

AD-A062 315

NORTH CAROLINA AGRICULTURAL AND TECHNICAL STATE UNIV --ETC F/G 9/5  
A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS AND A STUD--ETC(U)  
NOV 78

N00014-77-C-0199

NL

UNCLASSIFIED

1 OF 2  
ADA  
062315



**LEVEL II**



**ADA062315**

**NORTH CAROLINA A & T STATE UNIVERSITY**

**FINAL REPORT  
NAVAL AIR SYSTEMS COMMAND  
Washington, D. C.**

**A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS  
AND A STUDY OF IMPROVED STATISTICAL AND NON-DESTRUCTIVE TESTING  
METHODS OF FAILURE PREDICTION IN BRITTLE MATERIALS**

**Contract No. N00014-77-C-0199**

**November 10, 1978**

**DDC FILE COPY**

**DDC  
RECEIVED  
DEC 19 1978  
RESERVE**

**DISTRIBUTION STATEMENT A**

**Approved for public release;  
Distribution Unlimited**

**78 11 20 027**

**LEVEL II**

NAVAIR 3022 E

①

*New*  
NORTH CAROLINA A & T STATE UNIVERSITY

Greensboro.

⑨ FINAL REPORT.  
NAVAL AIR SYSTEMS COMMAND  
Washington, D. C.

⑥  
A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS  
AND A STUDY OF IMPROVED STATISTICAL AND NON-DESTRUCTIVE TESTING  
METHODS OF FAILURE PREDICTION IN BRITTLE MATERIALS

⑤ Contract No. N00014-77-C-0199

⑪ 14 Nov 1978

⑫ 122 p.

This report contains separate topical reports as indicated below:

	<u>Page</u>
A Study of Two Dimensional Recursive Digital Filters by Winsor E. Alexander .....	2
Indenter Hardness Studies by George J. Filatovs .....	45
Ceramic Fracture Analysis Through Biaxial Weibull Theory by William J. Craft .....	73

Project Director: William J. Craft  
School of Engineering  
N. C. A & T State University  
Greensboro, N. C. 27411  
(919) 379-7549

410 976

*JW*  
DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited

78 11 20 027

D D C  
R R R R R R R R R R  
DEC 19 1978  
D

SUMMARY

Research efforts covered by the subject contract have been directed toward the stability and synthesis problems for two dimensional digital recursive digital filters. Such filters are advantageous when a premium is placed upon computer storage or computation time.

Previously defined results on stability analysis have been refined and a revised paper has been submitted to the IEEE Transactions on Circuits and Systems for publication. This research effort will continue to address this complex problem.

Previously defined results on synthesis of bandpass and band enhancement filters have been refined. Results of these improvements will be installed in the Naval Intelligence Support Center's Spatial Domain Filtering Package which was designed and implemented by this author. This work will be done under a separate effort. This package has provided very satisfactory results to date and the refinements should significantly improve the overall performance of the package. Documentation on this spatial domain filtering package should be available within the next year.

Approximately circularly symmetric lowpass, highpass, bandpass, band-stop, low frequency boost and high frequency boost filters have been designed. Evaluation of these filters with actual images of various types need to be addressed and has been proposed as a follow on to the effort described in this report.

ADDRESSOR BY		
DTIC	White Section	<input checked="" type="checkbox"/>
DDC	Red Section	<input type="checkbox"/>
UNANNOUNCED <input type="checkbox"/>		
IDENTIFICATION		
Per Hc. on file		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist. AVAIL. and/or SPECIAL		
A		

## INTRODUCTION

The two-dimensional recursive digital filter is particularly suited to image processing applications when there is a premium placed on computer memory requirements and time for processing. Due to these considerations, it has definite advantages over the Fast Fourier Transform (FFT) algorithm for many image processing operations[1]. The application of recursive digital filters to image processing, however, has been hampered by two problems: stability and synthesis[2]. The synthesis problem is the problem of expressing the desired impulse response in closed form and thus determining the filtering coefficients. The stability problem occurs because the recursive filter requires feedback of past output values and therefore it can become unstable.

The research for the first year has been on both problems with progress made in both areas. This report discusses the progress made in both areas and the directions for future research.

## STABILITY

The stability problem for one dimensional digital recursive filters is straight forward. The roots of the polynomial in the closed form of the one dimensional  $Z$ -Transform for the filter impulse response must have magnitudes less than one. Stability analysis is therefore reduced to finding roots of  $n$ th degree polynomials with constant coefficients. For the two dimensional problem, stability is not straightforward because a two variable polynomial is not generally factorable into distinct roots. When the polynomial in the denominator of the two dimensional  $Z$ -Transform[3] for the impulse response is factorable, then the stability analysis procedure is the same as for the one dimensional problem.

The two dimensional stability problem is very complicated if the poly-

nomial in the denominator of the Z-Transform is not factorable into distinct roots. Efforts by other researchers have been directed toward examining regions of roots for two variable polynomials which is computationally feasible only for very simple filters[4].

The method used by this researcher is to express the two dimensional digital recursive equation as a matrix recursive equation. The description of the matrix recursive equation and its derivation is given in Appendix C. The resulting matrix recursive equation has three coefficient matrices,  $B_1$ ,  $B_2$  and  $A$ . Appendix A gives a summary of stability analysis results to date[5]. A paper entitled "Stability Analysis of two dimensional Recursive Filters" by W. E. Alexander and S. A. Pruess was revised as a part of this research effort and resubmitted for publication to the IEEE Transactions on Circuits and Systems. A preprint of this paper is given in Appendix D.

In practice, the stability analysis procedure which only involves finding the spectral radius of a matrix with real coefficients is very simple and easily implemented. Computer algorithms are readily available to perform the necessary computations. The procedure is regularly used by this researcher for stability analysis of two dimensional recursive digital filters.

#### SYNTHESIS

Often it is possible to express a desired two dimensional digital recursive filter as the product or sum of two one dimensional digital filters. That is the two dimensional Z-Transform of the digital recursive filter can be expressed as the product or sum of two one-dimensional Z-Transforms. In either case, the two dimensional synthesis problem is reduced to the synthesis of two one-dimensional filters. However, it is not possible to design sum

separable or product separable digital recursive filters for all applications. For those applications, the design of the required two dimensional digital recursive filter is considerably more complicated.

Many imaging systems have a natural circular symmetry. In general, the optical transfer function of a circularly symmetric imaging system is uniform with respect to direction. The natural consequence is that filters with circularly symmetric impulse response functions are generally very desirable for image processing. The relationship between circular symmetry of the impulse response and the frequency response dictates that the design requirement is for these filters to have a circularly symmetric frequency response[6].

#### LOW PASS FILTER DESIGN

The design goal is to design a low pass filter with circularly symmetric frequency magnitude characteristics. No attempt is made to control the phase response of the desired filter. This presents no difficulties in implementing the designed filters because the two pass, linear phase recursive digital filtering procedure can be used to obtain linear phase[5].

The magnitude characteristic for the one dimensional Butterworth approximation filter in the Laplace Transform variable is given by

$$h(s)h(-s) = \frac{1}{1 + (-1)^n \epsilon^2 \left( \frac{s}{\omega_0} \right)^{2n}} \quad (1)$$

The corresponding equation for two dimensional filters is given by

$$h(s_1, s_2) = \frac{1}{1 + (-1)^n \epsilon^2 \left( \frac{s_1^2 + s_2^2}{\omega_x^2 + \omega_y^2} \right)^n} \quad (2)$$

where  $s_1$  and  $\omega_x$  are respectively Laplace Transform and cutoff frequency variables for the x direction and  $s_2$  and  $\omega_y$  are respectively Laplace Transform and cutoff frequency variables for the y direction.

If the bilinear transformation[7] is applied to (2) to obtain a two dimensional Z-Transform, we obtain

$$H(z, w)^2 = \frac{[(z+1)^2 (w+1)^2]^n}{[(z+1)^2 (w+1)^2]^n + \epsilon^2 (-1)^n c^n [(z-1)^2 (w+1)^2 + (z+1)^2 (w-1)^2]} \quad (3)$$

$$\text{Let } c = 1 / [\tan^2(\omega_R T/2)] \quad \omega_R^T = \omega_x^2 T_x^2 + \omega_y^2 T_y^2 \quad (4)$$

Note that  $\omega_y$  is the effective radial cutoff frequency. In continuing the design procedure in manner similar to that used for one dimensional digital recursive filters[8] difficulties are encountered because the denominator of (3) is not factorable in distinct roots of z and w. However, a suitable approximation may be obtained by factoring along the w-z plane.

Thus in this plane, one obtains

$$\left| H(z, z) \right|^2 = \frac{(z+1)^{4n}}{(z+1)^{4n} + (-1)^n c^n [2(z-1)^2 (z+1)^2]^n} \quad (5)$$

Simplifying, we obtain

$$\left| H(z, z) \right|^2 = \frac{(z+1)^{2n}}{(z+1)^{2n} + \epsilon^2 (-1)^n c^n [2(z-1)^2]^n} \quad (6)$$

Thus the poles of the magnitude response in the w=z plane occur in reciprocal pairs as roots of the denominator of (6).

As with one dimensional filters, consideration should be given to round off errors and truncation errors in implementing two dimensional digital recursive filters. Thus a cascade realization is very desirable

because it acts to minimize round off error. Also, it is desirable to avoid using complex arithmetic when implementing two dimensional recursive filters. This leads to a natural selection of implementing a basic filter with either one pole and one zero or two poles and two zeros to accommodate complex conjugate pairs of poles. Then any general filter would be implemented as stages of the one pole or two pole filter.

If we let  $n=1$  in (6), we obtain a factorization of  $|H(z,z)|^2 = H_x(z)H_x(z^{-1})$  such that for a stable filter design.

$$H_x(z) = \frac{A(z+1)}{(z+P)} \quad (7)$$

$$A = \frac{1}{1-2C\epsilon^2} \quad (8)$$

$$P = \frac{+(1+2C\epsilon^2) - 2\sqrt{2C\epsilon^2}}{1-2C\epsilon^2} \quad (9)$$

except that  $P=0$  for  $C=0.5/\epsilon^2$ .

For the case where  $n$  is equal to or greater than 2, factorization becomes more complicated and the computer is used to find the roots with magnitudes less than one.

Forming the two dimensional Z-Transform for the final low pass filter design for  $n$  equal to one, we obtain

$$H_L(z,w) = \frac{A^2(z+1)(w+1)}{(z+P)(w+P)} \quad (10)$$

Note that this filter design is product separable and inherently stable because we have selected  $P$  such that  $|P|$  is always less than one. In a similar fashion, we can design filters for  $n$  greater than one.

#### BAND PASS FILTER DESIGN

Once a low pass filter has been designed, it is possible to obtain highpass, band pass and band stop filters as well as low frequency boost

and high frequency boost filters from the low pass design. In this section, we discuss the design of a general boost filter which can be used with proper parameter values to obtain the above mentioned filters. With the low pass filter design ( $n=1$ ) given in (10), we can obtain a filter with the desired magnitude as given by

$$|H(z,w)| = \alpha + \beta |H_L(z,w)|^2 \quad (11)$$

where  $H_L(z,w) = H_L(z,w) H_L(z^{-1}, w^{-1})$ . Thus

$$H(z,w) = \alpha + \beta A^4 \frac{(z+1)(z^{-1}+1)(w+1)(w^{-1}+1)}{(z+P)(z^{-1}+P)(w+P)(w^{-1}+P)} \quad (12)$$

$$|H(z,w)| = \frac{\alpha[(z+P)(1+Pz)(w+P)(1+Pw)] + \beta A^4 [(z+1)^2(w+1)^2]}{(z+P)(1+Pz)(w+P)(1+Pw)} \quad (13)$$

Note that the filter represented by (13) is unstable because of the terms  $(1+Pz)$  and  $(1+Pw)$  in the denominator. The poles corresponding to these terms have magnitudes greater than one since  $P$  has a magnitude less than one. However, we can stabilize (13) by using the minimum phase version of the denominator. This does not change the magnitude since the magnitude of  $(1+Pz)$  is equal to the magnitude of  $(z+P)$  and the magnitude of  $(1+Pw)$  is equal to the magnitude of  $(w+P)$ . Thus the desired boost filter design has the two dimensional Z-Transform

$$H_B(z,w) = \frac{\alpha [Pz^2 + (1+P^2)z + P] [Pw^2 + (1+P^2)w + P] + \beta A^4 [(z+1)^2 (w+1)^2]}{(z+P)^2 (w+P)^2} \quad (14)$$

thus if express (14) in the form

$$H_B(z,w) = \frac{\sum_{J=0}^L \sum_{K=0}^L a_{JK} z^{-J} w^{-K}}{\sum_{J=0}^L \sum_{K=0}^L b_{JK} z^{-J} w^{-K}} \quad (15)$$

it follows that

$$\begin{aligned} a_{00} &= a_{22} = \alpha P^2 + \beta A^4 \\ a_{10} &= a_{01} = a_{12} = \alpha P(1+P^2) + 2\beta A^4 \\ a_{02} &= a_{20} = \alpha P^2 + \beta A^4 \\ a_{11} &= \alpha(1+P^2)^2 + 4\beta A^4 \end{aligned} \quad (16)$$

(17)

$$\begin{aligned}
 b_{00} &= 1.0 \\
 b_{01} &= b_{10} = 2P \\
 b_{02} &= b_{20} = P^2 \\
 b_{12} &= b_{21} = 2P^3 \\
 b_{11} &= 4P^2 \\
 b_{22} &= P^4
 \end{aligned}$$

It only remains to determine the value of  $\epsilon$  for both the low pass and boost filters. Note that the squared magnitude of  $H_L(Z,W)$  is equal to  $1/(1+\epsilon^2)$  when the radial frequency,  $\omega$ , is equal to the radial cutoff frequency,  $\omega_R$ . Thus for  $n=1$   $|H_L(Z,W)|^2 = 1/(1+\epsilon^2)$  at the cutoff frequency. If we use the double pass linear phase filter which is desirable[5], the magnitude of the resultant filter is the squared magnitude of the original filter. If we desire the magnitude of the resulting filter to be down 3db at the cutoff, we obtain

$$\frac{1}{\sqrt{2}} = \frac{1}{(1+\epsilon^2)^2} \quad (18)$$

The same value for  $\epsilon^2$  is appropriate for the low frequency boost filter.

For the high pass filter, we have  $\alpha=1$  and  $\beta=-1$ . Thus the magnitude of the frequency response for the double pass linear phase filter at the cutoff frequency is given by the relationship

$$\frac{1}{2} = 1 - \frac{1}{(1+\epsilon^2)^2} \quad (19)$$

The same value for  $\epsilon^2 = 1.0/(2^{1/2}-1)$  is appropriate for the high frequency boost filter.

If we designated B as the magnitude of the desired boost, then the

low frequency boost filter values for  $\alpha$  and  $\beta$  are given by

$$\alpha = 1.0 ; \quad \beta = B-1.0 \quad (20)$$

Correspondingly, for the high frequency boost filter, the values of  $\alpha$  and  $\beta$  are given by

$$\alpha = B ; \quad \beta = - B+1.0 \quad (21)$$

Examples of two dimensional recursive filter designs are given in Appendix B.

#### ROTATED ONE DIMENSIONAL FILTERS

A problem of interest in image processing is to filter with a one dimensional filter with the orientation of the filter specified and independent of the sampling directions. This type of filter would be useful for enhancing or suppressing linear features, for system noise suppression or for image correction (i.e. linear smear). However, any one dimensional digital recursive filter which is rotated becomes a two dimensional filter associated problems in stability and synthesis.

Constraints with regard to angle of rotation and stability of rotated filters have been developed by Costa and Ventsonopoulos[9]. They have used several rotated low pass filters to obtain circularly symmetric lowpass filters. This approach is currently being evaluated with regard to use with single rotated filters.

#### RECOMMENDATIONS FOR FUTURE RESEARCH

Approximately circularly symmetric band pass and band enhancement filters have been designed. These filter designs must be evaluated with regard to performance on actual images of various types and with regard

to circular symmetry as a function of critical frequency. Methods of improving response and eliminating errors in circular symmetry need to be investigated.

Techniques for obtaining rotated one dimensional digital filters need to be investigated with regard to practical use. The most practical method will be developed into an algorithm for image processing.

The use of recursive digital filters for image correction has been hampered by problems in designing filters with arbitrarily specified magnitude and phase characteristics. Long term future research efforts need to be directed toward this problem.

Recursive digital filters are practical for image processing applications using small computers, minicomputers and special signal processors. Applications range from real time image processing and data acquisition to medical applications and industrial process monitoring. However, special algorithms must be developed for many of these applications. Such practical research problems provide the real payoff for research on two dimensional digital recursive filters.

## REFERENCES

- [1] Ernest L. Hall, "A Comparison of Computations for Spatial Frequency Filtering", Proceedings of the IEEE, Vol. 60, No. 7, 1972, pp 887-891
- [2] N. K. Bose, "Problems and Progress in Multidimensional Systems Theory", Proceedings of the IEEE, Vol. 65, No. 6, 1977, pp. 824-840.
- [3] Lawrence R. Rabiner and Bernard Gold, Theory and Applications of Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975, pp. 442-455.
- [4] Dennis Goodman, "Some Stability Properties of the Two Dimensional Linear Shift-Invariant Digital Filters", IEEE Transactions on Circuits and Systems, Vol. CAS24, No. 4, 1977, pp. 201-208.
- [5] Winser E. Alexander, Stability and Synthesis of Two-Dimensional Digital Recursive Filters, Ph.D. Dissertation, University of New Mexico, Albuquerque, N. M., 1974 (University Microfilms, Ann Arbor, Mich.).
- [6] A. Papoulis, Systems and Transforms with Applications in Optics, McGraw-Hill, Inc., New York, N. Y., 1968, p. 140.
- [7] Bernard Gold and Charles Rader, Digital Processing of Signals, McGraw-Hill, Inc., New York, N. Y., 1969, Chapter 6.
- [8] Samuel Stearns, Digital Signal Analysis, Hayden Publishing Company, Rochelle Park, N. J., 1975, Chapter 12.
- [9] J. M. Costa and A. N. Venetsonopoulos, "Design of Circularly Symmetric Two-Dimensional Recursive Filters", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-22, No. 6, 1974, pp. 432-442.

## APPENDIX A

Summary of Two Dimensional Digital Recursive  
Digital Filter Stability analysis Results

Given the two dimensional digital recursive filter with the corresponding biavariate recursive equation

$$g(m,n) = \sum_{J=0}^L \sum_{K=0}^L a_{JK} f(m-J,n-K) - \sum_{\substack{J=0 \\ J+K>0}}^L \sum_{K=0}^L b_{JK} g(m-J,n-K) \quad (A.1)$$

where  $g(m,n)$  is the current output,  $g(m-J,n-K)$  represents past output values and  $f(m-J,n-K)$  represents current and past input values for all permissible values of  $J$  and  $K$ . It is possible to represent this relationship by a matrix recursive equation (see Appendix C)

(A.2)

$$G_{m,n} = \bar{B}_1 G_{m-1,n} + \bar{B}_2 G_{m,n-1} + \bar{A} F_{m,n}$$

where  $G_{m,n}$  is a column vector such that all its elements are the outputs,  $g(m-J,n-K)$ , where  $0 \leq J \leq L$  and  $0 \leq K \leq L$ .  $F_{m,n}$  is a column vector such that its elements are the inputs,  $f(m-J,n-K)$  and  $\bar{B}_1, \bar{B}_2$  and  $\bar{A}$  are appropriate coefficient matrices such that (A.1) and (A.2) are equivalent.

**Theorem 1:** Given the discrete system represented by the matrix recursive equation in (A.2). If either spectral radii,  $\rho(\bar{B}_1)$  or  $\rho(\bar{B}_2)$  is greater than or equal to one, then the system is computationally unstable.

