AFFDL-TR-76-36
VOLUME II

DEVELOPMENT OF RELIABILITY-BASED AIRCRAFT SAFETY CRITERIA: AN IMPACT ANALYSIS
VOLUME II: COMPUTER MANUAL

MODERN ANALYSIS INC.
RIDGEWOOD, NEW JERSEY

APRIL 1976

TECHNICAL REPORT, AFFDL-TR-76-36, VOLUME II
FINAL REPORT FOR PERIOD APRIL 1975 - APRIL 1976

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

This technical report has been reviewed and is approved for publication.

Robert L. Neulieb
ROBERT L. NEULIEB
Project Engineer

R.M. Bader
ROBERT M. BADER, Chief
Structural Integrity Br
Structures Division

FOR THE COMMANDER

Howard L. Farmer
HOWARD L. FARMER, Colonel, USAF
Chief, Structural Mechanics Division

AIR FORCE - 29 OCTOBER 76 - 300

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-76-36, Volume II	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): DEVELOPMENT OF RELIABILITY-BASED AIRCRAFT SAFETY CRITERIA: AN IMPACT ANALYSIS	5. TYPE OF REPORT & PERIOD COVERED: Final Report April, '75-April, '76
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Masanobu Shinozuka	8. CONTRACT OR GRANT NUMBER(s): F33615-75-C-3066
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Modern Analysis Inc. 229 Oak Street Ridgewood, N.J. 07450	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 13670123
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433	12. REPORT DATE: April 1976
	13. NUMBER OF PAGES: 80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Safety, Reliability, Durability, Damage Tolerant Design, Fail Safe Design, Random Loading, Rise and Fall of Random Processes, Crack Propagation, Fracture Mechanics, Proof Load Testing, Full-Scale Testing, Inspections, NDI Techniques.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The random stress processes, composite Gaussian or single Gaussian, are constructed on the basis of exceedance curves for different aircraft under various flight conditions. The fatigue crack propagations under such stress processes are estimated with the aid of fracture mechanics method. The crack propagation rate under random loading is assumed to be proportional to the expected value of a power of the range of stress intensity factor which is in turn

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) proportional to the expected value of the same power of the rise and fall of the stress process involved. The residual strength is evaluated as a function of crack size either under the assumption of slow crack growth design or under the assumption of fail safe design. Two different models as to the basic mechanical nature of the crack are used for the analysis and their interrelationships are discussed. One of these is the crack initiation model in which a crack of certain size is assumed to initiate at "randomly distributed time" t_o and then propagate in accordance with the propagation law of fracture mechanics. The other is the pre-existing crack model which assumes pre-existence of cracks of random size. With the aid of the random process theory, the failure rate is evaluated as the expected rate of upcrossing of the residual strength by the stress process. The probability of aircraft failure is obtained from the failure rate taking into consideration the effect of inspection procedures and proof loads. The latter, however, is considered only in conjunction with pre-existing crack model. The significance of full-scale testing is also reviewed from the veiw point of reliability implementation and demonstration. Numerical examples are worked out to illustrate the numerical significance of various analytical models and to indicate the sensitivity of the probability of failure to the parameters involved in the models. This volume contains computer program lists, input and output descriptions, input and output samples, flow charts and other data necessary for numerical work associated with the tasks described above.

FOREWORD

This users manual report was prepared by Modern Analysis Inc., Ridgewood, N.J., under Air Force Contract No. F33615-75-C-3066. The work was monitored by the Air Force Flight Dynamics Laboratory, FBE, Air Force Systems Command, United States Air Force, Wright-Patterson Air Force Base, Ohio, project 1367, task number 01, with Dr. Robert L. Neulieb (AFFDL/FBE) as Project Engineer.

The project was conducted under the general direction of Dr. M. Shinozuka, President, Modern Analysis Inc.

Dr. R. Vaicaitis and Mr. P. Wai, provided the programming and documentation efforts. The programs documented herein are written for operation on a CDC6600 computer system.

The work reported herein was performed during the period April 1975 to March 1976. The final report in two volumes was submitted on April 30, 1976.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONA

LIST OF TABLES

SECTION 1

INTRODUCTION

The probabilities of first aircraft failure under a prescribed mission loading and a number of rigorous and cursory inspections are calculated analytically. Three programs were written for this purpose. The first program, called Maneuver Loading (MANLOD), calculates the average of rise and fall of the stress process due to aircraft maneuvers or gust utilizing a simulation technique of random processes. The results from this program serve as an input to the other two programs. The second program, called Aircraft Reliability when Time to Crack Initiation is a Random Variable (AREL1), evaluates probabilities of first aircraft failure based on the condition that time to fatigue crack initiation is distributed according to a two-parameter Weibull distribution. The third program, called Aircraft Reliability when Initial Cracks are Random Variables (AREL2), calculates probabilities of first aircraft failure based on the initial crack size distribution model. The following sections of this document discuss these three programs, their options, the input data, and the output.

SECTION 2

MANEUVER LOADING PROGRAM (MANLOD)

The purpose of MANLOD program is to determine the rise and fall of the stress response process due to aircraft maneuvers or atmospheric turbulence. The results from this program serve as an input to the aircraft reliability programs AREL1 and AREL2 which are described in Sec. 3 and 4. A simplified flow chart of the MANLOD program is shown in Figure 1.

2.1 SIMULATION AND RANGE COUNT

The time history of stress response due to aircraft maneuvers or gust is generated from the specified stress spectral densities utilizing simulation techniques of Gaussian random processes. For the purpose of this study, the stress history due to gust is generated from a specified normalized spectral density function corresponding to a truncated Gaussian white noise spectra. The area under this spectral density is unity. The width of this spectra is controlled by a parameter called BETA which ranges $0.0 \leq \text{BETA} < 1$. For example, BETA = 0 corresponds to a wide band process and BETA = 0.9 to a narrow band process. The stress time history due to aircraft maneuvers is generated from the specified spectral density in Function FUNSPT. If the spectral density due to maneuvers is specified in terms of accelerations g^2, the conversion to stress spectral density is achieved through the parameter g. To account for the unsymmetric exceedance observed in actual maneuver exceedance

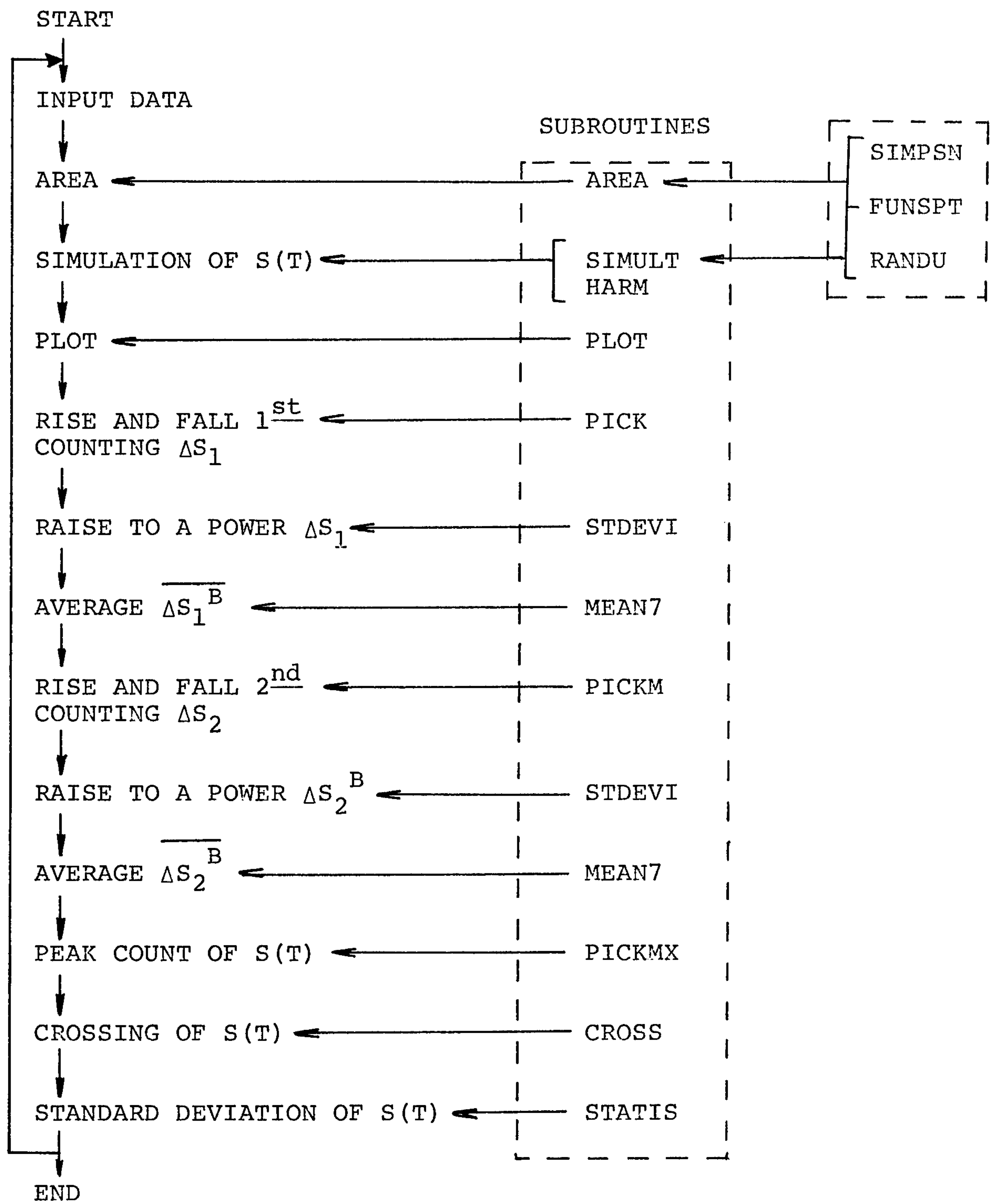
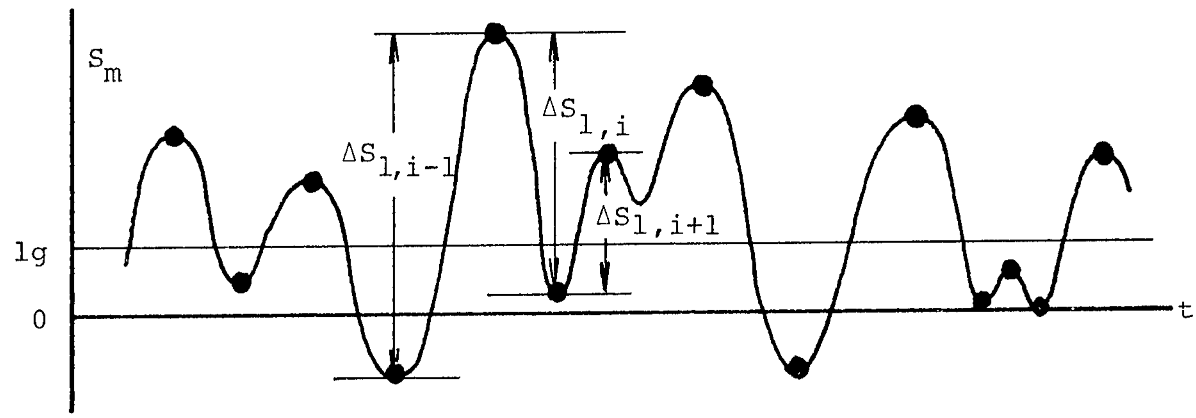


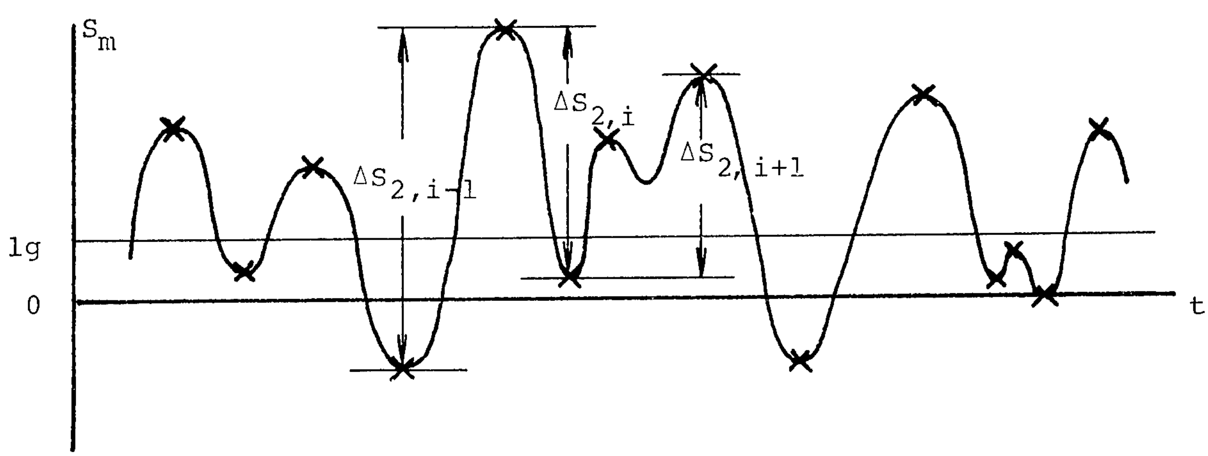
Figure 1. Simplified Flow Chart of MANLOD Program

curves, the stress time history is modified by multiplying the negative portion by a factor called REDUCE = standard deviation for negative maneuvers/standard deviation for positive maneuvers.

The rise and fall, ΔS_i, of the simulated process $S_m(t)$ are determined using two separate range count methods. In the first method, the $\Delta S_{1,i}$ values are determined by selecting the difference



between the consecutive peak and valley as shown above. In the second counting method, the $\Delta S_{2,i}$ values are selected from the difference between those peaks and valleys which are separated by the 1g line as shown below.



2.2 INPUT DATA

There are no input data cards used for the Maneuver Loading Program. The necessary data parameters need to be specified at the beginning of the main program. The input spectral density function of the maneuver loading in g^2 units need to be specified. An option is provided to use a normalized truncated white noise spectrum. The input parameters are as follow: maneuver loading spectral density function SS(W) with parameters A, FB and PHIO where A = a shape parameter; FB = break frequency in Hz; PHIO = peak value of the spectral density function SS(W); IX = an odd integer with less than 9 digits where by changing IX a new sample function is generated; N = number of frequency subdivisions on the spectral density function SS(W) in powers of 2; NPT = a number of simulated points of the output sample function specified by NPT = 2 ** M1 with 2 * N being equal to or less than NPT, WU = the upper cut-off frequency on SS(W); WL = the lower cut-off frequency on SS(W); M1 = a positive integer such that NPT = 2 ** M1, NSIGN = an option parameter where for NSIGN = 0 a normalized white noise spectral density with a unit standard deviation is provided by the program, and for NSIGN = 1 the user must supply a spectral density for the maneuver loading or use the one specified in the subroutine FUNSPT; REDUCE = ratio of the standard deviations such that REDUCE = σ_n/σ_p where σ_n is the standard deviation determined from the time history below the 1g level and σ_p is the standard deviation obtained from the time history above the 1g level; NP = number of points plotted by the subroutine PLOT; BETA = parameter which specifies the width of the truncated Gaussian white noise spectrum; NSAMPL = number of samples to be simulated.

2.3 PROGRAM DESCRIPTION

The Maneuver Loading Program consists of a main program unit and fourteen subroutines. The program listing is given in Appendix A. The main program unit called MANLOD and the fourteen subroutines: AREA, SIMULT, PLOT, PICK, STDEVI, MEAN7, PICKM, PICKMX, CROSS, SIMPSN, FUNSPT, HARM, RANDU, STATIS are described as follows.

2.3.1 The Main Program Unit MANLOD

The main program initializes the input data and with a call to the subroutine AREA calculates the area under the maneuver loading spectral density specified in Function FUNSPT. The time realizations of the loading (or stress) are generated by calling subroutine SIMULT. When NSIGN = 0, stress time history corresponding to gust input is provided. When NSIGN = 1, stress time history due to maneuvers is given. The rise and fall of the simulated time history are determined by calling subroutine PICK for the first counting method and subroutine PICKM for the second counting method. The rise and fall values are raised to an arbitrary power B ranging from 1 to 10 by subroutine STDEVI. Then, the average of rise and fall raised to B power is calculated by calling subroutine MEAN7. The total number of peaks of the simulated process $S_m(t)$ is counted by the subroutine PICKMX and the crossing rate is determined from subroutine CROSS. A subroutine PLOT is called to plot the output time history for one realization.

2.3.2 Subroutine AREA (S, N, WUP, WL)

Subroutine AREA calculates the area under the curve specified in FUNSPT. It utilizes a Simpson rule type integration procedure given in Function SIMPSN. The arguments for this subroutine are

S = input data from FUNSPT;

N = number of subdivisions on spectral density SS;

WUP = upper cut-off frequency on SS;

WL = lower cut-off frequency on SS.

2.3.3 Subroutine SIMULT (A, N, NPT, WU, WL, IX, M1, NSIGN, INV, S)

The subroutine SIMULT is used to generate samples of the time history corresponding to gust or maneuver loading. The program utilizes the Fast Fourier Transform (FFT) algorithm by calling subroutine HARM. The essential equations for this simulation are

Real A(2 * J + 1) = SQRT(2.0 * SP) * COS(PHI)

Imaginary A(2 * J + 2) = SQRT(2.0 * SP) * SIN(PHI)

where J = 1, 2, . . ., N, SP = specified spectral density, PHI = random phase angles uniformly distributed between 0 and 2π which are generated by subroutine RANDU. The arguments for this subroutine are A = specified by the above equations; N = number of subdivisions on spectral density SP; NPT = number of time points to be simulated; WU = upper cut-off frequency on SP; WL = lower cut-off frequency on SP; IX = a number specified in the input; NSIGN = specified in the input; INV, S = internal parameters of the subroutine.

2.3.4 Subroutine HARM (A, M, INV, S, IFSET, IFERR)

Subroutine HARM perform a Fast Fourier Transform (FFT) on the specified function A(I), (I = 1, 2, . . ., N). The arguments for this subroutine are

A = generated input by the subroutine SIMULT;

M = number of points to be simulated, i.e., for one dimensional case M(1) = M1, M(2) = 0, M(3) = 0.

INV, S = internal parameters of the subroutine;

IFSET = 1, implies a forward Fourier Transform, IFSET = -1 implies an inverse Fourier Transform;

IFERR = internal parameter of the subroutine.

2.3.5 Subroutine PLOT (NP, DT, A, 0.0, M)

Subroutine PLOT plots a given function A with a dimension NP at step intervals DT. The argument M indicates how many points from given NP needs to be plotted. For example, if M = 1, it implies that every point is plotted, if M = 2, it implies that every second point is plotted.

2.3.6 Subroutine PICK (A, NPT, S, NPK(J))

Subroutine PICK determines the rise and fall ΔS_{li} corresponding to the first counting procedure as described in Sec. 2.1. The arguments for PICK program are

A = input array from which rise and fall needs to be found;

NPT = number of points in the input array A;

S = an array in which the rise and fall are stored;

NPK = number of rise and fall values found from the specified input A.

2.3.7 Subroutine STDEVI (NST, NPK(J), S, STD, RMS, A1(J), A2(J), . . ., A5(J), AN)

This subroutine raises the rise and fall values calculated in PICK to powers A1, A2, . . ., A5, where

NST = parameter calculated by the main program;

NPK = number of rise and fall values;

S = rise and fall;

STD, RMS, AN = parameters calculated inside the subroutine.

2.3.8 Subroutine MEAN7 (NSAMPL, A1, . . ., A5, AM1, . . ., AM5, AA, AN)

Subroutine MEAN7 calculates the average of rise and fall raised to arbitrary power in subroutine STDEVI.

NSAMPL = sample size for which average is calculated;

A1, . . ., A5 = parameters obtained in subroutine STDEVI;

AM1, . ., AM5 = calculated in the subroutine;

AA = calculated in the main program;

AN = calculated in the subroutine

2.3.9 Subroutine PICKM (A, NPT, S, MPK(J))

This subroutine calculates the rise and fall ΔS_{2i} corresponding to the second counting procedure as described in Sec. 2.1. The meaning of the arguments is the same as for subroutine PICK.

2.3.10 Subroutine PICKMX (Y, NPT, PMAX, LPEAK, PMIN, MTROG)

Subroutine PICKMX picks all the peaks and troughs from a given process.

NPT = number of points;

Y = input process;

PMAX = array for peak values;

PMIN = array for trough values;

LPEAK = number of peaks found;

MTROG = troughs found.

2.3.11 Subroutine CROSS (STDEV, PMAX, PMIN, LLL, MMM, VZERO, RATIO)

This subroutine determines the crossing of the threshold level specified by STDEV for a given process.

STDEV = standard deviation of the given process.

PMAX = peak value array of the process;

PMIN = trough value array of the process;

LLL = number of peaks;

MMM = number of troughs;

VZERO = indicates zero threshold crossing;

RATIO = when RATIO = 1, crossing for the total given time history is determined, when RATIO = 0, crossing per hour is found.

2.3.12 Subroutine SIMPSN (AF, NPT, DSTEP, SIMP)

Subroutine SIMPSN integrates a given function AF utilizing a Simpson rule integration procedure.

NPT = number of points to be integrated (must be an odd number)

DSTEP = integration step;

SIMP = resulting area of integration.

