USER'S MANUAL

FATIGUE LIFE ANALYSIS PROGRAM (FLAP)

A. M. Zarembski
V. Sobash

R273
Fatigue

MATHEMATICAL MODEL
MATHEMATICAL MODEL

AAR
02-Track-Train Dynamics

rnational Government-Industry
Research Program on Track Train Dynamics

R-273

1. REPORT NO.	2. REPORT DATE	3. PERIOD COVERED
R-273	July 1979	

4. TITLE AND SUBTITLE

User's Guide to Fatigue Life Analysis Program (FLAP)

5. AUTHOR(S)

A.M.Zarembski, V. Sobash

6. PERFORMING ORGANIZATION NAME AND ADDRESS

Association of American Railroads
Track-Train Dynamics Phase II
3140 South Federal Street,
Chicago, Illinois 60616

7. TYPE OF REPORT

Computer program documentation

8. CONTRACT OR GRANT NO.

DOT-FR 64228

9. SPONSORING AGENCY NAME AND ADDRESS

Department of Transportation
Federal Railroad Administration
Washington, D.C. 20590

10. NO. OF PAGES

90

11. NO. OF REFERENCES

12. SUPPLEMENTARY NOTES

13. ABSTRACT

This guide documents the use of the Fatigue Life Analysis Program which is intended for use in the implementation of the Interim AAR Guidelines for Fatigue Analysis of Freight Cars. This computer program is a tool by means of which the freight car design engineer can readily incorporate fatigue life calculations within the design process.

14. SUBJECT TERMS

Freight Car Design
Fatigue Analysis
Fatigue Design

15. AVAILABILITY STATEMENT

Director, Technical Center
Association of American Railroads
3140 South Federal Street,
Chicago, Illinois 60616

Steering Committee for the Track Train Dynamics Program

Chairman
J. L. Cann
Vice President
Operations
CN Rail

Vice Chairman
W. J. Harris, Jr.
Vice President
Research and Test Department
Association of American Railroads

E. F. Lind
Technical Director
Tech Center
Association of American Railroads

W. S. Autrey
Chief Engineer
Atchison, Topeka & Santa Fe Railway Co.

Peter Detmold
Chairman
Canadian Railway Advisory Committee

M. Ephraim
Chief Engineer
Electro Motive Division
General Motors Corporation

J. G. German
Vice President
Engineering
Missouri Pacific Co.

W. S. Hansen
President
A. Stucki Co.

S. B. Hobbs
Deputy Director
Systems Development
Department of Transportation
Transportation Systems Center

W. P. Manos
General Superintendent—Cars
Chicago, Rock Island &
Pacific Railroad Company

G. C. Martin
Director
Mechanical Engineering
ConRail

E. R. Mathews
Director
Transportation Test Center
Federal Railroad Administration

R. A. Matthews
Vice President
Railway Progress Institute

W. McLaren
Chief Current Technology Division
Transport Canada-
Research & Development Centre

D. K. McNear
President
Southern Pacific Transportation Co.

L. A. Peterson
Director
Office of Rail Safety Research
Federal Railroad Administration

G. E. Reed
Director
Railroad Sales
AMCAR Division
ACF Industries

R. E. Rinehart
Manager, Systems Analysis
General Electric Company

D. V. Sartore
Chief Engineer Design
Burlington Northern, Inc.

J. J. Schmidt
Assistant Vice President
Equipment Engineering
National Railroad Passenger Corp

P. S. Settle
Vice-President Commercial
Transportation Product Group
Portec, Inc.

W. W. Simpson
Vice President
Engineering
Southern Railway Company

W. K. Smith
Acting Chairman,
United States Railway Association

D. L. Spanton
Director
Office of Freight Systems
Federal Railroad Administration

D. R. Sutliff
Chief Engineer
Engineering Analysis
ACF Industries, AMCAR Division

***R. D. Spence** (Chairman)
President
Louisville & Nashville Railroad Company

***L. S. Crane** (Chairman)
President and Chief
Executive Officer
Southern Railway System

***D. Y. Clem**
President
McConway & Torley Corporation

***C. Bruce Ward**
President
Gunderson, Inc

***Edward J. Ward**
Senior Railroad Transport Specialist
Transportation Research Board
National Research Council

***J. B. Stauffer**
Director—Industrial Engineering
ConRail

***R. G. Maughan**
Former Chairman
Railroad Advisory Committee
Transportation Development Agency

***Former members of this committee**

ABSTRACT

This guide documents the use of the Fatigue Life Analysis Program, which is intended for use in the implementation of the Interim AAR Guidelines For Fatigue Analysis of Freight Cars. This computer program is a tool, through which the freight car design engineer can readily incorporate fatigue life calculations within the design process.

BACKGROUND INFORMATION

ON THE

TRACK-TRAIN DYNAMICS PROGRAM

The Track-Train Dynamics Program encompasses studies of the dynamic interaction of a train consist with track as affected by operating practices, terrain, and climatic conditions.

Trains cannot move without these dynamic interactions. Such interactions, however, frequently manifest themselves in ways climaxing in undesirable and costly results. While often differing and sometimes necessarily so, previous efforts to reasonably control these dynamic interactions have been reflected in the operating practices of each railroad and in the design and maintenance specifications for track and equipment.

Although the matter of track-train dynamics is by no means a new phenomenon, the increase in train lengths, car sizes, and loadings has emphasized the need to reduce wherever possible excessive dynamic train action. This, in turn, requires a greater effort to achieve more control over the stability of the train as speeds have increased and railroad operations become more systematized.

The Track-Train Dynamics Program is representative of many new programs in which the railroad industry is pooling its resources for joint study and action.

A major planning effort on track-train dynamics was initiated in July 1971 by the Southern Pacific Transportation Company under contract to the AAR and carried out with AAR staff support. Completed in early 1972, this plan clearly indicated that no individual railroad had both the resources and the incentive to undertake the entire program. Therefore, AAR was authorized by its Board to proceed with the Track-Train Dynamics Program.

In the same general period, the FRA signaled its interest in vehicle dynamics by development of plans for a major test facility. The design of a track loop for train dynamic testing and the support of related research programs were also pursued by FRA.

In organizing the effort, it was recognized that a substantial body of information and competence on this program resided in the railroad supply industry and that significant technical and financial resources were available in government.

Through the Railroad Progress Institute, the supply industry coordinated its support for this program and has made available men, equipment, data from earlier proprietary studies, and monetary contributions.

Through the FRA, contractor personnel and direct financial resources have been made available.

Through the Transport Canada Research and Development Centre (TDC), the Canadian Government has made a major commitment to work on this problem and to coordinate that work with the United States' effort.

Through the Office de Recherces et D'Essais, the research arm of the Union Internationale des Chemins de Fer, the basis for a full exchange of information with European groups active in this field has been arranged.

The Track-Train Dynamics Program is managed by the Research and Test Department of the Association of American Railroads under the direction of an industry-government steering committee. Railroad members are designated by elected members of the AAR's Operation-Transportation General Committee, supply industry members by the Federal Railroad Administration, and Canadian Government members by the Transport Development Centre. Appropriate task forces and advisory groups are established by the Steering Committee on an ad hoc basis as necessary to pursue and resolve elements of the program.

The staff of the program comprises AAR employees, personnel contributed on a full- or part-time basis by railroads or members of the supply industry, and personnel under contract to the Federal Railroad Administration or the Transportation Development Agency.

The program plan as presented in 1972 comprises:

1) Phase I -- 1972-1974

   Analysis of an interim action regarding the present dynamic aspects of track, equipment, and operations to reduce excessive train action.

2) Phase II -- 1974-1977

   Development of improved track and equipment specifications and operating practices to increase dynamic stability.

3) Phase III -- 1977-1982

   Application of more advanced scientific principles to railroad track, equipment, and operations to improve dynamic stability.

Phase I officially ended in December of 1974. The major technical elements of Phase I included:

a) The establishment of the dynamic characteristics of track and equipment.

b) The development and validation of mathematical models to permit the rapid analysis of the effects on dynamic stability of modifications in design, maintenance, and use of equipment and track structures.

c) The development of interim guidelines for train handling, makeup, track structures, and engineer training to reduce excessive train action.

Reports on all elements of Phase I activities have been completed and are available through the AAR. A list of the Track Train Dynamics publications is available upon request.

The major technical elements of Phase II include:

a) The adaptation of Phase I analytical models to allow for conducting parameter investigations in the area of track, trucks, draft gear and cushion units, and vehicle behavior.

b) The development of fatigue analysis guidelines.

c) The development of a comprehensive program for identifying the loads to which track, vehicles, and vehicle components are subjected.

As research on this program proceeds, reports on other elements of Phase II will be issued, and existing reports updated at appropriate intervals.

TABLE OF CONTENTS

		Page
1.	Introduction	1
	1.1 Purpose and Scope	1
	1.2 Related Publications	1
2.	Analytical Background	2
	2.1 General	2
	2.2 Miner's Rule	3
	2.3 Material Properties	3
	2.4 Load Spectra	4
	2.5 Data Acquisition and Reduction	5
3.	Program Description	9
	3.1 Introduction	9
	3.2 Main Program	10
	3.3 Subroutines	14
	3.3.1 Loaded	14
	3.3.2 Empty	16
	3.3.3 Impact	16
	3.3.4 CONVRT	20
	3.3.5 GMULTY	21
	3.3.6 ADDNT	25
4.	Program Use	29
	4.1 Versions	29
	4.2 Input - DEC	29
	4.3 Control Cards - IBM	31
	4.4 Input - IMB	32
	4.5 Output - IBM and DEC	37

TABLE OF CONTENTS (con't)

Page

5. Symbols 38

Appendix A: Sample Problem A.1

A.1 Problem Description A.1

A.2 IBM Version A.4

A.2.1 FLAP Input Description A.4

A.2.2 FLAP Listing - IBM A.9

A.2.3 FLAP Output - IBM A.22

A.3 DEC Version A.28

A.3.1 FLAP Input Description A.28

A.3.2 FLAP Listing - DEC A.31

A.3.3 FLAP Output - DEC A.45

Appendix B: List of Ratio of Empty Car at Loaded Car Miles B.1

LIST OF FIGURES

Figure No.		Page
1	Damage Calculation Flow Chart.	7
2	Fatigue Analysis Program Flow Chart. .	8
3	Main Routine Flow Chart.	12
4	Loaded Subroutine Flow Chart	15
5	Empty Subroutine Flow Chart	17
6	Impact Subroutine Flow Chart	19
7	GMULTY Matrix.	23
8	GMULTY Subroutine Flow Chart	24
9	Complete Flow Chart.	26
10	Car Group Layout	33
11	Data Input When JNT=3.	34
12	Card Format.	35
A.1	Beam Reinforced With Partial Length Cover Plate	A.2
A.2	Sample Problem Card Format	A.7
A.3	Sample Problem Card Format	A.8
A.4	Load Spectra Suitable For Input to FLAP.	A.29
A.5	User's Program Conversion.	A.30

LIST OF TABLES

Table		Page
1	Input Data.	6
2	Options Available in Main Route . .	13
3	Option Control Variables.	27
4	Program Designators for Engineering Values	36
A.1	Road Environment Percent Occurrence Spectrum	A.3

1. INTRODUCTION

The use of fatigue analysis for the calculation of freight car and/or component life is a necessary addition to the overall freight car design procedure. To meet this need, the Interim AAR Guidelines for Fatigue Analysis of Freight Cars was introduced. In order to facilitate the implementation of these guidelines, a user oriented computer program was developed.

1.1 Purpose and Scope

This volume is intended to serve as a user's manual for the application of the freight car Fatigue Life Analysis Program. The program is based on the Interim AAR Guidelines for Fatigue Analysis of Freight Cars. It is not intended to supercede or make obsolete any existing AAR design requirements. Rather, it is to be used as a tool, by means of which the freight car designer can readily incorporate fatigue life calculations within the design process.

1.2 Related Publications

- Interim AAR Guidelines for Fatigue Analysis of Freight Cars
- User's Guide to Rainflow Counting Program
- Specification for the Design, Fabrication, and Construction of Freight Cars. AAR. Adopted September 1964.

2. ANALYTICAL BACKGROUND

The basic fatigue analysis method used in this program corresponds to that described in the Interim AAR Guidelines. As stated in that Guideline, this method is not considered to be the only reliable method of fatigue analysis. However, users of other methods must be able to verify the method's general acceptance in the field of fatigue analysis and provide reference to sufficient comparisons between predictions and test results.

2.1 General

The fatigue analysis portion of this program is based on Miner's linear cumulative damage rule. This rule is used to calculate fatigue damage due to over-the-road and/or yard impact loading. By use of either applicable modified Goodman Diagrams or appropriate stress concentration factors, the effect of geometric details is included.

The basic input data is generally in the form of environmental load spectra, as included in the Interim AAR Guidelines. These spectra can then be converted to stress spectra for critical car locations by means of structural analysis techniques. Alternately, raw data such as that obtained from field measurement on instrumented test cars (strains, accelerations, etc.) can be converted from time history form to suitable load spectra by means of the AAR Rainflow Counting Program. This counting program gives the user the capability of taking raw field test data and converting it into a form suitable for input into the Fatigue Analysis Program.

A summary of the program is shown as a basic fatigue life calculation sequence in Figure 1. Using the input data needed to fill the requirements of Table 1, the fatigue damage is calculated and the total number of cycles to failure is determined. The loaded and unloaded fatigue lives are then determined by using the ratios of occurrences (or cycles) per mile and empty to loaded car-miles (Appendix B). Additional options (Figure 2) allow for the combination of yard-impact life to obtain a total anticipated fatigue life.

It should be noted that conversion of load spectra to stress spectra, as done in this program, is based on a separate vehicle stress analysis which is beyond the scope of this program. Finally, the decision as to which location in the freight car should be investigated for the fatigue life analysis is beyond the capability of this program. This decision, which involves considerable engineering judgement, must be made by the car designer.

2.2 Miner's Rule

Miner's linear cumulative damage rule is a widely accepted rule of fatigue damage accumulation in a structure subject to repeated loads. Consequently, it was chosen as the basis for the standard AAR fatigue analysis guideline. According to this rule, whenever a component is stressed to some value below the yield stress of the material and above the fatigue limit for that detail component (at that applied stress level), damage occurs (Figure 1). For components subject to repeated loadings, this damage is linearly cumulative, and failure will result when the sum of the damage equals unity.

$$\Sigma \frac{N_i'}{N_i} = N_T \Sigma \frac{\alpha i}{Ni} = 1$$

This method of analysis, which is described in detail in the Interim AAR Guidelines, assumes no damage for negative-negative stress cycles, and no dependenace on order of load. For the case where the absolute value of the compressive stress is greater than or equal to the tensile stress for a given cycle, the cycle is assumed to be a complete reversal cycle with a value equal to one-half the sum of the absolute value of the compressive stress and tensile stress. The compressive stress then becomes the negative of the new tensile stress.

2.3 Material Properties

The Modified Goodman Diagrams (MGD's) such as those given in the Interim AAR Guidelines are convenient means of giving basic material property data and have the additional capability of being able to incorporate detailed features.

The MGD's are assumed to have constant slope which are cut off at the yield point of the material. The curves given in the Interim AAR Guidelines and the corresponding analysis assume a fatigue limit for steels of 2,000,000 cycles.

Two basic types of MGD's are included in the AAR Guidelines. One type gives the basic fatigue properties of plates in the as-rolled condition. When using this type of diagram, an appropriate stress concentration factor must be included in the stress analysis to take into consideration the effect of structural details. The second type of diagram already includes the effect of a structural detail and thus, no additional stress concentration factor is needed for the corresponding analysis.

The material properties needed for the Fatigue Life Analysis Program are:

b Fatigue limit for the stress ratio R=0, (y intercept of the MGD curve)

m Slope of the MGD curve

k Absolute value of the slope of the S-N curve for the material under consideration. (The S-N curve is assumed to be a straight line on log-log paper which for steel goes from 100,000 cycles to 2,000,000 cycles. The slope is independent of the stress ratio R).

2.4 Load Spectra

a. Environmental (Over-the-Road) Load Spectra

Several environmental load spectra for various categories of freight cars are given in the Interim AAR Guidelines. These spectra are based on range-mean counting of field test data. Supplementary spectra can be generated through the use of the AAR Rainflow Counting Program to reduce field test data.

The spectra takes the form of three-dimensional folded spectra which give, for each maximum-minimum load combination, the percent occurrence that the cycles are encountered in the environment. It is recommended as a conservative approach that the stress values for each increment be chosen so as to provide the greatest range between maximum and minimum stress.

Conversion of load spectra to stress spectra involves application of stress analysis techniques beyond the scope of this manual. However, after a conversion of load to stress spectra has been effected, it can be incorporated into the Fatigue Life Analysis Program in a manner corresponding to that shown in the Interim AAR Guidelines.

The user should note that conversion of load spectra to stresses may involve more than one load spectra (combined loadings). Furthermore, stress spectra may differ for a freight car subject to a given environmental load condition when it is in a loaded vs. an unloaded state.

b. Impact (Yard) Load Spectra

At present there is little data available on impact loading in the form of load spectra. This area, which is not covered in the Interim AAR Guidelines, is included in the Fatigue Life Analysis Program so that impact effects, when data becomes available, can be readily incorporated into the design procedure. It has been observed that impact loading will generally be in the form of zero to maximum loads, and consequently will be represented by only one column of the three dimensional folded spectra.

The conversion of yard damage to equivalent car life involves the development of a β_i, the equivalent ratio of impact cycles per mile. One suggested procedure is to establish a ratio of service miles per impact, β_M. (1 impact every 200 road miles is a number that has been suggested, but not validated). Then $\beta_i = \beta/\beta_M$, where β is the total number of spectrum cycles per mile for the road environment under consideration. This method, which appears to be acceptable, has not been validated at this time, and consequently, the choice of a value for β_i is left to the individual car designer.

2.5 Data Acquisition and Reduction

As an alternative to the use of the given car environmental load spectra, new load or stress spectra can be generated directly from test measurement data. This can be done through instrumentation of a "test" vehicle for over-the-road measurements of either acceleration or stress time histories. These time histories can be reduced to the form of three dimensional folded spectra by means of the AAR Rainflow Counting Program which is described in the User's Guide to the AAR Rainflow Counting Program.

It should be noted that development of an over-the-road stress spectra, at a point already determined as being fatigue critical, eliminates the need for a stress analysis to convert the load spectra to stress spectra, since the stress spectra is obtained directly. However, care should be used to insure that the instrumented point is, in fact, the area where fatigue failures will arise and be of a form so as to put the car out of service. Furthermore, the load distribution on the test car should be representative of the actual service environment. A list of freight car components and attachments which require fatigue analysis is given in the Interim AAR Guidelines.