**Theorem 2:** Given the discrete system represented by the matrix recursive equation in (A.2). The system is computationally unstable if  $\rho(\bar{B}_1 + \bar{B}_2)$  is greater than or equal to one.

**Theorem 3:** Given the discrete system represented by the matrix recursive equation in (A.2). The system is stable is

$$\rho \left[ \text{abs}(\bar{B}_1) + \text{abs}(\bar{B}_2) \right] < 1 \quad (A.3)$$

where  $\text{abs}(B_1)$  and  $\text{abs}(B_2)$  refers to taking the absolute value of all elements in  $B_1$  and  $B_2$  respectively .

**Theorem 4:** Given the discrete system represented by the matrix recursive equation in (A.2). Define a particular permutation matrix  $S$  (See Appendix C). The system is stable if  $\rho(B_1+B_2)<1$ ,  $\rho(B_1S)<1/2$  and  $\rho(B_2S)<1/2$ .

**Conjecture:** Given the discrete system represented by the matrix recursive equation in (A.2). The system is stable if and only if  $\rho(B_1)<1$ ,  $\rho(B_2)<1$  and  $\rho(B_1+B_2)<1$ .

Proofs for Theorems 1 through 4 have been developed. Current research is directed toward verifying the practical usage of these theorems and to further investigation of the conjecture.

## APPENDIX B

## Filter Design Examples

The filter synthesis procedure for designing two dimensional digital recursive filters in this research effort is an extension of a one dimensional filter synthesis procedure. The squared magnitude characteristic of the desired circularly symmetric two dimensional filter is chosen in the Laplace Transform domain. The Butterworth filter characteristic has been chosen because of its wide spread use in band pass filter applications. The bilinear transformation is then used to map the squared magnitude characteristic into the two dimensional Z-Transform domain. The coefficient matrices,  $B_1$  and  $B_2$  of the corresponding matrix recursive equation (See Appendix C) are obtained and the eigenvalues of the matrix sum  $(B_1 + B_2)$  are determined. These eigenvalues occur in reciprocal pairs because the original function was a magnitude response. The eigenvalues with magnitudes less than one are then used as roots of a product separable denominator to form the denominator of the two dimensional Z-Transform for a stable filter. The numerator for the filter is retained from the mapping of the squared magnitude characteristic to the two dimensional Z-Transform.

This procedure has been used to design and implement two dimensional recursive lowpass, highpass, low frequency boost and high frequency boost filters. Some examples are given below. It should be emphasized that this design procedure always results in stable filters. Also, the filter algorithm necessary to implement these filters has been developed for a CDC 6400 System by this researcher at the Naval Intelligence Support Center in Suitland, Maryland.

There are still some minor problems remaining with this procedure with

regard to obtaining circular symmetry when the critical frequency is near the Nyquist frequency or near zero. These problems are being studied with the intent of providing necessary corrective improvements in these areas.

Preliminary results indicate that the problem near the Nyquist frequency is caused by the mapping of the squared magnitude characteristic to the two dimensional Z-Transform using the bilinear transformation. However, it does appear that significant correction can be made for the case when the critical frequency is near zero.

Figure B.1 shows the contour plot of a low pass filter design with a cutoff frequency of 0.4 of the Nyquist frequency. The contour labeled D is the half power point. Figure B.2 is the contour plot of a high frequency boost filter with a break frequency of 0.5 and relative boost of high frequency to 25.6. The contour labeled D is the half power point. Figure B.3 is The prospective plot of this filter. The design goal is that the contours and specifically the break frequency contour be circularly symmetric.

FIG. B.1: Contour Plot for Low pass Filter.  
Cutoff frequency is 0.4 Nyquist Frequency

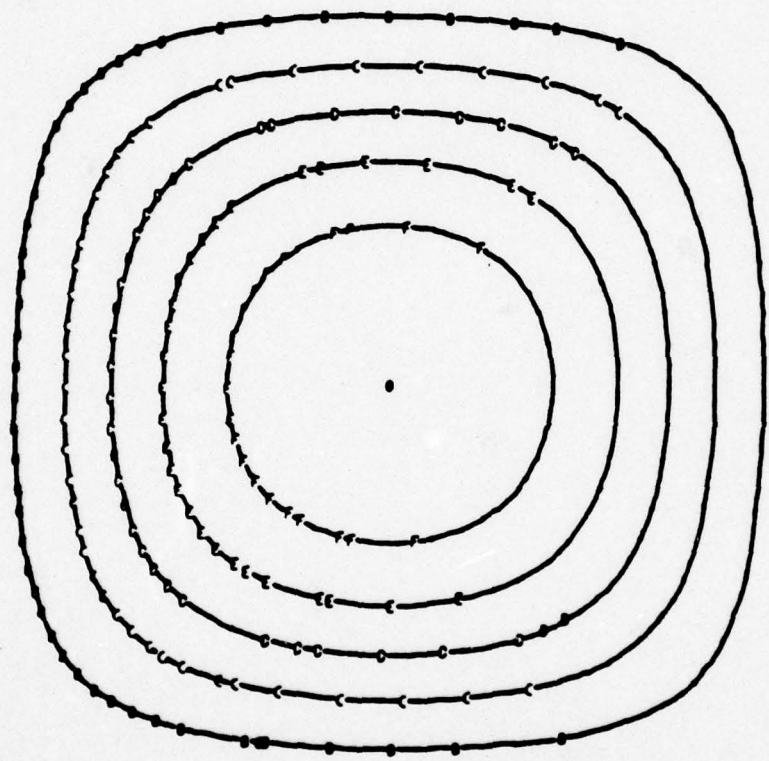


FIG. B.2: Contour Plot for high frequency boost filter.  
Break frequency is 0.5 Nyquist frequency.  
Relative boost of high frequencies is 25.6

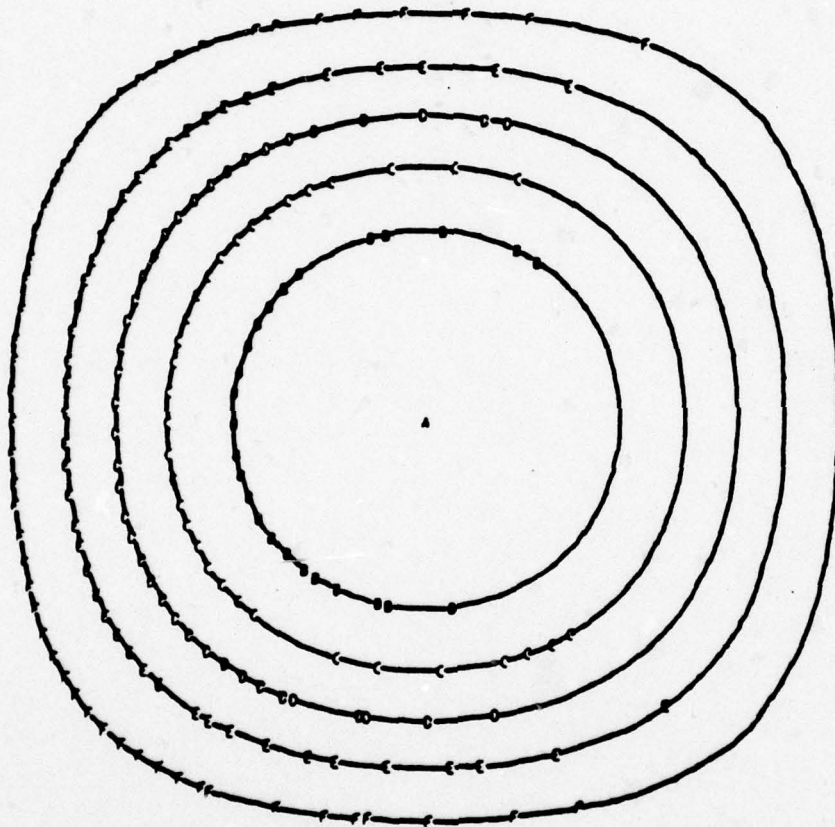
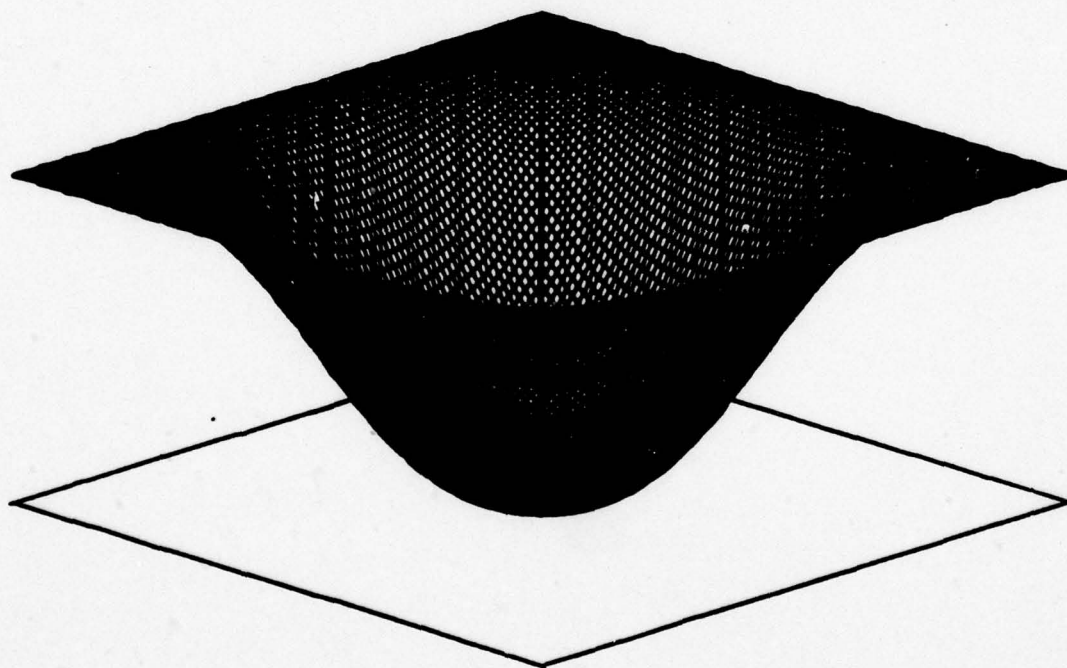


FIG.B.3: Perspective plot for high frequency boost filter of Figure B.2



## APPENDIX C

## Description of the Matrix Recursive Equation

## C.1 Introduction

In this appendix, the matrix representation of the two dimensional digital recursive filter is presented in detail. The matrix S which is used for the Proof of Theorem 4 for stability analysis as presented in Appendix A is also described.

## C.2 The Matrix Recursive Equation

Consider the two dimensional digital recursive filter which has the ZW-Transform

$$H(z,w) = \frac{\sum_{J=0}^L \sum_{K=0}^L a_{JK} z^{-J} w^{-K}}{\sum_{J=0}^L \sum_{K=0}^L b_{JK} z^{-J} w^{-K}} \quad (C.1)$$

The corresponding two dimensional recursive algorithm for the filter is given by

$$g(m,n) = \sum_{J=0}^L \sum_{K=0}^L a_{JK} f(m-J, n-K) - \sum_{\substack{J=0 \\ J+K > 0}}^L \sum_{K=0}^L b_{JK} g(m-J, n-K) \quad (C.2)$$

Define the matrix V such that the element of V in the Jth row and Kth column is given by  $b_{J-1, K-1}$ . That is

$$V = \begin{bmatrix} b_{00} & b_{01} & \cdot & \cdot & \cdot & b_{0L} \\ b_{01} & b_{11} & \cdot & \cdot & \cdot & b_{1L} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{L0} & b_{L1} & \cdot & \cdot & \cdot & b_{LL} \end{bmatrix}$$

or

$$v = [v_{JK}] = [b_{J-1, K-1}] \quad (C.4)$$

$V$  is the filtering matrix corresponding to the denominator of the ZW-Transform  $H(z,w)$ . A similar filtering matrix can be defined for the numerator of the ZW-Transform. However, this matrix is not used in this development.

Define the vector  $G_{m,n}$  which contains all of the outputs in the recursive algorithm for the filter as given in (C.2). Order the outputs such that the outputs  $g(m-J, n)$ , for all values of  $J$ , occur before the output  $g(m, n-1)$  and the outputs  $g(m-J, n-1)$ , for all values of  $J$ , occur in order before the output  $g(m, n-2)$ . Continue in this manner until all outputs are included. The vector  $G_{m,n}$  is then given by

$$G = \begin{bmatrix} g(m,n) \\ g(m-1,n) \\ \vdots \\ g(m-L,n) \\ g(m,n-1) \\ \vdots \\ g(m-L,n-1) \\ \vdots \\ g(m-L,n-L) \end{bmatrix} \quad (C.5)$$

The vector  $G_{m-1,n}$  is then obtained by decreasing the first parameter of each output in  $G_{m,n}$  by one. That is

$$G_{m-1,n} = \begin{bmatrix} g(m-1,n) \\ g(m-2,n) \\ \vdots \\ g(m-L,n) \\ g(m-1,n-1) \\ \vdots \\ g(m-L-1,n-1) \\ \vdots \\ g(m-L-1,n-L) \end{bmatrix} \quad (C.6)$$

The vector  $G_{m,n-1}$  is obtained by decreasing the second parameter of each of the outputs in  $G_{m,n}$ . Thus

$$G_{m,n-1} = \begin{bmatrix} g(m,n-1) \\ g(m-1,n-1) \\ \vdots \\ g(m-L,n-1) \\ g(m,n-2) \\ \vdots \\ g(m-L,n-2) \\ \vdots \\ g(m-L,n-L-1) \end{bmatrix} \quad (C.7)$$

Define  $F_{m,n}$  as the vector which contains all of the inputs in the recursive algorithm for the filter as given in (C.2). Order the inputs in the same as the outputs were ordered for the vector  $G_{m,n}$ . Then

$$F_{m,n} = \begin{bmatrix} f(m,n) \\ f(m-1,n) \\ \vdots \\ f(m-L,n) \\ f(m,n-1) \\ \vdots \\ f(m-L,n-1) \\ \vdots \\ f(m-L,n-L) \end{bmatrix} \quad (C.8)$$

Define the matrix  $B_1$  with elements  $b_{JK}^{(1)}$  and the matrix  $B_2$  with elements  $b_{JK}^{(2)}$ .  $B_1$  and  $B_2$  are  $(L+1)^2$  by  $(L+1)^2$  matrices such that

(C.9)

$$B_1 = \begin{bmatrix} b_{JK}^{(1)} \end{bmatrix}$$

$$B_2 = \begin{bmatrix} b_{JK}^{(2)} \end{bmatrix}$$

(C.10)

The elements  $b_{JK}^{(1)}$  are given by the algorithm

for  $J = 1$  to  $L + 1$

For  $I = 1$  to  $L$

Let  $K = I + (J-1)(L+1)$

If  $J = 1$ ,  $b_{IK}^{(1)} = -b_{I,J-1} = -v_{I+1,J}$

If  $J = 1$ ,  $b_{K+1,K}^{(1)} = 1$

If  $J > 1$ ,  $b_{IK}^{(1)} = -\frac{1}{2} b_{I,J+1} = -\frac{1}{2} v_{I+1,J}$

If  $J > 1$ ,  $b_{K+1,K}^{(1)} = \frac{1}{2}$

Otherwise  $b_{JK}^{(1)} = 0$

The elements  $b_{JK}^{(2)}$  are given by the algorithm

For  $J = 1$  to  $L$

For  $I = 1$  to  $L + 1$

Let  $K = I + (J-1)(L+1)$

If  $I = 1$ ,  $b_{IK}^{(2)} = -b_{I-1,J} = -v_{I,J+1}$

If  $I = 1$ ,  $b_{K+L,K}^{(2)} = 1.0$

If  $I > 1$ ,  $b_{IK}^{(2)} = -\frac{1}{2} b_{I-1,J} = -\frac{1}{2} v_{I,J+1}$

If  $I > 1$ ,  $b_{K+L,K}^{(2)} = \frac{1}{2}$

Otherwise  $b_{JK}^{(2)} = 0$

Define the  $(L+1)^2$  by  $(L+1)^2$  matrix  $A$  with elements  $\alpha_{JK}$ . Here we depart from the standard notation to avoid confusion between the elements of the matrix  $A$  and the coefficients of the filter specified by (C.1). Then

$$A = [\alpha_{JK}] \quad (C.11)$$

The elements  $\alpha_{JK}$  are given by the algorithm:

For  $I = 1$  to  $L + 1$

For  $J = 1$  to  $L + 1$

Let  $K = I + (J-1)(L+1)$

$$\alpha_{1,K} = a_{I-1,J-1}$$

Otherwise  $\alpha_{J,K} = 0$

### Example C.1

Consider the filter specified by (C.1) or (C.1) where  $L$  is equal to 2.

In that case,  $G_{m,n}$ ,  $G_{m-1,n}$ ,  $G_{m,n-1}$  and  $F_{m,n}$  are  $9 \times 1$  vectors and  $B_1$ ,  $B_2$ , and  $A$  are  $9 \times 9$  matrices. The vectors  $G_{m,n}$ ,  $G_{m-1,n}$ ,  $G_{m,n-1}$  and  $F_{m,n}$  are given by

$$G_{m,n} = \begin{bmatrix} g(m,n) \\ g(m-1,n) \\ g(m-2,n) \\ g(m,n-1) \\ g(m-1,n-1) \\ g(m-2,n-1) \\ g(m,n-2) \\ g(m-1,n-2) \\ g(m-2,n-2) \end{bmatrix} \quad (C.12)$$

$$G_{m-1,n} = \begin{bmatrix} g(m-1,n) \\ g(m-2,n) \\ g(m-3,n) \\ g(m-1,n-1) \\ g(m-2,n-1) \\ g(m-3,n-1) \\ g(m-1,n-2) \\ g(m-2,n-2) \\ g(m-3,n-2) \end{bmatrix} \quad (\text{C.13})$$

$$G_{m,n-1} = \begin{bmatrix} g(m,n-1) \\ g(m-1,n-1) \\ g(m-2,n-1) \\ g(m,n-2) \\ g(m-1,n-2) \\ g(m-2,n-2) \\ g(m,n-3) \\ g(m-1,n-3) \\ g(m-2,n-3) \end{bmatrix} \quad (\text{C.14})$$

and

$$F_{m,n} = \begin{bmatrix} f(m,n) \\ f(m-1,n) \\ f(m-2,n) \\ f(m,n-1) \\ f(m-1,n-1) \\ f(m-2,n-1) \\ f(m,n-2) \\ f(m-1,n-2) \\ f(m-2,n-2) \end{bmatrix} \quad (\text{C.15})$$

The matrices  $B_1$ ,  $B_2$  and  $A$  are given by

$$B_1 = \begin{bmatrix} -b_{10} & -b_{20} & 0 & -\frac{1}{2}b_{11} & \frac{1}{2}b_{21} & 0 & -\frac{1}{2}b_{12} & -\frac{1}{2}b_{22} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} \quad (\text{C.16})$$

$$B_2 = \begin{bmatrix} -b_{01} & -\frac{1}{2}b_{11} & -b_{21} & -b_{02} & -\frac{1}{2}b_{12} & -\frac{1}{2}b_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \end{bmatrix} \quad (C.17)$$

$$A = \begin{bmatrix} a_{00} & a_{10} & a_{20} & a_{01} & a_{11} & a_{21} & a_{02} & a_{12} & a_{22} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (C.18)$$

We can then express (C.2) in the matrix form

$$G_{m,n} = B_1 G_{m-1,n} + B_2 G_{m,n-1} + A F_{m,n} \quad (C.19)$$

Note that if we write the equations corresponding to the rows of (C.19) we obtain for the first row

$$g(m, n) = - \sum_{J=0}^L \sum_{K=0}^L b_{JK} g(m-J, n-K) + \sum_{\substack{J=0 \\ J+K>0}}^L \sum_{K=0}^L a_{JK} f(m-J, n-K) \quad (C.20)$$

For the subsequent rows, we obtain

$$g(m-J, n-K) = g(m-J, n-K) \quad (C.21)$$

or the outputs are equated to themselves. It follows directly that (C.19) is equivalent to (C.2).

