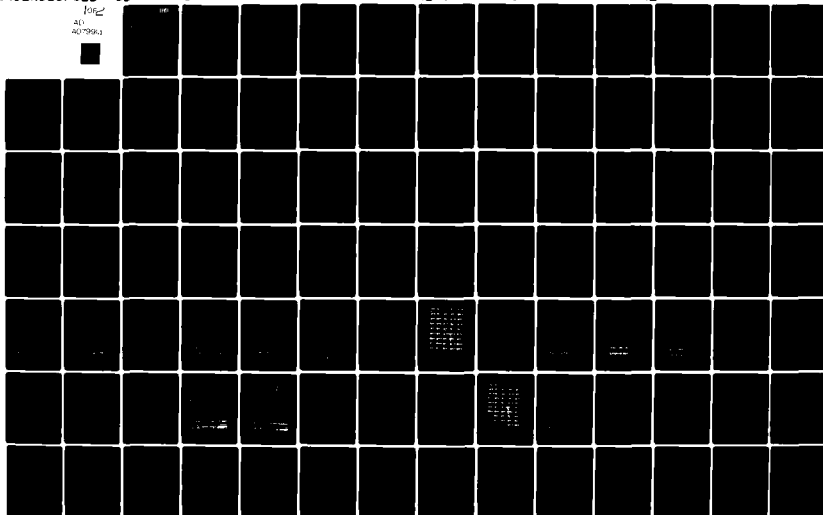


AO-A079 914

DAYTON UNIV OH RESEARCH INST F/G 9/2
COMPUTERIZED PROCESSING AND GRAPHIC REPRESENTATION OF VISCOELAS--ETC(U)
AUG 79 C S KING F33615-76-C-5137
UDR-TR-78-49 AFML-TR-79-4099 NL

UNCLASSIFIED

10/2
21
AO-79914



AFML-TR-79-4099

(2) LEVEL II

ADA079914

COMPUTERIZED PROCESSING AND GRAPHIC REPRESENTATION
OF VISCOELASTIC MATERIAL PROPERTY DATA

Carl S. King, Jr.
University of Dayton Research Institute
300 College Park Avenue
Dayton, Ohio 45469

August 1979

Interim Report for Period October 1977 through March 1978

Approved for public release; distribution
unlimited.

DDC FILE COPY

AIR FORCE MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

DDC
RECEIVED
JAN 30 1980
B

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

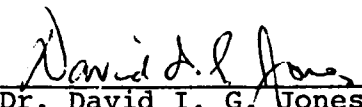
80 1 28 073

NOTICE

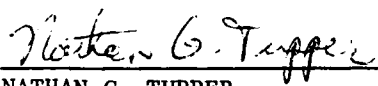
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

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



Dr. David I. G. Jones
Project Engineer



NATHAN G. TUPPER
Chief, Metals Behavior Branch
Metals and Ceramics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFML/LLN , W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. PERIODICITY'S CATALOG NUMBER	
18 AFML-TR-79-499			
4. TITLE (and Subtitle)		5. FUNDING NUMBERS	
6 COMPUTERIZED PROCESSING AND GRAPHIC REPRESENTATION OF VISCOELASTIC MATERIAL PROPERTY DATA		7. AUTHOR	
		10 UDR-TR-78-49	
8. AUTHOR		9. PERFORMING ORG. REPORT NUMBER	
10 Carl S. King, Jr		15 F33615-76-C-5137	
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
University of Dayton Research Institute 300 College Park Dayton, Ohio 45469		16 62102F 2418,735L 24180302	
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE	
Air Force Materials Laboratory Air Force Systems Command Wright-Patterson AFB, Ohio 45433		11 August 1979	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES	
12 109		150	
15. SECURITY CLASS. (of this report)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
Unclassified		N.A.	
16. DISTRIBUTION STATEMENT (of this Report)			
Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
Viscoelastic Material Properties Data Empirical Curve Equations Curve Parameters Computer Program Computer Graphics			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)			
A means of fitting viscoelastic material properties data to specified curves and graphing the data and curves adds tremen- dously to the technology advancements in the state of the art for data processing. A computer program is developed to accomplish the technology advancements for the viscoelastic material properties data.			

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

105400 Gue

FOREWORD

The work reported herein was performed by the University of Dayton Research Institute, Dayton, Ohio under Air Force Contract No. F33615-76-C-5137, Project Number 7351 "Metallic Materials," Task Number 735106, "Behavior of Metals", for the Air Force Materials Laboratory, Wright-Patterson Air Force Base, Ohio. The work described was conducted during the period October 1977 through March 1978 under the general supervision of Mr. D. H. Whitford, Supervisor of the Aerospace Mechanics Division, and Mr. M. L. Drake, Principal Investigator. The principal programmer was Mr. C. S. King, Jr.

The author acknowledges the secretaries and special thanks to Mr. M. L. Drake for the equipment and preparation of this report. Thanks are also extended to the Air Force Materials Laboratory to foresee the need for the application of this program.

ACCESSION for	
NTIS	White Section ☒
DDC	Buff Section ☐
UNANNOUNCED	☐
JUSIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist. AAIL. and/or SPECIAL	
A	23 C

TABLE OF CONTENTS

SECTION	PAGE
I INTRODUCTION	1
II DEVELOPMENT OF THEORY	3
2.1 Editing Procedure	3
2.2 Empirical Shift Parameter Equation	4
2.3 Empirical Modulus Equation	6
2.4 Empirical Loss Factor Equation	7
2.5 Parameter Constraints	8
III PROGRAM DEVELOPMENT	10
3.1 General Program Flow	10
3.2 Available Equipment	12
3.3 Type of Minimization	14
IV PROGRAM DESCRIPTION	16
4.1 Interactive Features	16
4.2 Data File Structure	17
4.3 Graphics Package	19
4.4 Minimization Technique	20
4.5 Initialization of Parameter Values	21
V SUBPROGRAM DESCRIPTION	23
5.1 Program VIBR2	23
5.2 Subroutine REDIN	24
5.3 Subroutine FITTO	25
5.4 Subroutine POLYF	25
5.5 Subroutine PLTTO	25
5.6 Subroutine PLTT1	26
5.7 Subroutine EDITO	26
5.8 Subroutine STORE	26

TABLE OF CONTENTS, continued

SECTION		PAGE
5.9	Subroutine MODUS	27
5.10	Subroutine ETA	27
5.11	Subroutine FCN	28
5.12	Function FCT	28
5.13	Function ALPHT	28
5.14	Function F1	29
5.15	Function F2	29
5.16	Function ALOG1	29
5.17	Subroutine VERTL	30
5.18	Subroutine SEARC	30
5.19	Subroutine GAXES	30
5.20	Subroutine PLTPK	31
5.21	Summary	31
VI	USER MANUAL WITH DESCRIPTIVE SAMPLE INPUT/OUTPUT	32
6.1	User Manual	32
6.2	Sample Input/Output	46
VII	SUMMARY AND RECOMMENDATIONS	72
REFERENCES		74
APPENDIX A	Parameter Name Descriptions	75
APPENDIX B	Program Listing	77

LIST OF FIGURES

FIGURE		PAGE
1	General Program Development	11
2	Detailed Program Diagram	13

SECTION I

INTRODUCTION

The motivation for this report was to develop a means for editing viscoelastic material data, predicting values for parameters of equations that closely describe the material data, plotting the material data, and storing the material data and parameters by use of a fast, dedicated digital computer in an interactive mode of operation. The availability of a computer program to accomplish the above operations would greatly decrease the time needed to characterize the viscoelastic material data. Up to this point in time, the above-mentioned material data operations had been accomplished manually, often taking many hours of calculations and plotting to represent the material data. A computer program could and would eliminate the need for the manual operations of calculating and plotting the data, leaving only the manual operations of providing input to the program and operation of the computer.

The analytical representations of viscoelastic material properties have evolved over many years. The first representation¹ displays graphically the data of the modulus and loss factor properties on a log-log scale with an independent variable called the reduced frequency. This variable combines the temperature into a factor that multiplies the frequency, the factor depending on the difference between the temperature and a suitably chosen reference temperature. A later representation displays these properties graphically on a log-log scale with an independent variable called the reduced temperature. The reference temperature is unique for each material and has been observed experimentally to be approximately equal to the temperature of peak damping for many materials.

Another representation³ displays the data as in the previous methods, and fits the properties data by use of representative curves. This method was the experimental data to determine a continuous analytical representation for the material properties.

This report will include descriptions of the detailed items that are features of the computer program. Also included are the equations used to curve fit the material data and a description of the curve-fitting routine. The plotting capabilities will be described, a user manual is provided, and output of the program is documented with notes to indicate the meaning of the input and output to the user. The listing of the program which performs the above operations on the viscoelastic material data is included as an appendix.

SECTION II

DEVELOPMENT OF THEORY

In order to facilitate more widespread utilization of viscoelastic damping material, it is desirable that an efficient, systematic, and standardized procedure be established to process, store, and retrieve the material properties. The approach adopted is to use equations³ which have been selected to best resemble the viscoelastic material properties. A minimization of the least squares will yield a fit to the material data for the best developed equations by determining the values of parameters in the equations. The basis for data processing, storage, and retrieval is the use of these parameters, representing the best fit of the material data, to aid in efficient, systematic, and standardized procedures for representing the viscoelastic material properties data.

As a precursor to the above fit of the equations, an editing procedure has been developed to assist in the sorting of data. The material data can be expected to include observations which are not consistent with the remaining data, and, to delete this data from consideration, an editing procedure has been implemented. After the data is edited, the equations are fitted to the remaining material data for further processing.

This section consists of subsections representing the editing procedure, the empirical shift parameter equation, the empirical modulus equation, and the empirical loss factor equation.

2.1 EDITING PROCEDURE

The editing procedure consists of a list of the material data and several plots. The plots are of observed frequency and observed loss factor versus observed temperature.

The loss factor versus temperature plot is important because the data will have the general shape of an inverted parabola, and a smooth curve can usually be fitted through the data. From this smooth curve, the peak can be determined and this point becomes the reference temperature (t_0) used to "shift," and minimize the scatter of the modulus and loss factor data.

The frequency versus temperature plot is important because the data will generally lie along nearly parallel curves for each resonant mode observed, at least for resonant beam type tests⁴.

The plots, along with a list of the material data, are used to edit the data. If a data point does not lie close to the general parabolic shape of the loss factor versus temperature curve or is not along the curves of frequency versus temperature, the data point should be rejected for the curve fit, since the observation generating that data point is not consistent with the remaining observations.

The editing procedure, therefore, is to eliminate any grossly inconsistent data points or observations and produce a reference temperature (t_0) for the shift parameter equation.

2.2 EMPIRICAL SHIFT PARAMETER EQUATION

The modulus (M) and loss factor (η) of a viscoelastic material are functions of two independent variables, namely temperature (t) and frequency (f). In order to plot the modulus and loss factor in two dimensions, these properties must be functions of one variable, where this one variable contains the two independent variables mentioned above. To accomplish this, a shift parameter (α_T), which is a function of the temperature, has been empirically developed¹. This shift parameter, when multiplied by the frequency, gives the variable of reduced frequency (f_r) of which the modulus and loss factor are functions.

The shift parameter equation is used to reduce the two dimensions f and t to one dimension $f\alpha_T$. The procedure reduces the scatter of the dependent modulus and loss factor equations

which have been reduced from surfaces in three dimensions with two independent variables, to curves in two dimensions with one variable (f_r). A universal shift parameter equation has been empirically determined by comparing the results of many materials⁵. It has been determined that the shift parameter equation is dependent only on temperature and is relative to a reference temperature (t_o) for the material, which reduces the scatter of the modulus and loss factor data to fit a smooth curve to the data.

Based on rationale presented in Reference 3, the "universal" shift parameter equation to be used for the processing of experimental viscoelastic material data is as follows:

$$\log_{10}(\alpha_T) = \frac{-12(t-t_o)}{\frac{525}{1.8} + (t-t_o)} \quad (1)$$

where the indicated temperature is in degrees Centigrade. The reduced frequency is computed as follows:

$$f_r = f \cdot \alpha_T$$

therefore:

$$f_r = f \cdot 10^{\left(\frac{-12(t-t_o)}{\frac{525}{1.8} + (t-t_o)} \right)}$$

or, in terms of the log:

$$\log_{10}(f_r) = \log_{10}(f) - \frac{12(t-t_o)}{\frac{525}{1.8} + (t-t_o)} \quad (2)$$

This equation therefore reduces the two independent variable problem to a one variable problem for the modulus and loss factor curves.

2.3 EMPIRICAL MODULUS EQUATION

The general shape of the log modulus versus log reduced frequency curve has horizontal asymptotes about which it is bounded. The curve is sigmoidal, monotonic, and an odd function about offsets. The equation considered in this report was derived in Reference 3 and is of the general form:

$$\log_{10}(M) = \log_{10}(M_\ell) + \frac{2 \log_{10} \left(\frac{M_{rom}}{M_\ell} \right)}{1 + \left(\frac{f_{rom}}{f_r} \right)^N} \quad (3)$$

This curve is asymptotic to M_ℓ for low values of reduced frequency (f_r), passes through the offset point M_{rom} at f_{rom} with a slope governed by the value of N , and has an asymptote for high values of reduced frequency. This curve, therefore, is an odd function about the point (f_{rom}, M_{rom}) in log-log coordinates.

A general least squares fit of the above curve to the material data will yield parameter values for f_{rom} , M_{rom} , N , and M_ℓ that will best describe the material data, where M is the observed modulus and f_r is the observed reduced frequency presented in Section 2.2. The indicated modulus values are in N/m^2 , the indicated reduced frequency values are as presented in Section 2.2, and the slope parameter value at the offset is dimensionless in the presented modulus curve.

This curve can be used to represent any type of modulus. In the case of shear modulus, simply replace the M 's in Equation 3 with G 's, indicating shear modulus. In the case of Young's modulus, simply replace the M 's in Equation 3 with E 's indicating Young's modulus. The equation of the modulus curve presented can be used to fit the material data to a smooth curve, and can also be represented on a graph of the viscoelastic material data. The parameter values determined by the least squares fit can

be stored on the computer, and the equation, along with the parameters, can be used at a future time to predict the modulus of the material for which the parameters are defined. Therefore, the storage and retrieval of data from the computer is important for further studies of viscoelastic material properties.

2.4 EMPIRICAL LOSS FACTOR EQUATION

The general shape of the log loss factor versus log reduced frequency curve is hyperbolic in nature. The general curve is an offset symmetric hyperbola with straight line asymptotes added to the equation.

The equation considered was derived in Reference 3 and is of the general form:

$$\log_{10}(\eta) = \log_{10}(\eta_{frol}) + \frac{C}{2} \left[(S_L + S_H) \frac{1}{C} \log_{10} \left(\frac{f_r}{f_{rol}} \right) + (S_L + S_H) \left(1 - \sqrt{1 + \left(\frac{1}{C} \log_{10} \left(\frac{f_r}{f_{rol}} \right) \right)^2} \right) \right] \quad (4)$$

This curve has a peak of η_{frol} at f_{rol} with a peak shape defined by C , is asymptotic to a line of slope S_L for low values of reduced frequency (f_r), and is asymptotic to a line of slope S_H for high values of reduced frequency.

A general least squares fit of the above curve to the material data will yield parameter values for η_{frol} , S_L , S_H , f_{rol} , and C that will best describe the material data, where η is the observed loss factor and f_r is the observed reduced frequency present in Section 2.2. The indicated reduced frequency values are as presented in Section 2.2, and the remaining values are dimensionless in the presented loss factor curve.

The equation of the loss factor curve can be used to fit the material data to a smooth curve, and can also be represented on a graph of the viscoelastic material data. The

parameter values determined by the least squares fit can be stored on the computer, and the equation along with the parameters can be used at a future time to predict the loss factor of the material for which the parameters are defined.

2.5 PARAMETER CONSTRAINTS

A study³ was performed to examine the effect of changing parameters and applying the equations to a limited span of observations, and it was found that constraints had to be imposed on the parameters for the modulus and loss factor curves. The study concluded that the least squares fit to the material data gave an excellent fit to the span data, but that the parameter values obtained were not realistic for the physical interpretation of the equations. Also, the study concluded that parameters should be held constant is possible and also may be constrained for the physical interpretation of the curves.

Based on the least squares fit in the study³, the following constraints for the given parameters are shown in general:

$$\begin{aligned}
 f_{rom_min} &\leq f_{rom} \\
 M_{rom_min} &\leq M_{rom} \leq M_{rom_max} \\
 M_{l_min} &\leq M_l \\
 \eta_{frol_min} &\leq \eta_{frol} \\
 S_{l_min} &\leq S_l \\
 S_h &\leq S_{h_max} \\
 f_{rol_min} &\leq f_{rol}
 \end{aligned} \tag{5}$$

By imposing the above constraints and allowing parameters to be held constant, the curve fits can now take on physical meaning.

To obtain the "best" possible fit to the material data, a wide range of observations must be taken to ensure the material data can be characterized by the empirical curves presented. The study³ shows that the viscoelastic material data can be characterized by the equations, and with some degree of certainty these equations, along with the desired material parameters, can be used to predict the viscoelastic material properties.

SECTION III

PROGRAM DEVELOPMENT

The computer program was developed as a result of discussions with Air Force Materials Laboratory (AFML) and Air Force Flight Dynamics Laboratory (AFFDL) personnel concerned with the characterization of material properties and usage of viscoelastic materials in damping treatments. The type of program developed was beneficial for the physical representation of the viscoelastic material data and utilizations by the intended users. The development included the general flow for the program, the use of available equipment, and the use of a curve fit routine to fit the data to the nonlinear curves presented in Section II. These items were the basis for the program development.

3.1 GENERAL PROGRAM FLOW

A general flow of the computer program was designed, as presented in Figure 1, with the ability to read in viscoelastic material property data, edit the data to eliminate any inconsistent data, obtain a reference temperature that reduces the scatter of the data about the given curves, fit the modulus data to a characteristic curve, and finally, plot the edited viscoelastic material property data on the desired grid with corresponding labels.

As the program developed and after further discussions with AFML and AFFDL personnel, extensions to the basic program were instituted to make the program more desirable for the user. An option was made available to store the data and parameters of the curves within the computer facility.

Since a store option was included, a retrieve option was added so the user can store the data at one time and execute the program at a future time, retrieve the data, and continue the analysis of the data. Also within the edit routine, an added option was included so the user could retrieve stored data, add more data for the analysis, and have the option to store the expanded data set.

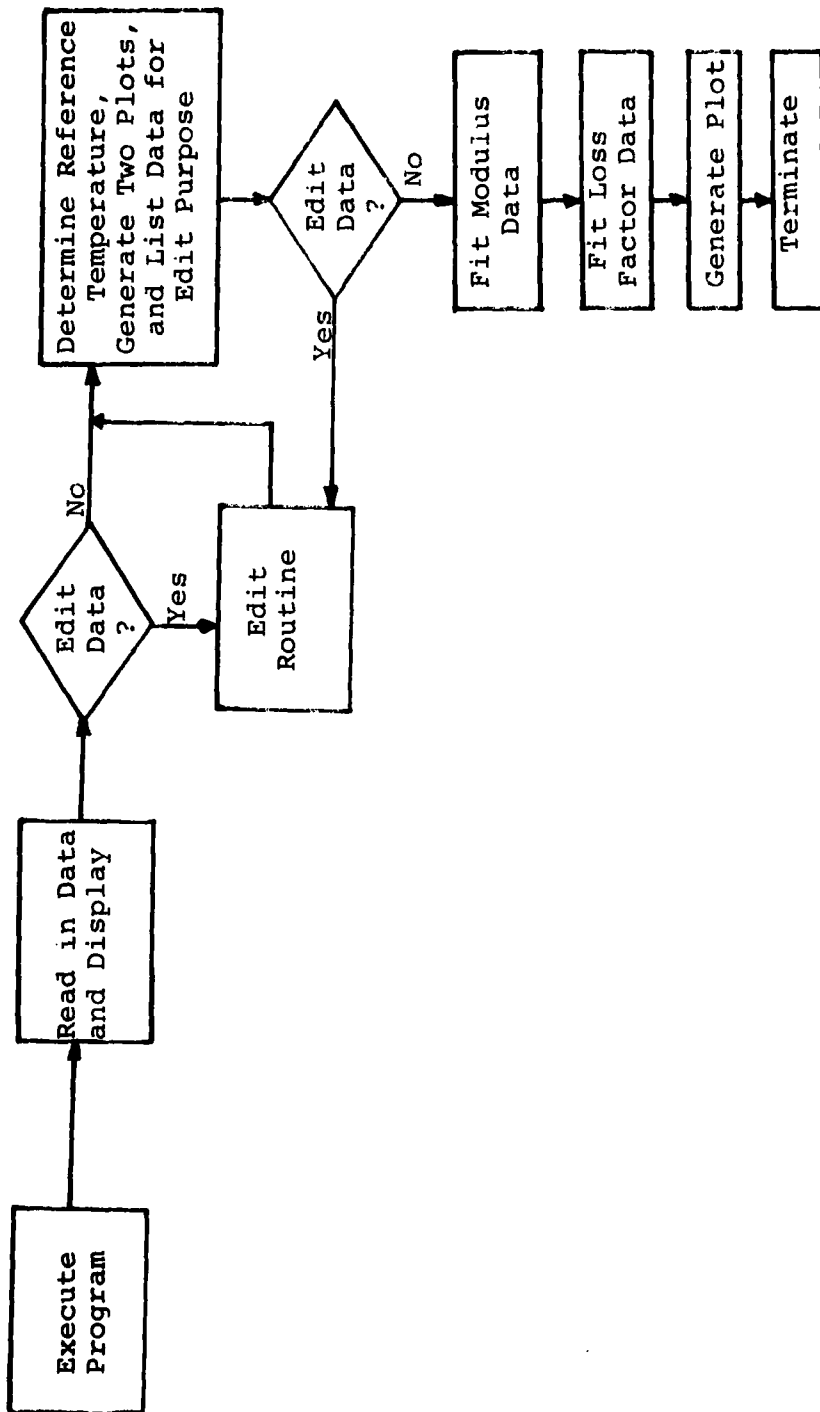


Figure 1. General Program Development.

