AD-A076 354 ALABAMA UNIV IN HUNTSVILLE F/G 13/4

METHODOLOGY FOR CONTAINER CUSHIONING MODEL DEVELOPMENT AND VALI--ETC(U)

AUG 79 R M WYSKIDA , J D JOHANNES DAAK40-78-C-0146

UNCLASSIFIED UAH-221-VOL-2 USAMICOM-TRT-CR-79-24-VOL- NL

1 OF 2

AD
A076354

MICOM Report No. TRT-CR-79-24
UAH Report No. 221

(2)

LEVEL II

A076353
VOL I

AD A076354

METHODOLOGY FOR CONTAINER CUSHIONING MODEL DEVELOPMENT AND VALIDATION

by

Richard M. Wyskida
James D. Johannes

Final Report
For the Period 8 June, 1978 - 30 September, 1979
Vol. II

This research work was supported by the
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809
Under Contract DAAK40-78-C-0146

The University of Alabama in Huntsville
P. O. Box 1247
Huntsville, Alabama 35807

August 1979

DISPOSITION INSTRUCTIONS

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

MICOM Report No. TRT-CR-79-24
UAH Report No. 221

METHODOLOGY FOR CONTAINER CUSHIONING MODEL DEVELOPMENT AND VALIDATION

by

Richard M. Wyskida
James D. Johannes

Final Report
For the Period 8 June, 1978 - 30 September, 1979
Vol. II

This research work was supported by the
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809
Under Contract DAAK40-78-C-0146

The University of Alabama in Huntsville
P. O. Box 1247
Huntsville, Alabama 35807

August 1979

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MICOM Report No. TRT-CR-79-24	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Methodology for Container Cushioning Model Development and Validation		5. TYPE OF REPORT & PERIOD COVERED Final 8 June 1978 - 30 September 79
		6. PERFORMING ORG. REPORT NUMBER 221 Vol. II
7. AUTHOR(s) Richard M. Wyskida James D. Johannes		8. CONTRACT OR GRANT NUMBER(s) DAAK40-78-C-0146
9. PERFORMING ORGANIZATION NAME AND ADDRESS The University of Alabama in Huntsville P. O. Box 1247 Huntsville, AL 35807		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, U. S. Army Missile Command Redstone Arsenal, AL 35809 Attn: DRSMI-RL		12. REPORT DATE August 1979
		13. NUMBER OF PAGES 126
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release - Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

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

Cushioning
Methodology
Models
Validation

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

This research report documents the procedure necessary to develop bulk cushioning mathematical models for use in either the tailored or encapsulated application.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The development of mathematical models for temperature sensitive bulk cushioning materials requires a very precise and exacting procedure. The procedure which has been developed during the MICOM Cushioning Material Research Program consists of the following aspects:

1. experimental design
2. data acquisition forms
   a) single material
   b) composite two material
3. test plan
4. temperature effects (analysis of variance)
5. outlier detection
6. individual dynamic dushioning curves
7. model development
8. model validation
9. model optimization
   a) tailored cushion (CUSHOP)
   b) encapsulation (ENCAP).

Each of these aspects will be discussed in the material which follows. For several aspects, a computer code exists to perform the calculations necessary to utilize that portion of the procedure. These computer codes have been tested extensively, and have been utilized in the development of nine bulk cushioning models.

This report will be confined to a general description of the developed container cushioning methodology. Complete details, together with extensive data analysis, are contained in [1 - 9] for the individual interested in a specific application.

EXPERIMENTAL DESIGN

The procedure begins with the experimental design which, for all MICOM/UAH temperature sensitive bulk cushioning efforts, has been a split-split plot design. The split-split plot design is a specialized form of a nested design in which a subtreatment factor is nested within a main treatment, and a sub-subtreatment is crossed with the subtreatment factor and the main treatment.

Under this design configuration, the mathematical model for each experimental observation may be written as

$$y_{ijkt} = \mu + S_i + R_t + SR_{it} \Big\} \quad \text{WHOLE PLOT}$$

$$+ Th_j + STh_{ij} + ThR_{jt} + SThR_{ijt} \Big\} \quad \text{SPLIT-PLOT}$$

$$\left. \begin{array}{l} + T_k + ST_{ik} + ThT_{jk} + SThT_{ijk} + TR_{kt} \\ + ThTR_{jkt} + STR_{ikt} + SThTR_{ijkt} \end{array} \right\} \quad \text{SPLIT-SPLIT PLOT}$$

where μ = the general mean

S_i = i^{th} static stress level

Th_j = j^{th} material thickness

T_k = k^{th} temperature level

R_t = t^{th} replication

SR_{it} = whole plot error

$ThR_{jt} + SThR_{ijt}$ = Split-plot error

$TR_{kt} + ThTR_{jkt} + STR_{ikt} + SThTR_{ijkt}$ = Split-split plot error.

DATA ACQUISITION FORMS

In an attempt to simplify the data acquisition procedure, while maintaining the randomization element of experimental design, a data collection form generator has been prepared for use on the UNIVAC 1100. This form generator prepares forms of the type shown in Table 1. The randomization scheme provides the order in which the drop height-thickness experimental combinations should be performed. For example, the first experimental combination to be performed for a given stress level, temperature, and replication, according to Table 1, is identified by a 1 at the intersection of a 2 inch thickness and a 30 inch drop height. Thus, the random order of experimentation is completely provided.

The numbers to the right of the matrix gives the experimenter the order of material samples to be loaded into the temperature controlled bins. Consequently, during the actual drop tests, the temperature conditioned samples can be drawn out in the proper sequence, with a minimum amount of time and temperature loss.

A computer code for single material forms generation is provided in Appendix A. The computer code for composite two material forms generation is located in Appendix B. A sample of a composite two material form is given in Table 2.

Table 1. Sample data collection form for single material

SINGLE MATERIAL

ETHAFOAM-2

STRESS LEVEL: REPLICATION:

TEMPERATURE:

DROP HEIGHT

THICKNESS	12''	18''	24''	30''
1''	7	12	4	5
2''	8	2	10	1
3''	3	11	6	9

2''
2''
3''
1''
1''
3''
1''
2''
3''
2''
3''
1''

Table 2. Sample data collection form for two materials

```
                         COMPOSITE  MATERIALS

                  INNER(BOTTOM)        OUTER(TOP)

                    ETHAFOAM2           ETHAFOAM4

      STRESS LEVEL:                   REPLICATION:

      TEMPERATURE:


                             DROP HEIGHT

             I   12''   I   18''   I   24''   I   30''   I
         ----I------I----I------I----I------I----I------I----I
             I      I    I      I    I      I    I      I    I
1''-ETHAFOAM4I      I    I      I    I      I    I      I    I
             I      I  2 I      I 10 I      I  7 I      I  9 I
1''-ETHAFOAM2I      I    I      I    I      I    I      I    I
             I      I----I      I----I      I----I      I----I
             I           I           I           I           I 2 - 2
             I           I           I           I           I
      T      I           I           I           I           I 1 - 1
      H      I           I           I           I           I
      I      I           I           I           I           I 3 - 3
      C      I           I           I           I           I
      K  ----I------I----I------I----I------I----I------I----I 3 - 3
      N      I      I    I      I    I      I    I      I    I
      E      I      I 11 I      I  5 I      I  1 I      I  8 I 2 - 2
      S      I      I    I      I    I      I    I      I    I
      S      I      I    I      I    I      I    I      I    I 3 - 3
             I      I----I      I----I      I----I      I----I
             I           I           I           I           I 1 - 1
             I           I           I           I           I
             I           I           I           I           I 2 - 2
2''-ETHAFOAM4I           I           I           I           I
             I           I           I           I           I 1 - 1
2''-ETHAFOAM2I           I           I           I           I
         ----I------I----I------I----I------I----I------I----I 1 - 1
             I      I    I      I    I      I    I      I    I
             I      I    I      I    I      I    I      I    I 2 - 2
             I      I  3 I      I  6 I      I  4 I      I 12 I
             I      I    I      I    I      I    I      I    I 3 - 3
             I      I----I      I----I      I----I      I----I
             I           I           I           I           I
             I           I           I           I           I
             I           I           I           I           I
3''-ETHAFOAM4I           I           I           I           I
             I           I           I           I           I
3''-ETHAFOAM2I           I           I           I           I
         ----I-----------I-----------I-----------I-----------I
```

TEST PLAN

The experimental testing of any bulk cushioning material requires a documented test procedure. The test plan follows the ASTM method D 1596-64 (dynamic drop test procedure), with MICOM compensation for temperature extremes, since the ASTM method considers a temperature of 73°F only. The procedure assumes a drop tester and related data measuring equipment are available as specified by MIL-C-26861.

The test plan follows:

1. Identify approximately nine well-spaced static stress values in the 0.03 to 5.0 psi range based on the specific material to be tested (not all materials can withstand psi values greater than 2.0). Testing should be in ascending static stress value order.
2. Procure an adequate number of test specimens (8.0 by 8.0 inches) to satisfy temperature conditioning requirements and replication demands.
3. The temperatures to be utilized are -65°F, -20°F, 20°F, 70°F, 110°F, and 160°F, while drop heights are 12", 18", 24", and 30".
4. Initially, condition the test specimens by dropping on them once, one at a time, from a 24 inch drop height measured from the top of the sample, at 70°F.
5. The test specimens are loaded into conditioning bins according to the right most column on the data form. Each bin is loaded to a specific replication with a total of three replications per temperature/static stress.
6. The test specimens are conditioned to the desired temperature in the conditioning bins.

7. Only one drop is permitted per test specimen per conditioning phase. This precludes dropping on compressed test specimens, which would bias the experimental data. Test specimens are to be permitted to return to ambient temperature prior to being temperature conditioned again (approximately 24 hours).
8. The testing sequence (1-12) has been randomly established by the representative numbers for each of the twelve blocks on the data sheet. The first test specimen to be tested is identified with a one in the upper right hand corner of one of the twelve identically shaped blocks on each data sheet.
9. Two experimental values are to be recorded for each drop test. The value to be recorded on the upper portion of the individual data location is the peak G value. Below this value should be recorded the pulse width in milliseconds.

This test plan will result in valid experimental data to be utilized in the development of superimposed dynamic cushioning curves. The sections which immediately follow will address the data analysis portion of the model development methodology.

TEMPERATURE EFFECTS

The effect of temperature upon foamed bulk cushioning materials has been proven conclusively in [1 - 3]. However, there may be an occasion when the experimenter desires to verify the temperature effect upon some new, exotic bulk cushioning material. Consequently, the material which follows in this section will describe the method of identifying a significant temperature effect.

In order to test the significance of the temperature effect on cushioning material properties, an ANalysis Of VAriance (ANOVA) is conducted on the data for each material type based on the experimental split-split plot design. The appropriate ANOVA is performed on the data by utilizing a UNIVAC 1108 STAT-PACK code entitled ANOSSP (Appendix C). A summary discussion of the assumptions and calculation methodology employed in this code is contained in [1].

A sample ANOVA table is shown in Table 3. In general, it is known that a significant difference exists between stress levels or between material thicknesses. This information is provided once again as part of the ANOVA procedure. The important significance test in Table 3 is the one related to temperature. In repeated testing upon Minicel, Ethafoam 2, Ethafoam 4, Urether 3, and Urester 4, the temperature effect has always been very significant. On a new bulk cushioning material, it would be important to test for a temperature effect.

Table 3. A Sample ANOVA Table

SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARES	F-STATISTICS	PROB. > F
Stress Levels					
Replicates				---	---
Error (Stress)				---	---
Material Thickness					
Stress x Thickness					
Error (Thickness)				---	---
Temperature					
Stress x Temperature					
Thickness x Temperature					
Stress x Thickness x Temperature					
Error (Temperature)				---	---
TOTAL			---	---	---

OUTLIER DETECTION

As every experimenter knows, an observation, purportedly taken under the same conditions, may be widely different from other observations, or an outlier from the rest. Thus, the problem confronting the experimenter is whether to keep the suspect observation in computation, or whether it should be discarded as a faulty measurement.

Although many criteria have been proposed for guiding the rejection of outliers, none were found in the literature which were particularly applicable to this case. Consequently, an extension of the extreme studentized deviate from the sample mean, or Nair Criterion, was developed and incorporated as a preface program to the CVREG curvilinear regression computer code.

Referring to Table 4, the first step in the outlier subroutine is to compute the sample variances for each set of three replications of G levels to find which set has the maximum sample variance. For the set of observations having the largest variance, each observation of the set is then tested individually as a candidate for rejection as an outlier by using the statistic

$$t = \frac{|x_e - \bar{x}|}{s_\nu}$$

where x_e = an individual observation in a set of three replications

$\bar{x}$ = the sample mean of the three observations

s_ν = an independent external estimate of the standard deviation from concurrent data.

To get s_ν, the set of replications of G's having the maximum sample variance, as indicated above, corresponding to a particular stress level, is eliminated from the calculations of s_ν. From the remaining sets of replications of G's, s_ν is calculated with the expression

Table 4. Actual Basic Data Printout

Drop Height of 12"

Minicel , -65F, 1"

Stress Levels			G's		
.0400	.0400	.0400	251.0000	205.0000	175.0000
.1000	.1000	.1000	148.0000	141.0000	156.0000
.2000	.2000	.2000	89.0000	91.0000	78.0000
.4000	.4000	.4000	44.0000	50.0000	53.0000
.8000	.8000	.8000	37.0000	37.0000	37.0000
1.0000	1.0000	1.0000	34.0000	30.0000	34.0000
1.6000	1.6000	1.6000	31.0000	29.0000	32.0000
2.0000	2.0000	2.0000	35.0000	34.0000	33.0000
2.4000	2.4000	2.4000	35.0000	38.0000	32.0000
3.0000	3.0000	3.0000	45.0000	46.0000	50.0000
3.6000	3.6000	3.6000	48.0000	43.0000	48.0000
4.0000	4.0000	4.0000	52.0000	61.0000	56.0000
4.4000	4.4000	4.4000	57.0000	65.0000	68.0000
4.6000	4.6000	4.6000	61.0000	71.0000	70.0000
5.0000	5.0000	5.0000	59.0000	61.0000	69.0000

REP 1

REP 2

REP 3

$$s_\nu = \sqrt{\frac{\sum_{i=1}^{n-1} s_i^2}{n-1}}$$

where s_i^2 = sample variance of the i^{th} set of replications of G's and

n = number of stress levels.

The values of the t statistic for each observation in the set of replicates being tested are compared with the appropriate value from a t table. A point is rejected as an outlier if t(calculated) > t(table value) [10].

If an observation is rejected in the first iteration of the outlier test, the set of replicates to which it belonged is no longer considered in further calculations, but the code then moves to the set of replicates with the <u>next highest</u> sample variance to check for outliers. Iteration is continued until a set of replications is checked and no points are rejected.

Safeguards are built into the outlier test code which restrict the number of data points that can be rejected in a set of replicates. For instance, only one point can be rejected in a set of three replications. Also, if two of the three values of t are the same and are greater than the test criterion value of t, then neither of the observations is rejected. This is essentially a tie rule for a relatively rare but possible eventuality.

The outlier detection code is contained in Appendix D.

INDIVIDUAL DYNAMIC CUSHIONING CURVES

With the outlying observations removed from the data, the data is sufficiently conditioned for input into the curvilinear regression code (CVREG), which determines the equation of the statistically best fitting polynomial for use as the dynamic cushioning design curve for a particular drop height and material thickness.

A fairly rigorous method for determining the degree of the polynomial to be fitted to a given set of data consists of first fitting a straight line to a set of data, i.e., $y = b_o + b_1x$ and testing the hypothesis $\beta_1 = 0$. Then, fit a second degree polynomial and test the null hypothesis $\beta_2 = 0$, namely, that nothing is gained by including the quadratic term. If this hypothesis can be rejected, a third-degree polynomial is fitted and the hypothesis $\beta_3 = 0$ is tested, etc. This procedure is continued until the null hypothesis $\beta_i = 0$ cannot be rejected in two successive steps and, consequently, there is no apparent advantage to carrying the extra terms.

The output of the CVREG code contains the coefficients of the first, second, third, and fourth degree polynomials for the experimental data under consideration. In addition, each polynomial is accompanied by an analysis of regression variance, and an F-statistic for determining whether the third and fourth terms should be retained. In most cases, a second order polynomial is sufficient to describe the experimental data.

The CVREG code is contained in Appendix D.

MODEL DEVELOPMENT

The experimental data for a particular cushioning material from which the individual dynamic cushioning curves were developed is now utilized in total to develop a generalized model. The generalized model is predicated upon a combination of drop height, cushion thickness, cushion temperature, and static stress level mathematically stated as:

$$c_0 + \sum_{\ell=0}^{1} h^{\ell/2} \sum_{k=0}^{1} \frac{1}{T^{(k+\frac{1}{2})}} \sum_{j=1}^{3} \theta^j \sum_{i=0}^{2} c_{ijk\ell}(\ell n\ 100\ \sigma_s)^i$$

$$+ \sum_{n=1}^{3} \theta^n \sum_{m=0}^{2} c_{mn}(\ell n\ 100\ \sigma_s)^m$$

where c_0 = constant

h = drop height

T = thickness

θ = temperature = $\frac{^\circ F + 460}{100}$

σ_s = static stress.

From this general model it is necessary to select the combination of terms which best describes the situation to be modeled.

A stepwise regression procedure (Appendix E) is utilized in acquiring coefficients for the general model. A total of 45 terms are considered by the stepwise regression procedure. These terms represent the combinations of variables found to describe bulk cushioning behavior very adequately. A summary of the variable combinations is given in Table 5. The variable combinations are identified by an x in the appropriate column.

Each time a variable is added to the general model, it is necessary to evaluate the resultant dynamic cushioning curves to assure the proposed model

Table 5. Variable combination summary

Variable	Coefficient	θ	θ^2	θ^3	$h^{1/2}$	$T^{-1/2}$	$T^{-3/2}$	$(\ell n\, \sigma_s)$	$(\ell n\, \alpha_s)^2$
0									
1		x				x			
2		x				x		x	
3		x				x			x
4		x			x		x		
5		x			x		x	x	
6		x			x		x		x
7		x			x	x			
8		x			x	x		x	
9		x			x	x			x
10			x			x			
11			x			x		x	
12			x			x			x
13			x		x		x		
14			x		x		x	x	
15			x		x		x		x
16			x		x	x			
17			x		x	x		x	
18			x		x	x			x
19				x		x			
20				x		x		x	
21				x		x			x
22				x	x		x		
23				x	x		x	x	
24				x	x		x		x
25				x	x	x			
26				x	x	x		x	
27				x	x	x			x
28		x					x		
29		x					x	x	
30		x					x		x
31			x				x		
32			x				x	x	
33			x				x		x
34				x			x		
35				x			x	x	
36				x			x		x
37		x							
38		x						x	
39		x							x
40			x						
41			x					x	
42			x						x
43				x					
44				x				x	
45				x					x

is providing the hypothesized U-shaped curves which do not possess negative peak accelerations, and the curves are distinct. Obviously, many of the developed models will be similar in their predictive ability. However, first-hand observartion during the experimental phase, supported by statistical evaluations at each step of the model development effort, are the best methods to evaluate the various developed models. In addition to the developed mathematical models, the regression procedure provides printer plots for each drop height/thickness/temperature combination, to assist in a valid model selection.

MODEL VALIDATION

The model validation procedure builds on the earlier aspects of the methodology being described. In [6], a procedure for validating the model dynamic cushioning curves is documented in detail, which is predicated upon prediction limits.

The developed generalized model procedure provides dynamic cushioning curves for individual combinations of drop height, temperature, static stress, and cushion thickness. The individual dynamic cushioning curves [IDCC] are compared with the developed prediction limits to ascertain whether the generalized model is predicting the IDCC in a consistent statistical fashion. In essence, the test of the generalized model is to determine if it can provide G-level values which include the significant static stress level portion of the IDCC and still remain within the prediction limits for the particular conditions under consideration. The significant portion of the IDCC is identified as the minimum IDCC G-level value bounded by $\pm$ 1.0 psi. This significant portion may be truncated at the lower or upper static stress level if the bounds fall outside the standard static stress range of 0.05 to 5.20 psi.

The validation code is provided in Appendix F. A sample output is given in Table 6 for the Ethafoam 2 model. The minimum G-level for the IDCC is 25.67 at a static stress of 1.6 psi. Consequently, the significant portion is 0.60 to 2.60 psi as identified by the double asterisk (**) to the left of the MODEL column. Over this static stress range, it is seen that all model values are contained within the developed prediction limits for this case. Thus, the general model is found to be predicting G-level values very adequately.

Table 6. Sample Validation Output

ETHAFOAM-2 12.0 IN. D.H. 1.0 IN. THICK -65.0 TEMPERATURE

STATIC STRESS PSI	IDCC	DECELERATION LOWER-P		(G) MODEL	UPPER-P
.05	173.75	156.18		187.51	191.32
.10	119.70	102.45		125.06	136.95
.15	93.74	76.81		95.17	110.68
.20	77.86	61.23		76.95	94.49
.25	67.00	50.67		64.51	83.33
.30	59.06	43.02		55.45	75.09
.35	53.00	37.25		48.57	68.76
.40	48.25	32.78		43.18	63.72
.45	44.43	29.23		38.86	59.63
.50	41.31	26.38		35.36	56.25
.55	38.74	24.06		32.47	53.42
.60	36.59	22.16	**	30.07	51.02
.65	34.78	20.59	**	28.06	48.97
.70	33.25	19.30	**	26.37	47.21
.75	31.96	18.23	**	24.95	45.68
.80	30.85	17.35	**	23.75	44.36
.85	29.91	16.62	**	22.73	43.20
.90	29.11	16.02	**	21.87	42.19
.95	28.43	15.54	**	21.15	41.31
1.00	27.85	15.15	**	20.54	40.55
1.20	26.34	14.34	**	19.03	38.34
1.40	25.72	14.29	**	18.52	37.14
1.60	25.67	14.72	**	18.66	36.63
1.80	26.01	15.42	**	19.22	36.59
2.00	26.60	16.30	**	20.07	36.91
2.20	27.39	17.28	**	21.13	37.50
2.40	28.31	18.33	**	22.33	38.28
2.60	29.32	19.43	**	23.63	39.21
2.80	30.40	20.56		25.01	40.25
3.00	31.54	21.72		26.44	41.36
3.20	32.71	22.89		27.91	42.53
3.40	33.91	24.09		29.40	43.73
3.60	35.12	25.30		30.91	44.94
3.80	36.35	26.53		32.42	46.16
4.00	37.58	27.77		33.94	47.38
4.20	38.81	29.03		35.46	48.59
4.40	40.04	30.29		36.97	49.79
4.60	41.27	31.56		38.47	50.98
4.80	42.49	32.83		39.97	52.16
5.00	43.71	34.08		41.46	53.33
5.20	44.92	35.32		42.93	54.52

MODEL OPTIMIZATION

The development of a valid mathematical model of impact response for a particular cushioning material creates a need for specific application techniques. A common application method is a tailored cushion, where the cushion thickness is determined first, and the size of the cushion determined second. This application technique is entitled CUSHOP, referring to tailored cushion optimization.

Another common application method is encapsulation (ENCAP), where the entire surface of the protected item is encased by the cushioning material. The maximum and minimum surface areas must be determined prior to calculating the cushion thickness when the encapsulation method is utilized.

Appendix G contains the computer codes for the CUSHOP and ENCAP application procedures. Detailed instructions for the use of these codes is contained in [9]. Table 7 is a typical CUSHOP code output, from which the superimposed dynamic cushioning curves may be plotted. In a similar manner, Table 8 is a typical ENCAP code output. In both cases, tabular values are provided over a static stress range of 0.10 to 5.00 psi for a cold, ambient, and hot temperature value. In addition, the codes identify the cushion thickness required to provide the desired item protection for the application technique selected.

The cushion designer who desires a plotted set of superimposed dynamic cushioning curves for a particular set of conditions should consult [8] for complete details. A typical set of superimposed dynamic cushioning curves is given in Figure 1 for the CUSHOP application technique utilizing the Minicel cushioning material. It is seen that three and one-half inches of Minicel provide the item to be protected to a 30 G fragility level over a temperature range of -65°F to 160°F, and a drop height of 26 inches. The

Table 7. Typical CUSHOP Output

CUSHION MATERIAL OPTIMIZATION
CUSHOP

MINICEL
LOWER SS .80 UPPER SS 1.40 DROP HEIGHT 26.0 G-LEVEL 30.0
MATERIAL THICKNESS 3.50
TEMPERATURES -65.00 70.00 160.00
WEIGHT 10.00 MIN. BEARING AREA 7.14 MAX. BEARING AREA 12.50

STATIC STRESS	COLD G-LEVEL	AMBIENT G-LEVEL	HOT G-LEVEL
.05	237.81	119.68	70.31
.10	160.96	73.72	45.15
.15	123.50	53.01	34.47
.20	100.27	41.08	28.69
.25	84.17	33.41	25.25
.30	72.26	28.17	23.10
.35	63.06	24.46	21.76
.40	55.74	21.78	20.94
.45	49.78	19.82	20.49
.50	44.84	18.40	20.30
.55	40.70	17.38	20.30
.60	37.18	16.67	20.44
.65	34.17	16.20	20.69
.70	31.58	15.92	21.03
.75	29.33	15.80	21.43
.80	27.37	15.80	21.89
.85	25.65	15.92	22.39
.90	24.15	16.11	22.92
.95	22.83	16.39	23.47
1.00	21.67	16.72	24.05
1.05	20.65	17.10	24.64
1.10	19.75	17.53	25.25
1.15	18.96	17.99	25.86
1.20	18.26	18.49	26.48
1.25	17.65	19.01	27.11
1.30	17.12	19.56	27.74
1.35	16.65	20.12	28.37
1.40	16.25	20.71	29.01
1.45	15.90	21.30	29.64
1.50	15.61	21.91	30.28
1.55	15.36	22.53	30.91
1.60	15.15	23.16	31.54
1.65	14.99	23.80	32.18
1.70	14.85	24.44	32.80
1.75	14.75	25.08	33.43
1.80	14.69	25.73	34.05
1.85	14.64	26.39	34.67
1.90	14.63	27.05	35.28
1.95	14.63	27.70	35.90
2.00	14.66	28.36	36.50
2.05	14.71	29.03	37.11
2.10	14.78	29.69	37.71
2.15	14.86	30.35	38.31
2.20	14.96	31.01	38.90
2.25	15.08	31.67	39.49

Table 7. Typical CUSHOP Output (concluded)

2.30	15.21	32.33	40.07
2.35	15.35	32.99	40.65
2.40	15.50	33.64	41.23
2.45	15.67	34.30	41.80
2.50	15.84	34.95	42.37
2.55	16.03	35.60	42.93
2.60	16.22	36.25	43.49
2.65	16.43	36.90	44.05
2.70	16.64	37.54	44.60
2.75	16.86	38.18	45.15
2.80	17.08	38.82	45.69
2.85	17.32	39.46	46.23
2.90	17.56	40.10	46.77
2.95	17.80	40.73	47.30
3.00	18.05	41.36	47.83
3.05	18.31	41.98	48.35
3.10	18.57	42.61	48.87
3.15	18.83	43.23	49.39
3.20	19.10	43.85	49.90
3.25	19.37	44.46	50.41
3.30	19.65	45.07	50.92
3.35	19.93	45.68	51.42
3.40	20.21	46.29	51.92
3.45	20.50	46.89	52.42
3.50	20.79	47.49	52.91
3.55	21.08	48.09	53.40
3.60	21.38	48.69	53.89
3.65	21.68	49.28	54.37
3.70	21.97	49.87	54.85
3.75	22.28	50.45	55.33
3.80	22.58	51.04	55.80
3.85	22.88	51.62	56.27
3.90	23.19	52.19	56.74
3.95	23.50	52.77	57.20
4.00	23.81	53.34	57.67
4.05	24.12	53.91	58.12
4.10	24.43	54.48	58.58
4.15	24.75	55.04	59.03
4.20	25.06	55.60	59.48
4.25	25.38	56.16	59.93
4.30	25.70	56.72	60.37
4.35	26.01	57.27	60.82
4.40	26.33	57.82	61.26
4.45	26.65	58.37	61.69
4.50	26.97	58.91	62.13
4.55	27.29	59.45	62.56
4.60	27.61	59.99	62.99
4.65	27.93	60.53	63.41
4.70	28.26	61.06	63.84
4.75	28.58	61.60	64.26
4.80	28.90	62.13	64.68
4.85	29.22	62.65	65.09
4.90	29.55	63.18	65.51
4.95	29.87	63.70	65.92
5.00	30.19	64.22	66.33