TABLE I

INPUT DATA

1. Option Control Data

2. Material and Detail Part Properties
   a. Material Properties
   b. Detail Part Properties
      1. Modified Goodman Diagram
      2. S - N curve

3. Ratio of Empty Car to Loaded Car - Miles

4. Ratio of Impact Occurrences to Road Miles

5. Conversion of Load Spectra to Stress Spectra

6. Load Environment Spectra
   a. Road
      1. Loaded
      2. Unloaded
   b. Impact (Yard)

Figure 1

Damage Calculation Flow Chart

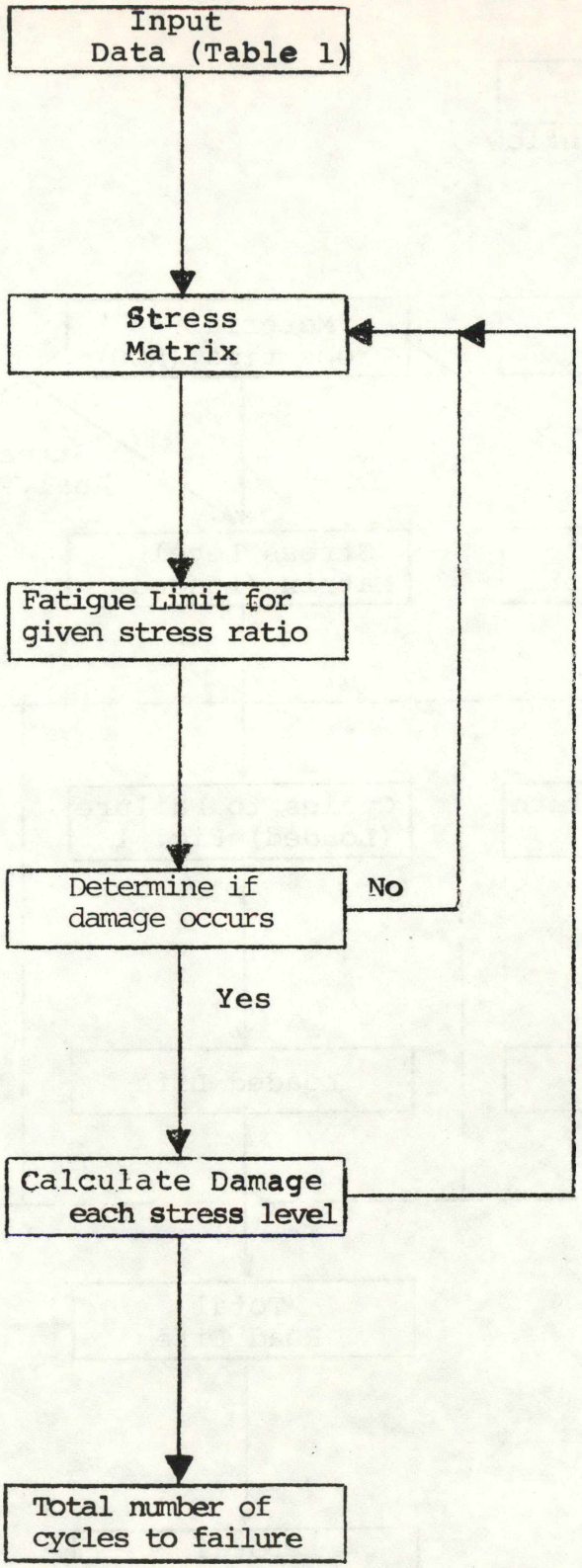
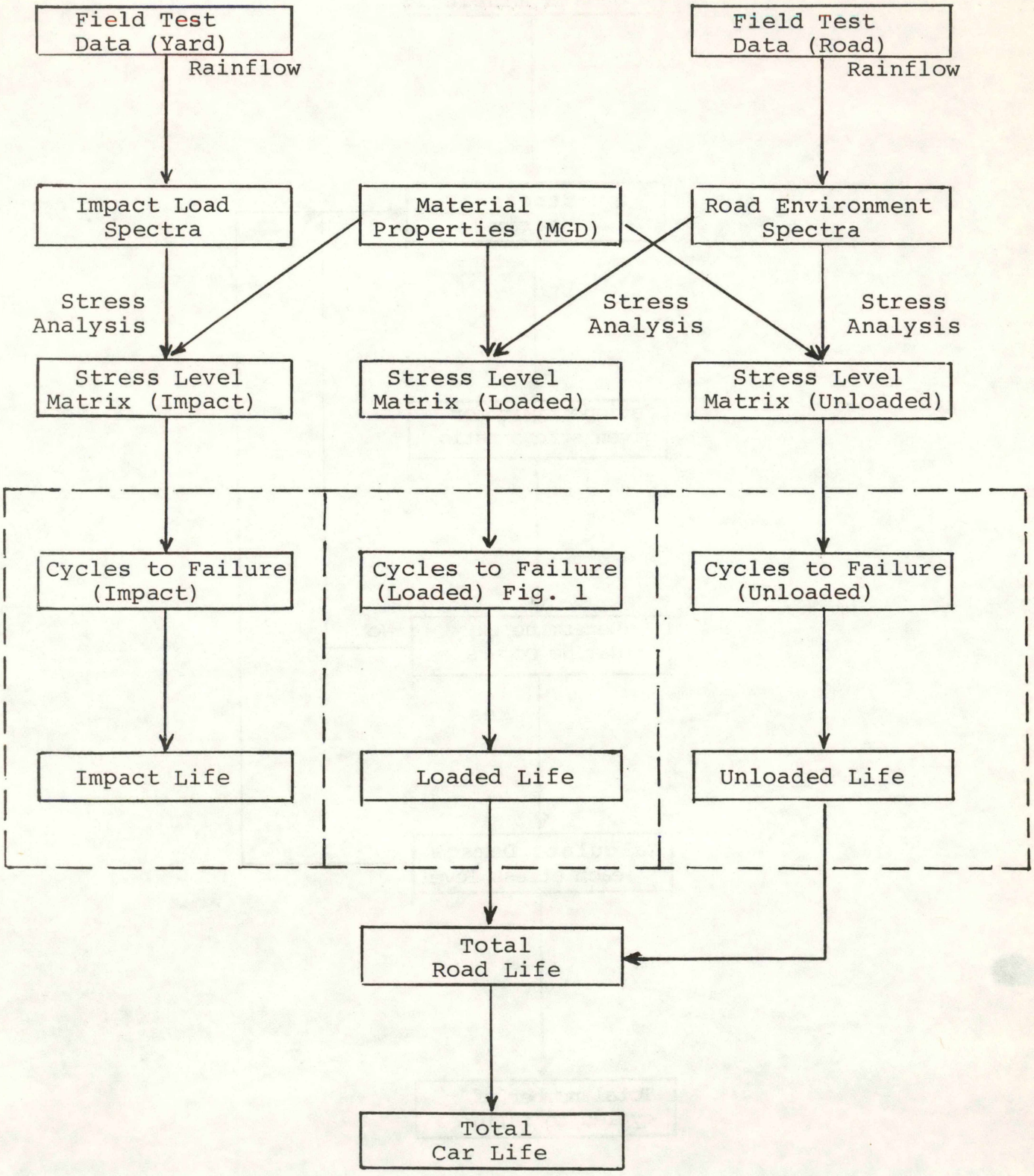


Figure 2

FATIGUE ANALYSIS PROGRAM FLOW CHART



3. PROGRAM DESCRIPTION

3.1 Introduction

FLAP consists of three subroutines which perform all fatigue life calculations. These routines are used by the main program and have no dependence upon each other. Three service subroutines are used to augment the main program and its routines in data assembly.

The three core subroutines are:

1. LOADED, which calculates fatigue life based on loaded road use of the car, is employed in every run.

2. EMPTY, which calculates fatigue life based on unloaded road use of the car, is called only when the user feels damage may have resulted from unloaded road use.

3. IMPACT, which calculates the fatigue life associated with damage caused in yard operations, is implemented only when knowledge of damage on this component is available.

The three service subroutines are:

1. GMULTY, which assembles the necessary matrices from input information, is only called when a direct input of the stress matrices is not available or convenient.

2. ADDNT combines multiple load spectra in any or all of the routines and is called only when more than one type of input loading is considered.

3. CONVRT* implements a conversion of a load spectra to form that is appropriate for input into the program.

The main program serves as a driver through which the desired subroutines are called, and through which the input and output functions are handled.

* DEC System-20 version only

3.2 Main

The main routine of the FLAP program can be broken up into four specialized functions:

1) Input and printout of data (Types 1-5, Table 1)

2) Accessing of needed subroutines

3) Conversion of fatigue damage cycles to expected life values in miles

4) Combining of the various contributing fatigue damages to obtain a value for the total fatigue life.

These functions are related to each other in the program as shown on the flow chart (Figure 3).

Main has been written with the intention of direct user-program communication and this dialogue should be system dependent. If direct user-program communication is not desired or cannot be implemented, this block must be structured toward a different format*.

The first function, the input of data, utilizes a block of read and write statements at the beginning of the program. A call to the service routine CONVRT*, can be used or bypassed upon decision of the user. A detailed explanation of the required input formats and the output format is given in Chapter 4.

The second function, providing access to the other five subroutines available, utilizes the option control data, which the user must input. To simplify the discussion at this point, the ADDNT subroutine will not be included in this explanation. The accessing function is accomplished by a series of tests acting on the option control variables which the user inputs in order to specify which subroutines are desired. The options available to the user are shown in Table 2.

After the fatigue damage cycles have been determined for loaded and unloaded road usage and yard operations in the respective subroutines, the main program then performs its third function, utilizing the variable Beta (β) to convert the fatigue damage cycles into expected life in miles. This value of Beta (cycles/mile) must be supplied by the user and cannot be generated in the program. The value is determined from the data acquisition procedure and accompanies the load spectra that is input to the program. Beta is number of load cycles per mile of vehicle travel. Thus, life (in miles) equals $N_{T/\beta}$.

* DEC System-20 version only

Each type of usage, i.e., loaded road service, unloaded road service, and yard operations (impacts), causes its own fatigue damage and thus requires its own Beta value. The Beta value for loaded and unloaded usage are generated from the basic value obtained in the data acquisition and are related to the empty car/loaded car usage ratios. Values for this ratio are provided in Appendix B. The user is required to input the basic Beta value and the empty car/loaded car miles ratio value, and from these values the program will solve for the resultant Beta Loaded and Beta Unloaded values. These are then used to divide the loaded fatigue damage cycles to failure and the unloaded fatigue damage cycles to failure, respectively, to obtain the number of miles of service which can be expected in each case. A Beta value for impact must also be determined and directly input by the user. This is discussed together with a suggested method in Section 2.4.b.

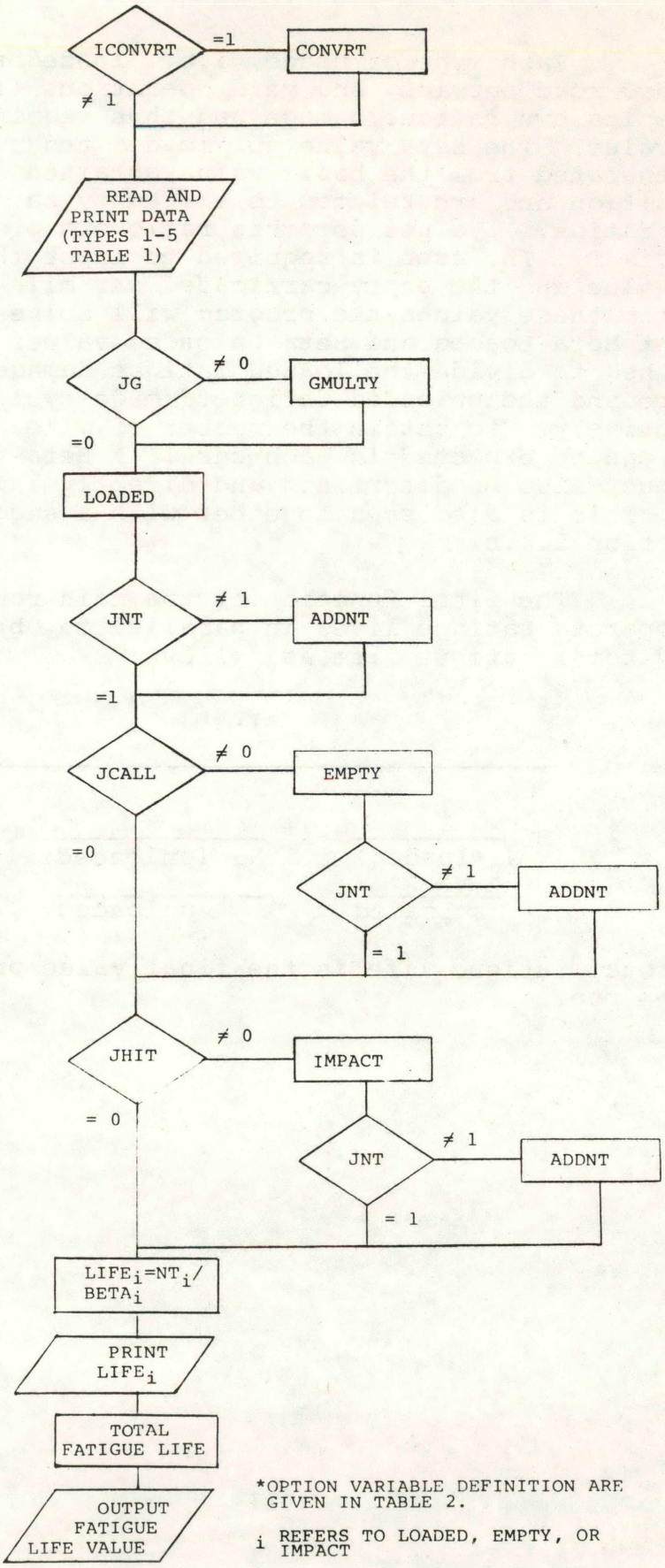
The final function of the main routine adds the separate fatigue lives in parallel to obtain the expected total fatigue life as follows:

$$\text{Life} = \frac{1}{\dfrac{1}{\dfrac{N_T(\text{loaded})}{\beta\ \text{loaded}}} + \dfrac{1}{\dfrac{N_T\ (\text{unloaded})}{\beta\ \text{unloaded}}} + \dfrac{1}{\dfrac{N_T\ (\text{impact})}{\beta\ \text{impact}}}}$$

This total fatigue life is the final value printed out for the run.

Figure 3

MAIN ROUTINE FLOW CHART



*OPTION VARIABLE DEFINITION ARE GIVEN IN TABLE 2.

i REFERS TO LOADED, EMPTY, OR IMPACT

TABLE 2

OPTIONS AVAILABLE IN THE MAIN ROUTINE

ICNVRT*	-	Option Control for Entering Load Spectra:
		ICNVRT = 1, reformat load spectra in a form suitable to FLAP
		ICNVRT = 0, load spectra is already in form suitable for input to FLAP
JG	-	Option Control for Entering GMULTY Subroutine:
		JG = 1, control is transferred to GMULTY
		JG = 0, GMULTY not utilized
JCALL	-	Option Control for Entering EMPTY Subroutine:
		JCALL = 1, fatigue damage due to unloaded road usage is calculated
		JCALL = 0, EMPTY not utilized
JHIT	-	Option Control for Entering IMPACT Subroutine:
		JHIT = 1, fatigue due to impact/yard conditions is calculated
		JHIT = 0, impact not utilized
JNT	-	Option Control for Entering ADDNT Subroutine:
		JNT > 1, multiple load data combined
		JNT = 1 ADDNT not utilized

* DEC SYSTEM-20 version only

3.3 Subroutines

In addition to the MAIN routine, six subroutines combine to provide a versatile means of fatigue life calculation. The user is given a variety of options, which he may or may not choose to employ.

Three of the subroutines, LOADED, EMPTY, and IMPACT form the core of the program. These subroutines determine the fatigue damage cycles to failure due to each stress spectra.

The other three subroutines, CONVRT, GMULTY, and ADDNT are service subroutines, which are available if needed to organize the matrices required in the core subroutines. Within GMULTY and ADDNT several options are available for generation of the stress matrices other than direct input. CONVRT is used for reformatting the load spectra to be input to the program.

A great deal of input preparation is saved when one of these options can be utilized.

3.3.1 Loaded

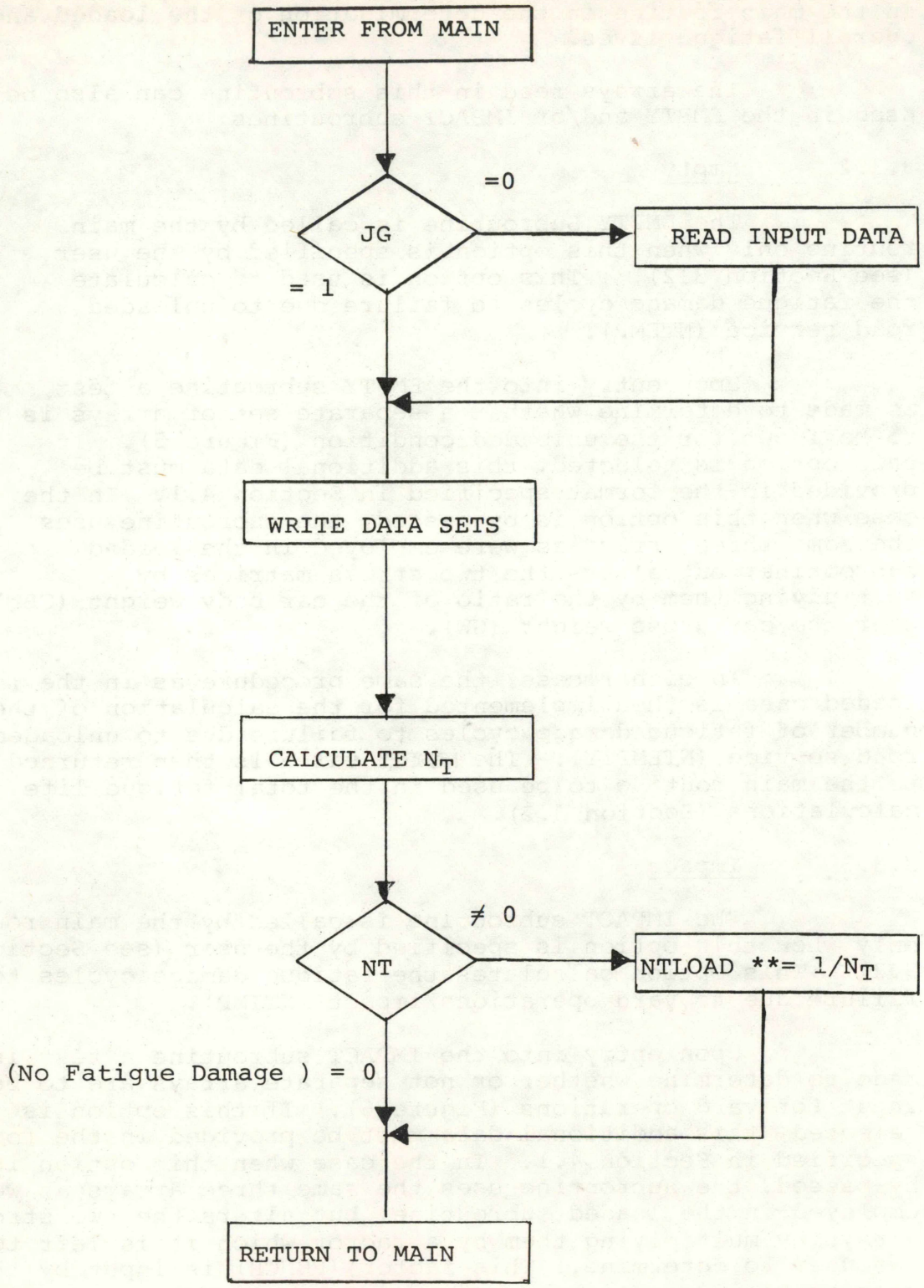
The LOADED subroutine is the only one of the six subroutines which is called no matter what options are chosen by the user. Upon entry into LOADED, a test to determine if data is being transferred from the GMULTY subroutine is made; otherwise the necessary arrays are read in directly (Figure 4).

The three arrays which are used to calculate the number of fatigue damage cycles to failure are a maximum stress array (STRSMX), a minimum stress array (STRSMN), and a percent occurrence array (ALPHA). These three arrays are printed out by LOADED before the calculation sequence is begun. These arrays are basic to the program and represent the stress spectra as defined in the user's guide. In all three arrays the values must be compatible. Thus STRSMX requires a positive stress value, STRSMN requires a stress value less than STRSMX, and ALPHA must have a value less than one, representing the percent occurrence of this particular stress combination in the total stress spectra. The format for these arrays is given in Section 4.1.

These arrays are used to determine a value for the number of fatigue damage cycles to failure due to loaded road service (NTLOAD). The procedure used in this calculation is discussed in Section 2.2 and a graphic representation is provided in Figure 1.

Figure 4

LOADED SUBROUTINE FLOW CHART*



*Option control variables defined in Table 3.

**Number of fatigue damage cycles due to loaded condition use.

The NTLOAD value and all three arrays are then returned to the main routine. The NTLOAD values are used in the main routine in the determination of the loaded and overall fatigue lives.

The arrays read in this subroutine can also be used in the EMPTY and/or IMPACT subroutines.

3.3.2 Empty

The EMPTY Subroutine is called by the main routine only when this option is specified by the user (see Section 3.2). This option is used to calculate the fatigue damage cycles to failure due to unloaded road service (NTEMP).

Upon entry into the EMPTY subroutine a test is made to determine whether a separate set of arrays is to be input for the unloaded condition (Figure 5). If this option is selected, this additional data must be provided in the format specified in Section 4.1. In the case when this option is by-passed, the subroutine uses the same three arrays as were employed in the loaded subroutine, but alters the two stress matrices by multiplying them by the ratio of the car body weight (CBW) over the car gross weight (GW).

In either case, the same procedure as in the loaded case is then implemented for the calculation of the number of fatigue damage cycles to failure due to unloaded road service (NTEMPTY). The NEMPTY value is then returned to the main routine to be used in the total fatigue life calculations (Section 3.2).

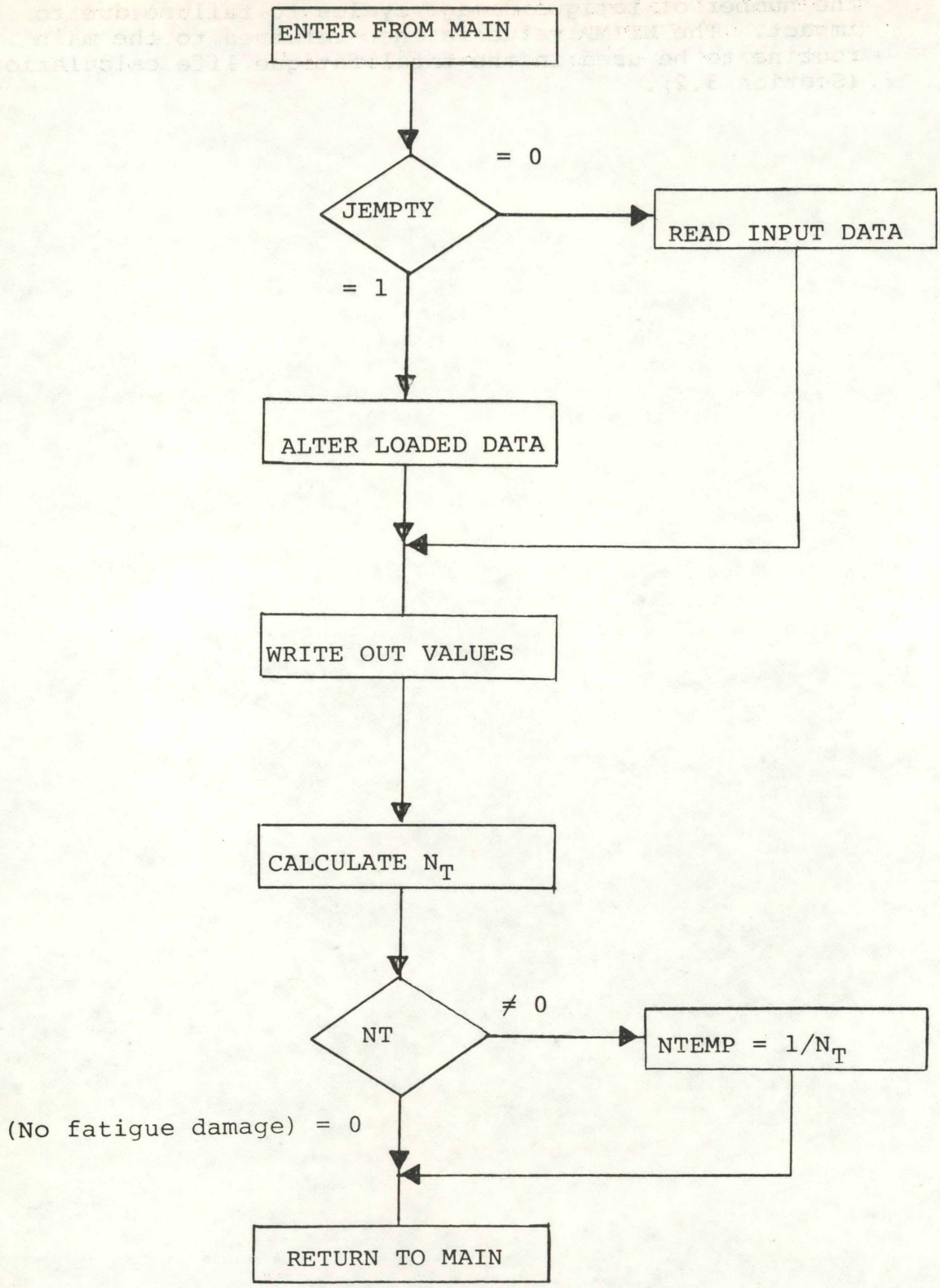
3.3.3 Impact

The IMPACT subroutine is called by the main routine only when this option is specified by the user (see Section 3.2). This option calculates the fatigue damage cycles to failure due to yard operations/impact (NTIMP).