Also, at this point a decision was made to perform all calculations and final plots in metric units, therefore, an option was included to allow the user to read in new data in either metric or English units, and to add new data in the edit routine, in metric or English units. A more detailed general flow of the program is shown in Figure 2.

Further discussions with AFML and AFFDL personnel have led to more additions to the basic program. These additions do not affect the flow of the program presented. These additions are basically items that interface with the user-program interaction within each block in the figure.

3.2 AVAILABLE EQUIPMENT

The program development was limited to the equipment available for implementation and usage of the computer program. The computer program was coded in FORTRAN IV for the HP-2100 minicomputer under the DOS-III operating system. This minicomputer is available at the AFML where the implementation and usage of the program is to be available for AFML personnel. The DOS-III operating system, with a FORTRAN IV compiler, allows the storage of data on files resident on a disk. This feature is useful in the program and in the program development by implementing disk storage of data by the program. The disk storage capability also allows the user to read and write data to files, and therefore to build a thorough data base of viscoelastic material property data.

Also available at the AFML is a Tektronix graphics screen terminal, which is needed to accomplish the required computer generated graphics of the program. This terminal is connected directly to the HP-2100 computer mentioned above. The terminal has a hard-copy unit which produces a copy, on a sheet of paper, of all information projected onto the screen of the terminal. This equipment is valuable for generating the plots of the material data, and has the ability to obtain paper copies of the screen for reference at a future time.

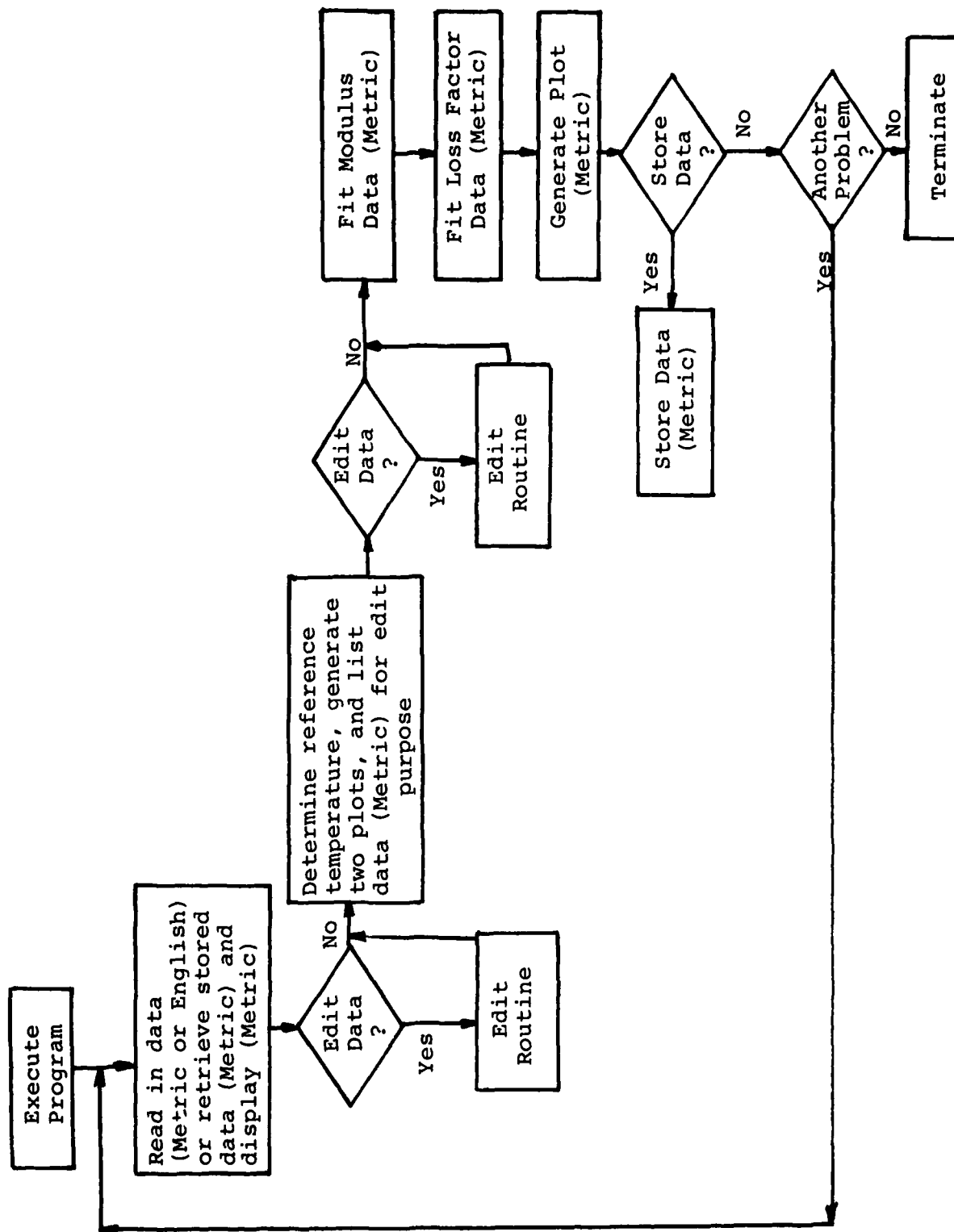


Figure 2. Detailed Program Diagram.

The software required for the implementation of this program is the DOS-III operating system which allows storage of data, the FORTRAN IV compiler, and a package of utility routines used to project the graphics or plots onto the screen of the terminal. These utility routines are commonly known as the PLOT-10 software package for the Tektronix hardware.⁶ These routines allow the user to display any information on any part of the screen. This is a very powerful package for projecting graphics and alpha-numerics on the screen of the terminal.

The decision was made to use the above mentioned equipment and software. Since AFML has the facility indicated, the program development and implementation was completed at AFML, for the purpose of program use by AFML personnel responsible for the characterization of viscoelastic material property data.

3.3 TYPE OF MINIMIZATION

The program development also led to the examination of curve fitting routines for the modulus and loss factor curves presented in Section II. The decision was made to use the minimization technique known as least-squares to determine the parameters that most closely describe these nonlinear curves of the data. In this case, the least-squares technique leads to the minimization of a nonlinear function, being the sum of the square deviations of the data about the defining curve. A direct search approach was used to scan the parameter space, within which the nonlinear curve is defined, and determine a point in the parameter space that minimizes the nonlinear function. The direct search approach, as applied to the computer program, implements some type of prescribed search pattern to arrive at the minimum of the nonlinear function being considered. Once this point has been found, the prescribed search pattern determines that any movement in the parameter space by the direct search approach will only yield an increase in the value of the nonlinear function being minimized. Therefore, this point must be the minimum of the nonlinear function within a region of the minimizing point in the parameter space.

This type of minimization procedure was chosen because it does search the parameter space directly to find the minimum value of the function to be minimized. Also, this type required a much smaller amount of coding than other techniques, thus allowing more coding for other parts of the overall program. Also, this type is relatively fast for small dimensionality of the parameter space being considered.

SECTION IV

PROGRAM DESCRIPTION

This section contains an overall general description of the program implemented to represent the viscoelastic material properties data. Several items that will be discussed are the interactive nature of the program including inputs and outputs, a description of the minimization technique, a description of the graphics package, a description of the structure of the data files for data storage, and a description of the initialization of the parameter values for the modulus and loss factor curves presented in Section 2.

4.1 INTERACTIVE FEATURES

The computer program was developed to accommodate the interactive features requested by AFML and AFFDL personnel. The user manual, presented in Section 6, presents all the possible input and output features available in the program which require user interaction. The interactive features provided are of three basic types.

The decision type involves alphanumeric input by the user for decision type of responses for the program. These input types are preceded by the output of a question, for which the user inputs the correct alphanumeric character for the response to the question. The program then interprets this decision and proceeds to the next sequential segment of the program based on the response decision. These decision alternatives are presented in Section 6, the User's Manual, for each question output by the program which requires a decision type of response by the user.

The numeric type of input involves numeric values to be entered by the user. These inputs are preceded by an output, for which the user inputs the correct numeric values for the response to the output. This output, in general, implies that the user is to enter data or values for the following input. This input must be in the form of a number, thus indicating a numeric value being

entered to the program. These numeric-type inputs are presented in Section 6, the User's Manual, for each output by the program which requires a numeric type of input or response by the user.

The control type input involves the entering of control instructions by the user. These control instructions are instructions to the operating system for the HP-2100 minicomputer. The user inputs a control instruction to initiate execution of the program. After execution begins, the program will periodically suspend and pause execution for the user to obtain copies of the terminal screen. The user then inputs a control instruction to continue execution of the program or discontinue execution of the program, causing an abnormal termination of the program. These control inputs are presented in Section 6, User's Manual.

4.2 DATA FILE STRUCTURE

The computer program development led to the discussion of data storage with AFML and AFFDL personnel. The discussion led to three data files for data storage by the program. A general description of these files and type of data acquisition is discussed.

The description of these data files is rather simple. The program development led to the use of three files to store data. The first file is a general file which stores the material name, data sources, and other items (i.e., number of data points, material code number, data type, etc.) for the material on the designated material code number. This number represents the storage location in the data file for the above-mentioned data. This file is used to indicate if any material data is stored on the designated material code (see Section 6). The second file is the data file to store the reduced and edited material data. This data file can be accessed through the first file and the material data can be found relative to the material code number. This, again, represents the storage location in the data file for the reduced and edited material data.

The third file is the data file to store the parameter values determined by the curve fits of this program. This data file can be accessed through the first file and the parameter data can be found relative to the material code number. This again represents the storage location in the data file for the parameter values. This third file cannot be retrieved directly by the user of this program as the other two can (Section 6). This file is the product of the computer program and will be used at a future time in another program, which will use the parameter values for a material to predict the material properties. As can be seen, these three files are combined by use of the material code number of the first file to completely store the data for this program and subsequent follow-on programs. This type of data file structure eliminates the need for the user to enter the entire data at the terminal at each execution of the program. This structure also allows the building of data sets for each material in the edit sequence of the program, and then store the expanded data set for the material. This structure also allows re-editing and re-fitting of the data to the curves until a representative graph of the material data has been achieved, without the need of re-entering the data each time. Finally, this structure allows the implementation of other programs to store data on these files for use by this program, and to retrieve data from these files for use by subsequent analysis-type programs.

The above mentioned data files are resident on the HP-2100 minicomputer disk at the AFML, along with the computer program. The operating system allows the programmer to implement, in the program, direct storage and retrieval of data on the disk. This data must have specified file names and be able to locate storage areas, in the files, relative to the beginning of the files. For this computer program, the three files are implemented. By use of a utility routine of the operating system callable by FORTRAN IV, the program can store or retrieve (write or read) data from the files mentioned and from the storage location indicated by the

material code for each file. This capability of the operating system assisted in the development of this computer program and aids in the storage and retrieval of material data. Thus, having stored a massive amount of material properties data, these files can be used for curve fitting, further analysis, and predictions of the viscoelastic material properties.

4.3 GRAPHICS PACKAGE

The hardware to accomplish the required graphics is the TEKTRONIX 4014-1 graphics screen terminal. This terminal is well adapted for graphics because it has what is commonly known as a vector mode for the terminal. In the vector mode the programmer has the ability to address individual points on the screen, thus having the ability to draw lines and have complete control over the addressability of the entire screen. In the alphanumeric mode, the programmer can use the FORTRAN IV input/output to display, on the screen, the input and output required by the program. This hardware is connected directly to the HP-2100 mini-computer on which this program is to be implemented.

To accomplish graphics on the terminal, the TEKTRONIX Corp. has supplied a package of subprograms which can be used by the programmer to generate the graphics and thus project the graphics on the screen of the terminal for display to the user. The package supplied involves complete terminal control subprograms which control the status of the terminal (i.e., vector mode or alphanumeric mode) so the programmer has complete control of the terminal at all times in the program. This type of programmer control allows the coexistence of graphics and FORTRAN input/output on the screen for display to the user. The subprograms implemented for this program are not described in the subprogram description (Section 7). These subprograms are described in Reference 6, which includes a description of each subprogram, what the subprogram accomplishes for the terminal, and how the subprogram controls the display on the screen of the terminal. This graphics package,

along with the hardware, allows the programmer to display any required graphics and/or input/output required for the implementation of the computer program to generate a representation of the viscoelastic material properties.

4.4 MINIMIZATION TECHNIQUE

The nonlinear function described for the minimization is the sum of square deviations or, as commonly known, least squares fit of data to a given curve. The type of minimization implemented in the computer program is the direct search method. In this method, the algorithm searches the parameter space directly to determine the minimum of the function. To find the values of the parameters in the parameter space, this method uses some predetermined search pattern to locate the minimum of the function in the given parameter space. The direct search technique uses what is commonly known as a SIMPLEX in the search pattern.

A SIMPLEX is a collection of points in the parameter space, for which each point is called a vertex and is represented by the function value and parameter values. The direct search method replaces each vertex with another vertex which yields a smaller functional value. This scheme continues until all the vertices converge to one point which should be the minimum functional value. This point, therefore, represents the function and parameter values that yield a minimum for the function within a given neighborhood of the point in the parameter space. Any search efforts beyond this point will only yield a larger value for the function in the parameter space. These parameter values of the point, therefore, represent the parameters that force the function to be a minimum. In this program the function is again the least squares fit of data to desired curves (Section 2), and the parameter values are those values for the parameters of the curve that cause the function to be a minimum. This direct search technique, therefore, has found parameter values for the desired curves that most closely represent the viscoelastic material properties.

4.5 INITIALIZATION OF PARAMETER VALUES

The minimization technique requires initial parameter values for the direct search procedure. These initial values are used to provide a starting point for the search and are also used to describe the search pattern required for the direct search method.

For the modulus curve, the initial parameter values are assigned in the following way. The f_{rom} parameter value is found by taking the square root of the product of the minimum and maximum observed reduced frequency values from the data. The M_{rom} parameter initial value is found by fitting a second-degree polynomial to the log modulus versus log reduced frequency data, substituting the value of the log f_{rom} above to determine the value of log M_{rom} , and then looking up the anti-log to find the value for M_{rom} . The M parameter initial value is found by taking one-half of the minimum observed modulus value from the data. The N parameter initial value is assigned to .4.

For the loss factor curve, the initial parameter values are assigned in the following way. The f_{rol} parameter initial value is found by fitting a second-degree polynomial to the log loss factor versus log reduced frequency data, determining the location of the peak for the polynomial, and taking the anti-log of the log reduced frequency value to assign the value of f_{rol} . The η_{frol} parameter initial value is found by determining the value of the peak for the polynomial fit, and taking the anti-log of the log loss factor to assign the value of η_{frol} . The S_l parameter initial value is assigned to .35. The S_h parameter initial value is assigned to -.45. The C parameter initial value is assigned to 4.0.

The initial parameter values are assigned by the above methods. The search technique uses these initial values to determine the search procedure to implement in the parameter space. The search technique then proceeds, from the initial values, to

search the parameter space and determine the parameter values that will most closely describe the desired curves for the viscoelastic material properties data.

SECTION V

SUBPROGRAM DESCRIPTION

This section includes a general description of the routines which are combined together to form the computer program which generates the representation of the viscoelastic material properties data. Many of the subprogram routines provide user interaction and input to control the representation of the viscoelastic material properties. This section does not include a description of the graphic routines needed for the overall computer program. A description of these routines can be found in Reference 6.

5.1 PROGRAM VIBR2

This routine is the main segment of the computer program. When execution of the program begins, this segment initializes the graphics package for further use by the program. This segment makes several calls to routines in the graphics package to set the appropriate values to be used by the graphics package. This segment first calls the read-in and retrieval routine to make available the data to be represented. This segment then inquires about editing the listed data. If so, this segment calls the editing routine. In either case, this segment then calls the routine to fit the loss factor data to a second degree polynomial curve of the temperature data. This segment then calls the graphics routine which generates two graphs and lists the data. This segment again inquires about editing the data. If so, the editing routine is called, and this segment proceeds back to call the routine to fit the loss factor versus temperature data. If not, this segment inquires about the changing of the reference temperature. If so, this segment has the user input the reference temperature. In either case this segment calls the routine to fit the modulus data to the specified curve presented in Section 2. This segment then calls the routine to fit the loss factor data to the specified curve presented in Section 2. This segment inquires about

the user input for the interval of temperature lines for the representative graph. This segment then calls the graphing routine which generates the appropriate grid, graphs the data, and graphs the curves determined by the above curve fits. This segment then calls the routine which outputs the curves used in the curve fits and the parameters that describe the curves for the listed data. This segment inquires about changing the reference temperature and/or parameters. If so, this segment has the user input the reference temperature and proceeds back to call the modulus curve fitting routine. If not, this segment inquires about the storing of the data. If so, this segment calls the storing routine. In either case, this segment inquires about the use of the program to represent another set of data. If so, this segment proceeds back to the read-in and retrieval routine after clearing the screen of the terminal to prepare another set of data to be represented. If not, this segment terminates the graphics package and terminates execution of the program. Control is then returned to the operating system.

5.2 SUBROUTINE REDIN

This segment of the program is used to make available the data to be represented. This segment inquires about the listing of labels for previously stored data. If so, this segment, by use of a utility routine of the operating system, displays the labels of the desired stored data. In either case, this segment inquires about the reading in of new data. If so, this segment asks many questions and requires much input from the user to characterize and label the data. The user has the option to enter the data in metric units or English units. The user then inputs the data in either unit and, after it is entered, the data is converted to metric units because all calculations and computations are accomplished in metric units. If not, this segment inquires a retrieval code which represents data that has already been stored by the program or by other programs. This segment uses a utility routine of the operating system to retrieve the stored data from the files of the

computer system. In either case, this segment erases the screen and redisplay this newly read-in data or retrieved data now output in metric units. This segment then pauses in its execution of the program, so that the user at his option can obtain copies of the terminal screen for further reference, and then returns control of the program to its calling routine.

5.3 SUBROUTINE FITTO

This segment is used to fit a second degree polynomial to the loss factor data as a function of the temperature, by use of a routine to fit the data. This segment also determines the peak of the polynomial to give the reference temperature of the material to the nearest multiple of 10°C. This segment then returns control to the calling routine.

5.4 SUBROUTINE POLYF

This segment is used to fit a second degree polynomial to any data. This segment is included because the program needs, in several instances, a polynomial fit to the data. This segment then returns control to its calling routine.

5.5 SUBROUTINE PLTTO

This segment of the program is used to set the specified grid size for the frequency versus temperature and loss factor versus temperature graphs. This segment clears the screen of the terminal, sets the specified grid sizes, and then calls a plotting routine twice to generate the two graphs, mentioned above, on the screen. This segment then prepares the terminal for further output by several calls to routines of the graphics package. This segment then outputs the data that is represented by the graphs displayed on the screen, and returns control of the program to its calling routine.

5.6 SUBROUTINE PLTT1

This segment of the program is used to generate a graph. This segment generates either the frequency versus temperature plot or the loss factor versus temperature plot. For the former, this segment outputs, through calls to the routines of the graphics package, the grid of the plot, the labels of the axes for the grid, and the plotting of the points, along with an identifying number, of the frequency versus temperature data. For the latter, this segment outputs, through calls to routines of the graphics package, the grid of the plot, the labels of the axes for the grid, the plotting of the points, along with an identifying number, of the loss factor versus temperature data, a curve through the loss factor versus temperature data mentioned above, and an output of the reference temperature for the material data. This segment then returns control to its calling routine.

5.7 SUBROUTINE EDITO

This segment is used to edit the material data which has been read in or retrieved. The options available to the user are the ability to add new data, delete data, or end the edit sequence. If the user wants to add new data, this segment allows adding of data in metric or English units. The user then inputs the new data, and the new data is then converted to metric units after input. If the user wants to delete data, this segment requires input, by the user, for the lines or point numbers from the data list to be deleted. These points are deleted and the data is resequenced to indicate that the data has been entirely deleted from the data being considered. If the user wants to end the edit sequence, this segment terminates and returns to its calling routine.