Table 8. Typical ENCAP Output

CUSHION MATERIAL OPTIMIZATION
ENCAP

ETHAFOAM-2
LOWER SS .83 UPPER SS 1.00 DROP HEIGHT 23.0 G-LEVEL 35.0
MATERIAL THICKNESS 2.50
TEMPERATURES -65.00 70.00 160.00
WEIGHT 10.00 MIN. BEARING AREA 10.00 MAX. BEARING AREA 12.00

STATIC STRESS	COLD G-LEVEL	AMBIENT G-LEVEL	HOT G-LEVEL
.05	284.36	154.25	61.86
.10	190.21	95.03	41.61
.15	144.57	67.93	33.28
.20	116.42	52.08	28.95
.25	96.99	41.71	26.49
.30	82.08	34.50	25.06
.35	71.69	29.28	24.27
.40	62.97	25.40	23.88
.45	55.91	22.49	23.77
.50	50.10	20.28	23.86
.55	45.24	18.61	24.10
.60	41.15	17.35	24.44
.65	37.66	16.41	24.85
.70	34.67	15.74	25.33
.75	32.10	15.29	25.85
.80	29.88	15.01	26.41
.85	27.96	14.87	26.99
.90	26.28	14.86	27.59
.95	24.83	14.95	28.21
1.00	23.56	15.12	28.83
1.05	22.46	15.37	29.47
1.10	21.50	15.69	30.11
1.15	20.67	16.05	30.76
1.20	19.96	16.47	31.40
1.25	19.35	16.92	32.05
1.30	18.82	17.42	32.70
1.35	18.38	17.94	33.34
1.40	18.01	18.49	33.98
1.45	17.71	19.06	34.62
1.50	17.47	19.65	35.26
1.55	17.28	20.26	35.89
1.60	17.14	20.88	36.52
1.65	17.05	21.52	37.15
1.70	17.00	22.17	37.77
1.75	16.98	22.83	38.39
1.80	17.00	23.50	39.00
1.85	17.05	24.18	39.61
1.90	17.13	24.86	40.21
1.95	17.24	25.55	40.81
2.00	17.37	26.24	41.40
2.05	17.53	26.94	41.99
2.10	17.70	27.63	42.57
2.15	17.90	28.34	43.15
2.20	18.11	29.04	43.72
2.25	18.34	29.75	44.29

Table 8. Typical ENCAP Output (concluded)

2.30	18.59	30.45	44.85
2.35	18.85	31.16	45.41
2.40	19.13	31.87	45.96
2.45	19.41	32.57	46.51
2.50	19.71	33.28	47.06
2.55	20.02	33.99	47.60
2.60	20.34	34.69	48.13
2.65	20.67	35.40	48.67
2.70	21.01	36.10	49.19
2.75	21.35	36.80	49.72
2.80	21.71	37.50	50.23
2.85	22.07	38.20	50.75
2.90	22.44	38.90	51.26
2.95	22.81	39.59	51.76
3.00	23.19	40.29	52.27
3.05	23.57	40.98	52.76
3.10	23.96	41.67	53.26
3.15	24.36	42.35	53.75
3.20	24.76	43.04	54.24
3.25	25.16	43.72	54.72
3.30	25.57	44.40	55.20
3.35	25.98	45.07	55.67
3.40	26.39	45.75	56.15
3.45	26.81	46.42	56.62
3.50	27.23	47.09	57.08
3.55	27.65	47.76	57.54
3.60	28.07	48.42	58.00
3.65	28.50	49.08	58.46
3.70	28.93	49.74	58.91
3.75	29.36	50.39	59.36
3.80	29.79	51.05	59.80
3.85	30.23	51.70	60.24
3.90	30.66	52.35	60.68
3.95	31.10	52.99	61.12
4.00	31.54	53.63	61.55
4.05	31.98	54.27	61.98
4.10	32.42	54.91	62.41
4.15	32.86	55.54	62.84
4.20	33.30	56.17	63.26
4.25	33.75	56.80	63.68
4.30	34.19	57.43	64.09
4.35	34.63	58.05	64.51
4.40	35.08	58.67	64.92
4.45	35.52	59.29	65.33
4.50	35.97	59.90	65.73
4.55	36.42	60.52	66.14
4.60	36.86	61.13	66.54
4.65	37.31	61.73	66.94
4.70	37.75	62.34	67.33
4.75	38.20	62.94	67.73
4.80	38.65	63.54	68.12
4.85	39.09	64.14	68.51
4.90	39.54	64.73	68.89
4.95	39.98	65.32	69.28
5.00	40.43	65.91	69.66

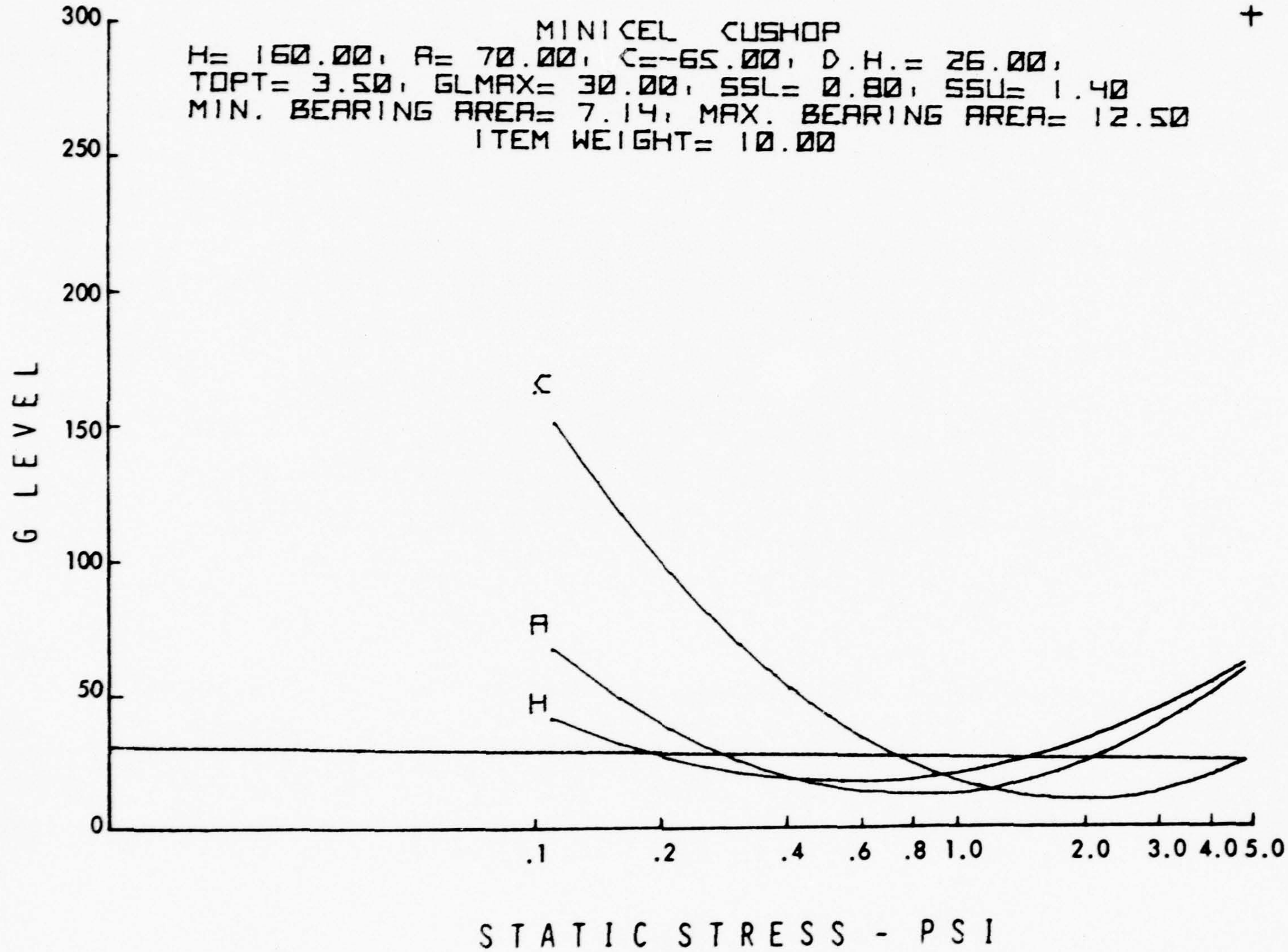

Figure 1. Minicel "CUSHOP"

tailored cushion would be 7.14 inches square at a static stress level of 0.80, or some combination between these two extremes. Only values of static stress which extend from 0.80 and including 1.40 psi can be utilized for the solution to this situation.

Similar superimposed dynamic cushioning curve results are obtained with the ENCAP code, for a particular set of conditions. Figure 2 is a typical output of the ENCAP application technique.

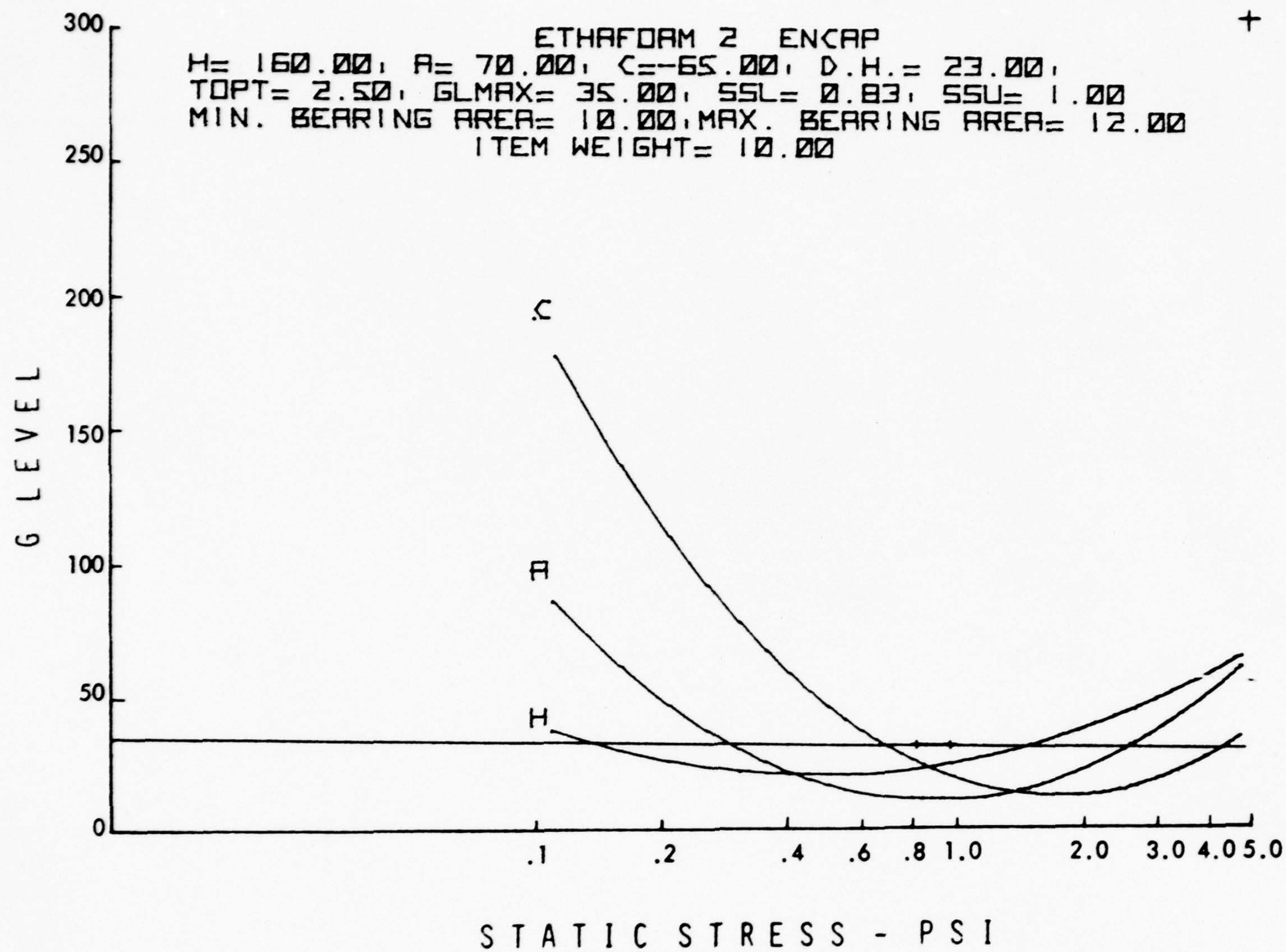

Figure 2. Ethafoam 2 "ENCAP"

CONCLUSIONS

This report documents the step by step procedure necessary to develop a bulk cushioning model for use in either a tailored or encapsulation application. Appropriate computer codes are included in the Appendices for the model development methodology. Any new bulk cushioning material can be modeled utilizing the developed methodology.

REFERENCES

1. Wyskida, R. M. and M. R. Wilhelm, Temperature Sensitive Dynamic Cushioning Function Development and Validation for Hercules Minicel Thermoplastic Foam, UAH Research Report No. 159, Vol. I, Huntsville, Alabama, September 1974.

2. Wyskida, R. M. M. R. Wilhelm, and J. D. Bynum, Temperature Sensitive Dynamic Cushioning Function Development and Validation for Polyester and Polyether Type Polyurethane Foam, UAH Research Report No. 159, Vol. III, Huntsville, Alabama, December 1974.

3. Wyskida, R. M., M. R. Wilhelm, and J. D. Bynum, Temperature Sensitive Dynamic Cushioning Function Development and Validation for DOW Ethafoam Polyethylene Foam, UAH Research Report No. 172, Vol. I, Huntsville, Alabama, July 1975.

4. Wyskida, R. M., M. R. Wilhelm, and J. D. Bynum, Temperature Sensitive Dynamic Cushioning Function Development and Validation for Blocksom Rubberized Hair, UAH Research Report No. 172, Vol. III, Huntsville, Alabama, July 1975.

5. Wyskida, R. M., M. R. Wilhelm, and J. D. Johannes, Development and Application of Confidence Intervals and Prediction Limits on Dynamic Cushioning Functions for Selected Temperature Sensitive Cushioning Materials, UAH Research Report No. 180, Vol. V, Huntsville, Alabama, October 1975.

6. Wyskida, R. M. and J. D. Johannes, Validation of Generalized Cushioning Models for Selected Temperature Sensitive Cushioning Materials, UAH Report No. 187, Vol. I, Huntsville, Alabama, October 1976.

7. Wyskida, R. M., J. D. Johannes, and M. R. Wilhelm, Container Cushioning Design Engineer Users Manual, UAH Report No. 187, Vol. II, Huntsville, Alabama, October 1976.

8. Wyskida, R. M., J. D. Johannes, and M. R. Wilhelm, Container Cushioning Design Engineer Users Manual (HP-9815A Version), UAH Report No. 203, Vol. I, Huntsville, Alabama, August 1977.

9. Wyskida, R. M. and J. D. Johannes, Container Cushioning Design Engineer Users Manual (CDC-6600 FORTRAN Version), UAH Report 203, Vol. II, June 1978.

10. Natrella, M. G., Experimental Statistics, National Bureau of Standards Handbook 91, August 1963.

APPENDIX A

Single Material Data Collection Form Generator Code

The Single Material Data Collection Form Generator Code will generate and print any specified number of special data collection forms for a given specific material, as exemplified by Table 1 of this report. The number of forms and material name are input by the user.

```
C
C****************************************************************
C
C      SINGLE MATERIAL DATA COLLECTION FORM GENERATOR
C
C****************************************************************
C
C      THIS PROGRAM RANDOMLY ORDERS THE NUMBERS FROM 1 TO 12 USING A
C      UNIFORM RANDOM NUMBER GENERATOR, THEN CALLS A SUBROUTINE WHICH
C      PRINTS FORMS CONTAINING THE RANDOMIZED CUSHION THICKNESSES.
C
       COMMON /NUMS/ NUM(12),X(12),NUMR(12),NEW(12)
       COMMON /MATT/ MAT(3),MT1,MT2,MT3
C
C      READ THE NUMBER OF FORMS TO PRINT, NF, AND THE RANDU SEED X(1).
C
       READ (5,400) NF,X(1)
C
C      EXAMPLE COL
C                 12345678901234567890
C                   5 4871.
C
C      READ THE MATERIAL NAMES AND THE SAMPLE THICKNESSES
C
       READ (5,500) (MAT(I),I=1,3),MT1,MT2,MT3
C
C      EXAMPLE COL
C                 12345678901234567890
C                 ETHAFOAM-2   1 2 3
C
       DO 100 I=1,12
       NUM(I)=I
100    CONTINUE
C
C      GENERATE AND PRINT NF FORMS
C
       DO 300 I=1,NF
       CALL RANDU (X,12)
C
C      ORDER THE SAMPLE THICKNESSES BY RANDOM ARRAY
C
       CALL SORT (12)
       NEW(1)=MT1
       NEW(2)=MT1
       NEW(3)=MT1
       NEW(4)=MT1
       NEW(5)=MT2
       NEW(6)=MT2
       NEW(7)=MT2
       NEW(8)=MT2
       NEW(9)=MT3
       NEW(10)=MT3
```

```
      NEW(11)=MT3
      NEW(12)=MT3
      DO 200 K=1,12
      ISUB=NUM(K)
      NUMR(ISUB)=NEW(K)
200   CONTINUE
C
C     GENERATE NEW RANDU SEED
C
      X(1)=X(12)*1.0E4+1.0
      CALL WRITE
300   CONTINUE
      WRITE (6,600)
      STOP
C
C
400   FORMAT (I3,F6.0)
500   FORMAT (3A4,3I2)
600   FORMAT (1H1)
      END
```

```
      SUBROUTINE SORT(N)
      COMMON /NUMS/ NUM(12),X(12),NUMR(12),NEW(12)
C
C     SUBROUTINE FOR SORTING N RANDOM NUMBERS IN ASCENDING ORDER
C     AND ARRANGES THE CORRESPONDING THICKNESSES, NUM, ACCORDING TO
C     THIS RANDOMIZATION SCHEME.
C
      M=N
100   CONTINUE
      M=M/2
      IF (M.EQ.0) RETURN
      K=N-M
      J=1
200   CONTINUE
      I=J
300   CONTINUE
      II=I+M
      IF (X(I).LT.X(II)) GO TO 400
      F=X(I)
      X(I)=X(II)
      X(II)=F
      NF=NUM(I)
      NUM(I)=NUM(II)
      NUM(II)=NF
      I=I-M
      IF (I.GE.1) GO TO 300
400   CONTINUE
      J=J+1
      IF (J.GT.K) GO TO 100
      GO TO 200
      END
```

```
      SUBROUTINE WRITE
C
C     SUBROUTINE WHICH PRINT THE RANDOMIZED FORMS FOR
C     DROP HEIGHTS OF 12'',18'',24'', AND 30''.
C
      COMMON /NUMS/ NUM(12),X(12),NUMR(12),NEW(12)
      COMMON /MATT/ MAT(3),MT1,MT2,MT3
      WRITE (6,100) (MAT(I),I=1,3)
      WRITE (6,200)
      WRITE (6,300)
      WRITE (6,400)
      WRITE (6,400)
      WRITE (6,500) (NUM(I),I=1,4)
      WRITE (6,1900) NUMR(1)
      WRITE (6,400)
      WRITE (6,600)
      WRITE (6,700) MT1
      WRITE (6,1900) NUMR(2)
      WRITE (6,800)
      WRITE (6,800)
      WRITE (6,1900) NUMR(3)
      WRITE (6,800)
      WRITE (6,800)
      WRITE (6,1900) NUMR(4)
      WRITE (6,300)
      WRITE (6,900)
      WRITE (6,400)
      WRITE (6,1000)
      WRITE (6,1900) NUMR(5)
      WRITE (6,400)
      WRITE (6,1100) (NUM(I),I=5,8)
      WRITE (6,400)
      WRITE (6,1200)
      WRITE (6,1900) NUMR(6)
      WRITE (6,600)
      WRITE (6,700) MT2
      WRITE(6,1300)
      WRITE (6,800)
      WRITE (6,1400)
      WRITE (6,1900) NUMR(7)
      WRITE (6,800)
      WRITE (6,1500)
      WRITE (6,800)
      WRITE (6,1600)
      WRITE (6,1900) NUMR(8)
      WRITE (6,800)
      WRITE (6,1600)
      WRITE (6,300)
      WRITE (6,1900) NUMR(9)
      WRITE (6,400)
      WRITE (6,400)
      WRITE (6,1900) NUMR(10)
```

```
      WRITE (6,500) (NUM(I),I=9,12)
      WRITE (6,400)
      WRITE (6,600)
      WRITE (6,1900) NUMR(11)
      WRITE (6,700) MT3
      WRITE (6,800)
      WRITE (6,600)
      WRITE (6,1900) NUMR(12)
      WRITE (6,600)
      WRITE (6,800)
      WRITE (6,1800)
      RETURN
C
C
100   FORMAT (1H1,25X,'SINGLE MATERIAL',//,28X,3A4,///,12X,'STRESS LEVEL
     1:',15X,'REPLICATION:'///12X,'TEMPERATURE:'////)
200   FORMAT (1H ,34X,'DROP HEIGHT'//16X,'I',4X,4H12'',3X,'I',4X,4H18'',
     13X,'I',4X,4H24'',3X,'I',4X,4H30'',3X,'I')
300   FORMAT (12X,'----I',4('------I----I'))
400   FORMAT (16X,'I',4(6X,'I     I'))
500   FORMAT (1H+,24X,4(I2,10X))
600   FORMAT (16X,'I',4(6X,'I----I'))
700   FORMAT (1H+,11X,I1,2H'')
800   FORMAT (16X,'I',4(11X,'I'))
900   FORMAT (1H+,9X,'T')
1000  FORMAT (1H+,9X,'H')
1100  FORMAT (1H+,9X,'I',5X,'I',8X,4(I2,10X))
1200  FORMAT (1H+,9X,'C')
1300  FORMAT (1H+,9X,2HK ,I1,2H'')
1400  FORMAT (1H+,9X,'N')
1500  FORMAT (1H+,9X,'E')
1600  FORMAT (1H+,9X,'S')
1700  FORMAT (1H+,11X,I1,2H'')
1800  FORMAT (12X,'----I',4('-----------I'))
1900  FORMAT (1H+,70X,I1,2H'')
      END
```

APPENDIX B

Composite Material Data Collection Form Generator Code

The Composite Material Data Collection Form Generator Code will generate and print any specified number of special data collection forms for a given specific material, as exemplified by Table 2 of the report. The number of forms and material names are input by the user.

```
C*****************************************************************************
C
C          COMPOSITE TWO MATERIAL DATA COLLECTION FORM GENERATOR
C
C*****************************************************************************
C      THIS PROGRAM RANDOMLY ORDERS THE NUMBERS FROM 1 TO 12 USING A
C      UNIFORM RANDOM NUMBER GENERATOR, THEN CALLS A SUBROUTINE WHICH
C      PRINTS FORMS CONTAINING THE RANDOMIZED NUMBERS.
C
       COMMON /NUMS/ NUM(12),X(12), NUMR(12),NEW(12)
       COMMON /MT/ MAT1(3),MAT2(3)
C
C      READ THE NUMBER OF FORMS TO PRINT, NF, AND THE RANDU SEED X(1).
C
       READ(5,600) NF,X(1)
C         EXAMPLE: COL
C                          123456789012345678901234567890
C DATA                     300 4672.
C
C
C      READ THE MATERIAL NAMES AND THE SAMPLE THICKNESSES
C
       READ(5,400) MAT1,MAT2,MT1,MT2,MT3
C
C         EXAMPLE: COL
C                          123456789012345678901234567890
C      DATA                ETHAFOAM2   ETHAFOAM4    1 2 3
C
       DO 100 I=1,12
       NUM(I)=I
100    CONTINUE
C
C      GENERATE AND PRINT NF FORMS
C
       DO 300 I=1,NF
       CALL RANDU (X,12)
       CALL SORT(12)
       NEW( 1) = MT1
       NEW( 2) = MT1
       NEW( 3) = MT1
       NEW( 4) = MT1
       NEW( 5) = MT2
       NEW( 6) = MT2
       NEW( 7) = MT2
       NEW( 8) = MT2
       NEW( 9) = MT3
       NEW(10) = MT3
       NEW(11) = MT3
       NEW(12) = MT3
       DO 200 K=1,12
       ISUB = NUM(K)
       NUMR(ISUB) = NEW(K)
```

```
200    CONTINUE
C
C      GENERATE NEW RANDU SEED
C
       X(1)=X(12)*1.0E4+1.0
       CALL WRITE
300    CONTINUE
       WRITE (6,500)
       STOP
C
400    FORMAT(6A4,3I2)
500    FORMAT (1H1)
600    FORMAT ()
       END
```

```
      SUBROUTINE SORT(N)
C
C     SUBROUTINE FOR SORTING N RANDOM NUMBERS IN ASCENDING ORDER
C     AND ARRANGES THE CORRESPONDING THICKNESSES, NUM, ACCORDING TO
C     THIS RANDOMIZATION SCHEME.
C
      COMMON /NUMS/ NUM(12),X(12), NUMR(12),NEW(12)
C
      M=N
100   CONTINUE
      M=M/2
      IF (M.EQ.0) RETURN
      K=N-M
      J=1
200   CONTINUE
      I=J
300   CONTINUE
      II=I+M
      IF (X(I).LT.X(II)) GO TO 400
      F=X(I)
      X(I)=X(II)
      X(II)=F
      NF=NUM(I)
      NUM(I)=NUM(II)
      NUM(II)=NF
      I=I-M
      IF (I.GE.1) GO TO 300
400   CONTINUE
      J=J+1
      IF (J.GT.K) GO TO 100
      GO TO 200
      END
```

```
      SUBROUTINE WRITE
C
C     GENERATE AND PRINT NF FORMS
C
      COMMON /NUMS/ NUM(12),X(12), NUMR(12),NEW(12)
      COMMON /MT/ MAT1(3),MAT2(3)
C
      K=1
      WRITE (6,100) MAT1,MAT2
      WRITE (6,200)
      WRITE (6,300)
      WRITE (6,400)
      WRITE (6,400)
      WRITE (6,1800) K,MAT2(1),MAT2(2),MAT2(3)
      WRITE (6,400)
      WRITE (6,500) (NUM(I),I=1,4)
      WRITE (6,400)
      WRITE (6,1800) K,MAT1(1),MAT1(2),MAT1(3)
      WRITE (6,600)
      WRITE (6,700)
      WRITE (6,1700) NUMR(1),NUMR(1)
      WRITE (6,700)
      WRITE (6,700)
      WRITE (6,800)
      WRITE (6,1700) NUMR(2),NUMR(2)
      WRITE (6,700)
      WRITE (6,900)
      WRITE (6,700)
      WRITE (6,1000)
      WRITE (6,1700) NUMR(3),NUMR(3)
      WRITE (6,700)
      WRITE (6,1100)
      WRITE (6,300)
      WRITE (6,1200)
      WRITE (6,1700) NUMR(4),NUMR(4)
      WRITE (6,400)
      WRITE (6,1300)
      WRITE (6,400)
      WRITE (6,1400)
      WRITE (6,1700) NUMR(5),NUMR(5)
      WRITE (6,500) (NUM(I),I=5,8)
      WRITE (6,400)
      WRITE (6,1500)
      WRITE (6,400)
      WRITE (6,1500)
      WRITE (6,1700) NUMR(6),NUMR(6)
      WRITE (6,600)
      WRITE (6,700)
      WRITE (6,1700) NUMR(7),NUMR(7)
      WRITE (6,700)
      WRITE (6,700)
      WRITE (6,1700) NUMR(8),NUMR(8)
```

```
      WRITE (6,700)
      K=K+1
      WRITE (6,1800) K,MAT2(1),MAT2(2),MAT2(3)
      WRITE (6,700)
      WRITE (6,1700) NUMR(9),NUMR(9)
      WRITE (6,700)
      WRITE (6,1800) K,MAT1(1),MAT1(2),MAT1(3)
      WRITE (6,300)
      WRITE (6,1700) NUMR(10),NUMR(10)
      WRITE (6,400)
      WRITE (6,400)
      WRITE (6,1700) NUMR(11),NUMR(11)
      WRITE (6,400)
      WRITE (6,500) (NUM(I),I=9,12)
      WRITE (6,400)
      WRITE (6,1700) NUMR(12),NUMR(12)
      WRITE (6,600)
      WRITE (6,700)
      WRITE (6,700)
      WRITE (6,700)
      WRITE (6,700)
      K=K+1
      WRITE (6,1800) K,MAT2(1),MAT2(2),MAT2(3)
      WRITE (6,700)
      WRITE (6,700)
      WRITE (6,1800) K,MAT1(1),MAT1(2),MAT1(3)
      WRITE (6,1600)
      RETURN
C
100   FORMAT (1H1,28X,'COMPOSITE  MATERIALS',//25X,'INNER(BOTTOM)',5X,'O
     1UTER(TOP)',//,26X,3A4,T45,3A4,///,12X,'STRESS LEVEL:',15X,'REPLICA
     2TION:'///12X,'TEMPERATURE:'///)
200   FORMAT (1H ,34X,'DROP HEIGHT'//20X,'I',4X,4H12'',3X,'I',4X,4H18'',
     13X,'I',4X,4H24'',3X,'I',4X,4H30'',3X,'I')
300   FORMAT (16X,'----I',4('------I----I'))
400   FORMAT (20X,'I',4(6X,'I    I'))
500   FORMAT (1H+,28X,4(I2,10X))
600   FORMAT (20X,'I',4(6X,'I----I'))
700   FORMAT (20X,'I',4(11X,'I'))
800   FORMAT (1H+,9X,'T')
900   FORMAT (1H+,9X,'H')
1000  FORMAT (1H+,9X,'I')
1100  FORMAT (1H+,9X,'C')
1200  FORMAT (1H+,9X,1HK)
1300  FORMAT (1H+,9X,'N')
1400  FORMAT (1H+,9X,'E')
1500  FORMAT (1H+,9X,'S')
1600  FORMAT (16X,'----I',4('-----------I'))
1700  FORMAT (1H+,69X,I1,3H - ,I1)
1800  FORMAT (1H+,6X,I1,3H''-,3A4)
      END
```

APPENDIX C

Analysis of Variance Code

The main driver of this code establishes dimensions, reads the input data, and calls the split-split plot analysis of variance UNIVAC STAT-PACK program ANOSSP as a subroutine.