### C.3 The S Matrix

We now give the algorithm for the matrix which is used to reorder the rows and columns of the matrices  $B_1$  and  $B_2$  for the proof of Theorem 4 as described in Appendix A. Define the  $(L+1)^2$  by  $(L+1)^2$  matrix  $S$  such that

$$S = [s_{JK}] \quad (C.22)$$

The elements  $S_{JK}$  are given by the algorithm

For  $I = 1$  to  $L + 1$

For  $M = 1$  to  $L + 1$

Let  $J = M + (I-1)(L+1)$

Let  $K = I + (M-1)(L+1)$

$s_{JK} = 1.0$

Otherwise  $s = 0$

### Example C.2

Consider the filter specified by (C.1) or (C.2) where  $L$  is equal to 2.

In that case, we have

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

## APPENDIX D

Stability Analysis of Two-Dimensional Recursive Filters  
(A Preprint)

Stability Analysis of Two-Dimensional Recursive Filters\*

Winser E. Alexander, Member, IEEE

and

Steven A. Pruess

---

\* Manuscript received October 10, 1975; revised January 31, 1977 and June 23, 1978. This work was partially supported by the U. S. Department of Energy and partially by the Naval Air Systems Command.

W. E. Alexander is with the Department of Electrical Engineering at North Carolina Agricultural and Technical State University

S. A. Pruess is with the Department of Mathematics at the University of New Mexico.

---

## ABSTRACT

A new approach to the stability problem for the two-dimensional digital recursive filter is presented. The bivariate difference equation representation of the two-dimensional recursive filter is converted to a multi-input multi-output (MIMO) system similar to the state space representation of the one dimensional digital recursive filter. In this paper, a pseudo-state space representation is used and three two-dimensional polynomial matrices are obtained. A general theorem for stability of two-dimensional digital recursive filters is derived and a very useful theorem is presented which expresses sufficient requirements for instability in terms of the spectral radii of these matrices.

## I. Introduction

A two-dimensional digital recursive filter can be characterized by the bivariate difference equation

$$g(m,n) = \sum_{J=0}^L \sum_{K=0}^L a_{JK} f(m-J, n-K) - \sum_{J=0}^L \sum_{K=0}^L b_{JK} g(m-J, n-K) \quad (1)$$

where the coefficients  $a_{JK}$  and  $b_{JK}$  are constants [1] and some of these constants may be zero. There are two major problems to consider in the design of recursive filters for two-dimensional signal processing: synthesis and stability. The synthesis problem consists of determining the filter coefficients so that the required frequency response is realized. If the resulting filter is to be useful, it must be bounded input-bounded output (BIBO) stable. In this paper the stability problem is considered and a new approach to stability analysis for the two-dimensional digital recursive filter is presented.

For the one-dimensional case, there are essentially two methods of determining necessary and sufficient conditions for stability of digital filters: examining regions of analyticity for the characteristic polynomial and by direct evaluation of the characteristics of the impulse response [2,3,4]. In particular, if the system corresponding to the digital filter is represented by a state space equation, then one can determine stability from the coefficient matrices in the state space equation[4]. For the two-dimensional case, generalizations of the first method involves examining regions of analyticity for bivariate polynomials which is computationally feasible only for very simple filters[5]. This paper attempts to generalize the second method for the two-dimensional case, i.e. to establish stability by computing the spectral radii of coefficient matrices with real coefficients.

## II. Pseudo State Space Representation

Fornasini and Marchesini [6] have defined a state space representation of the two-dimensional digital recursive filter. In this paper, we use a particular case of the Fornasini-Marchesini model where one of the coefficient matrices is the null matrix. Thus, we obtain the pseudo state space representation

$$\begin{aligned} G_{m,n} &= B_1 G_{m-1,n} + B_2 G_{m,n-1} + AF_{m,n} \\ g(m,n) &= DG_{m,n} \end{aligned} \quad (2)$$

$G_{m,n}$  is a column vector such that its elements are the outputs,  $g(m-J, n-K)$  where  $0 \leq J \leq L$  and  $0 \leq K \leq L$ . Note that  $G_{m,n}$  contains all of the outputs that are represented in (1) including  $g(m,n)$ . Similarly,  $F_{m,n}$  is a column vector such that its elements are the inputs,  $f(m-J, n-K)$  where  $0 \leq J \leq L$  and  $0 \leq K \leq L$ .

We can then define matrices  $B_1$ ,  $B_2$  and  $A$  [7] such that (1) and (2) are equivalent. The matrices  $B_1$ ,  $B_2$  and  $A$  are all of order  $(L+1)^2$  by  $(L+1)^2$ . The vector  $D$  is a row vector with  $L+1$  elements.

The ordering of the outputs in  $G_{m,n}$  and of the inputs in  $F_{m,n}$  is not unique. However, the ordering does affect the relative position of the elements of the corresponding coefficient matrices. Also note that there are identical elements in  $G_{m-1,n}$  and  $G_{m,n-1}$ . Where this occurs, the corresponding elements of  $B_1$  and  $B_2$  can be divided such that the magnitude of each is no larger than that of the corresponding  $b_{JK}$  or one as appropriate. It is convenient to consistently divide equally and choose a particular ordering scheme.

### III. Stability Analysis

The stability analysis herein will be confined to the linear shift invariant (LSI) two-dimensional discrete system. Such a system is BIBO stable if and only if the discrete impulse response of the system,  $h(m,n)$ , is absolutely summable, i.e.,  $\sum_{m,n=0}^{\infty} |h(m,n)| < \infty$  [1].

Let us define the particular vector  $H_{J,K}$  as that input vector which represents a single unit sample at the  $(J,K)$  position of the two-dimensional data array and all other inputs are zero. Let us further define the initial condition vector,  $G_{J-1,K}$  and  $G_{J,K-1}$ , as null vectors. Then for  $m=J$  and  $n=K$ , (2) reduces to

$$\begin{aligned} G_{J,K} &= AH_{J,K} \\ h(J,K) &= DG_{J,K} \end{aligned} \quad (3)$$

Define the term  $C(B_1^J, B_2^K)$  as the sum of all unique products involving  $B_1$  as a factor  $J$  times and  $B_2$  as a factor  $K$  times. It is helpful to note that if  $B_1$  and  $B_2$  commute, then  $C(B_1^J, B_2^K) = \binom{J+K}{K} B_1^J B_2^K = (J+K)! B_1^J B_2^K / (J!K!)$ . In general, the matrices do not commute. Therefore, we give as an example  $C(B_1^2, B_2^1) = B_1^2 B_2 + B_1 B_2 B_1 + B_2 B_1^2$ .

Lemma 1: Given the discrete LSI system represented by (2), the contribution to the output vector,  $G_{m,n}$ , by a single input vector,  $H_{J,K}$ , which corresponds to a unit impulse at the  $(J,K)$  position where  $J \leq m$  and  $K \leq n$  is given by  $G_{m,n} = C(B_1^{m-J}, B_2^{n-K}) AH_{J,K}$ .

The proof of Lemma 1 is given in the Appendix. Lemma 1 provides a convenient means of finding the output of the two-dimensional digital recursive filter for all values of  $m$  and  $n$  when the filter is excited by a single input at any point in the array. Since the filter is linear and shift invariant, we can use the principle of superposition to find the output for

any particular sequence of inputs.

Thus, the unit impulse response of the filter is given by

$$\begin{aligned} G_{m,n} &= C(B_1^m, B_2^n) A H_{0,0} \\ h(m,n) &= D G_{m,n} = DC(B_1^m, B_2^n) A H_{0,0} \end{aligned} \quad (4)$$

**Lemma 2:** Given the discrete LSI system represented by (2) for which the corresponding transfer function has mutually prime numerator and denominator polynomials. If the contribution to the output vector  $G_{m,n}$  by a bounded sequence of input vectors  $F_{J,K}$  where  $0 \leq J < M$  and  $0 \leq K < N$  can be expressed by  $G_{m,n} = Q^m A F_{J,K}$  or  $G_{m,n} = Q^n A F_{J,K}$ , then the system is unstable if  $\rho(Q)$ , the spectral radius of  $Q$ , is greater than one. The proof of Lemma 2 is given in the Appendix.

**Theorem 1:** The discrete LSI system represented by (2) is stable if and only if for at least one matrix norm

$$S_2 = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \|DC(B_1^m, B_2^n) A H_{0,0}\| < \infty$$

Theorem 1 follows directly from (4) and the requirement that the discrete impulse response be absolutely summable. Since  $h(m,n)$  is a scalar, its matrix norm is equivalent to its absolute value and the proof of theorem 1 is obvious.

**Theorem 2:** The discrete LSI system represented by (2) and for which the numerator and denominator polynomials of the corresponding transfer function are mutually prime is unstable if any one of the spectral radii  $\rho(B_1)$ ,  $\rho(B_2)$  or  $\rho(B_1+B_2)$  is greater than or equal to one. The proof of Theorem 2 is given in the Appendix.

In the practical application of two-dimensional digital recursive filters,

any filter with  $\rho(B_1)$ ,  $\rho(B_2)$  or  $\rho(B_1+B_2)$  equal to one can be considered to be unstable and should be avoided[8]. Goodman [5] has shown by clever examples that two-dimensional filters with nonessential singularities of the second kind on the unit bidisc may be stable. Such a filter may have  $\rho(B_1)$ ,  $\rho(B_2)$  or  $\rho(B_1+B_2)$  equal to one. However, roundoff errors and coefficient truncation would prevent satisfactory performance by such a filter for most applications.

Several other theorems relating to sufficient conditions for stability have been found [7]. However, it has been shown that these constraints are too restrictive for general use. That is, useful stable filters can be found which do not satisfy the corresponding sufficient conditions for stability.

Computer algorithms are readily available to find the spectral radius of a matrix with real coefficients. Thus, Theorem 2 presents a convenient and easily implemented technique to assess the stability of two-dimensional digital recursive filters.

## APPENDIX

In this appendix, the proofs for Lemmas 1 and 2 and Theorem 2 are given. Since all vector norms are equivalent, any convenient norm may be used for either input or output.

## A1. Proof of Lemma 1

We proceed with a proof by induction. If we use (2) and (3) to obtain  $G_{J+1,K}$ ,  $G_{J,K+1}$  and  $G_{J+1,K+1}$  for input vector  $H_{J,K}$  and all initial condition vectors are null vectors, we obtain

$$\begin{aligned} G_{J+1,K} + B_1 G_{J,K} &= B_1 A H_{J,K} \\ G_{J,K+1} &= B_2 G_{J,K} = B_2 A H_{J,K} \\ G_{J+1,K+1} &= B_1 G_{H,K+1} + B_2 G_{K+1,J} = (B_1 B_2 + B_2 B_1) A H_{J,K} \end{aligned} \quad (A1)$$

If we use Lemma 1, we obtain

$$\begin{aligned} G_{J+1,K} &= C(B_1^0, B_2^1) A H_{J,K} = B_1 A H_{J,K} \\ G_{J,K+1} &= C(B_1^0, B_2^1) A H_{J,K} = B_2 A H_{J,K} \\ G_{J+1,K+1} &= C(B_2^1, B_2^1) A H_{J,K} = (B_1 B_2 + B_2 B_1) A H_{J,K} \end{aligned} \quad (A2)$$

Thus for any arbitrary  $m$  and  $n$  such that  $m > J$  and  $n > K$ , we can use (2) to write

$$G_{m+1,n} = B_1 G_{m,n} + B_2 G_{m+1,n-1} \quad (A3)$$

Then using (4) to find expressions for  $G_{m,n}$  and  $G_{m+1,n-1}$ , we have

$$G_{m+1,n} = [B_1 C(B_1^{m-J}, B_2^{n-J}) + B_2 C(B_2^{m-J+1}, B_2^{n-K-1})] A H_{J,K} \quad (A4)$$

Consider the term,  $C(B_1^J, B_2^K)$ . All of the products in the term either have  $B_1$  as the first factor or  $B_2$  as the first factor. If  $B_1$  is the first factor we must postmultiply by the sum of all possible products such that the power of  $B_1$  is decreased by one. If  $B_2$  occurs as the first factor, we must postmultiply by the sum all possible products such that the power of  $B_2$  is decreased by one. We conclude that

$$C(B_1^J, B_2^K) = B_1 C(B_1^{J-1}, B_2^K) + B_2 C(B_1, B_2^{K-1}), \quad (A5)$$

for all  $J$  and  $K$  such that both  $J$  and  $K$  are greater than or equal to one. It follows directly that

$$G_{m+1, n} = C(B_1^{m+1-J}, B_2^{n-K}) A_{H_{J, K}}. \quad (A6)$$

Similarly from (2) we write

$$G_{m, n+1} = B_1 G_{m-1, n} + B_2 G_{m, n}. \quad (A7)$$

Using (4) to find expressions for  $G_{m-1, n+1}$  and  $G_{m, n}$ , we have

$$G_{m, n+1} = [B_1 C(B_1^{m-J-1}, B_2^{n+1-K}) + B_2 C(B_1^{m-J}, B_2^{n-K})] A_{H_{J, K}}. \quad (A8)$$

It follows that

$$G_{m, n+1} = C(B_1^{m-J}, B_2^{n+1-K}) A_{H_{J, K}} \quad (A9)$$

Finally, from (2) we obtain

$$G_{m+1, n+1} = B_1 G_{m, n+1} + B_2 G_{m+1, n} \quad (A10)$$

Using Lemma 1 to express  $G_{m, n+1}$  and  $G_{m+1, n}$  we obtain

$$G_{m+1, n+1} = [B_1 C(B_1^{m-J}, B_2^{n+1-K}) + B_2 C(B_1^{m+1-J}, B_2^{n-K})] A_{H_{J, K}} \quad (A11)$$

It follows from (A5) and (A11) that

$$G_{m+1, n+1} = C(B_1^{m+1-J}, B_2^{n+1-K}) A_{H_{J, K}} \quad (A12)$$

and Lemma 1 holds.

### A2. Proof of Lemma 2

In the proof of Lemma 2, we shall show that if the response to a particular sequence of input vectors can be represented as given in Lemma 2, then the system is unstable if  $\rho(Q) > 1$  [9].

Define the eigenvalue corresponding to the spectral radius of  $Q$  as  $\lambda_Q$  and the corresponding eigenvector as  $P_Q$ . Then if the system transfer function has mutually prime numerator and denominator polynomials we can select an input vector such that

$$AF_{J,K} = \epsilon P_Q + R_{J,K} \text{ for all } J \text{ and } K \quad (A13)$$

where  $\epsilon$  is an arbitrary nonzero finite constant and  $R_{J,K}$  is not in the direction of  $P_Q$ . We then have

$$G_{m,n} = Q^m AF_{J,K} = \epsilon Q^m P_Q + Q^m R_{J,K} \quad (A14)$$

Then since  $\lambda_Q$  is the eigenvalue corresponding to the spectral radius, the norm of  $G_{m,n}$  is dominated by the term  $\epsilon Q^m P_Q$  in the limit as  $m$  approaches infinity.

Thus

$$S = \lim_{m \rightarrow \infty} \|G_{m,n}\| = \lim_{m \rightarrow \infty} \|\epsilon Q^m P_Q\| = \lim_{m \rightarrow \infty} \|\epsilon \lambda_Q^m P_Q\|$$

Note that  $S$  is infinite if  $\lambda_Q$  is greater than one and Lemma 2 holds.

### A3. Proof of Theorem 2

For this proof, we show that we can find a particular sequence of inputs that give unbounded output if either of the spectral radii specified in Theorem 2 is greater than one.

From Lemma 1 and 2 the output from a single arbitrary bounded input at the (J,K) position can be given by

$$G_{M,N} = f(J,K) C(B_1^{M-J}, B_2^{N-K}) A_{H_{J,K}} \quad (A16)$$

$$g(M,N) = DG_{M,N}$$

where  $F(J,K)$  is the scalar input at the (J,K) position. If we let  $K=N$  and  $J=0$  in (A16), we have

$$G_{M,N} = F(0,N) C(B_1^M, B_2^0) A_{H_{J,K}} = f(0,N) B_1^M A_{H_{J,K}} \quad (A17)$$

If we apply Lemma 2, we see that the system is unstable if  $\rho(B_1) > 1$ . If we let  $J = M$  and  $K = 0$  in (A16), we have

$$G_{M,N} = f(M,0) C(B_1^0, B_2^N) A_{H_{J,K}} = f(M,0) B_2^N A_{H_{J,K}} \quad (A18)$$

If we apply Lemma 2, we see that the system is unstable if  $\rho(B_2) > 1$ .

If we use a particular sequence of inputs  $f(J, M-J)$  for  $0 \leq J \leq M$  where all  $f(J, M-J)$  are bounded and equal. Using the principle of superposition and (A16) we have

$$G_{M,N} = \sum_{J=0}^M f(J, M-J) C(B_1^{M-J}, B_2^J) A_{H_{J, M-J}} \quad (A19)$$

Since all inputs are equal, we can write

$$G_{M,N} = f(0,M) \left[ \sum_{J=0}^M C(B_1^{M-J}, B_2^J) \right] A_{H_{0,M}} \quad (A20)$$

$$G_{M,M} = f(0,M) (B_1 + B_2)^M A_{H_{0,M}} \quad (A21)$$

If we apply Lemma 2, we see that the system is unstable if  $\rho(B_1 + B_2) > 1$ .

#### ACKNOWLEDGEMENT

The author wishes to thank Professor E. I. Jury of the Department of

Electrical Engineering, University of California Berkley and the reviewers of this paper for supplying helpful comments and references.

## REFERENCES

- [1] Lawrence R. Rabiner and Bernard Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975, pp. 442-455.
- [2] Samuel Stearns, Digital Signal Analysis, Hayden Publishing Company, 1975, p. 134.
- [3] E. I. Jury, "Theory and Application of Inners", Proc. IEEE, Vol. 63 No. 7, 1975, pp. 1044-1068.
- [4] Katsuhiko Ogata, State Space analysis of Control Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1967, p. 487.
- [5] Dennis Goodman, "Some Stability Properties of Two-Dimensional Linear Shift-Invariant Digital Filters", IEEE Transactions on Circuits and Systems, Vol CAS24, No. 4, 1977, pp. 201-208.
- [6] E. Fornasini and G. Marchesini, "State Space Realization Theory for Two-Dimensional Filters", IEEE Transactions on Automatic Control, Vol. AU 197, pp. 484-492.
- [7] Winser E. Alexander, Stability and Synthesis of Two-Dimensional Digital Recursive Filters, Ph.D. dissertation, University of New Mexico, Albuquerque, N. M., 1974 (University Microfilms, ann Arbor, Mich.).
- [8] N. K. Bose, "Problems and Progress in Multidimensional System Theory", Proc. IEEE, Vol. 65, No. 6, 1977, pp. 824-840.
- [9] Alston S. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell Publishing Company, New York, N. Y., 1964, Chapter 2.

**INDENTER HARDNESS STUDIES**

by

**George J. Filatovs**

## TABLE OF CONTENTS

	Page
Introduction .....	46
Hardness Testing .....	47
Experimental Conjectures .....	49
Experimental Material .....	53
Experimental Apparatus .....	53
Results .....	57
Discussion .....	57
Acknowledgment .....	64
References .....	66
Appendix 1 .....	68

## INTRODUCTION

Technology would be much advantaged by the availability of better methods for anticipating and preventing the mechanical failure of ceramics. In particular, the dependence of mechanical properties on the microstructural character is so complex, and ceramic microstructures so inherently variable, that the technological use of ceramics has been persistently retarded.