Upon entry into the IMPACT subroutine a test is made to determine whether or not separate arrays are to be input for yard operations (Figure 6). If this option is selected, this additional data must be provided in the format specified in Section 4.1. In the case when this option is by-passed, the subroutine uses the same three arrays as were employed in the loaded subroutine, but alters the two stress arrays by multiplying them by a factor which it is left to the user to determine. This factor (PUNCH) is input by the user at his discretion.

Figure 5

EMPTY SUBROUTINE FLOW CHART *

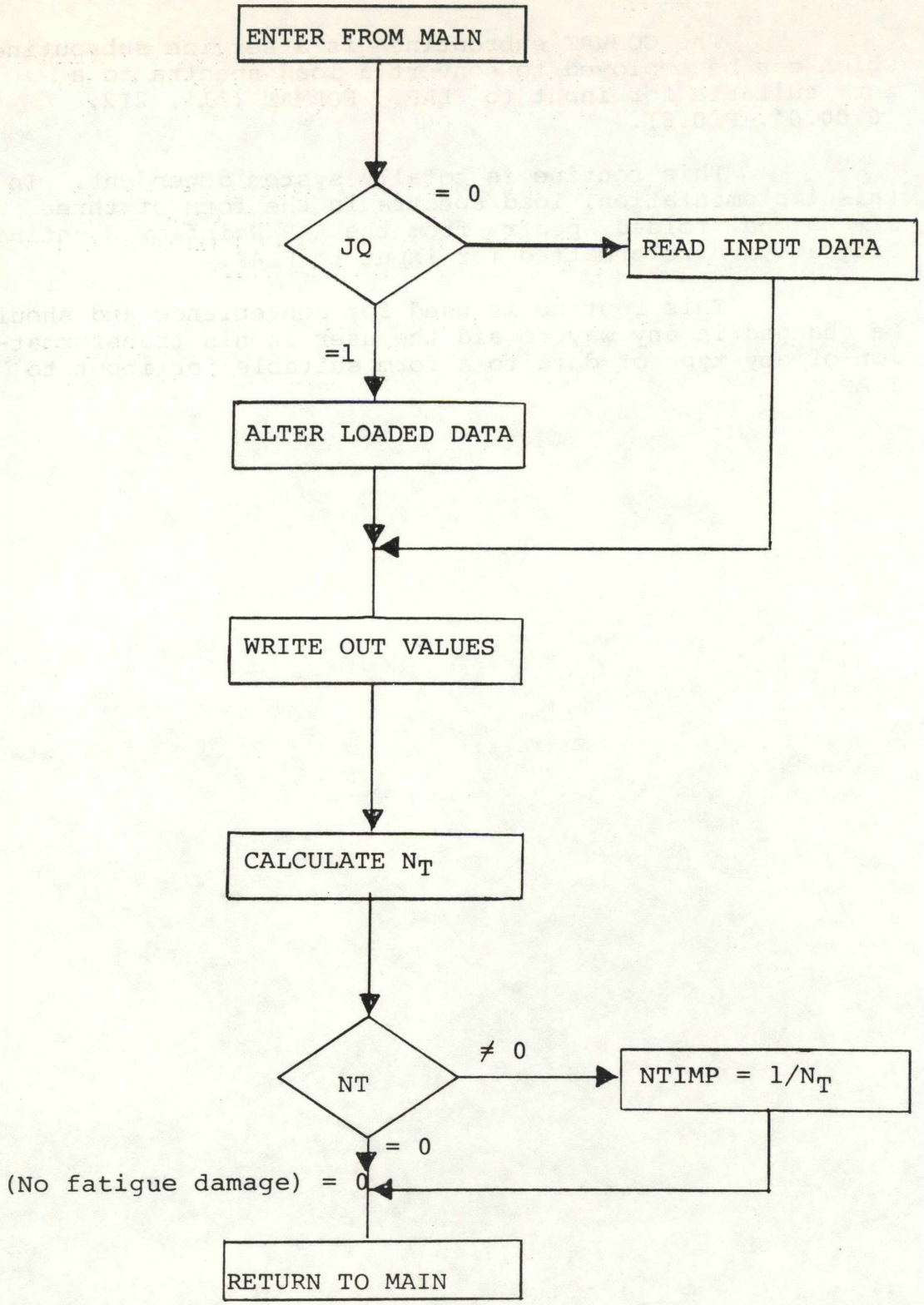


* Option control variables defined in Table 3.

In either case, the same procedure as in the loaded case is then implemented for the calculation of the number of fatigue damage cycles to failure due to impact. The NTIMP value is then returned to the main routine to be used in the total fatigue life calculations (Section 3.2).

Figure 6

IMPACT SUBROUTINE FLOW CHART *



* Option control variables defined in Table 3.

3.3.4 CONVRT

The CONVRT subroutine* is a service subroutine which can be employed to convert a load spectra to a form suitable for input to FLAP. FORMAT ('1', 2I2, '0.00.0', F10.8).

This routine is totally system dependent. In this implementation, load spectra in the form of three dimensional folded spectra from the AAR Rainflow Counting Program, are reformatted for input to FLAP.

This routine is used for convenience and should be changed in any way to aid the user in his transformation of any type of data to a form suitable for input to FLAP.

* CONVRT only available on DEC-20 version

3.3.5 GMULTY

The GMULTY subroutine is a service subroutine which can be employed to produce the two stress arrays STRSMX and STRSMN required in the core subroutine (Figure 7). The user must input all three arrays, STRSMX, STRSMN, and ALPHA if the GMULTY subroutine is by-passed. When used GMULTY requires the input of several stress/load factors in addition to the ALPHA array. A complete ALPHA array is required irrespective of the selected options.

Within the GMULTY subroutine, there are three options available (Figure 8):

1) use of an incrementing factor (G) together with a static stress (STATIC) to generate the stress arrays that correspond to the input ALPHA arrays,

2) use of a stress concentration factor (SCF) and a stress/force ratio (STRE/FOE) with a static stress (STATIC) to convert input load arrays into usable stress arrays,

3) use of a stress/force ratio (STRE/FOE) to convert input load arrays into usable stress arrays.

Note 1: Only one option can be employed in each run of the program. They cannot be used together.

Note 2: Option 3 is a special case of Option 2, where SCF = 1.0 and STATIC = 0.0.

The subroutine begins by determining which of the three options has been chosen. The alpha matrix must be input by the user in all three options.

Selection of the first option requires the user to input a static stress (STRESS) and an incrementing acceleration value (G). These values are used to generate the STRSMX and STRSMN arrays. Note the dynamic stresses are obtained by multiplying G and the static stress and are then added to the static stress to generate the final stress arrays (Figure 8). These data sets are now in a form acceptable for use in the core subroutines without the need for a direct input of the stress values.

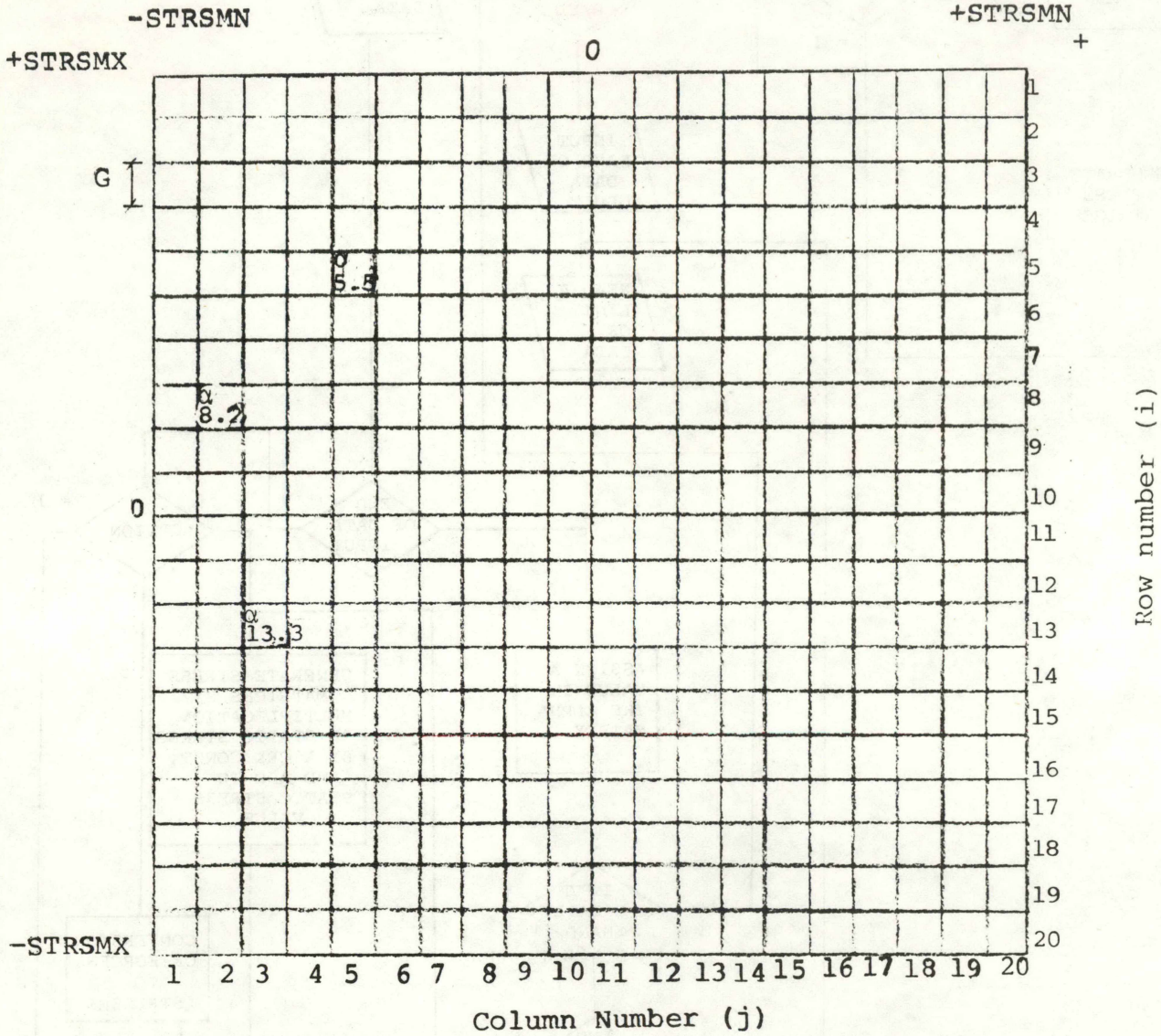
The second option requires the user to input two load arrays in addition to the ALPHA array and to input a stress concentration factor (SCF) which must be selected by the user along with an appropriate stress/force ratio (STRE/FOE) and a static stress (STATIC). (This static stress will often be zero and is assigned a default value of zero unless some other value is input). In this option, the two load arrays corresponding to STRSMX and STRSMN are multiplied by the stress/force ratio to convert the loads into stresses. In the same step, this value is modified by multiplication with the stress concentration factor. The result of this operation is added to the static stress value, if one is input, to obtain the necessary STRSMX and STRSMN arrays.

The third option provided in GMULTY converts load arrays directly into stress arrays. The user must input the load and ALPHA arrays and an appropriate stress (STRE) and force (FOE). The subroutine then multiplies each value in the load sets by the stress/force ratio to obtain STRSMN and STRSMN directly.

The three arrays are returned to the main routine to be used when the LOADED subroutine is called.

Figure 7

GMULTY MATRIX



The dynamic STRSMX and STRSMN values for a given $\alpha_{i,j}$ are equal to: G STATIC $(i_{zero} - i + 1)$ and G·STATIC·$(j-1-i_{zero})$ respectively e.g For the $\alpha_{8,2}$ value:

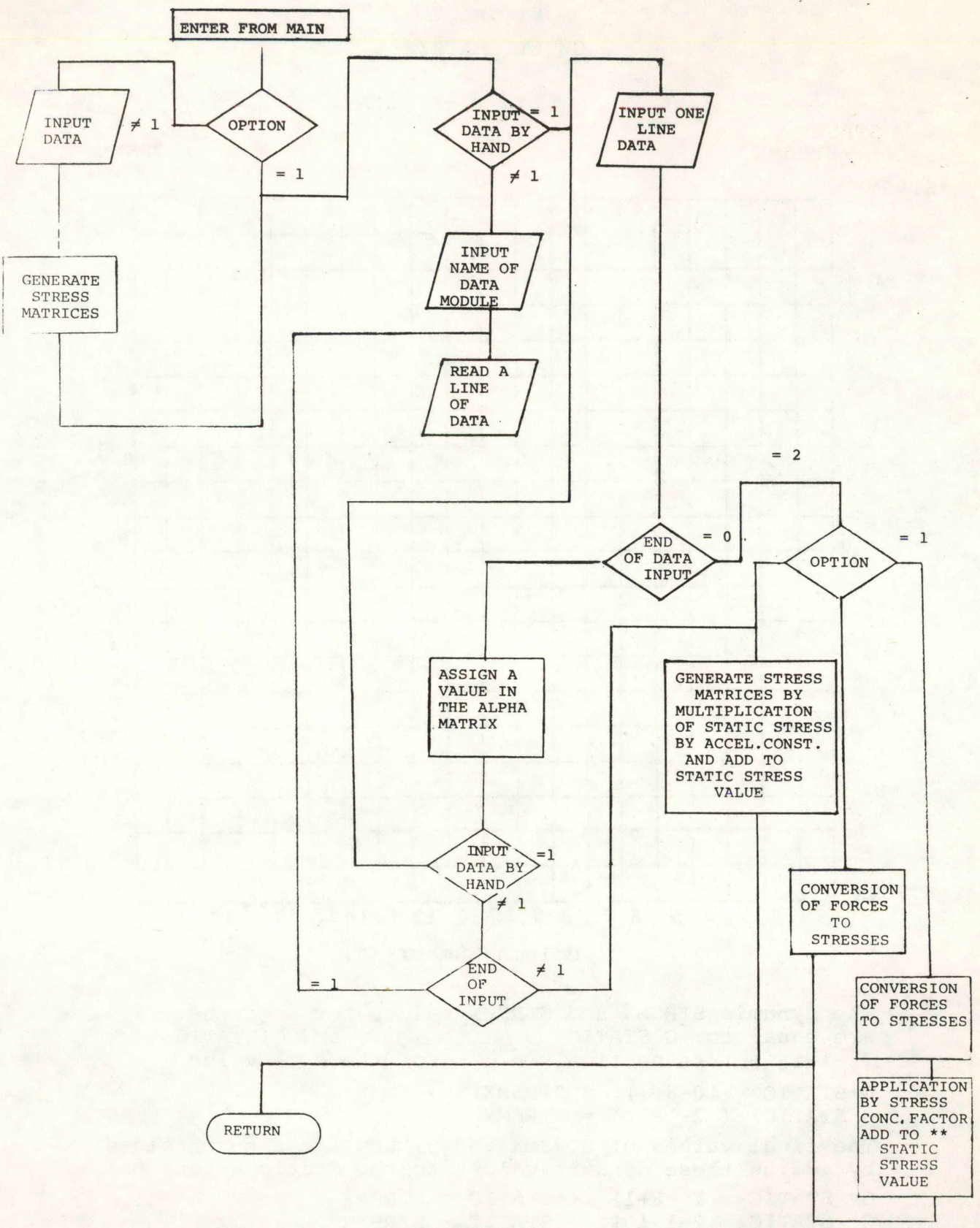
G·STATIC· (10-8+1) = STRSMX
G·STATIC· (2-1-10) = STRSMN

The final values of STRSMX and STRSMN are then obtained by adding these dynamic values to the Static stress value.

G ·STATIC· (10-8+1) + STATIC = STRSMX
G ·STATIC· (2-1-10) + STATIC - STRSMN

Figure 8

GMULTY SUBROUTINE FLOW CHART



3.3.6 ADDNT

The ADDNT subroutine is employed to combine up to three simultaneous fatigue producing spectra. The fatigue damage cycles to failure for each spectra are added in parallel, i.e., Figure 9.

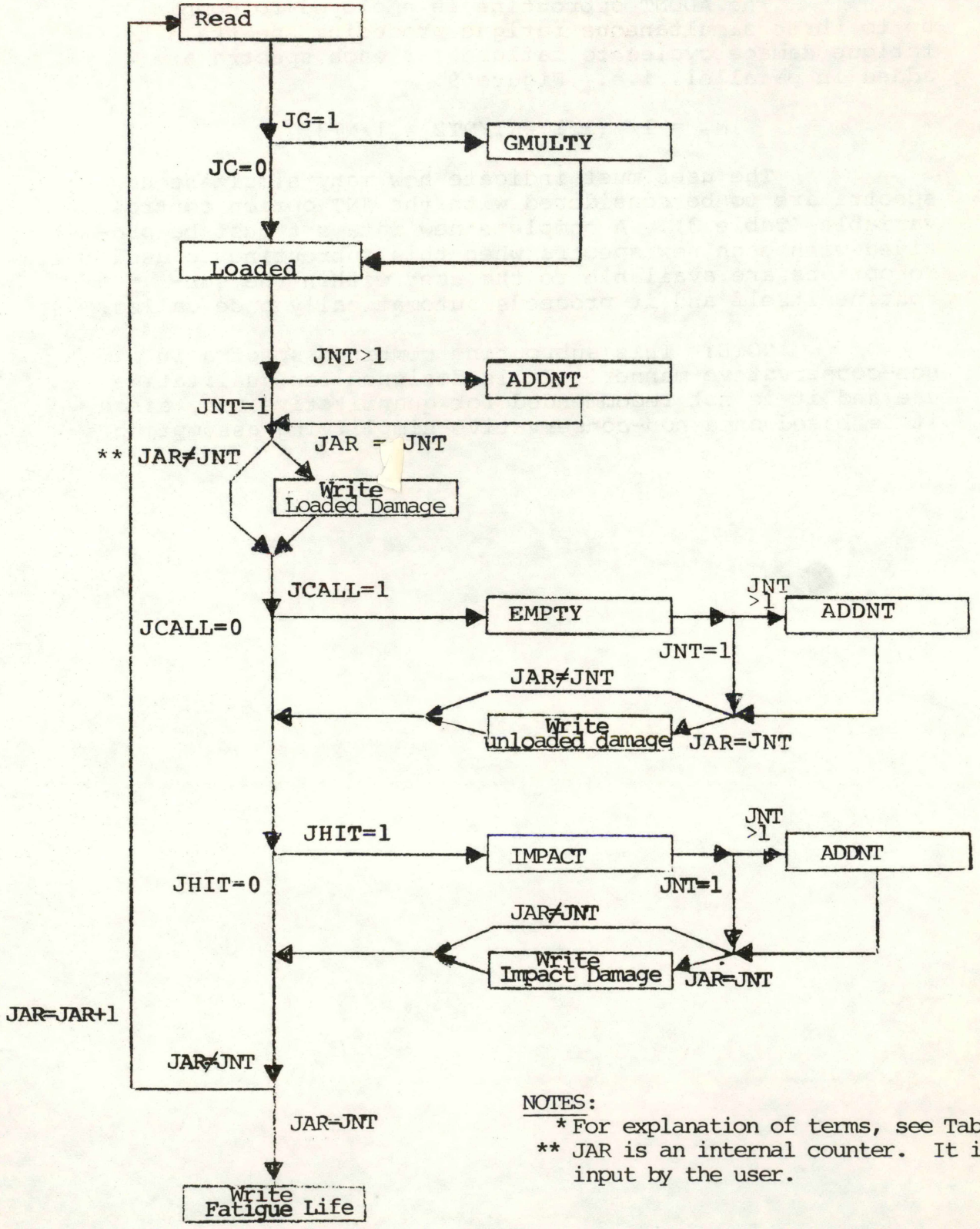
$$N_T = 1/ (1T1 = 1/NT2 = 1/NT3)$$

The user must indicate how many simultaneous spectra are to be considered with the JNT option control variable (Table 3). A complete new data set must be provided with each new spectra when this subroutine is used. No options are available to the user within the subroutine itself and it proceeds automatically once called.

NOTE: This subroutine combines spectra in a non-conservative manner. It is designed for qualitative use and it is not recommended for quantitative use, since it is based on a non-conservative simplifying assumption.

Figure 9

COMPLETE FLOW CHART *



NOTES:

* For explanation of terms, see Table 3.