The following is a description of the necessary input parameters and floating point arrays that are created in ANOSSP.

Y	is a NI by NJ by NK by NL, four-dimensional array of observations.	floating-point array; input
NI	is the number of main treatment factor A levels.	FORTRAN integer; input
NJ	is the number of subtreatment factor B levels.	FORTRAN integer; input
NK	is the number of subtreatment factor C levels.	FORTRAN integer; input
NL	is the number of replications per experimental cell, NL > 1.	FORTRAN integer; input
SS	is a 12-element array containing sums of squares.	floating-point array; output
IDF	is a 12-element array containing degrees of freedom.	FORTRAN integer array; output
F	is a 10-element array containing F-statistics.	floating-point array; output
PF	is a 10-element array containing probabilities that the F-statistics are exceeded.	floating-point array; output
YMNSQ	is a 11-element array containing mean squares.	floating-point array; output

The above five arrays are ordered as follows:

In program execution, ANOSSP employs the logic given below:

(1) Sum the observations for the grand total, factor level sums, and cross level sums.

(2) Divide grand total by the number of observations to obtain grand mean.

(3) Calculate factor A level means, estimates of effects, and sum of squares.

(4) Calculate factor B level means, estimates of effects, and sum of squares.

(5) Calculate factor C level means, estimates of effects, and sum of squares.

(6) Calculate replication level means, and sum of squares.

(7) Process the first order interaction parameters.

(8) Calculate AxB interaction level means, estimates of effects and sum of squares.

(9) Calculate AxC interaction level means, estimates of effects and sum of squares.

(10) Calculate Ax replication interaction level means and sum of squares for Error (A).

(11) Calculate BxC interaction level means, estimates of effects and sum of squares.

(12) Calculate the Bx replication interaction level means, and sum of squares for one component of Error (B).

(13) Process the second order interaction parameters.

(14) Calculate the AxBxC interaction level means, estimates of effects, and sum of squares.

(15) Calculate the AxBx replication interaction level means and sum of squares for the second component of Error (B).

(16) Multiply each source's sum of squares by an appropriate factor to obtain the final sum of squares.

(17) Compute total sum of squares, the residuals, and the residual sum of squares.

(18) Subtract the replication, Error (A), and Error (B) sum of squares from the residual sum of squares to obtain Error (C) sum of squares.

(19) Set up the degrees of freedom.

(20) Set up the mean squares.

(21) Divide each source's mean square by the appropriate error mean square term in each Error (C), Error (B), or Error (A) group to obtain the F-statistics.

(22) Call FISH to calculate the probabilities that the F-statistics are exceeded. FISH is a required subprogram but other subprograms which compute F-statisitcs can be utilized.

```
C       ******************************************************************
C
C        PROGRAM USES  STAT-PACK ROUTINE ANOSSP (SPLIT-SPLIT PLOT DESIGN)
C        TO CALCULATE ANALYSIS OF VARIANCE ON CUSHIONING MATERIAL
C       DATA USING NI STRESS LEVELS, 3 THICKNESSES, 3 TEMPERATURES, AND
C       3 REPLICATIONS.
C
C       ******************************************************************
C
      PARAMETER NI=15,NJ=3,NK=3,NL=3,ISUM4=24,IPROD=255
C     ISUM4 IS NI+NJ+NK+NL
C     IPROD IS ((NI+1)*(NJ+1)*(NK+1)-1)
      DIMENSION Y(NI,NJ,NK,NL),SS(12),IDF(12),F(12),PF(12),YMNSQ(12),
     1ALPHA(IPROD),YFTMN(ISUM4),YMNIL(NI,NL),YMNJL(NJ,NL),YMNIJ(NI,NJ),
     2YMIJL(NI,NJ,NL),YMIJK(NI,NJ,NK),YMNIK(NI,NK),YMNJK(NJ,NK),
     3R(NI,NJ,NK,NL)
      MI=NI
      MJ=NJ
      MK=NK
C     THERE IS A SEPARATE CARD FOR EACH DATA SHEET. LOOPS FROM INSIDE
C     OUT ARE TEMPERATURE, THICKNESS, REPLICATION, AND STRESS.
C     INPUT FORMAT IS FREE FIELD.
      DO 110 I=1,NI
      DO 105 L=1,NL
      READ(5,1000) ((Y(I,J,K,L),K=1,NK),J=1,NJ)
  105 CONTINUE
  110 CONTINUE
      CALL ANOSSP(Y,NI,NJ,NK,NL,SS,IDF,F,PF,YMNSQ,ALPHA,YFTMN,YMNIL,
     1YMNJL,YMNIJ,YMIJL,YMEAN,YMIJK,YMNIK,YMNJK,IERR,$130,R,MI,MJ,MK)
C     WRITE OUTPUT
      WRITE(6,1900) ((((Y(I,J,K,L),K=1,NK),J=1,NJ),L=1,NL),I=1,NI)
  120 WRITE(6,2000) (I,SS(I),IDF(I),YMNSQ(I),F(I),PF(I),I=1,12)
      WRITE(6,2010) YMEAN,(ALPHA(I),I=1,IPROD)
      WRITE(6,2020)((((R(I,J,K,L),L=1,NL),K=1,NK),J=1,NJ),I=1,NI)
      WRITE(6,2030) (YFTMN(I),I=1,ISUM4)
      WRITE(6,2040) ((YMNIL(I,L),L=1,NL),I=1,NI)
      WRITE(6,2050) ((YMNJL(J,L),L=1,NL),J=1,NJ)
      WRITE(6,2060) ((YMNIJ(I,J),J=1,NJ),I=1,NI)
      WRITE(6,2070)(((YMIJL(I,J,L),L=1,NL),J=1,NJ),I=1,NI)
      WRITE(6,2080)(((YMIJK(I,J,K),K=1,NK),J=1,NJ),I=1,NI)
      WRITE(6,2090) ((YMNIK(I,K),K=1,NK),I=1,NI)
      WRITE(6,2100) ((YMNJK(J,K),K=1,NK),J=1,NJ)
      STOP
  130 WRITE(6,2110) IERR
      GO TO 120
C     FORMATS
 1000 FORMAT()
 1900 FORMAT(1H ,9E14.6)
 2000 FORMAT('1 I',2X,'SUMS OF SQUARES',3X,'DFS',5X,'MEAN SQUARES',5X,
     1'F-STATISTICS',5X,'PROB.GT.F'/(1X,I2,2X,E15.6,3X,I3,2X,E15.6,2X,
     2E15.6,2X,E15.6))
 2010 FORMAT('0ARRAY OF GRAND MEAN AND EFFECTS'/(1X,9E14.6))
```

```
2020 FORMAT('0RESIDUALS'/9X,'REPLICATE 1',9X,'REPLICATE 2',9X,
    1'REPLICATE 3'/(5X,E15.6,5X,E15.6,5X,E15.6))
2030 FORMAT('0FACTOR LEVEL MEANS'/(1X,9E14.6))
2040 FORMAT('0STRESS LEVEL X REPLICATIONS'/(1X,3E15.6))
2050 FORMAT('0THICKNESS X REPLICATIONS'/(1X,3E15.6))
2060 FORMAT('0STRESS LEVEL X THICKNESS'/(1X,3E15.6))
2070 FORMAT('0STRESS LEVEL X THICKNESS X REPLICATIONS'/(1X,6E15.6))
2080 FORMAT('0STRESS LEVEL X THICKNESS X TEMPERATURE'/(1X,6E15.6))
2090 FORMAT('0STRESS LEVEL X TEMPERATURE'/(1X,3E15.6))
2100 FORMAT('0THICKNESS X TEMPERATURE'/(1X,3E15.6))
2110 FORMAT('0ERROR EXIT FROM SUBROUTINE, IERR = ',I2)
     END
```

```
C        ***********************************************************************
C        SUBROUTINE FOR A SPLIT SPLIT PLOT DESIGN ANALYSIS OF VARIANCE
C        ***********************************************************************
         SUBROUTINE ANOSSP (Y,NI,NJ,NK,NL,SS,IDF,F,PF,YMNSQ,ALPHA,YFTMN,
        1YMNIL,YMNJL,YMNIJ,YMIJL,YMEAN,YMIJK,YMNIK,YMNJK,IERR,$,R,
        2MI,MJ,MK)
         DIMENSION Y(MI,MJ,MK,1),SS(1),IDF(1),F(1),PF(1),YMNSQ(1),ALPHA(1),
        1YFTMN(1),YMNIL(MI,1),YMNJL(MJ,1),YMNIJ(MI,1),YMIJL(MI,MJ,1),
        2YMIJK(MI,MJ,1),YMNIK(MI,1),YMNJK(MJ,1),R(MI,MJ,MK,1)
C        ***********************************************************************
C        INITIALIZATIONS
C        ***********************************************************************
         XNI=NI
         XNJ=NJ
         XNK=NK
         XNL=NL
         NIJ=NI+NJ
         NIJK=NIJ+NK
         NIJKL=NIJK+NL
         SDUM6=0.
         XDIJ=XNI*XNJ
         XDIK=XNI*XNK
         XDIL=XNI*XNL
         XDJK=XNJ*XNK
         XDKL=XNK*XNL
         XDJL=XNJ*XNL
         XDIJK=XDIJ*XNK
         XDJKL=XDJK*XNL
         XDIKL=XDIK*XNL
         XDIJL=XDIJ*XNL
         DO 1 I=1,12
       1 SS(I)=0.
         DO 2 I=1,NIJKL
       2 YFTMN(I)=0.
         DO 5 I=1,NI
         DO 5 J=1,NJ
         DO 4 K=1,NK
         DO 3 L=1,NL
         YMIJL(I,J,L)=0.
         YMNJL(J,L)=0.
       3 YMNIL(I,L)=0.
         YMIJK(I,J,K)=0.
         YMNJK(J,K)=0.
       4 YMNIK(I,K)=0.
       5 YMNIJ(I,J)=0.
         YMEAN=0.
C        ***********************************************************************
C        BEGIN SUMMING LEVELS
C        ***********************************************************************
         DO 40 I=1,NI
         DO 30 J=1,NJ
         DO 20 K=1,NK
```

```
      DO 10 L=1,NL
      YMNJK(J,K)=YMNJK(J,K)+Y(I,J,K,L)
      YMNIK(I,K)=YMNIK(I,K)+Y(I,J,K,L)
      YMNIL(I,L)=YMNIL(I,L)+Y(I,J,K,L)
      YMNJL(J,L)=YMNJL(J,L)+Y(I,J,K,L)
      YMNIJ(I,J)=YMNIJ(I,J)+Y(I,J,K,L)
      YMIJL(I,J,L)=YMIJL(I,J,L)+Y(I,J,K,L)
      YMIJK(I,J,K)=YMIJK(I,J,K)+Y(I,J,K,L)
   10 YFTMN(NIJK+L)=YFTMN(NIJK+L)+Y(I,J,K,L)
   20 YFTMN(NIJ+K)=YFTMN(NIJ+K)+YMIJK(I,J,K)
      YFTMN(NI+J)=YFTMN(NI+J)+YMNIJ(I,J)
   30 YFTMN(I)=YFTMN(I)+YMNIJ(I,J)
   40 YMEAN=YMEAN+YFTMN(I)
C     ****************************************************************
C     GRAND MEAN
C     ****************************************************************
      YMEAN=YMEAN/XDIJ/XDKL
C     ****************************************************************
C     COMPUTE FACTOR MEANS AND EFFECTS, THEN FACTOR SUMS OF SQUARES
C
C     FACTOR A LEVEL MEANS, EFFECTS, AND SS
C     ****************************************************************
      DO 50 I=1,NI
      YFTMN(I)=YFTMN(I)/XDJKL
      ALPHA(I)=YFTMN(I)-YMEAN
   50 SS(1 )=SS(1 )+ALPHA(I)*ALPHA(I)
C     ****************************************************************
C     FACTOR B LEVEL MEANS, EFFECTS, AND SS
C     ****************************************************************
      DO 55 J=1,NJ
      YFTMN(NI+J)=YFTMN(NI+J)/XDIKL
      ALPHA(NI+J)=YFTMN(NI+J)-YMEAN
   55 SS(4)=SS(4)+ALPHA(NI+J)*ALPHA(NI+J)
C     ****************************************************************
C     FACTOR C LEVEL MEANS, EFFECTS, AND SS
C     ****************************************************************
      DO 60 K=1,NK
      YFTMN(NIJ+K)=YFTMN(NIJ+K)/XDIJL
      ALPHA(NIJ+K)=YFTMN(NIJ+K)-YMEAN
   60 SS(7)=SS(7)+ALPHA(NIJ+K)*ALPHA(NIJ+K)
C     ****************************************************************
C     REPLICATION LEVEL MEANS AND SS
C     ****************************************************************
      DO 65 L=1,NL
      YFTMN(NIJK+L)=YFTMN(NIJK+L)/XDIJK
   65 SS(2 )=SS(2 )+(YFTMN(NIJK+L)-YMEAN)**2
C     ****************************************************************
C     FIRST ORDER INTERACTION LEVEL MEANS, EFFECTS AND SS
C     ****************************************************************
      DO 80 I=1,NI
C     ****************************************************************
C     FACTOR A X B INTERACTION LEVEL MEANS, EFFECTS, AND SS
```

```
C     ******************************************************************
      DO 70 J=1,NJ
      YMNIJ(I,J)=YMNIJ(I,J)/XDKL
      NDUM=NIJK+(I-1)*NJ+J
      ALPHA(NDUM)=YMNIJ(I,J)-YFTMN(I)-YFTMN(NI+J)+YMEAN
   70 SS(5)=SS(5)+ALPHA(NDUM)*ALPHA(NDUM)
C     ******************************************************************
C     FACTOR A X C INTERACTION LEVEL MEANS, EFFECTS, AND SS
C     ******************************************************************
      DO 75 K=1,NK
      YMNIK(I,K)=YMNIK(I,K)/XDJL
      NDUM=NIJK+NI*NJ+(I-1)*NK+K
      ALPHA(NDUM)=YMNIK(I,K)-YFTMN(I)-YFTMN(NIJ+K)+YMEAN
   75 SS(8)=SS(8)+ALPHA(NDUM)*ALPHA(NDUM)
C     ******************************************************************
C     FACTOR A X REPLICATION LEVEL MEANS FOR ERROR(A) SS
C     ******************************************************************
      DO 80 L=1,NL
      YMNIL(I,L)=YMNIL(I,L)/XDJK
   80 SS(3 )=SS(3 )+(YMNIL(I,L)-YFTMN(I)-YFTMN(NIJK+L)+YMEAN)**2
      DO 90 J=1,NJ
C     ******************************************************************
C     FACTOR B X C INTERACTION LEVEL MEANS, EFFECTS, AND SS
C     ******************************************************************
      DO 85 K=1,NK
      YMNJK(J,K)=YMNJK(J,K)/XDIL
      NDUM=NIJK+NI*NJ+NI*NK+(J-1)*NK+K
      ALPHA(NDUM)=YMNJK(J,K)-YFTMN(NI+J)-YFTMN(NIJ+K)+YMEAN
   85 SS(9)=SS(9)+ALPHA(NDUM)*ALPHA(NDUM)
C     ******************************************************************
C     FACTOR B X REPLICATION LEVEL MEANS, AND SS FOR ERROR(B) SS
C     ******************************************************************
      DO 90 L=1,NL
      YMNJL(J,L)=YMNJL(J,L)/XDIK
   90 SDUM6=SDUM6+(YMNJL(J,L)-YFTMN(NI+J)-YFTMN(NIJK+L)+YMEAN)**2
C     ******************************************************************
C     SECOND ORDER INTERACTION LEVEL MEANS, EFFECTS, AND SS
C     ******************************************************************
      DO 110 I=1,NI
      DO 110 J=1,NJ
C     ******************************************************************
C     FACTOR A X B X C INTERACTION LEVEL MEANS, EFFECTS, AND SS
C     ******************************************************************
      DO 100 K=1,NK
      YMIJK(I,J,K)=YMIJK(I,J,K)/XNL
      NDUM=NIJK+NI*NJ+NI*NK+NJ*NK+(I-1)*NJ*NK+(J-1)*NK+K
      ALPHA(NDUM)=YMIJK(I,J,K)-YMNIJ(I,J)-YMNIK(I,K)-YMNJK(J,K)+YFTMN(I)
     1+YFTMN(NI+J)+YFTMN(NIJ+K)-YMEAN
  100 SS(10)=SS(10)+ALPHA(NDUM)*ALPHA(NDUM)
C     ******************************************************************
C     FACTOR A X B X REPLICATION LEVEL MEANS AND SS FOR ERROR (B)
C     ******************************************************************
```

```
      DO 110 L=1,NL
      YMIJL(I,J,L)=YMIJL(I,J,L)/XNK
  110 SS(6)=SS(6)+(YMIJL(I,J,L)-YMNIJ(I,J)-YMNIL(I,L)-YMNJL(J,L)+YFTMN(I
     1)+YFTMN(NI+J)+YFTMN(NIJK+L)-YMEAN)**2
C     ******************************************************************
C     MULTIPLY SS BY CORRECT FACTOR TO OBTAIN SUMS OF SQUARES
C     ******************************************************************
      SS(10)=SS(10)*XNL
      SS(9)=SS(9)*XDJL
      SS(8)=SS(8)*XDJL
      SS(7)=SS(7)*XDIJL
      SS(6)=SS(6)*XNK+SDUM6*XDIK
      SS(5)=SS(5)*XDKL
      SS(4)=SS(4)*XDIKL
      SS(3 )=SS(3 )*XDJK
      SS(2 )=SS(2 )*XDIJK
      SS(1 )=SS(1 )*XDJKL
C     ******************************************************************
C     COMPUTE TOTAL SUM OF SQUARES, RESIDUALS, AND THE RESIDUAL SS
C     ******************************************************************
      DO 140 I=1,NI
      DO 140 J=1,NJ
      DO 140 K=1,NK
      DO 140 L=1,NL
      SS(12)=SS(12)+(Y(I,J,K,L)-YMEAN)**2
      CALL OVERFL (IERR)
      IF (IERR.EQ.1) GO TO 99
C     ******************************************************************
C     RESIDUALS
C     ******************************************************************
      NDUM= NIJK+I*NJ-NJ+J
      NDUM1=NIJK+NI*NJ+I*NK-NK+K
      R(I,J,K,L)=Y(I,J,K,L)-YMEAN-ALPHA(I)-ALPHA(NI+J)-ALPHA(NIJ+K)-
     1ALPHA(NDUM)-ALPHA(NDUM1)
      NDUM= NIJK+NI*NJ+NI*NK+J*NK-NK+K
      NDUM1=NIJK+NI*NJ+NI*NK+NJ*NK+I*NJ*NK-NJ*NK+J*NK-NK+K
      R(I,J,K,L)=R(I,J,K,L)-ALPHA(NDUM)-ALPHA(NDUM1)
C     ******************************************************************
C     RESIDUAL SS
C     ******************************************************************
  140 SS(11)=SS(11)+R(I,J,K,L)*R(I,J,K,L)
C     ******************************************************************
C     ERROR(C) SS IS RESIDUAL SS LESS E(B) SS LESS E(A) SS LESS REP SS
C     ******************************************************************
      SS(11)=SS(11)-SS(6)-SS(3)-SS(2)
C     ******************************************************************
C     SET UP DEGREES OF FREEDOM
C     ******************************************************************
      IDF(12)=NI*NJ*NK*NL-1
      IDF(11)=NI*NJ*(NK-1)*(NL-1)
      IDF(10)=(NI-1)*(NJ-1)*(NK-1)
      IDF(9)=(NJ-1)*(NK-1)
```

```
      IDF(8)=(NI-1)*(NK-1)
      IDF(7)=NK-1
      IDF(6)=NI*(NJ-1)*(NL-1)
      IDF(5)=(NI-1)*(NJ-1)
      IDF(4)=NJ-1
      IDF(3 )=(NI-1)*(NL-1)
      IDF(2 )=NL-1
      IDF(1 )=NI-1
C     *****************************************************************
C     SET UP MEAN SQUARES
C     *****************************************************************
      DO 150 I=1,11
  150 YMNSQ(I)=SS(I)/IDF(I)
C     *****************************************************************
C     COMPUTE F-STATISTICS AND PROBABILITIES F-STATS EXCEEDED
C     ERROR(C) GROUP
C     *****************************************************************
      DO 160 I=7,10
      F(I)=YMNSQ(I)/YMNSQ(11)
      CALL DVCHK  (IERR)
      IF (IERR.EQ.1) GO TO 99
  160 PF(I)=1.-FISH(F(I),IDF(I),IDF(11))
C     *****************************************************************
C     ERROR(B) GROUP
C     *****************************************************************
      DO 170 I=4,5
      F(I)=YMNSQ(I)/YMNSQ(6)
      CALL DVCHK  (IERR)
      IF (IERR.EQ.1) GO TO 99
  170 PF(I)=1.-FISH(F(I),IDF(I),IDF(6))
C     *****************************************************************
C     ERROR(A) FOR FACTOR A
C     *****************************************************************
      F(1)=YMNSQ(1)/YMNSQ(3)
      CALL DVCHK  (IERR)
      IF (IERR.EQ.1) GO TO 99
      PF(1)=1.-FISH(F(1),IDF(1),IDF(3))
      RETURN
   99 RETURN 1
      END
```