5.8 SUBROUTINE STORE

This segment of the program is used to store the material label, material data, and parameters of the curve fit (Section 2) on specified files. These files are set to be data files for this

program and, through the use of a utility routine of the operating system, this data can be stored or written to these files on the disk of the computer. This data is stored for future reference and possibly future analysis needed for the given material. This segment outputs a message that the data has been stored and pauses execution of the program so the user can obtain copies of the terminal screen for further reference, and returns control of the program to its calling routine.

5.9 SUBROUTINE MODUS

This segment of the program is used to fit the modulus data to the curve presented in Section 2. This segment clears the screen of the terminal and outputs a message to the user. This segment then initializes the parameters for the modulus curve fit from the given data with the use of the polynomial fit routine. This segment then inquires about the constraints, for which the user enters the constraint values as presented in Section 2. This segment then calls the minimization routine to determine the values of the parameters for the modulus curve fit. This segment calls the routine to output the parameter values and modulus curve fit to the terminal. This segment then inquires if the user wants to accept these parameter values. If he does, this segment returns control to its calling routine. If not, this segment inquires about the initial values the user wants to input for the parameters, and the ability of the user to hold parameters constant in the minimization process. This segment then proceeds back to call the minimization routine.

5.10 SUBROUTINE ETA

This segment of the program is used to fit the loss factor data to the curve presented in Section 2. This segment clears the screen of the terminal and outputs a message to the user. This segment then initializes the parameters for the loss factor curve fit from the given data with the use of the polynomial fit routine.

This segment then inquires about the constraints for which the user enters the constraint values as presented in Section 2. This segment then calls the minimization routine to determine the values of the parameters for the loss factor curve fit. This segment calls the routine to output the parameter values and loss factor curve fit to the terminal. This segment then inquires if the user wants to accept these parameter values. If he does, this segment returns control to its calling routine. If not, this segment inquires about the initial values the user wants to input for the parameters and the ability of the user to hold parameters constant in the minimization process. This segment then proceeds back to call the minimization routine.

5.11 SUBROUTINE FCN

This segment is used by the minimization routine to obtain the value of the function to be minimized, namely, the sum of square deviations. This segment uses the reference temperature routine to obtain the value of the independent variable, namely, reduced frequency. This segment then uses the curve selection routine, the parameter values, and the independent variable value to determine the sum of square deviations of the data being considered. This segment then returns control to its calling routine.

5.12 FUNCTION FCT

This segment is used to select the curves to fit the data. The parameter values and independent variable value are used to select either the modulus value or the loss factor value for the parameter values and independent variable value being considered. This segment then returns a value to its calling routine.

5.13 FUNCTION ALPHT

This segment uses the temperature of the data point along with the reference temperature for the material to calculate the value of the α_T curve presented in Section 2. This α_T value is

then multiplied by the frequency of the data point to determine the independent variable value for the data point, namely, reduced frequency, which is then used in the modulus and loss factor curves and curve fits. This segment then returns the α_T value to its calling routine.

5.14 FUNCTION F1

This segment of the program is used to evaluate the modulus curve for the given parameter values and the independent variable value. If any of the parameter values are outside the bounds of the constraints presented in Section 2, the value of the modulus curve is set to a large number in hope that the minimization process will bring the parameter values within the constraint bounds and obtain the appropriate curve fit to the modulus data. This segment then returns the modulus curve value to its calling routine.

5.15 FUNCTION F2

This segment of the program is used to evaluate the loss factor curve for the given parameter values and the independent variable value. If any of the parameter values are outside the bounds of the constraints presented in Section 2, the value of the loss factor curve is set to a large number in the hope that the minimization process will bring the parameter values within the constraint bounds and obtain the appropriate curve fit to the loss factor data. This segment then returns the loss factor value to its calling routine.

5.16 FUNCTION ALOG1

This segment of the program is used to evaluate the logarithm, to base ten, of a number. The HP-2100 mini-computer has this routine available as a utility, but has been shown to have several bugs in the routine. Therefore, to ensure proper operation, this short segment is added to the program. This segment then returns the value to its calling routine.

5.17 SUBROUTINE VERTL

This segment of the program is used to output a string of characters, for graph labeling purposes, in the vertical direction. The graphics package needed for the program does not have a routine to output a vertical string of characters on the terminal. This segment is implemented to accomplish this by using calls to routines in the graphics package. This segment then returns control of the program to its calling routine.

5.18 SUBROUTINE SEARC

This segment is used to minimize the function of the parameters, namely, the sum of square deviations or least squares. This segment is a package routine which uses the direct search technique to minimize the function. This segment uses the function routine to evaluate the function as this segment searches directly in the parameter space to determine the parameter values that minimize the function being considered. This segment then returns control to its calling routine after this segment has determined the parameter values in the parameter space that minimizes the sum of square deviations.

5.19 SUBROUTINE GAXES

This segment of the program generates on the screen of the terminal the characteristic graph for the material data. Through extensive calls to the graphics package routines, this segment clears the screen, displays the appropriate grid to accommodate the data, graphs the modulus data, graphs the modulus curve, graphs the loss factor data, graphs the loss factor curve, graphs the temperature lines², outputs the appropriate scales for each of the axes, and labels each of the four axes. This segment pauses execution of the program so the user can examine the graph and obtain copies of the graph for further reference and returns control of the program to its calling routine.

5.20 SUBROUTINE PLTPK

This segment of the program is used to output, for display by the user, the curves to which the data was fit along with the value of the reference temperature and values for the parameters of the curve fits. This segment also outputs the material code and material name along with the final curve and parameter values which are represented by the characteristic graph. This segment pauses execution of the program for the user to obtain copies of the screen, and then returns control to its calling routine.

5.21 SUMMARY

Mentioned in the above subsections are only brief descriptions of the subprograms that form the complete program. Any detailed description would require a line-for-line analysis of the code for each subprogram to thoroughly examine the entire computer program.

SECTION VI

USER MANUAL WITH DESCRIPTIVE SAMPLE INPUT/OUTPUT

This section presents the user interaction with the program to generate the desired representation of the experimental visco-elastic material data. This section is divided into two subsections. The first subsection gives the user a general walk-through of the program and describes the commands the user must input from the terminal. The second subsection presents sample outputs of the program along with a descriptive note indicating the input by the user.

The user is to note that all underscored items which follow indicate user input to the program. The underscored upper-case letters must be entered as indicated to ensure proper format within the program. The lower-case letters indicate free-format input by the user to the program. In this case the items may be entered as desired by the user. If several underscored items appear in the user manual on a single line, the user only needs to separate these free-format items by a comma or a space to ensure proper entry to the program. After each input, the user enters the carriage return-line feed sequence (CR/LF) to enter the input to the program.

6.1 USER MANUAL

The user must first initiate execution of the program. To accomplish this, the user must input:

:RUN, VIBR2 CR/LF . (1)

The program begins by clearing the screen and outputs the following:

LIST THE STORED DATA?(Y,N) (2)

for which the user inputs either:

Yes CR/LF or No CR/LF .

If No is entered for (2), the program proceeds to Step 2.

If Yes is entered for (2), the program outputs:

ENTER THE MAXIMUM NUMBER OF MATERIAL CODES (3)

for which the user inputs:

max CR/LF,

the maximum number of material codes available to the user (currently 200 3-14-78). The program searches the data files created and outputs the material code, material name, and data sources for all stored material data, and then proceeds to Step 2.

STEP 2

The program responds:

READ IN NEW DATA?(Y,N) (4)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If No is entered for (4), meaning the user wants to retrieve data already stored on the data files, the program outputs:

ENTER THE CODE OF RETRIEVAL MATERIAL (5)

for which the user inputs:

m CR/LF,

where m is the material code for the material data to be retrieved. If the material code does not match the material data stored on the files, the program outputs:

MISMATCH IN RETRIEVED MATERIAL CODES (6)

and proceeds back to output (5) until a match for retrieved data occurs upon which the program proceeds to Step 3.

If Yes is entered for (4) the program prepares for the entry of new data to the program.

The program outputs:

ENTER HEADER DATA

ENTER THE CODE OF THE MATERIAL (7)

for which the user inputs:

i CR/LF,

where the number i, between 1 and 200 inclusive, corresponds to the position the following data is to be stored on the data files.

The program then outputs:

ENTER THE MATERIAL NAME (MAX 60 CHAR.) (8)

for which the user inputs:

name CR/LF,

the name of desired material.

The program then outputs:

ENTER THE MANUFACTURER DATA SOURCE (MAX 60 CHAR.) (9)

for which the user inputs:

source CR/LF,

the manufacturer source of material data.

The program then outputs:

ENTER THE AFML DATA SOURCE (MAX 60 CHAR.) (10)

for which the user inputs:

source CR/LF,

the AFML source of material data.

The program then outputs:

ENTER OTHER DATA SOURCES (MAX 60 CHAR.) (11)

for which the user inputs:

sources CR/LF,

any other sources of material data.

The program then outputs:

ENTER THE NUMBER OF DATA POINTS (MAX 64 PTS.) (12)

for which the user inputs:

n CR/LF,

the number of data points to input.

The program then outputs:

IS THE DATA TO BE ENTERED METRIC? (Y,N) (13)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If Yes is entered for (13), a flag is set to assume the data to be entered is metric.

If No is entered for (13), a flag is set to assume the data to be entered is English and will be converted to metric immediately after entering.

The program then outputs:

IS THE MODULUS DATA SHEAR?(Y,N) (14)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If Yes is entered for (14), a flag is set to indicate the data is for shear modulus of the material.

If No is entered for (14), a flag is set to indicate the data is for Young's modulus of the material.

The program then outputs several lines for the user, to indicate the program is now ready to enter the viscoelastic material data.

The user inputs:

i, M_i, η _i, t_i, f_i, m_i CR/LF,

for i between 1 and n inclusive, where i is the ith data point, M_i is the modulus, η _i is the loss factor, t_i is the temperature, f_i is the frequency, and m_i is the mode number for that data point. The program converts the data to metric if entered in English since all computations are done in metric. The program then proceeds to Step 3.

STEP 3

The program clears the screen and proceeds to output the new or retrieved material data including headings and labels, where all the units are indicated as metric since all computations are done in metric. The program then pauses so that the user can obtain copies of the screen for future reference. The program outputs:

VIBR2: PAUSE 0000

VIBR2 SUSP. (15)

@

to continue the program when the user is ready the user inputs:

:GO CR/LF.

The program outputs:

EDIT THE DATA?(Y,N) (16)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If No is entered for (16), the program proceeds to Step 4.

If Yes is entered for (16), the program proceeds to edit the data shown on the screen. The editing portion of the program outputs:

EDIT THE ABOVE DATA: ADD, DELETE, OR END?(A,D,E) (17)

for which the user inputs:

Add CR/LF or Delete CR/LF or End CR/LF.

If Add is entered for (17), the program outputs:

ENTER THE NUMBER OF DATA POINTS TO ADD (18)

for which the user inputs:

m CR/LF,

the number of data points to add to the data shown. The total number of data points cannot exceed 64.

The program then outputs:

IS THE DATA TO BE ADDED METRIC?(Y,N) (19)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If Yes is entered for (19), a flag is set to indicate the data to be added is metric.

If No is entered for (19), a flag is set to indicate the data to be added is English and a conversion to metric will be performed after the data is entered.

The program then outputs:

READY TO ADD DATA FOR MATERIAL IN FORMAT SHOWN ABOVE (20)

for which the user inputs:

$i, M_i, \eta_i, t_i, f_i, m_i$ CR/LF ,

for i , between 1 and m inclusive, being the i^{th} added data point for the material data. The program makes the conversion to metric if necessary. The program then proceeds to output (17) again.

If Delete is entered for (17), the program outputs:

ENTER THE NUMBER OF DATA POINTS

TO DELETE AND THE INDICES OF THE ROWS TO DELETE (21)

for which the user inputs:

$m, i_1, i_2, \dots, i_m$ CR/LF ,

and the specified i_j row in the data shown on the screen will be deleted from the list. The program then proceeds to output (17) again.

If End is entered for (17), the program terminates the edit sequence and proceeds to Step 4.

STEP 4

The program clears the screen, produces the frequency vs. temperature plot and loss factor vs. temperature plot, and lists the data for viewing by the user. A computer selected T_o is indicated and at this time the user can examine the data closely.

The program will finally output (16) again and the user has the same options to edit the data as in Step 3. If the user chooses not to edit, at this point the program pauses and suspends by outputting:

VIBR2: PAUSE 0000
VIBR2 SUSP (22)
@

at this time the user can obtain copies of the screen for future reference. To continue the program the user inputs:

:GO CR/LF,

and proceeds to Step 5.

STEP 5

To begin the curve fit of the material data the program outputs:

CHANGE TO VALUE OR PARAMETER VALUES? (Y,N) (23)

for which the user inputs:

Yes CR/LF or No CR/LF.

If No is entered for (24), the program proceeds to Step 6.

If Yes is entered for (24), the program outputs:

ENTER TO VALUE (24)

for which the user inputs:

t₀ CR/LF,

the t₀ value for the material to the nearest 10°C by either user judgment or prior experience when testing the material. The program then proceeds to Step 6.

STEP 6

The program now fits the material modulus data to a specified curve. The program clears the screen and outputs:

FIT MODULUS DATA

ENTER THE MINIMUM VALUE OF FROM PARAMETER (25)

for which the user inputs:

x CR/LF,

the minimum value of the reduced frequency (FROM) parameter in Hz. This parameter corresponds to the value of $F_{\alpha T}$ at the inflection point of the modulus curve. The program then outputs:

ENTER THE MINIMUM VALUE OF MROM PARAMETER (26)

for which the user inputs:

x CR/LF,

the minimum value of the modulus (MROM) parameter in N/m^2 . This parameter corresponds to the value of the modulus curve at the inflection point.

The program then outputs:

ENTER THE MAXIMUM VALUE OF MROM PARAMETER (27)

for which the user inputs:

x CR/LF,

the maximum value of the modulus (MROM) parameter in N/m^2 . The program then outputs:

ENTER THE MINIMUM VALUE OF ML PARAMETER (28)

for which the user inputs:

x CR/LF,

the minimum value of the modulus (ML) parameter in N/m^2 . This parameter corresponds to the lower horizontal asymptote of the modulus curve.

The program proceeds to fit the modulus data to the specified modulus curve. The program then outputs the equation of the modulus curve, the parameter names, and the values of the parameters that most closely describe the curve fit for the modulus data. The program then outputs:

ACCEPT THESE VALUES?(Y,N) (29)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If Yes is entered for (29), the program proceeds to Step 8.

If No is entered for (29), the program outputs:

HOLD ANY PARAMETERS CONSTANT?(Y,N) (30)

for which the user inputs:

Yes CR/LF or No CR/LF.

If No is entered for (30), the program proceeds to Step 7.

If Yes is entered for (30), the program outputs:

ENTER THE NUMBER AND INDICES OF THE CONSTANT
PARAMETERS (31)

for which the user inputs:

n, i₁, i₂, ... i_n CR/LF,

where the i_j are the indices of the array elements that correspond to the indicated modulus parameters. Through experience, the user may want to hold parameters constant for a given material, therefore this option is available to the user. The program proceeds to Step 7.

STEP 7

The program outputs:

ENTER FROM(A1) PARAMETER VALUE (32)

for which the user inputs:

x CR/LF,

the user defined constant or variable value of the modulus (FROM) parameter.

The program then outputs:

ENTER MROM(A2) PARAMETER VALUE (33)

for which the user inputs:

x CR/LF,

the user defined constant or variable value of the modulus (MROM) parameter. The program then outputs:

ENTER N(A3) PARAMETER VALUE (34)

for which the user inputs:

x CR/LF,

the user defined constant or variable value of the modulus (N) parameter. This parameter corresponds to the slope of modulus curve at the point of inflection.

The program then outputs:

ENTER ML(A4) PARAMETER VALUE (35)

for which the user inputs:

x CR/LF,

the user defined constant or variable value of the modulus (ML) parameter. The program proceeds to curve fit the modulus data with these user defined constant and/or variable modulus parameter values. The program then outputs the equation of the modulus curve, the parameter names, and the values of the parameters that now most closely describe the curve fit for the modulus data. The program then proceeds to output (29) again for the user option input.

STEP 8

Next the program fits the material loss factor data to a specified curve. The program clears the screen and outputs:

FIT LOSS FACTOR DATA

ENTER THE MINIMUM VALUE OF ETAFROL PARAMETER (36)

for which the user inputs:

x CR/LF,

the minimum value of the loss factor (ETAFROL) parameter. This parameter corresponds to the value at the peak of the loss factor curve. The program then outputs:

ENTER THE MINIMUM VALUE OF SL PARAMETER (37)

for which the user inputs:

x CR/LF,

the minimum value of the slope (SL) parameter. This parameter corresponds to the slope of the asymptotic line for the low values of reduced frequency for the loss factor curve. The program then outputs:

ENTER THE MINIMUM VALUE OF FROL PARAMETER (39)

for which the user inputs:

x CR/LF,

the minimum value of the loss factor (FROL) parameter in Hz.

This parameter corresponds to the reduced frequency value at the peak of the loss factor curve.

The program proceeds to fit the loss factor data to the specified loss factor curve. The program then outputs the equation of the loss factor curve, the parameter names, and the values of the parameters that most closely describe the curve fit for the loss factor data. The program then outputs:

ACCEPT THESE VALUES?(Y,N) (40)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If Yes is entered for (40), the program proceeds to Step 10.

If No is entered for (40), the program outputs:

HOLD ANY PARAMETERS CONSTANT?(Y,N) (41)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If No is entered for (41), the program proceeds to Step 9.

If Yes is entered for (41), the program outputs:

ENTER THE NUMBER AND INDICES OF THE CONSTANT
PARAMETERS (42)

for which the user inputs:

n, i₁, i₂, ... i_n CR/LF,

where the i_j are the indices of the array elements that correspond to the indicated loss factor parameters. Through experience the user may want to hold parameters constant for a given material, therefore this option is available to the user. The program proceeds to Step 9.

STEP 9

The program outputs:

ENTER ETAFROL(B1) PARAMETER VALUE (43)

for which the user inputs:

x CR/LF ,

the user defined constant or variable value of the loss factor (ETAFROL) parameter. The program then outputs:

ENTER SL(B2) PARAMETER VALUE (44)

for which the user inputs:

x CR/LF ,

the user defined constant or variable value for loss factor (SL) parameter. The program then outputs:

ENTER SH(B3) PARAMETER VALUE (45)

for which the user inputs:

x CR/LF ,

the user defined constant or variable value for the loss factor (SH) parameter. The program then outputs:

ENTER FROL(B4) PARAMETER VALUE (46)

for which the user inputs:

x CR/LF ,

the user defined constant or variable value of the loss factor (FROL) parameter. The program then outputs:

ENTER C(B5) PARAMETER VALUE (47)

for which the user inputs:

x CR/LF ,

the user defined constant or variable value of the loss factor (C) parameter. This parameter corresponds to the shape of the peak of the loss factor curve (larger C implies flatter peak, smaller C implies sharper peak).

The program proceeds to curve fit the loss factor data with these user defined constant and/or variable loss factor parameter values. The program then outputs the equation of the loss factor curve, the parameter names, and the value of the parameters that now most closely describe the curve fit for the loss factor data. The program then proceeds to output (40) again for the user option input.

STEP 10

The program outputs:

```
ENTER THE INTERVAL OF TEMPERATURE  
LINES (48)
```

for which the user inputs:

```
x CR/LF
```

the interval of the displayed temperature lines. If the interval value entered is less than 20°C, the half-band width is assigned to 100°C. If the interval value entered is greater than 20°C, the half-band width is assigned to 200°C.

The desired graphs are then generated. Since the program has determined the values of the parameters for each curve and the user has accepted these values, the program clears the screen and generates the desired grid. The grid is generated in such a way that all the data points will fall within its bounds. The data points for both the modulus and loss factor are plotted as a function of reduced frequency. The curves for both are plotted using the parameters determined and being a function of reduced frequency. The axes are labeled with the correct label in metric units along with the scale for each axis.

The final plot is now complete and the program pauses and suspends by outputting:

```
VIBR2: PAUSE 0000  
VIBR2 SUSP  
@ (49)
```

the user can obtain copies of the screen for future reference. To continue the program the user then inputs:

:GO CR/LF .

The program clears the screen and outputs the material name, equations, parameter names, parameter values, and proceeds to pause and suspend by outputting:

VIBR2: PAUSE 0000

VIBR2 SUSP (50)

@

the user can obtain copies of the screen which identify the previous plot. To continue the program the user then inputs:

:GO CR/LF .

STEP 11

The program outputs (23) again and the user has the options of changing t_0 and/or the parameters given to define the relationship of the curves generated for the experimental data.

If Yes is entered for (23), the program goes to output (24) and proceeds from that point.

If No is entered for (23), the program outputs:

STORE THE DATA?(Y,N) (51)

for which the user inputs either:

Yes CR/LF or No CR/LF .

If No is entered for (51), the program proceeds to Step 12.