** JAR is an internal counter. It is not input by the user.

TABLE 3

OPTION CONTROL VARIABLES

JDIM - User specified dimensions of spectra arrays, thus generated arrays are JDIM by JDIM; JDIM $\leq$ 20

JG - Option control for entering GMULTY subroutine:
JG = 1, control is transferred to GMULTY
JG = 0, GMULTY not utilized

JCALL - Option control for entering EMPTY subroutine:
JCALL = 1, fatigue damage due to unloaded road usage is calculated,
JCALL = 0, EMPTY not utilized

JHIT - Option control for entering IMPACT subroutine:
JHIT = 1, fatigue damage due to impact yard conditions is calculated
JHIT = 0, IMPACT not utilized

JNT - Option control for entering ADDNT subroutine:
JNT = 1, ADDNT not utilized,
JNT = 2, two simultaneous sets of input data are used and the ADDNT routine is employed to combine them,
JNT = 3, three simultaneous sets of input data are used and the ADDNT routine is employed to combine them

JOR - Option control in GMULTY subroutine:
JOR = 1, GMULTY utilizes the matrix constant, G (input data), and a static stress value, STATIC, to directly generate the dynamic stress matrices
JOR = 2, GMULTY utilizes a stress concentration factor, SCF (input data), and a stress/load conversion ratio to generate the needed stress matrices from the load matrices
JOR = 3, GMULTY utilizes a stress/load conversion ratio to generate the needed stress matrices from the load matrices

JEMPTY - Option control within the EMPTY subroutine:
JEMPTY = 1, the unloaded stress matrices are obtained by multiplying the loaded stress matrices by the car body weight (CBW) gross

weight (GW) ratio
JEMPTY = 0, the unloaded stress matrices are input directly by the user

NOTE: If JCALL = 0, the value of JEMPTY is never used

JQ - Option control within the IMPACT subroutine:
JQ = 1, the impact stress matrices are obtained by multiplying the loaded stress matrices by the conversion factor, PUNCH
JQ = 0, the impact stress matrices are input directly by the user

NOTE: If JHIT = 0, the value of JQ is never used

4. PROGRAM USE

4.1 Versions

Two versions of FLAP are currently available; an IBM System 370 configuration and a DECSYSTEM-20 configuration. The two versions differ in input/output and user program handling.

The IBM version employs the use of cards for input. Card format is explained in Sections 4.4 and 4.5.

The DEC version employs direct user-program interaction through an on-line system. The routine CONVRT is an additional part of this version.

4.2 Input - DECSYSTEM-20

FLAP interacts directly with the user through the use of prompts. Data may reside on a storage medium that is accessible by FLAP, or may be input by hand by the user at a terminal on-line to the system.

When FLAP is called by the user, the following series of prompts appear. The program will not proceed to the next prompt until a suitable response is received:

```
CONVERT REPOS TO FLAP FORM (1=YES)? *
 (Option to reformat a storage medium resident
  load spectra to FLAP's input standards - see
  Section 3.3.4, CONVRT).

     0 or 1 **

GIVE THE OUTPUT DEVICE NUMBER: *
 (Pick device that output will be directed to)

     x (Integer)  **

ENTER JDIM, JG, JCALL, JHIT, JOR, JEMPTY,JQ,JNT *
 (Option variables - see Section 3., Table 3)

     X,X,X,X,X,X,X,X,  (X is an integer) **

ENTER BETA,B,M,K,NE,BETAIM,EOVERL,CBW,GW,G,SCF,
STATIC,STRE,FOE, PUNCH
 (Engineering values - see Table 4)

     X,X,X,X,X,X,X,X,X,X,X,X,X,X,X **
     (X conforms to definition of variable)
```

ARE YOU INPUTTING DATA BY HAND (1=YES)? *

0 or 1 **

If the answer to the above was 0:

GIVE FILE WHERE DATA IS LOCATED: *
(File is storage medium resident)

Filename **

If the answer to the above was 1:

ENTER J,JA,JB,SMAX,SMIN *
(This prompt will appear until the user has furnished enough data. J should remain 1 as long as the user continues inputting data. When a 0 is input for J, the prompt will not reappear, and processing will begin. JA and JB are the matrix indices. SMAX and SMIN are stress values).

* Denotes FLAP prompt

** Denotes user response.

4.3 Control Cards - IBM

The input deck must consist of a minimum of six cards providing option control values and needed property values, followed by a card group which provides the loaded use ALPHA matrix (and its corresponding stress sets), and terminated by a blank card (Figure 10). Depending on which options are being utilized, a maximum of eight more ALPHA arrays and their corresponding stress sets, can be input. The program will attempt to read nine arrays when all three core subroutines are directed to read in their own arrays, and JNT=3. Depending on the user's option choices, any number of data sets from 1 to 9 may be required as input. Each data set must be terminated by a blank card, therefore the last card input must always be blank, and there must be the same number of blank cards as there are ALPHA sets being input.

This input stream is altered slightly when JNT > 1, (Figure 11). When two or three simultaneous environmental spectra are being used in the calculation of fatigue damage, each new set of option control and property values (6 cards) must be input before each additional spectra can be read in. Thus, for JNT=3, i.e., three simultaneous spectra, the order of input is:

Option control card	(1)
Property value cards	(5)
Loaded stress/ALPHA array	(400 max**)
*Unloaded stress/ALPHA array	(400 max**)
*Impact stress/ALPHA array	(400 max**)
Second option control card	(1)
Second property value cards	(5)
Loaded stress/ALPHA array	(400 max**)
*Unloaded stress/ALPHA array	(400 max**)
*Impact stress/ALPHA array	(400 max**)
Third option control card	(1)
Third property value cards	(5)
Loaded stress/ALPHA array	(400 max**)
*Unloaded stress/ALPHA array	(400 max**)
*Impact stress/ALPHA array	(400 max**)

NOTE: With the first option in GMULTY (Section 3.3.4), no stress sets need be input with the ALPHA array and this portion of the card would be left blank.

* Optional data sets

** Last card must be blank

4.4 Input - IBM

The first card required as input, the option control card, must be input using an I5 format for each of the eight variables (Figure 12 A). Only one value may be input in each of the five column blocks and it must be an integer. Only columns 1 thru 40 are read on this card. Only the prescribed values (Table 4) are meaningful in the program.

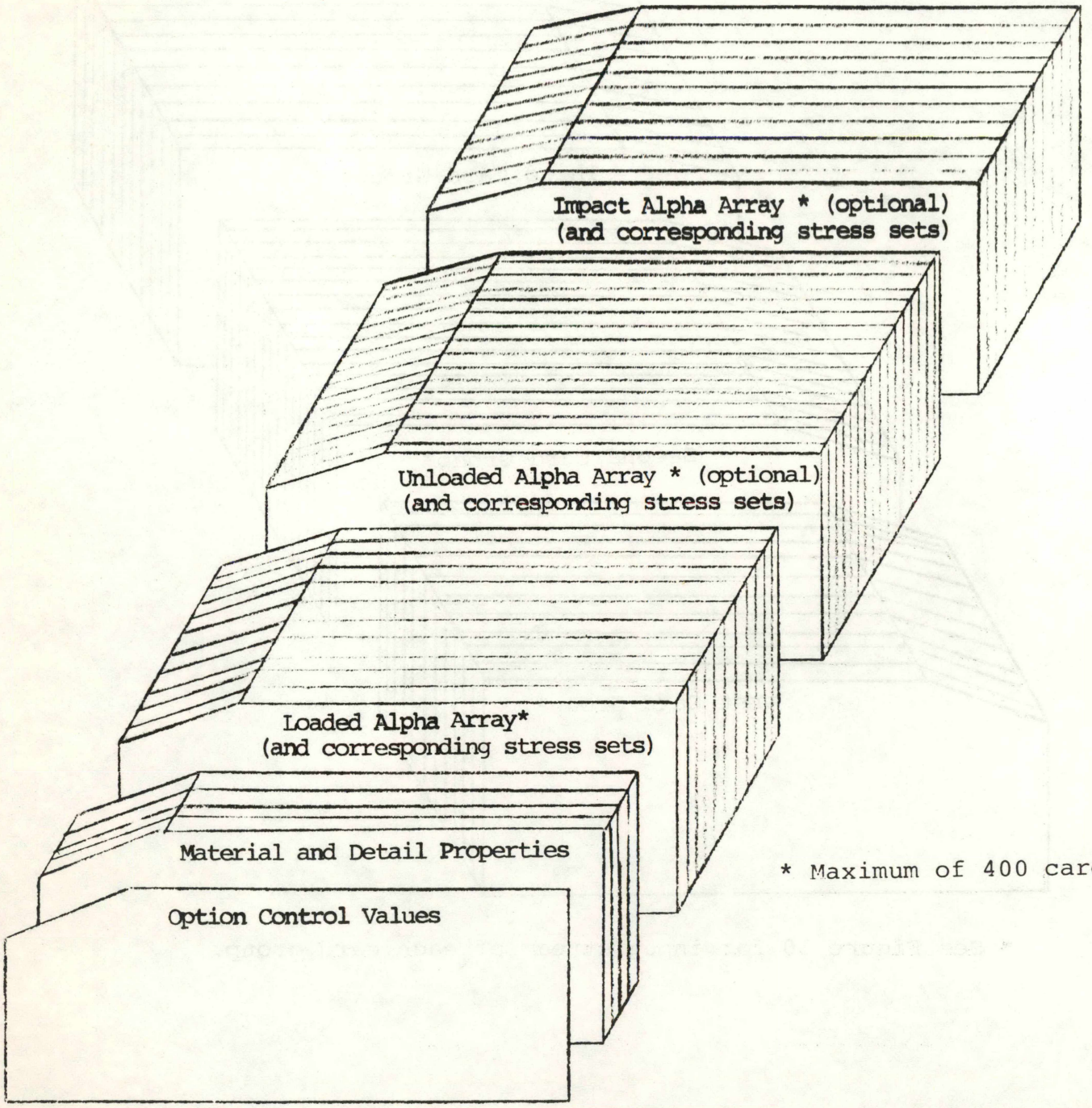
The next five cards required in the input stream (Figure 10) each contain three values input in F20.8 format; the card format then is 3F20.8 (Figure 12 B).

The remainder of the input involves the ALPHA arrays their corresponding stress sets. These values are input on cards with a 3I5, 3F20.8 format. The first integer must be 1 unless it is a blank card to signal the end of a block of data. The next two integers are the row number, followed by the column number, which locate the remaining values on the card in the load spectra array. The first floating point number is read as the STRSMX value, the second number as the STRSMN value and the final number is loaded into the ALPHA array at the location specified by the row and column numbers read on the card. The ALPHA value (ALPHA must be < 1.0) at this location represents the percent occurrence of the stress cycle specified by the STRSMX and STRSMN values read from the same card (unless these values are generated in GMULTY). A blank card then signals the end of a data set.

In Appendix A a sample problem is presented to illustrate the proper input forms and procedure.

Figure 10

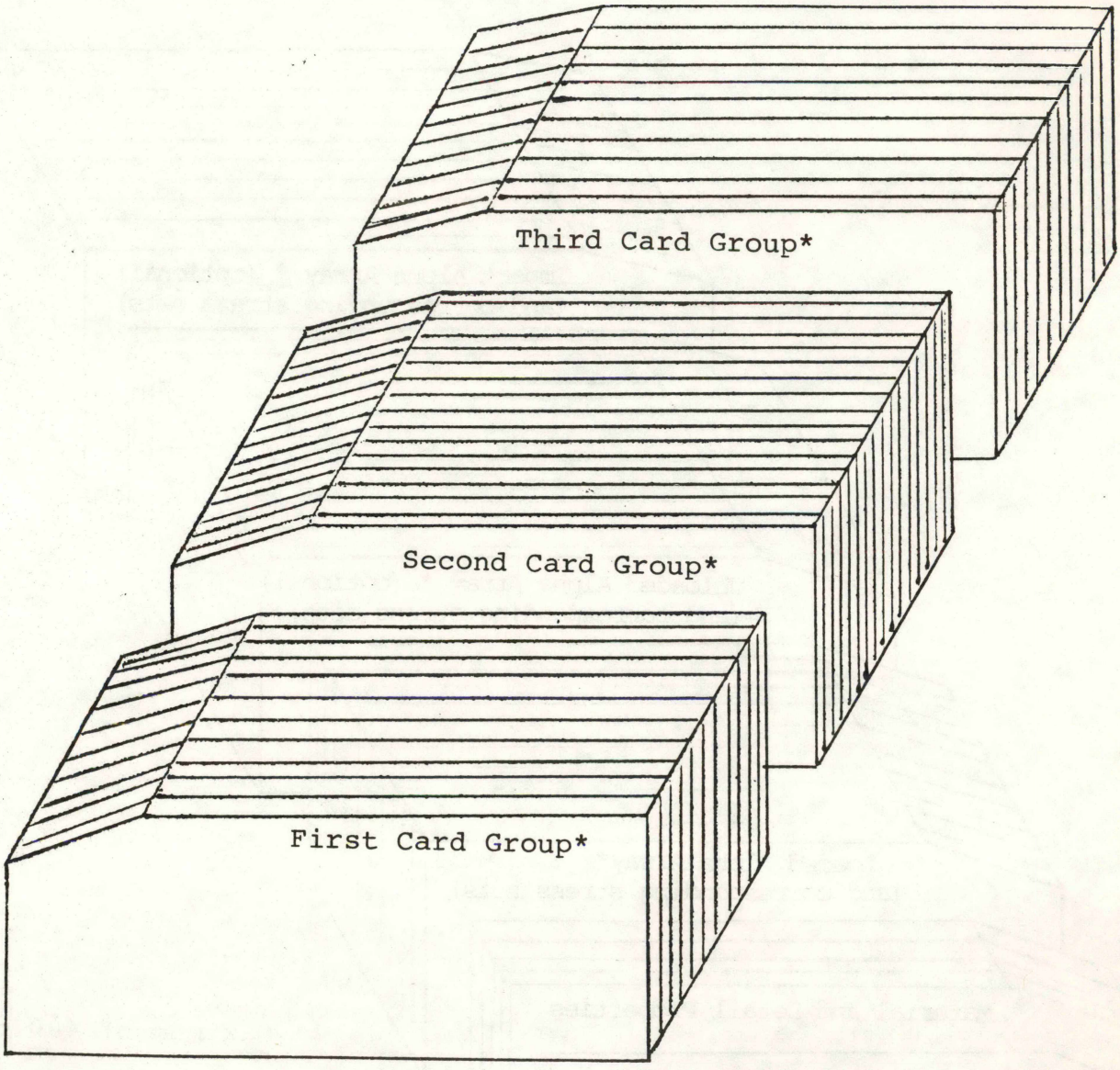
CARD GROUP LAYOUT



* Maximum of 400 cards

Figure 11

DATA INPUT WHEN JNT - 3



* See Figure 10 for input stream of each card group.

Figure 12

CARD FORMAT

A. CARD 1 IN INPUT STREAM:*

	1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40
#1	JDIM	JG	JCALL	JHIT	JOR	JEMPTY	JQ	JNT

*See Table 3 for values and meanings

B. CARD 2 THRU 6 IN INPUT STREAM:*

	1 20	21 40	41 60
#2	BETA	B	M
#3	K	NE	BETAIM
#4	EOVERL	CBW	GW
#5	G	SCF	STATIC
#6	STRE	FOE	PUNCH

*See Table 4 for abbreviation meanings

C. CARD 7 AND BEYOND IN INPUT STREAM:

	1 5	6 10	11 15	16 35	36 55	56 75
#7 ↓	1	Matrix Row	Matrix Column	Maximum Stress	Minimum Stress	Alpha

TABLE 4

PROGRAM DESIGNATORS FOR ENGINEERING VALUES

BETA	-	Total number of spectrum cycles/mile. (β)
B	-	The fatigue limit for the stress ratio R=0, Y intercept on modified Goodman diagram curve. (b)
M	-	The slope of the modified Goodman Diagram curve. (m)
K	-	The absolute value of the slope of the S-N curve. (k)
NE	-	The number of cycles at the fatigue limit. (N_e)*
BETAIM	-	The total number of spectrum cycles/mile due to impact. (β_i)
EOVERL	-	The ratio of Empty mile usage to Loaded mile usage.
CBW	-	The car body weight
G	-	Acceleration increment needed in GMULTY
SCF	-	The value of the stress concentration factor. (K_t or K_f)
STATIC	-	The stresses due to 1G acceleration
STRE	-	Stress value corresponding to the load FOE
FOE	-	Load input for use with STRE for conversion of loads to stresses.
PUNCH	-	Impact load factor employed to modify the LOADED stresses for use in the IMPACT subroutine.
GW	-	The car's gross weight (loaded)

* For steel N_e = 2,000,000 cycles

4.5 Output - IBM and DEC

A sample output is provided in Appendix A for both IBM and DEC versions. The first page of output is a listing of the option control variable values which were input. The second page is a listing of all the variables listed in Table 4. These two pages are designed to eliminate any input errors of the user by providing a convenient check of the input data.

The alpha matrices and their corresponding stress sets are printed out by the respective subroutine using them. Each subroutine identifies the values being used in that particular subroutine and prints out a complete listing of all the values. Each subroutine prints its values on a separate page.

Finally, each type of fatigue damage is listed, i.e., the number of fatigue damage cycles due to loaded, unloaded, and impact situations and their respective fatigue lives in miles. Following these contributory results, the total expected fatigue life in miles is the final listing bracketed by asterisks.

5. SYMBOLS

α_i - Fraction of the total number of cycles applied at given stress levels $S_{max\ i}$ and $S_{min\ i}$.

β - Total number of spectrum cycles per mile.

β_i - Equivalent impact cycles per mile.

b - Fatigue limit for stress ratio R = 0; Y - intercept on modified Goodman diagram curve.

k - Absolute value of the slope of the S-N curve.

m - Slope of the modified Goodman diagram curve.

N_T - Fatigue life (total number of cycles to failure) under a given spectrum loading.

N_i' - Number of cycles applied at each level of $S_{max\ i}$ and $S_{min\ i}$.

N_i - Number of cycles to produce failure at each level of $S_{max\ i}$ and $S_{min\ i}$.

R - Stress ratio, S_{min}/S_{max}.

S_e - Fatigue limit or stress which produces failure in N_e cycles (N_e assumed 2×10^6 cycles for steel).

β_M - Equivalent ratio service miles per impact.

S_i - Fatigue limit computed for failure at N_i cycles.

S_{max} - Maximum or highest algebraic applied stress.

S_{min} - Minimum or lowest algebraic applied stress.

K_t - Theoretical stress concentration factor.

K_f - Fatigue notch factor.

APPENDIX A

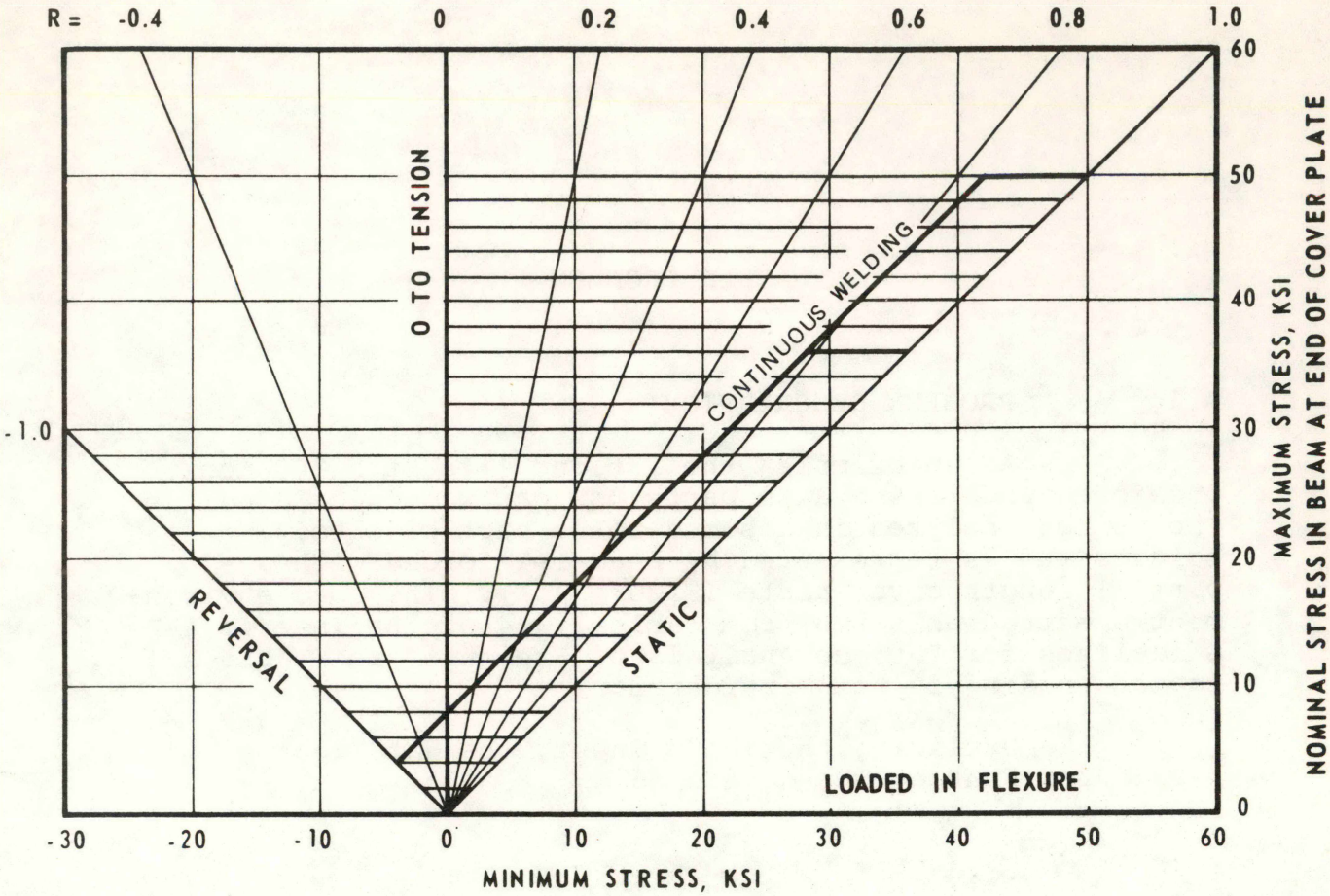
SAMPLE PROBLEM

A.1 PROBLEM DESCRIPTION

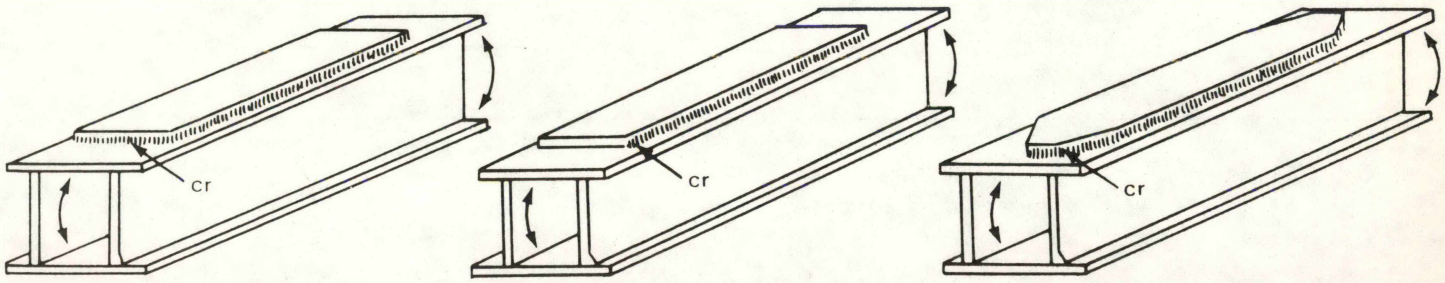
As an example* the fatigue life of a 100 ton cushioned underframe auto parts box car was calculated. The member analyzed is assumed to be part of a body bolster and is represented by beam reinforced with a partial length cover plate (Figure A.1). The road environment spectra was taken from Table 3.3.4 of the Interim AAR Guidelines for Fatigue Analysis of Freight Cars (Table A.1). Impact loading is not considered.