APPENDIX D

Data Analysis Code

The main driver of this code reads the data, initializes arrays, and calls the outlier (OUTLR) and curvilinear regression (CVREG) subroutines. Subroutine SORT is utilized in locating the set of replications having the largest sample variance in the OUTLR subroutine. This criterion for rejection of outlying observations is explained in detail in the Outlier Detection section of this report. This code and the code contained in Appendix E assume that the input data is sorted ascending, by temperature, then by thickness, then by stress-level, and finally by replication.

```
C*******************************************************************************
C
C                         DATA ANALYSIS PROGRAM
C
C*******************************************************************************
C
C
C      MAIN DRIVER READS THE CUSHION MATERIAL DATA RECORD CONSISTING OF
C      TEMPERATURE, DROP HEIGHT, THICKNESS, STRESS-LEVEL, G-VALUE,
C         REPLICATION AND MATERIAL TYPE     (IN THIS ORDER).
C      INITIALIZES ARRAYS CALL OUTLR (OUTLIER SUBPROGRAM), AND CALLS
C      CVREG (CURVILINEAR REGRESSION SUBPROGRAM).
C
C      THE OUTPUT IS IN THE FORM OF:
C      FOR A PARTICULAR
C         1.    DROP HEIGHT,TEMPERATURE,AND MATERIAL THICKNESS A
C               TABLE CONTAINING THE STRESS LEVELS AND G-VALUES,
C               AND ANY POINT THAT WAS REJECTED BY OUTLR IS ALSO LISTED.
C
C         2.    THE F - STATISTIC AND THE FIRST, SECOND, THIRD AND
C               FOURTH DEGREE POLYNOMIAL COEFFICIENTS ARE LISTED.
C
C
C*******************************************************************************
       DIMENSION STR(75), G(75), ALPH(3), X(75)
       DIMENSION COEF(10), YI(75), SIGHAT(4)
       DIMENSION XNEW(75), YNEW(75)
       DIMENSION BK(5,4)
       REWIND 12
       KNT=0
       IFLAG=0
       KFLAG=0
       GO TO 200
100    KNT=1
       GO TO 300
200    READ( 5,1700,END=400) A1,A2,A3,A4,A5,A6,(ALPH(I),I=1,3)
       IF(KNT.EQ.0)DTJ=A3
       IF (A3.NE. DTJ ) GO TO 500
       KNT=KNT+1
300    TEMP=A1
       DHT=A2
       THCK=A3
       DTJ = A3
       STR(KNT)=A4
       G(KNT)=A5
       REP=A6
       GO TO 200
400    IFLAG=1
500    NPTS=KNT
       DO 600 I=1,NPTS
       X(I) = ALOG(100.*STR(I))
600    CONTINUE
```

```
      WRITE (6,1800) ALPH,DHT,TEMP,THCK
      WRITE (6,1900)
      I=1
      ITEMP=NPTS
700   IF (I.GT.ITEMP) GO TO 1000
      IF (STR(I).EQ.STR(I+1).AND.STR(I+1).EQ.STR(I+2)) GO TO 900
      IF (STR(I).EQ.STR(I+1)) GO TO 800
      WRITE (6,2100) STR(I),X(I),G(I)
      KFLAG=1
      I=I+1
      GO TO 700
800   WRITE (6,2200) STR(I),STR(I+1),X(I),X(I+1),G(I),G(I+1)
      KFLAG=1
      I=I+2
      GO TO 700
900   WRITE (6,2000) STR(I),STR(I+1),STR(I+2),X(I),X(I+1),X(I+2),G(I),G(
     1I+1),G(I+2)
      I=I+3
      GO TO 700
1000  CONTINUE
C***********************************************************************
      CALL OUTLR (NPTS,X,G,NEWPTS,XNEW,YNEW)
C***********************************************************************
      DO 1070 IK = 1,NEWPTS
      XNEW(IK)=X(IK)
      YNEW(IK) = G(IK)
1070  CONTINUE
      NI=0
1100  NI=NI+1
C***********************************************************************
      CALL CVREG (NI,NEWPTS,XNEW,YNEW,DEGRE,COEF)
C***********************************************************************
      SIGHAT(NI)=0.
C     POLYNOMIAL CALCULATION
      DO 1200 I=1,NEWPTS
      YI(I)=COEF(1)+COEF(2)*XNEW(I)
      IF(NI.EQ.2)YI(I)=YI(I)+COEF(3)*XNEW(I)**2
      IF(NI.EQ.3)YI(I)=YI(I)+COEF(4)*XNEW(I)**3
      IF(NI.EQ.4)YI(I)=YI(I)+COEF(5)*XNEW(I)**4
      SIGHAT(NI)=SIGHAT(NI)+(YNEW(I)-YI(I))**2
1200  CONTINUE
      DO 1150 K=1,5
1150  BK(K,NI) = COEF(K)
      DF=NEWPTS-NI-1
      SIGHAT(NI)=SIGHAT(NI)/DF
      IF (NI.GT.1) GO TO 1300
      GO TO 1100
1300  F=ABS((SIGHAT(NI-1)-SIGHAT(NI))/SIGHAT(NI))
      IF(NI.EQ.4)F=ABS((SIGHAT(2)-SIGHAT(4))/SIGHAT(4))
      NII=NI-1
      WRITE (6,1600) F,NII,SIGHAT(NII),NI,SIGHAT(NI)
      IF (NI.LT.4) GO TO 1100
```

```
      WRITE(12) NEWPTS
      WRITE(12) DHT,TEMP,THCK,(ALPH(II),II=1,2),((BK(KK,JJ),KK=1,5),
     1      JJ=1,4),   (X(LL),LL=1,NEWPTS), (Y(LL),LL=1,NEWPTS)
1400  IF (IFLAG.EQ.1) GO TO 1500
      GO TO 100
1500  IPT = 99999
      WRITE(12) IPT
      STOP
C
1600  FORMAT (///2X,'F =',E15.8,5X,'SIG',I1,'SQ =',E15.8,5X,'SIG',I1,'SQ
     1 =',E15.8)
1700  FORMAT (3F10.0,F10.2,2F10.0,8X,3A4)
1800  FORMAT (1H1,3A4,' DROP HEIGHT OF',F5.1,2H'',', TEMPERATURE ',F6.1
     1,'F, THICKNESS',F4.1,2H'')
1900  FORMAT (1H ,8X,'STRESS LEVELS',21X,'LN(100*STRESS LEVELS)',25X,3HG
     1'S)
2000  FORMAT (1H ,3(3F10.4,5X))
2100  FORMAT (1H ,3(F10.4,25X))
2200  FORMAT (1H ,3(2F10.4,15X))
      END
```

```
      SUBROUTINE OUTLR (NPTS,X,Y,NEWPTS,XNEW,YNEW)
C***********************************************************************
C
C                     OUTLIER POINT REJECTION SUBPROGRAM
C
C***********************************************************************
C     METHOD BASED ON THE NAIR CRITERION (AN EXTENSION OF THE
C     EXTREME STUDENTIZED DEVIATE FROM THE SAMPLE MEAN).
C
C     FOR THE SET OF OBSERVATIONS HAVING THE LARGEST VARIANCE
C     EACH OBSERVATION OF THE SET IS TESTED INDIVIDUALLY AS A
C     CANDIDATE FOR REJECTION AS AN OUTLIER.
C
C***********************************************************************
      DIMENSION X(1), Y(1), XNEW(1), YNEW(1), XA(25), XB(25), XC(25), YA
     1(25), YB(25), YC(25), YMEAN(25), S2N(25), KEY(25)
      TESTV = 1.66
      DO 100 I=1,25
100   KEY(I)=0
      NTEMP=NPTS-2
      L=1
      I=1
200   CONTINUE
      J=I+1
      K=I+2
      IF (I.GT.NTEMP) GO TO 500
      IF (X(J).NE.X(K)) GO TO 400
      IF (X(I).NE.X(J)) GO TO 300
      XA(L)=X(I)
      XB(L)=X(J)
      XC(L)=X(K)
      YA(L)=Y(I)
      YB(L)=Y(J)
      YC(L)=Y(K)
      KEY(L)=0
      L=L+1
      I=I+3
      GO TO 200
300   CONTINUE
      I=I+1
      GO TO 200
400   CONTINUE
      I=I+2
      GO TO 200
500   CONTINUE
      L=L-1
      NP=L
      NPTS=(L*3)-1
C     CALCULATE MEANS & VARIANCES
      DO 700 I=1,NP
      IF (YA(I).NE.YB(I).OR.YA(I).NE.YC(I)) GO TO 600
      YMEAN(I)=YA(I)
```

```
      S2N(I)=0.0
      WRITE (6,2300) YMEAN(I),S2N(I)
      GO TO 700
600   CONTINUE
      YMEAN(I)=(YA(I)+YB(I)+YC(I))/3.
      S2N(I)=((YA(I)-YMEAN(I))**2+(YB(I)-YMEAN(I))**2+(YC(I)-YMEAN(I))**
     12)/2.
700   CONTINUE
C***********************************************************************
C     SORT IN ORDER OF DECREASING VARIANCE
      CALL SORT (NP,S2N,YMEAN,YA,YB,YC,XA,XB,XC)
C***********************************************************************
C     CALCULATE SNU
      L=1
C     SUM VARIANCES
800   KNT=0
      SUMV=0.
      DO 900 I=1,NP
      IF (KEY(I).EQ.1) GO TO 900
      IF (I.EQ.L) GO TO 900
      SUMV=SUMV+S2N(I)
      KNT=KNT+1
900   CONTINUE
      SNU=SQRT(SUMV/(KNT-1))
C     TEST
      TA=ABS((YA(L)-YMEAN(L))/SNU)
      TB=ABS((YB(L)-YMEAN(L))/SNU)
      TC=ABS((YC(L)-YMEAN(L))/SNU)
      IF (TA.EQ.TB.OR.TA.EQ.TC.OR.TB.EQ.TC) GO TO 1300
      TE=MAX(TA,TB,TC)
      IF (TE.LE.TESTV) GO TO 1300
      IF (TE.EQ.TA) GO TO 1000
      IF (TE.EQ.TB) GO TO 1100
      IF (TE.EQ.TC) GO TO 1200
      GO TO 1300
1000  WRITE (6,2400) L,XA(L),YA(L),TE,SNU
      XA(L)=999.
      KEY(L)=1
      GO TO 1300
1100  WRITE (6,2400) L,XB(L),YB(L),TE,SNU
      XB(L)=999.
      KEY(L)=1
      GO TO 1300
1200  WRITE (6,2400) L,XC(L),YC(L),TE,SNU
      XC(L)=999.
      KEY(L)=1
C     CHECK TO SEE IF ALL VALUES
1300  IF (L.EQ.NP) GO TO 1400
      L=L+1
      GO TO 800
C     PUT IN NEW ARRAYS
1400  L=0
```

```
      NEWPTS=0
1500  L=L+1
      IF (XA(L)-999.) 1600,1700,1600
C     POINT ACCEPTED
1600  NEWPTS=NEWPTS+1
      XNEW(NEWPTS)=XA(L)
      YNEW(NEWPTS)=YA(L)
1700  IF (XB(L)-999.) 1800,1900,1800
1800  NEWPTS=NEWPTS+1
      XNEW(NEWPTS)=XB(L)
      YNEW(NEWPTS)=YB(L)
1900  IF (XC(L)-999.) 2000,2100,2000
2000  NEWPTS=NEWPTS+1
      XNEW(NEWPTS)=XC(L)
      YNEW(NEWPTS)=YC(L)
2100  IF (L.GE.NP) GO TO 2200
      GO TO 1500
2200  RETURN
C
2300  FORMAT (1H0,12X,'SAME VALUES    ',2F6.0)
2400  FORMAT ('0REJECT POINT',I3,', X =',F6.2,', Y =',F6.2,',T =',F6.2,'
     1,SNU =',F6.2)
      END
```

```
      SUBROUTINE CVREG (N1,NPT,X,Y,DEGRE,COEF)
C***********************************************************************
C
C     CURVILINEAR REGRESSION
C     CURVILINEAR REGRESSION DETERMINES THE EQUATIONS
C     OF THE STATISTICALLY BEST FITTING POLYNOMIALS OF FIRST,
C     SECOND, THIRD AND FOURTH ORDER.
C
C***********************************************************************
      DIMENSION A(10,11), X(1), Y(1), COEF(1), KON(10)
      N=N1+1
      M=N+1
      ANPT=NPT
      DO 100 I=1,N
      DO 100 J=1,M
100   A(I,J)=0.
      A(1,1)=NPT
      DO 400 K=1,NPT
      DO 300 I=1,N
      DO 300 J=1,N
      IPJ2=I+J-2
      IF (IPJ2) 300,300,200
200   A(I,J)=A(I,J)+X(K)**IPJ2
300   CONTINUE
      DO 400 I=2,N
400   A(I,M)=A(I,M)+X(K)**(I-1)*Y(K)
      DO 500 J=1,NPT
500   A(1,M)=A(1,M)+Y(J)
      DO 600 J=1,M
      DO 600 I=1,N
600   A(I,J)=A(I,J)/ANPT
      IERR=0
      M=N+1
      DO 1300 I=1,N
      IF (A(I,I)) 800,700,800
700   IERR=1
      GO TO 1400
800   TEMP=1.0/A(I,I)
      IP1=I+1
      DO 900 J=IP1,M
900   A(I,J)=A(I,J)*TEMP
      DO 1200 K=1,N
      IF (I-K) 1000,1200,1000
1000  DO 1100 J=IP1,M
1100  A(K,J)=A(K,J)-A(K,I)*A(I,J)
1200  CONTINUE
1300  CONTINUE
      N=N1+1
      M=N+1
1400  IF (IERR) 1500,1600,1500
1500  WRITE (6,3100)
      GO TO 2300
```

```
1600  CONTINUE
      DO 1700 K=1,N
1700  COEF(K)=A(K,M)
      SUMR2=0.0
      DO 1900 I=1,NPT
      YC=COEF(1)
      DO 1800 K=2,N
1800  YC=YC+COEF(K)*X(I)**(K-1)
      R=Y(I)-YC
1900  SUMR2=SUMR2+R*R
      SIGMA=SQRT(SUMR2/ANPT)
      SSERR=SUMR2
      SUMR2=Y(1)
      DO 2000 I=2,NPT
2000  SUMR2=SUMR2+Y(I)
      BARY1=SUMR2/NPT
      SUMR2=0.0
      DO 2100 I=1,NPT
      R=Y(I)-BARY1
2100  SUMR2=SUMR2+R*R
      SSTOT=SUMR2
      SSREG=SSTOT-SSERR
      DSREG=SSREG/N1
      DSERR=SSERR/(NPT-(N1+1))
      FRATO=DSREG/DSERR
      DEGFT=N1
      DEGFB=NPT-(N1+1)
      DEGRE=NPT-1
      ETS=SSERR/SSTOT
      IF(ETS.GE.1.0)ETS=1.0
      CORR=SQRT(1.0-ETS)
      WRITE (6,3000) N1
      WRITE (6,2600)
      WRITE (6,2700) SSREG,DEGFT,DSREG,FRATO,CORR
      WRITE (6,2800) SSERR,DEGFB,DSERR
      WRITE (6,2900) SSTOT,DEGRE
      WRITE (6,2400)
      MM=N1+1
      DO 2200 I=1,10
2200  KON(I)=I-1
      WRITE (6,2500) (KON(I),COEF(I),I=1,MM)
2300  RETURN
C
2400  FORMAT (//,2X,'CURVE COEFFICIENTS')
2500  FORMAT (//2X,2HB(,I1,1H),3X,E15.7)
2600  FORMAT(//,2X,'SOURCE',9X,'S.S',9X,'D.F',9X,'M.S',9X,'F',12X,'R')
2700  FORMAT (//,2X,'DUE TO',5(4X,E10.4))
2800  FORMAT (//,2X,'ABOUT',3(4X,E10.4))
2900  FORMAT (//,2X,'TOTAL',2(4X,E10.4))
3000  FORMAT (1H1,2X,'ANOVA FOR CURVE OF ORDER',I3)
3100  FORMAT (//,4X,'SINGULAR MATRIX ',/,4X,'CURVE FIT IMPOSSIBLE')
      END
```

```
      SUBROUTINE SORT(N,VAL,X1,X2,X3,X4,X5,X6,X7)
C*****************************************************************************
C
C     SUBROUTINE FOR SORTING N NUMBERS IN DESCENDING ORDER
C
C*****************************************************************************
      DIMENSION VAL(1),X1(1),X2(1),X3(1),X4(1),X5(1),X6(1),X7(1)
      M=N-1
      DO 100 I=1,M
      L=I+1
      DO 100 II=L,N
      IF(VAL(I) .GE. VAL(II)) GOTO 100
      F = VAL(I)
      VAL(I) = VAL(II)
      VAL(II) = F
      F=X1(I)
      X1(I)=X1(II)
      X1(II)=F
      F=X2(I)
      X2(I)=X2(II)
      X2(II)=F
      F=X3(I)
      X3(I)=X3(II)
      X3(II)=F
      F=X4(I)
      X4(I)=X4(II)
      X4(II)=F
      F=X5(I)
      X5(I)=X5(II)
      X5(II)=F
      F=X6(I)
      X6(I)=X6(II)
      X6(II)=F
      F=X7(I)
      X7(I)=X7(II)
      X7(II)=F
100   CONTINUE
      RETURN
      END
```