The field of mechanical properties is vast, therefore this phase addressed itself to a well defined and limited objective; to acquire, through literature reviews and laboratory testing, and understanding of the principles of microhardness testing of ceramics. This understanding was then used to devise a modification of the microhardness test which allowed the extraction of additional information.

The isolation of the hardness test as the centerpiece for this task can be easily defended. Hardness tests have long been used for classification and surveying of various materials, and their application to ceramics promises to be useful, for example to determine fracture toughness. And in a general way, indentation serves as a model system for studying the strength degradation in a wide range of phenomena such as machining flaws and impact abrasion.

## HARDNESS TESTING

The hardness test has long suggested itself as a simple means of obtaining mechanical properties: the ease of the test and the small sample size needed being the principal inducements. As a result, a number of technically valuable correlations have been established, for example, the tensile strength-Brinell hardness formula for certain metals. Unfortunately, the hardness test has not become a routine tool for investigating ceramic materials. The principal factors in this was the controversy in the relative roles of plastic deformation and cracking, (it was not until the 70's that plasticity was accepted as an important process), and the lack of theoretical underpinning to sort out confusing results.

The mechanics of point indentation has been slowly developing. As early as 1881 Hertz<sup>(1)</sup> analyzed the general elastic contact between two curved bodies, and in 1885 Boussinesq<sup>(2)</sup> solved the stress field for the case of an infinitely sharp indenter on a flat surface. However, as the stress field produced in non-homogeneous, anisotropic crystalline materials is complex, and there are no solutions for stress fields in terms of general geometry for an elastic indenter, it seems unlikely that any solution will emerge. Anyone who has made hardness tests on ceramics will have been impressed with the diversity of results which arise from minor variations in test conditions.

Nevertheless, a number of models have been devised to understand the indentation of actual materials, containing drastic simplifications and based on systematic studies of ceramic materials<sup>(3,4)</sup>. Most of

these models have in common the inclusion of elastic-plastic processes, and the assumption of spherical symmetry; in addition, the initiation and propagation of cracks are usually treated as separate events. There are also two extreme categories of contact situations; blunt and sharp. For blunt indentors<sup>(5)</sup>, the crack nucleates from pre-existing flaws and develops into a Hertzian cone, while sharp indentors<sup>(6)</sup> nucleate cracks from the plastic zone at the contact, which then develops into half-penny cracks.

Recent theoretical and experimental results have attempted to establish the actual macroscopic events in indentation fracture. The post-mortem examination of fracture surfaces and indentation impressions have been important in directing theoretical attempts. Perhaps the most dramatic development is the apparent correlation between fracture mirror patterns and fracture stress for glass and ceramics<sup>(7-12)</sup>. Unfortunately, after much intense study and confirmation of the general empirical relationships, it appears that these relationships may have no useful fundamental implications<sup>(13-15)</sup>.

Notwithstanding such analytical limitations, a number of studies have established certain common features. The greatest attention has been directed to the blunt indenter; the crack systems, their nucleation, propagation, and geometry have been studied<sup>(16-21)</sup>. Generally, the blunt indenter probes the cleavage tendencies. The effect of loads<sup>(22)</sup> and indenter angle (sharpness)<sup>(3,6)</sup> have been studied. The theoretical understanding of the crack growth in these tests has been primarily based on the Griffith energy criterion.

The sharp indenter is probably more pertinent to actual contact

situations, such as grinding and hardness testing. The principal difference from the blunt indenter is the plastic flow preceding the formation of the crack and the complex stress field. Once the crack system develops, the influence of indenter geometry becomes less important. Specific discussion of the sharp indentation studies will be in terms of the hardness test, which will be considered a semi-sharp indenter.

In summary, while there have been determined various functional relationships for indentation processes, the theoretical description remains incomplete.

#### HARDNESS TESTING OF CERAMICS

In spite of the formidable theoretical and experimental difficulties a number of attempts have been made to use the hardness test on ceramics. Most of these have involved the post-mortem examination of crack patterns and dimensions, and the indenter has most frequently been the Vickers diamond pyramid. This indenter is usually considered a semi-sharp indenter and its indentation stress field is favorable for plastic flow. Specific examples are the measurement of stresses in tempered glass surfaces<sup>(23)</sup>, fracture toughness determination<sup>(24)</sup>, and compressive strengths<sup>(25)</sup>. The results of some of these will be referred to in the discussion of experimental results.

#### EXPERIMENTAL CONJECTURES

The experimental idea developed here is an extension of the attempts to use the features of the hardness impression such as the extent and pattern of cracking. The surface crack pattern as a clue to fracture toughness was originated by Palmquist in 1957<sup>(27)</sup> and since has been considerably refined in theory and procedure<sup>(28-29)</sup>. The shortcoming

of these methods is that measurement of the crack features is difficult and unreliable, usually requiring SEM and etching techniques<sup>(30-32)</sup>. A simpler and less ambiguous method is desirable.

Although there is no solution of the stress field for hardness indentors, the deformation-fracture for well developed cracks in brittle materials is known, and can be used to orient our thoughts. Fig. 1 shows a Vickers Diamond Pyramid Indentor (DPI) and the primary crack pattern. The sharp edges of the indentor tend to initiate and develop half-penny cracks along the diagonals. Therefore, to prove the extent of these cracks it seems reasonable to search for some interaction of the cracks with stress singularities such as a surface or another indentation. For example, as shown in Fig. 2, for the orientation of indentor shown, the cracks would extend to the surface, and the crack length would be unambiguously revealed when breakthrough at the edge occurred. Another possibility is shown in Fig. 3, where the distance between successive impressions is varied until chipping occurs.

Figure 1. Diamond Pyramid Indentation and Resulting Crack Pattern

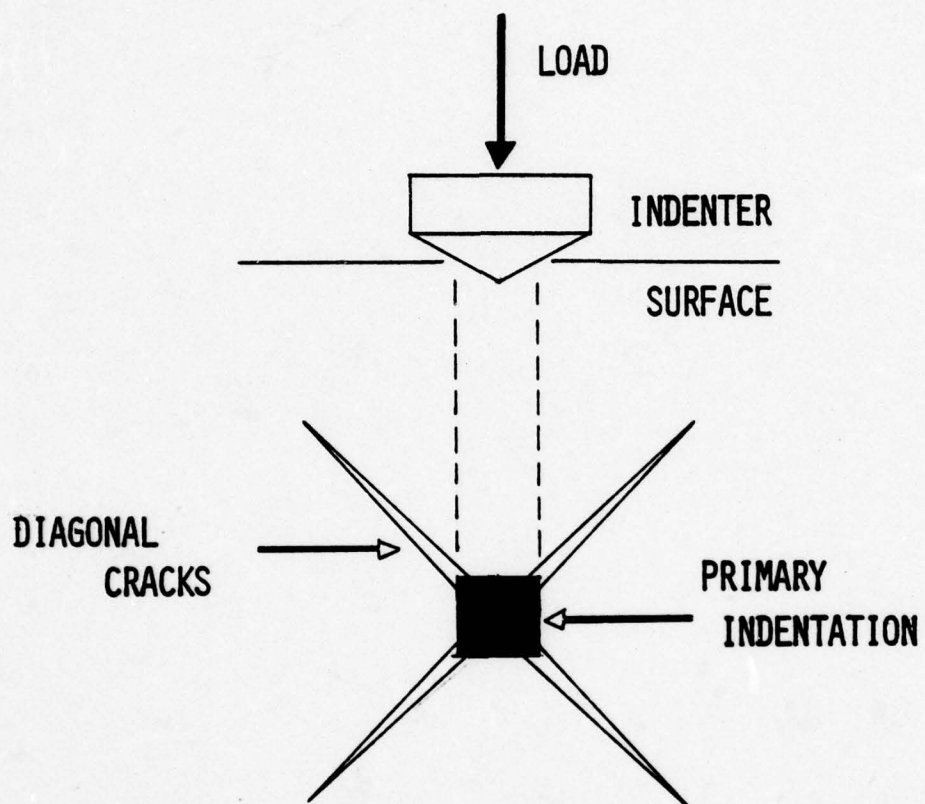
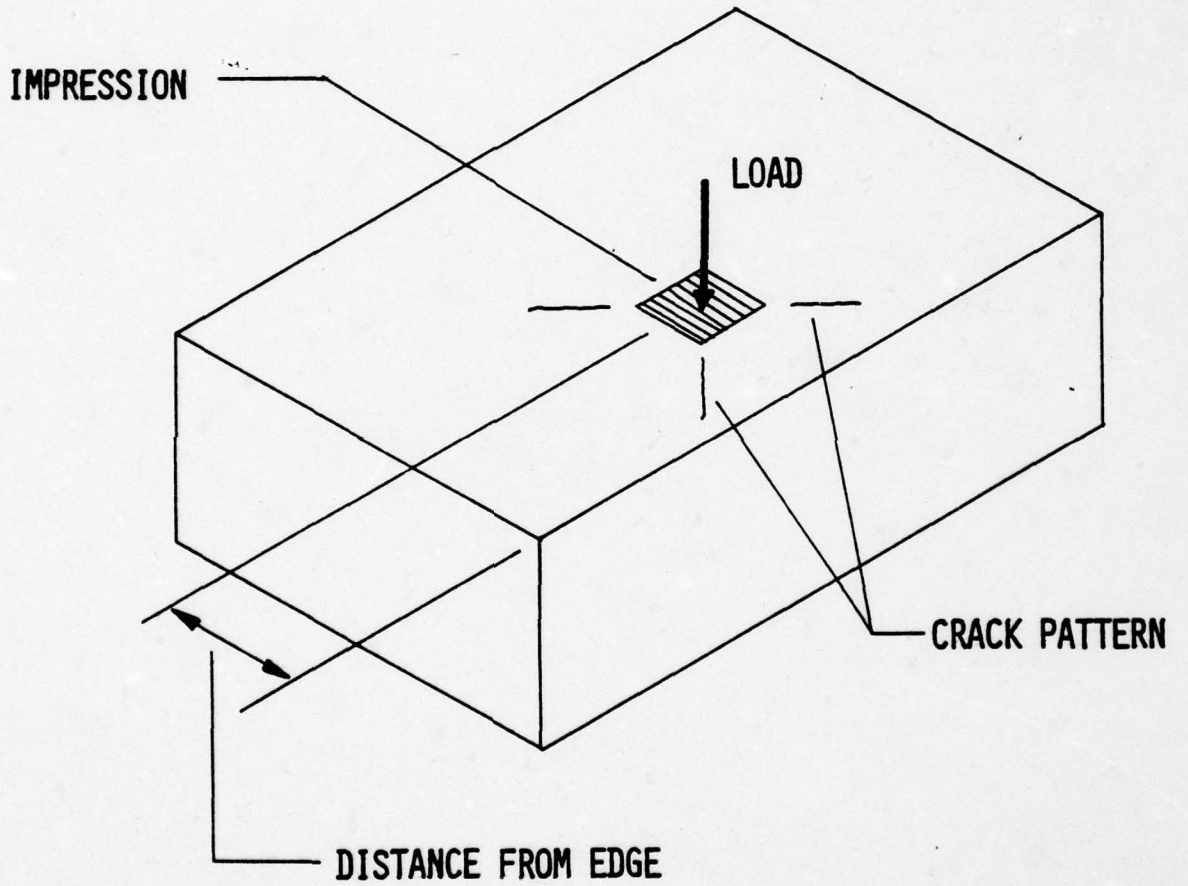


Figure 2. Diamond Pyramid Indentation made close to an edge. Cracks break out to the perpendicular surface and cause chipping.



## EXPERIMENTAL MATERIAL

The materials used were remnants from an evaluation program on ceramic vane materials conducted by IIT Research Institute and the Air Force Materials Laboratory and were extensively characterized by them<sup>(33-35)</sup>. The materials are briefly described below, and some properties summarized in Table 1. For testing by North Carolina A & T State University, the surfaces were polished with diamond paste, final polishing being with 1 -  $\mu$  diamond.

NC-350: Reaction bonded  $\text{Si}_3\text{N}_4$ . The microstructure has 25% porosity, with the porosity and silicon nitride phase uniformly distributed. The porosity was one-half to one-third open. The phases present were  $\alpha$ - $\text{Si}_3\text{N}_4$  (major) and  $\beta$ - $\text{Si}_3\text{N}_4$  (minor).

NC-435: Siliconized SiC. A two-phase material with about 20% Silicon. The SiC phase is the  $\alpha$ -form, and low porosity which is mostly closed.

NC-132: Hot pressed  $\text{Si}_3\text{N}_4$ . The phases present were  $\alpha$ - $\text{Si}_3\text{N}_4$ , the major phase, and  $\text{Si}_2\text{N}_4\text{O}$ , the minor phase. There were also traces of WC. The microstructure showed fine, elongated grains with virtually no open porosity.

## EXPERIMENTAL APPARATUS

The tests were conducted on an Kentron microhardness tester, using a 136° Diamond Pyramid Indentor. This indentor is cut in the shape of a square-based pyramid having an apex angle of 136°. The loads were applied for 15 seconds, and care was taken to reduce machine vibration. The tests were

conducted in a laboratory environment at room temperature. Considerable practice was initially expended in learning to obtain consistent results.

Figure 3. Expected chipping between two correctly oriented Diamond Pyramid Indentations.

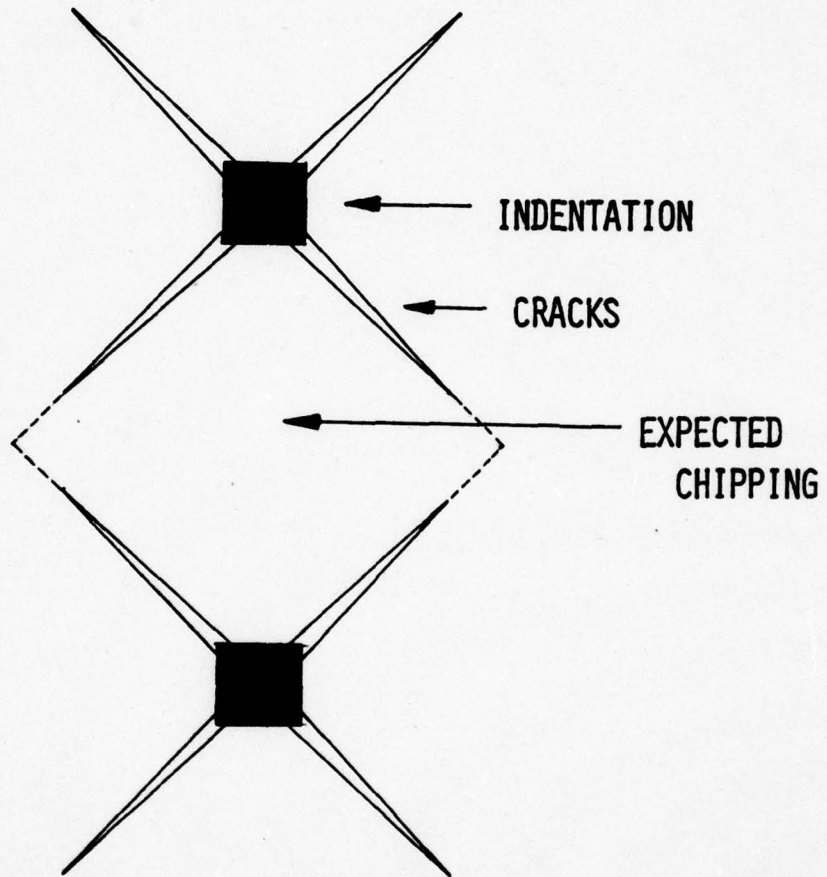


Table 1

<u>Material</u>	<u>Batch</u>	<u>Fracture Stress</u>	<u>Modulus</u>	<u>Density</u>
NC-350	1	30,860 psi	25.5x10 psi	2.52 gm/cc
	2	23,530	23.5	2.40
	3	33,960	27.7	2.54
NC-435	1	50,460	53.9	2.936
	3	65,980	49.5	2.997
	4	55,150	48.8	2.962
NC-132	1	90,910	48.4	3.177
	2	115,210	45.7	3.186

## RESULTS

Based on the previous speculations, a number of different indentation procedures were tried. The two-indentor method did yield some success; however, the experimental procedure proved troublesome. At least on the machine used, it proved difficult to line up the impression accurately, and so not enough usable data resulted. The test using an edge, shown in Fig. 3, gave better results. In this test, the distance from an edge which caused breaking of the edge at a particular load, was proved. The results are given in Tables 2 and 3. No results are included for NC-132, as this material frequently deformed by flaking rather than forming an impression and cracking. A few usable data points indicated that the results for this material followed the trends set by the other materials. This material is included because it apparently represents the upper strength limits for materials which can be used for the distance from the edge test, although anisotropy might be playing a role.

## DISCUSSION

The data of Table 2 are plotted in Fig. 4 is the fracture stress as determined by four point bending from Table 1. While there is no single straight line which appears to fit this data well, it is possible that a line can be fitted through each of the sets of points for the two materials. Certainly the hardness test appeared to rank the materials correctly, both as to batch and type; this certainly indicates some accuracy and shows that surface finish, machine operation and microstructural variability are not scattering the data beyond use.