2.3.13 Subroutine RANDU (IX, IY, YFL)

Subroutine RANDU generates independent random numbers uniformly distributed between 0 and 1. By multiplying these numbers by 2π, random phase angles, which are necessary for the digital simulation procedure, are obtained. An odd integer, IX, needs to be specified in the main program. By changing IX, a new seed of random numbers is generated.

IX = any odd integer less than 9 digits;

IY = internal parameter in the program;

YFL = output - random numbers between 0 and 1.

2.3.14 Subroutine FUNSPT (SP, W1)

Subroutine FUNSPT is used to array the normalized empirical spectral density function corresponding to aircraft maneuvers.

SS(W) = 2π * PHIO/(1 + (2π * W/FB) ** A)/ANORML

where

W = frequency in HZ;

FB, A, PHIO = data parameters;

ANORML = area under SS which is determined in subroutine AREA.

2.3.15 Subroutine STATIS (A, NP, STD, RMS)

Subroutine STATIS determines the standard deviation and rms of maneuver loading.

A = input array;

NP = number of points;

STD = output - standard deviation;

RMS = output - root mean square.

2.4 PROGRAM OUTPUT

The MANLOD program generates printed output. In Table 1, the average of rise and fall for one realization is given using the first range counting method described in Sec. 2.1.

TABLE 1. RISE AND FALL
(FIRST COUNTING METHOD)

```
BETA= 0.0          NPEAK=  2285       WU=   0.2000       WL=   0.0070
MEAN1=   0.692    MEAN2=   0.907    MEAN3=  1.547    MEAN35=   2.163    MEAN4=  3.129
MEAN6= 0.17808E 02  MEAN7= 0.48034E 02  MEAN8= 0.13826E 03  MEAN9= 0.42070E 03

 MEAN45=    4.661   MEAN5=    7.126
MEAN10= 0.13425E 04

STDEV= 0.6505576E 00       RMS= 0.9499773E 00
```

NPEAK = number of rise and fall;

WU = upper cut-off frequency on spectral density;

WL = lower cut-off frequency on spectral density;

STDEV = standard deviation of the rise and fall;

RMS = root mean square of the rise and fall;

MEAN1, MEAN2. . . ., MEAN10 = indicates the mean values if rise and fall which were raised to powers 1, 2, . . ., 10, respectively.

Similar results are shown in Table 2 for the second range counting method. The standard deviation and rms of $S_m(t)$ are also included in this table.

TABLE 2. RISE AND FALL
(SECOND COUNTING METHOD)

```
BETA= 0.0          NPEAK=  1000      WJ=  0.2000      WL=  0.0070
MEAN1=  1.292   MEAN2=  2.274   MEAN3=  4.791   MEAN35=  7.295
MEAN6= 0.84403E 02  MEAN7= 0.25703E 03  MEAN8= 0.83775E 03

  STDEV= 0.7776743E 00      RMS= 0.1508034E 01
  MEAN4= 11.401    MEAN45=  18.237    MEAN5=  29.802
MEAN9= 0.29053E 04  MEAN10= 0.10559E 05
```

```
STANDARD DEVIATION OF THE TIME HISTORY
STDEV= 0.6485804E 00      R.M.S.= 0.7130958E 00
```

The average of rise and fall corresponding to averaged eight realizations is given in Table 3 for the first range counting method and in Table 4 for the second range counting method. The number of rise and fall for each method is also indicated in these tables.

TABLE 3. AVERAGE OF POWER OF RISE AND FALL
(FIRST COUNTING METHOD)

CASE I

AVERAGE OF POWER OF RISE AND FALL
AND NUMBER OF RISE AND FALL

BETA	POWER 1.0	2.0	3.0	3.5	4.0	4.5	5.0	RISE & FALL
0.0	0.696	0.933	1.670	2.397	3.568	5.479	8.645	2269

BETA	6.0	7.0	8.0	9.0	10.	PEAKS
0.0	23.05	56.38	203.68	659.41	2235.61	2269

TABLE 4. AVERAGE OF POWER OF RISE AND FALL
(SECOND COUNTING METHOD)

CASE II

AVERAGE OF POWER OF RISE AND FALL
AND NUMBER OF RISE AND FALL

BETA	POWER 1.0	2.0	3.0	3.5	4.0	4.5	5.0	RISE & FALL
0.0	1.276	2.264	4.371	7.519	11.932	19.395	32.211	1031

BETA	6.0	7.0	8.0	9.0	10.	PEAKS
0.0	94.00	292.95	966.31	3350.72	12147.27	1031

The level of threshold crossing per hour against standard deviations of the process is given in Table 5.

TABLE 5. THRESHOLD CROSSING

NUMBER OF TOTAL CROSSINGS AT DIFFERENT LEVEL OF SIGMA,
INCLUDES CROSSINGS WITH BOTH +VE AND -VE SLOP

SIMULATED RESULT

-5.0SIGMA=	0	CROSSINGS
-4.8SIGMA=	0	CROSSINGS
-4.5SIGMA=	0	CROSSINGS
-4.4SIGMA=	0	CROSSINGS
-4.2SIGMA=	0	CROSSINGS
-4.0SIGMA=	0	CROSSINGS
-3.8SIGMA=	0	CROSSINGS
-3.6SIGMA=	0	CROSSINGS
-3.4SIGMA=	0	CROSSINGS
-3.2SIGMA=	0	CROSSINGS
-3.0SIGMA=	0	CROSSINGS
-2.8SIGMA=	0	CROSSINGS
-2.6SIGMA=	0	CROSSINGS
-2.4SIGMA=	0	CROSSINGS
-2.2SIGMA=	0	CROSSINGS
-2.0SIGMA=	0	CROSSINGS
-1.8SIGMA=	0	CROSSINGS
-1.6SIGMA=	0	CROSSINGS
-1.4SIGMA=	0	CROSSINGS
-1.2SIGMA=	0	CROSSINGS
-1.0SIGMA=	0	CROSSINGS
-0.8SIGMA=	4	CROSSINGS
-0.6SIGMA=	25	CROSSINGS
-0.4SIGMA=	84	CROSSINGS
-0.2SIGMA=	188	CROSSINGS
0.0SIGMA=	350	CROSSINGS
0.2SIGMA=	308	CROSSINGS
0.4SIGMA=	267	CROSSINGS
0.6SIGMA=	229	CROSSINGS
0.8SIGMA=	193	CROSSINGS
1.0SIGMA=	163	CROSSINGS
1.2SIGMA=	134	CROSSINGS
1.4SIGMA=	108	CROSSINGS
1.6SIGMA=	78	CROSSINGS
1.8SIGMA=	54	CROSSINGS
2.0SIGMA=	40	CROSSINGS
2.2SIGMA=	24	CROSSINGS
2.4SIGMA=	13	CROSSINGS
2.6SIGMA=	8	CROSSINGS
2.8SIGMA=	3	CROSSINGS
3.0SIGMA=	2	CROSSINGS
3.2SIGMA=	1	CROSSINGS
3.4SIGMA=	0	CROSSINGS
3.6SIGMA=	0	CROSSINGS
3.8SIGMA=	0	CROSSINGS
4.0SIGMA=	0	CROSSINGS
4.2SIGMA=	0	CROSSINGS
4.4SIGMA=	0	CROSSINGS
4.6SIGMA=	0	CROSSINGS
4.8SIGMA=	0	CROSSINGS
5.0SIGMA=	0	CROSSINGS

Section 3

AIRCRAFT RELIABILITY WHEN TIME TO CRACK INITIATION IS A RANDOM VARIABLE (AREL1)

The purpose of this program is to calculate the probability of first aircraft failure either of a single airplane or of an airplane in a fleet. The analytical probability of failure model is constructed for the case when time to fatigue crack initiation at a critical location on the aircraft is a random variable having a Weibull type distribution. The analytical model allows the inclusion of rigorous base type inspections which are performed at equal time intervals over the design life of the aircraft. A simplified flow chart of the program is shown in Figure 2. The following paragraphs discuss the input data, the structure of the program, and the output. The section on the structure of the program includes only the key analytical expressions. For a detailed description of all the equations, the reader should consult the Technical Report.

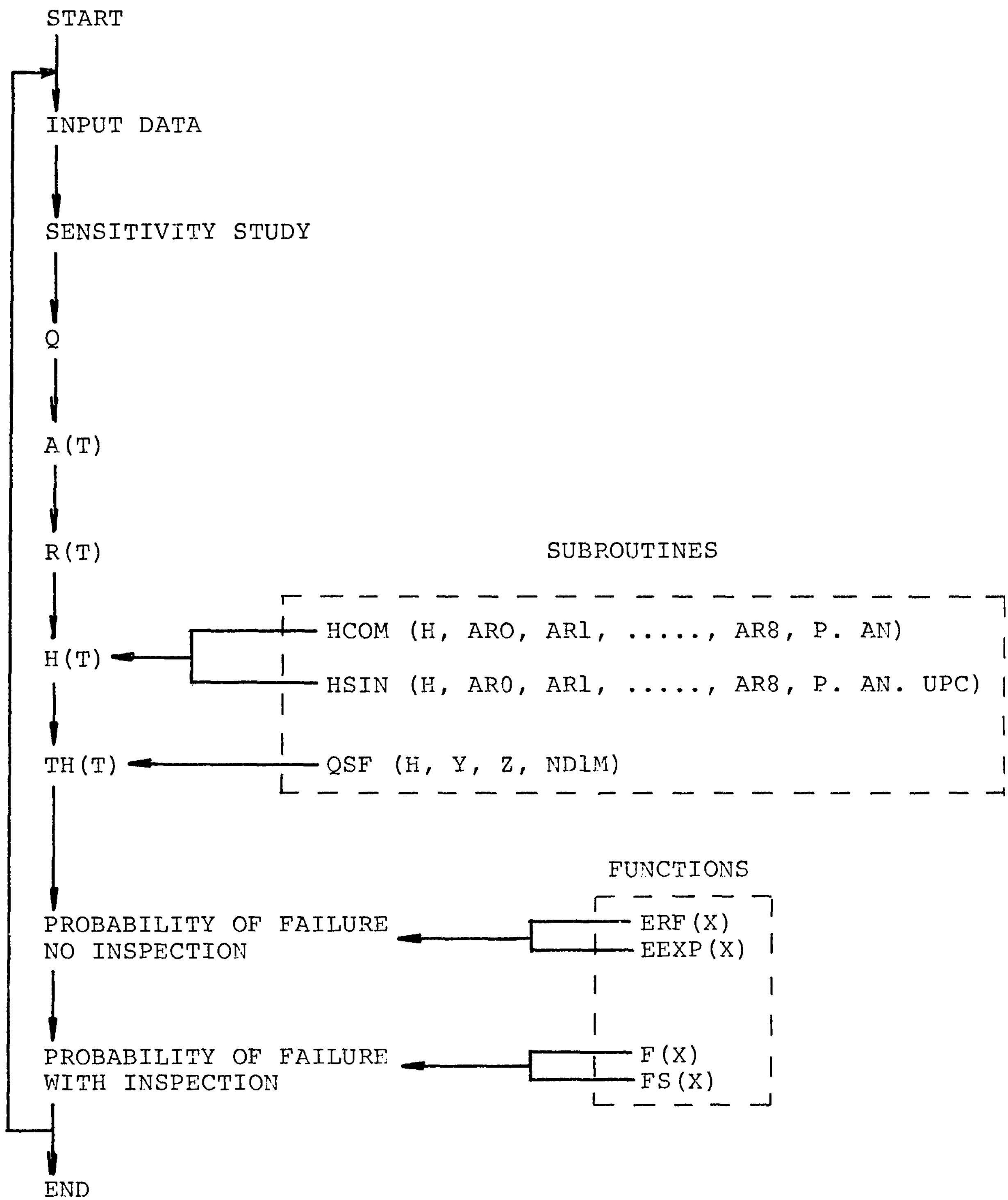


Figure 2. Simplicied Flow Chart of the AREL1 Program

3.1 INPUT DATA

There is one input data card used for the AREL1 program. The Format of this data card is illustrated in Table 6.

TABLE 6.

DATA CARD FORMAT FOR AREL1 PROGRAM

Column	Contents	Format
1-10	XY = a number from 1 to 32 which references the 32 parameters in the system	F 10.0
11-20	YZ1 = new input data if only one parameter is changed	F 10.0
21-30	YZ2 = new input data if two parameters are changed	F 10.0
31-40	YZ3 = new input data if three parameters are changed	F 10.0

The remaining data parameters are specified at the beginning of the main program. For the physical description of these parameters see the Technical Report.

The data are as follow: EF = λ, material constant; BE = b, a parameter in the fatigue crack propagation model; DKTH = threshold value of the range of stress intensity factor; EFD = λ', material constant, C = c, material constant: KC = critical stress intensity factor; AO = initial fatigue crack size; ALPHA = shape parameter in Weibull distribution; BETA = scale parameter in Weibull distribution; CMP(1) = 1., CMP(2) = 1., CMP(3) = 1. implies that the aircraft loads due to nonstorm turbulence, thunderstorms, and maneuvers each is taken as composite Gaussian processes; CMP(1) = 0.0, CMP(2) = 0.0, CMP(3) = 0.0 implies that the aircraft loads due to nonstorm turbulence, thunderstorms, and maneuvers each is taken as a single Gaussian process; UPC(1) = 1., UPC(2) = 1., UPC(3) = 1. implies that the input loads are truncated at certain level (single Gaussian case only); EL = the truncation index on maneuver loads; G = gravitational constant in ksi; AA(4), AAA(4) = rise and fall parameters of the random stress process due to gust which can be determined from MANLOD program by selecting the option NSIGN = 0; AAAA(4) = rise and fall parameter of the random stress process due to maneuvers which can be determined from MANLOD program by selecting the option NSIGN = 1; BAB(4) = (2 * I - 1)!! where I = BE/2 and (2 * I - 1)!! = 1 · 3 · 5 . . .; AN(1), AN(2) = characteristic stress response frequencies corresponding to gust; AN(3) = characteristics stress response frequency corresponding to maneuvers; ANA = characteristic stress response frequency corresponding to ground - air - ground loads; PP(1), PP(2), PP(3) = fractions of aircraft flight time spent in clear air turbulence, thunderstorms, and maneuvers, respectively; SGC(1), SGC(2), SGC(3) = stress intensities due to clear air turbulence, thunderstorms, and maneuvers, respectively, when each loading is treated as a composite Gaussian process; SGS(1), SGS(2), SGS(3) = stress intensities due to clear air turbulence, thunderstorms, and maneuvers, respectively, when each loading is treated as a single Gaussian process; XO = 1G level loading (level flight); ZR = stress intensity due to ground-air-ground loads; VO = coefficient of material strength variation; AMVO and RO = mean of the material strength; FSF = 1.0 indicates a fail-safe type design, FSF = 0.0 indicates a slow crack propagation model; ZETA = factor indicating the residual strength at the fail-safe crack size AS; AS = fail-safe crack size; TT = design service life of the airplane; A1 = minimum crack size below which the crack cannot be detected by inspection; A2 = maximum crack size beyond which the crack is always detected by inspection; RM = power exponent for the crack detection probability model; U = inspection probability; NN = number of rigorous base type inspections; NPT = number of divisions on Weibull distribution; NOI = number of output points when no inspections are included in the aircraft probability of failure model; NF = number of aircraft in a fleet. The units of the input data are time = hours, length = inches, strength = ksi; force = pounds.

3.2 PROGRAM DESCRIPTION

The Aircraft Reliability Program AREL1 consists of a main program unit and seven subroutines HCOM, HSIN, QSF, ERF, EEXP, F and FS which are described in the following paragraphs. The program listing is given in Appendix B.

3.2.1 The Main Program Unit AREL1

The main program reads and initializes the specified data as described in Sec. 3.1. The first part of the program is designated for the sensitivity study of the 32 parameters in the aircraft reliability model. This is achieved by specifying the value of XY from 1 to 32. Each number of XY corresponds to a particular parameter in the system which are defined in Sec. 3.1.

The input load parameter, Q, due to the combined effect of the clear air turbulence, thunderstorms, maneuvers, and ground - air - ground loads is calculated from

A = AN(1) * PP(1) * SB(1) + AN(2) * PP(2) * SB(2)
+ AN(3) * PP(3) * SB(3)+ANA * ZBB

where SB(1), SB(2), SB(3) and ZBB are the averages of the BE power of the rise and fall of the stress response process corresponding to clear air turbulence, thunderstorms, maneuvers, and ground - air - ground loads, respectively. These parameters are evaluated for a composite Gaussian process and for a single Gaussian process. The fatigue crack size at flight time T, A(I), (I = 1, 2, . . ., NPT) is determined utilizing the fail-safe design and the slow crack propagation models. The residual strength, R(I), is determined from the Griffith-Irwin equation

R(I) = FLOAT(KC) * SQRT(2.0/PI/A(I))/RO

for the slow crack propagation model and from

R(I) = 1.0 - (1.0 - ZETA) * SQRT((ABS(A(I)
- AO)/(AS - AO)))

for the fail-safe design, where RO = ultimate strength. Then, the failure rate, H(I), is determined by calling subroutine HCOM when the input loading is treated as a composite Gaussian process and subroutine HSIN when the input loading is taken as a single Gaussian process. The summation of the failure rate from the fatigue crack initiation to T flight hours after crack initiation is obtained by integrating the failure rate H(I) by calling subroutine QSF. The time to crack initiation having a two parameter Weibull distribution is calculated from

W(I) = ALPHA/BETA * TBETA ** ALPH1
* EXP(-TBETA ** ALPHA)

where TBETA = T/BETA, ALPH1 = ALPHA - 1. Then, by specifying NN = 0, which implies no inspections, the probability of first aircraft failure, P(I), is calculated. When NN $\neq$ 0 with the inspection probability F specified in FUNCTION F, the probability of first aircraft failure, P(J), (J = TT/NN) is determined at each inspection interval. The probability of first aircraft failure, TP(J), in a fleet of NF airplanes is calculated from

TP(J) = 1.0 - (1.0 - P(J)) ** NF

3.2.2 Subroutine HCOM (H, AR0, AR1, . . ., AR8, P, AN)

The subroutine HCOM determines the failure rate H(I) for the case when input loading is taken as a composite Gaussian process. The necessary arguments of the program are

H = failure rate - output;
AR0, AR1, . . ., AR8 - calculated in the main program;
P = input data corresponding to PP(1), PP(2) and PP(3);
AN = characteristic frequency calculated in the main program.

3.2.3 Subroutine HSIN (H, AR0, AR1, . . ., AR8, P, AN, UPC)

The subroutine HSIN calculates the failure rate, H(I), for the case when input loading is considered as a single Gaussian process. The subroutine provides an option for truncating the failure rate at certain truncation level EL (for example maneuvers). This is achieved by setting UPC = 1.0 in the main program. The necessary arguments of the program are the same as in subroutine HCOM and UPC is specified as input data.

3.2.4 Subroutine QSF (H, Y, Z, NDIM)

Subroutine QSF integrates the given input function Y at specified integration steps H and stores the results in Z at each integration step. NDIM = number of input points to be integrated.

3.2.5 Function EEXP (ARG)

Function EEXP (ARG) is used to calculate EEXP (ARG) = EXP (ARG) - 1.0 by utilizing an expansion procedure for the case when ARG > -50.0 and $0 \leq ARG \leq .01$. When ARG < -50.0, EEXP (ARG) = -1.0, and when ARG > .01 no expansion is used.

3.2.6 Function F(A)

Function F(A) specifies the crack detection probability corresponding to rigorous type inspections in terms of the crack size A. The necessary data for this function is specified in the main program.

3.2.7 Function FS(A)

Function FS(A) determines probabilities of not detecting a fatigue crack.

3.2.8 Function ERF

Function ERF calculates the error function in terms of exponentials.

3.3 PROGRAM OUTPUT

The AREL1 Program generates printed output. First the input data XY, YZ1, YZ2, YZ3 is printed as shown below for a particular example.

```
XY,YZ1,YZ2,YZ3   0.3000000E 02   0.0          0.0          0.0
```

The calculated parameters SB1, SB2, SB3, ZBB, Q, ANS, ATH, TB, AC, TC, CDS, FSF are printed as shown below

```
SB1,SB2,SB3,ZBB,Q,ANS
0.8283E 02( 1.0)  0.3622E 04( 1.0)  0.2400E 03( 1.0)  0.5063E 05

0.6267E 05  0.3300E 03

ATH,TB,AC,TC,CDS,FSF
0.1039E 00  0.4247E 04  0.7054E 00  0.4827E 04  0.1434E-07  0.1000E 01
```

where

SB1, SB2, SB3, ZBB = average stress intensities raised to BE power corresponding to clear air turbulence, thunderstorms, maneuvers, and ground-air-ground loads, respectively.

Q = combined input loading;

ANS = PP(1) * AN(1) + PP(2) * AN(2) + PP(3) * AN(3) is the average characteristic frequency of gust and maneuver loads combined;

ATH = fatigue crack size at the threshold level;

TB = time to reach threshold crack size;

AC = initial critical crack size;

TC = time to reach AC;

CDS = c ;calculated material constant in the crack propagation model

FSF = index denoting fail-safe or slow crack propagation model.

Table 6 contains a listing of the output sample for A, R, H, TH and W as functions of flight time T.