Utilizing the above input, the data deck needed for FLAP can be developed.

* This example is used in Section 2.4.1 of the Interim AAR Guidelines for Fatigue Analysis of Freight Cars.



k, slope of S-N curve = 0.32



BEAMS REINFORCED WITH PARTIAL LENGTH COVER PLATE

FIGURE A.1

ROAD ENVIRONMENT PERCENT OCCURRENCE SPECTRUM

VERTICAL G'S @ BOLSTER
100-TON AUTO PARTS BOX CAR

UP - DOWN

UP - UP

COLUMN NUMBER

ROW NUMBER	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1																				
2																				
3																				
4																				
5					0.0020															
6				0.0020	0.0040	0.0020	0.0040	0.0020												
7				0.0010	0.0030	0.0190	0.0260	0.0140	0.0060	0.0010										
8					0.0020	0.0170	0.0670	0.1740	0.1160	0.0180	0.0010									
9				0.0010	0.0010	0.0030	0.0470	0.4530	1.6290	0.7750	0.0130									
10						0.0010	0.0090	0.2520	5.4630	38.1790	1.4150									
11							0.0010	0.0070	1.3630	49.8420										
12							0.0010		0.0010											
13				0.0010			0.0010													
14																				
15																				
16																				
17																				
18																				
19																				
20																				

MAXIMUM PEAK
VERTICAL ACCELERATION G'S

1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 -0.1 -0.2 -0.3 -0.4 -0.5 -0.6 -0.7 -0.8 -0.9 -1.0

-0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

MINIMUM PEAK
VERTICAL ACCELERATION G'S

DOWN - DOWN

ROAD TEST OF: 100 TON AUTO PART BOX CAR (CUSHION UNDERFRAME)
GROSS RAIL Wt.: 235.500 Lbs. TEST DATE: 11/71
FROM-TO: ST.LOUIS - KANSAS CITY
ON: MO.PAC. RR
SIZE TRAIN:
TEST CAR LOCATED:
MAX. MPH: 60-70 AVG.MPH: 32.3
RECORDED MILEAGE: 157.7
RECORDING TIME: 5.0 Hrs.

SPECTRUM TOTAL: 99.9489
POSITIVE FORCE IS: UP
NEGATIVE FORCE IS: DOWN

TOTAL OCCURRENCES: 111.856
AVG.OCCURRENCES/MILE: 709.322
AVG.CYCLES/MILE: 354.661

TABLE A.1

A.2 IBM VERSION

A.2.1 FLAP Input Description

Utilizing the input described in Section A.1., the data deck needed for FLAP can be developed. For the option control card, which is the first card input into the program, the following values are assigned:

JDIM=20;	Input spectrum is 20x20 matrix
JG=1;	GMULTY subroutine will be utilized to convert input acceleration values into stresses
JCALL=1;	EMPTY subroutine is called to calculate fatigue damage due to unloaded road service
JHIT=0;	Impact damage is not considered. IMPACT subroutine is not to be called.
JOR=1;	The first option in GMULTY is selected
JEMPTY=1;	Stresses for unloaded road service are obtained by multiplying stresses in loaded service by the ratio CBW/GW
JQ=0;	Default value. This value is not required since IMPACT subroutine is not called
JNT=1;	ADDNT is not utilized since only one set of spectra is input

Once the options have been selected, the necessary material and detail property values are input. Note: Table 4 contains an explanation of terms.

For the second car of the input deck:

BETA = 354.661*
B = 8000.0*
M = 1.00*

For the third card:

K = 0.32*
NE = 2000000.0*
BETAIM Blank; no value needed to input since JHIT was set at zero.

For the fourth card:

EOVERL = 0.98* (See Appendix B for appropriate values)
CBW = 90000.0 } These values must be input
GW = 240000.0 } since JCALL=1

For the fifth card:

G = 0.1 This value must be input since JOR=1
SCF No value need be input since JOR=1
STATIC = 10000.0 This value must be input since JOR=1

For the sixth card:

This card may be input as a blank since none of these values are required. The STRE and FOE values are needed only for options 2 and 3 of GMULTY and the PUNCH value is only needed in the IMPACT subroutine.

The proper format for cards 1-6 is shown in Figure A.2.

The remainder of the cards required for the input into FLAP will contain the occurrence (ALPHA) values of the input spectrum.

* This value must always be input

No values will be input for the stresses since all values will be generated by the GMULTY subroutine. The ALPHA values input are the values in Table A.1 divided by 100, since the values in Table A.1 are expressed as percentages and FLAP requires ALPHA values $\leq$ 1.0. The proper format for these cards are shown in Figure A.3.

A listing of the FLAP program and the resulting output follow. It is noted on pages A.23 and A.24 that all option values and input parameters are printed out. Additionally, the STRSMX, STRSMN, and ALPHA arrays (loaded road service) are printed out on page A.25. Note: if ADDNT is selected, i.e., more than one set of input loads, all input values and arrays are printed out. Finally, the number of cycles to failure and the fatigue life, in miles, are printed out for loaded car service (page A.25), unloaded car service (page A.26) and total service (page A.27).

FIGURE A.2

SAMPLE PROBLEM CARD FORMAT

A. CARD 1 IN INPUT STREAM

	1 5	6 10	11 15	16 20	21 25	26 30	31 35	36 40
#1	20	1	1	0	1	1	0	1

B. CARD 2 THRU 6 IN INPUT STREAM

	1 20	21 40	41 60
#2	354.661	8000.0	1.00
#3	0.32	2000000.0	
#4	0.98	90000.0	240000.0
#5	0.1		10000.0
#6			

FIGURE A.3

SAMPLE PROBLEM CARD FORMAT

1	5 6 10	11 15	16 35	36 55	56 75
1	5	5			.00002
1	6	4			.00002
1	6	5			.00004
1	6	6			.00002
1	6	7			.00004
1	6	8			.00002
1	7	4			.00001
1	7	5			.00003
1	7	6			.00019
1	7	7			.00026
1	7	8			.00014
1	7	9			.00006
1	7	10			.00001
1	8	5			.00002
1	8	6			.00017
1	8	7			.00067
1	8	8			.00174
1	8	9			.00116
1	8	10			.00018
1	8	11			.00001
1	9	4			.00001
1	9	5			.00001
1	9	6			.00003
1	9	7			.00047
1	9	8			.00453
1	9	9			.01629
1	9	10			.00775
1	9	11			.00013
1	10	6			.00001
1	10	7			.00009
1	10	8			.00252
1	10	9			.05463
1	10	10			.38179
1	10	11			.01415
1	11	7			.00001
1	11	8			.00007
1	11	9			.01363
1	11	10			.49842
1	12	7			.00001
1	12	9			.00001
1	13	4			.00001
1	13	7			.00001
blank card					

A.2.2 FLAP Listing - IBM

```
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(20,20),STRSMN(20,20),ALPHA(20,20)
      DIMENSION IMSTMX(20,20),IMSTMN(20,20),IMALF(20,20)
      DIMENSION EMSTMX(20,20),EMSTMN(20,20),EMALF(20,20)
      DIMENSION NT1(3),NT2(3),NT3(3)
      NTLOAD=0.0
      NTEMP=0.0
      NTIMP=0.0
      JAR=1
      JIFF=1
      JEFF=2
      JYFF=3
   23 READ (5,914) JDIM,JG,JCALL,JHIT,JOR,JEMPTY,JQ,JNT
  914 FORMAT (8I5)
      WRITE (6,915) JDIM,JG,JCALL,JHIT,JOR,JEMPTY,JQ,JNT
  915 FORMAT ('1',' JDIM   =',I3,//,' JG     =',I3,//,' JCALL  =',I3,//,
     1' JHIT   =',I3,//,' JOR    =',I3,//,' JEMPTY =',I3,//,' JQ     =',
     2I3,//,' JNT    =',I3)
      READ (5,4) BETA,B,M,K,NE,BETAIM
    4 FORMAT (3F20.8/3F20.8)
      READ (5,1) EOVERL,CBW,GW
    1 FORMAT (3F20.8)
      READ (5,29) G,SCF,STATIC,STRE,FOE,PUNCH
   29 FORMAT (3F20.8/3F20.8)
      WRITE (6,32) BETA,B,M,K,NE,BETAIM,EOVERL,CBW,GW,G
   32 FORMAT ('1',' TOTAL NUMBER OF SPECTRUM CYCLES PER MILE = ',F20.8,
     1//,' THE FATIGUE LIMIT FOR THE STRESS RATIO R =0, Y-INTERCEPT ON M
     2ODIFIED GOODMAN DIAGRAM CURVE = ',F20.8,//,' THE SLOPE OF THE MODI
     3FIED GOODMAN DIAGRAM CURVE = ',F20.8,//,' THE ABSOLUTE VALUE OF TH
     4E SLOPE OF THE S-N CURVE = ',F20.8,//,' THE NUMBER OF CYCLES AT TH
     5E FATIGUE LIMIT = ',F20.8,//,' THE TOTAL NUMBER OF SPECTRUM CYCLES
     6 PER MILE DUE TO IMPACT = ',F20.8,//,' THE RATIO OF EMPTY MILE USA
     7GE TO LOADED MILE USAGE = ',F20.8,//,' THE CAR BODY WEIGHT = ',
     8F20.8,//,' THE GROSS WEIGHT = ',F20.8,//,' THE NUMBER OF G S ACCEL
     9ERATION ENCOUNTERED DURING USAGE = ',F20.8,//)
      WRITE (6,2) SCF,STATIC,STRE,FOE,PUNCH
    2 FORMAT (' THE VALUE OF THE STRESS CONCENTRATION FACTOR = ',F20.8,
     1//,' THE STATIC LG STRESS DUE TO DEAD AND LIVE LOADS = ',F20.8,//,
     2' FOR CONVERSION REQUIREMENTS A STRESS OF ',F20.8,' IS EXPERIENCED
     3 WHEN A LOAD OF ',F20.8,' IS APPLIED',//,' THE IMPACT LOAD FACTOR
     4= ',F20.8)
      BETALD=BETA/(1+EOVERL)
      BETAEM=BETA-BETALD
      IF (JG.EQ.0) GO TO 111
      CALL GMULTY (JDIM,STRSMX,STRSMN,ALPHA,G,SCF,STATIC,STRE,FOE,JOR)
  111 CALL LOADED (JDIM,JG,STRSMX,STRSMN,ALPHA,NTLOAD,B,M,NE,K)
      IF (JNT.NE.1) CALL ADDNT (NTLOAD,JAR,JIFF,NT1,NT2,NT3)
   15 IF (JCALL.EQ.0) GO TO 16
      CALL EMPTY (JDIM,JEMPTY,CBW,GW,NTEMP,STRSMX,STRSMN,ALPHA,B,M,NE,
     1K,EMSTMX,EMSTMN,EMALF)
      IF (JNT.NE.1) CALL ADDNT (NTEMP,JAR,JEFF,NT1,NT2,NT3)
   16 IF (JHIT.EQ.0) GO TO 18
      CALL IMPACT (JDIM,JHIT,NTIMP,STRSMX,STRSMN,ALPHA,B,M,NE,K,JQ,
```

```
  1IMSTMX,IMSTMN,IMALF,PUNCH)
   IF (JNT.NE.1) CALL ADDNT (NTIMP,JAR,JYFF,NT1,NT2,NT3)
18 IF (JNT.EQ.JAR) GO TO 24
   JAR=JAR+1
   GO TO 23
24 WRITE (6,41) NTLOAD
41 FORMAT ('1',' FATIGUE CYCLES TO FAILURE FROM LOADED CAR ROAD
  1 SERVICE IS ',F16.2,/)
   WLIFE=NTLOAD/BETALD
   IF (WLIFE.NE.0.0) GO TO 26
   WRITE (6,21)
21 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE LOADED CONDITION IS
  1 CONSIDERED, THE FATIGUE LIFE IS INFINITE.',////)
   GO TO 60
26 WRITE (6,42) WLIFE
42 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE LOADED CONDITION IS
  1 CONSIDERED, THE FATIGUE LIFE IS ',F16.2,' MILES',////)
60 IF (JCALL.EQ.0) GO TO 61
   WRITE (6,43) NTEMP
43 FORMAT ('1',' FATIGUE CYCLES TO FAILURE FROM EMPTY CAR ROAD SERVIC
  1E IS ',F16.2,/)
   WLIFE=NTEMP/BETAEM
   IF (WLIFE.NE.0.0) GO TO 22
   WRITE (6,25)
25 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE EMPTY CONDITION IS
  1 CONSIDERED, THE FATIGUE LIFE IS INFINITE.',////)
   GO TO 61
22 WRITE (6,44) WLIFE
44 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE EMPTY CONDITION IS
  1 CONSIDERED, THE FATIGUE LIFE IS ',F16.2,' MILES',////)
61 IF (JHIT.EQ.0) GO TO 62
   WRITE (6,49) NTIMP
49 FORMAT ('1',' FATIGUE CYCLES TO FAILURE FROM YARD OPERATIONS IS',
  1F16.2,/)
   WLIFE=NTIMP/BETAIM
   IF (WLIFE.NE.0.0) GO TO 28
   WRITE (6,27)
27 FORMAT (' IF ONLY DAMAGE RESULTING FROM IMPACT IS CONSIDERED, THE
  1 FATIGUE LIFE IS INFINITE.',////)
   GO TO 62
28 WRITE (6,40) WLIFE
40 FORMAT (' IF ONLY DAMAGE RESULTING FROM IMPACT IS CONSIDERED, THE
  1 FATIGUE LIFE IS ',F16.2,' MILES',////)
62 IF (NTLOAD.EQ.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.EQ.0.0) GO TO 11
   IF (NTLOAD.NE.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.EQ.0.0)
  1LIFE=NTLOAD/BETALD
   IF (NTLOAD.EQ.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.EQ.0.0)
  1LIFE=NTEMP/BETAEM
   IF (NTLOAD.EQ.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.NE.0.0)
  1LIFE=NTIMP/BETAIM
   IF (NTLOAD.NE.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.EQ.0.0)
  1LIFE=1/(BETALD/NTLOAD+BETAEM/NTEMP)
   IF (NTLOAD.NE.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.NE.0.0)
  1LIFE=1/(BETALD/NTLOAD+BETAIM/NTIMP)
```

```
      IF (NTLOAD.EQ.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.NE.0.0)
     1LIFE=1/(BETAEM/NTEMP+BETAIM/NTIMP)
      IF (NTLOAD.NE.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.NE.0.0)
     1LIFE=1/(BETALD/NTLOAD+BETAEM/NTEMP+BETAIM/NTIMP)
      WRITE (6,10) LIFE
   10 FORMAT ('1',' **********',//,' THE TOTAL EXPECTED FATIGUE LIFE IS
     1',E14.6,' MILES',//,' **********')
      GO TO 13
   11 WRITE (6,12)
   12 FORMAT (' **********',//,' NO DAMAGE',//,' **********')
   13 STOP
      END
@
```

```
      SUBROUTINE LOADED (JDIM,JG,STRSMX,STRSMN,ALPHA,NTLOAD,B,M,NE,K)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      IF (JG.EQ.1) GO TO 200
      DO 3 JA=1,JDIM
      DO 2 JB=1,JDIM
      STRSMX(JA,JB)=0.0
      STRSMN(JA,JB)=0.0
      ALPHA(JA,JB)=0.0
    2 CONTINUE
    3 CONTINUE
    5 READ (5,6) J,JA,JB,SMAX,SMIN,ALF
    6 FORMAT (3I5,3F20.8)
      IF (J.EQ.0) GO TO 200
      STRSMX(JA,JB)=SMAX
      STRSMN(JA,JB)=SMIN
      ALPHA(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
  200 NT=0.0
      WRITE (6,49)
   49 FORMAT ('1',//, '            ********************          THE FOLLOW
     1ING ARE THE INPUTTED ARRAYS:          *********************',//)
      DO 76 JA=1,JDIM
      DO 75 JB=1,JDIM
      IF (ALPHA(JA,JB).EQ.0.0) GO TO 75
      WRITE (6,77) JA,JB,STRSMX(JA,JB),JA,JB,STRSMN(JA,JB),JA,JB,
     1ALPHA(JA,JB)
   77 FORMAT (' STRSMX(',I2,',',I2,') =',F20.8,'   *  STRSMN(',I2,',',I2,
     1') =',F20.8,'   *  ALPHA(',I2,',',I2,') =',F20.8)
   75 CONTINUE
   76 CONTINUE
      DO 9 JA=1,JDIM
      DO 8 JB=1,JDIM
      IF (STRSMX(JA,JB).LE.0.0) GO TO 8
      IF (ALPHA(JA,JB).EQ.0.0) GO TO 8
      STRESS=STRSMN(JA,JB)
      IF (STRESS.LT.0.0) STRESS=-(STRESS)
      IF (STRESS.LT.STRSMX(JA,JB)) GO TO 7
      STRSMX(JA,JB)=(STRESS+STRSMX(JA,JB))/2.0
      STRSMN(JA,JB)=-(STRSMX(JA,JB))
    7 R=STRSMN(JA,JB)/STRSMX(JA,JB)
      SE=B/(1-M*R)
      IF (SE.GT.STRSMX(JA,JB)) GO TO 8
      NI=NE/((STRSMX(JA,JB)/SE)**(1/K))
      X=ALPHA(JA,JB)/NI
      NT=X+NT
    8 CONTINUE
    9 CONTINUE
      IF (NT.EQ.0.0) GO TO 13
      NTLOAD=1/NT
   13 RETURN
      END
@
```

```
      SUBROUTINE EMPTY (JDIM,JEMPTY,CBW,GW,NTEMP,STRSMX,STRSMN,ALPHA,
     1B,M,NE,K,EMSTMX,EMSTMN,EMALF)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      DIMENSION EMSTMX(JDIM,JDIM),EMSTMN(JDIM,JDIM),EMALF(JDIM,JDIM)
      NT=0.0
      DO 3 JA=1,JDIM
      DO 7 JB=1,JDIM
      EMSTMX(JA,JB)=0.0
      EMSTMN(JA,JB)=0.0
      EMALF(JA,JB)=0.0
    7 CONTINUE
    3 CONTINUE
      IF (JEMPTY.EQ.1) GO TO 1
    5 READ (5,6) J,JA,JB,SMAX,SMIN,ALF
    6 FORMAT (3I5,3F20.8)
      IF (J.EQ.0) GO TO 200
      EMSTMX(JA,JB)=SMAX
      EMSTMN(JA,JB)=SMIN
      EMALF(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
  200 WRITE (6,45)
   45 FORMAT ('1',//, '            ********************          THE FOLLOW
     1ING ARE THE INPUTTED ARRAYS:          ********************',//)
      DO 76 JA=1,JDIM
      DO 75 JB=1,JDIM
      IF (EMALF(JA,JB).EQ.0.0) GO TO 75
      WRITE (6,77) JA,JB,EMSTMX(JA,JB),JA,JB,EMSTMN(JA,JB),JA,JB,
     1EMALF(JA,JB)
   77 FORMAT (' EMSTMX(',I2,',',I2,') =',F20.8,'     EMSTMN(',I2,',',I2,
     1') =',F20.8,'     EMALF(',I2,',',I2,') =',F20.8)
   75 CONTINUE
   76 CONTINUE
      DO 9 JA=1,JDIM
      DO 8 JB=1,JDIM
      IF (EMSTMX(JA,JB).LE.0.0) GO TO 8
      IF (EMALF(JA,JB).EQ.0.0) GO TO 8
      STRESS=EMSTMN(JA,JB)
      IF (STRESS.LT.0.0) STRESS=-(STRESS)
      IF (STRESS.LT.EMSTMX(JA,JB)) GO TO 8
      EMSTMX(JA,JB)=(STRESS+EMSTMX(JA,JB))/2
      EMSTMN(JA,JB)=-(EMSTMX(JA,JB))
    8 CONTINUE
    9 CONTINUE
      GO TO 2
    1 DO 49 JA=1,JDIM
      DO 94 JB=1,JDIM
      EMSTMX(JA,JB)=STRSMX(JA,JB)*CBW/GW
      EMSTMN(JA,JB)=STRSMN(JA,JB)*CBW/GW
      EMALF(JA,JB)=ALPHA(JA,JB)
   94 CONTINUE
   49 CONTINUE
    2 DO 34 JA=1,JDIM
```

```
      DO 43 JB=1,JDIM
      IF (EMSTMX(JA,JB).LE.0.0) GO TO 43
      IF (EMALF(JA,JB).EQ.0.0) GO TO 43
      R=EMSTMN(JA,JB)/EMSTMX(JA,JB)
      SE=B/(1-M*R)
      IF (SE.GT.EMSTMX(JA,JB)) GO TO 43
      NI=NE/((EMSTMX(JA,JB)/SE)**(1/K))
      X=EMALF(JA,JB)/NI
      NT=X+NT
   43 CONTINUE
   34 CONTINUE
      IF (NT.EQ.0.0) GO TO 13
      NTEMP=1/NT
   13 RETURN
      END
@
```

```
      SUBROUTINE IMPACT (JDIM,JHIT,NTIMP,STRSMX,STRSMN,ALPHA,B,M,NE,K,
     1JQ,IMSTMX,IMSTMN,IMALF,PUNCH)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      DIMENSION IMSTMX(JDIM,JDIM),IMSTMN(JDIM,JDIM),IMALF(JDIM,JDIM)
      NT=0.0
      DO 3 JA=1,JDIM
      DO 7 JB=1,JDIM
      IMSTMX(JA,JB)=0.0
      IMSTMN(JA,JB)=0.0
      IMALF(JA,JB)=0.0
    7 CONTINUE
    3 CONTINUE
      IF (JQ.EQ.1) GO TO 1
    5 READ (5,6) J,JA,JB,SMAX,SMIN,ALF
    6 FORMAT (3I5,3F20.8)
      IF (J.EQ.0) GO TO 200
      IMSTMX(JA,JB)=SMAX
      IMSTMN(JA,JB)=SMIN
      IMALF(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
  200 WRITE (6,45)
   45 FORMAT ('1',//, '            ********************          THE FOLLOW
     1ING ARE THE INPUTTED ARRAYS:          ********************',//)
      DO 76 JA=1,JDIM
      DO 75 JB=1,JDIM
      IF (IMALF(JA,JB).EQ.0.0) GO TO 75
      WRITE (6,77) JA,JB,IMSTMX(JA,JB),JA,JB,IMSTMN(JA,JB),JA,JB,
     1IMALF(JA,JB)
   77 FORMAT (' IMSTMX(',I2,',',I2,') =',F20.8,'     IMSTMN(',I2,',',I2,
     1') =',F20.8,'     IMALF(',I2,',',I2,') =',F20.8)
   75 CONTINUE
   76 CONTINUE
      DO 9 JA=1,JDIM
      DO 8 JB=1,JDIM
      IF (IMSTMX(JA,JB).LE.0.0) GO TO 8
      IF (IMALF(JA,JB).EQ.0.0) GO TO 8
      STRESS=IMSTMN(JA,JB)
      IF (STRESS.LT.0.0) STRESS=-(STRESS)
      IF (STRESS.LT.IMSTMX(JA,JB)) GO TO 8
      IMSTMX(JA,JB)=(STRESS+IMSTMX(JA,JB))/2
      IMSTMN(JA,JB)=-(IMSTMX(JA,JB))
    8 CONTINUE
    9 CONTINUE
      GO TO 2
    1 DO 49 JA=1,JDIM
      DO 94 JB=1,JDIM
      IMSTMX(JA,JB)=STRSMX(JA,JB)*PUNCH
      IMSTMN(JA,JB)=STRSMN(JA,JB)*PUNCH
      IMALF(JA,JB)=ALPHA(JA,JB)
   94 CONTINUE
   49 CONTINUE
    2 DO 34 JA=1,JDIM
```

```
      DO 43 JB=1,JDIM
      IF (IMSTMX(JA,JB).LE.0.0) GO TO 43
      IF (IMALF(JA,JB).EQ.0.0) GO TO 43
      R=IMSTMN(JA,JB)/IMSTMX(JA,JB)
      SE=B/(1-M*R)
      IF (SE.GT.IMSTMX(JA,JB)) GO TO 43
      NI=NE/((IMSTMX(JA,JB)/SE)**(1/K))
      X=IMALF(JA,JB)/NI
      NT=X+NT
   43 CONTINUE
   34 CONTINUE
      IF (NT.EQ.0.0) GO TO 13
      NTIMP=1/NT
   13 RETURN
      END
@
```

```
      SUBROUTINE GMULTY (JDIM,STRSMX,STRSMN,ALPHA,G,SCF,STATIC,STRE,FOE,
     1JOR)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      DO 8 JA=1,JDIM
      DO 9 JB=1,JDIM
      STRSMX(JA,JB)=0.0
      STRSMN(JA,JB)=0.0
      ALPHA(JA,JB)=0.0
    9 CONTINUE
    8 CONTINUE
    5 READ (5,4) J,JA,JB,SMAX,SMIN,ALF
    4 FORMAT (3I5,3F20.8)
      IF (J.EQ.0) GO TO 6
      STRSMX(JA,JB)=SMAX
      STRSMN(JA,JB)=SMIN
      ALPHA(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
    6 GO TO (7,14,20),JOR
    7 JZERO=JDIM/2
      DO 3 JA=1,JDIM
      DO 2 JB=1,JDIM
      IF (ALPHA(JA,JB).EQ.0.0) GO TO 2
      QUAN1=JZERO-JA+1
      QUAN2=JB-JZERO-1
      STRSMX(JA,JB)=G*QUAN1*STATIC+STATIC
      STRSMN(JA,JB)=G*QUAN2*STATIC+STATIC
      IX=STRSMX(JA,JB)
      IN=STRSMN(JA,JB)
      RIX=IX
      RIN=IN
      RX=STRSMX(JA,JB)-RIX
      RN=STRSMN(JA,JB)-RIN
      IF (RX.GE.0.5) GO TO 10
      STRSMX(JA,JB)=RIX
      GO TO 11
   10 STRSMX(JA,JB)=RIX+1.0
   11 IF (RN.GE.0.5) GO TO 12
      STRSMN(JA,JB)=RIN
      GO TO 2
   12 STRSMN(JA,JB)=RIN+1.0
    2 CONTINUE
    3 CONTINUE
      GO TO 15
   14 DO 18 JA=1,JDIM
      DO 19 JB=1,JDIM
      STRSMX(JA,JB)=STRSMX(JA,JB)*SCF*STRE/FOE+STATIC
      STRSMN(JA,JB)=STRSMN(JA,JB)*SCF*STRE/FOE+STATIC
   19 CONTINUE
   18 CONTINUE
      GO TO 15
   20 DO 21 JA=1,JDIM
      DO 22 JB=1,JDIM
```

```
      STRSMX(JA,JB)=STRSMX(JA,JB)*STRE/FOE
      STRSMN(JA,JB)=STRSMN(JA,JB)*STRE/FOE
   22 CONTINUE
   21 CONTINUE
   15 RETURN
      END
@
```

```
      SUBROUTINE ADDNT (NT,JAR,JIFF,NT1,NT2,NT3)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION NT1(3),NT2(3),NT3(3)
      IF (JIFF.NE.1) GO TO 1
      NT1(JAR)=NT
      IF (JAR.EQ.1) RETURN
      IF (JAR.NE.2) GO TO 4
      IF (NT1(1).EQ.0.0.AND.NT1(2).EQ.0.0) NT=0.0
      IF (NT1(1).NE.0.0.AND.NT1(2).EQ.0.0) NT=NT1(1)
      IF (NT1(1).EQ.0.0.AND.NT1(2).NE.0.0) NT=NT1(2)
      IF (NT1(1).NE.0.0.AND.NT1(2).NE.0.0) NT=1/(1/NT1(1)+1/NT1(2))
    4 IF (JAR.NE.3) GO TO 1
      IF (NT1(1).EQ.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).EQ.0.0) NT=0.0
      IF (NT1(1).NE.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).EQ.0.0) NT=NT1(1)
      IF (NT1(1).EQ.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).EQ.0.0) NT=NT1(2)
      IF (NT1(1).EQ.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).NE.0.0) NT=NT1(3)
      IF (NT1(1).NE.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).EQ.0.0)
     1NT=1/(1/NT1(1)+1/NT1(2))
      IF (NT1(1).NE.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).NE.0.0)
     1NT=1/(1/NT1(1)+1/NT1(3))
      IF (NT1(1).EQ.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).NE.0.0)
     1NT=1/(1/NT1(2)+1/NT1(3))
      IF (NT1(1).NE.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).NE.0.0)
     1NT=1/(1/NT1(1)+1/NT1(2)+1/NT1(3))
    1 IF (JIFF.NE.2) GO TO 2
      NT2(JAR)=NT
      IF (JAR.EQ.1) RETURN
      IF (JAR.NE.2) GO TO 5
      IF (NT2(1).EQ.0.0.AND.NT2(2).EQ.0.0) NT=0.0
      IF (NT2(1).NE.0.0.AND.NT2(2).EQ.0.0) NT=NT2(1)
      IF (NT2(1).EQ.0.0.AND.NT2(2).NE.0.0) NT=NT2(2)
      IF (NT2(1).NE.0.0.AND.NT2(2).NE.0.0) NT=1/(1/NT2(1)+1/NT2(2))
    5 IF (JAR.NE.3) GO TO 2
      IF (NT2(1).EQ.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).EQ.0.0) NT=0.0
      IF (NT2(1).NE.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).EQ.0.0) NT=NT2(1)
      IF (NT2(1).EQ.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).EQ.0.0) NT=NT2(2)
      IF (NT2(1).EQ.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).NE.0.0) NT=NT2(3)
      IF (NT2(1).NE.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).EQ.0.0)
     1NT=1/(1/NT2(1)+1/NT2(2))
      IF (NT2(1).NE.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).NE.0.0)
     1NT=1/(1/NT2(1)+1/NT2(3))
      IF (NT2(1).EQ.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).NE.0.0)
     1NT=1/(1/NT2(2)+1/NT2(3))
      IF (NT2(1).NE.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).NE.0.0)
     1NT=1/(1/NT2(1)+1/NT2(2)+1/NT2(3))
    2 IF (JIFF.NE.3) GO TO 3
      NT3(JAR)=NT
      IF (JAR.EQ.1) RETURN
      IF (JAR.NE.2) GO TO 6
      IF (NT3(1).EQ.0.0.AND.NT3(2).EQ.0.0) NT=0.0
      IF (NT3(1).NE.0.0.AND.NT3(2).EQ.0.0) NT=NT3(1)
      IF (NT3(1).EQ.0.0.AND.NT3(2).NE.0.0) NT=NT3(2)
      IF (NT3(1).NE.0.0.AND.NT3(2).NE.0.0) NT=1/(1/NT3(1)+1/NT3(2))
```

```
    6 IF (JAR.NE.3) GO TO 3
      IF (NT3(1).EQ.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).EQ.0.0) NT=0.0
      IF (NT3(1).NE.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).EQ.0.0) NT=NT3(1)
      IF (NT3(1).EQ.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).EQ.0.0) NT=NT3(2)
      IF (NT3(1).EQ.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).NE.0.0) NT=NT3(3)
      IF (NT3(1).NE.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).EQ.0.0)
     1NT=1/(1/NT3(1)+1/NT3(2))
      IF (NT3(1).NE.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).NE.0.0)
     1NT=1/(1/NT3(1)+1/NT3(3))
      IF (NT3(1).EQ.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).NE.0.0)
     1NT=1/(1/NT3(2)+1/NT3(3))
      IF (NT3(1).NE.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).NE.0.0)
     1NT=1/(1/NT3(1)+1/NT3(2)+1/NT3(3))
    3 RETURN
      END
@
```