APPENDIX E
Cushioning Model Development Code

The Stepwise Regression code generates the coefficients for the terms in the general cushion model. It uses the data as generated by the Data Analysis program.

```
C        ******************************************************************
C
C   THIS PROGRAM IS USED TO GENERATE THE DYNAMIC CUSHIONING MODEL.
C           *** THE PLOT ROUTINES ARE STANDARD CDC 6600 SCOPE ROUTINES ***
C
C   PASS 1 IS USED TO GENERATE DATA FOR THE MLR PROGRAM.
C
C   PASS 2 IS USED TO EVALUATE THE COEFFICIENTS FROM THE MLR PROGRAM
C           AND PLOT THE FINAL RESULTS.
C
C        ******************************************************************
C
      PROGRAM MODEL (TAPE1=130,TAPE2=65,TAPE11,
     *    TAPE98=65, TAPE99=130,
     *    PUNCH=65, INPUT=65, OUTPUT=130, TAPE7=PUNCH,TAPE5=INPUT)
C
      DIMENSION  TEMP(10), ATEMP(10),DROP(10), DL(10)
      DIMENSION  HEAD(8), LEFT(8), BOTTOM(8)
      DIMENSION  RIGHT(15)
      DIMENSION  IND(51),COEFF(51),A(855),X(101),Y(101)
      DIMENSION  NC(28), NCARD(8), V(51)
      DIMENSION  THICK(10), NSYM(10)
      DATA  LEFT/8*1H /
C
      DATA HEAD/8*1H /
      DATA  TEMP/10*0./, DROP/10*0./ , ATEMP, DL / 20*10H********** /
      DATA  THICK/10*0./, NSYM/10*10H**********/
      DATA HEAD/8*1H /, RIGHT/15*1H /
      DATA  TP,DH,TC,SS,GL/5*1./
      CALL DATE( DAY )
       NVR = 45
C INSERT NUMBER  OF EQUATION TERMS HERE ................................
C
      REWIND 7
      READ(7,904)  NTEST
      IF( EOF(7) .EQ. 0.0 )  GO TO 200
C
C************ PASS 1 ***************************************************
    9 NBRT = 1
C        COPY TAPE 99 TO  FILE 1
      NBRD = 1
      NBRC = 1
      NP = 0
      REWIND 1
      REWIND 99
   10 READ( 5,905) NCARD,   TP, DH, TC, SS, GL,  REP, TP1,TP2
  905 FORMAT(8A10, T1,6F10.5,8X,,A10,A2)
      IF( EOF( 5  ) .NE. 0.0 )  GO TO 14
C     IF(DH .EQ. 21) GO TO 13
C     IF(TC .GE. 4.)GO TO 13
C
C
```

```
      ENCODE(50,9000,NCARD(1))TP,DH,TC,SS,GL
 9000 FORMAT(5F10.5)
      PRINT 9001,NCARD
 9001 FORMAT(*  TEST *,8A10)
      WRITE(1,920)  NCARD
      NP = NP + 1
C          COUNT DIFFERENT TEMPERATURES AND CONSTRUCT LABELS
      DO 15 J=1,NBRT
      IF( TP .EQ. TEMP(J) )  GO TO 16
   15 CONTINUE
      TEMP(NBRT) = TP
      JTP = TP
      ENCODE(10,912,ATEMP(NBRT))    JTP
  912 FORMAT(I3,* DEGREE*)
      NBRT = NBRT + 1
C          COUNT DIFFERENT DROP HEIGHTS AND CONSTRUCT LABELS
   16 DO 17 J=1,NBRD
      IF( DH .EQ. DROP(J) )  GO TO 18
   17 CONTINUE
      DROP(NBRD) = DH
      JDH = DH
      ENCODE(10,914,DL(NBRD))   JDH
  914 FORMAT(I3,* INCHES*)
      NBRD = NBRD + 1
C
C          COUNT DIFFERENT THICKNESSES
   18 DO 19 J=1,NBRC
      IF( TC .EQ. THICK(J) )  GO TO 20
   19 CONTINUE
      THICK(NBRC) = TC
      JTC = TC
      ENCODE(10,915,NSYM(NBRC))  JTC
  915 FORMAT(I1,9X)
      NBRC = NBRC + 1
C
   20 CONTINUE
      GO TO 13
   14 END FILE 1
      REWIND 1
      REWIND 99
      NBRT = NBRT - 1
      NBRD = NBRD - 1
      NBRC = NBRC - 1
      IF( NP*NBRT*NBRD*NBRC .NE. 0 )  GO TO 21
      PRINT 916
  916 FORMAT(*1WRONG INPUT FILE*////)
C        MODE 1
      N = -LOCF(TEMP(1)) - 10000
      TEMP(N) = 0.
   21 CONTINUE
      IF( NBRD .GT. 5 )  NBRD = 5
      WRITE(98,930) TP1,TP2,NBRT,NBRD,NBRC,TEMP,ATEMP,DROP,DL,THICK,NSYM
```

```
  930 FORMAT(2A10,3I2/3(10F10.4/10A10/))
      REWIND 98
C
      NRETURN = 1
      NC(3)=NC(4)=NC(6)=NC(12)=NC(17)=NC(18)=NC(19)=NC(28) = 1
      NC(5)=NP
      NC(7)=6
      NC(9)=999
      NC(12) = 1
      PRINT 900,   DAY
  900 FORMAT(*1DYNAMIC CUSHIONING ANALYSIS*90X,A10)
      DO  150  NT = 1,1
      REWIND 1
      NP = 0
      NC(1)=NC(26)=NT
      NC(2)=NC(13)=NC(14)=    NVR
      TT = TEMP(NT)
C
  100 READ(1,901)  TP, DH, TC, SS, GL
  901 FORMAT(5F10.5)
      IF( EOF(1) .NE. 0.0 )  GO TO 120
C
C INSERT DATA CONDITIONS HERE... ALSO IN AFFECTED DO LOOPS IN PASS 2 ....
C
      YVAR = GL
C
C          COMPUTE DYNAMIC CUSHIONING MODEL VARIABLES
      GO TO 800
  105 CONTINUE
C
C INSERT YVAR TRANSFORMATION HERE .......................................
      WRITE(11)  ( V(J), J=1, NVR ),  YVAR
      NP = NP + 1
      GO TO 100
C
  120 END FILE 11
      PRINT 902,NP
  902 FORMAT(//*  NP = *I5)
      WRITE(2,904)  NC
      PRINT   904,  NC
  904 FORMAT( 14I5 )
      WRITE(2,906) TP1,TP2
      PRINT   906, TP1,TP2
  906 FORMAT(80X,T2,*DYNAMIC CUSHIONING MODEL* 5X2A10)
  150 CONTINUE
      PRINT 907, ( TEMP(J), J=1,NBRT )
      PRINT 908, ( DROP(J), J=1,NBRD )
      PRINT 909, ( THICK(J),J=1,NBRC )
  907 FORMAT(///* TEMPERATURES* 11F6.0)
  908 FORMAT(  /* DROP HEIGHTS* 11F6.0)
  909 FORMAT(  /* THICKNESSES * 11F6.0)
      REWIND 11
```

```
      REWIND 2
      GO TO 999
C
C************* PASS 2 *********************************************************
C
  200 REWIND 7
C           READE CONSTANTS FOR PASS 2
      REWIND 98
      READ (98,950) TP1,TP2,NBRT,NBRD,NBRC,TEMP,ATEMP,DROP,DL,THICK,NSYM
      IF( NBRT .EQ. 0 )  STOP 'ERROR'
      NRETURN = 2
      HEAD(1) = TP1
      HEAD(2) = TP2
      XMIN = ALOG10( .03 )
      XMAX = ALOG10( 10.0)
      DX = ( XMAX - XMIN ) / 100.
  210 READ(7,920)   NCARD,   K, CONST, NT, NV,ND, NSTEP, ND, SE, R
  920 FORMAT(8A10,T1, I5 ,E20.8,5I5,F20.4,F10.7)
      IF( EOF(7) .NE. 0.0 )  GO TO 999
C
      PRINT 922,               NV, R
  922 FORMAT(*1* / *1*         I5* VARIABLES     CORRELATION COEFFICIENT IS*
     *F8.6///* MLR OUTPUT CARDS*//)
      PRINT 910, NCARD
  910 FORMAT(1X,8A10)
      DO 220 J=1,NV
      READ(7,920)   NCARD, IND(J), COEFF(J)
  220 PRINT 910, NCARD
       ENCODE(150,911,RIGHT)  ( IND(J), J=1,NV )
  911 FORMAT(100X,50X,T1, 50I3)
C
      DO 260 NT = 1,NBRT
      HEAD(6) = ATEMP(NT)
      TP = TEMP(NT)
C           DROP HEIGHT
      DO 250 ND = 1,NBRD
      DH = DROP(ND)
C**
      HEAD(8) = DL(ND)
      CALL SETGRID( A, -76, XMIN, XMAX, 0.0,350.)
      CALL LABGRID( A, 1, 20, 20HLOG( STATIC STRESS ) )
      CALL LABGRID( A, 2, 20, 20H                  G LEVEL )
      CALL LABGRID( A, 3, 80, HEAD )
      CALL LABGRID( A, 4, 76, RIGHT )
C
C           THICKNESS
      DO 240 NTC =1,NBRC
      TC = THICK(NTC)

      XX = XMIN
      DO 230 JP=1,101
      SS=10**XX
```

```
C         COMPUTE DYNAMIC CUSHIONING MODEL VARIABLES
      GO TO 600
  222 CONTINUE
      Y(JP) = CONST
      DO 225 J=1,NV
      I = IND(J)
  225 Y(JP) = Y(JP) + COEFF(J) * V(I)
      X(JP) = XX
C
C INSERT REVERSE YVAR TRANSFORMATION HERE.    Y(JP) = ----  ..............
C
  230 XX = XX + DX
C
      CALL PLTGRID( A, NSYM(NTC), 101, X, Y )
  240 CONTINUE
C
      CALL PRINTPL( A, 6LOUTPUT )
  250 CONTINUE
  260 CONTINUE
      GO TO 210
C
  600 CONTINUE
C********* DYNAMIC CUSHIONING MODEL ***************************************

      SS100 = SS * 100.
      AL = ALOG( SS100 )
      AL2 = AL * AL
      SRDH = SQRT( DH )
      TC5H = TC ** ( -3.5)
       TR = (TP+460)/100.
      TR2 = TR * TR
      TR3 = TR * TR2
      TR4 = TR3 * TR
C
      TCOH = TC**(-0.5)
      TCTH = TC**(-1.5)
      TCINV = TC**(-2.5)
      V(01) = TR * TCOH * 1.0
      V(02) = TR * TCOH * 1.0 * AL
      V(03) = TR * TCOH * 1.0 * AL2
      V(04) = TR * TCTH * SRDH
      V(05) = TR * TCTH * SRDH * AL
      V(06) = TR * TCTH * SRDH * AL2
      V(07) = TR * TCOH * SRDH
      V(08) = TR * TCOH * SRDH * AL
      V(09) = TR * TCOH * SRDH * AL2
      V(10) = TR2 * TCOH * 1.0
      V(11) = TR2 * TCOH * 1.0 * AL
      V(12) = TR2 * TCOH * 1.0 * AL2
      V(13) = TR2 * TCTH * SRDH
      V(14) = TR2 * TCTH * SRDH * AL
      V(15) = TR2 * TCTH * SRDH * AL2
```

```
      V(16) = TR2 * TCOH * SRDH
      V(17) = TR2 * TCOH * SRDH * AL
      V(18) = TR2 * TCTH * SRDH * AL2
      V(19) = TR3 * TCOH * 1.0
      V(20) = TR3 * TCOH * 1.0 * AL
      V(21) = TR3 * TCOH * 1.0 * AL2
      V(22) = TR3 * TCTH * SRDH
      V(23) = TR3 * TCTH * SRDH * AL
      V(24) = TR3 * TCTH * SRDH * AL2
      V(25) = TR3 * TCOH * SRDH
      V(26) = TR3 * TCOH * SRDH * AL
      V(27) = TR3 * TCTH * SRDH * AL2
      V(28) = TR * TCTH
      V(29) = TR * TCTH * AL
      V(30) = TR * TCTH * AL2
      V(31) = TR2 * TCTH
      V(32) = TR2 * TCTH * AL
      V(33) = TR2 * TCTH *AL2
      V(34) = TR3 * TCTH
      V(35) = TR3 * TCTH * AL
      V(36) = TR3 * TCTH * AL2
      V(37) = TR
      V(38) = TR * AL
      V(39) = TR * AL2
      V(40) = TR2
      V(41) = TR2 * AL
      V(42) = TR2 * AL2
      V(43) = TR3
      V(44) = TR3 * AL
      V(45) = TR3 * AL2
C
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      GO TO ( 105 , 222 ), NRETURN
C
  999 CONTINUE
      END
```

```
C      ******************************************************************
C
C      STEPWISE MULTIPLE LINEAR REGRESSION
C
C      ******************************************************************
C
C          BASED UPON PROCEDURES IN DRAPER'S APPLIED REGRESSION ANALYSIS
C          AND SHARE NUMBER 1333
C          TAPES  9 (A6) AND 10 (B6)  ARE USED AS WORK TAPES.
C          TAPES 11 (B5) AND 10 (B6)  ARE USED AS BINARY INPUT TAPES.
C A. MLR CONTROL CARD 1  FORMAT( 14I5 ) ********************************
C 01-05 NPROB  = NUMBER TO IDENTIFY PROBLEM.
C 06-10 NXV    = TOTAL NUMBER OF INDEPENDENT VARIABLES IN INPUT DATA.
C 11-15 NYV    = TOTAL NUMBER OF   DEPENDENT VARIABLES IN INPUT DATA.
C 16-20 INDEXY = INDEX OF THE DEPENDENT VARIABLE FOR THE PROBLEM.
C 21-25 NDATA  = TOTAL NUMBER OF DATA OBSERVATIONS FOR THE PROBLEM.
C                  IF UNKNOWN- SET EQUAL MAXIMUM EXPECTED AND SET LAST
C                  DATA OBSERVATION EQUAL TO  99999999.
C 26-30 IDEN   = NUMBER OF ALPHABETIC HEADER CARDS ( SEE C ).
C 31-35 INTYPE = 0 FOR REGULAR RUN WITH DATA ON CARDS.
C                1 TO REWIND 10 AND STORE CARD DATA FOR LATER PROBLEM.
C                2 TO REWIND 10 AND USE DATA STORED BY A PREVIOUS PROB.
C                3 TO STORE CARD DATA ON TAPE 10 WITHOUT REWIND.
C                4 TO USE DATA ON TAPE 10 WITHOUT FIRST REWINDING.
C                5 TO USE TAPE 11 (B5) AS INPUT AFTER REWINDING.
C                6 TO USE TAPE 11 (B5) AS INPUT WITHOUT REWINDING.
C                7 REWIND B5, USE AS INPUT, THEN REWIND FOR LATER USE.
C 36-40 NREAR  = 0 TO USE DATA WITHOUT REARRANGING IT.
C                1 TO REARRANGE DATA ACCORDING TO CONTROL CARD F.
C 41-45 MAXSTP = MAXIMUN NUMBER OF STEPS OR ITERATIONS ALLOWED.
C                  TO BYPASS PRINTOUT OF CALCULATIONS PRIOR TO SUMMARY,
C                  SET EQUAL TO 999.
C 46-50 IFBACK = STEP AT WHICH BACK SOLUTION STARTS (ACTUAL VS PRED.).
C                  SET EQUAL TO 0 FOR NO BACK SOLUTION.
C                  SET EQUAL TO 999 FOR BACK SOLUTION OF SUMMARY ONLY.
C                  NOTE - IF NDATA*(NXV+1) IS GREATER THAN 3000, A6 IS
C                  USED TO STORE DATA THEREBY INCREASING RUN TIME.
C 51-55 NSTART = NUMBER OF INDEPENDENT VARIABLES THAT YOU WISH TO START
C                  THE REGRESSION WITH (SEE D).    NORMAL VALUE IS 0.
C                  IF NSTART = -1 THE PROGRAM WILL AUTOMATICALLY PUT
C                  ALL NXV VARIABLES IN REGRESSION AT START WITH A
C                  TEST OF ONE, WITHOUT CONTROL CARDS IN D. TEST IS ZERO
C                  FOR OTHER NEGATIVE VALUES.
C 56-60 MINSUM = MIN NBR OF IND VAR IN SUMMARY O/P. NORMAL VALUE IS 1.
C 61-65 MAXSUM = MAX NBR OF IND VAR IN SUMMARY O/P. NORMAL VALUE IS NXV.
C 66-70 MAXREG = MAX NBR OF IND VAR IN REGRESSION.  NORMAL VALUE IS NXV.
C*********
C    MLR CONTROL CARD 2  ***************
C 01-05 IFWT   = 0 FOR UNWEIGHTED DATA
C              = 1 IF WEIGHTS ARE READ IN AS INPUT.
C 06-10 IFCNST = 0 IF CONSTANT TERM IS TO BE CALCULATED.
C              = 1 TO DELETE CONSTANT TERM
```

```
C                  =-1 IF CONST TERM IS TO BE CONSIDERED AS THE COEFFICIENT
C                      OF A NEW INDEPENDENT VARIABLE XD WHICH ALWAYS HAS THE
C                      VALUE 1. THE SIGNIFICANCE OF THE CONSTANT WILL BE
C                      INDICATED BY ITS STANDARD ERROR.
C 11-15 IFLIST = 0 TO LIST INPUT DATA, 1 OTHERWISE.
C 16-20 IFSUMS = 0 TO LIST SUM(XI*XJ), 1 OTHERWISE.
C 21-25 IFRES  = 0 TO LIST SUM(XI-XBARI)(XJ-XBARJ), 1 OTHERWISE.
C 26-30 IFCORR = 0 TO LIST SIMPLE CORRELATION COEFFICIENTS, 1 OTHERWISE.
C 31-35 NCROSS = 1 IF YOU WISH TO INCLUDE AS ADDITIONAL INDEPENDENT
C                      VARIABLES, THE SQUARES AND THE CROSS PRODUCTS OF
C                      THE INDEPENDENT VARIABLES.  ZERO OTHERWISE.
C                      GENERATED VARIBLES ARE X(NXV+1)=X1*X1
C                      X(NXV+2)=X1*X2, X(NXV+3)=X1*X3,...X(NXV+NXV)=X1*XNXV
C                      X(NXV+NXV+1)=X2*X2, X(NXV+NXV+2)=X2*X3 .... ETC.
C 36-40 IFTRA  = ALLOWS FOR TRANSFORMATIONS OF INPUT DATA. SEE G FOR USE
C              = 0 FOR NO TRANSFORMATIONS
C              = 1 FOR TRANSFORMATIONS.
C              =-1 TO USE PREVIOUS TRANS WHICH ARE STILL IN CORE.
C 41-45 NVOID  = 0 TO PROCESS ALL OBSERVATIONS. = 1 TO READ UP TO 14
C                    OBSERVATIONS TO BE LEFT OUT OF REGRESSION. SEE CONTROL
C                    CARD E. = -1 TO USE PREVIOUS E CARD .
C 46-50 IFSUB  = ZERO FOR NORMAL RUN. POSITIVE VALUE CALLS IN A
C                    USER SUPPLIED SUBROUTINE CALLED  NBROFX(IFSUB,NXV)
C                    TO CHANGE NXV (NOTE- NBROFX MUST BE SUPPLIED EVEN
C                    IF IT IS JUST A RETURN). A USER SUPPLIED SUBROUTINE
C                    CALLED  EQUAT(IFSUB,DATA) IS USED TO MAKE THE DESIRED
C                    CROSS PRODUCT AND TRANSFORMATIONS (NOTE- DEPENDENT
C                    VARIABLE SHOULD BE DEFINED AS THE VARIABLE DATA(NXV+1)
C                    WHERE DATA IS A SET OF OBSERVATIONS GOING INTO THE
C                    S/R AND THE TRANSFORMED SET COMMING OUT ).
C 51-55 NFMT   = 0 FOR REGULAR INPUT FORMAT(7F10.0).
C              = 1 TO READ INPUT FORMAT( SEE I ).
C              =-1 TO USE FORMAT FROM PREVIOUS RUN.
C 56-60 IFPNCH = 0 TO DELETE PUNCHING OF EQUATION COEFFICIENTS.
C 61-65 IFDATE = 0 TO PRINT DATE OF COMPUTER RUN
C 66-70 IFNAME = 0 TO READ NAMES OF VARIABLES. SEE H FOR FORMAT
C              =-1 TO USE PREVIOUS H CARD, STILL IN CORE.
C              = 1 TO ASSUME BLANK NAMES
C**********
C C. ALPHABETIC HEADER  CARDS.     DO NOT USE IF  IDEN=0 .
C    IDEN CARDS WITH FORMAT(16A5)       LAST CARD REPEATED ON EACH PAGE
C**********
C D. CARDS FOR VARIABLES IN REGRESSION AT START AND CORRESPONDING TESTS.
C    DO NOT USE IF NSTART=0. THERE SHOULD BE 'NSTART' FIELDS.7(F8.0,I2)
C 01-08 TEST(1) = A TEST CONDITION WHICH DETERMINES WHETHER A VARIABLE
C                      WILL BE DELEATED OR ADDED TO THE REGRESSION. ITS
C                      VALUE IS 1-R**2. ZERO CORRESPONDS TO A MULTIPLE CORR
C                      COEFFICIENT OF 1, WHICH MAKES IT IMPOSSIBLE FOR THE
C                      PROGRAM TO DELETE THAT VARIABLE FROM THE SET OF IND
C                      VARIABLES. TEST=1 CORRESPONDS TO MULT CORR COEF OF 0
C                      WHICH MAKES SUCH A DELETION CERTAIN.
C 09-10 INDEX(1)= FIRST  VARIABLE TO BE INCLUDED IN REGRESSION AT START.
```

```
C 11-18 TEST (2)= TEST FOR TWO VARIABLE SET.
C 19-20 INDEX(2)= SECOND VARIABLE TO BE INCLUDED IN REGRESSION AT START.
C 21-28 TEST (3)= TEST FOR THREE VARIABLE SET.
C 29-30 INDEX(3)= ETC.
C*********
C E. OBSERVATIONS TO BE REMOVED FROM REGRESSION. USE ONLY IF  NVOID=1 .
C 01-05 NOGOOD(1) = INDEX OF 1ST POINT TO BE REMOVED. ( FORMAT(14I5) )
C 06-10 NOGOOD(2) = ETC.
C*********
C F. CONTROL CARD TO REARRANGE MLR DATA.  DO NOT USE IF NREAR=0.  (14I5)
C 01-05 NWORDS = NUMBER OF WORDS IN TAPE OR CARD RECORD.
C 06-10 LOCY   = LOCATION OF DEPENDENT VARIABLE Y.
C 10-15 LOCX(J),J=1,NXV = LOCATIONS OF INDEPENDENT VARIABLES.
C                     IF  IFWT=0 LAST LOCATION IS FOR WEIGHTS.
C*********
C G. TRANSFORMATION CONTROLS. DO NOT USE IF IFTRA=0.  FORMAT(7(F8.0,I2))
C      PUT TRANSFORMATIONS AND CORRESPONDING CONSTANTS IN SAME ORDER AS
C      X AND Y VARIABLES.
C      TRANSFORMATIONS  0=NONE, 1=X+C, 2=X*C, 3=X/C, 4=C/X, 5=X**C,
C      6=C**X, 7=LN(X+C), 8=LOG(X+C), 9=E**(C*X), 10=E**(C/X),
C      11=SIN(C*X), 12=COS(C*X), 13=TAN(C*X)
C 01-08 CONST  = CONSTANT FOR FIRST VARIABLE
C 09-10 NBRTRA = TRANSFORMATION FOR FIRST VARIABLE
C 11-18        = CONST FOR SECOND VARIABLE ,  ETC.
C 19-20        = TRA FOR SECOND VARIABLE,     ETC.
C*********
C H. INPUT VARIABLE NAMES IN ORDER OF INPUT.  USE ONLY IF IFNAME=0.
C 01-10 = NAME OF FIRST INPUT VARIABLE
C 11-20 = NAME OF SECOND INPUT VARIABLE, ETC.     FORMAT( 7(A6,A4) )
C*********
C I. VARIABLE FORMAT FOR INPUT DATA (12A6). USE ONLY IF NFMT=1.
C*********
C J. MLR DATA CARDS SHOULD BE PUNCHED WITH FORMAT(7F10.0), OBSERVATION
C      BY OBSERVATION IN THE FOLLOWING ORDER (IF INTYPE=0,1) X1, X2, X3,
C      X4, ...., XNXV,  Y1, Y2, Y3, ...., YNYV,  WT(IF IFWT=1 ).
C     DATA CARDS FOR INTYPE=0,1,3 ONLY. BINARY TAPE INPUT FOR OTHER CODES
C      IF NREAR=1 ORDER OF DATA IS DETERMINED BY CONTROL CARD F.
C*********
C
C TO USE THE PROGRAM FOR AN ORDINARY MULTIPLE REGRESSION (I.E. NO
C  ADJOINING OR DELETING), PUT ALL VARIABLES IN THE REGRESSION AT
C  THE OUTSET ( NSTART = -NXV ) AND PUT MAXSTP = 1.
C***********************************************************************
C********* CARD OUTPUT  ( IF IFPNCH IS NOT EQUAL ZERO )
C
C           ONE CARD FOR EACH VARIABLE IN EQUATION
C           FORMAT( I5,E20.8,5I5,F20.4,F10.7 )
C       01-05 = I = INDEX OF INDEPENDENT VARIABLES IN EQUATION
C       06-25 = COEFF(I) = COEFFICIENT FOR VARIABLE I
C       26-30 = NPROB  = PROBLEM NUMBER
C       31-35 = NBRNOW = NUMBER OF VARIABLES IN EQUATION
C       36-40 = INDEXY = INDEX OF DEPENDENT VARIABLE
```

```
C      41-45 = NOSTEP = STEP NUMBER IN WHICH THE EQUATION WAS COMPUTED
C      46-50 = IFPNCH = INPUT VALUE GREATER THAN ZERO
C      51-70 = SIGPCT = STANDARD ERROR OF EQUATION AS A PERCENT OF Y MEAN
C      71-80 = REGRCO = CORRELATION COEFFICIENT OF EQUATION
C
C********* BASIC STATISTICS OUTPUT *********
C
C           XI = X = VALUE OF OBSERVATION FOR VARIABLE I
C           SUM( XI ) = SUMMATION OF VARIABLE I
C           N = NUMBER OF OBSERVATIONS
C           XN = WEIGHTED NUMBER OF OBSERVATIONS
C
C      MEAN = WEIGHTED AVERAGE = SUM( XI ) / XN
C      STANDARD DEVIATION = SQRT( (SUM(XI**2)-SUM(XI)*MEAN)/(XN-1) )
C
C      SUM OF VARIABLES  = SUM( XI )
C      RAW SUM OF SQUARES AND CROSS PRODUCTS = SUM( XI*XJ )
C      SUM OF SQUARES AND CROSS PRODUCTS ABOUT THE MEAN = CORRECTED SUMS=
C           = SS( I,J ) = SUM(XI*XJ)-SUM(XI)*SUM(XJ)/XN
C      SIMPLE CORRELATION COEFFICIENTS =
C           = R(I,J) = SS(I,J)/SQRT( SS(I,I)*SS(J,J) )
C
C********* RESIDUAL ANALYSIS ( ACTUAL VS PREDICTED ) PRINTOUT *********
C
C      ACTUAL = Y = DEPENDENT VARIABLE
C      PREDICTED = YC = COMPUTED Y USING REGRESSION EQUATION
C                     = A0 + A1*X1 + A2*X2 + ... + AN*XN
C      RESIDUAL = E = YC - Y
C      NORMALIZED DEVIATE = RESIDUAL / STANDARD ERROR
C      PERCENT DEVIATION  = 100 * RESIDUAL / ACTUAL
C      WEIGHT = INPUT WEIGHT OF OBSERVATION
C      SSE = RESIDUAL SUM OF SQUARES
C      CHI SQUARE = SUM( ( RESIDUAL**2 ) / YC )
C
C********* ANALYSIS PRINTOUT *********
C
C      TOTAL(CORRECTED) SUM OF SQUARES = SUM OF SQUARES ABOUT THE MEAN
C           = SUM((Y-YMEAN)**2) = SUM((YC-YMEAN)**2)+SUM((Y-YC)**2) =SS(Y)
C      TOTAL( ORIGIN ) SUM OF SQUARES = SUM OF SQUARES ABOUT THE ORIGIN.
C           USED INSTEAD OF SS(Y) WHEN REGRESSION IS FORCED THRU ORIGIN.
C      REGRESSION SUM OF SQUARES = SUM OF SQUARES DUE TO REGRESSION
C           = EXPLAINED VARIATION = SUM((YC-YMEAN)**2) = SS(R)
C      RESIDUAL SUM OF SQUARES =   SUM OF SQUARES ABOUT THE REGRESSION
C           = UNEXPLAINED VARIATION = SUM((Y-YC)**2) = SS(E)
C      THE MEAN SQUARES COLUMN IS OBTAINED BY DIVIDING THE SUM OF SQUARES
C              ENTRY BY ITS CORRESPONDING DEGREES OF FREEDOM.
C      RESIDUAL MEAN SQUARE = VARIANCE ABOUT THE REGRESSION = S**2 =MS(E)
C      COEFFICIENT OF MULTIPLE DETERMINATION = PCT OF EXPLAINED VARIATION
C           = (SS DUE TO REGRESSION)/(SS ABOUT MEAN) = CORR. FORM SS(R)
C      CORRELATION COEFFICIENT = R = SQRT( CORR. FORM OF SS(R) )
C      STANDARD ERROR OF ESTIMATE = S = SQRT( MS(E) )
C      S.E. AS PCT. OF MEAN = 100 * S / YMEAN
```

```
C      F  TEST FOR SIGNIFICANCE OF REGRESSION = MS(R)/MS(E)
C
C      CONSTANT = A(0) = YMEAN-SUM(A(I)*XMEAN(I)) = CONSTANT TERM
C      COEFFICIENT = A(I) = THE EFFECT ON Y OF A UNIT INCREASE IN XI WHEN
C           THE OTHER VARIABLES ARE HELD CONSTANT.
C      STANDARD ERROR = STANDARD ERROR OF REGRESSION COEFFICIENT.
C           THE 95 PERCENT CONFIDENCE LIMITS FOR A UNIVERSE REGRESSION
C           COEFFICIENT ARE GIVEN BY THE SAMPLE COEFFICIENT PLUS AND MINUS
C           T(0.025) TIMES THE ESTIMATED STANDARD OF THE COEFFICIENT.
C      COEFF/SE = USED IN HYPOTHESIS THAT COEFFICIENT = 0.
C           = COEFFICIENT DIVIDED BY ITS STANDARD ERROR TO GIVE THE NUMBER
C           OF S.E. AWAY FROM HYPOTHEIZED ZERO. SHOULD BE GREATER THAN T
C           VALUE TO REJECT THE HYPOTHESIS THAT THE COEFFICIENT IS NOT
C           SIGNIFICANTLY DIFFERENT FROM ZERO.
C      F    = F VALUE TO REMOVE VARIABLE FROM REGRESSION. **** NOT USED **
C      BETA COEFFICIENT = MEASURE OF THE NET EFFECT OF EACH VARIABLE ON Y
C      RSQ CHANGE = DECREASE IN RSQ IF THE VARIABLE IS REMOVED FROM REGRE
C
C      PARTIAL RSQ = THE SQUARE OF THE PARTIAL CORRELATION COEFFICIENT OF
C           VARIABLE K NOT IN THE REGRESSION WITH THE RESPONSE Y.
C           = R(KY.LMN... )**2 WHERE L,M,N,... ARE ALREADY IN REGRESSION.
C           = RELATIVE AMOUNT OF IMPROVEMENT THAT IS BROUGHT ABOUT IF
C           VARIABLE K WERE ADDED TO THE REGRESSION.