The data from Table 3 are shown plotted on log-log paper in Fig. 5. Comparison of this data with that in Table 1 shows that the materials are

Table 2

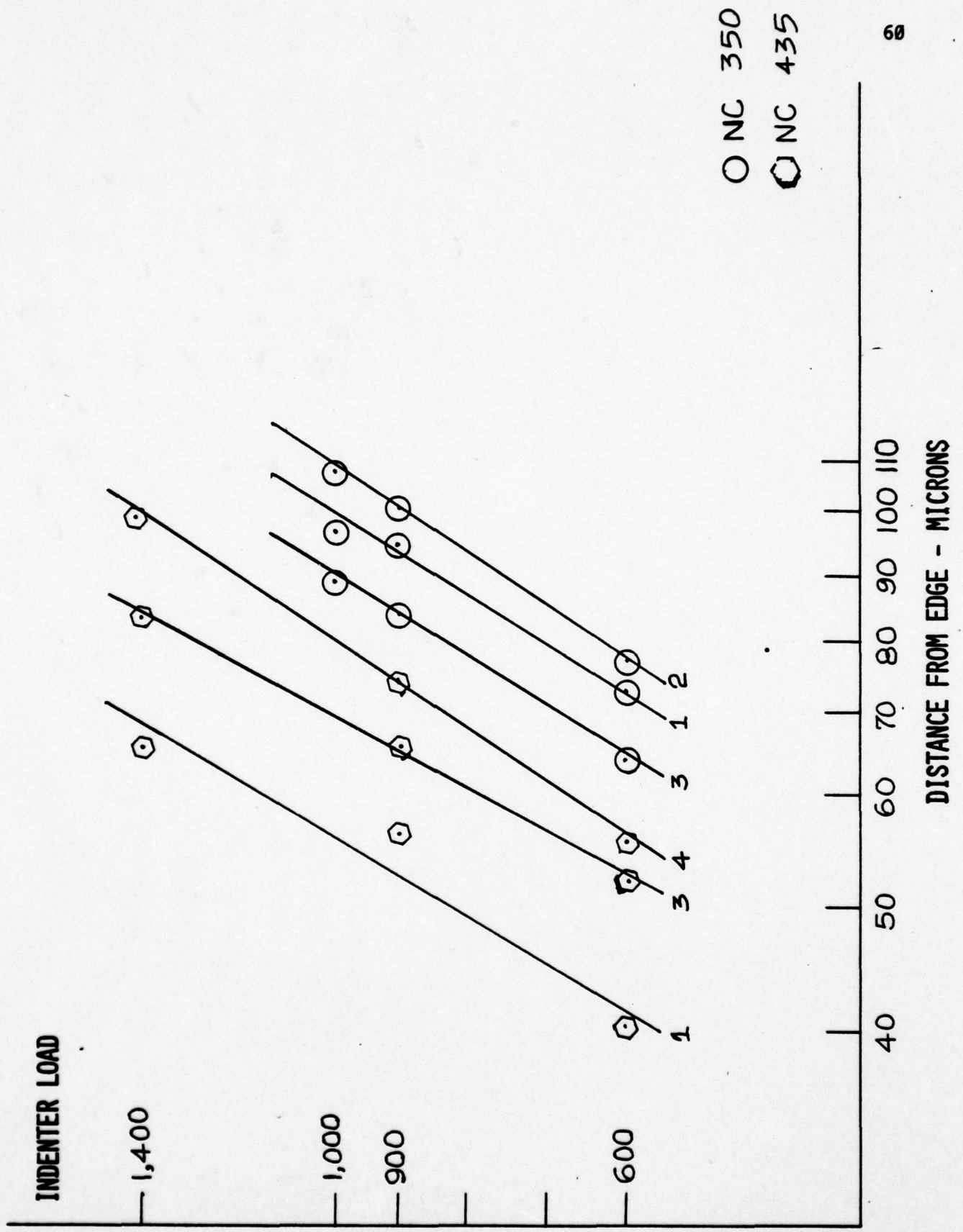
<u>Material</u>	<u>Batch</u>	<u>Hardness*</u>
NC-350	1	859 DPN
	2	662
	3	1087
NC-435	1	1086
	3	1780
	4	1646

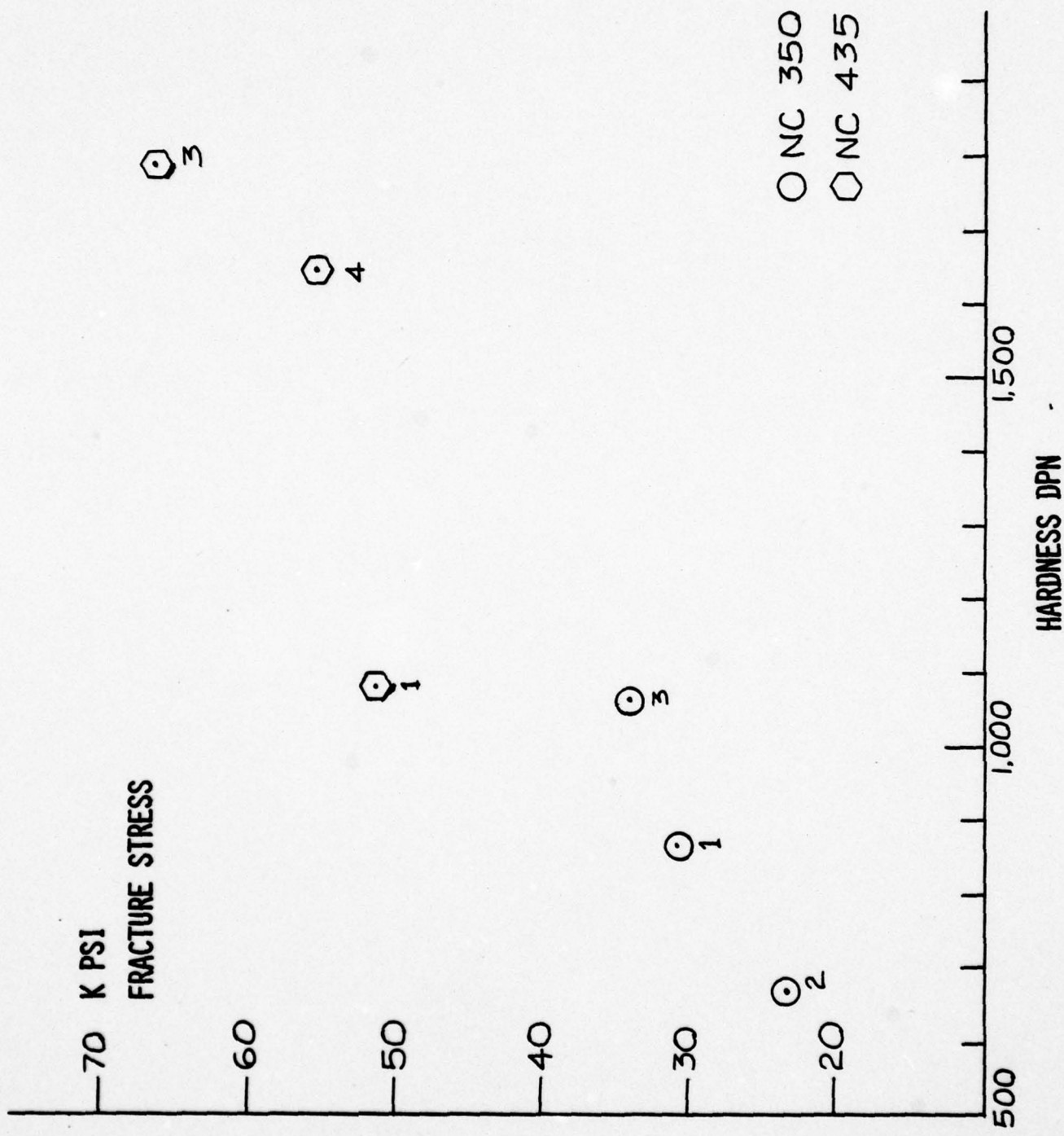
\*Average of 10 readings, 600 gram load.

Table 3

<u>Material</u>	<u>Batch</u>	<u>Load</u>	<u>Distance from Edge*</u>
NC-350	1	600 grams	72 microns
	2		76
	3		64
NC-435	1	600	39
	3		52
	4		56
NC-350	1	900	94
	2		99
	3		83
NC-435	1	900	56
	3		65
	4		74
NC-350	1	1000	196
	2		105
	3		88
NC-435	1	1400	65
	3		82
	4		97

\*Average of 10 readings.





70 K PSI  
FRACTURE STRESS

HARDNESS DPN

○ NC 350  
⬡ NC 435

ranked by modulus rather than by strength. In trying to understand this and other features about this data, we need some theoretical underpinning. The first obvious correlation to look for is some analogy to the mirror boundary relationship<sup>(9,10,14)</sup>

$$\sigma_f r_m^{1/2} = A$$

where  $\sigma_f$  is the fracture stress,  $r_m$  the fracture mirror radius, and A the mirror constant. Although, as mentioned previously, the theoretical basis for this relationship has been denied, it has nevertheless been confirmed by many studies, and so there is some expectation that the present data would fit this equation. If the distance from the edge is taken as an indication of the mirror size, then the functional dependence in the above equation must be reversed. That is, the mirror boundary represents the surface created by the fracture stress, so that a strong material would create a large fracture, while in the present test, as the load is constant (or normalized), the stronger material will break closer to the edge and thus create a smaller area.

Even though the functional dependence is correct, the data slope is not the required 1/2, and no amount of manipulation, for converting the distances to areas, can make it conform to the mirror boundary equation.

A second approach is to consider the distance from the edge as proportional to the characteristic crack dimension. This would mean that

the crack extension stage would predominate, and that the force would be proportional to the  $3/2$  power of the crack dimension (or distance from the edge), as expected for a half-penny crack<sup>(34,35)</sup>. The data does appear to cluster around a  $3/2$  slope, although it is impossible to know if the scatter in slopes is due to test or material variability, or is a natural material function. If the distance from the edge is truly an indicator of the force necessary to extend a crack, then this method may eventually make it possible to determine such factors as the fracture surface energy. IITRI determined  $K_{Ic}$  by the double torsion method for the materials tested here. See Table 4. Those values correlate to the ranking determined here, if the small differences between batches 3 and 1 are ignored.

If we then review all of the data, the strengths of these materials correlate with the densities, which correlate with the hardness, and the modulus correlates with the  $K_{Ic}$  which correlates with the distance from the edge. Also, a plot of the distance from the edge vs load gives roughly the slope  $3/2$  for all batches, the relationship between the force and extension of a half-penny crack. Intriguing as these results are, it would be overoptimistic at this time to suggest that anything other than a method for ranking materials or batches, by modulus or  $K_{Ic}$ , has been developed. An extremely limited range of materials was tested, and as the difficulties with the NC-132 material showed, some materials may be difficult to accommodate to this test. It is hoped that these results will encourage further exploitation of the hardness test.

**ACKNOWLEDGMENT**

We are indebted to Dr. R. Ruh of the Air Force Materials Laboratory and Dr. G. C. Walther of the IIT Research Institute for supplying the test materials and to Dr. H. P. Leighly of the University Missouri-Rolla for allowing the use of the Hardness tester.

Table 4

<u>Material</u>	<u>Batch</u>	<u>K</u>
NC-350	1	1.88 MN
	2	2.15
	3	2.08
NC-435	1	3.93
	3	3.91
	4	3.40

## REFERENCES

- 1) H. Hertz, "Hertz's Miscellaneous Papers", MacMillan, London 1896.
- 2) J. Boussinesq, Discussed in "Theory of Elasticity", by Timoshenko and Goodier, McGraw-Hill, 1970. p398.
- 3) B. Lawn and R. Wilshaw, J. Mat. Sci., 10, 1049(1975). This is an extensive review paper.
- 4) B. Lawn and M. Swain, J. Mat. Sci., 10, 113(1975).
- 5) B. Lawn, S. M. Wiederhorn, and H. H. Johnson, J. Amer. Ceram. Soc., 58 (9-10), 428(1975).
- 6) B. Lawn, E. R. Fuller, and S. M. Wiederhorn, J. Amer. Ceram. Soc., 59 (5-6), 193(1976).
- 7) J. R. Varner and H. J. Oel, Glass. Tech. Ber., 48, 73(1975).
- 8) D. A. Krohn and D. P. H. Hasselman, J. Amer. Ceram. Soc., 54, 54(1971).
- 9) J. J. Melcholsky, S. W. Freiman, and R. W. Rice, J. Mater. Sci., 11, 1310(1976).
- 10) J. J. Melcholsky, R. W. Rice, and S. W. Freiman, J. Amer. Ceram. Soc., 57, 440(1974).
- 11) N. Shinkai, Jap. J. Appl. Phys. 14, 147(1975).
- 12) J. Congleton and N. J. Petch, Phil. Mag., 16,749(1967).
- 13) K. R. McKinney, J. Amer. Ceram. Soc., 56, 115(1973).
- 14) A. I. A. Abdel-Latif, R. C. Bradt, and R. E. Tressler, Int. J. Of Fract. 13(3), 349(1977).
- 15) L. R. F. Rose, Int. J. of Fract. 12(6), 799(1976). This is a review paper.
- 16) F. C. Frank and B. R. Lawn, Proc. Roy. Soc. Lond., A299, 291(1967).
- 17) B. R. Lawn, J. Appl. Phys., 39, 4828(1968).
- 18) A. G. Mikosza and B. R. Lawn, J. Appl. Phys., 42, 5540(1971).
- 19) T. R. Wilshaw, J. Phys. D:Appl. Phys., 4, 1567(1971).
- 20) J. S. Williams, B. R. Lawn, and M. V. Swain, Phys. Stat. Soli(a), 2

7(1979).

- 21) N. E. W. Hartley and T. R. Wilshaw, *J. Mater. Sci.*, 8, 265(1973).
- 22) K. Phillips, Ph.D. Thesis, University of Sussex, 1975.
- 23) D. B. Marshall and B. R. Lawn, *J. Amer. Ceram. Soc.*, 60(1-2), 86(1977).
- 24) A. G. Evans and E. A. Charles, *J. Amer. Ceram. Soc.*, 59(7-8), 371(1976).
- 25) R. W. Rice, "The Science of Hardness Testing and Its Research Applications", ASM, 1973.
- 26) S. Palmqvist, *Jernkontorets Ann.*, 141, 300(1957).
- 27) J. H. Westbrook, *J. Amer. Ceram. Soc.*, 41, 433(1958).
- 28) H. E. Exner, *Trans. Met. Soc. AIME*, 245, 677(1969).
- 29) V. M. Glazov and V. N. Vigdorovich, "Microhardness of Metals and Semiconductors", Consultants Bureau, 1979.
- 30) E. K. Pavelcheck and R. H. Doremus, *J. Mater. Sci.*, 9, 1802(1974).
- 31) Idem, *J. Appl. Phys.*, 46, 4096(1975).
- 32) R. H. Doremus and W. A. Johnson, *J. Mater. Sci.*, 13, 855(1978).
- 33) D. C. Larsen, "Property Screening and Evaluation of Ceramic Vane Materials", Interim Technical Report 3. Contract No. F33615-75-C-5196.
- 34) D. C. Larsen and G. C. Walther, "Property Screening and Evaluation of Ceramic Vane Materials", Interim Report 4.
- 35) D. C. Larsen and G. C. Walther, "Property Screening and Evaluation of Ceramic Turbine Engine Materials", Interim Technical Report 5.
- 36) B. R. Lawn and E. R. Fuller, *J. Mater. Sci.*, 19, 2016(1975).
- 37) B. R. Lawn, T. Jesen, A. Arora, *J. Mater. Sci.*, 11, 573(1966).

APPENDIX 1



600 Grams

Microns for Edge

		AV			AV
41T1A	40		31T1A	80	
	39			73	
	35			79	
	35			59	
	33	39		76	72
	37			75	
	32			78	
	42			60	
	45			67	
	47			62	
44T1A	45		33T1A	59	
	50			61	
	60			69	
	62			66	
	55	56		71	64
	47			62	
	65			58	
	57			70	
	63			60	
	51			68	
43T1A	47		32T1A	68	
	50			81	
	53			77	
	61			73	
	45	52		70	76
	55			82	
	49			79	
	57			72	
	48			80	
	60			78	

900 Grams

## Microns for Edge

		AV		AV
41T1A	50		33T1A	101
	49			97
	61			88
	52	56		89
	49			91
	52			98
	60			95
	63			90
	65			91
	59			100
44T1A	81		33T1A	91
	72			86
	71	74		79
	78			80
	70			85
	68			90
	69			84
	80			76
	76			85
	75			78
43T1A	58		32T1A	94
	74			95
	71			101
	70	65		107
	63			104
	66			110
	57			92
	61			96
	63			89
	68			97

1,000 Grams

## Distance

		AV
31T1A	101	
	104	
	98	
	89	96
	91	
	91	
	93	
	95	
	98	
	101	
33T1A	95	
	92	
	94	
	82	88
	91	
	84	
	86	
	93	
	81	
	83	
32T1A	98	
	99	
	97	
	109	105
	108	
	102	
	114	
	97	
	112	
	110	

1,400 Grams LoadMicrons for Edge  
AV

41T1A	66	
	59	
	61	
	64	65
	70	
	68	
	62	
	71	
	67	
61		
43T1A	87	
	85	
	89	
	80	82
	78	
	79	
	86	
	81	
	76	
78		
44T1A	91	
	89	
	95	
	94	97
	103	
	102	
	98	
	96	
	100	
	102	

**CERAMIC FRACTURE ANALYSIS  
THROUGH BIAXIAL WEIBULL THEORY**

by

**William J. Craft**

## TABLE OF CONTENTS

	Page
Introduction .....	75
Development of Simple Test Analysis .....	77
Pure Bending Stress Field in a Rectangular Section Bar .....	79
Pure Bending for a Circular Cross-Section Rod .....	80
Pure Torsion for a Circular Bar .....	81
Fourier Analysis of Biaxial Data .....	82
Finite Element Analysis .....	84
Conclusion .....	85
References .....	86

## APPENDIX

Appendix 1 - Three Parameter Weibull Simple Tension Program .....	88
Appendix 2 - Fourie.For .....	90
Appendix 3 - Finite.For .....	94

## CERAMIC FRACTURE ANALYSIS THROUGH BIAXIAL WEIBULL THEORY

## INTRODUCTION

The second project of this investigation was entitled "Strength characterization of brittle materials by means of simple-model destructive testing and by indenter-initiated crack growth".

A simple destructive test is appealing due to: (1) its decisiveness in quantifying failure; (2) its ability, if successful, in transferrability of results to complex designs and shapes; (3) its incorporation of size effects; and (4) in that it requires inexpensive testing equipment.

Literature on ceramics largely resides in three camps, fracture mechanics, continuum mechanics, and statistical failure. Fracture mechanics predictions of failure couple with elasticity theory where strain energy release rate is equated with a critical flaw length or size [1-2] frequently on the basis of standard 'crack' models such as interface delamination in the case of composite material problems or as voids with sharp edges as in the case of the penny-shaped crack [2]. The difficulty in such approaches involving elasticity theory is the intractability of the field equations even for linear elastostatic problems of simple geometry [3,4].

Finite element and difference methods offer some advances in the numerical techniques [5,6], in particular when coupled with ancillary methods that predict the stress intensity factor [7,8]. These methods all fail, however, to give an account of the customarily larger experimental scatter in apparent failure strengths of numerous grainy brittle materials.

The classical work on statistical failure by Weibull [9-11] has been applied to numerous stochastic processes capacitor discharge, fatigue time life, and

others largely due to the ease of use, simplicity of assumptions and non-stochastic processes as ceramic fracture in relation to a size effect. The major problems posed by this theory are the mathematical determination of the Weibull parameters, in particular for the three parameter distribution, and in the application of the results in a meaningful manner to the elasticity solution in order that a failure probability for the body can be found.

In order to develop a technique that would utilize Weibull Theory and an elasticity solution to generate fracture probabilities for a body of brittle material, it was necessary to investigate what techniques were proposed and used to generate the Weibull parameters to begin with.

Sample earlier work was soon found based on moment generating methods and on rank-order theory [12,13] to differ with results obtained by several iterative methods whereby the sum of squares of deviations were minimized (residuals) as is the case of standard least-squares techniques. Moment generation can involve large truncation error and rank-order assumptions should be significantly less precise than comparison with cumulative totals of experimental data. Further, although least-squares analyses tends to emphasize deviations in the experimental curve ordinates, incorporation of appropriate weights can counter this effect while preserving the methods basic simplicity.

Another problem with earlier techniques is that an assumption was generally required for multiaxial stress states necessitated by the uniaxial stress state under which data had been taken. Many persons assumed statistical independence in the actions of principal direction stresses, i.e., the probability of failure of an element equal to the product of probabilities of failure due to each principal direction stress [12]. Some writers have proposed

extensions of the uniaxial Weibull Theory based on fracture mechanics where distributions of penny-shaped cracks initiate failure [14,15,16].

In this report it was proposed to couple the results of multiaxial tests, notably biaxial ones, with an analysis and development of Weibull parameters obtained at specific stress ratios. This would aid the utilization of experimental data without the need to develop further fracture mechanics assumptions. Further, it has been hoped that such a technique when coupled with effective testing of standard specimens and stress fields could help lend a standardization in the analysis of experimentally obtained statistical fracture.

#### DEVELOPMENT OF SIMPLE TEST ANALYSES

An extension to publications where standard specimen shapes are used to generate Weibull statistics [17,18] are developed below where the three parameter Weibull fit is used.

#### UNIAXIAL CONSTANT STRESS FIELDS

The Weibull distribution for a uniaxial constant stress state is given below:

$$F(\sigma) = 1 - \exp\left(-\kappa \int \left(\frac{\sigma - \sigma_u}{\sigma_0}\right)^m \frac{dU}{U_{un}}\right)$$

$U_{un}$  = Unit Volume

$\kappa, \sigma_u, \sigma_0, m$  = Weibull Parameters which can be expressed uniquely as three independent ones.