A.2.3 FLAP Output - IBM

```
JDIM   = 20
JG     = 1
JCALL  = 1
JHIT   = 0
JOR    = 1
JEMPTY = 1
JQ     = 0
JNT    = 1
```

TOTAL NUMBER OF SPECTRUM CYCLES PER MILE = 354.66100000

THE FATIGUE LIMIT FOR THE STRESS RATIO R =0, Y-INTERCEPT ON MODIFIED GOODMAN DIAGRAM CURVE = 8000.00000000

THE SLOPE OF THE MODIFIED GOODMAN DIAGRAM CURVE = 0.99900000

THE ABSOLUTE VALUE OF THE SLOPE OF THE S-N CURVE = 0.32000000

THE NUMBER OF CYCLES AT THE FATIGUE LIMIT = 2000000.00000000

THE TOTAL NUMBER OF SPECTRUM CYCLES PER MILE DUE TO IMPACT = 0.0

THE RATIO OF EMPTY MILE USAGE TO LOADED MILE USAGE = 0.98000000

THE CAR BODY WEIGHT = 90000.00000000

THE GROSS WEIGHT = 240000.00000000

THE NUMBER OF G S ACCELERATION ENCOUNTERED DURING USAGE = 0.10000000

THE VALUE OF THE STRESS CONCENTRATION FACTOR = 0.0

THE STATIC LG STRESS DUE TO DEAD AND LIVE LOADS = 10000.00000000

FOR CONVERSION REQUIREMENTS A STRESS OF 0.0 IS EXPERIENCED WHEN A LOAD OF 0.0 IS APPLIED

THE IMPACT LOAD FACTOR = 0.0

********** THE FOLLOWING ARE

STRSMX(5, 5) =	16000.00000000	*	STRSMN(5, 5) =
STRSMX(6, 4) =	15000.00000000	*	STRSMN(6, 4) =
STRSMX(6, 5) =	15000.00000000	*	STRSMN(6, 5) =
STRSMX(6, 6) =	15000.00000000	*	STRSMN(6, 6) =
STRSMX(6, 7) =	15000.00000000	*	STRSMN(6, 7) =
STRSMX(6, 8) =	15000.00000000	*	STRSMN(6, 8) =
STRSMX(7, 4) =	14000.00000000	*	STRSMN(7, 4) =
STRSMX(7, 5) =	14000.00000000	*	STRSMN(7, 5) =
STRSMX(7, 6) =	14000.00000000	*	STRSMN(7, 6) =
STRSMX(7, 7) =	14000.00000000	*	STRSMN(7, 7) =
STRSMX(7, 8) =	14000.00000000	*	STRSMN(7, 8) =
STRSMX(7, 9) =	14000.00000000	*	STRSMN(7, 9) =
STRSMX(7,10) =	14000.00000000	*	STRSMN(7,10) =
STRSMX(8, 5) =	13000.00000000	*	STRSMN(8, 5) =
STRSMX(8, 6) =	13000.00000000	*	STRSMN(8, 6) =
STRSMX(8, 7) =	13000.00000000	*	STRSMN(8, 7) =
STRSMX(8, 8) =	13000.00000000	*	STRSMN(8, 8) =
STRSMX(8, 9) =	13000.00000000	*	STRSMN(8, 9) =
STRSMX(8,10) =	13000.00000000	*	STRSMN(8,10) =
STRSMX(8,11) =	13000.00000000	*	STRSMN(8,11) =
STRSMX(9, 4) =	12000.00000000	*	STRSMN(9, 4) =
STRSMX(9, 5) =	12000.00000000	*	STRSMN(9, 5) =
STRSMX(9, 6) =	12000.00000000	*	STRSMN(9, 6) =
STRSMX(9, 7) =	12000.00000000	*	STRSMN(9, 7) =
STRSMX(9, 8) =	12000.00000000	*	STRSMN(9, 8) =
STRSMX(9, 9) =	12000.00000000	*	STRSMN(9, 9) =
STRSMX(9,10) =	12000.00000000	*	STRSMN(9,10) =
STRSMX(9,11) =	12000.00000000	*	STRSMN(9,11) =
STRSMX(10, 6) =	11000.00000000	*	STRSMN(10, 6) =
STRSMX(10, 7) =	11000.00000000	*	STRSMN(10, 7) =
STRSMX(10, 8) =	11000.00000000	*	STRSMN(10, 8) =
STRSMX(10, 9) =	11000.00000000	*	STRSMN(10, 9) =
STRSMX(10,10) =	11000.00000000	*	STRSMN(10,10) =
STRSMX(10,11) =	11000.00000000	*	STRSMN(10,11) =
STRSMX(11, 7) =	10000.00000000	*	STRSMN(11, 7) =
STRSMX(11, 8) =	10000.00000000	*	STRSMN(11, 8) =
STRSMX(11, 9) =	10000.00000000	*	STRSMN(11, 9) =
STRSMX(11,10) =	10000.00000000	*	STRSMN(11,10) =
STRSMX(12, 7) =	9000.00000000	*	STRSMN(12, 7) =
STRSMX(12, 9) =	9000.00000000	*	STRSMN(12, 9) =
STRSMX(13, 4) =	8000.00000000	*	STRSMN(13, 4) =
STRSMX(13, 7) =	8000.00000000	*	STRSMN(13, 7) =

THE INPUTTED ARRAYS: **********

4000.00000000	*	ALPHA(5, 5)	=	0.00002000
3000.00000000	*	ALPHA(6, 4)	=	0.00002000
4000.00000000	*	ALPHA(6, 5)	=	0.00004000
5000.00000000	*	ALPHA(6, 6)	=	0.00002000
6000.00000000	*	ALPHA(6, 7)	=	0.00004000
7000.00000000	*	ALPHA(6, 8)	=	0.00002000
3000.00000000	*	ALPHA(7, 4)	=	0.00001000
4000.00000000	*	ALPHA(7, 5)	=	0.00003000
5000.00000000	*	ALPHA(7, 6)	=	0.00019000
6000.00000000	*	ALPHA(7, 7)	=	0.00026000
7000.00000000	*	ALPHA(7, 8)	=	0.00014000
8000.00000000	*	ALPHA(7, 9)	=	0.00006000
9000.00000000	*	ALPHA(7,10)	=	0.00001000
4000.00000000	*	ALPHA(8, 5)	=	0.00002000
5000.00000000	*	ALPHA(8, 6)	=	0.00017000
6000.00000000	*	ALPHA(8, 7)	=	0.00067000
7000.00000000	*	ALPHA(8, 8)	=	0.00174000
8000.00000000	*	ALPHA(8, 9)	=	0.00116000
9000.00000000	*	ALPHA(8,10)	=	0.00018000
10000.00000000	*	ALPHA(8,11)	=	0.00001000
3000.00000000	*	ALPHA(9, 4)	=	0.00001000
4000.00000000	*	ALPHA(9, 5)	=	0.00001000
5000.00000000	*	ALPHA(9, 6)	=	0.00003000
6000.00000000	*	ALPHA(9, 7)	=	0.00047000
7000.00000000	*	ALPHA(9, 8)	=	0.00453000
8000.00000000	*	ALPHA(9, 9)	=	0.01629000
9000.00000000	*	ALPHA(9,10)	=	0.00775000
10000.00000000	*	ALPHA(9,11)	=	0.00013000
5000.00000000	*	ALPHA(10, 6)	=	0.00001000
6000.00000000	*	ALPHA(10, 7)	=	0.00009000
7000.00000000	*	ALPHA(10, 8)	=	0.00252000
8000.00000000	*	ALPHA(10, 9)	=	0.05463000
9000.00000000	*	ALPHA(10,10)	=	0.38179000
10000.00000000	*	ALPHA(10,11)	=	0.01415000
6000.00000000	*	ALPHA(11, 7)	=	0.00001000
7000.00000000	*	ALPHA(11, 8)	=	0.00007000
8000.00000000	*	ALPHA(11, 9)	=	0.01363000
9000.00000000	*	ALPHA(11,10)	=	0.49842000
6000.00000000	*	ALPHA(12, 7)	=	0.00001000
8000.00000000	*	ALPHA(12, 9)	=	0.00001000
3000.00000000	*	ALPHA(13, 4)	=	0.00001000
6000.00000000	*	ALPHA(13, 7)	=	0.00001000

FATIGUE CYCLES TO FAILURE FROM LOADED CAR ROAD SERVICE IS 1645466470.99

IF ONLY DAMAGE OCCURING DURING THE LOADED CONDITION IS CONSIDERED, THE FATIGUE LIFE IS 9186303.58 MILES

FATIGUE CYCLES TO FAILURE FROM EMPTY CAR ROAD SERVICE IS 0.0

IF ONLY DAMAGE OCCURING DURING THE EMPTY CONDITION IS CONSIDERED, THE FATIGUE LIFE IS INFINITE.

```
**********
THE TOTAL EXPECTED FATIGUE LIFE IS   0.918630D+07 MILES
**********
```

A.3 DEC VERSION

A.3.1 FLAP Input Description

Utilizing the input described in Section A.1, a data file can be created for the input of a load spectra. This file may already exist in a different format and use of the CONVRT routine (see Sections 3.3.4 and 4.2) will reformat it for FLAP input standards. Input from a storage medium can be bypassed, but there must then be input by hand.

Input for the sample problem resides in a storage medium and is already in the proper format for FLAP input (Figure A.4).

Options chosen and engineering values for variables are the same as in the IBM input description (see Section A.2.1).

User-Program conversation is shown in Figure A.5.