If Yes is entered for (51), the program outputs:

ABOVE PLOTTED DATA AND

PARAMETERS ARE STORED ON (52)

FILES: VIBR3, VIBR4, AND VIBR5

to indicate to the user the data has been stored on the designated files for further reference at another time. The program then pauses and suspends by outputting:

VIBR2: PAUSE 0000

VIBR2 SUSP

@

(53)

for which the user inputs:

:GO CR/LF,

to continue the program to Step 12.

STEP 12

The program outputs:

BEGIN ANOTHER PROBLEM? (Y,N)

(54)

for which the user inputs either:

Yes CR/LF or No CR/LF.

If Yes is entered for (54), the program proceeds to output (2) and the user begins another problem (i.e., enter or retrieve another material data set, etc.).

If No is entered for (54), the program terminates.

6.2 Sample Input/Output

The following pages contain example input/output for the program user manual described above. All the options cannot be presented in that the number of pages would be tremendous, so just brief examples of the input/output are presented to illustrate the features present in the program and described in the above user manual.

•:RU,VIBR2

Y LIST THE STORED DATA(Y,N)
 _ ENTER THE MAXIMUM NUMBER OF MATERIAL CODES

MATERIAL CODE : 2
 MATERIAL : UNOBTAINIUM
 DATA SOURCES
 MANUFACTURER : NONE
 AFIL : NONE
 OTHER : NONE

MATERIAL CODE : 3
 MATERIAL : 1467
 DATA SOURCES
 MANUFACTURER : NONE
 AFIL : NONE
 OTHER : NONE

MATERIAL CODE : 4
 MATERIAL : 1467
 DATA SOURCES
 MANUFACTURER : NONE
 AFIL : NONE
 OTHER : NONE

MATERIAL CODE : 7
 MATERIAL : 3M-467
 DATA SOURCES
 MANUFACTURER : NONE
 AFIL : JONES D.I.G.
 OTHER : WASHIP, A.D.

Y READ IN NEW DATA(Y,N)

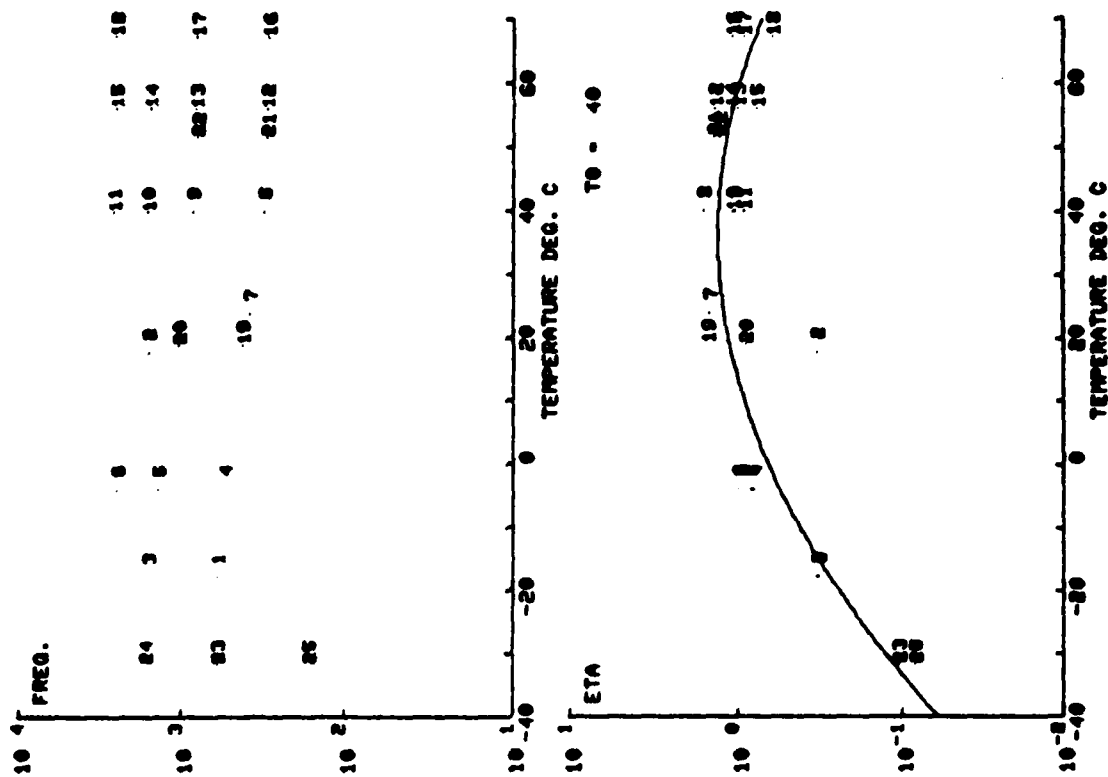
ENTER HEADER DATA
 ENTER THE CODE OF THE MATERIAL
 3 ENTER THE MATERIAL NAME(MAX 60 CHAR.)
 N467
 ENTER THE MANUFACTURER DATA SOURCE(MAX 60 CHAR.)
 NONE
 ENTER THE AFRL DATA SOURCE(MAX 60 CHAR.)
 NONE
 ENTER OTHER DATA SOURCES(MAX 60 CHAR.)
 NONE
 ENTER THE NUMBER OF DATA POINTS(MAX 64 PTS.)
 25
 IS THE DATA TO BE ENTERED METRIC(Y,N)
 Y
 IS THE MODULUS DATA SHEAR(Y,N)
 Y
 READY TO ENTER DATA FOR MATERIAL
 NO. MODULUS LOSS FACTOR DEG. C FREQ. MODE
 NO. N/Hz Hz Hz Hz
 1 6.64241E7 328 -17.8 597.2
 2 1.20775E8 335 -17.8 1609.3
 3 1.20775E8 335 -17.8 1609.3
 4 1.24655E7 803 -3.9 537.2
 5 2.4883E7 826 -3.9 1307.3
 6 2.8261E7 9922 -3.9 2521.4
 7 1.6271E6 1.4723 9 374.2
 8 4.92975E5 1.6148 40 313.8
 9 1.30311E6 1.6284 40 854.3
 10 1.7395E6 1.6764 40 1629.4
 11 2.4752E6 1.9159 40 2645.6
 12 2.8190E5 1.3756 1.239.2
 13 6.27423E5 987.56 1.814.3
 14 7.99792E5 1.093 56.1 1572.4
 15 1.50905E5 765.56 1.2583.5
 16 1.9374E5 1.077 67.5 203.2
 17 3.7852E5 916.67 2.788.3
 18 1.0204E6 622.67 2.2550.5
 19 2.54417E6 1.495 18.9 414.2
 20 5.5227E6 905 18.9 1042.3
 21 2.27527E5 1.394 51.7 206.2
 22 4.3643E5 1.28 51.7 804.2
 23 7.8138E7 1070 -31.7 600.2
 24 2.0409E8 885 -31.7 1049.3
 25 2.0409E8 885 -31.7 1049.4

MATERIAL CODE : 3
 MATERIAL IN467
 MANUFACTURE INCOME
 APPL INCOME
 OTHER INCOME

NO.	MODULUS N*REZ	LOSS FACTOR	TEMP. DEG. C	FREQ. HZ	MODE NO.
1	6.64241E+07	.3290	-17.8	597.0	1
2	1.20775E+08	.3350	-17.8	1600.0	2
3	1.20775E+08	.3350	-17.8	1600.0	3
4	1.34655E+07	.8030	-3.9	1397.0	4
5	2.46832E+07	.8250	-3.9	2521.0	5
6	2.82616E+07	.9922	23.9	3774.0	6
7	1.62716E+06	1.4700	40.0	313.0	7
8	4.92975E+05	1.6100	40.0	854.0	8
9	1.30311E+06	1.6280	40.0	1629.0	9
10	1.73058E+06	1.6760	40.0	2645.0	10
11	2.47522E+06	.9160	56.1	290.0	11
12	2.81996E+05	1.3750	56.1	814.0	12
13	6.27422E+05	.9870	56.1	1572.0	13
14	7.99522E+05	1.9930	56.1	2583.0	14
15	1.50995E+06	1.7650	67.2	798.0	15
16	1.93743E+05	1.0770	67.2	2550.0	16
17	3.78522E+05	.9160	67.2	414.0	17
18	1.02042E+06	.6320	18.9	1042.0	18
19	2.54417E+06	1.4950	18.9	206.0	19
20	5.52270E+06	.9050	51.7	804.0	20
21	2.27527E+05	1.3940	51.7	600.0	21
22	4.36438E+05	1.2800	51.7	1640.0	22
23	7.81383E+07	.1070	-31.7	1640.0	23
24	2.04092E+08	.0850	-31.7	1640.0	24
25	2.04092E+08	.0850	-31.7	1640.0	25

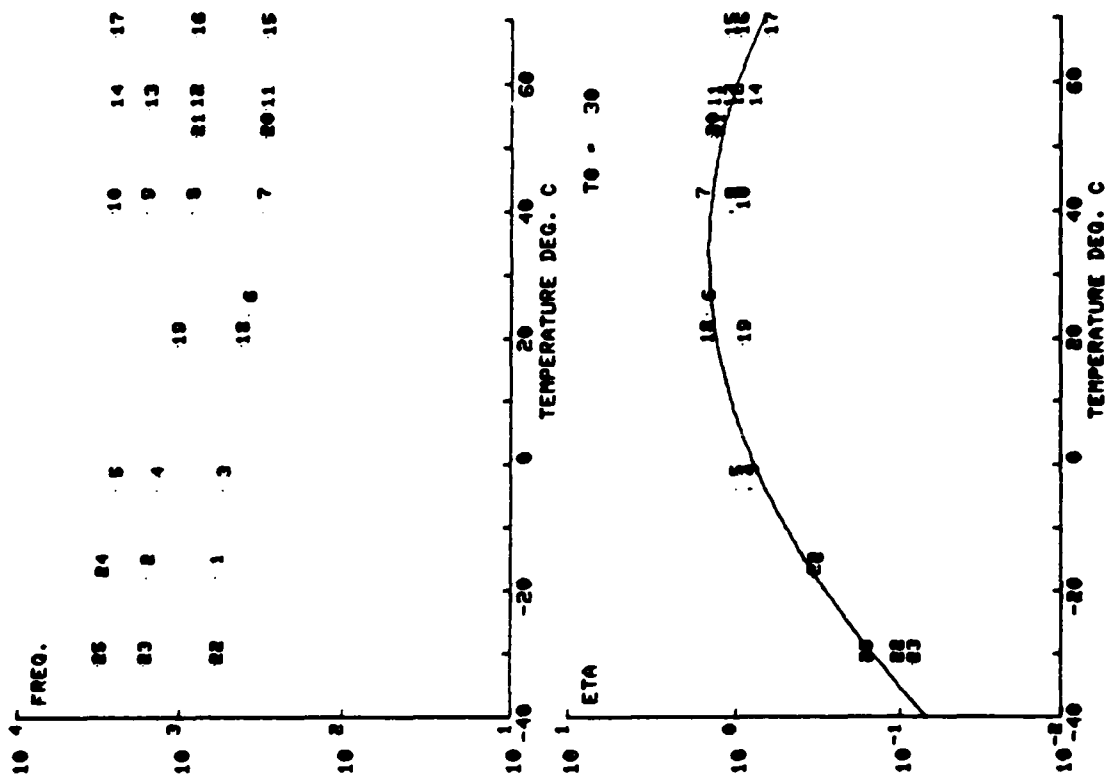
CUFIT 1 PAUSE 0000
 CUFIT SUSP
 C: GO
 N EDIT THE DATA?(Y,N)

MATERIAL CODE : 3
 MATERIAL IN467
 NO. MODULUS LOSS FREQ. TEMP. MODE
 N M112 DEG. C NO.
 1 6.64241E+07 .3280 -17.8 8 597.0 8
 2 1.20775E+08 .3350 -17.8 8 1000.0 3
 3 1.34775E+08 .3350 -17.8 8 1000.0 3
 4 1.34655E+07 .8250 -3.9 8 1537.0 3
 5 2.48332E+07 .9322 -3.9 8 1537.0 3
 6 2.82616E+07 1.4700 23.9 8 2521.0 4
 7 1.62716E+06 1.6100 40.0 8 374.0 2
 8 1.39311E+06 1.0280 40.0 8 313.0 8
 9 1.73058E+06 1.0760 40.0 8 254.0 4
 10 1.47522E+06 1.9160 40.0 8 162.0 4
 11 2.81996E+05 1.3750 56.1 8 2845.0 5
 12 6.27423E+05 1.9870 56.1 8 299.0 2
 13 7.99292E+05 1.0930 56.1 8 814.0 3
 14 1.50952E+06 1.7650 56.1 8 1572.0 4
 15 1.93743E+05 1.0770 67.2 8 2583.0 5
 16 3.78522E+05 1.9160 67.2 8 293.0 2
 17 1.02042E+06 1.6220 67.2 8 798.0 3
 18 2.54417E+06 1.4950 18.9 8 414.0 5
 19 5.5270E+06 1.9050 18.9 8 1042.0 3
 20 2.27527E+05 1.3940 51.7 8 206.0 2
 21 4.36438E+05 1.2800 51.7 8 804.0 3
 22 7.81383E+07 1.070 31.7 8 600.0 2
 23 2.94892E+08 .0850 -31.7 8 1640.0 3
 24 2.04892E+08 .0850 -31.7 8 164.0 3
 25 EDIT THE DATA?(Y,N)
 Y EDIT THE ABOVE DATA: ADD, DELETE, OR END?(A,D,E)
 D ENTER THE NUMBER OF DATA POINTS
 TO DELETE AND THE INDICES OF THE
 ROWS TO DELETE
 2 2,25
 A EDIT THE ABOVE DATA: ADD, DELETE, OR END?(A,D,E)
 A ENTER THE NUMBER OF DATA POINTS TO ADD
 2 IS THE DATA TO BE ADDED METRIC?(Y,N)
 Y READY TO ADD DATA FOR MATERIAL
 IN FORMAT SHOWN ABOVE
 1 1.43004E8,-3410,-17.8,3043.-4.
 2 3.01184E8,-168,-31.7,3194.-4.
 EDIT THE ABOVE DATA: ADD, DELETE, OR END?(A,D,E)
 E



MATERIAL CODE 1 3
 MATERIAL IN457
 NO. MODULUS LOSS FREQ. TEMP. MODE
 N MPR FACTOR HZ DEG. C NO.
 1 6.64241E+07 3800 -17.8 8 2
 2 1.20775E+08 3750 -17.8 8 2
 3 1.34655E+07 8020 -3.9 3 2
 4 2.46832E+07 8250 -3.9 3 2
 5 2.82616E+07 9922 23.9 4 2
 6 1.62716E+06 1.4700 40.0 2 2
 7 4.92975E+06 1.6100 40.0 2 2
 8 1.30311E+06 1.8200 40.0 2 2
 9 1.3058E+06 1.0700 40.0 2 2
 10 2.47522E+06 1.9160 56.1 5 2
 11 2.81964E+05 1.3750 56.1 5 2
 12 6.27423E+05 1.9870 56.1 5 2
 13 7.59792E+05 1.0930 56.1 5 2
 14 1.50995E+06 1.7550 67.2 5 2
 15 1.53743E+05 1.0770 67.2 5 2
 16 3.78522E+05 1.9160 67.2 5 2
 17 1.02042E+06 1.6220 67.2 5 2
 18 2.54417E+06 1.4950 18.9 3 2
 19 5.22270E+06 1.9050 18.9 3 2
 20 2.27527E+05 1.3940 51.7 3 2
 21 4.36438E+05 1.2000 51.7 3 2
 22 7.81383E+07 1.0700 -31.7 3 2
 23 2.04092E+08 1.0700 -31.7 3 2
 24 1.43004E+08 1.0700 -31.7 3 2
 25 3.01184E+08 1.0700 -31.7 3 2
 EDIT THE DATA(Y,N)
 N CUFIT : PAUSE 0000
 CUFIT SUSP

0:GO
 N CHANGE TO VALUE OR PARAMETER VALUES(Y,N)



```

FIT MODULUS DATA
ENTER THE MINIMUM VALUE OF FROM PARAMETER
.0 ENTER THE MINIMUM VALUE OF FROM PARAMETER
689500.
ENTER THE MAXIMUM VALUE OF FROM PARAMETER
689500.
ENTER THE MINIMUM VALUE OF ML PARAMETER
6895.
LOG(M)-LOG(ML)+(2*LOG(WROR/ML))/(1+(FROM/FR)28N)
TO C      A1      A2      A3      A4
          39.0  2.8421E+03  4.5595E+06  .187  6.8067E+03
ACCEPT THESE VALUES?(Y,N)

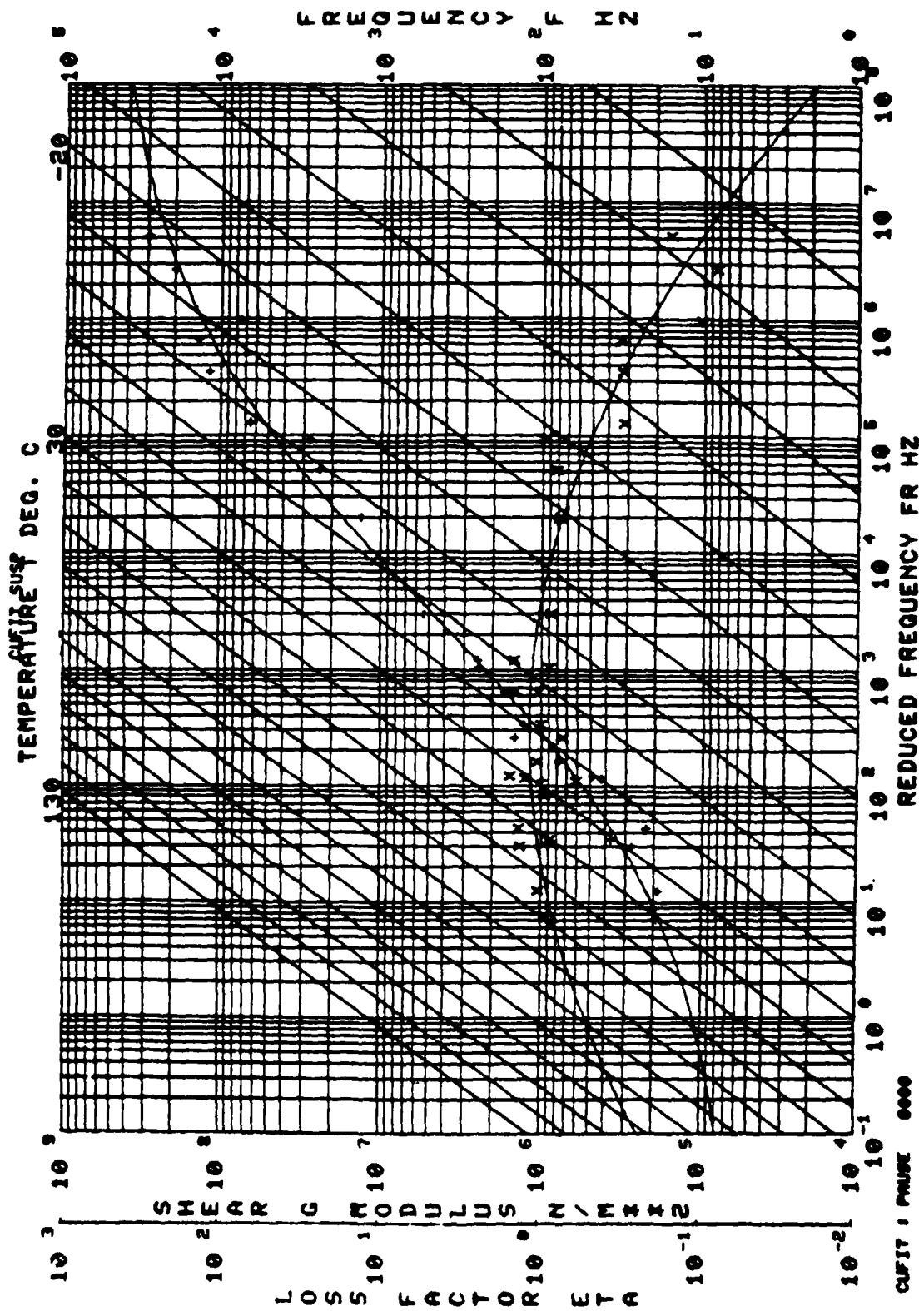
N HOLD ANY PARAMETERS CONSTANT?(Y,N)
Y
ENTER THE NUMBER AND INDICES OF
THE CONSTANT PARAMETERS
1.3 ENTER FROM(A1) PARAMETER VALUE
2.8E3
ENTER WROR(A2) PARAMETER VALUE
4.559E6
ENTER N(A3) PARAMETER VALUE
.3
ENTER ML(A4) PARAMETER VALUE
6.8067E3
LOG(M)-LOG(ML)+(2*LOG(WROR/ML))/(1+(FROM/FR)28N)
TO C      A1      A2      A3      A4
          39.0  3.8976E+03  5.6656E+06  .300  5.4615E+04
ACCEPT THESE VALUES?(Y,N)
Y