TABLE 6. CRACK SIZE A, RESIDUAL STRENGTH R, FAILURE RATE H, SUMMATION OF FAILURE RATE TH, TIME TO CRACK INITIATION W.

I,TIME,A,R,H,TH,W

I	TIME	A	R	H	TH	W
1	0.0	0.0400	1.0000	0.330E-10	0.0	0.0
51	500.0	0.0412	0.9924	0.411E-10	0.191E-07	0.617E-09
101	1000.0	0.0427	0.9888	0.455E-10	0.407E-07	0.494E-08
151	1500.0	0.0445	0.9856	0.499E-10	0.646E-07	0.167E-07
201	2000.0	0.0467	0.9823	0.547E-10	0.907E-07	0.395E-07
251	2500.0	0.0496	0.9789	0.604E-10	0.119E-06	0.772E-07
301	3000.0	0.0536	0.9748	0.679E-10	0.151E-06	0.133E-06
351	3500.0	0.0603	0.9692	0.796E-10	0.188E-06	0.212E-06
401	4000.0	0.0755	0.9593	0.106E-09	0.233E-06	0.316E-06
451	4500.0	0.2145	0.9097	0.439E-09	0.331E-06	0.450E-06
501	5000.0	1.5828	0.7316	0.781E-07	0.501E-05	0.617E-06
551	5500.0	7.0000	0.4300	0.674E-03	0.187E 00	0.821E-06
601	6000.0	7.0000	0.4300	0.674E-03	0.523E 00	0.106E-05
651	6500.0	7.0000	0.4300	0.674E-03	0.860E 00	0.135E-05
701	7000.0	7.0000	0.4300	0.674E-03	0.120E 01	0.169E-05
751	7500.0	7.0000	0.4300	0.674E-03	0.153E 01	0.208E-05
801	8000.0	7.0000	0.4300	0.674E-03	0.187E 01	0.252E-05
851	8500.0	7.0000	0.4300	0.674E-03	0.221E 01	0.301E-05
901	9000.0	7.0000	0.4300	0.674E-03	0.254E 01	0.357E-05
951	9500.0	7.0000	0.4300	0.674E-03	0.288E 01	0.419E-05
1001	10000.0	7.0000	0.4300	0.674E-03	0.322E 01	0.488E-05
1051	10500.0	7.0000	0.4300	0.674E-03	0.355E 01	0.563E-05
1101	11000.0	7.0000	0.4300	0.674E-03	0.389E 01	0.646E-05
1151	11500.0	7.0000	0.4300	0.674E-03	0.423E 01	0.735E-05
1201	12000.0	7.0000	0.4300	0.674E-03	0.456E 01	0.832E-05
1251	12500.0	7.0000	0.4300	0.674E-03	0.490E 01	0.936E-05
1301	13000.0	7.0000	0.4300	0.674E-03	0.524E 01	0.105E-04
1351	13500.0	7.0000	0.4300	0.674E-03	0.557E 01	0.117E-04
1401	14000.0	7.0000	0.4300	0.674E-03	0.591E 01	0.129E-04
1451	14500.0	7.0000	0.4300	0.674E-03	0.625E 01	0.143E-04
1501	15000.0	7.0000	0.4300	0.674E-03	0.659E 01	0.157E-04

NN= 0

In Table 7 the probabilities of first aircraft failure for a single airplane and in a fleet of 50 airplanes are shown for the case when no inspections are performed.

TABLE 7. PROBABILITY OF FIRST AIRCRAFT FAILURE (NO INSPECTIONS)

I,T,P(I),TP(I)		Single	Fleet
1	500.0	0.16522E-07	0.82608E-06
2	1000.0	0.33043E-07	0.16521E-05
3	1500.0	0.49586E-07	0.24793E-05
4	2000.0	0.66197E-07	0.33098E-05
5	2500.0	0.82990E-07	0.41495E-05
6	3000.0	0.99535E-07	0.49767E-05
7	3500.0	0.11640E-06	0.58200E-05
8	4000.0	0.13458E-06	0.67288E-05
9	4500.0	0.15530E-06	0.77649E-05
10	5000.0	0.17369E-06	0.86845E-05
11	5500.0	0.19418E-06	0.97090E-05
12	6000.0	0.25774E-06	0.12887E-04
13	6500.0	0.73039E-06	0.36519E-04
14	7000.0	0.27108E-05	0.13553E-03
15	7500.0	0.83435E-05	0.41709E-03
16	8000.0	0.21126E-04	0.10558E-02
17	8500.0	0.45866E-04	0.22907E-02
18	9000.0	0.89012E-04	0.44409E-02
19	9500.0	0.15871E-03	0.79041E-02
20	10000.0	0.26461E-03	0.13145E-01
21	10500.0	0.41823E-03	0.20699E-01
22	11000.0	0.63304E-03	0.31166E-01
23	11500.0	0.92381E-03	0.45161E-01
24	12000.0	0.13075E-02	0.63325E-01
25	12500.0	0.18027E-02	0.86270E-01
26	13000.0	0.24298E-02	0.11453E 00
27	13500.0	0.32106E-02	0.14853E 00
28	14000.0	0.41690E-02	0.18851E 00
29	14500.0	0.53304E-02	0.23450E 00
30	15000.0	0.67216E-02	0.28624E 00

The probabilities of first aircraft failure for a single airplane and a fleet of 50 airplanes with 5 scheduled inspections are given in Table 8.

TABLE 8. PROBABILITY OF FIRST AIRCRAFT FAILURE

(INSPECTIONS INCLUDED)

I,T,P(I),TP(I)		Single	Fleet
1	2500.0	0.82990E-07	0.41495E-05
2	5000.0	0.17184E-06	0.85919E-05
3	7500.0	0.72494E-06	0.36246E-04
4	10000.0	0.10433E-04	0.52150E-03
5	12500.0	0.53212E-04	0.26571E-02
6	15000.0	0.16889E-03	0.84099E-02

SECTION 4

AIRCRAFT RELIABILITY WHEN INITIAL CRACK SIZE IS A RANDOM VARIABLE (AREL2)

The purpose of AREL2 program is to determine the probability of first aircraft failure either of a single airplane or of an airplane in a fleet. The probability of aircraft failure is constructed on the condition that the initial fatigue crack size is a random variable for which the probability density function is obtained by appropriate transformation of the Weibull distribution of the time to fatigue crack initiation. The probability of failure model allows to include rigorous base type inspections which are performed at equal time intervals over the design life of the aircraft and cursory inspections which are performed at specified time intervals between the rigorous inspections. The probabilities of aircraft failure are calculated at rigorous inspections. The program includes options to study the effect of proof test and input load level truncation on the probability of first aircraft failure. A simplified flow chart of the program is shown in Figure 3. The following paragraphs discuss the input data, the structure of the program, and the output.

4.1 INPUT DATA

There is one input data card which is of the same format as shown in Table 6 except the parameter XY now ranges from 1 to 34. The remaining data parameters are specified at the beginning of the main program. The main contents of the data parameters for AREL2 program is the same as for AREL1 program. These are described in Sec. 3.1. The additional data parameters are as follow: NBN = number of subdivisions on the probability density function G(A0) corresponding to initial fatigue cracks; AL = lower limit of A0 for G(A0); AU = upper limit of A0 for G(A0); ACT = truncation level due to proof test on G(A0), CSRY = 0.0 implies no cursory inspections, CSRY = 1.0 implies cursory inspections included; NR = number of rigorous inspections; NC = number of cursory inspections; CA1 = minimum crack size which is detected by cursory inspection.

4.2 PROGRAM DESCRIPTION

The Aircraft Reliability Program AREL2 consists of a main program unit and eleven subroutines DENST, HCOM, HSIN, QSF, CIN, HIN, ERF, OMEXP, FF and FC which are described in the following paragraphs. The program listing is given in Appendix C.

4.2.1 The Main Program Unit AREL2

The main program reads and initializes data as described in Sec. 4.1. The first part of the program is designated for the sensitivity study of the 34 parameters in the reliability model. This is accomplished by specifying the value of XY from 1 to 34. Each number corresponds to a particular parameter in

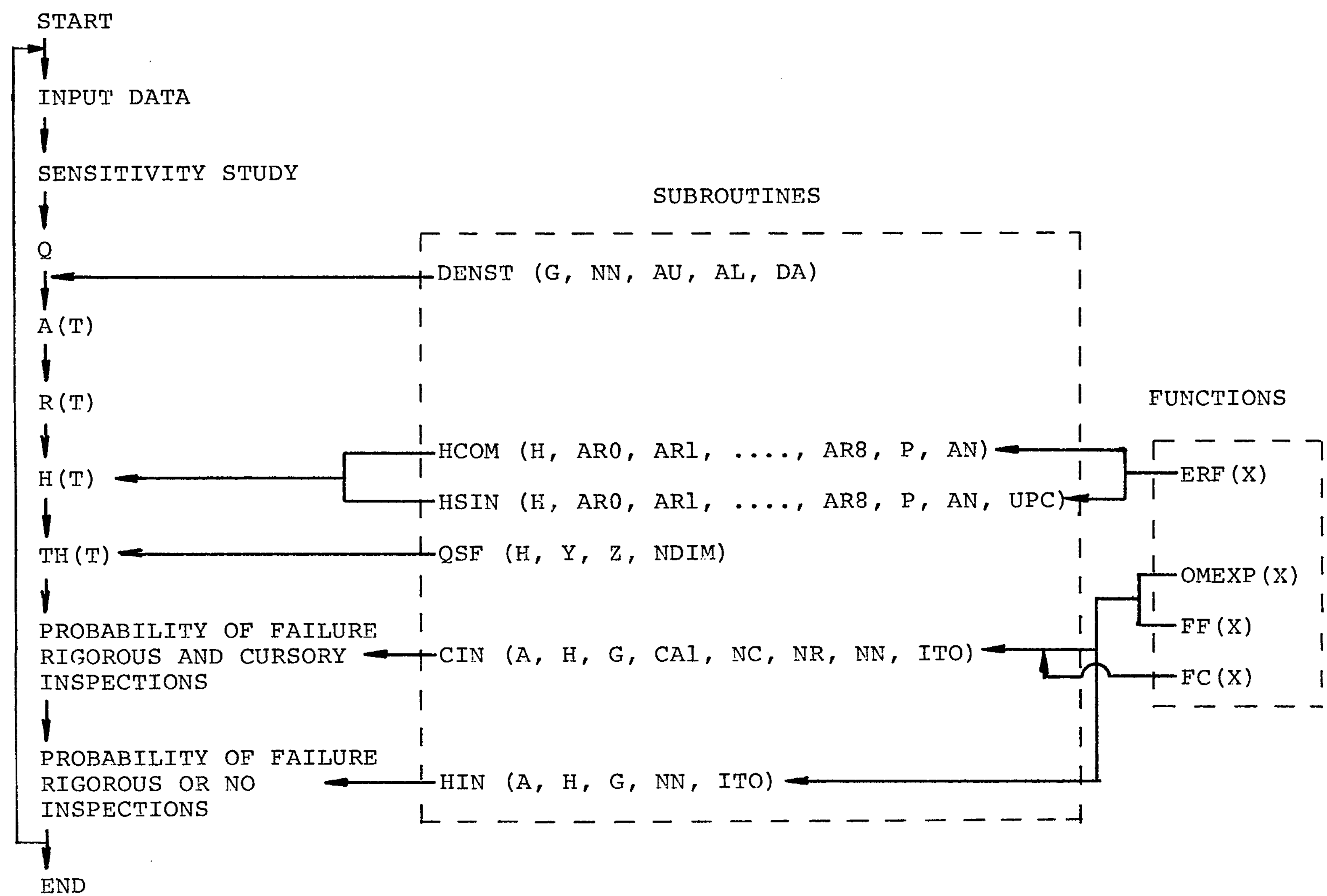


Figure 3. Simplified Flow Chart of the AREL2 Program

the system which are described in Sec. 4.1. The combined input load Q is determined in the same fashion as described in Sec. 3.2.1. Then with a call to subroutine DENST a probability density function G(A0) for the initial crack size A0 is generated. The crack size A(I), the residual strength R(I) and the failure rate H(I) are calculated in terms of A0 in a similar fashion as described in Sec. 3.2.1. The summation of the failure rate H(I) is obtained by calling subroutine QSF. The probability of first aircraft failure with rigorous and cursory inspections included is calculated from subroutine CIN. Similarly, the probability of first aircraft failure where no inspections or only rigorous inspections are performed is determined by calling subroutine HIN. The effect of proof test on the probability of failure is determined by specifying ACT to be equal or less than AU.

4.2.2 Subroutine DENST (G, NN, AU, AL, DA)

Subroutine DENST calculates the probability density function G(A0) of the initial crack size A0. This is achieved by transforming the probability density function W(T) corresponding to the time to crack initiation. The output of the transformation is stored in G. The other arguments of the subroutine are

NN = number of division points on W and number of output points on G;

AU = maximum specified initial crack size value;

AL = lower limit crack size of G.

4.2.3 Subroutine HCOM (H, AR0, AR1, . . ., AR8, P, AN)

Same as described in Sec. 3.2.3.

4.2.4 Subroutine HSIN (H, AR0, AR1, . . ., AR8, P, AN, UPC)

Same as described in Sec. 3.2.4.

4.2.5 Subroutine QSF (H, Y, Z, NDIM)

Same as described in Sec. 3.2.5.

4.2.6 Subroutine CIN (A, H, G, CA1, NC, NR, NN, ITO)

Subroutine CIN determines probabilities of first aircraft failure with both rigorous and cursory inspections included. The arguments of the subroutine are

A = input data - fatigue crack size which is calculated in the main program;

H = summation of the failure rate;

G = probability density function of the initial crack size which is determined by subroutine DENST;

CA1 = minimum crack size which is detected by cursory inspection and is specified as input;

NC = number of cursory inspections in an interval between rigorous inspections;

NR = number of rigorous inspections;

NN = total number of inspections (cursory and rigorous);

ITO = an index for output print.

4.2.7 Subroutine HIN (A, H, G, NN, ITO)

Subroutine HIN calculates probabilities of first aircraft failure when no inspections or only rigorous inspections are performed. The meaning of the arguments for this subroutine are the same as for subroutine CIN.

4.2.8 Function ERF(X)

Function ERF(X) is the same as in AREL1 program.

4.2.9 Function OMEXP(X)

Function OMEXP(X) evaluates OMEXP(X) = 1 - EXP(X) utilizing an expansion procedure.

4.2.10 Function FF(X)

Function FF(X) calculates the crack detection probability corresponding to rigorous type inspections. The necessary data for this function is specified in the main program.

4.2.11 Function FC(X)

Function FC(X) specifies the crack detection probability for cursory inspection. The input argument in terms of the crack detection size is specified in the main program.

4.3 PROGRAM OUTPUT

The AREL2 program generates printed output. The input data XY, YZ1, YZ2, YZ3 is printed for a chosen example. The number of rigorous inspections NN, the crack size A, the failure rate H, the probability of first aircraft failure of a single airplane PA, and the probability of first aircraft failure TPA in a fleet of airplanes are printed as shown in Tables 9 and 10 corresponding to fail-safe and slow crack propagation models, respectively.

TABLE 9. CRACK SIZE A, FAILURE RATE H, PROBABILITY OF FAILURE FOR SINGLE AIRPLANE PA, PROBABILITY OF FAILURE IN A FLEET TPA (FAIL-SAFE)

NN= 14

T	A	H	PA	TPA
11	0.4270466E-01	0.4072061E-07	0.3394086E-07	0.1697041E-05
21	0.4667349E-01	0.9069237E-07	0.6815583E-07	0.3407785E-05
31	0.5364176E-01	0.1513895E-06	0.1024292E-06	0.5121445E-05
41	0.7546765E-01	0.2331089E-06	0.1367163E-06	0.6835792E-05
51	0.1582803E 01	0.5295597E-05	0.1710062E-06	0.8550273E-05
61	0.7000000E 01	0.5207602E 00	0.2053084E-06	0.1026537E-04
71	0.7000000E 01	0.1194295E 01	0.2396573E-06	0.1198279E-04
81	0.7000000E 01	0.1867827E 01	0.2740477E-06	0.1370229E-04
91	0.7000000E 01	0.2541360E 01	0.3084521E-06	0.1542248E-04
101	0.7000000E 01	0.3214892E 01	0.3428797E-06	0.1714383E-04
111	0.7000000E 01	0.3888425E 01	0.3773134E-06	0.1886550E-04
121	0.7000000E 01	0.4561957E 01	0.4117510E-06	0.2058732E-04
131	0.7000000E 01	0.5235490E 01	0.4461873E-06	0.2230910E-04
141	0.7000000E 01	0.5909022E 01	0.4806233E-06	0.2403087E-04
151	0.7000000E 01	0.6582555E 01	0.5150594E-06	0.2575263E-04

TABLE 10. CRACK SIZE A, FAILURE RATE H, PROBABILITY OF FAILURE FOR SINGLE AIRPLANE PA, PROBABILITY OF FAILURE IN A FLEET TPA (SLOW CRACK GROWTH)

NN= 14

T	A	H	PA	TPA
11	0.4270466E-01	0.3304082E-07	0.3304045E-07	0.1652020E-05
21	0.4667349E-01	0.6608151E-07	0.6607974E-07	0.3303981E-05
31	0.5364176E-01	0.9912242E-07	0.9911918E-07	0.4955946E-05
41	0.7546765E-01	0.1321632E-06	0.1321584E-06	0.6607897E-05
51	0.1582803E 01	0.1765910E-04	0.1651974E-06	0.8259838E-05
61	0.1000000E 04	0.7670562E 03	0.1984070E-06	0.9920299E-05
71	0.1000000E 04	0.1767055E 04	0.2322512E-06	0.1161249E-04
81	0.1000000E 04	0.2767054E 04	0.2671156E-06	0.1335569E-04
91	0.1000000E 04	0.3767052E 04	0.3021181E-06	0.1510579E-04
101	0.1000000E 04	0.4767039E 04	0.3372525E-06	0.1686247E-04
111	0.1000000E 04	0.5767020E 04	0.3725574E-06	0.1862769E-04
121	0.1000000E 04	0.6767000E 04	0.4078362E-06	0.2039160E-04
131	0.1000000E 04	0.7766980E 04	0.4431927E-06	0.2215938E-04
141	0.1000000E 04	0.8766961E 04	0.4785293E-06	0.2392617E-04
151	0.1000000E 04	0.9766941E 04	0.5138896E-06	0.2569414E-04