Figure A.4

LOAD SPECTRA SUITABLE FOR INPUT TO FLAP

```
@TYPE FLDAT.DAT
1 5 50.00.0 .00002000
1 6 40.00.0 .00002000
1 6 50.00.0 .00004000
1 6 60.00.0 .00002000
1 6 70.00.0 .00004000
1 6 80.00.0 .00002000
1 7 40.00.0 .00001000
1 7 50.00.0 .00003000
1 7 60.00.0 .00019000
1 7 70.00.0 .00026000
1 7 80.00.0 .00014000
1 7 90.00.0 .00006000
1 7100.00.0 .00001000
1 8 50.00.0 .00002000
1 8 60.00.0 .00017000
1 8 70.00.0 .00067000
1 8 80.00.0 .00174000
1 8 90.00.0 .00116000
1 8100.00.0 .00018000
1 8110.00.0 .00001000
1 9 40.00.0 .00001000
1 9 50.00.0 .00001000
1 9 60.00.0 .00003000
1 9 70.00.0 .00047000
1 9 80.00.0 .00453000
1 9 90.00.0 .01629000
1 9100.00.0 .00775000
1 9110.00.0 .00013000
110 60.00.0 .00001000
110 70.00.0 .00009000
110 80.00.0 .00252000
110 90.00.0 .05463000
110100.00.0 .38179000
110110.00.0 .01415000
111 70.00.0 .00001000
111 80.00.0 .00007000
111 90.00.0 .01363000
111100.00.0 .49842000
112 70.00.0 .00001000
112 90.00.0 .00001000
113 40.00.0 .00001000
113 70.00.0 .00001000
00000000000000
@
```

Figure A.5

USER'S PROGRAM CONVERSATION

```
@
@RUN FLAP

CONVERT REPOS TO FLAP FORM(1=YES)?
0

GIVE THE OUTPUT DEVICE NUMBER:
3

ENTER JDIM,JG,JCALL,JHIT,JOR,JEMPTY,JQ,JNT
20,1,1,0,1,1,0,1

ENTER BETA,B,M,K,NE,BETAIM,EOVERL,CBW,GW,G,SCF,STATIC,STRE,FOE,PUNCH
354.661,8000,1,.32,2000000,0,.98,90000,240000,.1,0,10000,0,0,0

ARE YOU INPUTTING DATA BY HAND (1=YES)?
0

GIVE FILE WHERE DATA IS LOCATED:
FLDAT

TOTAL EXPECTED FATIGUE LIFE      0.920204E+07 MILES
STOP

END OF EXECUTION
CPU TIME: 1.26
EXIT
@
```

A.3.2 FLAP Listing - DEC

```
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(75,75),STRSMN(75,75),ALPHA(75,75)
      DIMENSION IMSTMX(75,75),IMSTMN(75,75),IMALF(75,75)
      DIMENSION EMSTMX(75,75),EMSTMN(75,75),EMALF(75,75)
      DIMENSION NT1(3),NT2(3),NT3(3)
      COMMON/WRT/JDEV
      NTLOAD=0.0
      NTEMP=0.0
      NTIMP=0.0
      JAR=1
      JIFF=1
      JEFF=2
      JYFF=3
      TYPE 106
106   FORMAT(' CONVERT REPOS TO FLAP FORM(1=YES)?')
      ACCEPT *,ICNVRT
      IF(ICNVRT.EQ.1) CALL CONVRT
      TYPE 137
137   FORMAT(' GIVE THE OUTPUT DEVICE NUMBER:')
      ACCEPT *,JDEV
      TYPE 1001
 1001 FORMAT(' ENTER JDIM,JG,JCALL,JHIT,JOR,JEMPTY,JQ,JNT')
   23  ACCEPT *,JDIM,JG,JCALL,JHIT,JOR,JEMPTY,JQ,JNT
      WRITE(JDEV,915) JDIM,JG,JCALL,JHIT,JOR,JEMPTY,JQ,JNT
  915 FORMAT('1',' JDIM =',I6,//,' JG   =',I6,//,' JCALL = ',I6,//,
     1' JHIT   =',I6,//,' JOR    =',I6,//,' JEMPTY =',I6,//,' JQ     =',
     2I6,//,' JNT    =',I3)
      TYPE 1002
1002  FORMAT(' ENTER BETA,B,M,K,NE,BETAIM,EOVERL,CBW,GW,G,SCF,STATIC,
     1STRE,FOE,PUNCH')
      ACCEPT *,BETA,B,M,K,NE,BETAIM,EOVERL,CBW,GW,G,SCF,STATIC,
     1STRE,FOE,PUNCH
      WRITE(JDEV,32) BETA,B,M,K,NE,BETAIM,EOVERL,CBW,GW,G
   32 FORMAT ('1',' TOTAL NUMBER OF SPECTRUM CYCLES PER MILE = ',F20.8,
     1//,' THE FATIGUE LIMIT FOR THE STRESS RATIO R =0, Y-INTERCEPT ON M
     2ODIFIED GOODMAN DIAGRAM CURVE = ',F20.8,//,' THE SLOPE OF THE MODI
     3FIED GOODMAN DIAGRAM CURVE = ',F20.8,//,' THE ABSOLUTE VALUE OF TH
     4E SLOPE OF THE S-N CURVE = ',F20.8,//,' THE NUMBER OF CYCLES AT TH
     5E FATIGUE LIMIT = ',F20.8,//,' THE TOTAL NUMBER OF SPECTRUM CYCLES
     6 PER MILE DUE TO IMPACT = ',F20.8,//,' THE RATIO OF EMPTY MILE USA
     7GE TO LOADED MILE USAGE = ',F20.8,//,' THE CAR BODY WEIGHT = ',
     8F20.8,//,' THE GROSS WEIGHT = ',F20.8,//,' THE NUMBER OF G S ACCEL
     9ERATION ENCOUNTERED DURING USAGE = ',F20.8,//)
      WRITE(JDEV,2) SCF,STATIC,STRE,FOE,PUNCH
    2 FORMAT (' THE VALUE OF THE STRESS CONCENTRATION FACTOR = ',F20.8,
     1//,' THE STATIC 1G STRESS DUE TO DEAD AND LIVE LOADS = ',F20.8,//,
     2' FOR CONVERSION REQUIREMENTS A STRESS OF ',F20.8,' IS EXPERIENCED
     3 WHEN A LOAD OF ',F20.8,' IS APPLIED',//,' THE IMPACT LOAD FACTOR
     4= ',F20.8)
      BETALD=BETA/(1+EOVERL)
      BETAEM=BETA-BETALD
      IF (JG.EQ.0) GO TO 111
      CALL GMULTY (JDIM,STRSMX,STRSMN,ALPHA,G,SCF,STATIC,STRE,FOE,JOR)
```

```
111 CALL LOADED (JDIM,JG,STRSMX,STRSMN,ALPHA,NTLOAD,B,M,NE,K)
    IF (JNT.NE.1) CALL ADDNT (NTLOAD,JAR,JIFF,NT1,NT2,NT3)
 15 IF (JCALL.EQ.0) GO TO 16
    CALL EMPTY (JDIM,JEMPTY,CBW,GW,NTEMP,STRSMX,STRSMN,ALPHA,B,M,NE,
   1K,EMSTMX,EMSTMN,EMALF)
    IF (JNT.NE.1) CALL ADDNT (NTEMP,JAR,JEFF,NT1,NT2,NT3)
 16 IF (JHIT.EQ.0) GO TO 18
    CALL IMPACT (JDIM,JHIT,NTIMP,STRSMX,STRSMN,ALPHA,B,M,NE,K,JQ,
   1IMSTMX,IMSTMN,IMALF,PUNCH)
    IF (JNT.NE.1) CALL ADDNT (NTIMP,JAR,JYFF,NT1,NT2,NT3)
 18 IF (JNT.EQ.JAR) GO TO 24
    JAR=JAR+1
    GO TO 23
 24 WRITE(JDEV,41) NTLOAD
 41 FORMAT ('1',' FATIGUE CYCLES TO FAILURE FROM LOADED CAR ROAD
   1 SERVICE IS ',F16.2,/)
    WLIFE=NTLOAD/BETALD
    IF (WLIFE.NE.0.0) GO TO 26
    WRITE(JDEV,21)
 21 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE LOADED CONDITION IS
   1 CONSIDERED, THE FATIGUE LIFE IS INFINITE.',////)
    GO TO 60
 26 WRITE(JDEV,42) WLIFE
 42 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE LOADED CONDITION IS
   1 CONSIDERED, THE FATIGUE LIFE IS ',F16.2,' MILES',////)
 60 IF (JCALL.EQ.0) GO TO 61
    WRITE(JDEV,43) NTEMP
 43 FORMAT ('1',' FATIGUE CYCLES TO FIALURE FROM EMPTY CAR ROAD SERVIC
   1E IS ',F16.2,/)
    WLIFE=NTEMP/BETAEM
    IF (WLIFE.NE.0.0) GO TO 22
    WRITE(JDEV,25)
 25 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE EMPTY CONDITION IS
   1 CONSIDERED, THE FATIGUE LIFE IS INFINITE.',////)
    GO TO 61
 22 WRITE(JDEV,44) WLIFE
 44 FORMAT (' IF ONLY DAMAGE OCCURING DURING THE EMPTY CONDITION IS
   1 CONSIDERED, THE FATIGUE LIFE IS ',F16.2,' MILES',////)
 61 IF (JHIT.EQ.0) GO TO 62
    WRITE(JDEV,49) NTIMP
 49 FORMAT ('1',' FATIGUE CYCLES TO FAILURE FROM YARD OPERATIONS IS',
   1F16.2,/)
    WLIFE=NTIMP/BETAIM
    IF (WLIFE.NE.0.0) GO TO 28
    WRITE(JDEV,27)
 27 FORMAT (' IF ONLY DAMAGE RESULTING FROM IMPACT IS CONSIDERED, THE
   1 FATIGUE LIFE IS INFINITE.',////)
    GO TO 62
 28 WRITE(JDEV,40) WLIFE
 40 FORMAT (' IF ONLY DAMAGE RESULTING FROM IMPACT IS CONSIDERED, THE
   1 FATIGUE LIFE IS ',F16.2,' MILES',////)
 62 IF (NTLOAD.EQ.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.EQ.0.0) GO TO 11
    IF (NTLOAD.NE.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.EQ.0.0)
   1LIFE=NTLOAD/BETALD
```

```
      IF (NTLOAD.EQ.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.EQ.0.0)
     1LIFE=NTEMP/BETAEM
      IF (NTLOAD.EQ.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.NE.0.0)
     1LIFE=NTIMP/BETAIM
      IF (NTLOAD.NE.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.EQ.0.0)
     1LIFE=1/(BETALD/NTLOAD+BETAEM/NTEMP)
      IF (NTLOAD.NE.0.0.AND.NTEMP.EQ.0.0.AND.NTIMP.NE.0.0)
     1LIFE=1/(BETALD/NTLOAD+BETAIM/NTIMP)
      IF (NTLOAD.EQ.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.NE.0.0)
     1LIFE=1/(BETAEM/NTEMP+BETAIM/NTIMP)
      IF (NTLOAD.NE.0.0.AND.NTEMP.NE.0.0.AND.NTIMP.NE.0.0)
     1LIFE=1/(BETALD/NTLOAD+BETAEM/NTEMP+BETAIM/NTIMP)
      TYPE 1010, LIFE
 1010 FORMAT(' TOTAL EXPECTED FATIGUE LIFE   ',E14.6,' MILES ')
      WRITE(JDEV,10) LIFE
   10 FORMAT ('1',' **********',//,' THE TOTAL EXPECTED FATIGUE LIFE IS
     1',E14.6,' MILES',//,' **********')
      GO TO 13
   11 WRITE(JDEV,12)
   12 FORMAT (' **********',//,' NO DAMAGE',//,' **********')
   13 STOP
      END
@
```

```
      SUBROUTINE LOADED (JDIM,JG,STRSMX,STRSMN,ALPHA,NTLOAD,B,M,NE,K)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      COMMON/WRT/JDEV
      IF (JG.EQ.1) GO TO 200
      DO 3 JA=1,JDIM
      DO 2 JB=1,JDIM
      STRSMX(JA,JB)=0.0
      STRSMN(JA,JB)=0.0
      ALPHA(JA,JB)=0.0
    2 CONTINUE
    3 CONTINUE
      TYPE 1003
 1003 FORMAT(' ENTER J,JA,JB,SMAX,SMIN,ALF ')
    5 ACCEPT *, J,JA,JB,SMAX,SMIN,ALF
      IF (J.EQ.0) GO TO 200
      STRSMX(JA,JB)=SMAX
      STRSMN(JA,JB)=SMIN
      ALPHA(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
  200 NT=0.0
      WRITE (JDEV,49)
   49 FORMAT ('1'//'          ********************          THE FOLLOW',
     1'ING ARE THE INPUT ARRAYS:              ********************'//)
      DO 76 JA=1,JDIM
      DO 75 JB=1,JDIM
      IF (ALPHA(JA,JB).EQ.0.0) GO TO 75
      WRITE (JDEV,77) JA,JB,STRSMX(JA,JB),JA,JB,STRSMN(JA,JB),JA,JB,
     1ALPHA(JA,JB)
   77 FORMAT (' STRSMX(',I2,',',I2,') =',F20.8,'   *  STRSMN(',I2,',',I2,
     1') =',F20.8,'   *  ALPHA(',I2,',',I2,') =',F20.8)
   75 CONTINUE
   76 CONTINUE
      DO 9 JA=1,JDIM
      DO 8 JB=1,JDIM
      IF (STRSMX(JA,JB).LE.0.0) GO TO 8
      IF (ALPHA(JA,JB).EQ.0.0) GO TO 8
      STRESS=STRSMN(JA,JB)
      IF (STRESS.LT.0.0) STRESS=-(STRESS)
      IF (STRESS.LT.STRSMX(JA,JB)) GO TO 7
      STRSMX(JA,JB)=(STRESS+STRSMX(JA,JB))/2.0
      STRSMN(JA,JB)=-(STRSMX(JA,JB))
    7 R=STRSMN(JA,JB)/STRSMX(JA,JB)
      SE=B/(1-M*R)
      IF(SE.GT.STRSMX(JA,JB))GO TO 8
      NI=NE/((STRSMX(JA,JB)/SE)**(1/K))
      X=ALPHA(JA,JB)/NI
      NT=X+NT
    8 CONTINUE
    9 CONTINUE
      IF (NT.EQ.0.0) GO TO 13
      NTLOAD=1/NT
   13 RETURN
      END
```

```
      SUBROUTINE EMPTY (JDIM,JEMPTY,CBW,GW,NTEMP,STRSMX,STRSMN,ALPHA,
     1B,M,NE,K,EMSTMX,EMSTMN,EMALF)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      DIMENSION EMSTMX(JDIM,JDIM),EMSTMN(JDIM,JDIM),EMALF(JDIM,JDIM)
      COMMON/WRT/JDEV
      NT=0.0
      DO 3 JA=1,JDIM
      DO 7 JB=1,JDIM
      EMSTMX(JA,JB)=0.0
      EMSTMN(JA,JB)=0.0
      EMALF(JA,JB)=0.0
    7 CONTINUE
    3 CONTINUE
      IF (JEMPTY.EQ.1) GO TO 1
      TYPE 1006
 1006 FORMAT('  INPUT J,JA,JB,SMAX,SMIN,ALF ')
    5  ACCEPT *, J,JA,JB,SMAX,SMIN,ALF
      IF (J.EQ.0) GO TO 200
      EMSTMX(JA,JB)=SMAX
      EMSTMN(JA,JB)=SMIN
      EMALF(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
  200 WRITE (JDEV,45)
   45 FORMAT('1'//'          ********************          THE FOLLOW',
     1'ING ARE THE INPUT ARRAYS:              ********************'//)
      DO 76 JA=1,JDIM
      DO 75 JB=1,JDIM
      IF (EMALF(JA,JB).EQ.0.0) GO TO 75
      WRITE (JDEV,77) JA,JB,EMSTMX(JA,JB),JA,JB,EMSTMN(JA,JB),JA,JB,
     1EMALF(JA,JB)
   77 FORMAT (' EMSTMX(',I2,',',I2,') =',F20.8,'      EMSTMN(',I2,',',I2,
     1') =',F20.8,'      EMALF(',I2,',',I2,') =',F20.8)
   75 CONTINUE
   76 CONTINUE
      DO 9 JA=1,JDIM
      DO 8 JB=1,JDIM
      IF (EMSTMX(JA,JB).LE.0.0) GO TO 8
      IF (EMALF(JA,JB).EQ.0.0) GO TO 8
      STRESS=EMSTMN(JA,JB)
      IF (STRESS.LT.0.0) STRESS=-(STRESS)
      IF (STRESS.LT.EMSTMX(JA,JB)) GO TO 8
      EMSTMX(JA,JB)=(STRESS+EMSTMX(JA,JB))/2
      EMSTMN(JA,JB)=-(EMSTMX(JA,JB))
    8 CONTINUE
    9 CONTINUE
      GO TO 2
    1 DO 49 JA=1,JDIM
      DO 94 JB=1,JDIM
      EMSTMX(JA,JB)=STRSMX(JA,JB)*CBW/GW
      EMSTMN(JA,JB)=STRSMN(JA,JB)*CBW/GW
      EMALF(JA,JB)=ALPHA(JA,JB)
   94 CONTINUE
```

```
49 CONTINUE
 2 DO 34 JA=1,JDIM
   DO 43 JB=1,JDIM
   IF (EMSTMX(JA,JB).LE.0.0) GO TO 43
   IF (EMALF(JA,JB).EQ.0.0) GO TO 43
   R=EMSTMN(JA,JB)/EMSTMX(JA,JB)
   SE=B/(1-M*R)
   IF (SE.GT.EMSTMX(JA,JB)) GO TO 43
   NI=NE/((EMSTMX(JA,JB)/SE)**(1/K))
   X=EMALF(JA,JB)/NI
   NT=X+NT
43 CONTINUE
34 CONTINUE
   IF (NT.EQ.0.0) GO TO 13
   NTEMP=1/NT
13 RETURN
   END
@
```

```
      SUBROUTINE IMPACT (JDIM,JHIT,NTIMP,STRSMX,STRSMN,ALPHA,B,M,NE,K,
     1JQ,IMSTMX,IMSTMN,IMALF,PUNCH)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      DIMENSION IMSTMX(JDIM,JDIM),IMSTMN(JDIM,JDIM),IMALF(JDIM,JDIM)
      COMMON/WRT/JDEV
      NT=0.0
      DO 3 JA=1,JDIM
      DO 7 JB=1,JDIM
      IMSTMX(JA,JB)=0.0
      IMSTMN(JA,JB)=0.0
      IMALF(JA,JB)=0.0
    7 CONTINUE
    3 CONTINUE
      IF (JQ.EQ.1) GO TO 1
      TYPE 1007
 1007 FORMAT(' INPUT J,JA,JB,SMAX,SMIN,ALF ')
    5 ACCEPT *, J,JA,JB,SMAX,SMIN,ALF
      IF (J.EQ.0) GO TO 200
      IMSTMX(JA,JB)=SMAX
      IMSTMN(JA,JB)=SMIN
      IMALF(JA,JB)=ALF
      IF (J.EQ.1) GO TO 5
  200 WRITE (JDEV,45)
   45 FORMAT('1'//'          ********************          THE FOLLOW',
     1'ING ARE THE INPUT ARRAYS:              ********************'//)
      DO 76 JA=1,JDIM
      DO 75 JB=1,JDIM
      IF (IMALF(JA,JB).EQ.0.0) GO TO 75
      WRITE (JDEV,77) JA,JB,IMSTMX(JA,JB),JA,JB,IMSTMN(JA,JB),JA,JB,
     1IMALF(JA,JB)
   77 FORMAT (' IMSTMX(',I2,',',I2,') =',F20.8,'      IMSTMN(',I2,',',I2,
     1') =',F20.8,'      IMALF(',I2,',',I2,') =',F20.8)
   75 CONTINUE
   76 CONTINUE
      DO 9 JA=1,JDIM
      DO 8 JB=1,JDIM
      IF (IMSTMX(JA,JB).LE.0.0) GO TO 8
      IF (IMALF(JA,JB).EQ.0.0) GO TO 8
      STRESS=IMSTMN(JA,JB)
      IF (STRESS.LT.0.0) STRESS=-(STRESS)
      IF (STRESS.LT.IMSTMX(JA,JB)) GO TO 8
      IMSTMX(JA,JB)=(STRESS+IMSTMX(JA,JB))/2
      IMSTMN(JA,JB)=-(IMSTMX(JA,JB))
    8 CONTINUE
    9 CONTINUE
      GO TO 2
    1 DO 49 JA=1,JDIM
      DO 94 JB=1,JDIM
      IMSTMX(JA,JB)=STRSMX(JA,JB)*PUNCH
      IMSTMN(JA,JB)=STRSMN(JA,JB)*PUNCH
      IMALF(JA,JB)=ALPHA(JA,JB)
   94 CONTINUE
```

```
49 CONTINUE
 2 DO 34 JA=1,JDIM
   DO 43 JB=1,JDIM
   IF (IMSTMX(JA,JB).LE.0.0) GO TO 43
   IF (IMALF(JA,JB).EQ.0.0) GO TO 43
   R=IMSTMN(JA,JB)/IMSTMX(JA,JB)
   SE=B/(1-M*R)
   IF (SE.GT.IMSTMX(JA,JB)) GO TO 43
   NI=NE/((IMSTMX(JA,JB)/SE)**(1/K))
   X=IMALF(JA,JB)/NI
   NT=X+NT
43 CONTINUE
34 CONTINUE
   IF (NT.EQ.0.0) GO TO 13
   NTIMP=1/NT
13 RETURN
   END
@
```

```
      SUBROUTINE CONVRT
      DOUBLE PRECISION DATA,OUTPUT
      DIMENSION DATAM(20,20)
      TYPE 1
1     FORMAT(' GIVE INPUT FILE NAME FOR CONVERSION OF REPOS')
      TYPE 44
44    FORMAT('        TO INPUT FORM FOR FLAP')
      ACCEPT 2,DATA
2     FORMAT(A10)
      OPEN(UNIT=22,FILE=DATA)
      TYPE 3
3     FORMAT(' GIVE OUTPUT FILE NAME FOR CONVERTED REPOS')
      ACCEPT 2,OUTPUT
      OPEN(UNIT=20,FILE=OUTPUT)
      DO 10 I=1,20
      K=21-I
      READ(22,4)(DATAM(I,J),J=1,K)
4    FORMAT (1X,10F9.6/1X,10F9.6)
10    CONTINUE
      DO 20 I=1,20
      DO 20 J=1,20
      IF(DATAM(I,J).EQ.0.0)GO TO 20
      DATAM(I,J)=DATAM(I,J)/100.
      WRITE(20,5)I,J,DATAM(I,J)
5     FORMAT('1',2I2,'0.00.0',F10.8)
20    CONTINUE
      WRITE(20,6)
6     FORMAT('00000000000')
      CLOSE(UNIT=20)
      RETURN
      END
@
```

```
      SUBROUTINE GMULTY (JDIM,STRSMX,STRSMN,ALPHA,G,SCF,STATIC,STRE,FOE,
     1JOR)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION STRSMX(JDIM,JDIM),STRSMN(JDIM,JDIM),ALPHA(JDIM,JDIM)
      DOUBLE PRECISION DATA
      IF(JOR.EQ.1)GO TO 99
      TYPE 107
107   FORMAT(' GIVE MAXIMUM,INCREMENT')
      ACCEPT *,MAX,INC
      SMAX=MAX+INC
      DO 8 JA=1,JDIM
      SMAX=SMAX-INC
      SMIN=-(MAX+INC)
      DO 9 JB=1,JDIM
      SMIN=SMIN+INC
      STRSMX(JA,JB)=SMAX
      STRSMN(JA,JB)=SMIN
      ALPHA(JA,JB)=0.0
    9 CONTINUE
    8 CONTINUE
      GO TO 98
   99 DO 81 J=1,JDIM
      DO 81 K=1,JDIM
      STRSMX(J,K)=0.0
      STRSMN(J,K)=0.0
      ALPHA(J,K)=0.0
   81 CONTINUE
  98  TYPE 141
141   FORMAT(' ARE YOU INPUTTING DATA BY HAND (1=YES)?')
      ACCEPT *,JINP
      IF(JINP.EQ.1)GO TO 152
      TYPE 147
147   FORMAT(' GIVE FILE WHERE DATA IS LOCATED:')
      ACCEPT 189,DATA
189   FORMAT(A10)
      OPEN(UNIT=1,FILE=DATA)
5     READ(1,35) J,JA,JB,SMAX,SMIN,ALF
35    FORMAT(I1,2I2,2F3.1,F10.8)
      GO TO 149
152   TYPE 164
164   FORMAT(' ENTER J,JA,JB,SMAX,SMIN,ALF')
      ACCEPT *,J,JA,JB,SMAX,SMIN,ALF
149   IF (J.EQ.0) GO TO 6
      ALPHA(JA,JB)=ALF
      IF(JINP.EQ.1)GO TO 152
      IF(J.EQ.1)GO TO 5
    6 GO TO (7,14,20),JOR
    7 JZERO=JDIM/2
      DO 3 JA=1,JDIM
      DO 2 JB=1,JDIM
      IF (ALPHA(JA,JB).EQ.0.0) GO TO 2
      QUAN1=JZERO-JA+1
      QUAN2=JB-JZERO-1
```

```
      STRSMX(JA,JB)=G*QUAN1*STATIC+STATIC
      STRSMN(JA,JB)=G*QUAN2*STATIC+STATIC
      IX=STRSMX(JA,JB)
      IN=STRSMN(JA,JB)
      RIX=IX
      RIN=IN
      RX=STRSMX(JA,JB)-RIX
      RN=STRSMN(JA,JB)-RIN
      IF (RX.GE.0.5) GO TO 10
      STRSMX(JA,JB)=RIX
      GO TO 11
   10 STRSMX(JA,JB)=RIX+1.0
   11 IF (RN.GE.0.5) GO TO 12
      STRSMN(JA,JB)=RIN
      GO TO 2
   12 STRSMN(JA,JB)=RIN+1.0
    2 CONTINUE
    3 CONTINUE
      GO TO 15
   14 DO 18 JA=1,JDIM
      DO 19 JB=1,JDIM
      STRSMX(JA,JB)=STRSMX(JA,JB)*SCF*STRE/FOE+STATIC
      STRSMN(JA,JB)=STRSMN(JA,JB)*SCF*STRE/FOE+STATIC
   19 CONTINUE
   18 CONTINUE
      GO TO 15
   20 DO 21 JA=1,JDIM
      DO 22 JB=1,JDIM
      STRSMX(JA,JB)=STRSMX(JA,JB)*STRE/FOE
      STRSMN(JA,JB)=STRSMN(JA,JB)*STRE/FOE
   22 CONTINUE
   21 CONTINUE
   15 RETURN
      END
@
```

```
      SUBROUTINE ADDNT (NT,JAR,JIFF,NT1,NT2,NT3)
      IMPLICIT REAL*8 (A-I,K-Z)
      DIMENSION NT1(3),NT2(3),NT3(3)
      IF (JIFF.NE.1) GO TO 1
      NT1(JAR)=NT
      IF (JAR.EQ.1) RETURN
      IF (JAR.NE.2) GO TO 4
      IF (NT1(1).EQ.0.0.AND.NT1(2).EQ.0.0) NT=0.0
      IF (NT1(1).NE.0.0.AND.NT1(2).EQ.0.0) NT=NT1(1)
      IF (NT1(1).EQ.0.0.AND.NT1(2).NE.0.0) NT=NT1(2)
      IF (NT1(1).NE.0.0.AND.NT1(2).NE.0.0) NT=1/(1/NT1(1)+1/NT1(2))
    4 IF (JAR.NE.3) GO TO 1
      IF (NT1(1).EQ.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).EQ.0.0) NT=0.0
      IF (NT1(1).NE.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).EQ.0.0) NT=NT1(1)
      IF (NT1(1).EQ.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).EQ.0.0) NT=NT1(2)
      IF (NT1(1).EQ.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).NE.0.0) NT=NT1(3)
      IF (NT1(1).NE.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).EQ.0.0)
     1NT=1/(1/NT1(1)+1/NT1(2))
      IF (NT1(1).NE.0.0.AND.NT1(2).EQ.0.0.AND.NT1(3).NE.0.0)
     1NT=1/(1/NT1(1)+1/NT1(3))
      IF (NT1(1).EQ.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).NE.0.0)
     1NT=1/(1/NT1(2)+1/NT1(3))
      IF (NT1(1).NE.0.0.AND.NT1(2).NE.0.0.AND.NT1(3).NE.0.0)
     1NT=1/(1/NT1(1)+1/NT1(2)+1/NT1(3))
    1 IF (JIFF.NE.2) GO TO 2
      NT2(JAR)=NT
      IF (JAR.EQ.1) RETURN
      IF (JAR.NE.2) GO TO 5
      IF (NT2(1).EQ.0.0.AND.NT2(2).EQ.0.0) NT=0.0
      IF (NT2(1).NE.0.0.AND.NT2(2).EQ.0.0) NT=NT2(1)
      IF (NT2(1).EQ.0.0.AND.NT2(2).NE.0.0) NT=NT2(2)
      IF (NT2(1).NE.0.0.AND.NT2(2).NE.0.0) NT=1/(1/NT2(1)+1/NT2(2))
    5 IF (JAR.NE.3) GO TO 2
      IF (NT2(1).EQ.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).EQ.0.0) NT=0.0
      IF (NT2(1).NE.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).EQ.0.0) NT=NT2(1)
      IF (NT2(1).EQ.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).EQ.0.0) NT=NT2(2)
      IF (NT2(1).EQ.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).NE.0.0) NT=NT2(3)
      IF (NT2(1).NE.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).EQ.0.0)
     1NT=1/(1/NT2(1)+1/NT2(2))
      IF (NT2(1).NE.0.0.AND.NT2(2).EQ.0.0.AND.NT2(3).NE.0.0)
     1NT=1/(1/NT2(1)+1/NT2(3))
      IF (NT2(1).EQ.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).NE.0.0)
     1NT=1/(1/NT2(2)+1/NT2(3))
      IF (NT2(1).NE.0.0.AND.NT2(2).NE.0.0.AND.NT2(3).NE.0.0)
     1NT=1/(1/NT2(1)+1/NT2(2)+1/NT2(3))
    2 IF (JIFF.NE.3) GO TO 3
      NT3(JAR)=NT
      IF (JAR.EQ.1) RETURN
      IF (JAR.NE.2) GO TO 6
      IF (NT3(1).EQ.0.0.AND.NT3(2).EQ.0.0) NT=0.0
      IF (NT3(1).NE.0.0.AND.NT3(2).EQ.0.0) NT=NT3(1)
      IF (NT3(1).EQ.0.0.AND.NT3(2).NE.0.0) NT=NT3(2)
      IF (NT3(1).NE.0.0.AND.NT3(2).NE.0.0) NT=1/(1/NT3(1)+1/NT3(2))
```

```
6 IF (JAR.NE.3) GO TO 3
  IF (NT3(1).EQ.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).EQ.0.0) NT=0.0
  IF (NT3(1).NE.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).EQ.0.0) NT=NT3(1)
  IF (NT3(1).EQ.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).EQ.0.0) NT=NT3(2)
  IF (NT3(1).EQ.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).NE.0.0) NT=NT3(3)
  IF (NT3(1).NE.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).EQ.0.0)
 1NT=1/(1/NT3(1)+1/NT3(2))
  IF (NT3(1).NE.0.0.AND.NT3(2).EQ.0.0.AND.NT3(3).NE.0.0)
 1NT=1/(1/NT3(1)+1/NT3(3))
  IF (NT3(1).EQ.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).NE.0.0)
 1NT=1/(1/NT3(2)+1/NT3(3))
  IF (NT3(1).NE.0.0.AND.NT3(2).NE.0.0.AND.NT3(3).NE.0.0)
 1NT=1/(1/NT3(1)+1/NT3(2)+1/NT3(3))
3 RETURN
  END
```