```

```

FIT LOSS FACTOR DATA
ENTER THE MINIMUM VALUE OF ETAFROL PARAMETER
.SS
ENTER THE MINIMUM VALUE OF SL PARAMETER
.EE
ENTER THE MAXIMUM VALUE OF SH PARAMETER
-1
ENTER THE MINIMUM VALUE OF FROL PARAMETER
.015
A=LOG(FR)-LOG(FROL))/C
LOG(ETA)-LOG(ETAFROL)+((SL+SH)*(SL-SH)*(1-SORT((1+A222)))C/2
TO C ETAFROL SL SH FROL C
B1 B2 B3 B4 B5
30.0 1.067 1.710 -1.916 2.1720E+03 15.302
ACCEPT THESE VALUES(Y,N)
N HOLD ANY PARAMETERS CONSTANT?(Y,N)
N
ENTER ETAFROL(B1) PARAMETER VALUE
.75
ENTER SL(B2) PARAMETER VALUE
2.
ENTER SH(B3) PARAMETER VALUE
-2.
ENTER FROL(B4) PARAMETER VALUE
2.172E4
ENTER C(B5) PARAMETER VALUE
4.
A=LOG(FR)-LOG(FROL))/C
LOG(ETA)-LOG(ETAFROL)+((SL+SH)*(SL-SH)*(1-SORT((1+A222)))C/2
TO C ETAFROL SL SH FROL C
B1 B2 B3 B4 B5
30.0 .742 1.972 -2.411 2.1271E+04 18.483
ACCEPT THESE VALUES(Y,N)
Y
ENTER THE INTERVAL OF
TEMPERATURE LINES
10

```



NOTE: Label to attach to graph
on previous page.

```

MATERIAL CODE: 3
MATERIAL NAME:
LOG(M) = LOG(M) * ((LOG(M) * M) / (1 + (FRON / FR) * M))
TO C FROM M A1 A2 A3 A4
30.0 3.897E+03 5.405E+05 .300 5.405E+04
A = (LOG(FR) - LOG(FR)) / C
LOG(ETA) = LOG(ETA * FROL) * ((SL * M) * ((SL * M) * (1 - SORT(1 + AS * M))) / C) / B
TO C ETAFROL SL SM FROL C
30.0 B1 742 1.978 -2.411 2.187E+04 18.483
LOG(FR) = LOG(F) - (1217 - T0) / (525 / 1.2 * T - T0)

```

```

CUEIT 1 PAUSE 0000
CUEIT 5USP
0:GO
N CHANGE TO VALUE OR PARAMETER VALUES(V,M)
N STORE THE DATA(V,M)
N BEGIN ANOTHER PROBLEM(V,M)
Y

```

LIST THE STORED DATA?(Y,N)
Y ENTER THE MAXIMUM NUMBER OF MATERIAL CODES

7

MATERIAL CODE : 2
MATERIAL : UNOBTAINIUM
DATA SOURCES
MANUFACTURER : NONE
AFIL : NONE
OTHER : NONE

MATERIAL CODE : 3
MATERIAL : M457
DATA SOURCES
MANUFACTURER : NONE
AFIL : NONE
OTHER : NONE

MATERIAL CODE : 4
MATERIAL : J457
DATA SOURCES
MANUFACTURER : NONE
AFIL : NONE
OTHER : NONE

MATERIAL CODE : 7
MATERIAL : 3A-457
DATA SOURCES
MANUFACTURER : NONE
AFIL : JONES D.I.G.
OTHER : NASHIF, A.D.

READ IN NEW DATA?(Y,N)

N ENTER THE CODE OF RETRIEVAL MATERIAL

4

MATERIAL CODE : 4
 MATERIAL : 1467
 MANUFACTURER DATA SOURCES
 AFPL INONE
 OTHER INONE

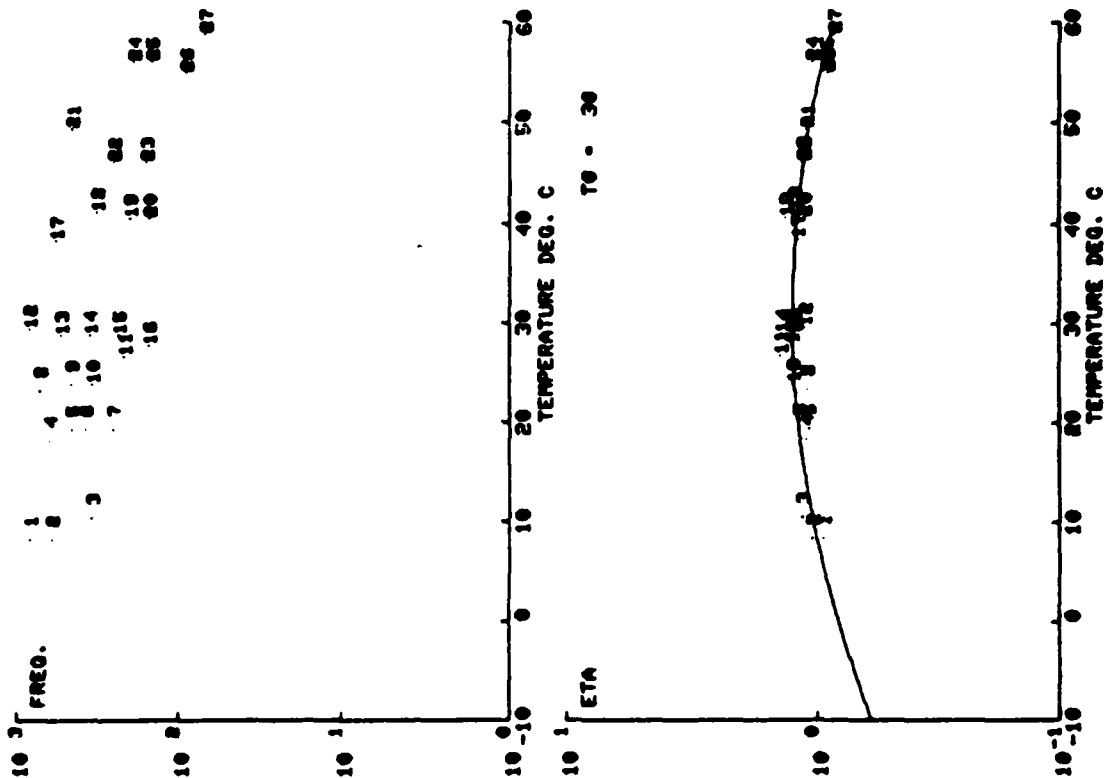
NO.	REMARKS	LOSS FACTOR	TEMP. REQ. C	FREQ. HZ	MODE NO.
1	1.2895E+07	1.3500	8.33	148.0	8
2	1.05697E+07	1.0500	8.33	14.0	8
3	5.37791E+06	1.1000	10.66	344.0	8
4	3.81290E+06	1.1100	18.33	636.0	8
5	3.83340E+06	1.0000	19.44	468.0	8
6	4.01854E+06	1.1000	19.44	379.0	8
7	3.1233E+06	1.1000	19.44	557.0	8
8	2.8582E+06	1.1000	23.33	736.0	8
9	2.06843E+06	1.2500	23.33	467.0	8
10	2.1500E+06	1.2500	23.33	346.0	8
11	1.3153E+06	1.4100	25.77	217.0	8
12	1.6892E+06	1.1200	28.44	53.0	8
13	1.55821E+06	1.2300	28.44	543.0	8
14	1.22037E+06	1.3000	28.44	360.0	8
15	1.00563E+06	1.2700	28.44	236.0	8
16	1.1100E+06	1.2000	27.83	153.0	8
17	8.0688E+05	1.2000	38.33	582.0	8
18	5.4468E+05	1.2300	41.11	322.0	8
19	3.76454E+05	1.3300	40.66	201.0	8
20	4.2747E+05	1.1200	40.66	154.0	8
21	4.9643E+05	1.1000	49.44	464.0	8
22	2.4087E+05	1.1200	46.11	257.0	8
23	2.3442E+05	1.1400	46.11	158.0	8
24	1.9829E+05	1.0400	56.11	190.0	8
25	2.06843E+05	.9100	56.11	148.0	8
26	1.74437E+05	.9000	55.00	90.0	8
27	1.3785E+05	.8500	53.00	70.0	8

0:60
 EDIY THE DATA7(V,N)
 N

MATERIAL CODE : 4
 MATERIAL 11487
 NO. MODULUS N RHE LOSS FACTOR TEMP. DEG. C FREQ. HZ. MODE NO.

1	1.2856E+07	1.3500	8.3	848.0	2
2	1.0597E+07	1.0500	8.3	614.0	2
3	5.3779E+06	1.1600	10.6	344.0	2
4	3.8128E+06	1.1100	18.3	236.0	2
5	3.8372E+06	1.0800	18.4	182.0	2
6	4.0194E+06	1.1800	18.4	139.0	2
7	3.1233E+06	1.1800	23.3	97.0	2
8	2.8622E+06	1.1000	23.3	738.0	2
9	2.0684E+06	1.2500	23.3	467.0	2
10	2.1586E+06	1.4100	23.7	346.0	2
11	1.3185E+06	1.1200	23.7	217.0	2
12	1.6882E+06	1.1200	23.7	153.0	2
13	1.5582E+06	1.2300	23.9	543.0	2
14	1.2207E+06	1.2300	23.9	360.0	2
15	1.0063E+06	1.2700	23.9	236.0	2
16	1.1066E+06	1.2600	23.8	152.0	2
17	8.0686E+05	1.2000	38.3	592.0	2
18	3.7645E+05	1.2300	41.1	322.0	2
19	5.4686E+05	1.3300	41.6	201.0	2
20	4.2745E+05	1.1200	40.6	154.0	2
21	4.9642E+05	1.1000	40.4	115.0	2
22	3.4875E+05	1.1200	46.1	257.0	2
23	2.3422E+05	1.1400	46.1	158.0	2
24	1.9035E+05	1.0400	56.1	190.0	2
25	2.0684E+05	1.0100	55.0	148.0	2
26	1.7437E+05	1.0000	55.0	90.0	2
27	1.3785E+05	1.3500	53.9	70.0	2

EDIT THE DATA(Y,N)
 N CUFIT 1 PAUSE 0000
 CUFIT SUSP
 0:00
 CHANGE TO VALUE OR PARAMETER VALUES(Y,N)
 N



```

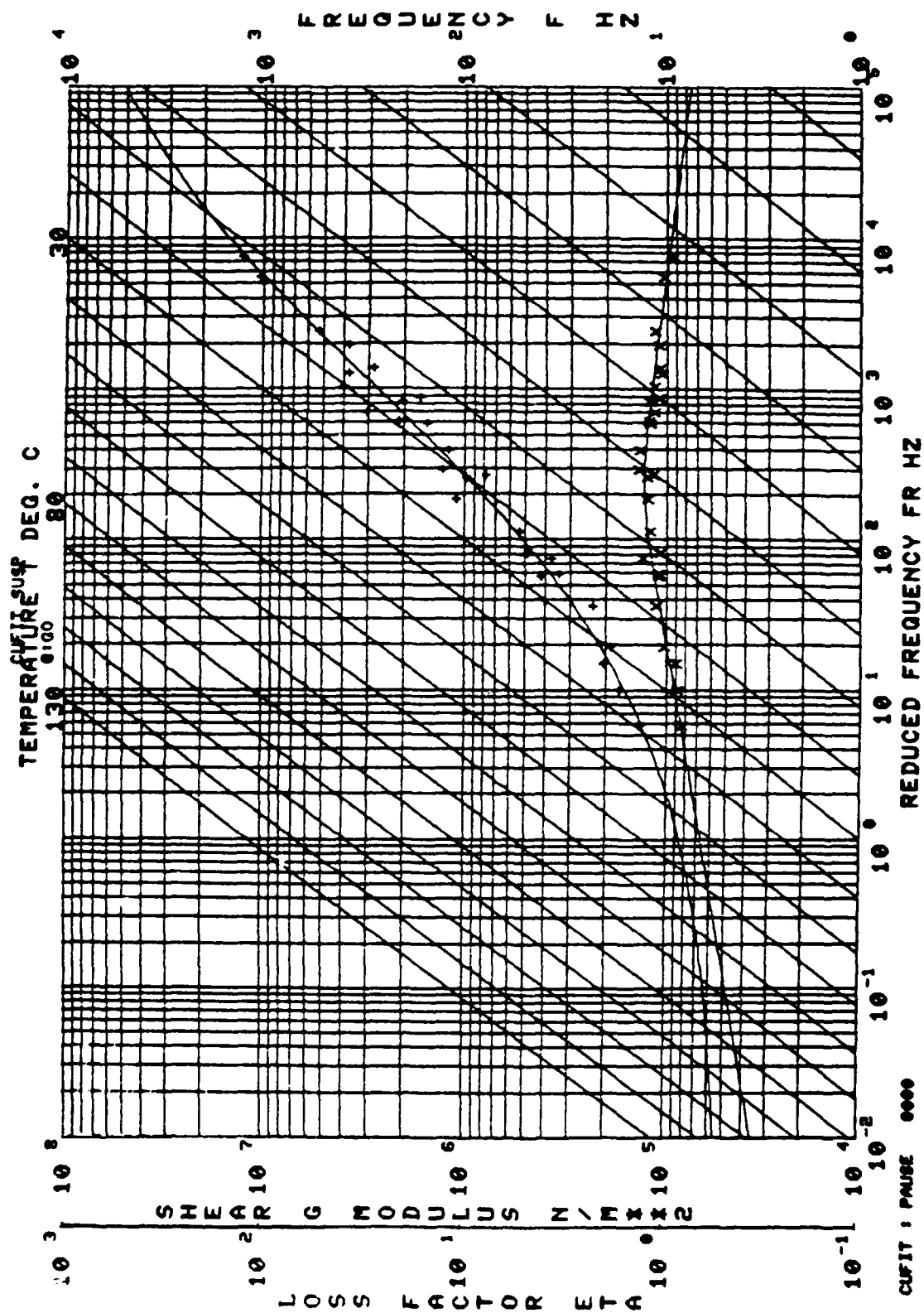
FIT NORMAL DATA
ENTER THE MINIMUM VALUE OF FROM PARAMETER
.01
ENTER THE MINIMUM VALUE OF FROM PARAMETER
689500.
ENTER THE MAXIMUM VALUE OF FROM PARAMETER
68950000.
ENTER THE MINIMUM VALUE OF ML PARAMETER
6895.
LOG(M)=-LOG(ML)+((2*LOG(MFROM/ML))/((1+(FROM/FR)32N)
TO C FROM MFROM N ML
A1 A2 A3 A4
30.0 1.157E+03 3.0340E+06 .379 5.041E+04
ACCEPT THESE VALUES?(Y,N)
Y

```

```

FIT LOSS FACTOR DATA
ENTER THE MINIMUM VALUE OF ETAFROL PARAMETER
.5
ENTER THE MINIMUM VALUE OF SL PARAMETER
.1
ENTER THE MAXIMUM VALUE OF SH PARAMETER
-.1
ENTER THE MINIMUM VALUE OF FROL PARAMETER
.01
A. LOG(1-FROL)/C
LOG.ETA-LOG(ETAFROL)+((SL+SH)*((1-SORT(1+ASB)))C/B
TO 0 ETAFROL SL SH FROL C B3 B4 B5
30.0 1.295 B2 137 -.100 2.808E+02 .284
ACCEPT THESE VALUES?(Y,N)
Y ENTER THE INTERVAL OF
TEMPERATURE LINES
10

```



NOTE: Label to attach to graph
on previous page.

```

PARAMETER CODE : 4
PARAMETER 1,467
LOG(M)=LOG(MN)*(2*LOG(MNOR/PL))/(1+(FROM/FR)2*2N)
TO C FROM N A3 A4 A5
30.0 1.157E+03 3.034E+06 .379 5.041E+04
A=(LOG(PW)-LOG(FROL))/C
LOG(ETA)=LOG(ETAFROL)*((SL+SH)*A+(SL-SH)*(1-SORT(1+ABS(2))))C/2
TO C ETAFROL SL SH FROL C
30.0 1.295 B3 B4 B5
LOG(FR)=LOG(F)-12*(T-T0)/(SJS/1.8+T-T0) .284

```

CUFFIT : PAUSE 0000
CUFFIT SUSP

0:60
CHANGE TO VALUE OR PARAMETER VALUES?(Y,N)

N STORE THE DATA?(Y,N)

Y ABOVE PLOTTED DATA AND
PARAMETERS ARE STORED ON
FILES: U1B03, U1B04, AND U1B05
CUFFIT : PAUSE 0000
CUFFIT SUSP

0:60
BEGIN ANOTHER PROBLEM?(Y,N)
Y

LIST THE STORED DATAP(Y,N)
N
READ IN NEW DATAP(Y,N)
N
ENTER THE CODE OF RETRIEVAL MATERIAL
7

MATERIAL CODE : ?
 MATERIAL : 3A-487
 DATA SOURCES
 MANUFACTURER INNOV
 AFRL JONES D.I.G.
 OTHER INSNIF. A.D.

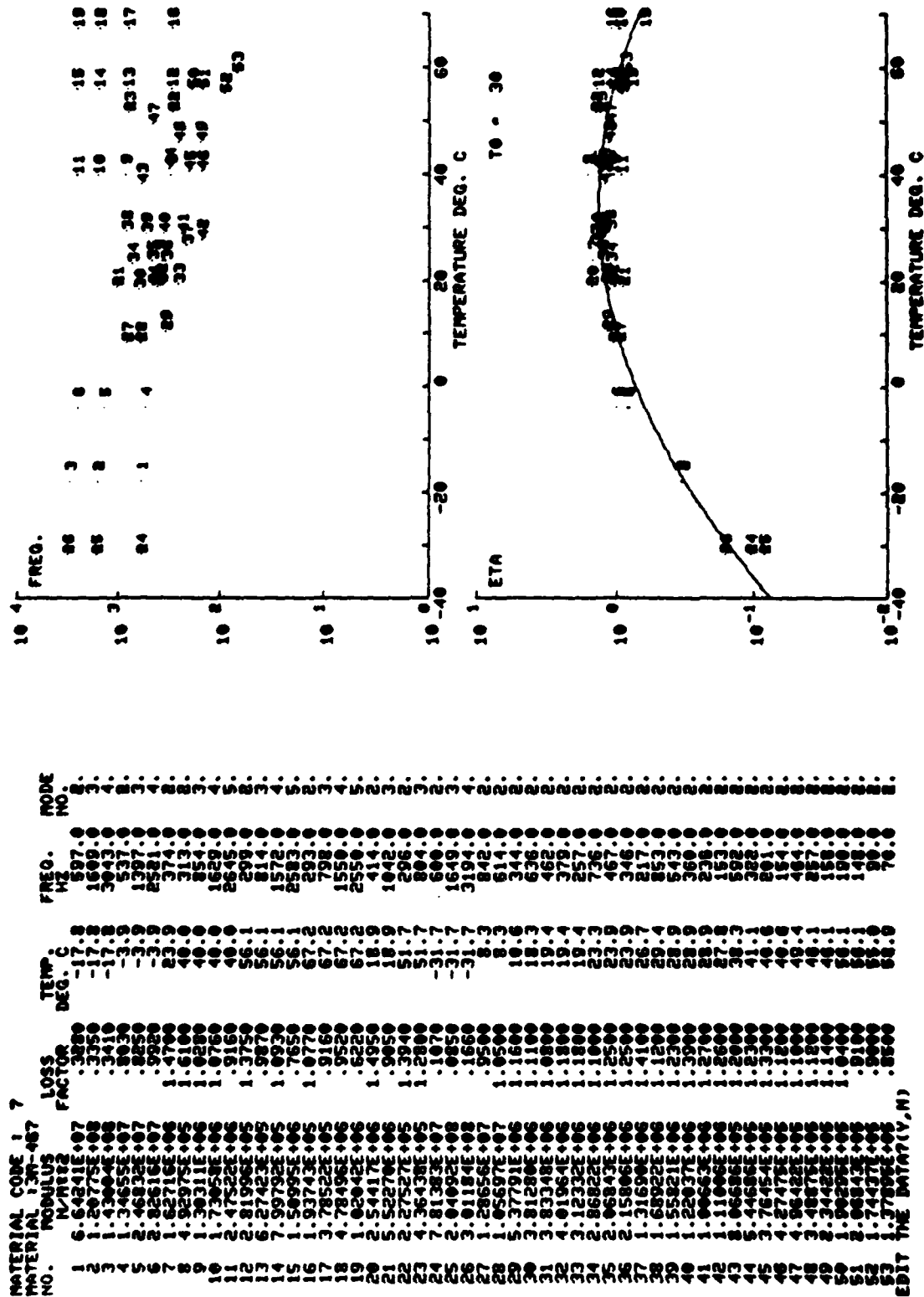
50 1.90805E+05 1.0400
 51 8.06843E+05 .9100
 52 1.74437E+05 .9000
 53 1.37891E+05 .8500
 CUFIT : PAUSE 0000
 CUFIT SUSP