APPENDIX A

```
C
C     MANUVER LOADING
C
      DIMENSION A(32768),INV(4096),S(4096),BETA(6)
      DIMENSION A1(8),A2(8),A3(8),A35(8),A4(8),A45(8),A5(8)
      DIMENSION B1(8),B2(8),B3(8),B35(8),B4(8),B45(8),B5(8)
      DIMENSION X1(6),X2(6),X3(6),X35(6),X4(6),X45(6),X5(6),XX(6,5)
      DIMENSION Y1(6),Y2(6),Y3(6),Y35(6),Y4(6),Y45(6),Y5(6),YY(6,5)
      DIMENSION NPK(8),MPK(8),NPEAK(6),MPEAK(6)
      DIMENSION AA(8,5),BB(8,5),AN(5),BN(5)
C     COMMON /AXC/Z(200)
      DATA BETA/0.0,0.25,0.5,0.75,0.8,0.9/
C     NSAMPL   --  NO OF SAMPLES USED FOR EACH BETA VALUE
C     NBETA    --  NO OF BETA VALUES
C     WUP      --  UPPER BOUND FREQUENCY IN HERTZ
C     WL       --  LOWER BOUND FREQUENCY IN HERTZ
C     N        --  NO OF DIVISIONS IN SPECTRAL DENSITY
C     NPT      --  NO OF SIMULATED TIME POINTS
C     M1       --  2**M1=NPT
C     NSIGN    --  0  USE WHITE NOISE SPECTRUM
C                  1  USE USER SUPPLY SPECTRUM
C     TIME     --  TOTAL SIMULATED PERIOD IN SECONDS
C     REDUCE  -- REDUCTION FACTOR OF THE NEGATIVE PART OF THE
C                TIME HISTORY
      REDUCE=0.25
      PI=3.141593
      NSAMPL=8
      NBETA=1
      WUP=0.2
      N=2048
      NPT=16384
      M1=14
      NSIGN=1
      DT=FLOAT(N)/(WUP*FLOAT(NPT))
      WL=0.007
      TIME=DT*NPT
C     CALL AREA TO FIND THE AREA OF SPECTRAL DENSITY IN ORDER TO
C     NORMALIZE S(W)
      CALL AREA (S,512,WUP,WL)
      A(1)=100000.
      A(2)=0.
      A(3)=.2
      SCALE=1.0
C     CALL GRAPH (A,S,512,2,' ',2,'  ',2,' ',2,'  ',SCALE)
      WRITE(6,312) TIME,DT,WUP,WL
312   FORMAT(10X,'PERIOD=',F10.2,4X,'DT=',F6.4,4X,'WU=',F5.3,
     1 4X,'WL=',F5.3/)
      NP=400
C     NPOTS  --  NO OF POINTS IN EACH GRAPH
C     NPLOTS --  NO OF GRAPHS TO BE PLOTTED
      NPOTS=1025
      IX=3234567
      DO 100 I=1,NBETA
      WU=WUP
```

```
      DO 80 J=1,NSAMPL
      CALL SIMULT (A,N,NPT,WU,WL,IX,M1,NSIGN,INV,S)
      DO 72 KK=1,NPT
      IF(A(KK).LT.0.0) A(KK)=A(KK)*REDUCE
   72 CONTINUE
      WRITE(6,20)
 20   FORMAT(1H1)
      IF (J.GT.1) GO TO 15
C     WRITE OUT PART OF THE SAMPLE TIME HISTORY (1ST SAMPLE)
      NS=NPT/8
      WRITE(6,74) (A(II),II=1,NS)
   74 FORMAT((1X,24F5.2))
      CALL PLOT (NP,DT,A,0.0,1)
   15 CALL PICK (A,NPT,S,NPK(J))
      CALL STDEVI(1,NPK(J),S,STD,RMS,A1(J),A2(J),A3(J),A35(J)
     1 ,A4(J),A45(J),A5(J),AN)
      DO 34 L=1,5
   34 AA(J,L)=AN(L)
      WRITE(6,25) BETA(I),NPK(J),WU,WL,STD,RMS
   25 FORMAT(//5X,'BETA=',F6.3,5X,'NPEAK=',I6,5X,'WU='
     1 F8.4,5X,'WL=',F8.4,5X,'STDEV=',E14.7,5X,'RMS=',E14.7/)
      WRITE(6,30) A1(J),A2(J),A3(J),A35(J),A4(J),A45(J),A5(J),
     1 (AA(J,L),L=1,5)
   30 FORMAT(5X,'MEAN1=',F7.3,3X,'MEAN2=',F7.3,3X,'MEAN3=',F7.3,
     1 3X,'MEAN35=',F7.3,3X,'MEAN4=',F7.3,3X,'MEAN45=',F8.3,
     2 3X,'MEAN5=',F8.3/5X,'MEAN6=',E12.5,2X,'MEAN7=',E12.5,2X,
     3 'MEAN8=',E12.5,2X,'MEAN9=',E12.5,2X,'MEAN10=',E12.5/)
C     WRITE(6,70) (A(L),L=1,NP)
C  70 FORMAT(/10(2X,F10.5))
C     WRITE(6,70) (S(L),L=1,50)
      CALL PICKM (A,NPT,S,MPK(J))
      CALL STDEVI(1,MPK(J),S,STD,RMS,B1(J),B2(J),B3(J),B35(J),
     1 B4(J),B45(J),B5(J),BN)
      DO 44 L=1,5
   44 BB(J,L)=BN(L)
      WRITE(6,25) BETA(I),MPK(J),WU,WL,STD,RMS
      WRITE(6,30) B1(J),B2(J),B3(J),B35(J),B4(J),B45(J),B5(J)
     1 ,(BB(J,L),L=1,5)
C     WRITE(6,70) (S(L),L=1,50)
      CALL STATIS (A,NPT,STDV,RMSV)
C     TO FIND THE STANDARD DEVIATION OF THE PROCESS
      WRITE(6,212) STDV,RMSV
  212 FORMAT(//5X,'STANDARD DEVIATION OF THE TIME HISTORY'
     1 //5X,'STDEV=',E14.7,5X,'R.M.S.=',E14.7)
   80 CONTINUE
      NSAM=NSAMPL
      CALL MEAN7 (NSAM,A1,A2,A3,A35,A4,A45,A5,AM1,AM2,AM3,
     1 AM35,AM4,AM45,AM5,AA,AN)
      WRITE(6,95) AM1,AM2,AM3,AM35,AM4,AM45,AM5,(AN(L),L=1,5)
   95 FORMAT(////10X,'AM  ',6(3X,E14.7)/15X,6(3X,E14.7)/)
      X1(I)=AM1
      X2(I)=AM2
      X3(I)=AM3
      X35(I)=AM35
      X4(I)=AM4
      X45(I)=AM45
```

```
      X5(I)=AM5
      DO 82 L=1,5
   82 XX(I,L)=AN(L)
      CALL MEAN7 (NSAM,B1,B2,B3,B35,B4,B45,B5,BM1,BM2,BM3,
     1 BM35,BM4,BM45,BM5,BB,BN)
      WRITE(6,96) BM1,BM2,BM3,BM35,BM4,BM45,BM5,(BN(L),L=1,5)
   96 FORMAT(//10X,'BM   ',6(3X,E14.7)/15X,6(3X,E14.7)/)
      Y1(I)=BM1
      Y2(I)=BM2
      Y3(I)=BM3
      Y35(I)=BM35
      Y4(I)=BM4
      Y45(I)=BM45
      Y5(I)=BM5
      DO 81 L=1,5
   81 YY(I,L)=BN(L)
      NPEAK(I)=0
      MPEAK(I)=0
      DO 98 II=1,NSAM
      NPEAK(I)=NPEAK(I)+NPK(II)
   98 MPEAK(I)=MPEAK(I)+MPK(II)
      NPEAK(I)=NPEAK(I)/FLOAT(NSAM)
      MPEAK(I)=MPEAK(I)/FLOAT(NSAM)
  100 CONTINUE
      WRITE(6,105)
  105 FORMAT(1H1////42X,'CASE I'//)
      WRITE(6,106)
  106 FORMAT(30X,'AVERAGE OF POWER OF RISE AND FALL'/
     1 /30X,' AND NUMBER OF RISE AND FALL '//43X,'POWER'/)
      WRITE(6,110)
  110 FORMAT(15X,'1.0',5X,'2.0',5X,'3.0',5X,'3.5',5X,'4.0',
     1 5X,'4.5',5X,'5.0',5X,'RISE & FALL'/6X,'BETA')
      DO 120 I=1,NBETA
  120 WRITE(6,140) BETA(I),X1(I),X2(I),X3(I),X35(I),X4(I),
     1 X45(I),X5(I),NPEAK(I)
  140 FORMAT(/6X,F4.2,2X,7F8.3,2X,I7)
  141 FORMAT(/6X,F4.2,2X,5(F10.3,1X),2X,I6)
      WRITE(6,145)
  145 FORMAT(1H1////42X,'CASE II'//)
      WRITE(6,106)
      WRITE(6,110)
      DO 150 I=1, NBETA
  150 WRITE(6,140) BETA(I),Y1(I),Y2(I),Y3(I),Y35(I),Y4(I)
     1 ,Y45(I),Y5(I),MPEAK(I)
      WRITE(6,105)
      WRITE(6,106)
      WRITE(6,111)
  111 FORMAT(15X,'6.0',8X,'7.0',8X,'8.0',8X,'9.0',8X,'10.',
     1 8X,'PEAKS'/6X,'BETA')
      DO 121 I=1,NBETA
  121 WRITE(6,141) BETA(I),(XX(I,L),L=1,5),NPEAK(I)
      WRITE(6,145)
      WRITE(6,106)
      WRITE(6,111)
      DO 151 I=1,NBETA
  151 WRITE(6,141) BETA(I),(YY(I,L),L=1,5),MPEAK(I)
```

```
      STDEV=1.
      CALL PICKMX (A,NPT,S(1),LLL,S(2049),MMM)
      WRITE(6,123) LLL,(S(L),L=1,LLL)
      WRITE(6,124) MMM,(S(L+2048),L=1,MMM)
  123 FORMAT(1H1/10X,'NO. OF PEAKS=',I6/(2X,10F8.4))
  124 FORMAT(1H1/10X,'NO. OF TROUGHS=',I6/(2X,10F8.4))
C     RATIO  --  1  GIVE TOTAL CROSSINGS OF THE HISTORY
C                3600/PERIOD GIVE CROSSINGS PER HOUR
      RATIO=1.
      CALL CROSS (STDEV,S(1),S(2049),LLL,MMM,VZERO,RATIO)
      RATIO=3600./TIME
      CALL CROSS (STDEV,S(1),S(2049),LLL,MMM,VZERO,RATIO)
      STOP
      END


      SUBROUTINE STDEVI(NST,NPT,FUN,STD,RMS,A1,A2,A3,A35,A4,A45,A5,AN)
      DIMENSION FUN(1),AN(1)
      ASUM1=0.0
      ASUM2=0.0
      ASUM3=0.0
      ASUM35=0.0
      ASUM4=0.0
      ASUM45=0.0
      ASUM5=0.0
      DO 10 I=1,5
   10 AN(I)=0.0
      XPOINT=FLOAT(NPT-NST+1)
      DO 5 JT=NST,NPT
      AA2=FUN(JT)*FUN(JT)
      AA3=FUN(JT)*AA2
      AA35=FUN(JT)**3.5
      AA4=AA2*AA2
      AA45=FUN(JT)**4.5
      AA5=AA3*AA2
      ASUM1=ASUM1+FUN(JT)
      ASUM2=ASUM2+AA2
      ASUM3=ASUM3+AA3
      ASUM35=ASUM35+AA35
      ASUM4=ASUM4+AA4
      ASUM45=ASUM45+AA45
      ASUM5=ASUM5+AA5
      AN(1)=AN(1)+AA3*AA3
      AN(2)=AN(2)+AA35*AA35
      AN(3)=AN(3)+AA4*AA4
      AN(4)=AN(4)+AA45*AA45
      AN(5)=AN(5)+AA5*AA5
    5 CONTINUE
      A1=ASUM1/XPOINT
      A2=ASUM2/XPOINT
      A3=ASUM3/XPOINT
      A35=ASUM35/XPOINT
      A4=ASUM4/XPOINT
      A45=ASUM45/XPOINT
      A5=ASUM5/XPOINT
```

```
      DO 15 K=1,5
   15 AN(K)=AN(K)/XPOINT
      RMS=SQRT(A2)
      STD=SQRT(A2-A1*A1)
      RETURN
      END
```

```
      SUBROUTINE MEAN7 (NSAMPL,A1,A2,A3,A35,A4,A45,A5,AM1,AM2,
     1 AM3,AM35,AM4,AM45,AM5,AA,AN)
      DIMENSION A1(1),A2(1),A3(1),A35(1),A4(1),A45(1),A5(1)
      DIMENSION AA(8,5),AN(5)
      AM1=0.
      AM2=0.
      AM3=0.
      AM35=0.
      AM4=0.
      AM45=0.
      AM5=0.
      DO 10 I=1,5
   10 AN(I)=0.
      DO 90 K=1,NSAMPL
      AM1=AM1+A1(K)
      AM2=AM2+A2(K)
      AM3=AM3+A3(K)
      AM35=AM35+A35(K)
      AM4=AM4+A4(K)
      AM45=AM45+A45(K)
      AM5=AM5+A5(K)
      DO 80 L=1,5
   80 AN(L)=AN(L)+AA(K,L)
   90 CONTINUE
      SAMPL=FLOAT(NSAMPL)
      AM1=AM1/SAMPL
      AM2=AM2/SAMPL
      AM3=AM3/SAMPL
      AM35=AM35/SAMPL
      AM4=AM4/SAMPL
      AM45=AM45/SAMPL
      AM5=AM5/SAMPL
      DO 85 L=1,5
   85 AN(L)=AN(L)/SAMPL
      RETURN
      END
```

```
      SUBROUTINE PICK (A,NP,PEAK,NPK)
C
C     PICK UP DIFFERNECE BETWEEN SUCESSIVE RISE AND FALL
C     A    -- INPUT ARRAY FROM WHICH RISE AND FALL ARE TO BE FOUND
C     NP   -- NO OF POINTS IN INPUT ARRAY A
C     PEAK -- ARRAY TO STORE THE RESULT OF (PEAK(I)-TROUGH(I-1))
C     NPK  -- NUMBER OF VALUE FOUND
C
      DIMENSION A(1),PEAK(1)
      PEAK(1)=A(1)
      IFLAG=1
      IF(A(1).GT.A(2)) IFLAG=0
      IPK=2
      DO 100 I=1,NP
      IF(IFLAG.LT.1) GO TO 60
C     PEAK TESTING
      IF(A(I).LT.A(I+1)) GO TO 100
      PEAK(IPK)=A(I)
      PEAK(IPK-1)=ABS(PEAK(IPK)-PEAK(IPK-1))
      IPK=IPK+1
      IFLAG=0
      GO TO 100
C     TROUGH TESTING
   60 IF(A(I).GT.A(I+1)) GO TO 100
      PEAK(IPK)=A(I)
      PEAK(IPK-1)=ABS(PEAK(IPK)-PEAK(IPK-1))
      IPK=IPK+1
      IFLAG=1
  100 CONTINUE
      NPK=IPK-2
      RETURN
      END


      SUBROUTINE PICKM (A,NP,PEAK,MPK)
C
C     PICK UP THE DIFFERENCES BETWEEN SUCESSIVE MAXIMUM RISE AND
C     MINIMUN FALL ( 2ND METHOD OF PICKING)
C
C     A    -- INPUT ARRAY FROM WHICH RISE AND FALL ARE TO BE PICKED
C     NP   -- NUMBER OF POINTS OF INPUT ARRAY A
C     PEAK -- ARRAY USE TO HOLD THE RISE AND FALL DIFFERENCES
C     MPK  -- NUMBER OF RISE AND FALL DIFFERNENCE FOUND
C
      DIMENSION A(1),PEAK(1)
      AMAX=0.
      AMIN=0.
      IFLAG=1
      IF(A(1).LT.0.) IFLAG=0
      IPK=1
```

```
      DO 100 I=1,NP
      IF(IFLAG.EQ.0) GO TO 40
      IF(A(I).GT.AMAX) AMAX=A(I)
      IF(A(I+1).LT.0.) GO TO 50
      GO TO 100
   50 IFLAG=0
      PEAK(IPK)=ABS(AMAX-AMIN)
      IPK=IPK+1
      AMIN=1.0
      GO TO 100
   40 IF(A(I).LT.AMIN) AMIN=A(I)
      IF(A(I+1).GT.0.) GO TO 70
      GO TO 100
   70 IFLAG=1
      PEAK(IPK)=ABS(AMAX-AMIN)
      IPK=IPK+1
      AMAX=-1.0
  100 CONTINUE
      MPK=IPK-1
      RETURN
      END

      SUBROUTINE PICKC (NPT,Y,PMAX,PMIN,LPEAK,MTROG)
C
C     TO PICK UP PEAKS AND TROUGHS FROM A PROCESS
C     Y  -- INPUT PROCESS FROM WHICH PEAKS AND TROUGHS ARE TO BE
C           FOUND
C     NPT - NO OF INPUT POINTS OF Y
C     PMAX  -- ARRAY USE TO HOLD SUCESSIVE PEAKS VALUE
C     PMIN  -- ARRAY USE TO HOLD SUCESSIVE TROUGH VALUES
C     LPEAK -- NUMBER OF PEAKS FOUND
C     MTROG -- NUMBER CF TROUGHS FCUND
C
      DIMENSION Y(1),PMAX(1),PMIN(1)
      I=1
      L=1
      M=1
  100 IF((I+1).GE.NPT) GO TO 30
   25 IF(Y(I).GT.Y(I+1)) GO TO 13
   20 IF(Y(I).GT.Y(I+1)) GO TO 15
   35 IF(I.GE.NPT) GO TO 100
   14 I=I+1
      GO TO 20
   15 PMAX(L)=Y(I)
      L=L+1
      GO TO 100
   13 IF(Y(I).LE.Y(I+1)) GO TO 16
   33 IF(I.GE.NPT) GO TO 100
   18 I=I+1
      GO TO 13
   16 PMIN(M)=Y(I)
      M=M+1
      GO TO 100
   30 LPEAK=L-1
      MTROG=M-1
      RETURN
      END
```

```
      SUBROUTINE AREA (S,NPT,FU,FL)
      DIMENSION S(1)
      COMMON ANORML
C
C     ANORML -- AREA OF THE SPECTRUM
C
C     FU  -- UPPER CUTOFF FREQUENCY IN HERTZ
C     FL  -- LOWER   "         "      "   "
C
      ANORML=1.0
      PI2=2.0*3.141593
      WUP=PI2*FU
      WLR=PI2*FL
      DW=WUP/FLOAT(NPT)
      NUMP=INT(WLR/DW)
      IF(NUMP.EQ.0) NUMP=1
      DO 45 I=1,NPT
   45 S(I)=0.
      DO 48 I=NUMP,NPT
      WI=FLOAT(I)*DW
      CALL FUNSPT (SP,WI)
   48 S(I)=SP
C     USE SIMPSN RULE TO COMPUTE THE AREA UNDER THE CURVE
      NPP=NPT
      IF(NPP/2*2.EQ.NPP) NPP=NPP-1
      CALL SIMPSN (S,NPP,DW,SAREA)
      ANORML=SAREA
      WRITE(6,60) SAREA,FU,FL,WUP,WLR,(S(I),I=1,128)
   60 FORMAT(10X,'AREA=',E14.7,5X,4(4X,F6.4)/8(2X,F8.4))
      RETURN
      END


      SUBROUTINE SIMPSN (AF,NPT,DSTEP,SIMP)
      DIMENSION AF(1)
C
C     NPT  -- NO. OF POINTS FOR INTEGRATION (MUST BE ODD NO.)
C     SIMP -- RESULTING AREA OF INTEGRATION
C
      NP=(NPT-1)/2
      MN=NP-1
      ODD=0.
      EVEN=0.
      END=AF(1)+AF(NPT)
      DO 10 I=1,NP
   10 ODD=ODD+AF(2*I)
      DO 20 I=1,MN
   20 EVEN=EVEN+AF(2*I+1)
      SIMP=(4.0*ODD+2.0*EVEN+END)*DSTEP/3.0
      RETURN
      END
```

```
      SUBROUTINE CROSS (STDEV,PMAX,PMIN,LLL,MMM,VZERO,RATIO)
      DIMENSION COUNT(51),PMAX(1),PMIN(1)
      INTEGER COUNT,SIGN
C
C     TO FIND CROSSINGS OF A PROCESS (INCLUDES +VE AND -VE SLOPS)
C     WE START CROSSINGS FROM THE FIRST TROUGH
C
C     STDEV  -- STANDARD DEVIATION OF THE PROCESS
C     PMAX   -- PEAK VALUE ARRAY OF THE PROCESS
C     PMIN   -- TROUGH VALUE ARRAY OF THE PROCESS
C     LLL    -- NO OF PEAKS
C     MMM    -- NO OF TROUGHS
C     VZERO  -- ZERO CROSSINGS
C     RATIO  -- SET TO 1 IF CROSSINGS FOR THE WHOLE PERIOD OF THE
C                         PROCESS IS TO BE FOUND
C               SET TO 3600/PERIOD IF CROSSINGS/HOUR IS TO BE
C                         FOUND
      KL=1
      STDEV5=STDEV/5.0
      DO 4 I=1,51
    4 COUNT(I)=0
  100 XXX=PMIN(KL)/STDEV5
      YYY=PMAX(KL)/STDEV5
      II=XXX
      II=II+26
      IF(PMIN(KL).LE.0.) GO TO 11
   15 II=II+1
   11 KK=YYY
      KK=KK+26
      IF(PMAX(KL).GT.0) GO TO 14
   12 KK=KK-1
   14 IF(KK.LE.51) GO TO 28
   33 KK=51
   28 IF(II) 31,31,22
   31 II=1
   22 IF(II.GT.51) GO TO 38
   20 COUNT(II)=COUNT(II)+1
   38 II=II+1
      IF(II.LE.KK) GO TO 20
   25 XXX=PMIN(KL+1)/STDEV5
      YYY=PMAX(KL)/STDEV5
      IJ=XXX
      IJ=IJ+26
      IF(PMIN(KL+1).LE.0.0) GO TO 21
   45 IJ=IJ+1
   21 KJ=YYY
      KJ=KJ+26
      IF(PMAX(KL).GT.0.0) GO TO 75
   74 KJ=KJ-1
   75 IF(KJ-51) 65,65,70
   70 KJ=51
   65 IF(IJ) 60,60,90
   60 IJ=1
   90 IF(IJ.GT.51) GO TO 82
```

```
 80 COUNT(IJ)=COUNT(IJ)+1
 82 IJ=IJ+1
    IF(IJ.LE.KJ) GO TO 80
 72 KL=KL+1
    IF(KL-MMM) 100,400,400
400 WRITE(6,500)
500 FORMAT(1H1,5X,'NUMBER OF TOTAL CROSSINGS AT DIFFERENT LEVEL OF'
   1,' SIGMA,' /6X,'INCLUDES CROSSINGS WITH BOTH +VE AND -VE SLOP'//
   2 6X,'SIMULATED RESULT'//)
177 Z1=-5.0
    DO 66 I=1,51
    COUNT(I)=IFIX(FLOAT(COUNT(I))*RATIO)
    WRITE(6,600) Z1,COUNT(I)
    Z1=Z1+.2
 66 CONTINUE
    VZERO=COUNT(26)
600 FORMAT(5X,F4.1,'SIGMA=',I6,2X,'CROSSINGS')
    RETURN
    END
```

```
C          SUBROUTINE SIMULT
C
C          PURPOSE
C             PERFORMS SIMULATION OF RANDOM SAMPLE FUNCTION BY THE USE
C             OF FAST FOURIER TRANSFORM METHOD
C
C          USAGE
C             CALL SIMULT (A,N,NPT,NPT2,NPT4,WU,WL,IX,M1,NSIGN,S,INV)
C
C          DESCRIPTION OF PARAMETERS
C             A       --A IS THE OUTPUT SAMPLE FUNCTION IN TIME DOMAIN
C                       IT IS A ONE-DIMENSION ARRAY