@

A.3.3 FLAP Output - DEC

```
JDIM =    20
JG   =    1
JCALL =     1
JHIT  =     0
JOR   =     1
JEMPTY =    1
JQ    =     0
JNT   =  1
```

TOTAL NUMBER OF SPECTRUM CYCLES PER MILE = 354.66100000

THE FATIGUE LIMIT FOR THE STRESS RATIO R =0, Y-INTERCEPT ON MODIFIED GOODMAN DIAGRAM CURVE = 8000.00000000

THE SLOPE OF THE MODIFIED GOODMAN DIAGRAM CURVE = 0.99900000

THE ABSOLUTE VALUE OF THE SLOPE OF THE S-N CURVE = 0.32000000

THE NUMBER OF CYCLES AT THE FATIGUE LIMIT = 2000000.00000000

THE TOTAL NUMBER OF SPECTRUM CYCLES PER MILE DUE TO IMPACT = 0.00000000

THE RATIO OF EMPTY MILE USAGE TO LOADED MILE USAGE = 0.98000000

THE CAR BODY WEIGHT = 90000.00000000

THE GROSS WEIGHT = 240000.00000000

THE NUMBER OF G S ACCELERATION ENCOUNTERED DURING USAGE = 0.10000000

THE VALUE OF THE STRESS CONCENTRATION FACTOR = 0.00000000

THE STATIC 1G STRESS DUE TO DEAD AND LIVE LOADS = 10000.00000000

FOR CONVERSION REQUIREMENTS A STRESS OF 0.00000000 IS EXPERIENCED WHEN A LOAD OF 0.00000000 IS APPLIED

THE IMPACT LOAD FACTOR = 0.00000000

********** THE FOLLOWING ARE THE INPUTTED ARRAYS: **********

STRSMX(5, 5) =	16000.00000000	*	STRSMN(5, 5) =	4000.00000000	*	ALPHA(5, 5) =	0.00002000
STRSMX(6, 4) =	15000.00000000	*	STRSMN(6, 4) =	3000.00000000	*	ALPHA(6, 4) =	0.00002000
STRSMX(6, 5) =	15000.00000000	*	STRSMN(6, 5) =	4000.00000000	*	ALPHA(6, 5) =	0.00004000
STRSMX(6, 6) =	15000.00000000	*	STRSMN(6, 6) =	5000.00000000	*	ALPHA(6, 6) =	0.00002000
STRSMX(6, 7) =	15000.00000000	*	STRSMN(6, 7) =	6000.00000000	*	ALPHA(6, 7) =	0.00004000
STRSMX(6, 8) =	15000.00000000	*	STRSMN(6, 8) =	7000.00000000	*	ALPHA(6, 8) =	0.00002000
STRSMX(7, 4) =	14000.00000000	*	STRSMN(7, 4) =	3000.00000000	*	ALPHA(7, 4) =	0.00001000
STRSMX(7, 5) =	14000.00000000	*	STRSMN(7, 5) =	4000.00000000	*	ALPHA(7, 5) =	0.00003000
STRSMX(7, 6) =	14000.00000000	*	STRSMN(7, 6) =	5000.00000000	*	ALPHA(7, 6) =	0.00019000
STRSMX(7, 7) =	14000.00000000	*	STRSMN(7, 7) =	6000.00000000	*	ALPHA(7, 7) =	0.00026000
STRSMX(7, 8) =	14000.00000000	*	STRSMN(7, 8) =	7000.00000000	*	ALPHA(7, 8) =	0.00014000
STRSMX(7, 9) =	14000.00000000	*	STRSMN(7, 9) =	8000.00000000	*	ALPHA(7, 9) =	0.00006000
STRSMX(7,10) =	14000.00000000	*	STRSMN(7,10) =	9000.00000000	*	ALPHA(7,10) =	0.00001000
STRSMX(8, 5) =	13000.00000000	*	STRSMN(8, 5) =	4000.00000000	*	ALPHA(8, 5) =	0.00002000
STRSMX(8, 6) =	13000.00000000	*	STRSMN(8, 6) =	5000.00000000	*	ALPHA(8, 6) =	0.00017000
STRSMX(8, 7) =	13000.00000000	*	STRSMN(8, 7) =	6000.00000000	*	ALPHA(8, 7) =	0.00067000
STRSMX(8, 8) =	13000.00000000	*	STRSMN(8, 8) =	7000.00000000	*	ALPHA(8, 8) =	0.00174000
STRSMX(8, 9) =	13000.00000000	*	STRSMN(8, 9) =	8000.00000000	*	ALPHA(8, 9) =	0.00116000
STRSMX(8,10) =	13000.00000000	*	STRSMN(8,10) =	9000.00000000	*	ALPHA(8,10) =	0.00018000
STRSMX(8,11) =	13000.00000000	*	STRSMN(8,11) =	10000.00000000	*	ALPHA(8,11) =	0.00001000
STRSMX(9, 4) =	12000.00000000	*	STRSMN(9, 4) =	3000.00000000	*	ALPHA(9, 4) =	0.00001000
STRSMX(9, 5) =	12000.00000000	*	STRSMN(9, 5) =	4000.00000000	*	ALPHA(9, 5) =	0.00001000
STRSMX(9, 6) =	12000.00000000	*	STRSMN(9, 6) =	5000.00000000	*	ALPHA(9, 6) =	0.00003000
STRSMX(9, 7) =	12000.00000000	*	STRSMN(9, 7) =	6000.00000000	*	ALPHA(9, 7) =	0.00047000
STRSMX(9, 8) =	12000.00000000	*	STRSMN(9, 8) =	7000.00000000	*	ALPHA(9, 8) =	0.00453000
STRSMX(9, 9) =	12000.00000000	*	STRSMN(9, 9) =	8000.00000000	*	ALPHA(9, 9) =	0.01629000
STRSMX(9,10) =	12000.00000000	*	STRSMN(9,10) =	9000.00000000	*	ALPHA(9,10) =	0.00775000
STRSMX(9,11) =	12000.00000000	*	STRSMN(9,11) =	10000.00000000	*	ALPHA(9,11) =	0.00013000
STRSMX(10, 6) =	11000.00000000	*	STRSMN(10, 6) =	5000.00000000	*	ALPHA(10, 6) =	0.00001000
STRSMX(10, 7) =	11000.00000000	*	STRSMN(10, 7) =	6000.00000000	*	ALPHA(10, 7) =	0.00009000
STRSMX(10, 8) =	11000.00000000	*	STRSMN(10, 8) =	7000.00000000	*	ALPHA(10, 8) =	0.00252000
STRSMX(10, 9) =	11000.00000000	*	STRSMN(10, 9) =	8000.00000000	*	ALPHA(10, 9) =	0.05463000
STRSMX(10,10) =	11000.00000000	*	STRSMN(10,10) =	9000.00000000	*	ALPHA(10,10) =	0.38179000
STRSMX(10,11) =	11000.00000000	*	STRSMN(10,11) =	10000.00000000	*	ALPHA(10,11) =	0.01415000
STRSMX(11, 7) =	10000.00000000	*	STRSMN(11, 7) =	6000.00000000	*	ALPHA(11, 7) =	0.00001000
STRSMX(11, 8) =	10000.00000000	*	STRSMN(11, 8) =	7000.00000000	*	ALPHA(11, 8) =	0.00007000
STRSMX(11, 9) =	10000.00000000	*	STRSMN(11, 9) =	8000.00000000	*	ALPHA(11, 9) =	0.01363000
STRSMX(11,10) =	10000.00000000	*	STRSMN(11,10) =	9000.00000000	*	ALPHA(11,10) =	0.49842000
STRSMX(12, 7) =	9000.00000000	*	STRSMN(12, 7) =	6000.00000000	*	ALPHA(12, 7) =	0.00001000
STRSMX(12, 9) =	9000.00000000	*	STRSMN(12, 9) =	8000.00000000	*	ALPHA(12, 9) =	0.00001000
STRSMX(13, 4) =	8000.00000000	*	STRSMN(13, 4) =	3000.00000000	*	ALPHA(13, 4) =	0.00001000
STRSMX(13, 7) =	8000.00000000	*	STRSMN(13, 7) =	6000.00000000	*	ALPHA(13, 7) =	0.00001000

FATIGUE CYCLES TO FAILURE FROM LOADED CAR ROAD SERVICE IS 1645466470.99

IF ONLY DAMAGE OCCURING DURING THE LOADED CONDITION IS CONSIDERED, THE FATIGUE LIFE IS 9186303.58 MILES

FATIGUE CYCLES TO FAILURE FROM EMPTY CAR ROAD SERVICE IS 0.00

IF ONLY DAMAGE OCCURING DURING THE EMPTY CONDITION IS CONSIDERED, THE FATIGUE LIFE IS INFINITE.

```
**********
THE TOTAL EXPECTED FATIGUE LIFE IS   0.918630E+07 MILES
**********
```

APPENDIX B

RATIO OF EMPTY TO LOADED CAR-MILES - BY CAR TYPE*

(Time and mileage basis cars, shipper/private line cars, and per-diem free cars combined. Average for United States)

TYPE OF CAR/AAR CAR TYPE CODES	RATIO
BOX-GENERAL SERVICE (UNEQUIPPED) (ALL B, L070, R-00, R-01)	.62
BOX-GENERAL SERVICE (EQUIPPED) (A-20, A-30, A-40, A-50, R-06, R-07)	.78
BOX-SPECIAL SERVICE (A-00, A-10)	.98
GONDOLA-GENERAL SERVICE (ALL G (EXCEPT G-9-))	.82
GONDOLA-SPECIAL SERVICE (G-9-, J-00, ALL C, ALL E)	1.01
HOPPER (OPEN TOP)-GENERAL SERVICE (ALL H (EXCEPT H-701))	.90
HOPPER (OPEN TOP)-SPECIAL SERVICE (H-70, J-10, J-20, ALL K)	1.01
HOPPER (COVERED) (L-5-)	1.02
TANK, UNDER 11,999 GALLONS (T--0, T--1, T--2, T--3)	1.14
TANK, 12,000 - 18,999 GALLONS (T--4)	1.04
TANK, 19,000 - 24,999 GALLONS (T--5, T--6)	1.01
TANK, 25,000 GALLONS & UP (T--7, T--8, T--9)	1.12
REFRIGERATOR (MEAT)-MECHANICAL (R-11, R-12)	1.07
REFRIGERATOR (OTHER THAN MEAT)-MECHANICAL (R-04, R-10)	.72
REFRIGERATOR (MEAT)-NON-MECHANICAL (R-02, R-08, R-09, R-14, R-15, R-17)	1.16
REFRIGERATOR (OTHER THAN MEAT)-NON-MECHANICAL (R-03, R-05, R-13, R-16)	.89
STOCK (ALL S)	1.19
AUTORACK (F-5-, F-6-)	1.00
FLAT-GENERAL SERVICE (F-0-)	.88
FLAT-SPECIAL SERVICE (F-1-, F-9-, F-20, F-30, F-40, L-2-, L-3-)	.99

TYPE OF CAR/AAR CAR TYPE CODES	RATIO
FLAT-TOFC (F-7-, F-8-)	.45
ALL OTHER (L-0-, L-1-, L-4-, L080, L090)	.89

Note: * Listing obtained from Interstate Commerce Commission Statement No. 152-72, December 1973, "Ratio of Empty to Loaded Freight Car-Miles by Type of Car and Performance Factors for Way, Through and All Trains Combined - 1972".

Fatigue Life Analysis Program (FLAP) - Users' manual, 1979

AM Zarembski, V sobash

PROPERTY OF FRA
RESEARCH & DEVELOPMENT
LIBRARY