For uniaxial constant stress,  $\sigma$  is a principal stress, independent of the volume so that

$$F(\sigma) = 1 - \exp\left(-\frac{V_g}{V_{un}} \left(\frac{\sigma - \sigma_u}{\sigma_0}\right)^m\right)$$

Upon using  $C = \frac{V_g}{V_{un} (\sigma_0)^m}$

$$F(\sigma) = 1 - \exp(-c(\sigma - \sigma_u)^m)$$

The simplest least-squares solution is obtained by noting that taking logs of this equation twice gives:

$$\ln \ln \left( \frac{1}{1 - F(\sigma)} \right) = \ln(c) + m \ln(\sigma - \sigma_u)$$

This is a linear relationship allowing a linear least-square analysis to be invoked for

$$y = A + Bx$$

where  $A = \ln(c)$

$$B = m$$

$$x_i = X(\sigma_i) = \ln(\sigma_i - \sigma_u)$$

$$y(\sigma_i) = y_i = \ln \ln \left( \frac{1}{1 - F(\sigma_i)} \right)$$

$i = 1, 2, \dots, n$  - the number of data points.

The solution that minimizes the residual is:

$$A = \left( \sum w_i y_i x_i \right) \left( \sum w_i \right) - \left( \sum w_i x_i \right) \left( \sum w_i y_i \right) / D$$

$$B = \left( \sum w_i x_i^2 \right) \left( \sum w_i y_i \right) - \left( \sum w_i x_i \right) \left( \sum w_i x_i y_i \right) / D$$

$$D = \left( \sum w_i \right) \left( \sum w_i x_i^2 \right) - \left( \sum w_i x_i \right)^2$$

Where all summations are  $i = 1, \dots, n$ .

In case equal weights are desired, the quantity  $\sum w_i$  is replaced by  $n$ , otherwise,  $w_i = 1$ . The obvious difficulty with this analysis is that  $A$  and  $B$  are dependent on  $\sigma_u$ . To determine the best  $\sigma_u$ , an iterative programming scheme, Appendix I, was developed which recomputes the residual sum of squares and chooses the  $\sigma_u$  minimizing this function. By a slight modification this scheme can be used to redefine the weights iteratively to emulate any curve-fitting power law.

PURE BENDING STRESS FIELD IN A RECTANGULAR SECTION BAR

Tests utilizing bending stress distributions are common and generally less expensive than ceramic tension tests — particularly in the base of three and four point loading tests. Following the two parameter derivation [17], where the stress distribution is variable and must be integrated over the volume in bending, the probability of failure is given by:

$$P = 1 - \exp - \left( \frac{V}{2(m+1)} \left( 1 - \frac{\sigma_u}{\sigma_b} \right) \left( \frac{\sigma_b - \sigma_u}{\sigma_0} \right)^m \right)$$

Where  $V = bLh$

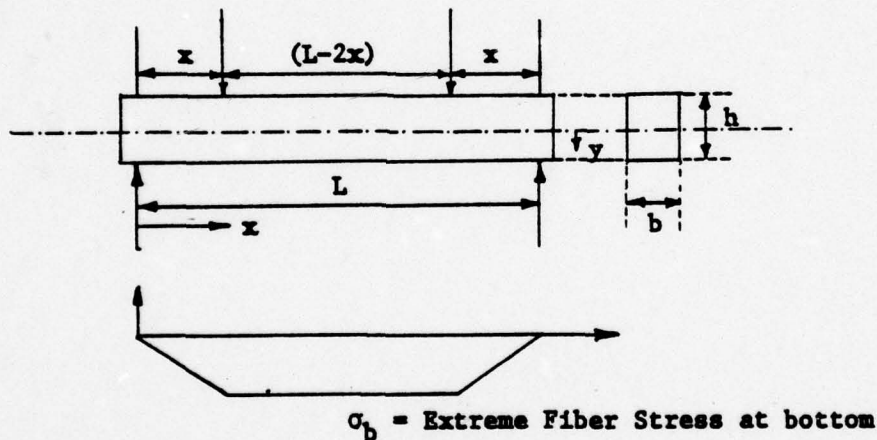


Figure 1. Prismatic beam under 4-point loading and distribution of extreme fiber stress

Fig. 1 depicts this case with its test volume. This distribution follows the same form of the case of a constant uniaxial stress field and may be solved iteratively by the technique where upon taking  $\ln \ln$ , one has

$$\ln \ln \left( \frac{1}{1 - P(\sigma_1)} \right) = \ln(V/2) - \ln(m+1) + (m+1) \ln(\sigma_{b1} - \sigma_u) - \ln(\sigma_{b1}) - m \ln(\sigma_0)$$

In this case

$$A = \ln(V/2) - \ln(m+1) - m \ln(\sigma_0)$$

$$B = m+1 .$$

#### PURE BENDING FOR A CIRCULAR CROSS-SECTION ROD

This case is related to the others except that the integration is conducted over a circular prism for the section in pure bending. The resulting integral form:

$$\int_{V_T} (my - \sigma_u I_y)^m \sqrt{R^2 - y^2} dy$$

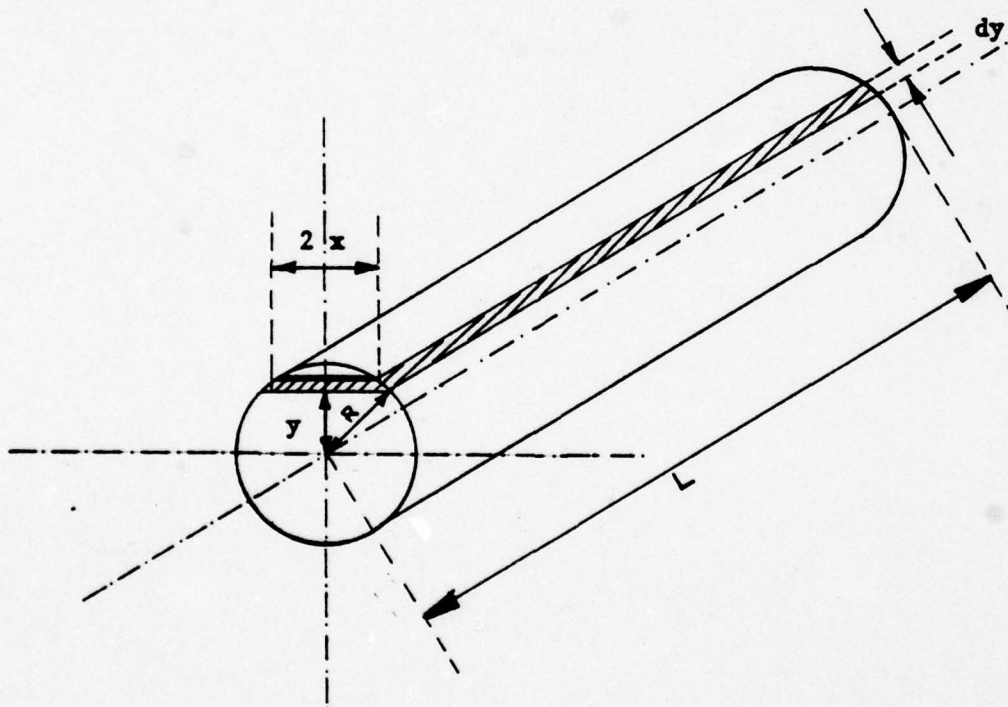


Figure 2. Bending for a circular cross-section rod

must be evaluated as a function of  $\sigma_u$  and  $M$  before the previously discussed  $\ln \ln$  least-square procedure can be applied. Fig. 2 depicts this case and its test volume,  $V_T$ .

### PURE TORSION FOR A CIRCULAR BAR

Under the action of pure torsion, the stress field  $\tau_{rz} = \frac{Tr}{J}$  results from which the principal direction stresses are:

$$\sigma_1 = \tau_{rz}, \quad \sigma_2 = -\tau_{rz} .$$

In order that a Weibull distribution based on this test can be evaluated, two parameters, discussed in the next section must be defined:

$$\sigma^2 = \sigma_1^2 + \sigma_2^2$$

$$\text{and } \theta = \text{Tan}^{-1} \left( \frac{\sigma_2}{\sigma_1} \right) = 135^\circ$$

They are the intensity of the biaxial stress field,  $\sigma$ , and the angle on the biaxial stress distribution that the point on the fracture surface makes with the  $\sigma_1$  axis, as depicted below:

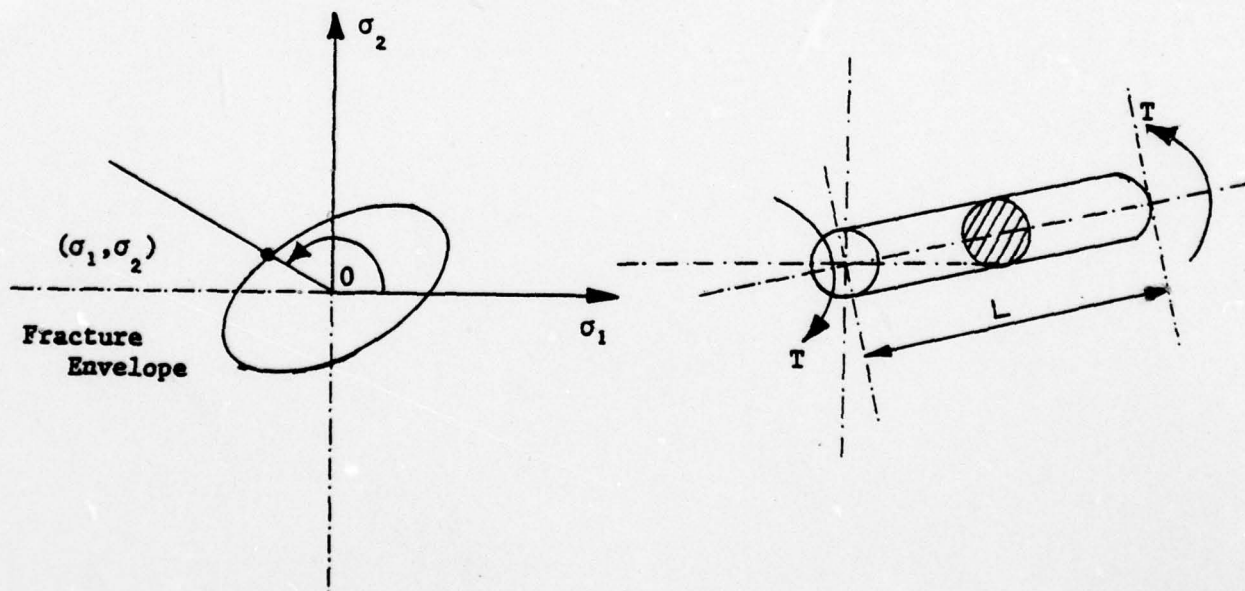


Fig. 3. Biaxial Stress Envelope

In this case, the stress distribution is integrated over the volume as before resulting in

$$P(\sigma) = 1 - e^{-B_n}$$

$$\text{Where } B_n = \frac{2\pi l}{(m+1)\sigma_0^m} \left\{ \frac{(\sigma - \sigma_n)^{m+1}}{\sigma} - \frac{(\sigma - \sigma_n)^{m+2}}{\sigma^2 (m+2)} \right\} \begin{matrix} R_0 \\ \frac{l\sigma_0}{T} \end{matrix}$$

This result has the same form as the case of the uniaxial tension field and can be correlated with the three parameter Weibull distribution by the Appendix I program.

#### FOURIER ANALYSIS OF BIAXIAL DATA

All three Weibull parameters are functions of the principal direction stress ratio or  $\theta$  where

$$\theta = \tan^{-1} \left( \frac{\sigma_2}{\sigma_1} \right)$$

Thus each Weibull parameter  $c, \sigma_0, m$  could be written by

$$c = c(\theta)$$

$$\sigma_0 = \sigma_0(\theta) \quad \text{or} \quad We = \left\{ \begin{matrix} c \\ \sigma_0 \\ m \end{matrix} \right\} = We(\theta)$$

$$m = m(\theta) = We(\theta \pm 2n\pi)$$

invoking periodicity we also have

$$We(\theta) = We(\theta \pm 2n\pi)$$

This same theory can be developed for the solid angle representation in

the case of three principal direction stress components; that extension is based on orthogonal polynomials and representation completeness of harmonic analysis [3,4,19].

Once the three Weibull parameters are known for a variety of  $\theta_1$ , the equivalent parameters can be obtained for any angle by means of trigonometric interpolation. The computer program FOURIE, Appendix II, requires the Weibull set  $W_{\theta_1}$  at a variety of  $\theta_1$ . At this point, the highest harmonic obtainable for the number of data points is used for the interpolation series or the highest harmonic is used which satisfies the null hypothesis condition [20]. The advantage of this program is that the coefficients to the Fourier harmonics, once generated, can be used for each element stress ratio without additional coefficient computation.

The computer program would naturally be much simpler with equally-spaced  $\theta_1$  data points but such is not sufficiently general, hence orthogonality cannot be invoked without the creation of special orthogonal sequences.

Basically given  $n$  ( $n$  odd) angles  $\theta_1$   $i=1, \dots, n$ , a set of functions can be generated with  $L = \frac{n-1}{2}$  to define  $y_m(\theta)$  where  $M \leq L$ . In this case:

$$y_m(\theta) = \frac{a_0^m}{2} + \sum_{j=1}^m a_j^m \cos(j, \theta) + b_j^m \sin(j, \theta) .$$

The residual function is:

$$H(a_0^m, \dots, a_m^m) = \sum_{i=1}^n w(\theta_1) \left\{ \bar{f}_1 - y_m(\theta) \right\}^2$$

This function is dependent on the weights  $w(\theta_1)$  associated with each Weibull 3-set  $W_{\theta_1} = \bar{f}_1$ . It can be shown that evaluation of the above, minimizing  $H$ , leads to the partitioned matrix equation:

$$\begin{array}{ccc|c|c}
 \frac{1}{2} \Sigma w_1 & \Sigma w_1 \cos(\theta_1) \dots & \Sigma w_1 \sin(\theta_1) \dots & a_0 & \Sigma w_1 \bar{r}_1 \\
 \frac{1}{2} \Sigma w_1 \cos(\theta_1) & \Sigma w_1 \cos^2(\theta_1) \dots & \Sigma w_1 \cos(\theta_1) \sin(\theta_1) & a_1 & \Sigma w_1 f_1 \cos(\theta_1) \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \frac{1}{2} \Sigma w_1 \sin(\theta_1) & \Sigma w_1 \cos(\theta_1) \sin(\theta_1) \dots & \Sigma w_1 \sin^2(\theta_1) \dots & b_1 & \Sigma w_1 f_1 \sin(\theta_1) \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & b_n & \vdots
 \end{array}$$

Summations are  $i = 1, \dots, n$ .

For brevity only the first term in each set is listed here. Modification of first row terms leads to symmetry of the coefficient matrix.

#### FINITE ELEMENT ANALYSIS

The remaining task in the use of biaxial Weibull statistics is the use of a general stress analysis method to find the stress distribution in a body so that this theory can be applied. The crux of this process is a finite element analysis model written by J. Brisbane [21] for bodies of revolution loaded axisymmetrically. This code has been modified to create files on the principal stresses for each element as well as volume for a particular material designation. The modified program, Appendix III, does generate triaxial stress data but for most loading conditions, one stress component, in or near the r-coordinate direction can usually be neglected.

As with all programs of this type, input information is comprehensive and complex. Only changes are described when they relate to the Weibull analysis. One additional parameter, the material number N is required, this is the number of the material subject to the Weibull modeling.

In order to analyse Weibull statistics, two additional disk files have been added, unit 43 and unit 47. Unit 43 stores the location of each model point of each element of material N. It also stores the volume on each record. Unit 47 records for each material N element, the principal stresses. These files are saved after completion of execution so that a Weibull summation can be performed in conjunction with the output of Appendix II. The program of Appendix requires data from the Appendix I program or a modification thereof.

#### CONCLUSION

During this next years program, this Weibull analysis will be performed and compared to data derived from experiments on test specimens. Each experiment is designed to provide Weibull parparameters for specific principal stress-ratios and will complement entirely the present work. In order to compare this biaxial theory to other methods, one test specimen is being designed with a variety of stress ratios present within its volume. Its fracture behavior will be noted and compared to predictions based on this and the independence of principal direction theory of failure. Thus what has remained is an experimental confirmation of this theory which should be provided within the 1978 - 1979 period.

## REFERENCES

- [1] Erdogan, F. and Ozbek, T., "Stresses in Fiber-Reinforced Composites with Imperfect Bonding," *Journal of Applied Mechanics*, Vol. 35, No. 4, 1969, p. 865.
- [2] Sueddon, I. N., Fourier Transforms, McGraw-Hill Book Co., Inc., New York, N. Y., 1951.
- [3] Blenkarn, Kenneth A., "Fourier Integral Solutions to Radially Symmetric Elasticity Problems," Ph.D. Dissertation in Mechanical Engineering, The Rice Institute, Houston, Texas, May, 1960.
- [4] Lure, A. I., Three-Dimensional Problems of the Theory of Elasticity, Interscience Publishers, a division of John Wiley & Sons, Inc., 1964.
- [5] Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, London, 1971.
- [6] Martin, Harold C. and Carey, Graham F., Introduction to Finite Element Analysis, McGraw-Hill Book Co., Inc., 1973.
- [7] Becker, E. B., and Dunham, R. S., "Some Stress Intensity Calculations Using Finite Elements," Finite Element Methods in Engineering, The University of New South Wales, Sydney, Australia, 1974.
- [8] Stern, Morris and Soni, Mahan Lal, "on the Computation of Stress Intensity Factors in Fibre Composite Media Using a Contour Integral Method," EMRL No. 1110, Department of Aerospace Engineering and Engineering Mechanics," University of Texas at Austin, Austin, Texas, August, 1974.
- [9] W. Weibull, "A Survey of Statistical Effects in the Field of Material Failure", *Applied Mechanics Reviews*, Vol. 5 #11, November, 1952.
- [10] W. Weibull, "A Statistical Distribution Function of Wide Applicability," *Journal of Applied Mechanics*, September, 1951.
- [11] "Weibull's Statistical Theory of Strength, Appendix III," WADC T-R 53-50 Pt 1.
- [12] "The Analysis of Brittle Components," Appendix I to Chapter 5, Report of the Materials Advisory Board Ad Hoc Committee on Ceramic Processing, Washington, D.C.
- [13] Gregory and Spruill, "Structural Reliability of Re-Entry Vehicles Using Brittle Materials in the Primary Structure," IAS Aerospace Systems Reliability Symposium, 1962.
- [14] S. B. Batdorf, "A Statistical Theory for Failure of Brittle Materials

Under Combined Stresses", AIAA Paper, #73-381.