C             N       --N IS THE NUMBER OF SUBDIVISIONS OF A GIVEN SPECTRUM
C                       SS(W),AND N MUST BE A NUMBER OF POWER OF 2
C                       THAT IS  N=2**M ,M IS A POSITIVE INTEGER
C             NPT     --NPT IS THE NUMBER OF POINTS OF THE OUTPUT SAMPLE
C                       FUNCTION SPECIFIED AS INPUT NPT=2**M1 WITH
C                       2*N EQUAL TO OR LESS THAN NPT
C             NPT2    --NPT2 IS AN INTEGER INPUT EQUAL TO 2*NPT
C             NPT4    --NPT4 IS EQUAL TO ONE-FOURTH OF NPT AND MUST
C                       BE SPECIFIED AS INPUT
C             FU      --FU IS THE UPPER BOUND FREQUENCY IN HERTZ OF THE
C                       GIVEN SPECTRUM SS(W) ,MUST BE SPECIFIED
C             FL      --FL IS THE LOWER BOUND FREQUENCY IN HERTZ OF THE
C                       GIVEN SPECTRUM SS(W) ,MUST BE SPECIFIED
C             IX      --AN ODD INTEGER INPUT LESS THAN 9 DIGITS,BY CHANGING
C                       IX,WE WILL GET A DIFFERENT SAMPLE FUNCTION FOR A
C                       GIVEN N
C             M1      --AN POSITIVE INTEGER INPUT SUCH THAT NPT=2**M1
```

```
C           NSIGN --AN OPTION PARAMETER
C                         0   USE WHITE NOISE SPECTRUM PROVIDE BY THE
C                             SUBROUTINE SIMULT WITH FREQUENCY DOMAIN WU TO
C                             WL TO SIMULATE THE SAMPLE FUNCTION,AND THE
C                             SAMPLE FUNCTION IS NORMALIZED WITH MEAN=0 AND
C                             STANDARD DEVIATION=1
C                         1   USER WILL HAVE TO PROVIDE SPECTRUM SS(W) FOR
C                             SUBROUTINE FUNSPT AND THE SAMPLE FUNCTION
C                             SIMULATED IS NOT NORMALIZED BUT THE MEAN IS
C                             STILL ZERO
C
C        REMARKS
C           THIS SUBROUTINE WILL GIVE A UP TO 2**M1 POINTS
C           WITH M1 MUST NOT BE LESS THAN 3 OR GREATER THAN 20
C
C        SUBROUTINES AND FUNCTIONS REQUIRED
C           HARM RANDU,FUNSPT
C
C        METHOD
C                N
C        F(T)= SUM SQRT(2.0*SS(W(K))*EXP( I*PHI(K))*EXP(I*W(K)*T)
C               K=1
C
C             I=IMAGINARY NUMBER    T=TIME      PHI =RANDOM PHASE ANGLE
C           AND
C                    A(T)=SQRT((WU-WL)/N)*REAL(F(T))
C
C           FOR N SUBDIVISIONS OF SPECTRUM, FFT ONLY GIVE N POINTS OF
C           TIME HISTORY, THAT IS CNE COMPLETE PERIOD,THEREFORE THE
C           RESOLUTION OF THE PROCESS IS BAD. TO IMPROVE THE RESOLUTION
C           WE HAVE TO ADD SS(W(K))=0 K=N+2,...,NPT AFTER SS(WU)
C           THE TIME INCREMENT DTIME IS
C                  DTIME=1.0/W(NPT)       W=FREQUENCY
C           NOTE:  SS(W) HAS PRESCRIBED VALUE ONLY UP TO N POINTS
C                  IF PHI    ARE FIXED UP TO K=N, THE SAMPLE FUNCTION
C                  WILL BE THE SAME, THE ONLY THING CHANGES IS DTIME
C                  THEREFORE WHEN WE INCREASE NPT WE DECREASE DTIME.
C                  A,S,AND INV    HAVE TO BE DIMENSIONED IN TME MAIN
C                  PROGRAM ACCORDINGLY.
C
C     ********************************************************************
C
      SUBROUTINE SIMULT(A,N,NPT,FU,FL,IX,M1,NSIGN,INV,S)
      DIMENSION A(1),M(3),INV(1),S(1)
      PI=3.141593
      NL=256
C
C     CONVERT HERTZS INTO RADIANS
C
      WU=FU*2.0*PI
      WL=FL*2.0*PI
      DW=WU/FLOAT(N)
      NUMP=INT(WL/DW)
      WL=DW*NUMP
```

```
      IF(NUMP.EQ.0) NUMP=1
      SQROOT=SQRT(2.0*WU/((WU-WL)*FLOAT(N)))
      M(1)=M1
      M(2)=0
      M(3)=0
      IFSET=1
      DO 5 I=1,NUMP
      A(I*2-1)=0.0
      A(I*2)=0.0
    5 CONTINUE
      NN=N+2
      DO 10 I=NN,NPT
      A(2*I-1)=0.0
      A(2*I   )=0.0
   10 CONTINUE
      IF(NSIGN.EQ.1) GO TO 40
C
C     TO GENERATE RANDOM PHASE ANGLES PHI
C
      DO 30 I=NUMP,N
      CALL RANDU (IX,IY,YFL)
      IX=IY
      PHI=2.0*PI*YFL
      A(2*I+1)=COS(PHI)
      A(2*I+2)=SIN(PHI)
   30 CONTINUE
      GO TO 60
   40 DO 50 J=NUMP,N
      W1=FLOAT(J)*DW
      CALL FUNSPT (SP,W1)
      CALL RANDU (IX,IY,YFL)
      IX=IY
      PHI=2.0*PI*YFL
      A(2*J+1)=SQRT(2.0*SP)*COS(PHI)
      A(2*J+2)=SQRT(2.0*SP)*SIN(PHI)
   50 CONTINUE
C
C     START SIMULATING SAMPLE FUNCTION FOR A GIVEN SET OF PHI
C
   60 CALL HARM (A,M,INV,S,IFSET,IFERR)
      IF(NSIGN.EQ.1) GO TO 80
C
C     SAMPLE FUNCTION FROM SS(W) EQUAL TO WHITE NOISE
C
      DO 70 I=1,NPT
      A(I)=A(2*I-1)*SQROOT
   70 CONTINUE
      GO TO 100
C
C     SAMPLE FUNCTION FROM A SPECIFIED SS(W)
C
   80 DO 90 I=1,NPT
      A(I)=A(2*I-1)*SQRT(DW)
   90 CONTINUE
  100 RETURN
      END
```

```
      SUBROUTINE FUNSPT (SP,W1)
C
C     SS(W)=... IS A FUNCTION STATEMENT NONEXECUTABLE AND MUST BE
C     SUPPLIED BY THE USER.   SS(W) IS THE SPECTRAL DENSITY FUNCTION
C
      COMMON ANORML
      SS(W)=PI2*PHI0/(1.0+(PI2*W/FB)**A)/ANORML
      PI2=1./(2.*3.141593)
      PHI0=30
      FB=0.031
      A=2.6
      SP=SS(W1)
      RETURN
      END
```

```
      SUBROUTINE PLOT (NPT,H,FUN,FACT,NSTEP)
C
C            NPT----NO. OF POINTS TO BE PLOTTED
C            FUN----FUNCTION TO BE PLOTTED
C            H------TIME INTERVAL (TO 2 DECIMAL PLACES)
C  REMARK: TITLE AND DIMENSION OF FUNCTION HAVE TO BE PRINTED OUT BEFORE
C          CALLING PLOT
C  REMARK: IF FACT=0.,THEN SCALE FACTOR IS COMPUTED BASED ON THE GIVEN
C          FUNCTION,OTHERWISE SCALE FACTOR=FACT
C
      DIMENSION ALINE(120),FUN(NPT),DAS(120),SCALE(11)
      DATA DASH/'-'/,STAR/'*'/,BAR/'I'/,BLANK/' '/
C
C     DETERMINE THE SCALE FACTOR
C
      IF(FACT.EQ.0.) GO TO 30
      FACTOR=FACT
      GO TO 80
   30 AMAX=1.0E-70
      DO 40 I=1,NPT,NSTEP
      IF(ABS(FUN(I)).GT.AMAX) AMAX=ABS(FUN(I))
   40 CONTINUE
      FACTOR=AMAX/50.0
C
C     START TO PLOT
C
   80 DO 95 I=1,11
   95 SCALE(I)=-60.+10.*FLOAT(I)
      DO 100 I=1,120
  100 DAS(I)=DASH
      DO 105 I=15,115,10
  105 DAS(I)=BAR
      DO 110 I=10,120
  110 ALINE(I)=BLANK
      ALINE(65)=BAR
      WRITE(6,200) FACTOR
      WRITE(6,302) SCALE
      WRITE(6,303) (DAS(I),I=10,120)
```

```
      DO 120 I=1,NPT,NSTEP
      TIME=H*FLOAT(I-1)
      NP=FUN(I)/FACTOR+65.
      ALINE(NP)=STAR
      WRITE(6,304)    TIME,(ALINE(J),J=10,120)
      ALINE(NP)=BLANK
      IF(NP.EQ.65) ALINE(65)=BAR
  120 CONTINUE
  200 FORMAT(///,50X,'SCALE FACTOR=', E11.3)
  302 FORMAT(1H0,2X,'TIME',5X,5(F4.0,6X),1X,6(F3.0,7X))
  303 FORMAT (1H ,8X,114A1)
  304 FORMAT(1H ,F8.4,111A1)
      RETURN
      END




      SUBROUTINE PICKMX(A,NP,PEAK,MPK,TROUG,NPK)
C
C     PICK UP THE DIFFERENCES BETWEEN SUCESSIVE MAXIMUM RISE AND
C     MINIMUN FALL ( 2ND METHOD OF PICKING)
C
C     A    -- INPUT ARRAY FROM WHICH RISE AND FALL ARE TO BE PICKED
C     NP   -- NUMBER OF POINTS OF INPUT ARRAY A
C     PEAK -- ARRAY USE TO HOLD THE RISE AND FALL DIFFERENCES
C     MPK  -- NUMBER OF RISE AND FALL DIFFERNENCE FOUND
C
      DIMENSION A(1),PEAK(1),TROUG(1)
      AMAX=0.
      AMIN=0.
      IFLAG=1
      IF(A(1).LT.0.) IFLAG=0
      IPK=1
      ITK=1
      NST=1
      IF(IFLAG.EQ.0) GO TO 20
      DO 10 I=1,NP
      IF(A(I).LT.0.0) GO TO 15
   10 CONTINUE
   15 NST=I
      IFLAG=0
   20 DO 100 I=NST,NP
      IF(IFLAG.EQ.0) GO TO 40
      IF(A(I).GT.AMAX) AMAX=A(I)
      IF(A(I+1).LT.0.) GO TO 50
      GO TO 100
   50 IFLAG=0
      PEAK(IPK)=AMAX
      IPK=IPK+1
      AMIN=1.0
      GO TO 100
```

```
 40 IF(A(I).LT.AMIN) AMIN=A(I)
    IF(A(I+1).GT.0.) GO TO 70
    GO TO 100
 70 IFLAG=1
    TROUG(ITK)=AMIN
    ITK=ITK+1
    AMAX=-1.0
100 CONTINUE
    MPK=IPK-1
    NPK=ITK-1
    RETURN
    END
```

```
      SUBROUTINE STATIS (A,NP,STD,RMS)
      DIMENSION A(1)
C
C     TO FIND THE STANDARD DEVIATION AND RMS VALUES OF A PROCESS
C
      AS=0.
      AS2=0.
      DO 10 I=1,NP
      AS=AS+A(I)
      AS2=AS2+A(I)*A(I)
   10 CONTINUE
      AS=AS/FLOAT(NP)
      AS2=AS2/FLOAT(NP)
      STD=SQRT(AS2-AS*AS)
      RMS=SQRT(AS2)
      RETURN
      END
```

```
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*65539
      IF(IY) 5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*0.4656613D-9
      RETURN
      END
```

```
      SUBROUTINE HARM(A,M,INV,S,IFSET, IFERR,NPT2,NPT4)
      DIMENSION A(NPT2),INV(NPT4),S(NPT4 ),N(3),M(3),NP(3),W(2),W2(2),
     1W3(2)
      EQUIVALENCE (N1,N(1)),(N2,N(2)),(N3,N(3))
   10 IF( IABS(IFSET) - 1) 900,900,12
   12 MTT=MAX0(M(1),M(2),M(3)) -2
      ROOT2 = SQRT(2.)
      IF (MTT-MT ) 14,14,13
   13 IFERR=1
      RETURN
   14 IFERR=0
      M1=M(1)
      M2=M(2)
      M3=M(3)
      N1=2**M1
      N2=2**M2
      N3=2**M3
   16 IF(IFSET) 18,18,20
   18 NX= N1*N2*N3
      FN = NX
      DO 19 I = 1,NX
      A(2*I-1) = A(2*I-1)/FN
   19 A(2*I) = -A(2*I)/FN
   20 NP(1)=N1*2
      NP(2)= NP(1)*N2
      NP(3)=NP(2)*N3
      DO 250 ID=1,3
      IL = NP(3)-NP(ID)
      IL1 = IL+1
      MI = M(ID)
      IF (MI)250,250,30
   30 IDIF=NP(ID)
      KBIT=NP(ID)
      MEV = 2*(MI/2)
      IF (MI - MEV )60,60,40
   40 KBIT=KBIT/2
      KL=KBIT-2
      DO 50 I=1,IL1,IDIF
      KLAST=KL+I
      DO 50 K=I,KLAST,2
      KD=K+KBIT
      T=A(KD)
      A(KD)=A(K)-T
      A(K)=A(K)+T
      T=A(KD+1)
      A(KD+1)=A(K+1)-T
   50 A(K+1)=A(K+1)+T
      IF (MI - 1)250,250,52
   52 LFIRST =3
      JLAST=1
      GO TO 70
   60 LFIRST = 2
      JLAST=0
```

```
 70 DO 240 L=LFIRST,MI,2
    JJDIF=KBIT
    KBIT=KBIT/4
    KL=KBIT-2
    DO 80 I=1,IL1,IDIF
    KLAST=I+KL
    DO 80 K=I,KLAST,2
    K1=K+KBIT
    K2=K1+KBIT
    K3=K2+KBIT
    T=A(K2)
    A(K2)=A(K)-T
    A(K)=A(K)+T
    T=A(K2+1)
    A(K2+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    T=A(K3)
    A(K3)=A(K1)-T
    A(K1)=A(K1)+T
    T=A(K3+1)
    A(K3+1)=A(K1+1)-T
    A(K1+1)=A(K1+1)+T
    T=A(K1)
    A(K1)=A(K)-T
    A(K)=A(K)+T
    T=A(K1+1)
    A(K1+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    R=-A(K3+1)
    T = A(K3)
    A(K3)=A(K2)-R
    A(K2)=A(K2)+R
    A(K3+1)=A(K2+1)-T
 80 A(K2+1)=A(K2+1)+T
    IF (JLAST) 235,235,82
 82 JJ=JJDIF   +1
    ILAST= IL +JJ
    DO 85 I = JJ,ILAST,IDIF
    KLAST = KL+I
    DO 85 K=I,KLAST,2
    K1 = K+KBIT
    K2 = K1+KBIT
    K3 = K2+KBIT
    R =-A(K2+1)
    T = A(K2)
    A(K2) = A(K)-R
    A(K) = A(K)+R
    A(K2+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    AWR=A(K1)-A(K1+1)
    AWI = A(K1+1)+A(K1)
    R=-A(K3)-A(K3+1)
    T=A(K3)-A(K3+1)
    A(K3)=(AWR-R)/ROOT2
    A(K3+1)=(AWI-T)/ROOT2
    A(K1)=(AWR+R)/ROOT2
    A(K1+1)=(AWI+T)/ROOT2
    T= A(K1)
```

```
      A(K1)=A(K)-T
      A(K)=A(K)+T
      T=A(K1+1)
      A(K1+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
      R=-A(K3+1)
      T=A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
   85 A(K2+1)=A(K2+1)+T
      IF(JLAST-1) 235,235,90
   90 JJ= JJ + JJDIF
      DO 230 J=2,JLAST
   96 I=INV(J+1)
   98 IC=NT-I
      W(1)=S(IC)
      W(2)=S(I)
      I2=2*I
      I2C=NT-I2
      IF(I2C)120,110,100
  100 W2(1)=S(I2C)
      W2(2)=S(I2)
      GO TO 130
  110 W2(1)=0.
      W2(2)=1.
      GO TO 130
  120 I2CC = I2C+NT
      I2C=-I2C
      W2(1)=-S(I2C)
      W2(2)=S(I2CC)
  130 I3=I+I2
      I3C=NT-I3
      IF(I3C)160,150,140
  140 W3(1)=S(I3C)
      W3(2)=S(I3)
      GO TO 200
  150 W3(1)=0.
      W3(2)=1.
      GO TO 200
  160 I3CC=I3C+NT
      IF(I3CC)190,180,170
  170 I3C=-I3C
      W3(1)=-S(I3C)
      W3(2)=S(I3CC)
      GO TO 200
  180 W3(1)=-1.
      W3(2)=0.
      GO TO 200
  190 I3CCC=NT+I3CC
      I3CC = -I3CC
      W3(1)=-S(I3CCC)
      W3(2)=-S(I3CC)
  200 ILAST=IL+JJ
```

```
      DO 220 I=JJ,ILAST,IDIF
      KLAST=KL+I
      DO 220 K=I,KLAST,2
      K1=K+KBIT
      K2=K1+KBIT
      K3=K2+KBIT
      R=A(K2)*W2(1)-A(K2+1)*W2(2)
      T=A(K2)*W2(2)+A(K2+1)*W2(1)
      A(K2)=A(K)-R
      A(K)=A(K)+R
      A(K2+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
      R=A(K3)*W3(1)-A(K3+1)*W3(2)
      T=A(K3)*W3(2)+A(K3+1)*W3(1)
      AWR=A(K1)*W(1)-A(K1+1)*W(2)
      AWI=A(K1)*W(2)+A(K1+1)*W(1)
      A(K3)=AWR-R
      A(K3+1)=AWI-T
      A(K1)=AWR+R
      A(K1+1)=AWI+T
      T=A(K1)
      A(K1)=A(K)-T
      A(K)=A(K)+T
      T=A(K1+1)
      A(K1+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
      R=-A(K3+1)
      T=A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
  220 A(K2+1)=A(K2+1)+T
  230 JJ=JJDIF+JJ
  235 JLAST=4*JLAST+3
  240 CONTINUE
  250 CONTINUE
      NTSQ=NT*NT
      M3MT=M3-MT
  350 IF(M3MT) 370,360,360
  360 IGO3=1
      N3VNT=N3/NT
      MINN3=NT
      GO TO 380
  370 IGO3=2
      N3VNT=1
      NTVN3=NT/N3
      MINN3=N3
  380 JJD3 = NTSQ/N3
      M2MT=M2-MT
  450 IF (M2MT)470,460,460
```

```
460 IGO2=1
    N2VNT=N2/NT
    MINN2=NT
    GO TO 480
470 IGO2 = 2
    N2VNT=1
    NTVN2=NT/N2
    MINN2=N2
480 JJD2=NTSQ/N2
    M1MT=M1-MT
550 IF(M1MT)570,560,560
560 IGO1=1
    N1VNT=N1/NT
    MINN1=NT
    GO TO 580
570 IGO1=2
    N1VNT=1
    NTVN1=NT/N1
    MINN1=N1
580 JJD1=NTSQ/N1
600 JJ3=1
    J=1
    DO 880 JPP3=1,N3VNT
    IPP3=INV(JJ3)
    DO 870 JP3=1,MINN3
    GO TO (610,620),IGO3
610 IP3=INV(JP3)*N3VNT
    GO TO 630
620 IP3=INV(JP3)/NTVN3
630 I3=(IPP3+IP3)*N2
700 JJ2=1
    DO 870 JPP2=1,N2VNT
    IPP2=INV(JJ2)+I3
    DO 860 JP2=1,MINN2
    GO TO (710,720),IGO2
710 IP2=INV(JP2)*N2VNT
    GO TO 730
720 IP2=INV(JP2)/NTVN2
730 I2=(IPP2+IP2)*N1
800 JJ1=1
    DO 860 JPP1=1,N1VNT
    IPP1=INV(JJ1)+I2
    DO 850 JP1=1,MINN1
    GO TO (810,820),IGO1
810 IP1=INV(JP1)*N1VNT
    GO TO 830
820 IP1=INV(JP1)/NTVN1
830 I=2*(IPP1+IP1)+1
    IF (J-I) 840,850,850
840 T=A(I)
    A(I)=A(J)
    A(J)=T
    T=A(I+1)
    A(I+1)=A(J+1)
    A(J+1)=T
850 J=J+2
860 JJ1=JJ1+JJD1
870 JJ2=JJ2+JJD2
880 JJ3 = JJ3+JJD3
890 IF(IFSET)891,895,895
```

```
891 DO 892 I = 1,NX
892 A(2*I) = -A(2*I)
895 RETURN
900 MT=MAXO(M(1),M(2),M(3)) -2
    MT = MAXO(2,MT)
904 IF (MT-18) 906,906,13
906 IFERR=0
    NT=2**MT
    NTV2=NT/2
910 THETA=.7853981634
    JSTEP=NT
    JDIF=NTV2
    S(JDIF)=SIN(THETA)
    DO 950 L=2,MT
    THETA=THETA/2.
    JSTEP2=JSTEP
    JSTEP=JDIF
    JDIF=JSTEP/2
    S(JDIF)=SIN(THETA)
    JC1=NT-JDIF
    S(JC1)=COS(THETA)
    JLAST=NT-JSTEP2
    IF(JLAST - JSTEP) 950,920,920
920 DO 940 J=JSTEP,JLAST,JSTEP
    JC=NT-J
    JD=J+JDIF
940 S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)
950 CONTINUE
960 MTLEXP=NTV2
    LM1EXP=1
    INV(1)=0
    DO 980 L=1,MT
    INV(LM1EXP+1) = MTLEXP
    DO 970 J=2,LM1EXP
    JJ=J+LM1EXP
970 INV(JJ)=INV(J)+MTLEXP
    MTLEXP=MTLEXP/2
980 LM1EXP=LM1EXP*2
982 IF(IFSET)12,895,12
    END
```