C      NORMED SUM/SQ = THE NORMALIZED SUM OF SQUARES OF RESIDUALS FOR
C           VARIABLE K  HAD IT TOO BEEN REGRESSED. USEFUL IN DRAWING
C           ATTENTION TO NEAR-LINEAR DEPENDENCIES AMONG THE IND. VARIABLES
C      DELTA RSQ = CHANGE IN RSQ IF VARIABLE K WERE ADDED TO REGRESSION.
C           VARIABLE WITH LARGEST DELTA IS ADDED TO REGRESSION NEXT.
C      F     = F VALUE TO ADD VARIABLE TO REGRESSION. **** NOT USED ****
C
C********* ADDING  AND  DELETING  VARIABLES *****************************
C
C STEP 1 - THE VARIABLE NOT IN THE EQUATION WHICH CAUSES THE GREATEST
C      CHANGE IN RSQ IS ADDED TO THE REGRESSION.
C STEP 2 - THE VARIABLES IN THE EQUATION ARE THEN CHECKED TO SEE IF ONE
C      CAN BE DELETED. THE VARIABLE WHICH CAUSES THE SMALLEST CHANGE IN
C      RSQ IS SELECTED FOR REMOVAL. IF THE EQUATION WITHOUT THIS VARIABLE
C      PRODUCES A SS(R) WHICH IS SMALLER THAN THE PREVIOUS SS(R) FOR THAT
C      NUMBER OF VARIABLES, THE VARIABLE IS REMOVED.
C STEP 3 - IF A VARIABLE WAS REMOVED, REPEAT STEP 2,
C           OTHERWISE REPEAT STEP 1 AND 2.
C
C***********************************************************************
C
C      IT SHOULD BE NOTED THAT THE STATISTICS FOR NON-LINEAR EQUATIONS
C      SHOULD BE USED WITH CARE, AND SHOULD NOT BE COMPARED WITH THOSE
C      FROM LINEAR EQUATIONS, AS THEY HAVE DIFFERENT MEANINGS.
C      FOR EXAMPLE - IF Y IS TRANSFORMED BY TAKING ITS LOGARITHM, THE
C      SUM OF THE SQUARES OF THE ACTUAL RESIDUALS BETWEEN THE CALCULATED
C      AND THE OBSERVED Y VALUES ARE NOT  MINIMIZED, RATHER THE SUM OF
C      SQUARES OF THE LOGARITHMS OF THE  RATIOS  OF THESE VALUES ARE
C      BEING MINIMIZED ( (LOG YC - LOG Y) = ( LOG(YC/Y) ) ).
```

```
C      THEREFORE, COMPARISION OF ANY STATISTICS THAT ARE BASED UPON THE
C      SUM OF THE SQUARES OF THE Y RESIDUALS SUCH AS THE F VALUE OR
C      CORRELATION COEFFICIENT MAY BE MISLEADING.
C
C      IT SHOULD ALSO BE NOTED THAT WHEN THE CURVE IS FORCED THROUGH THE
C      ORIGIN OR SOME OTHER SPECIFIED Y INTERCEPT, THE DEGREES OF FREEDOM
C      ARE CHANGED AND THE CURVE NO LONGER GOES THROUGH THE MEANS OF THE
C      VARIABLES, THEREBY, CHANGING THE VALUES OF THE STATISTICS AND
C      MAKING COMPARISIONS OF CURVES WITH UNSPECIFIED Y INTERCEPTS
C      MISLEADING.   ALSO, COMPARISON OF F VALUES WITH THE STANDARD F
C      DISTRIBUTION IS NOT NECESSARILY VALID.
C**********************************************************************
C
C      THE USERS OF THIS PROGRAM ARE URGED TO REVIEW THE STANDARD TEXTS
C      ON REGRESSION ANALYSIS FOR THE USES AND LIMITATIONS OF THIS
C      TECHNIQUE, AND BEAR IN MIND THAT THE STATISTICAL RELATIONSHIPS ARE
C      NO BETTER THAN THE DATA THAT WAS USED TO COMPUTE THEM.
C**********************************************************************
       COMMON SIGMA(60),A(52,52),SIMCOR(52,52),AVG(60),TEST(60)
       COMMON POINT(60),STRING(3000),INDPAC(30,30),INDEXP(61)
       COMMON INDEX(60),NOUT(60),KSTEP(60),ALPHA(16),YMEAN,IDEN,IFAVE
       COMMON MAXSTP,IFPNCH,NSUMRY,NSKIP,NTAPE9,NEW
       COMMON NOVAR,NBRNOW,NOSTEP,NDATA,NBRXYW,NBRX,LPATH,DEFRM,K
       COMMON IFBACK,IFCNST,IFCORR,NPROB,NBRPVR,TOL,REMARK
       COMMON INDEXY,LBAD,NOGOOD(28)
       COMMON IFWT,YCONST,NYTRA,V(2,51),YTRA(2)
       COMMON  STDERR(50),CORSQR(50)
       DOUBLE PRECISION A,SIMCOR,SIGMA,AVG,TEST
C
C
       DIMENSION  VNAME(2,51),CROSS(8)
       DIMENSION XDATA(255), LOCX(60), CONST(60), NBRTRA(60), RMT(12), FM
      1T(12)
       DIMENSION JVOID(14)
       DIMENSION TRA(3), ATRA(2,17)
       DATA (ATRA(1,J),J=1,17) / 'X+C','X*C','X/C','C/X','X**C','C**X',
      1 '-NX+C','LGX+C','E**C*X','E**C/X','SINC*X','COSC*X','TNC**X',
      2  'SNHC*D',  'CSHC*D','TNHC*D','C TERM'  /
       DATA(CROSS(I),I=1,8) /'V(1)','V(2)','V(3)','V(4)','V(5)',
      1 'V(6)','V(7)','V(8)' /
       DATA NREELS,NEOF/1,0/
       DATA (RMT(J),J=1,12)/6H(7E10.,2H0),10*1H /
       DATA BLANK/1H /
       DATA AX/1HX/,AY/1HY/
       DATA START,VOIDED,SEARCH,TRAN,FORAMT/6HSTART ,6HVOIDED,6HSEARCH,6H
      1 TRA= ,6HJFMT  /
C
       DATA (XDATA(I),I=1,10) /
      1 60H1SELECTED PERCENTILE VALUES OF THE STUDENT T DISTRIBUTION   /,
      2(XDATA(I),I=11,23)       /78H               .10          .05        .01
      3   FOR    DIRECTIONAL (ONE-TAILED) TEST  /,
      4(XDATA(I),I=24,36)/                                           78H  D.F.
```

```
     5    .20         .10         .02      FOR NONDIRECTIONAL (TWO-TAILED) TES
     6T    /
      DATA ( XDATA(J),J=37,188)/4H    1,3.078,6.313,31.821,4H    2,1.886,
     1 2.920,6.965,4H    3,1.638,2.353,4.541,4H    4,1.533,2.132,3.747,
     2 4H    5,1.476,2.015,3.365, 4H    6,1.440,1.943,3.143,
     3 4H    7,1.415,1.895,2.998, 4H    8,1.397,1.860,2.896,
     4 4H    9,1.383,1.833,2.821, 4H   10,1.372,1.812,2.764,
     5 4H   11,1.363,1.796,2.718, 4H   12,1.356,1.782,2.681,
     6 4H   13,1.350,1.771,2.650, 4H   14,1.345,1.761,2.624,
     7 4H   15,1.341,1.753,2.602, 4H   16,1.337,1.746,2.538,
     8 4H   17,1.333,1.740,2.567, 4H   18,1.330,1.734,2.552,
     9 4H   19,1.328,1.729,2.539, 4H   20,1.325,1.725,2.528,
     $ 4H   21,1.323,1.721,2.518, 4H   22,1.321,1.717,2.508,
     $ 4H   23,1.319,1.714,2.500, 4H   24,1.318,1.711,2.492,
     $ 4H   25,1.316,1.708,2.485, 4H   26,1.315,1.706,2.479,
     $ 4H   27,1.314,1.703,2.473, 4H   28,1.313,1.701,2.467,
     $ 4H   29,1.311,1.699,2.462, 4H   30,1.310,1.697,2.457,
     $ 4H   40,1.303,1.684,2.423, 4H   50,1.298,1.676,2.403,
     $ 4H   60,1.296,1.671,2.390, 4H   80,1.292,1.664,2.374,
     $ 4H  100,1.290,1.660,2.365, 4H  200,1.286,1.653,2.345,
     $ 4H  500,1.283,1.648,2.334, 4H  INF,1.282,1.645,2.326 /
      WRITE (6,950) (XDATA(J),J=1,188)
C
10    CALL SLITE (0)
      DO 20 J=1,30
      DO 20 K=1,30
20    INDPAC(J,K)=0
      WRITE (6,960)
C
      READ (5,1100,END=940) NPROB,NXV,NYV,INDEXY,NDATA,IDEN,INTYPE,NREAR
     1,MAXSTP,IFBACK,NSTART,MINSUM,MAXSUM,MAXREG,IFWT,IFCNST,IFLIST,IFSU
     2MS,IFAVE,IFCORR,NCROSS,IFTRA,NSKIP,IFSUB,NFMT,IFPNCH,IFDATE,IFNAME
C
      WRITE (6,1180) NPROB,NXV,NYV,INDEXY,NDATA,IDEN,INTYPE,NREAR,MAXSTP
     1,IFBACK,NSTART,MINSUM,MAXSUM,MAXREG,IFWT,IFCNST,IFLIST,IFSUMS,IFAV
     2E,IFCORR,NCROSS,IFTRA,NSKIP,IFSUB,NFMT,IFPNCH,IFDATE,IFNAME
      D1=BLANK
      D2=BLANK
C         DATE IS MAP S/R TO PICK DATE OFF SEQUENCE CARD
      IF (IFDATE.EQ.0) CALL DATE (D1,D2)
      PRINT 1130, NPROB,NXV,NYV,INDEXY,INTYPE
      NBRNOW=NSTART
      IF (NFMT.NE.0) GO TO 40
      DO 30 J=1,12
30    FMT(J)=RMT(J)
40    CONTINUE
      IF(NDATA.LE.0)NDATA=10000
      NWTT=NXV+NYV
      IF(IFWT.NE.0)NWTT=NWTT+1
      IF (IFSUB.GT.0) CALL NBROFX (IFSUB,NXV)
      NSTEP=0
      NSUMRY=0
```

```
      TOL=0.00001
      NBRXY=NXV+1
C         NBRX   = NUMBER OF INDEPENDENT VARIABLES
C         NBRXY  = NUMBER OF INDEPENDENT VARIABLES + DEPENDENT VARIABLE
C         NBRXYW= SIZE OF ARRAY = NBRXY + 1
C         NBRNOW = NUMBER OF COEFFICIENTS FOR PRESENT  EQUATION
C         INDEX  = INDEX OF PRESENT  EQUATION
      IF(MAXSTP.EQ.0)MAXSTP=998
      IF(MAXREG.LT.5)MAXREG=5
C
C       TO STORE DATA ON TAPE FOR USE IN ANOTHER PROBLEM, SET INTYPE=1
C         FOR REWIND,  INTYPE=3 FOR NO REWIND
C         TO USE DATA FROM A PREVIOUS PROBLEM    SET INTYPE = 2 OR 5,
C         THEREBY CAUSING TAPE TO REWIND.
C         INTYPE = 4 OR 6 FOR BIN. TAPE 10 OR 11 TO BE USED AS INPUT
C                ALSO PREVENTS TAPE REWIND AT START OF PROBLEM.
      NREWB5=1
      IF (INTYPE.NE.7) GO TO 50
      NREWB5=0
      INTYPE=5
50    NTAPE=10
      NREAD=0
      NWRITE=0
      IF (INTYPE.EQ.0) GO TO 60
      IF(INTYPE.EQ.1.OR.INTYPE.EQ.3)NWRITE=1
      IF(INTYPE.EQ.5.OR.INTYPE.EQ.6)NTAPE=11
      IF (INTYPE.EQ.1.OR.INTYPE.EQ.2.OR.INTYPE.EQ.5) REWIND NTAPE
      IF(INTYPE.NE.1.AND.INTYPE.NE.3)NREAD=1
C
C         NUMBER OF INDEPENDENT + DEPENDENT VARIABLES
60    NTOTAL=NXV+NYV
      NEW=NXV+INDEXY
      IF (NTOTAL.LE.52) GO TO 70
C         TOO MANY VARIABLES
      NTOTAL=NEW
      IF (NTOTAL.LE.52) GO TO 70
      WRITE (6,970) NTOTAL
      CALL EXIT
70    NOSTEP=0
      IF (NCROSS.EQ.0) GO TO 80
      NOVAR=(NBRXY*(NBRXY+1))/2
      IF (NOVAR.LE.51) GO TO 90
      NCROSS=0
      WRITE (6,1000)
80    NOVAR=NBRXY
90    IF(IFCNST.LT.0)NOVAR=NOVAR+1
      NBRXYW=NOVAR+1
      NBRX=NOVAR-1
      MAXVAR=MIN0(MAXSUM,MAXREG,NBRX)
C         READ CONTROL CARD C
      IF (IDEN) 120,100,130
100   DO 110 J=1,16
```

```
110   ALPHA(J)=BLANK
120   IDEN=IABS(IDEN)
      GO TO 150
130   DO 140 I=1,IDEN
      READ (5,980) (ALPHA(J),J=1,16)
140   WRITE (6,990) (ALPHA(J),J=1,16)
150   CONTINUE
C          READ CONTROL CARD D
      DO 160 J=1,60
      KSTEP(J)=1
160   TEST(J)=+2.0E+30
      IF (NSTART) 190,220,170
170   READ (5,1080) (TEST(J),INDEX(J),J=1,NBRNOW)
C          PACK INDEX
      DO 180 J=1,NBRNOW
180   CALL PACK (NBRNOW,J,INDEX(J),1)
      GO TO 210
190   NBRNOW=NXV
      DO 200 J=1,NBRNOW
      CALL PACK (NBRNOW,J,J,1)
      TEST(J)=1.0D0
200   IF(NSTART.LT.-1)TEST(J)=0.0
210   WRITE (6,1090) START,(TEST(J),INDEX(J),J=1,NBRNOW)
220   MINVAR=MAX0(1,MINSUM)
      NTAPE9=1
      IF (IFBACK.EQ.0) GO TO 230
      IF (NDATA*NBRXYW.LE.3000) GO TO 230
      NTAPE9=0
      REWIND 9
230   DO 240 I=1,NBRXYW
      DO 240 J=1,NBRXYW
240   A(I,J)=0.0
C          READ CONTROL CARD E
      IF (NSKIP.LE.0) GO TO 290
      READ (5,1100) (JVOID(J),J=1,14)
      JV=0
      DO 280 L=1,28,2
      NOGOOD(L)=0
      NOGOOD(L+1)=0
250   IF (JV.EQ.14) GO TO 280
      JV=JV+1
      IF (JVOID(JV)) 270,250,260
260   NOGOOD(L)=JVOID(JV)
      NOGOOD(L+1)=JVOID(JV)
      LBAD=L+1
      IF (JVOID(JV+1)) 250,280,280
270   NOGOOD(L+1)=IABS(JVOID(JV))
280   CONTINUE
290   NSKIP=IABS(NSKIP)
      IF (NSKIP.NE.0) WRITE (6,1110) VOIDED,(NOGOOD(J),J=1,LBAD)
C          READ CONTROL CARD F
      LOOK=0
```

```
      IF (NREAR) 310,320,300
C          IF NREAR=1      READ SET OF SEARCH PARAMETERS.
300   NNXJ=NXV
      IF(IFWT.NE.0)NNXJ=NXV+1
      READ (5,1100) NWORDS,LOCY,(LOCX(J),J=1,NNXJ)
310   WRITE (6,1110) SEARCH,NWORDS,LOCY,(LOCX(J),J=1,NNXJ)
      LOOK=1
320   CONTINUE
C          READ CONTROL CARD G
C               READ TRANSFORMATIONS
      IF (IFTRA.GT.0) READ (5,1080) (CONST(I),NBRTRA(I),I=1,NTOTAL)
      IF (IFLIST.NE.0) GO TO 330
      WRITE (6,1160) ALPHA,D1,D2,(AX,J,J=1,NXV),(AY,J,J=1,NYV)
330   NW1=NTOTAL
      IF (IFTRA.NE.0) WRITE (6,1090) TRAN,(CONST(I),NBRTRA(I),I=1,NTOTAL
     1)
C          READ CONTROL CARD H      NAME OF VARIABLES
      IF (IFNAME) 370,360,340
340   DO 350 J=1,2
      DO 350 L=1,NTOTAL
350   VNAME(J,L)=BLANK
      GO TO 370
360   READ (5,1140) ((VNAME(J,L),J=1,2),L=1,NTOTAL)
370   IF (IFLIST.EQ.0) WRITE (6,1150) ((VNAME(J,L),J=1,2),L=1,NTOTAL)
C
C          READ CONTROL CARD I  ( VARIABLE FORMAT )
      IF (NFMT.GT.0) READ (5,1120) FMT
      IF (NFMT.NE.0) WRITE (6,1120) FORAMT,FMT
      IF(IFWT.NE.0)NWT=NTOTAL+1
      NSKIP=IABS(NSKIP)
      NBAD=0
      YCONST=CONST(NEW)
      DATA  ACT,UAL/6HACTUAL,1H /
      YTRA(1)=ACT
      YTRA(2)=UAL
      J=NBRTRA(NEW)
      NYTRA=0
      IF (IFTRA.EQ.0.OR.J.EQ.0) GO TO 380
      IF(NBRTRA(NEW).EQ.7)NYTRA=-1
      IF(NBRTRA(NEW).EQ.8)NYTRA=+1
      YTRA(1)=ATRA(1,J)
      YTRA(2)=ATRA(2,J)
380   CONTINUE
C
C
      DO 590 N=1,NDATA
      IF (NREAD) 440,440,390
390   IF (LOOK) 400,400,420
400   IF (ENDFIL(NTAPE,NREELS,NEOF)) 410,930,410
410   READ (NTAPE) (POINT(J),J=1,NWTT)
      GO TO 490
420   IF (ENDFIL(NTAPE,NREELS,NEOF)) 430,930,430
```

```
430   READ (NTAPE) (XDATA(J),J=1,NWORDS)
      GO TO 470
C
440   IF (LOOK) 450,450,460
450   READ (5,FMT) (POINT(J),J=1,NWTT)
      IF (NWRITE.NE.0) WRITE (10) (POINT(J),J=1,NWTT)
      GO TO 490
460   READ (5,FMT) (XDATA(J),J=1,NWORDS)
      IF (NWRITE.NE.0) WRITE (10) (XDATA(J),J=1,NWORDS)
470   DO 480 J=1,NNXJ
      JLOC=LOCX(J)
480   POINT(J)=XDATA(JLOC)
      POINT(NWT)=POINT(NNXJ)
      POINT(NEW)=XDATA(LOCY)
C
C          CHECK FOR END OF DATA INDICATOR
490   IF (POINT(1).EQ.99999999.) GO TO 600
      JDATA=N
      IF (IFSUB.GT.0) CALL EQUAT (IFSUB,POINT)
      IF (IFLIST.EQ.0) WRITE (6,1060) N,(POINT(J),J=1,NWT)
      IF (IFTRA.EQ.0) GO TO 500
      CALL CHANGE (POINT,NBRTRA,CONST,NTOTAL)
      IF (IFLIST.EQ.0) WRITE (6,1070) N,(POINT(J),J=1,NWT)
C
500   CONTINUE
C
      POINT(NOVAR)=POINT(NEW)
      WHT=1.0
      IF(IFWT.NE.0)WHT=POINT(NWT)
      POINT(NBRXYW)=WHT
      IF (NCROSS.EQ.0) GO TO 520
C                    CROSS PRODUCTS ARE USED AS INDEPENDENT VARIABLES
      L=NBRXY
      DO 510 I=2,NBRXY
      DO 510 J=I,NBRXY
      POINT(L)=POINT(I-1)*POINT(J-1)
510   L=L+1
520   IF(IFCNST.LT.0)POINT(NOVAR-1)=1.0
      IF (IFBACK.EQ.0) GO TO 550
      IF (NTAPE9.EQ.0) GO TO 540
C          STORE IN STRING IF DATA POINTS * VARIABLES LESS THAN 3000
      DO 530 J=1,NBRXYW
      JJ=NBRXYW*(N-1)+J
530   STRING(JJ)=POINT(J)
      GO TO 550
C     STORE DATA ON TAPE 9 IF DATA POINTS * VARIABLES EXCEED 3000
540   WRITE (9) (POINT(K),K=1,NOVAR),WHT
C
550   CONTINUE
      IF (NSKIP.EQ.0) GO TO 570
C          CHECK TO SEE IF POINT IS TO BE DELETED FROM REGRESSION
      DO 560 J=1,LBAD,2
```

```
      IF (N.LT.NOGOOD(J).OR.N.GT.NOGOOD(J+1)) GO TO 560
      NBAD=NBAD+1
      GO TO 590
560   CONTINUE
C
570   DO 580 I=1,NOVAR
C          SUM X(I)
      A(I,NBRXYW)=A(I,NBRXYW)+POINT(I)*WHT
      DO 580 J=I,NOVAR
C          SUM X(I)*X(J)
580   A(I,J)=A(I,J)+POINT(I)*POINT(J)*WHT
      A(NBRXYW,NBRXYW)=A(NBRXYW,NBRXYW)+WHT
590   CONTINUE
C
C     ************************************************************************
C
600   NDATA=JDATA-NBAD
      DEFRM=NDATA
      DENOM=A(NBRXYW,NBRXYW)-1.0
      IF(IFWT.NE.0)DENOM=DENOM+1.0
      IF (NTAPE9.EQ.0) REWIND 9
      IF (NREWB5.EQ.0) REWIND 11
C
      WRITE (6,1170) NPROB,ALPHA,NDATA,NOVAR,A(NBRXYW,NBRXYW),D1,D2
      K=2
      IF(IFTRA.EQ.0)K=1
      WRITE (6,1010) (BLANK,J=1,K)
      WRITE (6,1020)
      IX1=60
      IX2=0
C
      DO 660 J=1,NOVAR
      YMEAN=A(J,NBRXYW)/A(NBRXYW,NBRXYW)
      STDEV=SQRT(ABS((A(J,J)-A(J,NBRXYW)*YMEAN)/DENOM))
      TRA(1)=BLANK
      TRA(2)=BLANK
      V(1,J)=VNAME(1,J)
      V(2,J)=VNAME(2,J)
      L=J
      K=2
      IF (J.GT.NXV) GO TO 630
610   IF (IFTRA.EQ.0) GO TO 650
      I=NBRTRA(L)
      IF (I.LE.0) GO TO 650
      K=3
      TRA(3)=CONST(L)
620   TRA(1)=ATRA(1,I)
      TRA(2)=ATRA(2,I)
      GO TO 650
630   I=17
      V(1,J)=ATRA(1,I)
      V(2,J)=ATRA(2,I)
```

```
      IF (IFCNST.LT.0.AND.J.EQ.NOVAR-1) GO TO 620
      L=NXV+INDEXY
      V(1,J)=VNAME(1,L)
      V(2,J)=VNAME(2,L)
      IF (J.EQ.NOVAR) GO TO 610
C          CROSS PRODUCTS
      IX1=IX1+1
      IF (IX1.LE.NXV) GO TO 640
      IX2=IX2+1
      IX1=IX2
640   WRITE (6,1040) J,YMEAN,STDEV,IX2,IX1
      V(1,J)=CROSS(IX2)
      V(2,J)=CROSS(IX1)
      GO TO 660
650   IF (J.EQ.NOVAR) GO TO 660
      WRITE (6,1030) V(1,J),V(2,J),J,YMEAN,STDEV,(TRA(L),L=1,K)
660   CONTINUE
      WRITE (6,1050) V(1,J),V(2,J),YMEAN,STDEV,(TRA(L),L=1,K)
C
      IF (IFSUMS.EQ.0) CALL PRTSUM
      IF (IFCNST.NE.0) GO TO 670
      CALL RESID
      DEFRM=DEFRM-1.0
670   CALL CORREL
      CALL SLITET (1,LIGHT)
      IF (LIGHT.EQ.1) GO TO 10
      NDATA=JDATA
      NSKIP=NBAD
      IF (MAXSTP.EQ.999) WRITE (6,1220)
C     ************************************************************************
C
680   LPATH=1
      KPATH=1
      IF (NBRNOW.GT.0) GO TO 820
690   IF (A(NOVAR,NOVAR).GT.0.0) GO TO 730
      WRITE (6,1200) A(NOVAR,NOVAR)
      PRINT 1200, A(NOVAR,NOVAR)
      GO TO 860
C
700   JP=0
      NBRPVR=NBRNOW
      NBRNOW=NBRNOW-1
      DO 710 J=1,NBRPVR
      CALL PACK (NBRPVR,J,0,1)
      IF (INDEX(J).EQ.K) GO TO 710
      JP=JP+1
      CALL PACK (NBRNOW,JP,INDEX(J),1)
710   CONTINUE
      DO 720 I=1,NOVAR
      DO 720 J=1,NOVAR
720   A(I,J)=SIMCOR(I,J)
      GO TO 680
```

```
C
730   CALL ADDTO
      CALL SLITET (1,LIGHT)
      GO TO (860,740), LIGHT
740   GO TO (800,830,800), LPATH
C
750   TEST(NBRNOW)=A(NOVAR,NOVAR)
760   CALL OUTPUT
      CALL SLITET (2,LIGHT)
      IF (LIGHT.EQ.1) GO TO 700
      NSTEP=NSTEP+1
      IF (NSTEP.GT.MAXSTP) GO TO 860
      IF (NBRNOW.GE.MAXREG) GO TO 860
      CALL SLITET (1,LIGHT)
      GO TO (860,770), LIGHT
770   GO TO (790,780,10), KPATH
780   KPATH=1
790   IF (NBRNOW.LE.2) GO TO 810
      CALL REMOVE
      IF (REMARK.EQ.0.0) GO TO 810
      LPATH=1
800   CALL MATRIX
      GO TO (760,850,810), LPATH
810   IF (NBRNOW.LT.NBRX) GO TO 690
      WRITE (6,1190)
      GO TO 860
820   KPATH=2
830   L=1
      LPATH=2
840   CONTINUE
      CALL PACK (NBRNOW,L,K,2)
      GO TO 800
C      ********************** TRY ADJUNCTION ****************************
850   L=L+1
      IF (L.LE.NBRNOW) GO TO 840
      GO TO (810,750,910), KPATH
C
C          SUMMARY
860   WRITE (6,1230) ALPHA,NPROB
      IF(IFBACK.EQ.998)IFBACK=NSTEP
      IF(IFBACK.EQ.999)IFBACK=1
      IF(MAXSTP.EQ.999)MAXSTP=998
      DO 880 J=1,NBRNOW
      DO 870 L=1,J
870   CALL PACK (J,L,INDEX(L),2)
      IF (INDEX(1).LE.0) GO TO 880
      WRITE (6,1240) J,KSTEP(J),STDERR(J),CORSQR(J),(INDEX(L),L=1,J)
880   CONTINUE
      KPATH=3
      NBRNOW=MINVAR
      NSUMRY=1
890   IF (NBRNOW.GT.MAXVAR) GO TO 10
```

```
      CALL PACK (NBRNOW,1,J,2)
      IF (J.LE.0) GO TO 920
      DO 900 I=1,NOVAR
      DO 900 J=1,NOVAR
900   A(I,J)=SIMCOR(I,J)
      GO TO 850
910   CALL OUTPUT
920   NBRNOW=NBRNOW+1
      GO TO 890
930   WRITE (6,1210) JDATA,(POINT(J),J=1,NWTT)
940   STOP
C
C
C
C
950   FORMAT (10A6//13A6/13A6//40(A4,2X,3F10.3/))
960   FORMAT (47H1 NONSIMPLE STEPWISE MULTIPLE LINEAR REGRESSION45X23HAR
     1MYCOMPUTATIONCENTER/92X20HARMY MISSILE COMMAND/92X25HREDSTONE ARSE
     2NAL, ALABAMA)
970   FORMAT (37H0 TOTAL NUMBER OF VARIABLES TOO LARGEI6)
980   FORMAT (16A5)
990   FORMAT (/5X,16A5)
1000  FORMAT (47H  CROSS PRODUCTS DELETED, AS THERE ARE TOO MANY//)
1010  FORMAT (20X35HVAR     WTED AVERAGE       STAND. DEV. ,2A1,3X,25HTRANS
     1FORMATION    CONSTANT)
1020  FORMAT (1X)
1030  FORMAT (7X,2A6,I3,1X,2E16.5,6X,2A6,E16.4)
1040  FORMAT (19X,I3,1X,2E16.5,6X,2HX(I2,6H) * X(I2,1H))
1050  FORMAT (7X,2A6,3H  Y,1X,2E16.5,6X,2A6,E16.4)
1060  FORMAT (I9,1X,10F12.3/(10X,10F12.3))
1070  FORMAT (5X,1H(I3,1H),10F12.3/(10X,10F12.3))
1080  FORMAT (7(F8.0,I2))
1090  FORMAT (1H0,3XA6,10(F10.3,I2)/(10X,10(F10.3,I2)))
1100  FORMAT (14I5)
1110  FORMAT (1H0,A6,24I5/(7X24I5))
1120  FORMAT (13A6)
1130  FORMAT (30X,8HDATA SETI6,10X,4I5)
1140  FORMAT (7(A6,A4))
1150  FORMAT (12X20A6)
1160  FORMAT (34H1  * *  I N P U T       D A T A  * *5X16A5,2XA6,A2//6(9X,
     110(7XA1,1H(I2,1H))/))
1170  FORMAT (37H1STEPWISE REGRESSION PROBLEM NUMBER  I5,10X16A5/23H NUM
     1BER OF OBSERVATIONS14X,I5/20H NUMBER OF VARIABLES17XI5/30H WEIGHTE
     2D DEGREES OF FREEDOM  F12.3,78X2A6//)
1180  FORMAT (//35H  P R O G R A M       C O N T R O L S////10H   NPROB =I
     15,10H    NXV    =I5,10H    NYV    =I5,10H    INDEXY=I5,10H    NDATA =I5,
     210H    IDEN   =I5,10H    INTYPE=I5//10H    NREAR =I5,10H    MAXSTP=I5,1
     30H    IFBACK=I5,10H    NSTART=I5,10H    MINSUM=I5,10H    MAXSUM=I5,10H
     4    MAXREG=I5//10H    IFWT   =I5,10H    IFCNST=I5,10H    IFLIST=I5,10H
     5  IFSUMS=I5,10H    IFRES =I5,10H    IFCORR=I5,10H    NCROSS=I5//10H
     6 IFTRA =I5,10H    NVOID =I5,10H    IFSUB =I5,10H    NFMT   =I5,10H    I
     7FPNCH=I5,10H    IFDATE=I5,10H    IFNAME=,I5)
```

```
1190  FORMAT (51H0 **************** VARIABLES EXHAUSTED ************/)
1200  FORMAT (80H0************* PROBLEM TERMINATED, SUM OF SQUARES IS NO
     1N-POSITIVE *************E13.5/)
1210  FORMAT (//34H END OF FILE REACHED AFTER READINGI5,8H POINTS./20H0L
     1AST OBSERVATION IS//(1X,8E16.8))
1220  FORMAT (33H1MULTIPLE LINEAR REGRESSION STEPS)
1230  FORMAT (34H1SUMMARY OF BEST SETS OF VARIABLES,2X16A5,9H  PROBLEMI6
     1//25H  N STEP    STD ERR    RSQ//)
1240  FORMAT (1X,I2,I4,F11.3,F9.6,1X,35I3/28X,35I3)
      END
```

```
      SUBROUTINE OUTPUT
      COMMON SIGMA(60),A(52,52),SIMCOR(52,52),AVG(60),TEST(60)
      COMMON POINT(60),STRING(3000),INDPAC(30,30),INDEXP(61)
      COMMON INDEX(60),NOUT(60),KSTEP(60),ALPHA(16),YMEAN,IDEN,IFAVE
      COMMON MAXSTP,IFPNCH,NSUMRY,NSKIP,NTAPE9,NEW
      COMMON NOVAR,NBRNOW,NOSTEP,NDATA,NBRXYW,NBRX,LPATH,DEFRM,K
      COMMON IFBACK,IFCNST,IFCORR,NPROB,NBRPVR,TOL,REMARK
      COMMON INDEXY,LBAD,NOGOOD(28)
      COMMON IFWT,YCONST,NYTRA,V(2,51),YTRA(2)
      COMMON  STDERR(50),CORSQR(50)
      DOUBLE PRECISION A,SIMCOR,SIGMA,AVG,TEST
C
      DIMENSION  COEFF(61), ABC(5)
      DOUBLE PRECISION COEFF,CONST
      DOUBLE PRECISION SUMSQ,TSS,SIGY2,SIGY
      DOUBLE PRECISION YPRED,YOBS,DEV,RSQ,SQREG,SQREG2
      DOUBLE PRECISION  DEVSQ,CHISQ,SUMSQU,CHISQU,DEVU,YO,YC
      DATA BLANK/1H /,VOID/6HVOIDED/,CHECK/6HREVIEW/
      DATA  ACTUAL/6HACTUAL/
C          NBRNOW = NUMBER OF COEFFICIENTS FOR PRESENT  EQUATION
C          INDEX  = INDEX OF PRESENT  EQUATION
C          NSUMRY = 0 FOR BUILDING PHASE. = 1 FOR SUMMARY PHASE.
      KPATH=1
      IF (NSKIP.NE.0) KPATH=2
      WHT=1.0
      TSS = SIGMA(NOVAR)*SIGMA(NOVAR)
      CALL SLITET (1,LIGHT)
      GO TO (10,20), LIGHT
   10 CALL SLITE (1)
      GO TO 30
   20 NOSTEP=NOSTEP+1
      IF (NSUMRY.EQ.1) NOSTEP=KSTEP(NBRNOW)
   30 DO 40 J=1,60
   40 NOUT(J)=0
      DO 50 J=1,NBRNOW
      CALL PACK (NBRNOW,J,I,2)
      INDEX(J)=I
      NOUT(I)=1
C          BETA = A(I,NOVAR)
   50 COEFF(J)=A(I,NOVAR)*SIGMA(NOVAR)/SIGMA(I)
      IF (IFCNST.EQ.0) GO TO 60
      CONST=0.0
      GO TO 80
   60 CONST=AVG(NOVAR)
      DO 70 I=1,NBRNOW
      J=INDEX(I)
   70 CONST=CONST-(COEFF(I)*AVG(J))
   80 SUMSQ = A(NOVAR,NOVAR) * TSS
      XVAR=NBRNOW
      DEFR=DEFRM-XVAR
      NDEFR=DEFR
      NDEFRM=DEFRM
```