50.1
 51.1
 52.0
 53.0

100.0
 140.0
 70.0

NO.	MODULUS N.MHR2	LOSS FACTOR	TEMP. DEG. C	FREQ. HZ	MODE NO.
1	6.44341E+07	3800	-17.8	597.0	2
2	1.20775E+08	3750	-17.8	1608.0	3
3	1.43804E+08	3410	-17.8	3993.0	4
4	1.34855E+07	8020	-3.9	5237.0	5
5	2.48332E+07	8250	-3.9	1597.0	6
6	2.82610E+07	9020	-3.9	2521.0	7
7	1.62710E+06	1.4700	23.9	374.0	8
8	4.95975E+05	1.6100	40.0	313.0	9
9	1.30311E+06	1.8200	40.0	854.0	10
10	1.73958E+06	1.0700	40.0	1608.0	11
11	1.47522E+06	1.9100	40.0	2645.0	12
12	2.81956E+05	1.3750	56.1	299.0	13
13	6.27423E+05	1.9870	56.1	814.0	14
14	7.99792E+05	1.0930	56.1	1522.0	15
15	1.50952E+06	1.7550	56.1	2583.0	16
16	1.93743E+05	1.0770	67.2	293.0	17
17	3.78522E+05	1.9160	67.2	798.0	18
18	4.78496E+06	1.9520	67.2	1550.0	19
19	1.05042E+06	1.6220	67.2	2550.0	20
20	3.54417E+06	1.4950	18.9	414.0	21
21	5.2270E+06	1.9050	18.9	1020.0	22
22	2.27527E+05	1.3940	51.7	286.0	23
23	4.36438E+07	1.2800	51.7	804.0	24
24	7.81383E+07	1.1070	51.7	1608.0	25
25	3.04092E+08	1.0850	31.7	319.0	26
26	3.91184E+08	1.5600	31.7	519.0	27
27	1.28656E+07	1.0500	8.3	842.0	28
28	1.05697E+07	1.1600	8.3	1314.0	29
29	5.37791E+06	1.1100	10.6	2140.0	30
30	3.81280E+06	1.1100	18.7	3500.0	31
31	3.37348E+06	1.0800	19.4	4620.0	32
32	4.01664E+06	1.1800	19.4	7390.0	33
33	3.12332E+06	1.1000	19.4	11900.0	34
34	2.96822E+06	1.1200	23.3	12500.0	35
35	2.06843E+06	1.2500	23.3	14700.0	36
36	2.15800E+06	1.2500	23.3	17000.0	37
37	1.31890E+06	1.4100	26.7	22200.0	38
38	1.08822E+06	1.1200	26.7	35000.0	39
39	1.58821E+06	1.2300	28.4	56400.0	40
40	1.22037E+06	1.3000	28.4	87000.0	41
41	1.06063E+06	1.2700	28.4	130000.0	42
42	1.11000E+06	1.1800	28.4	200000.0	43
43	8.00000E+05	1.2300	30.3	300000.0	44
44	5.40000E+05	1.3300	41.1	450000.0	45
45	3.70464E+05	1.3300	40.0	680000.0	46
46	4.27476E+05	1.1800	40.0	1010000.0	47
47	4.00422E+05	1.1800	40.0	1510000.0	48
48	3.48876E+05	1.1100	40.0	2210000.0	49
49	2.24820E+05	1.1400	40.0	3310000.0	50

0:00
 EDIT THE DATA(Y,N)
 N



N CUFIT : PAUSE 0000
CUFIT SUSP
0100
N CHANGE TO VALUE OR PARAMETER VALUES(Y,N)

```

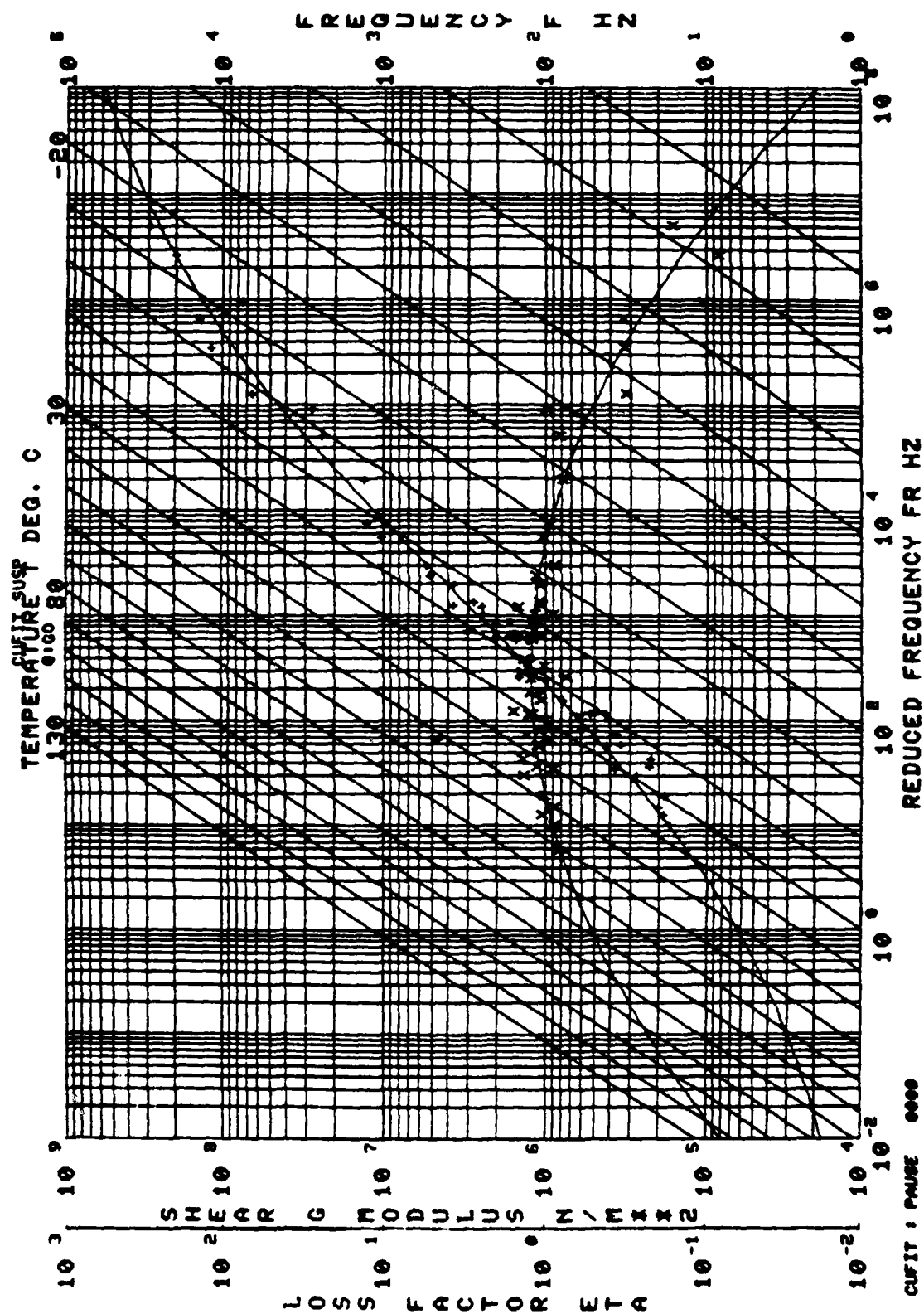
FIT MODULUS DATA
ENTER THE MINIMUM VALUE OF FROM PARAMETER
.01 ENTER THE MINIMUM VALUE OF FROM PARAMETER
ES9500.
ENTER THE MAXIMUM VALUE OF FROM PARAMETER
65950000.
ENTER THE MINIMUM VALUE OF HL PARAMETER
LOG(M)=-LOG(ML)/((2*LOG(MROR/ML))/((1+(FROM/FR)22N)
TO 0 FROM MROR N HL
A1 A2 A3 A4
30.0 1.6169E+03 3.5442E+06 .208 6.9029E+03
ACCEPT THESE VALUES(Y,N)
Y

```

```

FIT LOSS FACTOR DATA
ENTER THE MINIMUM VALUE OF ETAFROL PARAMETER
.5
ENTER THE MINIMUM VALUE OF SL PARAMETER
.1
ENTER THE MINIMUM VALUE OF SH PARAMETER
-.1
ENTER THE MINIMUM VALUE OF FROL PARAMETER
.01
A. LOG(IFR'-LOG(FROL))/C
LOG(ETA)-LOG(ETAFROL)+((SL+SH)/A+(SL-SH)/(1-SORT((1+ABS2)))C/2
TO 0 ETAFROL SL SH FROL C
30.0 B1 B2 B3 B4 B5
ACCEPT THESE VALUES(V,N)
-1.840 1.6940E+03 14.336
Y
ENTER THE INTERVAL OF
TEMPERATURE LINES
13

```



CUT : PAUSE 0000

NOTE: Label to attach to graph
on previous page.

```

MATERIAL CODE : 7
MATERIAL 130-487
LOG(M)=LOG(ML)*((SL*LOG(MROR/ML))/((1+(FROM/FR))2SH))
      FROM M ML
      A1 A2 A3 A4
      20.0 1.616E+03 3.544E+06 .000 6.000E+03
      A=(LOG(FR)-LOG(FROL))/C
      LOG(ETA)-LOG(ETAFROL)*((SL*SH)*((SL-SH)*(1-SORT(1+ABS2)))C/2
      TO C ETAFROL SL SH FROL C
      20.0 1.12E 1.000 -1.840 1.034E+03 14.336
      LOG(FR)-LOG(F)-12(T-T0)/(SES/1.8+T-T0)

```

CUFIT : PAUSE 0000
CUFIT SUSP

0:GO
CHANGE TO VALUE OR PARAMETER VALUES?(V,M)

M STORE THE DATA?(V,M)

V ABOVE PLOTTED DATA AND
PARAMETERS ARE STORED ON
FILES: UIRN3, UIRN4, AND UIRN5
CUFIT : PAUSE 0000
CUFIT SUSP

0:GO
BEGIN ANOTHER PROBLEM?(V,M)

CUFIT : STOP 0000

SECTION VII

SUMMARY AND RECOMMENDATIONS

The computer program described in this report is operational on the computer equipment at AFML. The program is being used to reduce, edit, and graphically represent viscoelastic material properties data for further development of a standardized material properties handbook.

The program currently requires practically the entire available core of the computer equipment when executing for the user. Therefore, any modifications to the computer program may be costly if they require the use of more core than is currently available for the equipment. If modifications are required which would increase the program size beyond the available core, the program would have to be split into suitable segments. By splitting the program into suitable segments, it could be executed in the available core of the equipment. Segmentation allows for the execution of segments as needed by the program, thus reducing the overall size of the executing program. Segmentation is available on the AFML equipment.

Future recommendations for this computer program would be to give the user the option of defining or using one of several curve-fit equations for the modulus and/or loss factor data. On this case, the user could select the equations that he feels best describe the material data. Also, the user should have the option of choosing a scheme to find the reference temperature (t_0) for the material, and in some way to find the shift parameter (a_t) curve that best describes the shift needed for each material. These capabilities if added to the program could give the user a better representation of the viscoelastic material properties data.

Several further considerations are the implementation of features into this program, or possible other programs, to use the

computer to control the experimental test facility. The use of the computer for all data recording, data processing, data editing, and data representation would reduce the time needed to collect the data, and also take over many more manual operations currently being performed. Through this major task, reported herein, the use of the computer for all the data processing operations has become more feasible. Future developments include the integration of the computer in to the test facility, and programs implemented on the computer to assist the user to effectively record data from the test facility, and proceed through the representation of the data by the computer. The amount of time spent by the user, therefore, would be reduced, because the user now will have a fast digital computer to accomplish in minutes many of the operations the user would currently do in several hours.

REFERENCES

1. Ferry, J.E., Viscoelastic Properties of Polymers. John Wiley and Sons, Inc., New York, 1961.
2. Jones, D.I.G., "A Reduced Temperature Nomogram for Characterization of Damping Material Behavior," Shock and Vibration Bulletin 48, Part 2, 1978.
3. Nashif, A.D. and L.C. Rogers, "Computerized Processing and Empirical Representation of Viscoelastic Material Property Data and Preliminary Constrained Layer Damping Treatment Design," 48th Shock and Vibration Symposium, October, 1977.
4. Nashif, A.D., "A New Method for Determining Damping Properties of Viscoelastic Materials," Shock and Vibration Bulletin 36, 1967.
5. Jones, D.I.G., "Temperature-Frequency Dependence of Dynamic Properties of Damping Materials," J. Sound and Vibration 33 (4), pp 451-470, 1974.
6. Plot-10 Terminal Control System; User's Manual, Tektronix, Inc., Beaverton, Oregon, 1974.
7. Nelder, J.A. and R. Mead, "A Simplex Method for Function Minimization," The Computer Journal, Vol. 7, No. 1, April, 1964.

APPENDIX A PARAMETER NAME DESCRIPTIONS

Following is a list of the parameter names used for the modulus and loss factor curves, indicating the units, and describing the parameter for the appropriate curve.

Name	Units	
TO	°C	Reference temperature to shift and minimize scatter for modulus and loss factor curves (t_0).
F	HZ	Frequency of observation (f).
FR	HZ	Frequency of observation reduced by the shift factor induced by the reference temperature ($f_r = \alpha_T f$).
T	°C	Temperature of observation (t).

MODULUS CURVE:

M	N/m ²	Observed modulus for material (M).
FROM	HZ	Reduced frequency at the inflection point of the modulus curve (f_{rom}).
MROM	N/m ²	Value of the modulus curve at the inflection point (M_{rom}).
N	-	Shape parameter for the modulus curve (N).
ML	N/m ²	Value of the lower asymptote for the modulus curve (M_l).

LOSS FACTOR CURVE:

ETA	-	Observed loss factor for material (η).
ETAFROL	-	Value of the loss factor curve at the peak (η_{frol}).
SL	-	Slope of asymptotic line for low values of reduced frequency (S_l).

Name	Units	
SH	-	Slope of asymptotic line for high values of reduced frequency (S_h).
FROL	HZ	Reduced frequency at the peak of the loss factor curve (f_{rol}).
C	-	Shape parameter for the loss factor curve (C).

APPENDIX B
PROGRAM LISTING

This section contains a compiled listing of the program described. This program has been compiled on the CDC computer system at Wright-Patterson Air Force Base, because of the inability of the HP-2100 and equipment available to present a neat and clean compiled listing of the program. There are no differences in the FORTRAN source presented herein and that stored on the HP-2100 computer system with permanent file name VIBR1.

FTN4

```
      PROGRAM VIBR2
      COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMPO,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)

C      VIBR3=#MATERIALS      (BLOCKS)
C      VIBR4=5*VIBR3
C      VIBR5=VIBR3
C
      IY=1HY
      IR=1
      IW=1
      CALL INITT(1200)
      CALL TERM(2,1024)
50     CALL ERAS
      CALL HOME
      CALL CHSIZ(3)
      CALL ANMOD
      CALL REDIN
      WRITE(IW,1000)
      READ(IR,1005) IYN
      IF(IYN.EQ.IY) CALL EDIT0
100    CALL FITT0
      CALL PLTT0
      WRITE(IW,1000)
      READ(IR,1005) IYN
      IF(IYN.NE.IY) GO TO 300
      CALL EDIT0
      GO TO 100
300    PAUSE
      WRITE(IW,1500)
      READ(IR,1005) IYN
      IF(IYN.EQ.IY) GO TO 450
400    CALL MODUS
      CALL ETA
      WRITE(IW,4000)
      READ(IR,*) DELT0
      CALL GAXES(DELT0)
      CALL PLTPK(3)
      WRITE(IW,1500)
      READ(IR,1005) IYN
      IF(IYN.NE.IY) GO TO 500
450    WRITE(IW,1600)
      READ(IR,*) TEMPO
      GO TO 400
500    WRITE(IW,3000)
      READ(IR,1005) IYN
      IF(IYN.EQ.IY) CALL STORE
      WRITE(IW,2000)
      READ(IR,1005) IYN
      IF(IYN.EQ.IY) GO TO 50
      CALL FINIT(0,100)
1000   FORMAT(2X,19HEDIT THE DATA?(Y,N))
1005   FORMAT(1A1)
```

```

1500  FORMAT(2X,41HCHANGE TO VALUE OR PARAMETER VALUES?(Y,N))
1600  FORMAT(2X,14HENTER TO VALUE)
2000  FORMAT(2X,27HBEGIN ANOTHER PROBLEM?(Y,N))
3000  FORMAT(2X,20HSTORE THE DATA?(Y,N))
4000  FORMAT(2X,21HENTER THE INTERVAL OF
1      /2X,17HTEMPERATURE LINES)
      END

```

```

SUBROUTINE REDIN
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMPO,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION BUF(320),NAME(3)
IY=1HY
WRITE(IW,5000)
READ(IR,1005) IYN
IF(IYN.NE.IY) GO TO 90
WRITE(IW,6000)
READ(IR,*) NMAX
ICR=14
LIBUF=128
ICON=3B
NAME(1)=2HVI
NAME(2)=2HBR
NAME(3)=2H3
DO 80 I=1,NMAX
INS=I-1
CALL EXEC(ICR,ICON,IBUF,LIBUF,NAME,INS)
IF(IBUF(121).NE.I) GO TO 80
WRITE(IW,3200) IBUF(121),(IBUF(J),J=1,120)
80  CONTINUE
90  WRITE(IW,1000)
    READ(IR,1005) IYN
    IF(IYN.NE.IY) GO TO 200
    WRITE(IW,2000)
    WRITE(IW,2100)
    READ(IR,*) MATL
    IBUF(121)=MATL
    WRITE(IW,2200)
    READ(IR,1010) (IBUF(I),I=1,30)
    WRITE(IW,2300)
    READ(IR,1010) (IBUF(I),I=31,60)
    WRITE(IW,2400)
    READ(IR,1010) (IBUF(I),I=61,90)
    WRITE(IW,2500)
    READ(IR,1010) (IBUF(I),I=91,120)
    WRITE(IW,2600)
    READ(IR,*) NPTS
    IBUF(122)=NPTS
    METRC=0
    WRITE(IW,2650)

```

```

READ(IR,1005) IYN
IF(IYN.EQ.IY) METRC=1
IBUF(124)=1
WRITE(IW,2675)
READ(IR,1005) IYN
IF (IYN.EQ.IY) IBUF(124)=0
WRITE(IW,2700)
WRITE(IW,2800)
IF(METRC.EQ.1) WRITE(IW,2850)
IF(METRC.EQ.0) WRITE(IW,2900)
DO 100 I=1,NPTS
100 READ(IR,*) N,(XIN(I,J),J=1,5)
IF(METRC.EQ.1) GO TO 500
DO 150 I=1,NPTS
150 XIN(I,1)=XIN(I,1)*6894.757
XIN(I,3)=5.*(XIN(I,3)-32.)/9.
GO TO 500
200 WRITE(IW,3000)
READ(IR,*) MATL
ICR=14
LBUF=640
LIBUF=128
ICON=3B
INS=MATL-1
IRS=5*INS
NAME(1)=2HVI
NAME(2)=2HBR
NAME(3)=2H3
CALL EXEC(ICR,ICON,IBUF,LBUF,NAME,INS)
IF(IBUF(121).EQ.MATL) GO TO 250
WRITE(IW,3100)
GO TO 200
250 NAME(3)=2H4
NPTS=IBUF(122)
CALL EXEC(ICR,ICON,BUF,LBUF,NAME,IRS)
DO 300 I=1,NPTS
DO 300 J=1,5
300 XIN(I,J)=BUF(5*(I-1)+J)
500 CALL ERAS
CALL HOME
CALL ANMOD
WRITE(IW,3200) IBUF(121),(IBUF(J),J=1,120)
WRITE(IW,2800)
WRITE(IW,2850)
DO 600 I=1,NPTS
600 WRITE(IW,3400) I,(XIN(I,J),J=1,5)
PAUSE
RETURN
1000 FORMAT(2X,22HREAD IN NEW DATA?(Y,N))
1005 FORMAT(1A1)
1010 FORMAT(30A2)
2000 FORMAT(2X,17HENTER HEADER DATA)
2100 FORMAT(2X,30HENTER THE CODE OF THE MATERIAL)
2200 FORMAT(2X,37HENTER THE MATERIAL NAME(MAX 60 CHAR.))
2300 FORMAT(2X,48HENTER THE MANUFACTURER DATA SOURCE(MAX 60 CHAR.))