APPENDIX B

```
// EXEC FORTGCG
//FORT.SYSIN DD *,DCB=BLKSIZE=3440
C AREL1
C SENSITIVITY STUDY IS AVAILABLE
C DELTA K THRESHOLD IS AVAILABLE
C THREE LOADS (COMPOSIT OR SINGLE GAUSSIAN) ARE AVAILABLE
      DIMENSION AA(10),AAA(10),BAB(10)
      DIMENSION A(1501),R(1501),H(1501),          TH(1501),W(1501)
      DIMENSION PP(3),SGC(3),RR(3),ANETA(3)
      DIMENSION DUMY(1501),TP(50),P(50),PS(50),RK(50),CIJ(50),QIJ(50)
      DIMENSION CMP(3),SGS(3),VX(3),AN(3),UPC(3),SB(3)
      DIMENSION AAAA(10)
   81 CONTINUE
C*********************************************************
C******************** MEAN VALUES ************************
C******************** FRACTURE DATA **********************
      EF=2.5
      BE=4.0
      DKTH=1.5
      EFD=10.0
      C=3.0E-7
      KC=60.0
C******************** INITIAL CRACK DATA *****************
      A0=0.04
      ALPHA=4.0
      BETA=30000.0
C******************** LOAD DATA **************************
C CMP (1.0=COMPOSIT GAUSSIAN, 0.0=SINGLE ONE)
      CMP(1)=1.0
      CMP(2)=1.0
      CMP(3)=1.0
C UPC (1.0=UPPER CUT SINGLE GAUSSIAN, 0.0=COMPLETE ONE)
      UPC(1)=0.0
      UPC(2)=0.0
      UPC(3)=0.0
C EL=UPPER CUT LEVEL (EL*X0)
      EL=8.0
      G=10.0
      AA(4)=115.0
      AAA(4)=115.0
      AAAA(4)=80.0
      BAB(4)=3.0
C AN(I) NUMBER OF LOAD CYCLES PER HOUR
      AN(1)=600.0
      AN(2)=600.0
      AN(3)=60.0
      ANA=0.5
      PP(1)=0.4975
      PP(2)=0.0025
      PP(3)=0.5
C SGC STANDARD DEVIATION OF COMPOSITE GAUSSIAN
      SGC(1)=0.07*G
      SGC(2)=0.18*G
      SGC(3)=0.1*G
C SGS STANDARD DEVIATIONS (SINGLE GAUSSIAN)
      SGS(1)=1.5*G
      SGS(2)=1.5*G
      SGS(3)=1.5*G
      X0=1.0*G
      ZR=1.5
```

```
C************** MATERIAL DATA ************************
      V0=0.056
      AMJ0=5.7*G
      R0=5.7*G
C************** FAIL SAFE DATA ***********************
      FSF=1.0
      ZETA=0.43
      AS=7.0
C*****************************************************
C TT   TOTAL FLIGHT HOUR
      TT=15000.0
C************** INSPECTION DATA **********************
      A1=0.02
      A2=0.3
      RM=0.125
      U=1.0
C NN   NUMBER OF INSPECTIONS
C TT/(NN+1) MUST BE INTEGER
C (TT/(NN+1))/(TT/(NPT-1)) MUST BE INTEGER
      NN=0
C*****************************************************
      PI=3.14159
C NPT   BASIC DATA POINTS
C EACH DIMENSION SIZE OF(A,R,H,TH,DUMY,W)MUST BE G.E. NPT
C (TT/(NPT-1)) MUST BE INTEGER
      NPT=1501
C NOI   DATA POINTS UNDER NO INSPECTION CASE
C EACH DIMENSION SIZE OF(TP P,PS,PK,CIJ,QIJ)MUST BE G.E. NOI AND (NN+1)
C (TT/NOI) MUST BE INTEGER
C (TT/NOI)/(TT/(NPT-1)) MUST BE INTEGER
      NOI=30
C************** FLEET SIZE ***************************
      NF=50
C*****************************************************
C*****************************************************
C SUB PROGRAM FOR SENSITIVITY STUDY
      READ (5,501) XY,YZ1,YZ2,YZ3
  501 FORMAT(4F10.0)
      WRITE(6,650) XY,YZ1,YZ2,YZ3
  650 FORMAT(1H ,///,' XY,YZ1,YZ2,YZ3',4E15.7)
      IF(XY.LE.0.0) GO TO 322
      IF(XY.EQ.1.0) EF=YZ1
      IF(XY.EQ.2.0)DKTH=YZ1
      IF(XY.EQ.3.0) EFD=YZ1
      IF(XY.EQ.4.0) C=YZ1
      IF(XY.EQ.5.0) AD=YZ1
      IF(XY.EQ.6.0) ANO=YZ1
      IF(XY.EQ.7.0) ANA=YZ1
      IF(XY.EQ.8.0) ANM=YZ1
      IF(XY.EQ.9.0) V0=YZ1
      IF(XY.EQ.10.0) ZETA=YZ1
      IF(XY.EQ.11.0) AS=YZ1
      IF(XY.EQ.12.0) KC=IFIX(YZ1)
      IF(XY.NE.13.0) GO TO 721
      PP(1)=YZ1
      PP(2)=YZ2
      PP(3)=YZ3
  721 CONTINUE
      IF(XY.NE.14.0) GO TO 722
      SGC(1)=YZ1
```

```
      SGC(2)=YZ2
      SGC(3)=YZ3
  722 CONTINUE
      IF(XY.NE.15.0) GO TO 723
      G=YZ1
      SGC(1)=0.07*G
      SGC(2)=0.18*G
      SGC(3)=0.1*G
      XO=1.0*G
      RO=5.7*G
      AMUO=5.7*G
  723 CONTINUE
      IF(XY.NE.16.0) GO TO 724
      AMUO=YZ1
      RO=YZ1
  724 CONTINUE
      IF(XY.EQ.17.0) ALPHA=YZ1
      IF(XY.EQ.18.0) BETA=YZ1
      IF(XY.EQ.19.0) FSF=YZ1
      IF(XY.EQ.20.0) BE=YZ1
      IF(XY.EQ.21.0) AA(4)=YZ1
      IF(XY.EQ.21.0) AAA(4)=YZ1
      IF(XY.EQ.22.0) AAAA(4)=YZ1
      IF(XY.EQ.23.0) ZR=YZ1
      IF(XY.NE.24.0) GO TO 725
      SGS(1)=YZ1
      SGS(2)=YZ2
      SGS(3)=YZ3
  725 CONTINUE
      IF(XY.NE.25.0) GO TO 726
      CMP(1)=YZ1
      CMP(2)=YZ2
      CMP(3)=YZ3
  726 CONTINUE
      IF(XY.EQ.26.0) A1=YZ1
      IF(XY.EQ.27.0) A2=YZ1
      IF(XY.EQ.28.0) RM=YZ1
      IF(XY.EQ.29.0) U=YZ1
      IF(XY.EQ.30.0) VV=YZ1
      IF(XY.NE.31.0) GO TO 727
      UPC(1)=YZ1
      UPC(2)=YZ2
      UPC(3)=YZ3
  727 CONTINUE
      IF(XY.EQ.32.0) EL=YZ1
C SUB PROGRAM FOR SENSITIVITY STUDY *** END
C*******************************************************************
      JKL=0
      KBE=BE
      IBE=BE
      IF(CMP(1).EQ.1.0)    SB(1)=AA(IBE)*BAB(IBE)*SGC(1)**KBE
      IF(CMP(1).EQ.0.0)    SB(1)=AA(IBE)         *SGS(1)**KBE
      IF(CMP(2).EQ.1.0)   SB(2)=AAA(IBE)*BAB(IBE)*SGC(2)**KBE
      IF(CMP(2).EQ.0.0)   SB(2)=AAA(IBE)         *SGS(2)**KBE
      IF(CMP(3).EQ.1.0)SB(3)=AAAA(IBE)*BAB(IBE)*SGC(3)**KBE
      IF(CMP(3).EQ.0.0)SB(3)=AAAA(IBE)         *SGS(3)**KBE
      ZBB=(ZR*G)**KBE
      Q=AN(1)*PP(1)*SB(1)+AN(2)*PP(2)*SB(2)+AN(3)*PP(3)*SB(3)
     1+ANA*ZBB
      AC=2./PI* (KC/RO)**2
```

```
      DT=TT/FLOAT(NPT-1)
C
      ANS=PP(1)*AN(1)+PP(2)*AN(2)+PP(3)*AN(3)
C ANS=CYCLES PER HOUR
C $$$ THRESHOLD NO IRETA A(I) NO KEISAN $$$
      IF(EF.EQ.2.0) EF=2.01
      IF(EFD.EQ.2.0) EFD=2.01
C
      ANTH=ALOG10(C)+EF*ALOG10(DKTH)
C
      CDST=ANTH-EFD*ALOG10(DKTH)
      CDS=10.0**CDST
C
      ATH=(DKTH/(Q/ANS)**(1.0/BE))**2*(2.0/PI)
C
      TB=2.0/(2.0-EFD)/CDS/Q**(EFD/BE)*ANS**((EFD-BE)/BE)
      TB=TB/(PI/2.0)**(EFD/2.0)
      TB=TB*(ATH**((2.0-EFD)/2.0)-AO**((2.0-EFD)/2.0))
C
      DO 8 I=1,NPT
      T=FLOAT(I-1)*DT
      IF(TB.GT.0.0) GO TO 43
C IN CASE OF ATH.LT.AO
      A(I)=AO**((2.0-EF)/2.0)
      A(I)=A(I)+((2.0-EF)/2.0)*C*Q**(EF/BE)*ANS**((BE-EF)/BE)*T
     1 *(PI/2.0)**(EF/2.0)
      IF(A(I).LT.0.00001) GO TO 45
      A(I)=A(I)**(2.0/(2.0-EF))
      GO TO 48
   43 CONTINUE
C IN CASE OF ATH.GT.AO
      IF(T.GT.TB) GO TO 49
      A(I)=AO**((2.0-EFD)/2.0)
      A(I)=A(I)+((2.0-EFD)/2.0)*CDS*Q**(EFD/BE)*ANS**((BE-EFD)/BE)*T
     1 *(PI/2.0)**(EFD/2.0)
      IF(A(I).LT.0.00001) GO TO 45
      A(I)=A(I)**(2.0/(2.0-EFD))
C
      GO TO 48
C
   49 A(I)=ATH**((2.0-EF)/2.0)
      A(I)=A(I)+((2.0-EF)/2.0)*C*Q**(EF/BE)*ANS**((BE-EF)/BE)*(T-TB)
     1 *(PI/2.0)**(EF/2.0)
      IF(A(I).LT.0.00001) GO TO 45
      A(I)=A(I)**(2.0/(2.0-EF))
      GO TO 48
   45 A(I)=10000.0
C
   48 IF(ESE.EQ.0.0) AS=1000.0
      IF(ABS(A(I)).GT.AS) GO TO 15
    8 CONTINUE
      GO TO 46
   15 KK=I
      DO 60 I=KK,NPT
   60 A(I)=AS
   46 CONTINUE
C ### A(I) CAL. END ###
      IF(ATH.GT.AC) GO TO 353
      TC=2.0/(2.0-EF )*(AC**((2.0-EF )/2.0)-ATH**((2.0-EF )/2.0))
      TC=TC/C/Q**(EF /BE)/ANS**((BE-EF )/BE)/(PI/2.0)**(EF /2.0)
```

```
      TC=TC+TB
      GO TO 364
  363 TC=2.0/(2.0-EFD)*(AC**((2.0-EFD)/2.0)- AO**((2.0-EFD)/2.0))
      TC=TC/CDS/Q**(EFD/BE)/ANS**((BE-EFD)/BE)/(PI/2.0)**(EFD/2.0)
  364 WRITE(6,800) SB(1),CMP(1),SB(2),CMP(2),SB(3),CMP(3),ZBB,Q,ANS
  800 FORMAT(1H ,/,'    SB1,SB2,SB3,ZBB,Q,ANS ',/,
     13(E12.4,'(',F4.1,')'), 3E12.4)
      WRITE(6,801) ATH,TB,AC,TC,CDS,FSF
  801 FORMAT(1H ,/,'  ATH,TB,AC,TC,CDS,FSF',/,5E12.4)
C
C RESIDUAL STRENGTH RATE ( FSF=1.0(FAIL SAFE)  =0.0(NO FAIL SAFE)  )
      DO 18 I=1,NPT
      IF(FSF.EQ.1.0) GO TO 189
      IF(A(I).LE.AC) GO TO 188
      R(I)=FLOAT(KC)*SQRT(2.0/PI/A(I))/R0
      GO TO 18
  188 R(I)=1.0
      GO TO 18
  189 SW=ABS((A(I)-AO)/(AS-AO))
      R(I)=1.0-(1.0-ZETA)*SQRT(SW)
   18 CONTINUE
C
C  $$$ CAL. OF H $$$
      DO 115 N=1,NPT
      HWA=0.0
      GMN=R(N)
      DO 116 I=1,3
      IF(CMP(I).EQ.0.0) SGC(I)=SGS(I)
      VX(I)=SGC(I)/XO
      ANETA(I)=GMN*AMUO/SGC(I)-XO/SGC(I)
      RR(I)=VO*GMN*AMUO/SGC(I)
      ARGO=ANETA(I)/SQRT(2.0)/RR(I)
      ARG1=-ANETA(I)+RR(I)**2/2.0
      ARG2=(ANETA(I)-RR(I)**2)/SQRT(2.0)/RR(I)
      ARG3=ANETA(I)/RR(I)/SQRT(2.0)
      ARG4=1.0/SQRT(1.0+RR(I)**2)
      ARG5=-0.5*(1.0/VX(I)**2+1.0/VO**2)
      ARG6=0.5*(RR(I)/VX(I)+1.0/VO)**2/(1.0+RR(I)**2)
      ARG7=ANETA(I)/SQRT(2.0)/RR(I)/SQRT(1.0+RR(I)**2)
      ARG8=1.0/SQRT(2.0)/SQRT(RR(I)**2+1.0)
     1*((FL-1.0)*RR(I)/VX(I)+FL/RR(I)/VX(I)-1.0/VO)
      IF(CMP(I).EQ.1.0) CALL HCOM(H(N),ARGO,ARG1,ARG2,ARG3,ARG4,
     1ARG5,ARG6,ARG7,ARG8,PP(I),AN(I) )
      IF(CMP(I).EQ.0.0) CALL HSIN(H(N),ARGO,ARG1,ARG2,ARG3,ARG4,
     1ARG5,ARG6,ARG7,ARG8,PP(I),AN(I),UPC(I) )
      HWA=HWA+H(N)
  116 CONTINUE
      H(N)=HWA
      IF(H(N).GT.1.0) H(N)=1.0
  115 CONTINUE
C ### CAL. OF H END ###
      CALL QSF(DT,H,TH,NPT)
      ALPH1=ALPHA-1
      DO 208 I=1,NPT
      T=FLOAT(I-1)*DT
      TBETA=T/BETA
  208 W(I)=ALPHA/BETA*TBETA**ALPH1*EXP(-TBETA**ALPHA)
      WRITE(6,1010)
 1010 FORMAT(1H ,/,'   I,TIME,A,R,H,TH,W',/)
C***IF NEED MORE ACCURATE PJT'DO 80 I=1,NPT,BIGGER'***
```

```
      DO 80  I=1,NPT,50
      T=FLOAT(I-1)*DT
      WRITE(6,1020) I,T,A(I),R(I),H(I),TH(I),W(I)
 1020 FORMAT(1H ,I4,F8.1,2F11.4,3F13.3)
   80 CONTINUE
      WRITE(6,600) NN
  600 FORMAT(1H ,'NN=',I5)
      DO 77  I=1,NPT
      DUMY(I)=H(I)
   77 CONTINUE
      H0=DUMY(1)
C
      IF(NN.NE.0) GO TO 100
C**************************NO INSPECTION CASE ***********************
      NN=NOI-1
      TI=TT/FLOAT(NN+1)
      NDT=TI/DT
      NNN=NN+1
      DO 10  I=1,NNN
      TT0=TI*FLOAT(I)
      IT=NDT*I+1
      DO 20  K=1,IT
      T=FLOAT(K-1)*DT
      NH=IT+1-K
      X=-T*H0-TH(NH)
      EX=EEXP(X)
      DUMY(K)=W(K)+W(K)*EX
   20 CONTINUE
      CALL QSF(DT,DUMY,DUMY,IT)
      X=-TT0*H0-(TT0/BETA)**ALPHA
      XX=EEXP(X)
      P(I)=-XX-DUMY(IT)
      IF(P(I).LT.1.0E-20) P(I)=1.0E-20
      WRITE(6,688)I,IT,TT0,X,XX,DUMY(IT),P(I)
  688 FORMAT(1H ,I3,I6,F10.1,4E15.7)
   10 CONTINUE
      GO TO 3000
C*************************** INSPECTION CASE ***********************
  100 CONTINUE
      NNN=NN+1
      TI=TT/FLOAT(NN+1)
      NDT=TI/DT
      IT=NDT+1
      DO 2000  J=1,NNN
      TT0=TI*FLOAT(J)
C     **PS(J) **
      DO 200 K=1,IT
      T=DT*FLOAT(K-1)
      NH=IT+1-K
      NW=(J-1)*NDT+K
      X=-(T+FLOAT(J-1)*TI)*H0-TH(NH)
      EX=EEXP(X)
      DUMY(K)=W(NW)+W(NW)*EX
  200 CONTINUE
      CALL QSF(DT,DUMY,DUMY,IT)
      X=-(FLOAT(J-1)*TI/BETA)**ALPHA
      Y=-(FLOAT(J)*TI/BETA)**ALPHA-FLOAT(J)*TI*H0
      EX=EEXP(X)
      EY=EEXP(Y)
      PS(J)=EX-EY-DUMY(IT)
```

```
      IF(J.NE.1) GO TO 210
      P(J)=PS(J)
      GO TO 220
  210 CONTINUE
C ********* QIJ *********
      WRITE(6,1009)
 1009 FORMAT(1H ,3X,'QIJ(I)=')
      JM=J-1
      DO 1000 I=1,JM
      DO 250 K=1,IT
      T=DT*FLOAT(K-1)
      NW=(I-1)*NDT+K
      ZZ=-HO*(FLOAT(I-1)*TI+T)
C *** TERM 1 ***
      NA=IT+1-K
      NH=NA
      NK=J-I
      X=ZZ-TH(NH)-RK(NK)
      EX=EEXP(X)
      AX=A(NA)
      TERM1=-F(AX,A1,A2,RM,U)*EX
C *** TERM 2 ***
      FSM=1.0
      JII=J-I
      DO 350 KK=1,JII
      NA=KK*NDT+2-K
      AX=A(NA)
  350 FSM=FSM*(1.0-F(AX,A1,A2,RM,U))
      NH=(J-I+1)*NDT+2-K
      X=ZZ-TH(NH)
      EX=EEXP(X)
      TERM2=-FSM*EX
C *** TERM 3 ***
      TERM3=0.0
      II=J-I
      IF(II.LT.2) GO TO 480
      DO 400 KK=2,II
      FSM=1.0
      KKM=KK-1
      DO 460 M=1,KKM
      NA=M*NDT+2-K
      AX=A(NA)
      X=F(AX,A1,A2,RM,U)
      FS=1.0-F(AX,A1,A2,RM,U)
  460 FSM=FSM*FS
      NA=KK*NDT+2-K
      AX=A(NA)
      NH=NA
      NK=J-I-KK+1
      X=ZZ-TH(NH)-RK(NK)
      EX=EEXP(X)
  400 TERM3=TERM3-FSM*F(AX,A1,A2,RM,U)*EX
  480 DUMY(K)=(TERM1+TERM2+TERM3)*W(NW)
  250 CONTINUE
      CALL QSF(DT,DUMY,DUMY,IT)
      QIJ(I)=DUMY(IT)
 1000 CONTINUE
C
      TQIJ=0.0
      DO 490 I=1,JM
```

```
  490 TOIJ=TOIJ+OIJ(I)
      P(J)=PS(J)+TOIJ
      IF(P(J).LT.1.0E-20) P(J)=1.0E-20
  220 RK(J)=P(J)+P(J)**2/2.0+P(J)**3/3.0
 2000 CONTINUE
C
 3000 CONTINUE
      WRITE(6,598)
  598 FORMAT(1H ,/,' I,T,P(I),TP(I)',/)
      DO 321 I=1,NNN
      TTO=TI*FLOAT(I)
      IF(P(I).GT.0.0001) GO TO 324
      TP(I)=FLOAT( NF)*P(I)-FLOAT(NF)*FLOAT(NF-1)*P(I)**2/2.0
      GO TO 325
  324 IF(P(I).GT.0.9)  GO TO 326
      TP(I)=1.0-(1.0-P(I))**NF
      GO TO 325
  326 TP(I)=1.0
  325 CONTINUE
      WRITE(6,599) I,TTO,P(I),TP(I)
  599 FORMAT(1H ,I5,F10.1,2E15.5)
  321 CONTINUE
      GO TO 81
  322 STOP
      END
      SUBROUTINE HCOM(H,AR0,AR1,AR2,AR3,AR4,AR5,
     1AR6,AR7,AR8,P,AN)
C CAL. FOR COMPOSITE GAUSSIAN
      H=0.5*AN*P*((1.0-ERF(AR0))+EXP(AR1)*(1.0+ERF(AR2)))
      RETURN
      END
      SUBROUTINE HSIN(H,AR0,AR1,AR2,AR3,AR4,AR5,
     1AR6,AR7,AR8,P,AN,UPC)
C CAL. FOR SINGLE GAUSSIAN
      IF(UPC.EQ.1.0) GO TO 10
C COMPLETE SINGLE GAUSSIAN