AD-A076 354 ALABAMA UNIV IN HUNTSVILLE F/G 13/4
METHODOLOGY FOR CONTAINER CUSHIONING MODEL DEVELOPMENT AND VALI--ETC(U)
AUG 79 R M WYSKIDA , J D JOHANNES DAAK40-78-C-0146
UNCLASSIFIED UAH-221-VOL-2 USAMICOM-TRT-CR-79-24-VOL- NL

2 OF 2
AD
A076354

END
DATE
FILMED
12-79
DDC

```
      IF (MAXSTP.EQ.999) GO TO 230
      IF (IFBACK.EQ.0.OR.IFBACK.GT.NOSTEP) GO TO 230
C
C     **** COMPUTE BACK SOLUTION
      SIGY=DSQRT(SUMSQ/DEFR)
      NABC = 1
      IF( IFWT.NE.0 ) NABC = 2
      IF(NYTRA.NE.0) NABC = 5
      IF( NDEFR.LE.0 )  SIGY = 0.0D0
      SUMSQ=0.0D0
      CHISQ = 0.0D0
      SUMSQU = 0.0D0
      CHISQU = 0.0D0
      NONO = 0
      NDROP=0
      LINE=50
C
      DO 220 N=1,NDATA
      IF (NTAPE9.NE.0) GO TO 90
      READ (9) (POINT(L),L=1,NOVAR),WHT
      GO TO 110
   90 JJ=NBRXYW*(N-1)
      DO 100 L=1,NOVAR
      KK=JJ+L
  100 POINT(L)=STRING(KK)
      JJ=NBRXYW*N
      WHT=STRING(JJ)
C
  110 YPRED=CONST
      DO 120 I=1,NBRNOW
      J=INDEX(I)
  120 YPRED=YPRED+COEFF(I)*POINT(J)
C
      YOBS=POINT(NOVAR)
      DEV=YPRED-YOBS
      DEVN=DEV/SIGY
      IF( NDEFR .LE. 0 )  DEVN = 0.0
      PCT=DEV/YOBS*100.0
      GOOD=BLANK
      IF (ABS(DEVN).GT.3.5) GOOD=CHECK
      GO TO (150,130), KPATH
  130 DO 140 J=1,LBAD,2
      IF (N.LT.NOGOOD(J).OR.N.GT.NOGOOD(J+1)) GO TO 140
      GOOD=VOID
      NBAD=NBAD+1
      IF (NBAD.EQ.NSKIP) KPATH=1
      GO TO 160
  140 CONTINUE
  150 DEVSQ = (DEV*DEV)*WHT
      SUMSQ = SUMSQ + DEVSQ
      CHISQ = CHISQ + DEVSQ/YPRED
  160 LINE=LINE+1
```

```
      M = NABC
      ABC(1) = GOOD
      ABC(2) = WHT
      IF (NYTRA) 170,190,180
  170 IF (YOBS.GT.15..OR.YPRED.GT.15.) GO TO 185
      YO = DEXP(YOBS) -YCONST
      YC = DEXP(YPRED)-YCONST
      GO TO 200
  180 IF (YOBS.GT.8..OR.YPRED.GT.8.) GO TO 185
      YO = 10.0D0**YOBS  - YCONST
      YC = 10.0D0**YPRED - YCONST
  200 CONTINUE
      DEVO = YC - YO
      ABC(3) = YO
      ABC(4) = YC
      ABC(5) = DEVO
      IF( GOOD.EQ.VOID ) GO TO 190
      DEVSQ = (DEVO*DEVO) * WHT
      SUMSQO = SUMSQO + DEVSQ
      CHISQO = CHISQO + DEVSQ/YC
      GO TO 190
  185 NONO = 1
      M = 2
  190 CONTINUE
      IF (LINE.LE.50) GO TO 210
      WRITE(6,460)NOSTEP,NPROB,(V(J,NOVAR),J=1,2),(YTRA(J),J=1,2),
     *     (BLANK, J=1,NABC)
      LINE=1
  210 WRITE(6,470) GOOD,N,YOBS,YPRED,DEV,DEVN,PCT,(ABC(J),J=1,M)
  220 CONTINUE
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SIGY = DSQRT( SUMSQ/DEFR )
      SIGYO = DSQRT(SUMSQO/DEFR)
      L=1
      IF(YTRA(1).NE.ACTUAL) L=2
      WRITE(6,490) SIGY, CHISQ, SUMSQ, (BLANK, J=1,L)
      IF( NABC.EQ.5 .AND. NONO.EQ.0 ) WRITE(6,500) SIGYO,CHISQO,SUMSQO
      IF (NTAPE9.EQ.0) REWIND 9
      SUMSQ = A(NOVAR,NOVAR) * TSS
C
  230 CONTINUE
      IF (A(NOVAR,NOVAR).LT.0.0) A(NOVAR,NOVAR)=0.0D0
      SUMSQ=DABS(SUMSQ)
      IF (NDEFR.GT.0) GO TO 240
      WRITE (6,370) NOSTEP
      SIGY=0.0
      SIGPCT=0.0
      REGRCO=0.0
      CALL SLITE (1)
      STDERR(NBRNOW) = 0.0
      CORSQR(NBRNOW) = 0.0
      IF (MAXSTP.EQ.999) RETURN
```

```
      GO TO 260
  240 CONTINUE
      SIGY2=SUMSQ/DEFR
      SIGY=DSQRT(SIGY2)
      RSQ=1.0D0-A(NOVAR,NOVAR)
      IF( NOSTEP .NE. KSTEP(NBRNOW) )  GO TO 245
      STDERR(NBRNOW) = SIGY
      CORSQR(NBRNOW) = RSQ
  245 CONTINUE
      IF (MAXSTP.EQ.999) RETURN
      AYY=A(NOVAR,NOVAR)
      TR=1.0
      REGRCO=SQRT(RSQ)
      R2=RSQ/XVAR
      V2=A(NOVAR,NOVAR)/DEFR
      FTEST=R2/V2
      SQREG=RSQ*TSS
      SQREG2=SQREG/XVAR
      DATA CORREC,TED/6HCORREC,3HTED/,ORI,GIN/6H ORIGI,1HN/
      BASE1=CORREC
      BASE2=TED
      IF (IFCNST.EQ.0) GO TO 250
      BASE1=ORI
      BASE2=GIN
  250 WRITE (6,440) NOSTEP,ALPHA,NPROB,BASE1,BASE2,NDEFRM,TSS,TR,NBRNOW,
     1SQREG,SQREG2,RSQ,R2,FTEST,NDEFR,SUMSQ,SIGY2,AYY,V2
      SIGPCT=ABS(SIGY/YMEAN*100.0)
C
  260 WRITE(6,380) SIGY,V(1,NOVAR),V(2,NOVAR),SIGPCT,REGRCO
      DO 290 J=1,NBRX
  280 POINT(J)=A(J,NOVAR)*A(NOVAR,J)/A(J,J)
  290 CONTINUE
C
      WRITE (6,400)
      KK=0
      IF (IFPNCH.NE.0.AND.NSUMRY.EQ.1) WRITE (7,480) KK,CONST,NPROB,NBRN
     1OW,INDEXY,NOSTEP,IFPNCH,SIGPCT,REGRCO
      IF (IFCNST.EQ.0) WRITE (6,390) CONST
C
C     **** LIST COEFFICIENTS
      DO 310 J=1,NBRNOW
      I=INDEX(J)
      IF (IFPNCH.EQ.0.OR.NSUMRY.EQ.0) GO TO 300
      WRITE (7,480) I,COEFF(J),NPROB,NBRNOW,INDEXY,NOSTEP,IFPNCH,SIGPCT,
     1REGRCO
  300 SEB=SQRT(ABS(A(I,I)))*SIGY/SIGMA(I)
      CT=COEFF(J)/SEB
      IF( SEB .EQ. 0.0 )  CT = 0.0
      WRITE(6,410) V(1,I),V(2,I), I,COEFF(J),SEB,CT,A(I,NOVAR),POINT(I)
      IF( POINT(I).LE.0.0 )  GO TO 310
      WRITE(6,510)
      CALL SLITE (2)
```

```
      K = I
  310 CONTINUE
C
      IF (NBRNOW.EQ.NBRX) GO TO 360
      IF (NDEFR.LE.0) GO TO 360
      WRITE (6,420)
      NP=0
      DO 350 I=1,NBRX
      PAR=POINT(I)/AYY
      IF( A(I,I).LE.TOL )  POINT(I) =3.33333333E33
      IF (NOUT(I)) 350,320,350
  320 IF (NP) 340,330,340
  330 IH=I
      RPAR=PAR
      SSN=A(I,I)
      DELT=POINT(I)
      NP=1
      GO TO 350
  340 WRITE (6,430) IH,RPAR,SSN,DELT,I,PAR,A(I,I),POINT(I)
      NP=0
  350 CONTINUE
      IF (NP.NE.0) WRITE (6,430) IH,RPAR,SSN,DELT
  360 RETURN
C
  370 FORMAT (29H1NO MORE DEGREES FREEDOM STEPI5/1H0,120(1H*))
  380 FORMAT (26H0STANDARD ERROR OF Y F16.6,5X2A6/24H S.E. AS PERCENT OF
     * MEAN F12.6/24H CORRELATION COEFFICIENT F12.6 // )
  390 FORMAT (7X,15HCONST. TERM   0,E17.8)
  400 FORMAT(20X,48HVAR    COEFFICIENT            STAND. ERROR    COEF/SE,7X,
     121HBETA          RSQ CHANGE)
  410 FORMAT (7X,A6,I3, E17.8,E19.6,F10.4,2F14.8)
  420 FORMAT(1HK21X58HREGRESSION OF THE VARIABLE K ON THE SET OF VARIABL
     1ES ABOVE/1HJ,2(4X,44HK   PARTIAL RSQ   NORMED SUM/SQ    DELTA RSQ,
     211X))
  430 FORMAT (2X,2(I4,F13.7,1X2E16.7,9X))
  440 FORMAT (5H1STEPI4,9X,16A5,9H  PROBLEMI6/6H0ANOVA,21X,30H....... OR
     1IGINAL UNITS .......9X,26H.... CORRELATION FORM ..../6H  SOURCE13X
     2,36HD.F.  SUM OF SQUARES    MEAN SQUARES9X,26HSUM SQUARES   MEAN S
     3QUARES11X,9HOVERALL F/7H0TOTAL(A6,A3,1H)I7,E17.8,20X,F15.8/11H REG
     4RESSIONI13,2E17.8,3X,2F15.8,F20.4/11H RESIDUAL  I13,2E17.8,3X,2F15
     5.8)
  460 FORMAT (5H1STEPI4,10X,43HRESIDUAL ANALYSIS  ( ACTUAL VS CALCULATED
     1 ),9X,7HPROBLEMI6/1H0,22X,A6,A4/7X,16HOBSERVATION      A6,A4,13H
     2CALCULATED,8X,33HRESIDUAL      NOR DEV     PCT DEV,A3,A6,6HWEIGHT,3A1
     3,30H   ACTUAL        CALC.       DEV //)
  470 FORMAT (1X,A6,I6,F17.5,2F16.5,F10.3,F11.3,1XA6,F8.4,F12.2,2F11.2)
  480 FORMAT (I5,E20.8,5I5,F20.4,F10.7)
  490 FORMAT (17HKSTANDARD ERROR =F10.3,15H , CHI SQUARE =F11.4,2X,
     *   7H  SSE =E15.8,2A1,34HFOR TRANSFORMED DEPENDENT VARIABLE )
  500 FORMAT (17H0STANDARD ERROR = F10.3,15H , CHI SQUARE =F11.4,2X,
     *   7H  SSE =E15.8,2X ,34HFOR RECONVERTED DEPENDENT VARIABLE )
  510 FORMAT(1X,50(1H*),53H ILL-CONDITIONED SET --- RESULTS IN DOUBT. DO
```

```
*NOT USE , 19(1H*) )
 END
```

```
C       S/R TO PRINT SUMS OF CROSS PRODUCTS
        SUBROUTINE PRTSUM
C         RAW SUMS OF SQUARES AND CROSS PRODUCTS
        COMMON SIGMA(60),A(52,52),SIMCOR(52,52),AVG(60),TEST(60)
        COMMON POINT(60),STRING(3000),INDPAC(30,30),INDEXP(61)
        COMMON INDEX(60),NOUT(60),KSTEP(60),ALPHA(16),YMEAN,IDEN,IFAVE
        COMMON MAXSTP,IFPNCH,NSUMRY,NSKIP,NTAPE9,NEW
        COMMON NOVAR,NBRNOW,NOSTEP,NDATA,NBRXYW,NBRX,LPATH,DEFRM,K
        COMMON IFBACK,IFCNST,IFCORR,NPROB,NBRPVR,TOL,REMARK
        COMMON INDEXY,LBAL,NOGOOD(28)
        COMMON IFW1,YCONST,NYTRA,V(2,51),YTRA(2)
        COMMON  STDERR(50),CORSQR(50)
        DOUBLE PRECISION A,SIMCOR,SIGMA,AVG,TEST
C
        DATA JB/1H /
        WRITE (6,150)
        WRITE (6,160) (JB,I,A(I,NBRXYW),I=1,NBRX)
        WRITE (6,170) A(NOVAR,NBRXYW)
        WRITE (6,180)
        WRITE (6,190) ((JB,I,J,A(I,J),J=1,NBRX),I=1,NBRX)
        WRITE (6,200) (JB,I,A(I,NOVAR),I=1,NBRX)
        WRITE (6,210) A(NOVAR,NOVAR)
        RETURN
C       ***************************************************************
C       ***************************************************************
C       S/R  TO CALCULATE AND PRINT THE RESIDUAL SUM OF SQUARES AND C.P.
           ENTRY RESID
        IF (A(NBRXYW,NBRXYW)) 10,10,20
     10 WRITE (6,220) NPROB
        CALL EXIT
     20 DO 40 I=1,NOVAR
        DO 30 J=I,NOVAR
     30 A(I,J)=A(I,J)-(A(I,NBRXYW)*A(J,NBRXYW)/A(NBRXYW,NBRXYW))
     40 AVG(I)=A(I,NBRXYW)/A(NBRXYW,NBRXYW)
        IF (IFAVE.NE.0) GO TO 50
        WRITE (6,230)
        WRITE (6,190) ((JB,I,J,A(I,J),J=I,NBRX),I=1,NBRX)
        WRITE (6,200) (JB,I,A(I,NOVAR),I=1,NBRX)
        WRITE (6,210) A(NOVAR,NOVAR)
     50 RETURN
C       ***************************************************************
C       ***************************************************************
C       S/R  TO CALCULATE AND PRINT THE SIMPLE CORRELATION COEFFICIENTS
           ENTRY CORREL
        DO 90 I=1,NOVAR
        IF (A(I,I)) 60,60,80
     60 WRITE (6,240) I
        IF( I.EQ.NOVAR )  CALL SLITE(1)
        SIGMA(I)=1.0
        DO 70 J=1,NOVAR
        A(I,J)=0.0
     70 A(J,I)=0.0
```

```
      GO TO 90
   80 SIGMA(I)=DSQRT(A(I,I))
   90 A(I,I)=1.0D0
      DO 100 I=1,NBRX
      IP1=I+1
      DO 100 J=IP1,NOVAR
      A(I,J)=A(I,J)/(SIGMA(I)*SIGMA(J))
  100 A(J,I)=A(I,J)
      DO 110 J=1,NOVAR
      DO 110 K=1,NOVAR
  110 SIMCOR(J,K)=A(J,K)
      IF (IFCNST.NE.0) GO TO 140
      IF (IFCORR) 140,120,140
  120 WRITE (6,250)
      IF( NBRX.LE.1 )  GO TO 135
      NOVM2=NBRX-1
      DO 130 I=1,NOVM2
      IP1=I+1
  130 WRITE (6,260) (JB,I,J,A(I,J),J=IP1,NBRX)
  135 WRITE (6,270) (JB,I,A(I,NOVAR),I=1,NBRX)
  140 RETURN
C
C
C
  150 FORMAT (1H0,40X,18HSUM  OF  VARIABLES//)
  160 FORMAT (4(A1,11H     SUM X(I2,3H) =F14.4))
  170 FORMAT (6X,11HSUM     Y  =F14.4)
  180 FORMAT (1H070H                                         RAW  SUM  OF  SQUARES  A
     1ND  CROSS PRODUCTS//)
  190 FORMAT (3(A1,6H    X(I2,7H) VS X(I2,3H) =F17.6))
  200 FORMAT (3(A1,6H    X(I2,12H) VS    Y  =F17.6))
  210 FORMAT (5X,16HY      VS     Y  =F17.6)
  220 FORMAT (32H0 ZERO NUMBER OF DATA, PROBLEM  I6/)
  230 FORMAT (1H025X56HSUMS  OF  SQUARES  AND  CROSS PRODUCTS  ABOUT  TH
     1E  MEAN//)
  240 FORMAT (10H0 VARIABLEI5,13H IS CONSTANT //)
  250 FORMAT (1H0,33X,33HSIMPLE  CORRELATION  COEFFICIENTS//)
  260 FORMAT (3(A1,6H    X(I2,7H) VS X(I2,3H) =F12.8,5X))
  270 FORMAT (3(A1,6H    X(I2,12H) VS    Y  =F12.8,5X))
      END
```

```
C     S/R  TO ADD A VARIABLE
      SUBROUTINE ADDTO
      COMMON SIGMA(60),A(52,52),SIMCOR(52,52),AVG(60),TEST(60)
      COMMON POINT(60),STRING(3000),INDPAC(30,30),INDEXP(61)
      COMMON INDEX(60),NOUT(60),KSTEP(60),ALPHA(16),YMEAN,IDEN,IFAVE
      COMMON MAXSTP,IFPNCH,NSUMRY,NSKIP,NTAPE9,NEW
      COMMON NOVAR,NBRNOW,NOSTEP,NDATA,NBRXYW,NBRX,LPATH,DEFRM,K
      COMMON IFBACK,IFCNST,IFCORR,NPROB,NBRPVR,TOL,REMARK
      COMMON INDEXY,LBAD,NOGOOD(26)
      COMMON IFWT,YCONST,NYTRA,V(2,51),YTRA(2)
      COMMON  STDERR(50),CORSQR(50)
      DOUBLE PRECISION A,SIMCOR,SIGMA,AVG,TEST
C
      DOUBLE PRECISION DA,VAR,VMIN,VMAX
C          NBRNOW = NUMBER OF COEFFICIENTS FOR PRESENT  EQUATION
C          NBRPRV = NUMBER OF COEFFICIENTS FOR PREVIOUS EQUATION
C          INDEX  = INDEX OF PRESENT  EQUATION
C          INDEXP = INDEX OF PREVIOUS EQUATION
      DO 10 J=1,NBRNOW
   10 CALL PACK (NBRNOW,J,INDEX(J),2)
      DO 20 J=1,NBRX
   20 NOUT(J)=0
      IF (NBRNOW) 50,50,30
   30 DO 40 J=1,NBRNOW
      NDUMMY=INDEX(J)
   40 NOUT(NDUMMY)=1
   50 VMAX=-1.0
C          FIND LARGEST DELTA                               (VAR = DELTA)
      DO 70 I=1,NBRX
C                                    BYPASS IF ALREADY IN EQUATION
      IF (NOUT(I).NE.0) GO TO 70
      IF (A(I,I).GE.TOL) GO TO 60
      WRITE (6,510) A(I,I),I,(INDEX(J),J=1,NBRNOW)
      GO TO 70
C
   60 VAR=A(I,NOVAR)*A(NOVAR,I)/A(I,I)
      IF (VAR.LE.VMAX) GO TO 70
      VMAX=VAR
      K=I
   70 CONTINUE
C               HAVE FOUND OPTIMAL VARIABLE
      NSTEP=NOSTEP+1
      IF (VMAX) 80,90,90
   80 WRITE (6,520) VMAX
      CALL SLITE (1)
      GO TO 260
   90 NBRPVR=NBRNOW
      NBRNOW=NBRNOW+1
      IF (TEST(NBRNOW)-A(NOVAR,NOVAR)+VMAX) 100,100,120
  100 WRITE (6,530) K,NBRNOW,NSTEP
      DO 110 I=1,NOVAR
      DO 110 J=1,NOVAR
```

```
  110 A(I,J)=SIMCOR(I,J)
      LPATH=2
      GO TO 260
C          ADD VARIABLE TO INDEX
  120 CONTINUE
      IF (NBRPVR) 230,230,130
  130 DO 140 J=1,NBRPVR
  140 CALL PACK (NBRPVR,J,INDEXP(J),2)
      DO 150 J=1,NBRNOW
  150 CALL PACK (NBRNOW,J,INDEX(J),2)
      DO 180 J=1,NBRPVR
      IF (INDEXP(J)-K) 180,160,170
  160 CALL SLITE (1)
      WRITE (6,480)
      GO TO 260
  170 JJ=J
      GO TO 190
  180 CONTINUE
      INDEXP(NBRNOW)=K
      GO TO 210
  190 L=NBRNOW-JJ
      DO 200 J=1,L
      NR=NBRNOW+1-J
      NS=NBRNOW-J
  200 INDEXP(NR)=INDEXP(NS)
      INDEXP(JJ)=K
C          CHECK TO SEE IF SET HAS ALREADY BEEN COMPUTED
  210 DO 220 J=1,NBRNOW
      CALL PACK (NBRNOW,J,I,2)
      IF (INDEXP(J).NE.I) GO TO 240
  220 CONTINUE
      WRITE (6,490) NOSTEP,K,NBRNOW,(INDEX(J),J=1,NBRNOW)
      LPATH=3
      GO TO 260
C
  230 INDEXP(1)=K
  240 TEST(NBRNOW)=A(NOVAR,NOVAR)-VMAX
C          NEW SET - PUT INDEXES IN MATRIX
      DO 250 J=1,NBRNOW
  250 CALL PACK (NBRNOW,J,INDEXP(J),1)
      LPATH=1
      IF(MAXSTP.EQ.999) GO TO 255
      IF(NSTEP.EQ.1) GO TO 256
  255 WRITE(6,500) NSTEP,K, V(1,K),V(2,K)
  256 KSTEP(NBRNOW)=NSTEP
      KSTEP(NBRNOW)=NSTEP
  260 RETURN
C     ******************************************************************
C     ******************************************************************
C
C          ADJUST CORRELATION MATRIX FOR ENTRANCE OF VARIABLE K
          ENTRY MATRIX
```

```
C
      DA=1.0/A(K,K)
      DO 300 I=1,NOVAR
      IF (I-K) 270,300,270
  270 DO 290 J=1,NOVAR
      IF (J-K) 280,290,280
  280 A(I,J)=A(I,J)-A(I,K)*A(K,J)*DA
  290 CONTINUE
      A(I,K)=-A(I,K)*DA
  300 CONTINUE
      DO 320 J=1,NOVAR
      IF (J-K) 310,320,310
  310 A(K,J)=A(K,J)*DA
  320 CONTINUE
      A(K,K)=DA
C
      IF (NOSTEP.LE.1) NCOMB=MIN1(1000.,2.**(NOVAR-1))
      IF (NOSTEP.LE.NCOMB) GO TO 330
      WRITE (6,540) NPROB
      CALL SLITE (1)
  330 RETURN
C     ******************************************************************
C     ******************************************************************
C     S/R  TO DELETE A VARIABLE
           ENTRY REMOVE
C
      DO 340 J=1,NBRNOW
  340 CALL PACK (NBRNOW,J,INDEX(J),2)
      REMARK = 0.0
      K=0
      NSTEP=NOSTEP+1
      VMIN=-2.0E+30
C          FIND SMALLEST DELTA                                 (VAR = DELTA)
      DO 360 J=1,NBRNOW
      I=INDEX(J)
      IF (A(I,I)-TOL) 360,360,350
  350 VAR=A(I,NOVAR)*A(NOVAR,I)/A(I,I)
      IF (VAR.LE.VMIN) GO TO 360
      VMIN=VAR
      K=I
  360 CONTINUE
      IF (TEST(NBRNOW-1)+VMIN-A(NOVAR,NOVAR)) 370,370,380
  370 CONTINUE
      GO TO 470
C          REMOVE VARIABLE TO BE DELETED FROM INDEX
  380 JP = 0
      DO 390 J=1,NBRNOW
      IF( INDEX(J).EQ.K ) GO TO 390
      JP = JP + 1
      INDEXP(JP) = INDEX(J)
  390 CONTINUE
      NBRPVR = NBRNOW - 1
```

```
C          CHECK TO SEE IF SET HAS ALREADY BEEN COMPUTED
      DO 440 J=1,NBRPVR
      CALL PACK (NBRPVR,J,I,2)
      IF (I.NE.INDEXP(J)) GO TO 450
  440 CONTINUE
      WRITE (6,550) K
      GO TO 470
C          NEW SET - PUT INDEXES IN MATRIX
  450 REMARK = 1.0
      NBRNOW = NBRNOW - 1
      TEST(NBRNOW)=-VMIN+A(NOVAR,NOVAR)
      DO 460 J=1,NBRNOW
  460 CALL PACK (NBRNOW,J,INDEXP(J),1)
      WRITE(6,560) NSTEP,K, V(1,K),V(2,K)
      KSTEP(NBRNOW)=NSTEP
  470 RETURN
C
C
C
  500 FORMAT ( 8H0AT STEPI4,20H,  ADJOINED VARIABLEI3 , 3X2A6 )
  560 FORMAT ( 8H0AT STEPI4,20H,  DELETED  VARIABLEI3 , 3X2A6 )
  490 FORMAT ( 8H0AT STEPI4,20H,  ADJUNCTION OF VARI3,25H PRODUCES THE S
     *AME SET OFI3,21H VARIABLES AS BEFORE ,16I3/42I3)
  550 FORMAT ( 1H0,11X      ,20H   DELETION   OF VARI3,45H PRODUCES THE S
     *AME SET OF VARIABLES AS BEFORE)
  480 FORMAT (42H0ERROR IN ADJOIN S/R... PROBLEM TERMINATED)
  530 FORMAT (21H0 ADJOINING VARIABLE I4,30H  PRODUCES NO IMPROVEMENT FO
     1R I3,21H  VARIABLES, AT STEP I4)
  510 FORMAT (38H0NORMALIZED RESIDUAL SUM OF SQUARES ISF11.6,17H .......
     1.VARIABLEI3,31H IS NEAR-DEPENDENT ON VARIABLES11I3,/2X40I3)
  540 FORMAT (49H0MAXIMUM NUMBER OF ITERATIONS EXCEEDED IN PROBLEMI5)
  520 FORMAT (14H0NEGATIVE VMAX F10.6, 25H ..... PROBLEM TERMINATED)
      END
```

```
C        PACK AND EXTRACT INDEX FROM ARRAY NN WHICH HAS 4 VALUES PER WD
C
      SUBROUTINE PACK (II,JJ,JVALUE,NGO)
      COMMON SIGMA(60),A(52,52),SIMCOR(52,52),AVG(60),TEST(60)
      COMMON POINT(60),STRING(3000),INDPAC(30,30),INDEXP(61)
      COMMON INDEX(60),NOUT(60),KSTEP(60),ALPHA(16),YMEAN,IDEN,IFAVE
      COMMON MAXSTP,IFPNCH,NSUMRY,NSKIP,NTAPE9,NEW
      COMMON NOVAR,NBRNOW,NOSTEP,NDATA,NBRXYW,NBRX,LPATH,DEFRM,K
      COMMON IFBACK,IFCNST,IFCORR,NPROB,NBRPVR,TOL,REMARK
      COMMON INDEXY,LBAD,NOGOOD(28)
      COMMON IFW1,YCONST,NYTRA,V(2,51),YTRA(2)
      COMMON  STDERR(50),CORSQR(50)
      DOUBLE PRECISION A,SIMCOR,SIGMA,AVG,TEST
      EQUIVALENCE (M,XM)
      DIMENSION MASK(4), MASKC(4), MASKS(4)
      DATA MASK/077,07700,0770000,077000000/
      DATA MASKC/0777777777700,0777777770077,0777777007777,0777700777777
     1/
      DATA MASKS/01,0100,010000,01000000/
C
C        INDEX IN NN  ARRAY
      I=(II+1)/2
      J=(JJ+1)/2
C        WORD TO BE PACKED
      M=INDPAC(I,J)
C        POSITION IN M TO BE PACKED ( 1-4 )
      NW=2*MOD(II,2)+MOD(JJ,2)+1
C
      GO TO (10,20), NGO
C        PACK INDEX
   10 XM=AND(M,MASKC(NW))
      XM=OR(M,JVALUE*MASKS(NW))
      INDPAC(I,J)=M
      RETURN
C        EXTRACT  INDEX
   20 XM=AND(M,MASK(NW))
      JVALUE=M/MASKS(NW)
      RETURN
C
      END
```

```
C     TRANSFORMATION OF DATA
      SUBROUTINE CHANGE (POINT,NBRTRA,CONST,NTOTAL)
C
      DIMENSION POINT(60), NBRTRA(60), CONST(60)
C
      DO 220 I=1,NTOTAL
      ITRA=NBRTRA(I)
      IF (ITRA.LE.0) GO TO 220
      IF (ITRA.LT.17) GO TO 10
      WRITE (6,230) ITRA
      NBRTRA(I)=0
      GO TO 220
C
   10 D=POINT(I)
      C=CONST(I)
      IF (ITRA.GT.8) GO TO 120
      GO TO (20,30,40,50,60,80,100,110), ITRA
   20 D=D+C
      GO TO 210
   30 D=D*C
      GO TO 210
   40 D=D/C
      GO TO 210
   50 D=C/D
      GO TO 210
   60 IF (C.LT.0.0) GO TO 70
      D=D**C
      GO TO 210
   70 D=1./D**(-C)
      GO TO 210
   80 IF (D.LT.0.0) GO TO 90
      D=C**D
      GO TO 210
   90 D=1./C**(-D)
      GO TO 210
  100 D=ALOG(D+C)
      GO TO 210
  110 D=ALOG10(D+C)
      GO TO 210
C
  120 ITRA=ITRA-8
      IF (C.EQ.0.0) C=1.
      GO TO (130,140,150,160,170,180,190,200), ITRA
  130 D=EXP(C*D)
      GO TO 210
  140 D=EXP(C/D)
      GO TO 210
  150 D=SIN(C*D)
      GO TO 210
  160 D=COS(C*D)
      GO TO 210
  170 D=SIN(C*D)/COS(C*D)
```

```
      GO TO 210
  180 D=SINH(C*D)
      GO TO 210
  190 D=COSH(C*D)
      GO TO 210
  200 D=TANH(C*D)
  210 POINT(I)=D
  220 CONTINUE
      RETURN
C
C
C
  230 FORMAT (/1X,14HTRANSFORMATIONI3,18H IS NOT IN TABLES,36H IT WILL B
     1E SET TO ZERO AND IGNORED.//)
      END
```

```
C          DUMMY SUBROUTINE TO MAKE UNUSUAL TYPES OF TRANSFORMATIONS.
C          THIS SUBROUTINE IS REPLACED AT OBJECT TIME WITH ONE WHICH
C          PERFORMS THE DESIRED TRANSFORMATIONS.
C          POINT= DATA OBSERVATION WHICH WAS READ IN ON CARDS OR TAPE
C          IFSUB = A NUMBER GREATER THAN ZERO WHICH CALLS S/R EQUAT AND
C                  NBROFX.  MAY BE USED AS A BRANCH INDICATOR.
C          NXV = NEW NUMBER OF INDEPENDENT VARIABLES
C          MAKE SURE DEPENDENT VARIABLE MATCHES  NXV + INDEXY.
C
C          S/R EQUAT IS USED TO ADD OR CHANGE DATA OBSERVATIONS
C
C          S/R  NBROFX   IS USED TO CHANGE THE VALUE OF NXV.
C                                                  AND MUST BE USED.
C     ***********************************************************************
C
C          EXAMPLE  -  POLYNOMIAL EQUATION     IFSUB = POWER OF X
C            Y = A0 + A1*X + A2*X**2 + A3*X**3 + ... + A(IFSUB)*X**IFSUB
C
      SUBROUTINE NBROFX (IFSUB,NXV)
      NXV=IFSUB
      NY=NXV+1
      NX=NXV
      RETURN
C
         ENTRY EQUAT(IFSUB,POINT)
      DIMENSION POINT(52)
C          STORE Y
      K=54-NX
      L=53
      DO 10 J=NY,52
      K=K-1
      L=L-1
   10 POINT(L)=POINT(K)
C
C          STORE POWERS OF X
      DO 20 J=2,NX
   20 POINT(J)=POINT(J-1)*POINT(1)
      RETURN
      END
```

```
SUBROUTINE  DATE(X,Y)
DATA B/1H          /
X=B
Y=B
RETURN
END
```