```

```

2400  FORMAT(2X,40HENTER THE AFML DATA SOURCE(MAX 60 CHAR.))
2500  FORMAT(2X,38HENTER OTHER DATA SOURCES(MAX 60 CHAR.))
2600  FORMAT(2X,44HENTER THE NUMBER OF DATA POINTS(MAX 64 PTS.))
2650  FORMAT(2X,38HIS THE DATA TO BE ENTERED METRIC?(Y,N))
2675  FORMAT(2X,31HIS THE MODULUS DATA SHEAR?(Y,N))
2700  FORMAT(2X,32HREADY TO ENTER DATA FOR MATERIAL)
2800  FORMAT(2X,3HNO.,3X,7HMODULUS,5X,4HLOSS,5X,5HTEMP.,4X,5HFREQ.,
1      3X,4HMODE)
2850  FORMAT(9X,6HN/M**2,4X,6HFACTOR,3X,6HDEG. C,5X,
1      2HHZ,5X,3HNO.)
2900  FORMAT(7X,8HLB/IN**2,4X,6HFACTOR,3X,6HDEG. F,5X,
1      2HHZ,5X,3HNO.)
3000  FORMAT(2X,36HENTER THE CODE OF RETRIEVAL MATERIAL)
3100  FORMAT(2X,36HMISMATCH IN RETRIEVED MATERIAL CODES)
3200  FORMAT(2X,15HMATERIAL CODE :,I3,
1      /2X,10HMATERIAL .,30A2,/12X,12HDATA SOURCES,
2      /2X,14HMANUFACTURER :,30A2,/2X,6HAFML :,30A2,
3      /2X,7HOTHER :,30A2/)
3400  FORMAT(2X,I3,1PE13.5,0PF9.4,F9.1,F9.1,F5.0)
5000  FORMAT(2X,26HLIST THE STORED DATA?(Y,N))
6000  FORMAT(2X,42HENTER THE MAXIMUM NUMBER OF MATERIAL CODES)
      END

```

```

      SUBROUTINE FITT0
      COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
      DIMENSION X(64),COF(3)
      DO 100 I=1,NPTS
      X(I)=XIN(I,3)
      ETA=XIN(I,2)
100    YD(I)=ALOG1(ETA)
      CALL POLYF(X,COF)
      DO 200 I=1,3
200    COFT0(I)=COF(I)
      T0=-COF(2)/(2.*COF(3))
      IT0=T0/10.+5*ABS(T0)/T0
      TEMP0=IT0*10.
      RETURN
      END

```

```

      SUBROUTINE POLYF(X,COF)
      COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
      DIMENSION X(64),COF(3),A(3,3),B(3,3),C(3)
      DO 100 I=1,3

```

```

      C(I)=0.
      DO 100 J=1,3
100    A(I,J)=0.
      A(1,1)=NPTS
      DO 200 I=1,NPTS
      A(1,2)=A(1,2)+X(I)
      A(1,3)=A(1,3)+X(I)**2
      A(2,3)=A(2,3)+X(I)**3
      A(3,3)=A(3,3)+X(I)**4
      C(1)=C(1)+YD(I)
      C(2)=C(2)+X(I)*YD(I)
200    C(3)=C(3)+YD(I)*X(I)**2
      A(2,1)=A(1,2)
      A(2,2)=A(1,3)
      A(3,1)=A(1,3)
      A(3,2)=A(2,3)
      B(1,1)=A(2,2)*A(3,3)-A(3,2)*A(2,3)
      B(2,1)=A(3,1)*A(2,3)-A(2,1)*A(3,3)
      B(3,1)=A(2,1)*A(3,2)-A(3,1)*A(2,2)
      B(1,2)=A(3,2)*A(1,3)-A(1,2)*A(3,3)
      B(2,2)=A(1,1)*A(3,3)-A(3,1)*A(1,3)
      B(3,2)=A(1,2)*A(3,1)-A(1,1)*A(3,2)
      B(1,3)=A(1,2)*A(2,3)-A(2,2)*A(1,3)
      B(2,3)=A(2,1)*A(1,3)-A(1,1)*A(2,3)
      B(3,3)=A(1,1)*A(2,2)-A(1,2)*A(2,1)
      DET=A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)
      DO 300 I=1,3
      DO 300 J=1,3
300    A(I,J)=B(I,J)/DET
      DO 400 I=1,3
      COF(I)=0.
      DO 400 J=1,3
400    COF(I)=A(I,J)*C(J)+COF(I)
      RETURN
      END

```

```

SUBROUTINE PLTTO
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMPO,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
CALL ERAS
ETMAX=XIN(1,2)
ETMIN=XIN(1,2)
TMAX=XIN(1,3)
TMIN=XIN(1,3)
FMAX=XIN(1,4)
FMIN=XIN(1,4)
DO 10 I=1,NPTS
IF(XIN(I,2).GT.ETMAX) ETMAX=XIN(I,2)
IF(XIN(I,2).LT.ETMIN) ETMIN=XIN(I,2)
IF(XIN(I,3).GT.TMAX) TMAX=XIN(I,3)

```

```

      IF(XIN(I,3).LT.TMIN) TMIN=XIN(I,3)
      IF(XIN(I,4).GT.FMAX) FMAX=XIN(I,4)
      IF(XIN(I,4).LT.FMIN) FMIN=XIN(I,4)
10    CONTINUE
      IEXPN=ALOG1(ETMIN)
      ETMIN=IEXPN-1
      IEXPN=ALOG1(ETMAX)
      ETMAX=IEXPN+1
      IMULT=TMIN/10.
      TMIN=10.*(IMULT-1)
      IMULT=TMAX/10.
      TMAX=10.*(IMULT+1)
      IEXPN=ALOG1(FMIN)
      FMIN=IEXPN-1
      IEXPN=ALOG1(FMAX)
      FMAX=IEXPN+1
      CALL PLTT1(4,TMIN,TMAX,FMIN,FMAX)
      CALL PLTT1(2,TMIN,TMAX,ETMIN,ETMAX)
      CALL RESET
      CALL HOME
      CALL CHSIZ(3)
      CALL ANMOD
      WRITE(IW,3000) I BUF(121), (I BUF(J),J=1,30)
      WRITE(IW,2800)
      WRITE(IW,2850)
      DO 300 I=1,NPTS
300  WRITE(IW,4000) I, (XIN(I,J),J=1,5)
      RETURN
3000  FORMAT(2X,15HMATERIAL CODE :,I3,/2X,10HMATERIAL :,30A2)
2800  FORMAT(2X,3HNO.,3X,7HMODULUS,5X,4HLOSS,5X,5HTEMP.,4X,5HFREQ.,
1      3X,4HMODE)
2850  FORMAT(9X,6HN/M**2,4X,6HFACTOR,3X,6HDEG. C,5X,
1      2HHZ,5X,3HNO.)
4000  FORMAT(2X,I3,1PE13.5,0PF9.4,F9.1,F9.1,F5.0)
      END

```

```

      SUBROUTINE PLTT1(NUM,TMIN,TMAX,ETMIN,ETMAX)
      COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMPO,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
      DIMENSION II(2),LABEL(9)
      ITEN=2H10
      CALL DWIND(TMIN,TMAX,ETMIN,ETMAX)
      IF(NUM.EQ.4) CALL TWIND(560,1010,420,770)
      IF(NUM.EQ.2) CALL TWIND(560,1010,30,380)
      CALL MOVEA(TMIN,ETMIN)
      CALL DRAWA(TMAX,ETMIN)
      CALL MOVEA(TMIN,ETMIN)
      CALL DRAWA(TMIN,ETMAX)
      TM=TMIN
80    CALL MOVEA(TM,ETMIN)

```

```

CALL DREL(0,5)
TM=TM+10.
IF(TM.LE.TMAX) GO TO 80
ETM=ETMIN
90 CALL MOVEA(TMIN,ETM)
CALL DREL(5,0)
ETM=ETM+1.
IF(ETM.LE.ETMAX) GO TO 90
CALL CHSIZ(4)
DO 100 I=1,NPTS
TM=XIN(I,3)
ETM=XIN(I,NUM)
ETE=ALOG1(ETM)
CALL PNTA(TM,ETE)
CALL CODE
WRITE(J,1000) I
CALL MREL(2,-5)
CALL AOPST(2,J)
100 CONTINUE
IF(NUM.EQ.4) GO TO 115
TM=TMIN
ETM=(COFT0(1)+TM*(COFT0(2)+TM*COFT0(3)))
CALL MOVEA(TM,ETM)
110 TM=TM+.5
ETM=(COFT0(1)+TM*(COFT0(2)+TM*COFT0(3)))
CALL DRAWA(TM,ETM)
IF(TM.LT.TMAX) GO TO 110
115 CALL CHSIZ(3)
TM=TMIN
MULT=(TMAX-TMIN)/100.+1.
120 IT=TM
CALL CODE
WRITE(I I,2000) I T
CALL MOVEA(TM,ETMIN)
CALL MREL(-24,-14)
CALL AOPST(4,II)
TM=TM+MULT*10.
IF(TM.LE.TMAX) GO TO 120
ETM=ETMIN
130 IETME=ETM
CALL CODE
WRITE(J,1000) I ETME
CALL MOVEA(TMIN,ETM)
CALL MREL(-36,-7)
CALL CHSIZ(3)
CALL AOPST(2,ITEN)
CALL MREL(1,7)
CALL CHSIZ(4)
CALL AOPST(2,J)
ETM=ETM+1.
IF(ETM.LE.ETMAX) GO TO 130
CALL CHSIZ(3)
IF(NUM.EQ.4) CALL MABS(754,390)
IF(NUM.EQ.2) CALL MABS(754,0)
LABEL(1)=2HTE

```

```

    LABEL(2)=2HMP
    LABEL(3)=2HER
    LABEL(4)=2HAT
    LABEL(5)=2HUR
    LABEL(6)=2HE
    LABEL(7)=2HDE
    LABEL(8)=2HG.
    LABEL(9)=2H C
    CALL AOPST(18,LABEL)
    IF(NUM.EQ.4) GO TO 140
    CALL MABS(570,360)
    LABEL(1)=2HET
    LABEL(2)=1HA
    CALL AOPST(3,LABEL)
    CALL MABS(900,360)
    LABEL(1)=2HT0
    LABEL(2)=2H =
    CALL AOPST(4,LABEL)
    IT=TEMPO
    CALL CODE
    WRITE(11,2000) I T
    CALL AOPST(4,11)
    RETURN
140  CALL MABS(570,750)
    LABEL(1)=2HFR
    LABEL(2)=2HEQ
    LABEL(3)=1H.
    CALL AOPST(5,LABEL)
    RETURN
1000 FORMAT(I 2)
2000 FORMAT(I 4)
    END

```

```

SUBROUTINE EDIT0
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMPO,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION IDE(64)
ID=1HD
IE=1HE
IY=1HY
10  WRITE(IW,1000)
    READ(IR,1005) IAD
    IF(IAD.EQ.ID) GO TO 100
    IF(IAD.EQ.IE) RETURN
    WRITE(IW,1100)
    READ(IR,*) NUM
    WRITE(IW,1150)
    READ(IR,1005) IYN
    METRC=0
    IF(IYN.EQ.IY) METRC=1

```

```

NL=NPTS+1
NH=NPTS+NUM
WRITE(IW,1200)
DO 50 I=NL,NH
50  READ(IR,*) N,(XIN(I,J),J=1,5)
    NPTS=NPTS+NUM
    IF(METRC.EQ.1) GO TO 10
    DO 60 I=NL,NH
    XIN(I,1)=XIN(I,1)*6894.757
60  XIN(I,3)=5.*(XIN(I,3)-32.)/9.
    GO TO 10
100  WRITE(IW,2000)
    READ(IR,*) NUM,(IDE(I),I=1,NUM)
    DO 200 I=1,NUM
    II=IDE(I)
    DO 200 J=1,5
200  XIN(II,J)=0.
    J=0
    DO 300 I=1,NPTS
    IF(XIN(I,1).EQ.0.) GO TO 300
    J=J+1
    DO 250 K=1,5
    XIN(J,K)=XIN(I,K)
250  CONTINUE
300  CONTINUE
    NPTS=NPTS-NUM
    GO TO 10
1000 FORMAT(2X,48HEDIT THE ABOVE DATA: ADD, DELETE, OR END?(A,D,E))
1005 FORMAT(1A1)
1100 FORMAT(2X,38HENTER THE NUMBER OF DATA POINTS TO ADD)
1200 FORMAT(2X,30HREADY TO ADD DATA FOR MATERIAL
1      / 7X,21HIN FORMAT SHOWN ABOVE)
2000 FORMAT(2X,31HENTER THE NUMBER OF DATA POINTS
1      / 2X,32HTO DELETE AND THE INDICES OF THE
2      / 11X,14HROWS TO DELETE)
1150 FORMAT(2X,36HIS THE DATA TO BE ADDED METRIC?(Y,N))
    END

```

```

SUBROUTINE STORE
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION BUF(320),NAME(3),BUFA(64)
DO 100 I=1,NPTS
DO 100 J=1,5
100  BUF(5*(I-1)+J)=XIN(I,J)
    IBUF(122)=NPTS
    IBUF(125)=1
    BUFA(1)=TEMP0
    BUFA(2)=COFT0(1)
    BUFA(3)=COFT0(2)

```

```

BUFA(4)=COFT0(3)
BUFA(5)=GPAR(1)
BUFA(6)=GPAR(2)
BUFA(7)=GPAR(3)
BUFA(8)=GPAR(4)
BUFA(9)=ETPAR(1)
BUFA(10)=ETPAR(2)
BUFA(11)=ETPAR(3)
BUFA(12)=ETPAR(4)
BUFA(13)=ETPAR(5)
BUFA(14)=DELTA(1)
BUFA(15)=DELTA(2)
BUFA(16)=DELTA(3)
BUFA(17)=DELTA(4)
BUFA(18)=DELTA(5)
ICR=15
LBUF=640
LIBUF=128
ICON=3B
INS=IBUF(121)-1
IRS=5*INS
NAME(1)=2HVI
NAME(2)=2HBR
NAME(3)=2H3
CALL EXEC(ICR,ICON,IBUF,LIBUF,NAME,INS)
NAME(3)=2H4
CALL EXEC(ICR,ICON,BUF,LBUF,NAME,IRS)
NAME(3)=2H5
CALL EXEC(ICR,ICON,BUFA,LIBUF,NAME,INS)
WRITE(IW,1000)
PAUSE
RETURN
1000 FORMAT(2X,22HABOVE PLOTTED DATA AND
1/2X,24HPARAMETERS ARE STORED ON
2/2X,30HFILES: VIBR3, VIBR4, AND VIBR5)
END

```

```

SUBROUTINE MODUS
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1 COFT0(3),TEMPO,CHISQ,NPAR,DELTA(5),DELM(5),
2 MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION A(5),ICON(5),FR(64),COF(3)
IY=1HY
CALL ERAS
CALL HOME
CALL ANMOD
WRITE(IW,1900)
NFUN=1
NPAR=4
TEMP=XIN(1,3)
FRMIN=ALPHT(TEMP)*XIN(1,4)

```

```

FRMAX=FRMIN
GMIN=XIN(1,1)
DO 20 I=1,NPTS
TEMP=XIN(I,3)
FRX=ALPHT(TEMP)*XIN(I,4)
IF(FRX.LT.FRMIN) FRMIN=FRX
IF (FRX.GT.FRMAX) FRMAX=FRX
IF(XIN(I,1).LT.GMIN) GMIN=XIN(I,1)
FR(I)=ALOG1(FRX)
G=XIN(I,1)
YD(I)=ALOG1(G)
20 CONTINUE
CALL POLYF(FR,COF)
A(1)=SQRT(FRMIN*FRMAX)
A1=ALOG1(A(1))
A(2)=10.**((COF(1)+A1*(COF(2)+A1*COF(3)))
A(3)=.4
A(4)=.5*GMIN
DO 30 I=1,NPAR
DELTA(I)=0.1*A(I)
IF(DELTA(I).EQ.0.) DELTA(I)=0.1
30 DELMI(I)=0.01*DELTA(I)
MASK(1)=1
MASK(2)=1
MASK(3)=0
MASK(4)=1
WRITE(IW,2000)
READ(IR,*) CNSTR(1)
WRITE(IW,2100)
READ(IR,*) CNSTR(2)
WRITE(IW,2200)
READ(IR,*) CNSTR(3)
WRITE(IW,2300)
READ(IR,*) CNSTR(4)
CALL SEARC(A,NFUN)
DO 40 I=1,NPAR
40 MASK(I)=0
50 CALL SEARC(A,NFUN)
DO 60 I=1,NPAR
60 GPAR(I)=A(I)
CALL PLTPK(NFUN)
WRITE(IW,1000)
READ(IR,1005) IYN
IF(IYN.EQ.IY) RETURN
DO 70 I=1,NPAR
70 MASK(I)=0
WRITE(IW,1100)
READ(IR,1005) IYN
IF(IYN.NE.IY) GO TO 300
WRITE(IW,1200)
READ(IR,*) NCON,(ICON(I),I=1,NCON)
DO 100 I=1,NCON
100 MASK(ICON(I))=1
300 WRITE(IW,1300)
READ(IR,*) A(1)

```

```

WRITE(IW,1400)
READ(IR,*) A(2)
WRITE(IW,1500)
READ(IR,*) A(3)
WRITE(IR,1600)
READ(IR,*) A(4)
GO TO 50
1900 FORMAT(2X,16HFIT MODULUS DATA)
2000 FORMAT(2X,41HENTER THE MINIMUM VALUE OF FROM PARAMETER)
2100 FORMAT(2X,41HENTER THE MINIMUM VALUE OF MROM PARAMETER)
2200 FORMAT(2X,41HENTER THE MAXIMUM VALUE OF MROM PARAMETER)
2300 FORMAT(2X,39HENTER THE MINIMUM VALUE OF ML PARAMETER)
1000 FORMAT(2X,25HACCEPT THESE VALUES?(Y,N))
1005 FORMAT(1A1)
1100 FORMAT(2X,34HHOLD ANY PARAMETERS CONSTANT?(Y,N))
1200 FORMAT(2X,31HENTER THE NUMBER AND INDICES OF
1 / 2X,23HTHE CONSTANT PARAMETERS)
1300 FORMAT(2X,30HENTER FROM(A1) PARAMETER VALUE)
1400 FORMAT(2X,30HENTER MROM(A2) PARAMETER VALUE)
1500 FORMAT(2X,27HENTER N(A3) PARAMETER VALUE)
1600 FORMAT(2X,28HENTER ML(A4) PARAMETER VALUE)
END

```

```

SUBROUTINE ETA
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1 COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2 MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION B(5),ICON(5),FR(64),COF(3)
IY=1HY
CALL ERAS
CALL HOME
CALL ANMOD
WRITE(IW,1900)
NFUN=2
NPAR=5
DO 20 I=1,NPTS
ETAX=XIN(I,2)
YD(I)=ALOG1(ETAX)
TEMP=XIN(I,3)
FRX=ALPHT(TEMP)*XIN(I,4)
20 FR(I)=ALOG1(FRX)
CALL POLYF(FR,COF)
B4=-COF(2)/(2.*COF(3))
B(1)=10.** (COF(1)+B4*(COF(2)+B4*COF(3)))
B(2)=.35
B(3)=-.45
B(4)=10.** B4
B(5)=4.
DO 30 I=1,NPAR
DELTA(I)=0.1*B(I)
IF(DELTA(I).EQ.0.) DELTA(I)=0.1

```

AJ-A079 914

DAYTON UNIV OH RESEARCH INST

F/G 9/2

COMPUTERIZED PROCESSING AND GRAPHIC REPRESENTATION OF VISCOELAS--ETC(U)

AUG 79 C S KING

F33615-76-C-5137

UNCLASSIFIED

UDR-TR-78-49

AFML-TR-79-4099

NL

ENC
AJ 90-4



END
DATE
FILMED

2-80

DDC

```

30   DELMI(I)=0.01*DELTA(I)
      MASK(1)=1
      MASK(2)=1
      MASK(3)=1
      MASK(4)=1
      MASK(5)=0
      WRITE(IW,2000)
      READ(IR,*) CNSTR(1)
      WRITE(IW,2100)
      READ(IR,*) CNSTR(2)
      WRITE(IW,2200)
      READ(IR,*) CNSTR(3)
      WRITE(IW,2300)
      READ(IR,*) CNSTR(4)
      CALL SEARC(B,NFUN)
      DO 40 I=1,NPAR
40   MASK(I)=0
50   CALL SEARC(B,NFUN)
      DO 60 I=1,NPAR
60   ETPAR(I)=B(I)
      CALL PLTPK(NFUN)
      WRITE(IW,1000)
      READ(IR,1005) IYN
      IF(IYN.EQ.IY) RETURN
      DO 70 I=1,NPAR
70   MASK(I)=0
      WRITE(IW,1100)
      READ(IR,1005) IYN
      IF(IYN.NE.IY) GO TO 300
      WRITE(IW,1200)
      READ(IR,*) NCON,(ICON(I),I=1,NCON)
      DO 100 I=1,NCON
100  MASK(ICON(I))=1
300  WRITE(IW,1300)
      READ(IR,*) B(1)
      WRITE(IW,1400)
      READ(IR,*) B(2)
      WRITE(IW,1500)
      READ(IR,*) B(3)
      WRITE(IW,1600)
      READ(IR,*) B(4)
      WRITE(IW,1700)
      READ(IR,*) B(5)
      GO TO 50
1900 FORMAT(2X,20HFIT LOSS FACTOR DATA)
2000 FORMAT(2X,44HENTER THE MINIMUM VALUE OF ETA FROL PARAMETER)
2100 FORMAT(2X,39HENTER THE MINIMUM VALUE OF SL PARAMETER)
2200 FORMAT(2X,39HENTER THE MAXIMUM VALUE OF SH PARAMETER)
2300 FORMAT(2X,41HENTER THE MINIMUM VALUE OF FROL PARAMETER)
1000 FORMAT(2X,25HACCEPT THESE VALUES?(Y,N))
1005 FORMAT(1A1)
1100 FORMAT(2X,34HHOLD ANY PARAMETERS CONSTANT?(Y,N))
1200 FORMAT(2X,31HENTER THE NUMBER AND INDICES OF
1    / 2X,23HTHE CONSTANT PARAMETERS)
1300 FORMAT(2X,33HENTER ETA FROL('E1) PARAMETER VALUE)