      H=0.5*AN*P*((1.0-ERF(AR3))+AR4
     1*EXP(AR5+AR6)*(1.0+ERF(AR7)))
      GO TO 25
C UPPER CUT SINGLE GAUSSIAN
   10 H=0.5*AN*P*((1.0-ERF(AR3))+AR4
     1*EXP(AR5+AR6)*(ERF(AR7)+ERF(AR8)))
   25 RETURN
      END
      FUNCTION EEXP(ARG)
      A=0.01
      B=-50.0
      C=1.0E-20
      IF(ABS(ARG).LT.C) GO TO 5
      IF(ABS(ARG).LT.A) GO TO 10
      IF(ARG.LT.B) GO TO 30
      EEXP=EXP(ARG)-1.0
      GO TO 20
    5 EEXP=ARG
      GO TO 20
   10 EEXP=ARG+ARG*ARG/2.0+ARG*ARG*ARG/6.0
      GO TO 20
   30 EEXP=-1.0
   20 CONTINUE
      RETURN
```

```
      END
      FUNCTION F(X,A1,A2,RM,U)
      IF(X.LT.A1) GO TO 10
      IF(X.GT.A2) GO TO 20
      F=((X-A1)/(A2-A1))**RM
      GO TO 1
   10 F=0.0
      GO TO 1
   20 F=1.0
    1 CONTINUE
      F=U*F
      RETURN
      END
      FUNCTION FS(X)
      FS=1.0-F(X)
      RETURN
      END
      FUNCTION ERF(X)
      IF(ABS(X).GT.10.0 )   GO TO 3
      X=X*1.414214
      T=1.0/(1.0+.2316419*ABS(X))
      D=0.398942 *EXP(-X*X/2.0)
      ERF=2.*(0.5-D*T*(((((1.33274*T-1.821256)*T+1.781478)*T
     1-0.356563 )*T+0.319381 ))
      IF(X) 1,2,2
    1 ERF=-ERF
      GO TO 2
    3 ERF=1.0
      IF(X.LT.0.0) ERF=-ERF
    2 RETURN
      END
      SUBROUTINE QSF(H,Y,Z,NDIM)
C
C
      DIMENSION Y(1),Z(1)
C
      HT=.3333333*H
      IF(NDIM-5)7,8,1
C
C     NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION LOOP
    1 SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+Y(6))
      SUM2=Y(5)+Y(5)
      SUM2=SUM2+SUM2
      SUM2=AUX2-HT*(Y(4)+SUM2+Y(5))
      Z(1)=0.
      AUX=Y(3)+Y(3)
      AUX=AUX+AUX
      Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
      Z(3)=SUM1
      Z(4)=SUM2
      IF(NDIM-6)5,5,2
C
C     INTEGRATION LOOP
    2 DO 4 I=7,NDIM,2
```

```
      SUM1=AUX1
      SUM2=AUX2
      AUX1=Y(I-1)+Y(I-1)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
      Z(I-2)=SUM1
      IF(I-NDIM)3,6,6
    3 AUX2=Y(I)+Y(I)
      AUX2=AUX2+AUX2
      AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
    4 Z(I-1)=SUM2
    5 Z(NDIM-1)=AUX1
      Z(NDIM)=AUX2
      RETURN
    6 Z(NDIM-1)=SUM2
      Z(NDIM)=AUX1
      RETURN
C     END OF INTEGRATION LOOP
C
    7 IF(NDIM-3)12,11,8
C
C     NDIM IS EQUAL TO 4 OR 5
    8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))
      SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      Z(1)=0.
      AUX1=Y(3)+Y(3)
      AUX1=AUX1+AUX1
      Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
      IF(NDIM-5)10,9,9
    9 AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
   10 Z(3)=SUM1
      Z(4)=SUM2
      RETURN
C
C     NDIM IS EQUAL TO 3
   11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
      SUM2=Y(2)+Y(2)
      SUM2=SUM2+SUM2
      Z(3)=HT*(Y(1)+SUM2+Y(3))
      Z(1)=0.
      Z(2)=SUM1
   12 RETURN
      END
//GO.SYSIN DD *
123.0
-123.0
/*
```

```
//    EXEC FORTHCG
//FORT.SYSIN DD *,DCB=BLKSIZE=3440
C  AREL2
C  SENSITIVITY STUDY IS AVAILABLE
C  DELTA K THRESHOLD IS AVAILABLE
C  THREE LOADS (COMPOSIT OR SINGLE GAUSSIAN) ARE AVAILABLE
C  ANY TOTAL FLIGHT HOUR IS AVAILABLE
      DIMENSION AA(10),AAA(10),BAB(10)
      DIMENSION AAAA(10)
      DIMENSION A(1501),R(1501),H(1501),TH(1501)
      DIMENSION PP(3),SGC(3),RR(3),ANETA(3)
      DIMENSION CMP(3),SGS(3),VX(3),AN(3),UPC(3),SB(3)
      DIMENSION D(126),GA(126,15),GTH(126,15),ITD(15)
      DIMENSION GME(125)
      COMMON A1,A2,RM,U,NT,NPT,DA,NF,PME
      COMMON/QWE/  ALPHA,BETA,EFD,CDS,D,ANS,BE
      COMMON/POI/  CA1,NC,NR
   31 CONTINUE
C****************************************************************
C****************** MEAN VALUES *********************************
C****************** FRACTURE DATA *******************************
      FF=2.5
      BE=4.0
      DKTH=1.5
      EFD=10.0
      C=3.0E-7
      KC=60
C***************** INITIAL CRACK DATA *****************************
      NBN=125
      A0=0.04
      AU=A0
      AL=0.015
      ALPHA=4.0
      BETA=30000.0
C***************** LOAD DATA *************************************
C  CMP (=1.0 COMPOSIT GAUSSIAN  ,=0.0 SINGLE ONE)
      CMP(1)=1.0
      CMP(2)=1.0
      CMP(3)=1.0
C  UPC  (=1.0 UPPER CUT SINGLE GAUSSIAN  ,=0.0 COMPLETE ONE)
      UPC(1)=0.0
      UPC(2)=0.0
      UPC(3)=0.0
C  EL   UPPER CUT LEVEL OF SINGLE GAUSSIAN (EL*X0)
      EL=8.0
      G=10.0
      AA(4)=115.0
      AAA(4)=115.0
      AAAA(4)=80.0
      BAB(4)=3.0
C  AN(I) NUMBER OF LOAD CYCLES PER HOUR
      AN(1)=600.0
      AN(2)=600.0
      AN(3)=60.0
      ANA=0.5
      PP(1)=0.4975
      PP(2)=0.0025
      PP(3)=0.5
C  SGC  STANDARD DEVIATION OF COMPOSITE GAUSSIAN
      SGC(1)=0.07*G
```

```
      SGC(2)=0.18*G
      SGC(3)=0.1*G
C SGS STANDARD DEVIATION OF SINGLE GAUSSIAN
      SGS(1)=1.5*G
      SGS(2)=1.5*G
      SGS(3)=1.5*G
      X0=1.0*G
      ZR=1.5
C**************** MATERIAL DATA ****************************
      V0=0.056
      AMU0=5.7*G
      R0=5.7*G
C**************** FAIL SAFE DATA ***************************
      FSF=1.0
      ZETA=0.43
      AS=7.0
C**********************************************************
C TT   TOTAL FLIGHT HOUR
      TT=15000.0
C************** INSPECTION DATA ****************
      A1=0.02
      A2=0.3
      RM=0.125
      U=1.0
C NN   NOMBER OF INSPECTIONS
C TT/(NN+1) MUST BE INTEGER
C (TT/(NN+1))/(TT/(NPT-1)) MUST BE INTEGER
      NN=5
C IF # OF INSPEC. GT. 5  GA( ,6),GTH( ,6),ITD(6) MUST BE CHANGED
C # OF INSPEC.=NN  GA( ,NN+1),GTH( ,NN+1),ITD(NN+1)
C************** PROOF TEST DATA ****************
      ACT=999.0
C ACT  CUT LEVEL OF INI. CRACK DISTRIBUTION
C ACT.GT.AU  MEANS NO PROOF TEST PERFORMED
C**********************************************************
C************** CURSORY INSPEC. DATA ***************
C     SRY=0.0
      NR=1
      NC=4
      CA1=0.6
C**********************************************************
      PI=3.14159
C NPT  BASIC DATA POINTS
C EACH DIMENSION SIZE OF(A,R,H,TH,DUMY,W)MUST BE G.E. NPT
C (TT/(NPT-1)) MUST BE INTEGER
      NPT=1501
C NOI  DATA POINTS UNDER NO INSPECTION CASE
C EACH DIMENSION SIZE OF(TP P,PS,RK,CIJ,CTJ)MUST BE G.E. NOI AND (NN+1)
C (TT/NOI) MUST BE INTEGER
C (TT/NOI)/(TT/(NPT-1)) MUST BE INTEGER
      NOI=30
C************** FLEET SIZE *********************
      NF=50
C**********************************************************
C**********************************************************
C**********************************************************
C SUB PROGRAM FOR SENSITIVITY STUDY
      READ (5,501) XY,YZ1,YZ2,YZ3
  501 FORMAT(4F10.0)
      WRITE(6,650) XY,YZ1,YZ2,YZ3
```

```
650  FORMAT(1H ,///,' XY,YZ1,YZ2,YZ3',4F15.7)
     IF(XY.LE.0.0) GO TO 322
     IF(XY.EQ.1.0) EE=YZ1
     IF(XY.EQ.2.0)DKTH=YZ1
     IF(XY.EQ.3.0) EEO=YZ1
     IF(XY.EQ.4.0) C=YZ1
     IF(XY.EQ.5.0) AO=YZ1
     IF(XY.EQ.5.0) AU=YZ1
     IF(XY.EQ.6.0) AN(1)=YZ1
     IF(XY.EQ.6.0) AN(2)=YZ1
     IF(XY.EQ.7.0) ANA=YZ1
     IF(XY.EQ.8.0) AN(3)=YZ1
     IF(XY.EQ.9.0) VO=YZ1
     IF(XY.EQ.10.0) ZETA=YZ1
     IF(XY.EQ.11.0) AS=YZ1
     IF(XY.EQ.12.0)  KC=YZ1
     IF(XY.NE.13.0) GO TO 721
     PP(1)=YZ1
     PP(2)=YZ2
     PP(3)=YZ3
721  CONTINUE
     IF(XY.NE.14.0) GO TO 722
     SGC(1)=YZ1
     SGC(2)=YZ2
     SGC(3)=YZ3
722  CONTINUE
     IF(XY.NE.15.0) GO TO 723
     G=YZ1
     SGC(1)=0.07*G
     SGC(2)=0.18*G
     SGC(3)=0.1*G
     XO=1.0*G
     RO=5.7*G
     AMUO=5.7*G
723  CONTINUE
     IF(XY.NE.16.0) GO TO 724
     AMUO=YZ1
     RO=YZ1
724  CONTINUE
     IF(XY.EQ.17.0) ALPHA=YZ1
     IF(XY.EQ.18.0) BETA=YZ1
     IF(XY.EQ.19.0) FSF=YZ1
     IF(XY.EQ.20.0) BE=YZ1
     IF(XY.EQ.21.0) AA(4)=YZ1
     IF(XY.EQ.21.0) AAA(4)=YZ1
     IF(XY.EQ.22.0) AAAA(4)=YZ1
     IF(XY.EQ.23.0) Z3=YZ1
     IF(XY.NE.24.0) GO TO 725
     SGS(1)=YZ1
     SGS(2)=YZ2
     SGS(3)=YZ3
725  CONTINUE
     IF(XY.NE.25.0) GO TO 726
     CMP(1)=YZ1
     CMP(2)=YZ2
     CMP(3)=YZ3
726  CONTINUE
     IF(XY.EQ.26.0) A1=YZ1
     IF(XY.EQ.27.0) A2=YZ1
     IF(XY.EQ.28.0) RM=YZ1
```

```
      IF(XY.EQ.29.0) U=YZ1
      IF(XY.EQ.30.0) NN=YZ1
      IF(XY.NE.31.0) GO TO 727
      UPC(1)=YZ1
      UPC(2)=YZ2
      UPC(3)=YZ3
  727 CONTINUE
      IF(XY.EQ.32.0) EL=YZ1
      IF(XY.EQ.33.0) ACT=YZ1
      IF(XY.EQ.34.0) CA1=YZ1
C SJ3 PROGRAM FOR SENSITIVITY STUDY ***END
C*****************************************************************
      JKL=0
      KBE=BE
      IBE=BE
      IF(CMP(1).EQ.1.0)  SB(1)=AA(IBE)*BAB(IBE)*SGC(1)**KBE
      IF(CMP(1).EQ.0.0)  SB(1)=AA(IBE)            *SGS(1)**KBE
      IF(CMP(2).EQ.1.0)  SB(2)=AAA(IBE)*BAB(IBE)*SGC(2)**KBE
      IF(CMP(2).EQ.0.0)  SB(2)=AAA(IBE)           *SGS(2)**KBE
      IF(CMP(3).EQ.0.0) SB(3)=AAAA(IBE)           *SGS(3)**KBE
      IF(CMP(3).EQ.1.0) SB(3)=AAAA(IBE)*BAB(IBE)*SGC(3)**KBE
      ZBB=(ZR*G)**KBE
      Q=AN(1)*PP(1)*SB(1)+AN(2)*PP(2)*SB(2)+AN(3)*PP(3)*SB(3)
     1+ANA*ZBB
      AC=2./PI* (KC/RO)**2
      DT=TT/FLOAT(NPT-1)
      ANS=PP(1)*AN(1)+PP(2)*AN(2)+PP(3)*AN(3)
C ANS=CYCLES PER HOUR
C
      IF(EF.EQ.2.0) EF=2.01
      IF(EFD.EQ.2.0) EFD=2.01
C
      ANTH=ALOG10(C)+EF*ALOG10(DKTH)
C
      CDST=ANTH-EFD*ALOG10(DKTH)
      CDS=10.0**CDST
C********************************
C********************************
      IF(CSRY.EQ.1.0) NN=NR+(NR+1)*NC
      IF(NN.EQ.0) GO TO 24
      NN1=NN+1
      DO 20 I=1,NN1
   20 ITO(I)=I*(NPT-1)/(NN+1)+1
      GO TO 22
   24 CONTINUE
      DO 23 I=1,6
   23 ITO(I)=I*(NPT-1)/6+1
      NN1=6
   22 CONTINUE
      NBN1=NBN+1
      NT=NBN1
      CALL DENST(Q,NT,AU,AL,DA)
      IF(ACT.GT.AU) PMF=1.0
      IF(ACT.GT.AU) GO TO 830
      NTC=(AU-ACT)/DA
      NTT=NT-NTC
      DO 831 I=NTT,NT
      Q(I)=0.0
  831 CONTINUE
      CALL QSF(DA,Q,GME,NT)
```

```
      PME=1.0/GMF(NT)
  830 CONTINUE
C*****************************************
C*****************************************
      KILE=9999
C KILE=9999 MEANS 'CRACK' TAKEN OFF
      DO 31 KIL=1,NBN1
      A0=AU-FLOAT(KIL-1)*DA
C
      ATH=(DKTH/(Q/ANS)**(1.0/BE))**2*(2.0/PI)
      TB=2.0/(2.0-EFD)/CDS/Q**(EFD/BE)*ANS**((EFD-BE)/BE)
      TB=TB/(PI/2.0)**(EFD/2.0)
      TB=TB*(ATH**((2.0-EFD)/2.0)-A0**((2.0-EFD)/2.0))
C
      ECQ=((2.0-EF)/2.0)*C*Q**(EF/BE)*ANS**((BE-EF)/BE)
     1*(PI/2.0)**(EF/2.0)
      ECQD=((2.0-EFD)/2.0)*CDS*Q**(EFD/BE)*ANS**((BE-EFD)/BE)
     1*(PI/2.0)**(EFD/2.0)
      DO 8 I=1,NPT
      T=FLOAT(I-1)*DT
      IF(TB.GT.0.0) GO TO 43
C IN CASE OF ATH.LT.A0
      A(I)=A0**((2.0-EF)/2.0)
      A(I)=A(I)+ECQ*T
      IF(A(I).LT.0.00001) GO TO 45
      A(I)=A(I)**(2.0/(2.0-EF))
      GO TO 48
   43 CONTINUE
      IF(T.GT.TB) GO TO 49
C IN CASE OF ATH.GT.A0
      A(I)=A0**((2.0-EFD)/2.0)
      A(I)=A(I)+ECQD*T
      IF(A(I).LT.0.00001) GO TO 45
      A(I)=A(I)**(2.0/(2.0-EFD))
C
      GO TO 48
C
   49 A(I)=ATH**((2.0-EF)/2.0)
      A(I)=A(I)+ECQ*(T-TB)
      IF(A(I).LT.0.00001) GO TO 45
      A(I)=A(I)**(2.0/(2.0-EF))
      GO TO 48
   45 A(I)=10000.0
C
   48 IF(FSF.EQ.0.0) AS=1000.0
      IF(ABS(A(I)).GT.AS) GO TO 15
      IF(KILE.NE.9999) GO TO 415
    8 CONTINUE
      GO TO 46
   15 KK=I
      DO 60 I=KK,NPT
   60 A(I)=AS
   46 CONTINUE
      GO TO 416
  415 DO 417 I=2,NPT
  417 A(I)=A(1)
  416 CONTINUE
C ### A(I) CAL. END ###
      IF(KILE.NE.9999) GO TO 430
      ABC=A0*1.2
```

```
      IF(A(NPT).LT.ABC) KILE=KIL+1
  430 CONTINUE
      IF(ATH.GT.AC) GO TO 353
      TC=2.0/(2.0-EF )*(AC**((2.0-EF )/2.0)-ATH**((2.0-EF )/2.0))
      TC=TC/C/Q**(EF /BE)/ANS**((BE-EF )/BE)/(PI/2.0)**(EF /2.0)
      TC=TC+TB
      GO TO 364
  353 TC=2.0/(2.0-EFD)*(AC**((2.0-EFD)/2.0)- AO**((2.0-EFD)/2.0))
      TC=TC/CDS/Q**(EFD/BE)/ANS**((BE-EFD)/BE)/(PI/2.0)**(EFD/2.0)
  364 CONTINUE
  800 CONTINUE
  801 CONTINUE
C
C RESIDUAL STRENGTH RATE ( FSF=1.0(FAIL SAFE)  =0.0(NO FAIL SAFE)
      DO 431 I=1,NPT
      IF(FSF.EQ.1.0) GO TO 189
      IF(A(I).LE.AC) GO TO 188
      R(I)=FLOAT(KC)*SQRT(2.0/PI/A(I))/RO
      GO TO 18
  188 R(I)=1.0
      GO TO 18
  189 SW=ABS((A(I)-AO)/(AS-AO))
      R(I)=1.0-(1.0-ZETA)*SQRT(SW)
   18 IF(KILE.NE.9999) GO TO 418
  431 CONTINUE
      GO TO 419
  418 DO 420 I=2,NPT
  420 R(I)=R(1)
  419 CONTINUE
C
C $$$ CAL. OF H $$$
      DO 115 N=1,NPT
      HWA=0.0
      GMN=R(N)
      DO 116 I=1,3
      IF(CMP(I).EQ.0.0) SGC(I)=SGS(I)
      VX(I)=SGC(I)/XO
      ANETA(I)=GMN*AMUO/SGC(I)-XO/SGC(I)
      RR(I)=VO*GMN*AMUO/SGC(I)
      IF(CMP(I).NE.1.0) GO TO 21
      ARG0=ANETA(I)/SQRT(2.0)/RR(I)
      ARG1=-ANETA(I)+RR(I)**2/2.0
      ARG2=(ANETA(I)-RR(I)**2)/SQRT(2.0)/RR(I)
      IF(CMP(I).EQ.1.0) CALL HCOM(H(N),ARG0,ARG1,ARG2,ARG3,ARG4,
     1ARG5,ARG6,ARG7,ARG8,PP(I),AN(I) )
      GO TO 411
   21 CONTINUE