APPENDIX F
Cushioning Model Validation Code

This code provides the calculations necessary for a specific model validation as outlined in the Model Validation Section. Input to this program is generated by the Data Analysis program.

```
C***********************************************************************
C
C                    VALIDATION OF CUSHION MATERIAL MODELS
C
C***********************************************************************
C
      INTEGER TYPEM
      COMMON/STD/IL,TYPEM(3,20),NV(20),CONST(20),COEF(51,20)
      COMMON /STD1/TP,TC,SS,GL,NVR,V(51)
      COMMON /STD2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
C
      RANGE=1.0
C
C     FILE 12 CONTAINS THE SPECIFIC MATERIAL DATA
      REWIND 12
C
C
100   CONTINUE
C
C     READ SPECIAL CASE FLAG AND MATERIAL TYPE NUMBER
C
      READ(5,2200) IS,IL
C
C     NUMBER OF COEFFICIENTS FOR THE MODEL
C
      NA =NV(IL)
C
C     SPECIAL CASE IS = 0  ALL DATA CASES IS = 1
C
300   IF (IS.EQ.1) GO TO 400
C
      READ (5,2100,END=1500) TC,DH,TP
400   CONTINUE
      READ (12) NPTS
      IF (NPTS.EQ.99999) GO TO 1500
      NPTA=NPTS+1
      READ (12) DH ,TP ,TC  ,(IA(II),II=1,2),((B(KK,JJ),KK=1,5),JJ=1,4),
     1(X(LL),LL=2,NPTA),(Y(LL),LL=2,NPTA)
      IF (IS.EQ.1) GO TO 500
500   PRINT 2300, TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL),DH,TC,TP
      X(1)=NPTS
      Y(1)=NPTS
C***********************************************************************
      CALL CONFID (X,Y,NPTS,YL,YU,B)
C***********************************************************************
      DELTA=0.05
      X(2)=0.05
      I=2
      N=2
600   CONTINUE
      IF (X(I).GT.5.2) GO TO 700
      IF(X(I).GE..95)DELTA=0.2
```

```
C
C       SECOND ORDER POLYNOMIAL
C
        Y(I)=B(1,N)+B(2,N)*(ALOG(X(I)*100.))
        I=I+1
        X(I)=X(I-1)+DELTA
        GO TO 600
700     CONTINUE
        X(1)=I-2
        Y(1)=I-2
800     CONTINUE
        K=I-1
        DO 900 L=2,K
        Y(L)=Y(L)+B(3,N)*(ALOG(X(L)*100.))**2
C
C       GENERAL CUSHION MATERIAL MODEL
C
        SS = X(L)
C***********************************************************************
        CALL MODEL
C***********************************************************************
        YM(L) =GL
900     CONTINUE
C
C       CALCULATE PREDICTION LIMITS
C
        NPTA=X(1)
C***********************************************************************
        CALL PREDIC (X,Y,NPTA,YL,YU,B,YPL,YPU)
C***********************************************************************
C
C       FIND MINIMUM IDCC G-LEVEL
C
        YMIN=1000.0
        DO 1200 I=2,K
        IF (YMIN.LE.Y(I)) GO TO 1200
        YMIN=Y(I)
        M=I
1200    CONTINUE
C
C       DETERMINE VALID MODEL RANGE FROM BOUNDED IDCC
C         AND PREDICTION LIMITS.
C
        XMIN=X(M)
        XL=XMIN-RANGE
        XU=XMIN+RANGE
        WRITE (6,2500)
        WRITE (6,2600)
        DO 1400 I=2,K
        N=3H
        IB=3H
        IF(X(I).GE.XL.AND.X(I).LE.XU)IB=3H **
```

```
      IF(YPL(I).GT.YM(I).OR.YPU(I).LT.YM(I))N=3H  *
      IF (YPL(I).LE.0.0) GO TO 1300
      WRITE (6,2700) X(I),Y(I),YPL(I),IB,YM(I),N,YPU(I)
      GO TO 1400
C     NEGATIVE G-VALUES SET TO - - .
1300  WRITE (6,2800) X(I),Y(I),IB,YM(I),N,YPU(I)
1400  CONTINUE
C
C     NEXT CASE
      GO TO 300
C
C     END OF JOB
1500  WRITE (6,2900)
      CALL EXIT
C
C
2000  FORMAT(1H1)
2100  FORMAT(2F5.1,F7.1)
2200  FORMAT(I1,I2)
2300  FORMAT (1H1,10X,3A4,4X,F4.1,' IN. D.H. ',F7.1,' IN. THICK',F8.1,'
     1TEMPERATURE')
2400  FORMAT ()
2500  FORMAT (///16X,'STATIC STRESS',17X,'DECELERATION  (G)')
2600  FORMAT (21X,'PSI',12X,'IDCC',4X,'LOWER-P',9X,'MODEL',6X,'UPPER-P')
2700  FORMAT(18X,F6.2,9X,F7.2,4X,F7.2,4X,A3,F7.2,A3,3X,F7.2)
2800  FORMAT(18X,F6.2,9X,F7.2,7X,'- -',5X,A3,F7.2,A3,3X,F7.2)
2900  FORMAT (1H1,'     END OF JOB')
      END
```

```
      SUBROUTINE CONFID (X,Y,NPTS,YL,YU,B)
C***********************************************************************
C
C        COMPUTE THE PREDICTION LIMITS OF THE CURVES
C
C***********************************************************************
      DIMENSION X(500), Y(500), YM(500),  YPL(500),
     1 YPU(500),B(5,4),IA(2),YL(500),YU(500)
      DIMENSION XAR(500,3), YAR(500), C(3), A(3,3), XIN(3,3), E(3)
      TAH = 1.66
      F2=0.0
      BX=0.0
      YS=0.0
      NPTS=X(1)
      DO 300 I=1,3
      C(I)=0.0
      E(I)=0.0
      DO 100 L=1,3
      XIN(I,L)=0.0
      A(I,L)=0.0
100   CONTINUE
      DO 200 J=1,NPTS
      XAR(J,I)=0.0
      YAR(J)=0.0
200   CONTINUE
300   CONTINUE
      F=0.0
      S=0.0
      SSQ=0.0
      DO 400 I=1,NPTS
      J=I+1
      XAR(I,1)=1.0
      XAR(I,2)=ALOG(X(J)*100.)
      XAR(I,3)=XAR(I,2)**2
      YAR(I)=Y(J)
400   CONTINUE
      DO 600 I=1,3
      DO 500 J=1,NPTS
      C(I)=C(I)+XAR(J,I)*YAR(J)
500   CONTINUE
      BX=BX+C(I)*B(I,2)
600   CONTINUE
      DO 700 J=1,NPTS
      YS=YS+YAR(J)**2
700   CONTINUE
2600  FORMAT(3X,'YS, BX    ',2F15.4        )
      SSQ=(YS-BX)/(X(1)-3.0)
      S=SQRT(SSQ)
      DO 800 J=1,NPTS
      A(1,1)=A(1,1)+XAR(J,1)*XAR(J,1)
      A(1,2)=A(1,2)+XAR(J,1)*XAR(J,2)
      A(1,3)=A(1,3)+XAR(J,1)*XAR(J,3)
```

```
      A(2,1)=A(2,1)+XAR(J,2)*XAR(J,1)
      A(2,2)=A(2,2)+XAR(J,2)*XAR(J,2)
      A(2,3)=A(2,3)+XAR(J,2)*XAR(J,3)
      A(3,1)=A(3,1)+XAR(J,3)*XAR(J,1)
      A(3,2)=A(3,2)+XAR(J,3)*XAR(J,2)
      A(3,3)=A(3,3)+XAR(J,3)*XAR(J,3)
800   CONTINUE
      D=A(1,1)*(A(2,2)*A(3,3)-A(3,2)*A(2,3))+A(1,2)*(A(3,1)*A(2,3)-A(2,1
     1)*A(3,3))+A(1,3)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
      XIN(1,1)=(A(2,2)*A(3,3)-A(3,2)*A(2,3))/D
      XIN(1,2)=(A(3,2)*A(1,3)-A(1,2)*A(3,3))/D
      XIN(1,3)=(A(1,2)*A(2,3)-A(2,2)*A(1,3))/D
      XIN(2,1)=(A(3,1)*A(2,3)-A(2,1)*A(3,3))/D
      XIN(2,2)=(A(1,1)*A(3,3)-A(3,1)*A(1,3))/D
      XIN(2,3)=(A(2,1)*A(1,3)-A(1,1)*A(2,3))/D
      XIN(3,1)=(A(2,1)*A(3,2)-A(3,1)*A(2,2))/D
      XIN(3,2)=(A(3,1)*A(1,2)-A(1,1)*A(3,2))/D
      XIN(3,3)=(A(1,1)*A(2,2)-A(2,1)*A(1,2))/D
      RETURN
C***********************************************************************
       ENTRY PREDIC (X,Y,NPTS,YL,YU,B,YPL,YPU)
C***********************************************************************
      DO 900 I=1,NPTS
      XAR(I,1)=1.
      J=I+1
      XAR(I,2)=X(J)
      XAR(I,3)=XAR(I,2)**2
900   CONTINUE
      YPL(1)=NPTS
      YPU(1)=NPTS
      YU(1)=NPTS
      YL(1)=NPTS
      DO 1400 J=1,NPTS
      DO 1100 I=1,3
      DO 1000 K=1,3
      E(I)=E(I)+XAR(J,K)*XIN(K,I)
1000  CONTINUE
1100  CONTINUE
      DO 1200 I=1,3
      F=F+E(I)*XAR(J,I)
1200  CONTINUE
      F2=1.0+F
      F2=SQRT(F2)
      F=SQRT(F)
      N=J+1
      YU(N)=Y(N)+TAH*S*F
      YL(N)=Y(N)-TAH*S*F
      YPU(N)=Y(N)+TAH*S*F2
      YPL(N)=Y(N)-TAH*S*F2
      F=0.0
      DO 1300 I=1,3
      E(I)=0.0
```

```
1300  CONTINUE
1400  CONTINUE
      RETURN
2000  FORMAT(3X,'B',5F15.4)
2100  FORMAT(3X,'SSQ, S, F, F2',3X,4F15.4)
      END
```

```
      SUBROUTINE MODEL
C*****************************************************************************
C********* DYNAMIC CUSHIONING MODEL ***************************************
C*****************************************************************************
      INTEGER TYPEM
      COMMON /STD1/TP,TC,SS,GL,NVR,V(51)
      COMMON /STD2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
      COMMON/STD/IL,TYPEM(3,20),NV(20),CONST(20),COEF(51,20)
C
      SS100 = SS * 100.
      AL = ALOG( SS100 )
      AL2 = AL * AL
      SRDH = SQRT( DH )
      TR  =(TP + 460.0)/100.
      TR2 = TR * TR
      TR3 = TR * TR2
      TR4 = TR3 * TR
      TCOH = TC ** (-0.5)
      TCTH = TC ** (-1.5)
C
      V(01) = TR    * TCOH
      V(02) = TR    * TCOH              * AL
      V(03) = TR    * TCOH              * AL2
      V(04) = TR    * TCTH    * SRDH
      V(05) = TR    * TCTH    * SRDH    * AL
      V(06) = TR    * TCTH    * SRDH    * AL2
      V(07) = TR    * TCOH    * SRDH
      V(08) = TR    * TCOH    * SRDH    * AL
      V(09) = TR    * TCOH    * SRDH    * AL2
      V(10) = TR2   * TCOH
      V(11) = TR2   * TCOH              * AL
      V(12) = TR2   * TCOH              * AL2
      V(13) = TR2   * TCTH    * SRDH
      V(14) = TR2   * TCTH    * SRDH    * AL
      V(15) = TR2   * TCTH    * SRDH    * AL2
      V(16) = TR2   * TCOH    * SRDH
      V(17) = TR2   * TCOH    * SRDH    * AL
      V(18) = TR2   * TCOH    * SRDH    * AL2
      V(19) = TR3   * TCOH
      V(20) = TR3   * TCOH              * AL
      V(21) = TR3   * TCOH              * AL2
      V(22) = TR3   * TCTH    * SRDH
      V(23) = TR3   * TCTH    * SRDH    * AL
      V(24) = TR3   * TCTH    * SRDH    * AL2
      V(25) = TR3   * TCOH    * SRDH
      V(26) = TR3   * TCOH    * SRDH    * AL
      V(27) = TR3   * TCOH    * SRDH    * AL2
      V(28) = TR    * TCTH
      V(29) = TR    * TCTH              * AL
      V(30) = TR    * TCTH              * AL2
      V(31) = TR2   * TCTH
      V(32) = TR2   * TCTH              * AL
```

```
      V(33) = TR2    * TCTH                  * AL2
      V(34) = TR3    * TCTH
      V(35) = TR3    * TCTH                  * AL
      V(36) = TR3    * TCTH                  * AL2
      V(37) = TR
      V(38) = TR * AL
      V(39) = TR * AL2
      V(40) = TR2
      V(41) = TR2 * AL
      V(42) = TR2 * AL2
      V(43) = TR3
      V(44) = TR3 * AL
      V(45) = TR3 * AL2
C
C         COMPUTE DYNAMIC CUSHIONING FUNCTION
C
      GL = CONST(IL)
      NA = NV(IL)
      DO 100 J=1,NA
  100 GL = GL +  COEF(J,IL) * V(J)
      RETURN
C
      END
```

```
      BLOCK DATA
C*****************************************************************************
C        MODEL COEFICIENTS
C*****************************************************************************
      INTEGER TYPEM
      COMMON /STD1/TP,TC,SS,GL,NVR,V(51)
      COMMON /STD2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
      COMMON/STD/IL,TYPEM(3,20),NV(20),CONST(20),COEF(51,20)
      DATA (NV(I),I=1,11)/ 4*36,7*45/
      DATA (TYPEM(1,I),I=1,5)/4HMINI ,4HETHA,4HETHA,4HURET,4HURES/
      DATA (TYPEM(2,I),I=1,5)/4HCEL ,4HFOAM,4HFOAM,4HHER-,4HTER-/
      DATA (TYPEM(3,I),I=1,5)/4H    ,4H-2  ,4H-4  ,4H3   ,4H4    /
      DATA(TYPEM(1,I),I=6,11)/ 4HURES,4HMIN+,4HETHA,4HMIN+,4HETHA,
     1    4HETHA/
      DATA(TYPEM(2,I),I=6,11)/ 4HTER4,4HURES,4H4+MI,4HETHA,4H2+ET,
     1    4H4+ET/
      DATA(TYPEM(3,I),I=6,11)/ 4H+MIN,3HTER,1HN,1H4,3HHA4,3HHA2/
      DATA (CONST(I),I=1,5)/-8.3931602,22.673363,32.918823,16.456637,
     1                       559.746/
      DATA (CONST(I),I=6,11)/ 176.96306,2204.5884,338.68941,
     1    495.97272,101.84126,364.8831 /
      DATA ((COEF(I,J),I=1,51),J=1,4)/
     0            2*0.,3.5419457,0.,-15.318724,3.340187,207.49326,
     1 -50.359563,1.4344456,0.,-6.6958791,4*0.,-54.656687,11.621323,0.,
     2 -1.3016393,2.0789886,-.226642,-.40141035,.61173036,-.0953190,
     3 3.9422841,-.8660377,0.,-233.77506,28.303458,0.,49.67875,26.32324,
     4 -6.0678372,0.,-6.1520847,1.0752888,15*0.,3*0.,-38.759499,0.,
     5 1.2835062,202.39926,-53.355374,2.9700805,3*0.,5.9886026,2*0.,
     6 -45.75217,8.1253357,3*0.,-.0096247456,2*0.,-.02080280 ,2.2706984,
     7  0.,-.067325251,-418.94060,156.40775,-15.003939,133.99966,
     8 -38.92069,2.2685693,-10.473401,2.1412173,16*0.,-48.167497,3*0.,
     9 -41.636209,6.0138922,216.47081,-33.44635,2*0.,4.339839,2*0.,
     A 9.4729454,-1.0063034,-51.761093,3.6887819,.55314362,2*0.,
     B -.10103962,-.21961366,-.33423026,0.,3.2311604,0.,-.054746082,
     C -253.15131,136.46772,-11.86598,72.15674,-25.154675,0.,-4.1437095,
     D 0.,.357168,17*0.,13.28693,-8.6616181,2*0.,63.974659,-9.0835167,
     E-4.6111397,0.,3.474952,-5.9426191,.40180109,1.8227191,0.,
     F -18.556063,0.,2.2398188,-.71661025,0.,.49953221,0.,-.15167925,
     G-.015276062,1.6300623,0.,-.19795351,21.394161,-18.657059,3*0.,
     H .40841278,2*0.,16*0./
      DATA ((COEF(I,J),I=1,51),J=5,5)/3*0.,-107.90113,32.951647,
     1 -3.7757261,206.02943,-49.889708,-5.0392425,3*0.,19.77191 ,2*0.,
     3 -56.422968,7.8434679,3.1828843,4*0.,-1.0933512,.14231548,
     4 3.7496497,0.,-.36983233,2*0.,10.179855,48.384712,-31.619216,0.,
     5 -9.0192091,5.953565,-.43214979,-213.634,0.,8.2119242,0.,14.258078
     6 ,-5.0076278,3.5699545,-2.4343742,.59891583,6*0./
      DATA((COEF(I,J),I=1,51),J= 6, 6)/
     1    23*0.0,0.010282159,13*0.0,-13.903136,3*0.0,0.26984519,9*0.0/
      DATA((COEF(I,J),I=1,51),J= 7, 7)/
     1    3*0.0,73.252522,-39.010025,-4.5392164,5.1754836,-1.3005991,
     A    6*0.0,
     2 3.6620029,6*0.0,-0.48309006,0.0,-0.29548739,4*0.0,118.01099,
```

```
3 0.0,-83.691243,0.0,-5.8215625,9.3710246,0.0,0.39998398,
4 -725.38749,-117.32628,3.4598770,58.689010,36.498087,-0.58720109,
5   0.0,-2.7682574,7*0.0/
 DATA((COEF(I,J),I=1,51),J= 8, 8)/
1    6*0.0,32.614392,-5.0961467,5*0.0,-3.6993450,0.63438176,
2 -4.312947,0.58519131,4*0.0,0.85493419,9*0.0,10.660799,
3 -1.8236532,-2.4597742,3*0.0,-53.449983,3.1497049,-8.8143942,
4 8.6115005,3*0.0,-0.078832109,6*0.0/
 DATA((COEF(I,J),I=1,51),J= 9, 9)/
1    3*0.0,30.352322,0.0,0.0,7.2316497,6*0.0,-3.8097934,9*0.0,
2    0.11994912,0.0,-0.055827933,0.0,-88.869467,3*0.0,11.080006,
3    3*0.0,-0.34449941,0.0,-63.71083,1.4256175,-19.209793,
4    10.554163 ,0.68592748,0.97094918,0.0,-0.14904712,6*0.0/
 DATA((COEF(I,J),I=1,51),J=10,10)/
1    -99.352813,21.354231,0.0,0.0,-1.4437391,0.0,15.056152,
2    -5.1632315,0.38916001,0.0,0.0,-0.33239932,4.351482,
3    6*0.0,0.22303519,0.0,0.0,-0.59521806,0.1135047,10*0.0,
4    1.3420541,-0.30672112,291.2326,-58.921384,0.0,-71.524129,
5    8.5114836,0.97569838,4.1622124,0.0,-0.14793199,6*0.0/
 DATA((COEF(I,J),I=1,51),J=11,11)/
1    6*0.0,41.173081,-13.052459,1.0084125, -67.963627,14.493859,
2    -1.1099463,3*0.0,-3.2333637,0.54214573,0.0,7.2069221,
A    -0.599362001,0.0,0.0,
3    -0.19269788,0.056610506,6*0.0,22.870867,3*0.0,-1.0732096,0.0,
4    70.269319,-47.453983,0.0,-15.995790,5.7049793,0.88065089,
5    0.0,0.0,-0.11323733,6*0.0/
C
 END
```

APPENDIX G

CUSHOP and ENCAP Model Codes

This code considers eleven different materials and is utilized by the Container Cushion Design Engineer System. Further documentation pertaining to this code is contained in [9].

```
C       ***************************************************************
C       *  CONTAINER CUSHIONING DESIGN ENGINEER                       *
C       *  CUSHOP AND ENCAP MODELS                                    *
C       *                                                             *
C       *                                                             *
C       *         MATERIAL TYPES                                      *
C       *            MINICEL                                          *
C       *            ETHAFOAM 2                                       *
C       *            ETHAFOAM 4                                       *
C       *            URETHER 3                                        *
C       *            URESTER 4                                        *
C       *            URESTER 4 + MINICEL                              *
C       *            MINICEL  + URESTER 4                             *
C       *            ETHAFOAM 4 + MINICEL                             *
C       *            MINICEL  + ETHAFOAM 4                            *
C       *            ETHAFOAM 2 + ETHAFOAM 4                          *
C       *            ETHAFOAM 4 + ETHAFOAM 2                          *
C       *                                                             *
C       ***************************************************************
        COMMON /STD1/TP,TC,SS,GL,NVR,V(51)
        COMMON /STD2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
        DIMENSION T(5)
        EQUIVALENCE (T(1),CH)
100     CONTINUE
C       INITIALIZATION
        READ(5, *)  MODEL,DH,GLMAX,CH,CA,CC,W,BMIN,BMAX
        IF(MODEL .EQ. 1) GOTO 200
        IF(MODEL .EQ. 2) GOTO 300
        IF(MODEL .EQ. 99) GOTO 400
        PRINT 500,MODEL
        GOTO 100
200     CALL CUSHOP
        GOTO 100
300     CALL ENCAP
        GOTO 100
400     PRINT 600
500     FORMAT(1H1,10X,'ERROR IN MODEL REQUEST-- MUST BE A 1,2,OR A 99')
600     FORMAT(1H1,10X,'END OF JOB')
        STOP
        END
```

```
      SUBROUTINE CUSHOP
C
C****************************************************************************
C               ' C U S H O P '
C     CUSHION MATERIAL THICKNESS OPTIMIZATION PROGRAM
C
C****************************************************************************
C
      INTEGER TYPEM
      COMMON /STU1/ TP,TC,SS,GL,NVR,V(51)
      COMMON /STU2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
      COMMON /STU/ IL,TYPEM(3,20),NV(20),CONST(20),COEF(51,20)
      DIMENSION T(5), TCC(3), YL(3), YU(3), TCMAX(11), YM(3)
      EQUIVALENCE (T(1),CH)
      DATA TCMAX/,4.0,4.0,4.0/
C
C      INITIALIZATION
      DATA TCMAX/4.0,4.0,4.0,4.0,4.0,6.0,6.0,6.0,6.0,6.0,6.0/
C      DSS = DELTA STATIC STRESS       SSMIN = MIN STATIC STRESS
C      SSMAX = MAX STATIC STRESS       DTC = DELTA MATERIAL THICKNESS
C      TCMIN = MIN MATERIAL THICKNESS  TCMAX = MAX MATERIAL THICKNESS
C
      DATA DSS,SSMIN,SSMAX,DTC,TCMIN/0.05,0.0,5.0,0.5,0.5/
C
      DO 1200 IL=1,11
      PRINT 1300
      TC = TCMIN
      TKMAX=TCMAX(IL)+0.5
      IF (T(3).LT.-20.0.AND.IL.GE.4) GO TO 1000
C
C          HAS MAXIMUM THICKNESS BEEN REACHED ;
C
100   IF (TC.GE.TKMAX) GO TO 800
C
C         INCREMENT THICKNESS
C
200   TC=TC+DTC
      DO 300 J=1,3
      SSU(J)=0.0
      TCC(J)=0.0
300   SSL(J)=0.0
C
C          INITIALIZE TEMPERATURE INDEX
C
      N=0
C
C          HAS THIRD TEMPERATURE BEEN REACHED ;
C
400   IF (N.GE.3) GO TO 800
C
C         INCREMENT TEMPERATURE
C
```

```
      N=N+1
      TP=T(N)
C
C          INITIALIZE STATIC STRESS
C
      SS=SSMIN
C
C          HAS MAXIMUM STATIC STRESS BEEN REACHED ;
C
500   IF (SS.GT.SSMAX) GO TO 100
C
C         INCREMENT STATIC STRESS
C
      SS=SS+DSS
C*********************************************************************
      CALL MODEL
C*********************************************************************
      IF (GL.GT.GLMAX) GO TO 500
      SSL(N)=SS
      TCC(N)=TC
      YL(N)=GL
C
C          HAS MAXIMUM STATIC STRESS BEEN REACHED ;
C
600   IF (SS.GT.SSMAX) GO TO 700
C
C         INCREMENT STATIC STRESS
C
      SS=SS+DSS
C*********************************************************************
      CALL MODEL
C*********************************************************************
      IF (GL.LT.GLMAX) GO TO 600
      IF(GL.GT.GLMAX)SS=SS-DSS
700   SSU(N)=SS
      YM(N)=GL
      YU(N)=GL
C
C          MAKE RANGE TEST
C
      TEST=SSU(N)-SSL(N)
      IF (TEST.GE.0.2) GO TO 400
      GO TO 100
800   CONTINUE
      IF (SSL(3).EQ.0.0) GO TO 900
      TEST=SSU(1)-SSL(3)
      IF (TEST.GE.0.2) GO TO 1100
900   CONTINUE
      IF (TC.LT.TKMAX) GO TO 200
      PRINT 1600, TKMAX,TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL),SSL(3),
     1 SSU(1),GLMAX,T(3),YM(3),T(2),YM(2),T(1),YM(1)
      GO TO 1200
```

```
1000     IF( IL .GE. 8) GOTO 100
      PRINT 1500, TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL)
      GO TO 1200
1100  CONTINUE
      BMIN=W/SSU(1)
      BMAX=W/SSL(3)
      PRINT 1400, TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL),SSL(3),SSU(1),DH,
     1GLMAX,TC,T(3),T(2),T(1),W,BMIN,BMAX
C
C     PRINT TABLE OF STATIC STRESS VERSUS G-LEVELS FOR THE THREE
C     TEMPERATURES  --COLD , AMBIENT AND HOT
C
      CALL TABLE
C
1200  CONTINUE
C
      RETURN
C
C
1300  FORMAT (1H1,T23,'CUSHION MATERIAL OPTIMIZATION',/,T33,'CUSHOP',/)
1400  FORMAT (T32,3A4,/,8X,'LOWER SS ',F5.2,2X,'UPPER SS ',F5.2,2X,'DROP
     1 HEIGHT',F5.1,2X,'G-LEVEL',F6.1,/,8X,'MATERIAL',' THICKNESS',F8.2,
     2/,8X,'TEMPERATURES',3X,3F12.2,/,8X,'WEIGHT',F8.2,2X,'MIN. BEARING
     3AREA',F8.2,2X,'MAX. BEARING AREA',F8.2,/)
1500  FORMAT (//,5X,'INVALID LOWER TEMPERATURE FOR  ',3A4)
1600  FORMAT (//,5X,'***BEYOND MAXIMUM THICKNESS (',F6.2,'  )  OF  ',3A4
     1,/,6X,'LOWER SS ',F5.3,2X,'UPPER SS ',F5.3,2X,'DROP HEIGHT',F6.1,2
     2X,'G-LEVEL',F6.1,/,10X,'HOT      TEMPERATURE',3X,F8.3,3X,'G-LEVEL',
     33X,F8.3,/,10X,'AMBIENT TEMPERATURE',3X,F8.3,3X,'G-LEVEL',3X,F8.3,/
     4,10X,'COLD     TEMPERATURE',3X,F8.3,3X,'G-LEVEL',3X,F8.3,//)
      END
```

```
      SUBROUTINE ENCAP
C*************************************************************************
C
C                    ' E N C A P '
C          ENCAPSULATION CUSHION   MATERIAL OPTIMIZATION PROGRAM
C
C      DETERMINE MATERIAL THICKNESS FOR GIVEN
C      STATIC STRESS DROP HEIGHT AND TEMPERATURES
C*************************************************************************
C
      INTEGER TYPEM
      COMMON /STD1/ TP,TC,SS,GL,NVR,V(51)
      COMMON /STD2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
      COMMON /STD/ IL,TYPEM(3,20),NV(20),CONST(20),COEF(51,20)
      DIMENSION TCMAX(5), YM(3), T(3)
      EQUIVALENCE (T(1),CH), (TC,THCK)
      DATA TCMAX/4.0,4.0,4.0,4.0,4.0/
C
C       INITIALIZATION
C
C       DSS = DELTA STATIC STRESS         SSMIN = MIN STATIC STRESS
C       SSMAX = MAX STATIC STRESS         DTC = DELTA MATERIAL THICKNESS
C       TCMIN = MIN MATERIAL THICKNESS    TCMAX = MAX MATERIAL THICKNESS
C
      DATA DSS,SSMIN,SSMAX,DTC,TCMIN/0.05,0.0,5.0,0.5,0.5/
C
      SU=W/BMIN
      SL=W/BMAX
C
100   DO 600 IL=1,5
      PRINT 1000
      TKMAX=TCMAX(IL)+8.5
      TC = TCMIN
200   TC=TC+0.5
      IF (TC.GT.TKMAX) GO TO 500
      YM(1)=0.0
      YM(2)=0.0
      YM(3)=0.0
      SS=SL
      IF (T(3).LT.-20.0.AND.IL.GE.4) GO TO 400
      I=3
300   CONTINUE
      TP=T(I)
C*************************************************************************
      CALL MODEL
C*************************************************************************
      YM(I)=GL
      IF (GL.GT.GLMAX) GO TO 200
      SS=SU
      I=I-1
      IF(I .GE. 1) GOTO 300
      PRINT 700, TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL),SL,SU,DH,GLMAX,TC,
```

```
     1T(3),T(2),T(1),W,BMIN,BMAX
C*************************************************************************
      CALL TABLE
C*************************************************************************
      GO TO 600
400   PRINT 900, TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL)
      GO TO 600
500   PRINT 800, TKMAX,TYPEM(1,IL),TYPEM(2,IL),TYPEM(3,IL),SL,SU,DH,GLMA
     1X,(T(J),YM(J),J=1,3)
600   CONTINUE
      RETURN
C
C
C
C
700   FORMAT (T32,3A4,/,8X,'LOWER SS ',F5.2,2X,'UPPER SS ',F5.2,2X,'DROP
     1 HEIGHT',F5.1,2X,'G-LEVEL',F6.1,/,8X,'MATERIAL',' THICKNESS',F8.2,
     2/,8X,'TEMPERATURES',3X,3F12.2,/,8X,'WEIGHT',F8.2,2X,'MIN. BEARING
     3AREA',F8.2,2X,'MAX. BEARING AREA',F8.2,/)
800   FORMAT (//,5X,'***BEYOND MAXIMUM THICKNESS (',F6.2,'  )  OF  ',3A4
     1,/,6X,'LOWER SS ',F5.2,2X,'UPPER SS ',F5.2,2X,'DROP HEIGHT',F6.1,2
     2X,'G-LEVEL',F6.1,/,10X,'HOT      TEMPERATURE',3X,F8.3,3X,'G-LEVEL',
     33X,F8.3,/,10X,'AMBIENT TEMPERATURE',3X,F8.3,3X,'G-LEVEL',3X,F8.3,/
     4,10X,'COLD     TEMPERATURE',3X,F8.3,3X,'G-LEVEL',3X,F8.3,//)
900   FORMAT (//,5X,'INVALID LOWER TEMPERATURE FOR  ',3A4)
1000  FORMAT (1H1,T23,'CUSHION MATERIAL OPTIMIZATION',/,T33,'ENCAP',/)
      END
```

```
      SUBROUTINE TABLE
C
C     PRINT TABLE OF STATIC STRESS VERSUS G-LEVELS FOR THE THREE
C     TEMPERATURES - COLD,AMBIENT AND HOT
C
      COMMON /STD/ IL,TYPEM(3,20),NV(20),CONST(20),COEF(51,20)
      COMMON /STD1/TP,TC,SS,GL,NVR,V(51)
      COMMON /STD2/ CH,CA,CC,GLMAX,SSL(3),SSU(3),W,DH,BMIN,BMAX
C
      SS = 0.05
      A = 0.0
      AST = 9999999.0
      PRINT 500
      DO 400 I=1,100
      TP = CC
      CALL MODEL
      YC=GL
      IF(YC .GE. 0.0) GOTO 100
      A = 1.0
      YC = AST
100   CONTINUE
      TP = CA
      CALL MODEL
      YA=GL
      IF(YA .GE. 0.0) GOTO 200
      A = 1.0
      YA = AST
200   CONTINUE
      TP = CH
      CALL MODEL
      YH=GL
      IF(YH .GE. 0.0) GOTO 300
      A = 1.0
      YH = AST
300   CONTINUE
      PRINT 600,SS,YC,YA,YH
      SS = SS+0.05
400   CONTINUE
      IF( A .EQ. 0.0) RETURN
      PRINT 700
      RETURN
C
500   FORMAT(/,  8X,'STATIC STRESS',4X,'COLD G-LEVEL',4X,
     1'AMBIENT G-LEVEL',4X,'HOT G-LEVEL',/)
600   FORMAT(10X,F5.2,3(12X,F6.2))
700   FORMAT(    /,10X,'ASTERISKS INDICATE CALCULATED G VALUE' ,
     1 ' DOES NOT EXIST')
      END
```