```

```

1400 FORMAT(2X,28HENTER SL(B2) PARAMETER VALUE)
1500 FORMAT(2X,28HENTER SH(B3) PARAMETER VALUE)
1600 FORMAT(2X,30HENTER FROL(B4) PARAMETER VALUE)
1700 FORMAT(2X,27HENTER C(B5) PARAMETER VALUE)
END

```

```

SUBROUTINE FCN(X,NFUN)
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION X(5)
F=0.
DO 30 I=1,NPTS
TEMP=XIN(I,3)
FR=ALPHT(TEMP)*XIN(I,4)
DF=YD(I)-FCT(FR,X,NFUN)
30  F=F+DF*DF
CHISQ=F
RETURN
END

```

```

FUNCTION FCT(FR,X,NFUN)
DIMENSION X(5)
GO TO (10,20),NFUN
10  FCT=F1(FR,X)
RETURN
20  FCT=F2(FR,X)
RETURN
END

```

```

FUNCTION ALPHT(TEMP)
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
ALPHT=10.**(-12.*(TEMP-TEMP0)/(525./1.8+TEMP-TEMP0))
RETURN
END

```

```

FUNCTION F1(FR,A)
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION A(5)
F1=0.
IF(NPAR.EQ.0) GO TO 100
IF(A(1).LT.CNSTR(1)) F1=1.E10
IF(A(2).LT.CNSTR(2)) F1=1.E10
IF(A(2).GT.CNSTR(3)) F1=1.E10
IF(A(4).LT.CNSTR(4)) F1=1.E10
IF(F1.NE.0.) RETURN
100 D=1.+(A(1)/FR)**A(3)
G=ALOG1(A(4))
F1=G+(2.*ALOG1(A(2)/A(4)))/D
RETURN
END

```

```

FUNCTION F2(FR,B)
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION B(5)
F2=0.
IF(NPAR.EQ.0) GO TO 100
IF(B(1).LT.CNSTR(1)) F2=1.E10
IF(B(2).LT.CNSTR(2)) F2=1.E10
IF(B(3).GT.CNSTR(3)) F2=1.E10
IF(B(4).LT.CNSTR(4)) F2=1.E10
IF(F2.NE.0.) RETURN
100 A=ALOG1(FR/B(4))/B(5)
C=ALOG1(B(1))
D=1.-SQRT(1.+A*A)
F2=C+((B(2)+B(3))*A+(B(2)-B(3))*D)*B(5)/2.
RETURN
END

```

```

FUNCTION ALOG1(X)
CONST=ALOG(10.)
ALOG1=ALOG(X)/CONST
RETURN
END

```

```

SUBROUTINE VERTL(NUM,LABEL)
DIMENSION LABEL(1)
DO 100 I=1,NUM
CALL AOPST(1,LABEL(I))
CALL MREL(-13,-21)
100 CONTINUE
RETURN
END

```

```

SUBROUTINE SEARC(X,NFUN)
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1 COFT0(3),TEMPO,CHISQ,NV,DELTA(5),DELM(5),
2 MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
DIMENSION CHI(6),Z(6,5),ZBAR(5),ZSTAR(5),X(5)
ALPHA=1.0
BETA=0.5
GAMMA=2.0
HUGE=1.0E37
SIGNI=1.E5
NVP=NV+1
NF=0
1 DO 10 J=1,NV
10 Z(NV+1,J)=X(J)
CALL FCN(X,NFUN)
NF=NF+1
CHI(NV+1)=CHISQ
DO 30 J=1,NV
DO 20 K=1,NV
20 Z(J,K)=X(K)
SIGN=1.0
23 IF(MASK(J).EQ.1) GO TO 21
Z(J,J)=Z(J,J)+SIGN*DELTA(J)
21 XS=X(J)
X(J)=Z(J,J)
CALL FCN(X,NFUN)
NF=NF+1
X(J)=XS
IF(CHI(NV+1)-CHISQ) 22,30,30
22 IF(SIGN.EQ.-2.0) GO TO 30
SIGN=-2.0
GO TO 23
30 CHI(J)=CHISQ
40 JH=1
JL=1
DO 80 J=2,NVP
IF(CHI(J)-CHI(JH))60,60,50
50 JH=J
60 IF(CHI(J)-CHI(JL))70,80,80
70 JL=J
80 CONTINUE
90 DO 120 J=1,NV
ZBAR(J)=0.0

```

```

DO 110 K=1,NVP
IF(K-JH)100,110,100
100 ZBAR(J)=ZBAR(J)+Z(K,J)
110 CONTINUE
120 ZBAR(J)=ZBAR(J)/FLOAT(NV)
DO 130 J=1,NV
IF(MASK(J).EQ.1) GO TO 130
X(J)=(1.0+ALPHA)*ZBAR(J)-ALPHA*Z(JH,J)
130 ZSTAR(J)=X(J)
CALL FCN(X,NFUN)
NF=NF+1
CHIST=CHISQ
IF(CHISQ-CHI(JL))160,180,180
160 DO 170 J=1,NV
IF(MASK(J).EQ.1) GO TO 170
X(J)=GAMMA*X(J)+(1.0-GAMMA)*ZBAR(J)
170 CONTINUE
CALL FCN(X,NFUN)
NF=NF+1
IF(CHISQ-CHIST) 420,230,230
180 DO 200 J=1,NVP
IF(J-JH)190,200,190
190 IF(CHISQ-CHI(J))230,200,200
200 CONTINUE
IF(CHISQ-CHI(JH))250,270,270
230 DO 240 J=1,NV
240 Z(JH,J)=ZSTAR(J)
CHI(JH)=CHIST
GO TO 440
250 DO 260 J=1,NV
260 Z(JH,J)=X(J)
CHI(JH)=CHISQ
270 DO 280 J=1,NV
IF(MASK(J).EQ.1) GO TO 280
X(J)=BETA*Z(JH,J)+(1.0-BETA)*ZBAR(J)
280 CONTINUE
CALL FCN(X,NFUN)
NF=NF+1
IF(CHISQ-CHI(JH))420,350,350
350 JS=JL
DO 390 J=1,NVP
IF(J-JL)360,390,360
360 DO 370 K=1,NV
Z(J,K)=0.5*(Z(J,K)+Z(JL,K))
370 X(K)=Z(J,K)
CALL FCN(X,NFUN)
NF=NF+1
IF(CHISQ-CHI(JL))380,391,391
380 JS=J
391 CHI(J)=CHISQ
390 CONTINUE
JL=JS
GO TO 440
420 DO 430 J=1,NV
430 Z(JH,J)=X(J)

```

```

440 CHI(JH)=CHISQ
DO 470 J=1,NV
  IF(MASK(J) .EQ. 1) GO TO 470
  ZMAX=-HUGE
  ZMIN=HUGE
  DO 450 K=1,NVP
    ZMAX=AMAX1(ZMAX,Z(K,J))
450   ZMIN=AMIN1(ZMIN,Z(K,J))
    IF(SIGNI*(ZMAX-ZMIN)-AMIN1(ABS(ZMAX),ABS(ZMIN))) 470,470,460
460   IF(ZMAX-ZMIN-ABS(DELM1(J))) 470,470,475
470   CONTINUE
    GO TO 480
475   JH=1
    JL=1
    DO 479 J=2,NVP
      IF(CHI(J)-CHI(JH)) 477,477,476
476   JH=J
477   IF(CHI(J)-CHI(JL)) 478,479,479
478   JL=J
479   CONTINUE
    IF(SIGNI*(CHI(JH)-CHI(JL))-CHI(JL)) 480,480,90
480   IF(CHI(JH)-CHI(JL)) 490,500,500
490   JL=JH
500   CHISQ=CHI(JL)
DO 510 J=1,NV
510   X(J)=Z(JL,J)
999  RETURN
END

```

SUBROUTINE GAXES(DELTO)

COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),

1 COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM1(5),

2 MASK(5),GPAR(4),ETPAR(5),CNSTR(4)

DIMENSION II(2),A(5),LABEL(15)

CALL ERAS

ITEN=2H10

IP=1H+

IX=1HX

NPAR=0

TEMP=XIN(1,3)

FRMIN=ALPHT(TEMP)*XIN(1,4)

FRMAX=FRMIN

GMIN=XIN(1,1)

GMAX=GMIN

ETMIN=XIN(1,2)

DO 50 I=1,NPTS

IF(XIN(I,1).LT.GMIN) GMIN=XIN(I,1)

IF(XIN(I,1).GT.GMAX) GMAX=XIN(I,1)

IF(XIN(I,2).LT.ETMIN) ETMIN=XIN(I,2)

TEMP=XIN(I,3)

FR=ALPHT(TEMP)*XIN(I,4)

IF(FR.LT.FRMIN) FRMIN=FR

IF(FR.GT.FRMAX) FRMAX=FR

CONTINUE

IEFR=ALOG1(FRMIN)

FRMIN=IEFR-2

50

```

IEFR=ALOG1(FRMAX)
FRMAX=IEFR+2
IGMIN=ALOG1(GMIN)
IGMIN=IGMIN-1
IGMAX=ALOG1(GMAX)
60  IGMAX=IGMAX+1
    IDIFG=IGMAX-IGMIN
    IF(IDIFG.LT.4) GO TO 60
    GMIN=IGMIN
    GMAX=IGMAX
    IETMI=ALOG1(ETMIN)
    ETMIN=IETMI-1
    ETMAX=IETMI-1+IDIFG
    FMIN=0.
    FMAX=IDIFG
    DELTA(1)=FRMIN
    DELTA(2)=FRMAX
    DELTA(3)=GMIN
    DELTA(4)=GMAX
    DELTA(5)=ETMIN
    INDEX=1
70  CONTINUE
    IF(INDEX.EQ.1) CALL DWIND(FRMIN,FRMAX,GMIN,GMAX)
    IF(INDEX.EQ.2) CALL DWIND(FRMIN,FRMAX,ETMIN,ETMAX)
    CALL TWIND(140,960,50,730)
    IF(INDEX.EQ.2) GO TO 117
    FR=FRMIN
100  DO 105 I=1,10
        XI=I
        XFR=FR+ALOG1(XI)
        CALL MOVEA(XFR,GMIN)
        CALL DRAWA(XFR,GMAX)
105  CONTINUE
        FR=FR+1.
        IF(FR.LT.FRMAX) GO TO 100
        G=GMIN
110  DO 115 I=1,10
        XI=I
        XG=G+ALOG1(XI)
        CALL MOVEA(FRMIN,XG)
        CALL DRAWA(FRMAX,XG)
115  CONTINUE
        G=G+1.
        IF(G.LT.GMAX) GO TO 110
117  CALL CHSIZ(3)
        DO 120 I=1,NPTS
        TEMP=XIN(I,3)
        FR=ALPHT(TEMP)*XIN(I,4)
        G=XIN(I,INDEX)
        EFR=ALOG1(FR)
        EG=ALOG1(G)
        CALL MOVEA(EFR,EG)
        CALL MREL(-4,-6)
        IF(INDEX.EQ.1) CALL AOPST(1,IP)
        IF(INDEX.EQ.2) CALL AOPST(1,IX)

```

```

120  CONTINUE
      IF(INDEX.EQ.2) GO TO 135
      DO 130 I=1,4
130  A(I)=GPAR(I)
      GO TO 137
135  DO 180 I=1,5
180  A(I)=ETPAR(I)
137  EFR=FRMIN
140  FR=10.**E FR
      IF(INDEX.EQ.1) EG=F1(FR,A)
      IF(INDEX.EQ.2) EG=F2(FR,A)
      IF(EFR.EQ.FRMIN) CALL MOVEA(E FR,EG)
      CALL DRAWA(E FR,EG)
      EFR=EFR+.05
      IF(E FR.LE.FRMAX) GO TO 140
      IF(INDEX.EQ.1) GO TO 145
      CALL MABS(68,50)
      CALL DRABS(68,730)
145  CONTINUE
      IF(INDEX.EQ.1) G=GMIN
      IF(INDEX.EQ.2) G=ETMIN
150  CALL MOVEA(FRMIN,G)
      IF(INDEX.EQ.1) GO TO 155
      CALL MREL(-68,0)
      CALL DREL(-5,0)
155  CALL MREL(-46,-5)
      CALL CHSIZ(2)
      CALL AOPST(2,ITEN)
      IEG=G
      CALL CODE
      WRITE(J,4000) IEG
      CALL MREL(1,11)
      CALL CHSIZ(3)
      CALL AOPST(2,J)
      G=G+1.
      IF(INDEX.EQ.1.AND.G.LE.GMAX) GO TO 150
      IF(INDEX.EQ.2.AND.G.LE.ETMAX) GO TO 150
      IF(INDEX.EQ.2) GO TO 170
      MULT=(FRMAX-FRMIN)/10.+1.
      FR=FRMIN
160  CALL MOVEA(FR,GMIN)
      CALL MREL(-26,-25)
      CALL CHSIZ(2)
      CALL AOPST(2,ITEN)
      IEFR=FR
      CALL CODE
      WRITE(J,4000) IEFR
      CALL MREL(1,11)
      CALL CHSIZ(3)
      CALL AOPST(2,J)
      FR=FR+MULT
      IF(FR.LE.FRMAX) GO TO 160
170  INDEX=INDEX+1
      IF(INDEX.LE.2) GO TO 70
      CALL DWIND(FRMIN,FRMAX,FMIN,FMAX)

```

```

CALL TWIND(140,960,50,730)
CALL CHSIZ(2)
WIDT0=200.
IF(DELT0.LT.20.) WIDT0=100.
T=TEMP0-WIDT0
TMAX=TEMP0+WIDT0
210 FR=ALPHT(T)
EFR=ALOG1(FR)
CALL MOVEA(EFR,FMIN)
EFR=EFR+FMAX
CALL DRAWA(EFR,FMAX)
IF(EFR.GE.FRMAX) GO TO 220
IT=T
TEM=(T-TEMP0)/50.
ITEM=TEM
IF(ITEM.NE.TEM) GO TO 220
CALL CODE
WRITE(II,5000) IT
CALL MREL(-39,3)
CALL AOPST(4,II)
220 T=T+DELT0
IF(T.LE.TMAX) GO TO 210
F=FMIN
230 CALL MOVEA(FRMAX,F)
CALL MREL(2,-5)
CALL CHSIZ(2)
CALL AOPST(2,ITEN)
IEF=F
CALL CODE
WRITE(J,4000) IEF
CALL MREL(1,11)
CALL CHSIZ(3)
CALL AOPST(2,J)
F=F+1.
IF(F.LE.FMAX) GO TO 230
CALL CHSIZ(2)
CALL MABS(400,0)
LABEL(1)=2HRE
LABEL(2)=2H DU
LABEL(3)=2HCE
LABEL(4)=2HD
LABEL(5)=2HFR
LABEL(6)=2HEQ
LABEL(7)=2HUE
LABEL(8)=2HNC
LABEL(9)=2HY
LABEL(10)=2HFR
LABEL(11)=2H H
LABEL(12)=1HZ
CALL AOPST(23,LABEL)
CALL MABS(430,755)
LABEL(1)=2HTE
LABEL(2)=2HMP
LABEL(3)=2HER
LABEL(4)=2HAT

```

```

LABEL(5)=2HUR
LABEL(6)=2HE
LABEL(7)=2HT
LABEL(8)=2HDE
LABEL(9)=2HG.
LABEL(10)=2H C
CALL AOPST(20,LABEL)
CALL MABS(5,530)
LABEL(1)=1HL
LABEL(2)=1HO
LABEL(3)=1HS
LABEL(4)=1HS
LABEL(5)=1H
LABEL(6)=1HF
LABEL(7)=1HA
LABEL(8)=1HC
LABEL(9)=1HT
LABEL(10)=1HO
LABEL(11)=1HR
LABEL(12)=1H
LABEL(13)=1HE
LABEL(14)=1HT
LABEL(15)=1HA
CALL VERTL(15,LABEL)
CALL MABS(75,635)
LABEL(1)=1HS
LABEL(2)=1HH
LABEL(3)=1HE
LABEL(4)=1HA
LABEL(5)=1HR
LABEL(6)=1H
LABEL(7)=1HG
LABEL(8)=1H
IF(IBUF(124).EQ.0) CALL VERTL(8,LABEL)
LABEL(1)=1HY
LABEL(2)=1HO
LABEL(3)=1HU
LABEL(4)=1HN
LABEL(5)=1HG
LABEL(6)=1HS
LABEL(7)=1H
LABEL(8)=1HE
LABEL(9)=1H
IF(IBUF(124).EQ.1) CALL VERTL(9,LABEL)
LABEL(1)=1HM
LABEL(2)=1HO
LABEL(3)=1HD
LABEL(4)=1HU
LABEL(5)=1HL
LABEL(6)=1HU
LABEL(7)=1HS
LABEL(8)=1H
LABEL(9)=1HN
LABEL(10)=1H/
LABEL(11)=1HM

```

```

LABEL(12)=1H*
LABEL(13)=1H*
LABEL(14)=1H2
CALL VERTL(14,LABEL)
CALL MABS(1008,520)
LABEL(1)=1HF
LABEL(2)=1HR
LABEL(3)=1HE
LABEL(4)=1HQ
LABEL(5)=1HU
LABEL(6)=1HE
LABEL(7)=1HN
LABEL(8)=1HC
LABEL(9)=1HY
LABEL(10)=1H
LABEL(11)=1HF
LABEL(12)=1H
LABEL(13)=1HH
LABEL(14)=1HZ
CALL VERTL(14,LABEL)
CALL MABS(0,0)
CALL CHSIZ(3)
CALL ANMOD
PAUSE
CALL ERAS
CALL RESET
CALL CHSIZ(3)
CALL HOME
CALL ANMOD
RETURN
4000 FORMAT(I 2)
5000 FORMAT(I 4)
END

```

```

SUBROUTINE PLTPK(NUM)
COMMON DUMMY(60),IR,IW,IBUF(128),NPTS,XIN(64,5),YD(64),
1      COFT0(3),TEMP0,CHISQ,NPAR,DELTA(5),DELM(5),
2      MASK(5),GPAR(4),ETPAR(5),CNSTR(4)
IF(NUM-2) 10,20,30
10 CONTINUE
WRITE(IW,1000)
WRITE(IW,1500) TEMP0,(GPAR(I),I=1,4)
RETURN
20 CONTINUE
WRITE(IW,2000)
WRITE(IW,2500) TEMP0,(ETPAR(I),I=1,5)
RETURN
30 CONTINUE
WRITE(IW,3000) IBUF(121),(IBUF(J),J=1,30)
WRITE(IW,1000)
WRITE(IW,1500) TEMP0,(GPAR(I),I=1,4)

```

```

WRITE(I W, 2000)
WRITE(I W, 2500) TEMP0, (ETPAR(I), I=1, 5)
WRITE(I W, 6000)
PAUSE
RETURN
1000 FORMAT(2X, 47HLOG(M)=LOG(ML)+(2LOG(MROM/ML))/(1+(FROM/FR)**N)
1 /4X, 4HT0 C, 6X, 4HFROM, 8X, 4HMROM, 7X, 1HN, 9X, 2HML,
2 /15X, 2HA1, 10X, 2HA2, 8X, 2HA3, 8X, 2HA4)
1500 FORMAT(2X, F8.1, 1PE12.4, 1PE12.4, 0PF8.3, 1PE12.4)
2000 FORMAT(2X, 23HA=(LOG(FR)-LOG(FROL))/C,
1 /2X, 59HLOG(ETA)=LOG(ETAFROL)+((SL+SH)A+(SL-SH)(1-SQRT(1+A**2)))C/
22, /4X, 4HT0 C, 2X, 7HETAFROL, 4X, 2HSL, 6X, 2HSH, 7X, 4HFROL, 7X, 1HC,
3 /13X, 2HB1, 6X, 2HB2, 6X, 2HB3, 8X, 2HB4, 8X, 2HB5)
2500 FORMAT(2X, F8.1, F8.3, F8.3, F8.3, 1PE12.4, 0PF8.3)
3000 FORMAT(/2X, 15HMATERIAL CODE :, I3, /2X, 10HMATERIAL :, 30A2)
6000 FORMAT(2X, 38HLOG(FR)=LOG(F)-12(T-T0)/(525/1.8+T-T0)/)
END

END$

```