      ARG3=ANETA(I)/RR(I)/SQRT(2.0)
      ARG4=1.0/SQRT(1.0+RR(I)**2)
      ARG5=-0.5*(1.0/VX(I)**2+1.0/VO**2)
      ARG6=0.5*(RR(I)/VX(I)+1.0/VO)**2/(1.0+RR(I)**2)
      ARG7=ANETA(I)/SQRT(2.0)/RR(I)/SQRT(1.0+RR(I)**2)
      ARG8=1.0/SQRT(2.0)/SQRT(RR(I)**2+1.0)
     1*((EL-1.0)*RR(I)/VX(I)+EL/RR(I)/VX(I)-1.0/VO)
      IF(CMP(I).EQ.0.0) CALL HSIN(H(N),ARG0,ARG1,ARG2,ARG3,ARG4,
     1ARG5,ARG6,ARG7,ARG8,PP(I),AN(I),JPC(I) )
  411 CONTINUE
      HWA=HWA+H(N)
  116 CONTINUE
      H(N)=HWA
```

```
      IF(H(N).GT.1.0) H(N)=1.0
      IF(KILE.NE.9999) GO TO 421
  115 CONTINUE
      GO TO 422
  421 DO 423 I=2,NPT
  423 H(I)=H(1)
  422 CONTINUE
C ### CAL. OF H END ###
      CALL OSF(DT,H,TH,NPT)
      KIP=NBN1-(KIL-1)
      DO 25 I=1,NN1
      GA(KIP,I)=A(ITO(I))
      GTH(KIP,I)=TH(ITO(I))
   25 CONTINUE
   31 CONTINUE
C******************************
C SUBROUTINE CIN   CURSORY+RIGOROUS INSPECTIONS
      IF(CSRY.EQ.1.0) CALL CIN(GA,GTH,O,CA1,NC,NR,NN,ITO)
      IF(CSRY.EQ.1.0) GO TO 81
C******************************
C SUBROUTINE HIN   RIGOROUS INSPECTION
      CALL HIN(GA,GTH,O,NN,ITO)
      GO TO 81
  322 STOP
      END
      SUBROUTINE HCOM(H,AR0,AR1,AR2,AR3,AR4,AR5,
     1AR6,AR7,AR8,P,AN)
C CAL. FOR COMPOSITE GAUSSIAN
      H=0.5*AN*P*((1.0-ERF(AR0))+EXP(AR1)*(1.0+ERF(AR2)))
      RETURN
      END
      SUBROUTINE HSIN(H,AR0,AR1,AR2,AR3,AR4,AR5,
     1AR6,AR7,AR8,P,AN,UPC)
C CAL. FOR SINGLE GAUSSIAN
      IF(UPC.EQ.1.0) GO TO 10
C COMPLETE SINGLE GAUSSIAN
      H=0.5*AN*P*((1.0-ERF(AR3))+AR4
     1*EXP(AR5+AR6)*(1.0+ERF(AR7)))
      GO TO 25
C UPPER CUT SINGLE GAUSSIAN
   10 H=0.5*AN*P*((1.0-ERF(AR3))+AR4
     1*EXP(AR5+AR6)*(ERF(AR7)+ERF(AR8)))
   25 RETURN
      END
      FUNCTION ERF(X)
      IF(ABS(X).GT.10.0) GO TO 3
      X=X*1.414214
      T=1.0/(1.0+.2316419*ABS(X))
      D=0.398942 *EXP(-X*X/2.0)
      ERF=2.*(0.5-D*T*((((1.33274*T-1.821256)*T+1.781478)*T
     1-0.356563 )*T+0.319381  ))
      IF(X) 1,2,2
    1 ERF=-ERF
      GO TO 2
    3 ERF=1.0
      IF(X.LT.0.0) ERF=-ERF
    2 RETURN
      END
      SUBROUTINE HIN(A,H,G,NN,ITO)
      DIMENSION A(126,15),H(126,15),G(126), ITO(125),PX(126),PA(15)
```

```
      DIMENSION TPA(15)
      COMMON A1,A2,RM,U,NT,NPT,DA,NF,PME
C NP=1 MEANS SKIP TO WRITE
      NP=1
      N=NN+1
      WRITE(6,600) NN
  600 FORMAT(1H ,'NN=',I3)
      IF(NN.NE.0) GO TO 987
C*************************************************
C*************************************************
C*** NO INSPECTION CASE ***
      N=6
      DO 900 I=1,N
      DO 901 J=1,NT
  901 PX(J)=OMEXP(-H(J,I))*G(J)
      CALL QSF(DA,PX,PX,NT)
      PA(I)=PX(NT)
      PA(I)=PA(I)*PME
      IF(PA(I).GT.0.0001) GO TO 324
      TPA(I)=FLOAT(NF)*PA(I)-FLOAT(NF)*FLOAT(NF-1)*PA(I)**2/2.0
      GO TO 325
  324 IF(PA(I).GT.0.9) GO TO 326
      TPA(I)=1.0-(1.0-PA(I))**NF
      GO TO 325
  326 TPA(I)=1.0
  325 CONTINUE
  900 CONTINUE
      WRITE(6,620)
  620 FORMAT(1H ' RESULT OF NO INSPEC. CASE')
      DO 902 I=1,N
      IT=((NPT-1)/N)*I+1
      WRITE(6,621) IT,PA(I),TPA(I)
  621 FORMAT(1H ,' IT=',I5,3X,' PA=',E15.7,3X,'TPA=',E15.7)
  902 CONTINUE
      GO TO 17
C
  987 CONTINUE
C*************************************************
C*************************************************
C*** INSPECTION CASE ***
      IF(NP.EQ.1) GO TO 738
      WRITE(6,699) (G(I),I=1,NT)
  699 FORMAT(1H ,5E15.7)
      DO 10 I=1,NT
      WRITE(6,666) ( A(I,M),M=1,6)
  666 FORMAT(1H ,6E15.7)
   10 CONTINUE
      DO 20 I=1,NT
      WRITE(6,666) ( H(I,M),M=1,6)
   20 CONTINUE
  738 CONTINUE
C    CALCULATE THE FIRST PROB.
      DO 240 I=1,NT
  240 PX(I)=OMEXP(-H(I,1))*G(I)
      CALL QSF(DA,PX,PX,NT)
      PA(1)=PX(NT)
      J=1
      IF(NP.EQ.1) GO TO 888
  280 WRITE(6,170) J,PA(1)
  170 FORMAT( /'PA',I4,' = ',E14.7)
```

```
  888 CONTINUE
C*************************
C       CALCULATE THE 2 - N PROB.
        DO 100 J=2,N
        DO 140 I=1,NT
C **    FIRST TERM
        RH1=H(I,1)
        RA1=A(I,1)
        RF1=FF(RA1)
        TER1=OMEXP(-RH1)
        AR1=RF1*PA(J-1)
        TER2=(1.-TER1)*AR1
        TERM1=TER1+TER2
        IF (NP .EQ. 1) GO TO 260
        WRITE(6,270) RH1,RA1,AR1,TER1,TER2,TERM1
  270   FORMAT(6X,E12.5,2F8.5,2E12.5,E14.7)
C **    THIRD TERM
  260   FA=0.
        JM1=J-1
        DO 250 KK=1,JM1
        M=J-KK
        RA3=A(I,M)
        RF3=1.-FF(RA3)
        RH3=H(I,M)
        RH4=H(I,M+1)
        TER3=RH4-RH3
        TER4=RF3*((1.-OMEXP(-RH3))*OMEXP(-TER3)+FA)
        IF (NP .EQ.1) GO TO 250
        WRITE(6,350) RH3,RH4,RA3,RF3,TER3,TER4
  350   FORMAT( 6X,2E12.5,2F8.5,E12.5,E14.7)
  250   FA=TER4
        TERM3=FA
C **    SECOND TERM
        TERM2=0.
        IF(J.LE.2)GO TO 200
        J2=J-2
        FA=1.0
        DO 550 K=1,J2
        RH2=H(I,K+1)
        RA2=A(I,K+1)
        RA=A(I,K)
        FA=FA*(1.-FF(RA))
        RF2=FF(RA2)
        ARG2=1.-OMEXP(-RH2)
        TERM=FA*RF2*ARG2* PA(J-K-1)
        TERM2=TERM2+TERM
        IF (NP .EQ.1) GO TO 550
        WRITE(6,160) RH2,RA2,RA,FA,RF2,ARG2,TERM
  160   FORMAT(2X,E12.5,4F8.5,E12.5,E14.7)
  550   CONTINUE
C       SUM OF THE ALL TERMS
  200   TERM4=TERM1+TERM2+TERM3
        PX(I)=TERM4*G(I)
        IF (NP .EQ. 1) GO TO 140
        WRITE(6,180) I,TERM4,G(I),PX(I)
  180   FORMAT(I4,'***',3E14.7)
  140   CONTINUE
        CALL QSF(DA,PX,PX,NT)
        PA(J)=PX(NT)
        IF(NP.EQ.1) GO TO 100
```

```
      WRITE(6,170) J,PA(J)
  100 CONTINUE
      DO 111 J=1,N
      PA(J)=PA(J)*PME
  111 CONTINUE
      DO 887 J=1,N
      IF(PA(J).GT.0.0001) GO TO 334
      TPA(J)=FLOAT(NF)*PA(J)-FLOAT(NF)*FLOAT(NF-1)*PA(J)**2/2.0
      GO TO 335
  334 IF(PA(J).GT.0.9) GO TO 336
      TPA(J)=1.0-(1.0-PA(J))**NF
      GO TO 335
  336 TPA(J)=1.0
  335 CONTINUE
  887 CONTINUE
      I=NT
      WRITE(6,510)
  510 FORMAT(4X,1HT,12X,'A',12X,'H',14X,'PA',11X,'TPA')
      WRITE(6,520) (ITO(J),A(I,J),H(I,J),PA(J), TPA(J),J=1,N)
  520 FORMAT(I5,4E14.7)
   17 CONTINUE
      RETURN
      END
      SUBROUTINE CIN(A,H,G,CA1,NC,NR,NN,ITO)
      DIMENSION A(126,15),H(126,15),G(126),ITO(15),PX(126),PA(15)
      DIMENSION TPA(15)
      COMMON A1,A2,RM,U,NT,NPT,DA,NF,PME
C NP=1 MEANS SKIP TO WRITE
      N
      WRITE(6,600) NN
  600 FORMAT(1H ,'NN=',I3)
      IF(NP.EQ.1) GO TO 738
      WRITE(6,699) (G(I),I=1,NT)
  699 FORMAT(1H ,5E15.7)
      DO 10 I=1,NT
      WRITE(6,666) ( A(I,M),M=1,6)
  666 FORMAT(1H ,6E15.7)
   10 CONTINUE
      DO 20 I=1,NT
      WRITE(6,666) (  H(I,M),M=1,6)
   20 CONTINUE
  738 CONTINUE
C   CALCULATE THE FIRST PROB.
      DO 240 I=1,NT
  240 PX(I)=OMEXP(-H(I,1))*G(I)
      CALL QSF(DA,PX,PX,NT)
      PA(1)=PX(NT)
      J=1
      IF(NP.EQ.1) GO TO 888
  230 WRITE(6,170) J,PA(1)
  170 FORMAT( /'PA',I4,' = ',E14.7)
  888 CONTINUE
C*********************
C     CALCULATE THE 2 - N PROB.
      DO 100 J=2,N
      DO 140 I=1,NT
C **  FIRST TERM
      RH1=H(I,1)
      RA1=A(I,1)
      NC1=NC+1
```

```
      JM1=J-1
      JO=JM1/NC1
      JOO=JO*NC1
      IF(JOO.EQ.JM1) RF1=FF(RA1)
      IF(JOO.NE.JM1) RF1=FC(RA1)
      TER1=OMEXP(-RH1)
      AR1=RF1*PA(J-1)
      TER2=(1.-TER1)*AR1
      TERM1=TER1+TER2
      IF (NP .EQ. 1) GO TO 260
      WRITE(6,270) RH1,RA1,AR1,TER1,TER2,TERM1
  270 FORMAT(6X,E12.5,2F8.5,2E12.5,E14.7)
C **  THIRD TERM
  260 FA=0.
      JM1=J-1
      DO 250 KK=1,JM1
      M=J-KK
      RA3=A(I,M)
      NC1=NC+1
      JMM=J-M
      JY=JMM/NC1
      JYY=JY*NC1
      IF(JYY.EQ.JMM) RF3=1.-FF(RA3)
      IF(JYY.NE.JMM) RF3=1.-FC(RA3)
      RH3=H(I,M)
      RH4=H(I,M+1)
      TER3=RH4-RH3
      TER4=RF3*((1.-OMEXP(-RH3))*OMEXP(-TER3)+FA)
      IF (NP .EQ.1) GO TO 250
      WRITE(6,350) RH3,RH4,RA3,RF3,TER3,TER4
  350 FORMAT( 6X,2E12.5,2F8.5,E12.5,E14.7)
  250 FA=TER4
      TERM3=FA
C **  SECOND TERM
      TERM2=0.
      IF(J.LE.2)GO TO 200
      J2=J-2
      FA=1.0
      DO 550 K=1,J2
      RH2=H(I,K+1)
      RA2=A(I,K+1)
      RA=A(I,K)
      NC1=NC+1
      JMK=J-K
      JT=JMK/NC1
      JTT=JT*NC1
      IF(JMK.EQ.JTT) FA=FA*(1.-FF(RA))
      IF(JMK.NE.JTT) FA=FA*(1.-FC(RA))
      JMK1=J-(K+1)
      JU=JMK1/NC1
      JUU=JU*NC1
      IF(JUU.EQ.JMK1) RF2=FF(RA2)
      IF(JUU.NE.JMK1) RF2=FC(RA2)
      ARG2=1.-OMEXP(-RH2)
      TERM=FA*RF2*ARG2* PA(J-K-1)
      TERM2=TERM2+TERM
      IF (NP .EQ.1) GO TO 550
      WRITE(6,160) RH2,RA2,RA,FA,RF2,ARG2,TERM
  160 FORMAT(2X,E12.5,4F8.5,E12.5,F14.7)
  550 CONTINUE
```

```
C         SUM OF THE ALL TERMS
  200     TERM4=TERM1+TERM2+TERM3
          PX(I)=TERM4*G(I)
          IF (NP .EQ. 1) GO TO 140
          WRITE(6,180) I,TERM4,G(I),PX(I)
  180     FORMAT(I4,'**',3E14.7)
  140     CONTINUE
          CALL QSF(DA,PX,PX,NT)
          PA(J)=PX(NT)
          IF(NP.EQ.1) GO TO 100
          WRITE(6,170) J,PA(J)
  100     CONTINUE
          DO 111 J=1,N
          PA(J)=PA(J)*PME
  111     CONTINUE
          DO 887 J=1,N
          IF(PA(J).GT.0.0001) GO TO 334
          TPA(J)=FLOAT(NF)*PA(J)-FLOAT(NF)*FLOAT(NF-1)*PA(J)**2/2.0
          GO TO 335
  334     IF(PA(J).GT.0.9) GO TO 336
          TPA(J)=1.0-(1.0-PA(J))**NF
          GO TO 335
  336     TPA(J)=1.0
  335     CONTINUE
  887     CONTINUE
          NR1=NR+1
          DO 26 I=1,NR1
          J=(NC+1)*I
          WRITE(6,670) I,PA(J),I,TPA(J),ITO(J)
  670     FORMAT(1H ,' P(',I2,')=',E15.7,7X,'TP(',I2,')=',F15.7,10X,I5)
   26     CONTINUE
   17     CONTINUE
          RETURN
          END
          FUNCTION FF(X)
          COMMON A1,A2,RM,J
          IF(X.LE.A1) GO TO 10
          IF(X.GE.A2) GO TO 20
          FF=U*((X-A1)/(A2-A1))**RM
          RETURN
   10     FF=0.0
          RETURN
   20     FF=1.0
          RETURN
          END
          SUBROUTINE DENST(G,  NN,AU,AL,DA)
          DIMENSION G(1)
          COMMON/QWE/ ALPHA,BETA,EFD,CDS,Q,ANS,BE
          C1=CDS
          S=EFD
          FNO=ANS
          AREA=ALPHA
          B=BE
          PI=3.14159
          S2=S/2.
          OMS2=1.-S2
          ASP=AU**OMS2
          H1=C1*(PI/2.)**S2*Q**(S/3)*FNO**(1.-S/B)
          H2=OMS2*H1
          DA=(AU-AL)/FLOAT(NN-1)
```

```
      DO 10 K=1,NN
      AK=AU-DA*FLOAT(K-1)
      ARK=AK/AU
      DELT=1.-ARK
      TK=(ASP-AK**OMS2)/H2
      L=NN-(K-1)
      G(L)=ARFA*AK**(-S2)*(TK/BETA)**(ARFA-1)*
     1EXP(-(TK/BETA)**ARFA)/(H1*BETA)
      IF(G(L).EQ.0.0) GO TO 10
      IF(G(L).LT.1.0E-50) GO TO 20
   10 CONTINUE
      GO TO 40
   20 CONTINUE
      DO 30 I=1,L
   30 G(I)=0.0
   40 RETURN
      END
      FUNCTION OMEXP(X)
      B=-50.0
      C=1.0E-20
      IF(ABS(X).LT.C) GO TO 5
      IF (ABS(X) .LT. 0.01 ) GO TO 10
      IF(X.LT.B) GO TO 30
      OMEXP=1.-EXP(X)
      RETURN
    5 OMEXP=-0.999999*X
      RETURN
   10 OMEXP=-0.166666*X*X*X-0.499999*X*X-0.999999*X
      RETURN
   30 OMEXP=1.
      RETURN
      END
      SUBROUTINE QSF(H,Y,Z,NDIM)
C
C
      DIMENSION Y(1),Z(1)
C
      HT=.3333333*H
      IF(NDIM-5)7,8,1
C
C     NDIM IS GREATER THAN 5. PREPARATIONS OF INTEGRATION LOOP
    1 SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(3)+AUX1+Y(5))
      AUX2=HT*(Y(1)+3.875*(Y(2)+Y(5))+2.625*(Y(3)+Y(4))+Y(6))
      SUM2=Y(5)+Y(5)
      SUM2=SUM2+SUM2
      SUM2=AUX2-HT*(Y(4)+SUM2+Y(6))
      Z(1)=0.
      AUX=Y(3)+Y(3)
      AUX=AUX+AUX
      Z(2)=SUM2-HT*(Y(2)+AUX+Y(4))
      Z(3)=SUM1
      Z(4)=SUM2
      IF(NDIM-6)5,5,2
C
C     INTEGRATION LOOP
```

```
    2 DO 4 I=7,NDIM,2
      SUM1=AUX1
      SUM2=AUX2
      AUX1=Y(I-1)+Y(I-1)
      AUX1=AUX1+AUX1
      AUX1=SUM1+HT*(Y(I-2)+AUX1+Y(I))
      Z(I-2)=SUM1
      IF(I-NDIM)3,6,6
    3 AUX2=Y(I)+Y(I)
      AUX2=AUX2+AUX2
      AUX2=SUM2+HT*(Y(I-1)+AUX2+Y(I+1))
    4 Z(I-1)=SUM2
    5 Z(NDIM-1)=AUX1
      Z(NDIM)=AUX2
      RETURN
    6 Z(NDIM-1)=SUM2
      Z(NDIM)=AUX1
      RETURN
C     END OF INTEGRATION LOOP
C
    7 IF(NDIM-3)12,11,8
C
C     NDIM IS EQUAL TO 4 OR 5
    8 SUM2=1.125*HT*(Y(1)+Y(2)+Y(2)+Y(2)+Y(3)+Y(3)+Y(3)+Y(4))
      SUM1=Y(2)+Y(2)
      SUM1=SUM1+SUM1
      SUM1=HT*(Y(1)+SUM1+Y(3))
      Z(1)=0.
      AUX1=Y(3)+Y(3)
      AUX1=AUX1+AUX1
      Z(2)=SUM2-HT*(Y(2)+AUX1+Y(4))
      IF(NDIM-5)10,9,9
    9 AUX1=Y(4)+Y(4)
      AUX1=AUX1+AUX1
      Z(5)=SUM1+HT*(Y(3)+AUX1+Y(5))
   10 Z(3)=SUM1
      Z(4)=SUM2
      RETURN
C
C     NDIM IS EQUAL TO 3
   11 SUM1=HT*(1.25*Y(1)+Y(2)+Y(2)-.25*Y(3))
      SUM2=Y(2)+Y(2)
      SUM2=SUM2+SUM2
      Z(3)=HT*(Y(1)+SUM2+Y(3))
      Z(1)=0.
      Z(2)=SUM1
   12 RETURN
      END
      FUNCTION FC(A)
      COMMON/POI/ CA1,NC,NR
      IF(A.LT.CA1) FC=0.
      IF(A.GE.CA1) FC=1.
      RETURN
      END
//GO.SYSIN DD *
123.0
-123.0
/*
```


☆ U. S. GOVERNMENT PRINTING OFFICE: 1976 — 657-630/1093