- [15] S. B. Batdorf, "Weibull Statistics for Polyaxial Stress State," Journal of American Soc., January, 1974.
- [16] A. G. Evans, "A General Approach for the Statistical Analysis of Multiaxial Fracture," August, 1978.
- [17] N. A. Weil, I. M. Daniel, "Analysis of Fracture Probabilities in Nonuniformly Stressed Brittle Materials," Journal of American Soc., Vol. 47, #6, June, 1964.
- [18] Daniel, I. M. and Weil, N. A., "The Influence of Stress Gradient Upon Fracture of Brittle Materials," Journal of American Society of Mechanical Engineers, Paper No. 63-WA-228.
- [19] Rubinstein, Zolman, A Course in Ordinary and Partial Differential Differential Equations, Academic Press, 1969.
- [20] Ralston, Anthony, A First Course in Numerical Analysis, McGraw-Hill Book Co., Inc., 1965.
- [21] Brisbane, John J., "Heat Conduction and Stress Analysis of Solid Propellant Rocket Motor Nozzles," Technical Report S-198, Rohm and Haas Co., Redstone Research Laboratory, Huntsville, Ala., 1969.

```

00100 C PROGRAM WEIBUL THREE PARAMETERS
00200 C THE PARAMETERS ARE SIGMA0, M ,AND C
00300 C THIS PROGRAM COMPUTES ALL PARAMETERS BY
00400 C LINEAR REGRESSION METHOD. FOR NOTATION SIGMA0=SIGO
00500 DIMENSION X(20),F(20),W(20),EK(20),P(20),WN(20),WM(20)
00600 WRITE(5,10)
00700 10 FORMAT(1H,8HGO AHEAD)
00800 C GO AHEAD REQUIRES YOU TO TYPE NO. OF DATA FILE
00900 READ(5,*) ND
01000 READ(ND,*) N
01100 C ND=NUMBER OF DATA FILE
01200 C N=NUMBER OF WEIBUL POINT PAIRS
01300 C W=ARBITRARY WEIGHTS
01400 C X=FRACTURE STRESS VALUES
01500 READ(ND,*) ((X(I),F(I),W(I)),I=1,N)
01600 DO 1 I=1,N
01700 1 WRITE(5,20) (I,X(I),F(I),W(I))
01800 20 FORMAT(1X,°X(I),F(I),W(I) FORI=°,I2,3(1X,E12.6))
01900 DO 2 I=1,N
02000 EK(I)=ALOG(ALOG(1./(1.-F(I))))
02100 2 CONTINUE
02200 NN=5
02300 JK=0
02400 INT=10
02500 RESO=1.E+30
02600 BEG=-.99*X(1)
02700 EMD=.99*X(1)
02800 98 H=(EMD-BEG)/FLOAT(INT)
02900 DO 95 I=0,INT
03000 SIK=BEG+FLOAT(I)*H
03100 DO 96 II=1,N
03200 96 P(II)=ALOG(X(II)-SIK)
03300 CALL LIN(W,WN,P,EK,N,SIK,A,B,RES)
03400 IF(RES.GT.RESO) GO TO 95
03500 RESO=RES
03600 SIG=SIK
03700 AG=A
03800 BG=B
03900 J=I
04000 95 CONTINUE
04100 WRITE(5,80) AG,BG,RESO,SIG,SIGO
04200 C REDIFINE EMD AND BEG AND CONTINUE
04300 IF (JK.GT.NN) GO TO 97
04400 JK=JK+1
04500 IF (J.EQ.0) J=1
04600 IF (J.EQ.INT) J=INT-1
04700 EMD=BEG+FLOAT(J+1)*H
04800 BEG=EMD-2.*H
04900 RESO=RES
05000 SIGO=SIG
05100 GO TO 98
05200 80 FORMAT(2X,°AG=°,E12.6,7X,°BG=°,E12.6/,2X,°RESO=°,E12.6,

```

```

05300      1  5X,°SIG=°,E12.6,5X,°SIGO=°,E12.6)
05400      97  EC=EXP(AG)
05500          ESIGO=SIG
05600          EM=BG
05700          WRITE(5,100)EC,EM,SIG
05800      100  FORMAT(/,2X,°EC=°,E12.6,5X,°EM=°,E12.6,5X,°SIG=°,E12.6/)
05900      C CALCULATED PROBABILITIES ARE LISTED AS BELOW
06000          DO 130 II=1,N
06100      130  P(II)=ALOG(X(II)-SIG)
06200          CALL LIN(W,WM,P,EK,N,SIG,A,B,RES)
06300          DO 108 I=1,N
06400          PCAL=1.0-EXP(-EC*(X(I)-ESIGO)**EM)
06500      108  WRITE(5,110) PCAL,F(I),W(I),WM(I)
06600      110  FORMAT(1X,°PCAL,F(I),W(I),WM(I)=°,E12.6,3(1X,E12.6))
06700          STOP
06800          END
06900          SUBROUTINE LIN(W,WN,P,EK,N,SIG,A,B,RES)
07000          DIMENSION W(20),P(20),EK(20),WN(20)
07100          C=0.
07200          D=0.
07300          E=0.
07400          G=0.
07500          H=0.
07600          DO 4 I=1,N
07700          C=C+(W(I))
07800          D=D+(W(I)*P(I))
07900          E=E+(W(I)*P(I)*P(I))
08000          G=G+(W(I)*EK(I))
08100          H=H+(W(I)*EK(I)*P(I))
08200      4    CONTINUE
08300          DEN=E*C-D*D
08400          A=((E*G-D*H)/(DEN))
08500          B=((C*H-D*G)/(DEN))
08600      C THE CONSTANTS A AND B ARE KNOWN
08700      C NOW FIND RESIDUE
08800          SUMM=0.
08900          DO 5 I=1,N
09000          WN(I)=(A+B*P(I))-EK(I)
09100      5    SUMM=SUMM+W(I)*WN(I)*WN(I)
09200          RES=SUMM
09300          RETURN
09400          END

```

```

00200 C THIS IS A BIAXIAL PROGRAM
00300     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
00400     DIMENSION TH(20),EM(20),W(20),A(20,20),B(20),XX(20)
00500     1 ,XK(20)
00600 C FOURIER LEAST SQUARE APPROXIMATION PROGRAM
00700 C ND=THE FILE NAME AS IN ND=5(THIS TERMINAL) OR
00800 C FORND.DAT=THE DISK FILE
00900     WRITE(5,40)
01000 40     FORMAT(2X,'ENTER DATA FILE NO.°)
01100     READ(5,*) ND
01200     WRITE(5,70)
01300 70     FORMAT(1X,'NOW ENTER NO. OF POINTS °)
01400     READ(5,*) N
01500 C N=NUMBER OF WEIBULL POINT PAIRS OR STRESS/FRACTURE/
01600 C PERCENT PAIRS.
01700     READ(ND,*) ((TH(I),EM(I),W(I)),I=1,N)
01800     DO 50 I=1,N
01900 50     WRITE(5,60) (I,TH(I),EM(I),W(I))
02000 60     FORMAT(1X,'TH(I),EM(I),W(I) FORI=°,I2,3(1X,E13.6))
02100 C W=THE WEIGHT AT EACH THETA
02200 C THETA IS IN RADIANS
02300 C EM=THE WEIBUL MODULUS
02400 C TO MAKE THE MATRIX SYMMETRIC WE MULTIPLY TOP ROW
02500 C BY .5 AFTER FORMULATION OF MATRIX
02600 C DATAS ARE UNEQUALLY SPACED
02700 C FOURIER APPROXIMATION FITTING THE DATA POINTS.
02800 C THEN THE NORMAL EQUATIONS ARE SET UP.
02900 C THEN THEY ARE SOLVED AND TESTED FOR SIGNIFICANCE
03000 C AND THE FOURIER APPROXIMATION IS INCREASED BY ONE DEGREE IS
03100 C NEEDED AND THE PROCESS REPEATED IF NEEDED.
03200 C ITER=THE NUMBER OF TIMES THE PROCESS IS EXECUTED.
03300     NP=5
03400     MP=N/2
03500     MP=2*MP
03600     ITER=0
03700     SIGO=30E+30
03800     M=1
03900 C     SET UP THE FIRST MATRIX
04000 13     I=1
04100     B(I)=PHO(I,N,W,EM,TH)
04200     DO 51 I=2,M+1
04300     K=I-1
04400 51     B(I)=RHO(K,N,W,EM,TH)
04500     DO 52 I=M+2,2*M+1
04600     K=I-M-1
04700 52     B(I)=GHO(K,N,W,EM,TH)
04800     WRITE(5,204)
04900 204    FORMAT(/,10X,'RIGHT HAND SIDE MATRIX°)
05000     WRITE(5,205) (I,B(I),I=1,2*M+1)
05100 205    FORMAT(/,10X,I2,4X,E13.6)
05200     I=1
           J=1

```

```

05300      A(I,J)=RMC(I,J,N,W,EM,TH)
05400      I=1
05500      DO 53 J=1,M
05600      KO=I
05700      LO=J+1
05800      53  A(KO,LO)=AMC(J,KO,LO,N,W,EM,TH)
05900      I=1
06000      DO 54 J=1,M
06100      K4=I
06200      L4=J+1+M
06300      54  A(K4,L4)=BMC(J,K4,L4,N,W,EM,TH)
06400      DO 55 I=1,M
06500      DO 55 J=1,M
06600      K1=I+1
06700      L1=J+1
06800      55  A(K1,L1)=DMC(I,J,K1,L1,N,W,EM,TH)
06900      DO 56 I=1,M
07000      DO 56 J=1,M
07100      K2=I+1
07200      L2=J+1+M
07300      56  A(K2,L2)=GMC(I,J,K2,L2,N,W,EM,TH)
07400      DO 57 I=1,M
07500      DO 57 J=1,M
07600      K3=I+1+M
07700      L3=J+1+M
07800      57  A(K3,L3)=HMC(I,J,K3,L3,N,W,EM,TH)
07900      DO 58 I=1,2*M
08000      DO 58 J=I+1,2*M+1
08100      58  A(J,I)=A(I,J)
08200      A(I,J)=A(J,I)
08300      WRITE(5,219)
08400      219  FORMAT(/,6X,'MATRIX OF COEFICIENTS',7X,'I',5X,'J',
08500      1  9X,'A(I,J)',/)
08600      WRITE(5,220) ((I,J,A(I,J)),J=1,2*M+1),I=1,2*M+1)
08700      220  FORMAT(6X,I2,4X,I2,5X,E13.6)
08800      C NOW SOLVE THE NORMAL MATRIX EQUATIONS
08900      C NP IS THE NUMBER OF PRINTER
09000      M1=2*M+1
09100      CALL SOLVE(A,XX,B,M1,DET,NP)
09200      WRITE(5,225) M,(XX(I),I=1,M1)
09300      225  FORMAT(2X,'M=',I5,/,1X,'XX(I)=',4(3E13.6))
09400      C NOW TEST THE STATISTICAL SIGNIFICANCE OF INCREASING M
09500      DO 59 I=1,2*M+1
09600      59  XK(I)=XX(I)
09700      IF(MP.EQ.N) GO TO 101
09800      IF(MP.LT.N) GO TO 102
09900      101  IF(2*M+1.EQ.(MP-1)) GO TO 99
10000      102  IF(2*M+1.EQ.N) GO TO 99
10100      C COMPUTE DELTA**2 THEN SIGMA**2
10200      DELN=0.
10300      DO 61 I=1,N
10400      TEMP=XX(1)/2

```

```

10500          DO 62 J=1,M
10600          J1=J+1
10700          J2=M+1+J
10800          62   TEMP=TEMP+(XX(J1)*COS(J*TH(I))+XX(J2)*SIN(J*TH(I)))
10900          61   DELN=DELN+W(I)*(EM(I)-TEMP)**2
11000          C NOW COMPUTE SIGMA**2
11100          XT=N-2*M-1
11200          SIGN=DELN/XT
11300          WRITE(5,200) DELN,SIGN,SIG0
11400          200  FORMAT(2X,'DELN=DELTA**2=°',E13.6/,
11500          1    2X,'SIGN=SIGMA**2=°',E13.6,2X,'SIG0=°',E13.6)
11600          IF(SIGN-SIG0) 11,99,99
11700          11   SIG0=SIGN
11800          M=M+1
11900          GO TO 13
12000          99   STOP
12100          END
12200          C FUNCTION PHO COMPUTES THE ELEMENTS IN R.H.S. MATRIX
12300          FUNCTION PHO(I,N,W,EM,TH)
12400          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
12500          DIMENSION W(20),EM(20),TH(20)
12600          SUMM=0.
12700          DO 1 L=1,N
12800          1    SUMM=SUMM+(W(L)*EM(L))* .5
12900          PHO=SUMM
13000          RETURN
13100          END
13200          C FUNCTION RHO COMPUTES THE ELEMENTS IN R.H.S. MATRIX
13300          FUNCTION RHO(J,N,W,EM,TH)
13400          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
13500          DIMENSION W(20),EM(20),TH(20)
13600          SUMM=0.
13700          DO 1 L=1,N
13800          1    SUMM=SUMM+(W(L)*EM(L)*COS(J*TH(L)))
13900          RHO=SUMM
14000          RETURN
14100          END
14200          C FUNCTION GHO COMPUTES THE ELEMENTS IN R.H.S. MATRIX
14300          FUNCTION GHO(J,N,W,EM,TH)
14400          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
14500          DIMENSION W(20),EM(20),TH(20)
14600          SUMM=0.
14700          DO 1 L=1,N
14800          1    SUMM=SUMM+(W(L)*EM(L)*SIN(J*TH(L)))
14900          GHO=SUMM
15000          RETURN
15100          END
15200          C ALL THE FUNCTIONS BELOW FORMS THE L.H.S. MATRIX
15300          FUNCTION RMC(I,J,N,W,EM,TH)
15400          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
15500          DIMENSION W(20),EM(20),TH(20)
15600          SUMM=0.

```

```

15700      DO 1 L=1,N
15800      1  SUMM=SUMM+(W(L))* .25
15900      RMC=SUMM
16000      RETURN
16100      END
16200      FUNCTION AMC(J,K0,L0,N,W,EM,TH)
16300      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
16400      DIMENSION W(20),EM(20),TH(20)
16500      E=0.
16600      DO 1 L=1,N
16700      1  E=E+(W(L)*COS(J*TH(L)))*.5
16800      AMC=E
16900      RETURN
17000      END
17100      FUNCTION BMC(J,K4,L4,N,W,EM,TH)
17200      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
17300      DIMENSION W(20),EM(20),TH(20)
17400      SUMM=0.
17500      DO 1 L=1,N
17600      1  SUMM=SUMM+(W(L)*SIN(J*TH(L)))*.5
17700      BMC=SUMM
17800      RETURN
17900      END
18000      FUNCTION DMC(I,J,K1,L1,N,W,EM,TH)
18100      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
18200      DIMENSION W(20),EM(20),TH(20)
18300      SUMM=0.
18400      DO 1 L=1,N
18500      1  SUMM=SUMM+(W(L)*COS(J*TH(L))*COS(I*TH(L)))
18600      DMC=SUMM
18700      RETURN
18800      END
18900      FUNCTION GMC(I,J,K2,L2,N,W,EM,TH)
19000      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
19100      DIMENSION W(20),EM(20),TH(20)
19200      SUMM=0.
19300      DO 1 L=1,N
19400      1  SUMM=SUMM+(W(L)*SIN(J*TH(L))*COS(I*TH(L)))
19500      GMC=SUMM
19600      RETURN
19700      END
19800      FUNCTION HMC(I,J,K3,L3,N,W,EM,TH)
19900      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
20000      DIMENSION W(20),EM(20),TH(20)
20100      SUMM=0.
20200      DO 1 L=1,N
20300      1  SUMM=SUMM+(W(L)*SIN(J*TH(L))*SIN(I*TH(L)))
20400      HMC=SUMM
20500      RETURN
20600      END
20700      END

```

```

.00      C   PROGRAM 00046(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10,TAPE11,
00200    C   1TAPE12,TAPE13,TAPE14,TAPE15,TAPE16,PUNCH)
00300      C   IMPLICIT REAL*8(A-H,O-Z)
00400      C   REAL*8 TITLE,WORD1,WORD2,WORD3
00500      C   REAL NU21,NU31,NU32
00600    C**AMG054 FINITE ELEMENT STRESS ANALYSIS OF ROCKET NOZZLES
00700      C   INTEGER CODE,ERR,PCODE,BW,TOPT
00800      C   DIMENSION R(30,50),Z(30,50),CODE(30,50),TITLE(13),UF(100),WF(100)
00900      C   1 ,TANF(100),IMAX(50),IMIN(50),BC(100,3),CONPR(16,15),IP(200)
01000      C   2 ,JP(200),P(200),TAU(200),PCODE(200),NEQ(50),          S(8,8)
01100      C   3 ,F(8),C(4,4),A(120,80),B(120),U(30,50),W(30,50)
01200      C   4 ,RT(2000),ZT(2000),TEMP(2000),PSI(4),SSMAX(14),SSMIN(14),IJSS(14)
01300      C   5 ,4)
01400      C   COMMON/TEM/A
01500      C   COMMON UF,WF,TANF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
01600      C   1     CONPR,IP,JP,P,TAU,PCODE,BW,NEQ,JRAN,          S,F,NPCARD
01700      C   2 ,TOPT,NP,MN
01800      C   COMMON/VARPRO/TABLE(12,10,5)
01900      C   EQUIVALENCE (BC,UF),(A(1),U),(A(1501),W),(A(3001),RT)
02000      C   1 ,(A(5001),ZT),(A(7001),TEMP)
02100      C   DATA WORD1/6HH ONLY/,WORD2/6HEND OF/,WORD3/6HAMG054/
02200      C   CALL ERRSET(209,256,-1,1,0)
02300      C   KPLOT=0
02400      C   MAX = 30
02500      C   JR1 = 120
02600      C   READ(40,101) N1,N2,N3
02700    C N1=INPUT DATA FILE NO.=40
02800    C N2=OUTPUT DATA FILE NO.=41
02900    C N3=MAT NO FOR WHICH WEIBULL STS. IS TO BE DONE!FOR WEIBUL
03000      C   4 READ(40,100)(TITLE(I),I=1,13)
03100      C   IF(TITLE(13).NE.WORD1)WRITE(46)TITLE
03200      C   1000 READ(40,101) JMIN,JMAX,          NCONT,TOPT,NTABLE
03300      C   REWIND 42
03400      C   REWIND 44
03500      C   REWIND 45
03600      C   CALL ERRSET(1)
03700      C   WRITE(41,205) (TITLE(I),I=1,13)
03800      C   WRITE(41,206) JMIN,JMAX,          NCONT,NTABLE
03900      C   ITOPT=TOPT+1
04000      C   GO TO(3000,3001,3002,3003,3004,3000),ITOPT
04100    3001 WRITE(41,218)
04200      C   GO TO 3000
04300    3002 WRITE(41,219)
04400      C   GO TO 3000
04500    3003 WRITE(41,220)
04600      C   GO TO 3000
04700    3004 WRITE(41,221)
04800    3000 CONTINUE
04900      C   JRAN=JMAX-JMIN+1
05000    C***** INITIALIZE
05100      C   DO 2 J=1,JRAN
05200      C   DO 1 I=1,MAX

```

AD-A062 315

NORTH CAROLINA AGRICULTURAL AND TECHNICAL STATE UNIV --ETC F/G 9/5  
A STUDY OF TWO-DIMENSIONAL RECURSIVE DIGITAL FILTERS AND A STUD--ETC(U)  
NOV 78 N00014-77-C-0199

