLOBAL VARIABLE	REAL	1 REFS
LCCA	GLOBAL VARIABLE	REAL	1 REFS
LCCP	GLOBAL VARIABLE	REAL	1 REFS
LDCA	GLOBAL VARIABLE	REAL	1 REFS
LDCP	GLOBAL VARIABLE	REAL	1 REFS
LSCA	GLOBAL VARIABLE	REAL	1 REFS
LSCP	GLOBAL VARIABLE	REAL	1 REFS
LICO	GLOBAL VARIABLE	REAL	1 REFS
LICP	GLOBAL VARIABLE	REAL	1 REFS
LISD	GLOBAL VARIABLE	REAL	1 REFS
MAX-F	PROCEDURE	REAL	1 REFS
MODEL	GLOBAL VARIABLE	REAL	1 REFS
NCINS	GLOBAL VARIABLE	REAL	1 REFS
NDINS	GLOBAL VARIABLE	REAL	1 REFS
PLACA	LOCAL RECURSIVE	REAL	1 REFS
PLCAC	LOCAL RECURSIVE	REAL	1 REFS

CDC 6600 CACI SIMSCRIPT II.5 VERSION /4.0-00/ KRONOS 2.1.2

PLNCR	LOCAL RECURSIVE	< 27	REAL	2 REFS
PLCCA	LOCAL RECURSIVE	< 41	REAL	2 REFS
PLCCR	LOCAL RECURSIVE	< 29	REAL	2 REFS
PLDCA	LOCAL RECURSIVE	< 43	REAL	2 REFS
PLDCR	LOCAL RECURSIVE	< 31	REAL	2 REFS
PLSCA	LOCAL RECURSIVE	< 45	REAL	2 REFS
PLSCR	LOCAL RECURSIVE	< 33	REAL	2 REFS
PLICO	LOCAL RECURSIVE	< 19	REAL	2 REFS
PLICR	LOCAL RECURSIVE	< 17	REAL	2 REFS
PLISD	LOCAL RECURSIVE	< 21	REAL	2 REFS
PSACA	LOCAL RECURSIVE	< 36	REAL	2 REFS
PSACR	LOCAL RECURSIVE	< 24	REAL	2 REFS
PSAVE	GLOBAL VARIABLE	(1-0)	REAL	5 REFS
PSBCA	LOCAL RECURSIVE	< 38	REAL	2 REFS
PSBCR	LOCAL RECURSIVE	< 26	REAL	2 REFS
PSCCA	LOCAL RECURSIVE	< 40	REAL	2 REFS
PSCCR	LOCAL RECURSIVE	< 28	REAL	2 REFS
PSDCA	LOCAL RECURSIVE	< 42	REAL	2 REFS
PSDCR	LOCAL RECURSIVE	< 30	REAL	2 REFS
PSSCA	LOCAL RECURSIVE	< 44	REAL	2 REFS
PSSCR	LOCAL RECURSIVE	< 32	REAL	2 REFS
PSICO	LOCAL RECURSIVE	< 14	REAL	2 REFS
PSICR	LOCAL RECURSIVE	< 16	REAL	2 REFS
PSISD	LOCAL RECURSIVE	< 20	REAL	2 REFS
PNAGE	LOCAL RECURSIVE	< 1	INTEGER	3 REFS
SACA	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
SACID	GLOBAL VARIABLE	(1-0)	INTEGER	2 REFS
SACR	GLOBAL VARIABLE	(1-0)	REAL	1 REFS
SAPID	GLOBAL VARIABLE	(1-0)	INTEGER	2 REFS
SBCA	GLOBAL VARIABLE		REAL	1 REFS
SBCR	GLOBAL VARIABLE		REAL	1 REFS
SCCA	GLOBAL VARIABLE		REAL	1 REFS
SCCR	GLOBAL VARIABLE		REAL	1 REFS
SDCA	GLOBAL VARIABLE		REAL	1 REFS
SDCP	GLOBAL VARIABLE		REAL	1 REFS
SELNB	GLOBAL VARIABLE	(1-0)	ALPHA	2 REFS
SELI	GLOBAL VARIABLE		ALPHA	1 REFS
SELF	GLOBAL VARIABLE		ALPHA	1 REFS
SEL3	GLOBAL VARIABLE	(1-0)	ALPHA	1 REFS
SELTHR	GLOBAL VARIABLE	(1-0)	INTEGER	2 REFS
SIZE.OF.FLEET	GLOBAL VARIABLE		INTEGER	1 REFS
SND	GLOBAL VARIABLE		INTEGER	1 REFS
SNRFS	GLOBAL VARIABLE		INTEGER	3 REFS
SNRFL	GLOBAL VARIABLE		INTEGER	2 REFS
SNRSL	GLOBAL VARIABLE		INTEGER	1 REFS
SNRSD	GLOBAL VARIABLE		INTEGER	1 REFS
SSCA	GLOBAL VARIABLE		REAL	1 REFS
SSCR	GLOBAL VARIABLE		REAL	1 REFS
SSTAN	GLOBAL VARIABLE	(1-0)	ALPHA	2 REFS
SSTIM	GLOBAL VARIABLE	(1-0)	INTEGER	2 REFS
STOL	GLOBAL VARIABLE		REAL	1 REFS
SUMMARY	PROCEDURE		INTEGER	1 REFS
SICO	GLOBAL VARIABLE		REAL	1 REFS

CDC 6488	CACI SIMSCRIPT II.5	VERSION /4.0-00/	KRONOS 2.1.2	93/02/78. 15.36.02.	PAGE 188
SICR	GLOBAL VARIABLE		REAL	1 REFS	
SISD	GLOBAL VARIABLE		REAL	1 REFS	
USAGE.LIFE	GLOBAL VARIABLE		REAL	1 REFS	
XCINS	GLOBAL VARIABLE		REAL	1 REFS	
ADINS	GLOBAL VARIABLE		REAL	1 REFS	
LABCD	GLOBAL VARIABLE		REAL	1 REFS	
		(1-D)	REAL	8 REFS	