UNCLASSIFIED

NL

2 OF 2  
ADA  
062315



END  
DATE  
FILMED

3 -79  
DDC

```

05300      R(I,J)=0.
05400      Z(I,J)=0.
05500      1 CODE(I,J)=0
05600      IMIN(J)= 1000
05700      2 IMAX(J)= 0
05800      DO 3 I=1,100
05900      DO 3 J=1,3
06000      3 BC(I,J)=0.
06100      ERR=0
06200      C*** READ NODAL POINT DATA
06300      CALL MESH
06400      C CALL GRIDSC(R,Z, CODE, IMIN, IMAX, J, J, K, K, PLOT, TITLE)
06500      IF(TOPT.EQ.5) TOPT=0
06600      IF(ERR.EQ.0) GO TO 6
06700      WRITE(41,212) ERR
06800      5 READ(40,100) (TITLE(I),I=1,13)
06900      IF(WORD2.NE.TITLE(1).AND.WORD3.NE.TITLE(1)) GO TO 5
07000      GO TO 2001
07100      6 IF (TITLE(13) .EQ. WORD1) GO TO 5
07200      C*** READ PRESSURE CARDS
07300      CALL PRESBC(IP,JP,P,TAU,PCODE,NPCARD,ERR)
07400      IF(ERR.NE.0)GO TO 5
07500      IF(NPCARD.EQ.0) GO TO 8
07600      WRITE(41,207)
07700      C*** ORDER THE PRESSURE CARDS BY INCREASING I AND J
07800      IF(NPCARD.EQ.1) GO TO 706
07900      N1=NPCARD-1
08000      DO 704 N=1,N1
08100      IS= IP(N)
08200      JS=JP(N)
08300      NS=N
08400      M1= N+1
08500      DO 703 M=M1,NPCARD
08600      IF(JP(M)-JS)702,701,703
08700      701 IF(IP(M).GT.IS)GO TO 703
08800      702 IS=IP(M)
08900      JS=JP(M)
09000      NS=M
09100      703 CONTINUE
09200      TS=TAU(NS)
09300      PS=P(NS)
09400      IPCS=PCODE(NS)
09500      IP(NS)=IP(N)
09600      JP(NS)=JP(N)
09700      P(NS)=P(N)
09800      TAU(NS)=TAU(N)
09900      PCODE(NS)=PCODE(N)
10000      IP(N)=IS
10100      JP(N)=JS
10200      P(N)=PS
10300      TAU(N)=TS
10400      PCODE(N)=IPCS

```

```

10500      704 WRITE(41,208)IS,JS,PS,TS,IPCS
10600      706 WRITE(41,208)IP(NPCARD),JP(NPCARD),P(NPCARD),TAU(NPCARD),PCODE(NPCA
10700      1  RD)
10800      C***      READ CONTINUUM MATERIAL PROPERTIES
10900      8  IF(NCONT.EQ.0) GO TO 13
11000      WRITE(41,203)
11100      DO 9 N=1,NCONT
11200      READ(40,103) MN,E1,E2,E3,NU21,NU31,NU32,PHI,G13,ALPHA1,
11300      1  ALPHA2,ALPHA3,CONPR(15,MN),CONPR(16,MN)
11400      WRITE(41,204) MN,E1,NU21,ALPHA1,G13,PHI,E2,NU31,ALPHA2,
11500      1  CONPR(15,MN),E3,NU32,ALPHA3,CONPR(16,MN)
11600      CALL PROP(E1,E2,E3,NU21,NU31,NU32,G13,PHI,ALPHA1,
11700      1  ALPHA2,ALPHA3,C,PSI)
11800      ICOUNT=0
11900      DO 801 II=1,4
12000      CONPR(II+10,MN)=PSI(II)
12100      DO 801 JJ=II,4
12200      ICOUNT=ICOUNT+1
12300      801 CONPR(ICOUNT,MN)=C(II,JJ)
12400      9  CONTINUE
12500      13  IF(NTABLE.EQ.0)GO TO 11
12600      C***      READ MATERIAL PROPERTY VS. TEMPERATURE TABLES
12700      WRITE(41,215)
12800      DO 10 N=1,NTABLE
12900      READ(40,104) MTN,NTEMP,BFR,BFZ,PHI,TO
13000      WRITE(41,216) MTN,BFR,BFZ,PHI
13100      MTN = MTN - 15
13200      DO 1001 N1=1,NTEMP
13300      READ(40,105)(TABLE(II,N1,MTN),II=1,11)
13400      1001 WRITE(41,217) (TABLE(II,N1,MTN),II=1,11)
13500      TABLE(12,1,MTN)=NTEMP
13600      TABLE(12,2,MTN)=BFR
13700      TABLE(12,3,MTN)=BFZ
13800      TABLE(12,4,MTN)=MTN+ 15
13900      TABLE(12,5,MTN)=TO
14000      TABLE(12,6,MTN) = PHI
14100      10  CONTINUE
14200      C***      DETERMINE TEMPERATURES FOR CONTINUUM ELEMENTS
14300      11  REWIND 42
14400      IF(TOPT.NE.0)CALL TEMPT(TOPT)
14500      REWIND 42
14600      CALL SETUP(A,B)
14700      C***      SOLVE FOR DISPLACEMENTS
14800      CALL BACSUB(A,B,NEQ,BW,JRAN)
14900      REWIND 45
15000      DO 14 J=1,JRAN
15100      J1=JRI-JRAN+J
15200      NEQJ = NEQ(J)
15300      14  WRITE(45) (A(J1,N1),N1=1,NEQJ)
15400      REWIND 45
15500      WRITE(41,213)
15600      DO 15 J=1,JRAN

```

```

15700      N2= NEQ(J)/2
15800      READ(45) (U(N1,J),W(N1,J) ,N1=1,N2)
15900      I1= IMIN(J)
16000      I2= IMAX(J)
16100      J1= JMIN-1+J
16200      DO 15 I=I1,I2
16300      I3= I+1-IMIN(J)
16400      NNN=0
16500      IF(J.EQ.JRAN.AND.I.EQ.I2)NNN=-1
16600      WRITE(46) R(I3,J),Z(I3,J),U(I3,J),W(I3,J),NNN
16700      RPDR=R(I3,J)+U(I3,J)
16800      ZPDZ=Z(I3,J)+W(I3,J)
16900      15 WRITE(41,214) I,J1, R(I3,J), Z(I3,J), U(I3,J), W(I3,J), RPDR, ZPDZ
17000      C**** CALCULATION OF ELEMENT STRESSES
17100      REWIND 44
17200      JRAN1= JMAX-JMIN
17300      IC=0
17400      DO 1501 I=1,14
17500      SSMAX(I)=-1.E20
17600      1501 SSMIN(I)=1.E20
17700      DO 16 J=1,JRAN1
17800      I1=IMIN(J)
17900      I2=IMAX(J)-1
18000      DO 16 I=I1,I2
18100      I3=I+1-I1
18200      J3=JMIN-1+J
18300      IN=I-IMIN(J+1)+1
18400      16 IF(CODE(I3,J)/1000000.LE.25)CALL STRESS(U,W,R,Z,I3,J,I,J3,CODE,IC,
18500      1 IN,SSMAX,SSMIN,IJSS)
18600      XX=1.E20
18700      NNN=-1
18800      WRITE(46)I3,JRAN,(XX,I=1,16),NNN
18900      WRITE(41,222) (IJSS(I,1),IJSS(I,2),SSMIN(I),IJSS(I,3),
19000      1 IJSS(I,4),SSMAX(I),I=1,14)
19100      222 FORMAT(1H112X59HMINIMUM AND MAXIMUM VALUES OF STRESS AND STRAIN IN
19200      1 THE BODY/1H041X7HMINIMUM23X,7HMAXIMUM/1H0,8HQANTITY25X1HI,4X1HJ
19300      210X5HVALUE14X1HI,4X1HJ10X5HVALUE/
19400      317HORADIAL STRESS      13X2I5,1PE15.4,10X2I5,E15.4/
19500      417HOHOOP STRESS      13X2I5,E15.4,10X2I5,E15.4/
19600      517HOAXIAL STRESS      13X2I5,E15.4,10X2I5,E15.4/
19700      617HOR-Z SHEAR STRESS 13X2I5,E15.4,10X2I5,E15.4/
19800      717HOMAX STRESS      13X2I5,E15.4,10X2I5,E15.4/
19900      817HOMIN STRESS      13X2I5,E15.4,10X2I5,E15.4/
20000      917HOMAX SHEAR STRESS 13X2I5,E15.4,10X2I5,E15.4/
20100      117HORADIAL STRAIN    13X2I5,E15.4,10X2I5,E15.4/
20200      117HOHOOP STRAIN     13X2I5,E15.4,10X2I5,E15.4/
20300      217HOAXIAL STRAIN    13X2I5,E15.4,10X2I5,E15.4/
20400      317HOR-Z SHEAR STRAIN 13X2I5,E15.4,10X2I5,E15.4/
20500      417HOMAX STRAIN      13X2I5,E15.4,10X2I5,E15.4/
20600      517HOMIN STRAIN      13X2I5,E15.4,10X2I5,E15.4/
20700      617HOMAX SHEAR STRAIN 13X2I5,E15.4,10X2I5,E15.4/
20800      READ(40,100) (TITLE(I),I=1,13)

```

```

20900      2001 CONTINUE
21000          IF(WORD1.NE.TITLE(13))WRITE(46)TITLE
21100          IF(WORD2.NE.TITLE(1)) GO TO 1000
21200          CALL CLEAN
21300          CALL EXIT
21400      100 FORMAT(13A6)
21500      101 FORMAT(8I5)
21600      102 FORMAT(2I5,2F10.5,I5)
21700      103 FORMAT(I10,6F10.5/7F10.5)
21800      104 FORMAT(2I10,4F10.5)
21900      105 FORMAT(7F10.5/10X4F10.5)
22000      201 FORMAT(2I10,2F20.4,I10)
22100      203 FORMAT(1H125X38H MATERIAL PROPERTIES )
22200      204 FORMAT(10HOMAT. NO.=I2,4X3HE1=1PE11.4,4X5HNU21=E11.4,2X7HALPHA1=
22300          1 E11.4,5X4HG13=E11.4,5X4HPHI=E11.4/
22400          2 16X3HE2=E11.4,4X5HNU31=E11.4,2X7HALPHA2=
22500          2 E11.4,5X4HBFZ=E11.4/16X3HE3=E11.4,4X5HNU32=E11.4,2X7HALPHA3=
22600          3 E11.4,5X4HBFZ= E11.4)
22700      205 FORMAT(49H1AMG054 FINITE ELEMENT STRESS ANALYSIS OF NOZZLES/
22800          184HOWRITTEN BY JOHN BRISBANE, ROHM AND HAAS CO., REDSTONE ARSENAL
22900          2RESEARCH LABORATORIES/1H013A6)
23000      206 FORMAT( 6H0JMIN=I5/6H JMAX=I5/
23100          1 18H NO. OF MATERIALS=I5/
23200          225H NO. OF MATERIAL TABLES = I3)
23300      207 FORMAT(1H116X34H APPLIED PRESSURES /
23400          19X1H19X1HJ13X1HP17X3HTAU11X5HPCODE )
23500      208 FORMAT(2I10,2F20.5,I10)
23600      212 FORMAT(I5,52H DATA ERRORS NOTED, PROGRAM PROCEEDS TO NEXT PROBLEM)
23700      213 FORMAT(1H13X1HI4X1HJ6X12HR-COORDINATE6X12HZ-COORDINATE6X14HR-DISPL
23800          1ACEMENT6X14HZ-DISPLACEMENT6X11HR + DELTA-R6X11HZ + DELTA-Z /)
23900      214 FORMAT( 2I5, 2F15.4, 2( 10X, 1PE15.4), 0P2F15.4)
24000      215 FORMAT( 1H1,35X,49H MATERIAL PROPERTY TABLE
24100          1 S)
24200      216 FORMAT(1H0/13H MATERIAL NO.I3,10X4HBFZ=E13.6,10X4HBFZ=E13.6,
24300          1 10X4HPHI=E13.6/1H0,7X,1HT,7X,2HE1,10X,2HE2,10X,2HE3,8X,4HNU21,
24400          2 4X,4HNU31,4X,4HNU32,6X,3HG13,8X,6HALPHA1,6X,6HALPHA2,6X,6HALPHA3)
24500      217 FORMAT(1H F9.0,3E12.4,3F8.5,4E12.4)
24600      218 FORMAT(35H0THE BODY HAS A UNIFORM TEMPERATURE)
24700      219 FORMAT(40H0THE TEMPERATURE TABLE WAS INPUT ON TAPE)
24800      220 FORMAT(41H0THE TEMPERATURE TABLE WAS INPUT ON CARDS)
24900      221 FORMAT(68HONODAL POINT TEMPERATURES WERE INPUT ON TAPE AS DETERMIN
25000          IED BY AMG065)
25100          STOP
25200          END
25300          SUBROUTINE PROP(E1,E2,E3,NU21,NU31,NU32,G13,PHI,ALPHA1,
25400          1 ALPHA2,ALPHA3,C,PSI)
25500          IMPLICIT REAL*8(A-H,O-Z)
25600          REAL*8 TITLE,WORD1,WORD2,WORD3
25700          REAL NU21,NU31,NU32
25800          DIMENSION C(4,4),PSI(4),CD(4,4),D(4,4)
25900          DATA D(2,2),D(1,2),D(3,2),D(4,2),D(2,1),D(2,3),D(2,4)/1.,6*0./
26000      C*** FORM THE C MATRIX

```

```

26100      SIN(X)=DSIN(X)
26200      COS(X)=DCOS(X)
26300      CF11 = 1./E2/E3-NU32*NU32/E2/E2
26400      CF12 = -NU21/E1/E3-NU31*NU32/E1/E2
26500      CF13 = NU21*NU32/E1/E2+NU31/E1/E2
26600      CF22 = 1./E1/E3-NU31*NU31/E1/E1
26700      CF23 = -NU32/E1/E2-NU21*NU31/E1/E1
26800      CF33 = 1./E1/E2-NU21*NU21/E1/E1
26900      DET = 1./E1*CF11+NU21/E1*CF12-NU31/E1*CF13
27000      C(1,1)=CF11/DET
27100      C(1,2)=-CF12/DET
27200      C(1,3)=CF13/DET
27300      C(2,2)=CF22/DET
27400      C(2,3)=-CF23/DET
27500      C(3,3)=CF33/DET
27600      C(4,4)=G13
27700      C(1,4)=0.
27800      C(2,4)=0.
27900      C(3,4)=0.
28000      C(2,1)=C(1,2)
28100      C(3,1)=C(1,3)
28200      C(4,1)=0.
28300      C(3,2)=C(2,3)
28400      C(4,2)=0.
28500      C(4,3)=0.
28600      C*** FORM THE VECTOR PSI
28700      DO 1 N=1,3
28800      1 PSI(N)=C(N,1)*ALPHA1+C(N,2)*ALPHA2+C(N,3)*ALPHA3
28900      PSI(4)=0.
29000      IF(PHI.EQ.0.) GO TO 8
29100      C*** ROTATE C AND PSI BY THE ANGLE PHI
29200      PHI=PHI/57.29578
29300      CP=COS(PHI)
29400      SP=SIN(PHI)
29500      D(1,1)=CP*CP
29600      D(1,3)=SP*SP
29700      D(1,4)=SP*CP
29800      D(3,1)=D(1,3)
29900      D(3,3)=D(1,1)
30000      D(3,4)=-D(1,4)
30100      D(4,1)=2.*D(3,4)
30200      D(4,3)=-D(4,1)
30300      D(4,4)=D(1,1)-D(1,3)
30400      DO 2 I=1,4
30500      CD(I,1)=0.
30600      DO 2 K=1,4
30700      2 CD(I,1)=CD(I,1)+D(K,I)*PSI(K)
30800      DO 3 I=1,4
30900      3 PSI(I)=CD(I,1)
31000      DO 5 I=1,4
31100      DO 5 J=1,4
31200      CD(I,J)=0.

```

```

31300      DO 5 K=1,4
31400      5 CD(I,J)=CD(I,J)+C(I,K)*D(K,J)
31500      DO 7 I=1,4
31600      DO 7 J=I,4
31700      C(I,J)=0.
31800      DO 6 K=1,4
31900      6 C(I,J)=C(I,J)+D(K,I)*CD(K,J)
32000      7 C(J,I)=C(I,J)
32100      8 RETURN
32200      END
32300      SUBROUTINE STIFFQ(I,J,PI,PT,IPT)
32400      C*****  CALCULATION OF STIFFNESS MATRIX OF A QUADRILATERAL COMPOSED
32500      C          OF FOUR TRIANGLES.
32600      C          CENTER NODAL DISPLACEMENTS AND ELEMENT PRESSURE ARE ELIMINATED
32700      C          AND DATA FOR THEIR CALCULATION IS WRITTEN ON TAPE 12
32800      IMPLICIT REAL*8(A-H,O-Z)
32900      REAL*8 TITLE,WORD1,WORD2,WORD3
33000      INTEGER ERR,CODE,      TOPT
33100      COMMON UF,WF,TANF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
33200      1  CONPR(16,15),DUM1,S,F,NPCARD,TOPT,NP,MN
33300      DIMENSION UF(100),WF(100),TANF(100),R(30,50),Z(30,50),CODE(30,50),
33400      1  TITLE(13),IMAX(50),IMIN(50),DUM1( 726),      S(8,8),F(8)
33500      2  ,IT(6,4),AK(10,10),F2(2),RR(5),C(4,4),ET(4),ZZ(5),F1(10)
33600      DOUBLE PRECISION AK,F1,F2,DET
33700      DATA IT(1,1),IT(2,1),IT(3,1),IT(4,1),IT(5,1),IT(6,1),IT(1,2),
33800      1IT(2,2),IT(3,2),IT(4,2),IT(5,2),IT(6,2),IT(1,3),IT(2,3),IT(3,3),
33900      2IT(4,3),IT(5,3),IT(6,3),IT(1,4),IT(2,4),IT(3,4),IT(4,4),IT(5,4),
34000      3IT(6,4)/1,2,3,4,9,10,3,4,7,8,9,10,7,8,5,6,9,10,5,6,1,2,9,10/
34100      1000 FORMAT(7HOSTIFFQ)
34200      I1= I+IMIN(J)-1
34300      IN=I1-IMIN(J+1)+1
34400      RR(1)=R(I,J)
34500      RR(2)=R(I+1,J)
34600      RR(3)=R(IN+1,J+1)
34700      RR(4)=R(IN,J+1)
34800      ZZ(1)=Z(I,J)
34900      ZZ(2)=Z(I+1,J)
35000      ZZ(3)=Z(IN+1,J+1)
35100      ZZ(4)=Z(IN,J+1)
35200      RK= (RR(1)+RR(2)+RR(3)+RR(4))/4.0
35300      ZK= (ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.0
35400      RR(5)=RR(1)
35500      ZZ(5)=ZZ(1)
35600      DT=0.
35700      IF(TOPT.NE.0)READ(42)DT
35800      1001 FORMAT(8HOSTIFFQ1,I10)
35900      IF(MN.LT.16) GO TO 9
36000      CALL INTERP(C,ET,BFR,BFZ,DT,MN)
36100      1002 FORMAT(7HOINTERP)
36200      ET(1) = ET(1)*DT
36300      ET(2) = ET(2)*DT
36400      ET(3) = ET(3)*DT

```

```

36500      ET(4) = ET(4)*DT
36600      GO TO 10
36700      9  ICOUNT=0
36800          DO 8 II=1,4
36900          ET(II)=CONPR(II+10,MN)*DT
37000          DO 8 JJ=II,4
37100          ICOUNT=ICOUNT+1
37200          C(II,JJ)=CONPR(ICOUNT,MN)
37300      8  C(JJ,II)=C(II,JJ)
37400          BFR=CONPR(15,MN)
37500          BFZ=CONPR(16,MN)
37600      10 IPT1=IPT/10
37700          IPT2=IPT-10*IPT1
37800          DO 6 N=1,10
37900          DO 7 M=1,10
38000      7  AK(N,M)=0.
38100      6  F1(N)=0.
38200          DO 1 N=1,4
38300          P=0.
38400          TAU=0.
38500          RI=RR(N)
38600          ZI=ZZ(N)
38700          RJ=RR(N+1)
38800          ZJ=ZZ(N+1)
38900          IF(IPT1.EQ.N.OR.IPT2.EQ.N) P=PI
39000          IF(IPT1.EQ.N.OR.IPT2.EQ.N) TAU=PT
39100      1003 FORMAT(9HOGOSTIFF3)
39200      13 CALL STIFF3(RI,RJ,RK,ZI,ZJ,ZK,P,TAU,IPT,C,ET,BFR,BFZ)
39300      1004 FORMAT(10HORESTIFF3 )
39400          DO 1 M1=1,6
39500          N1=IT(M1,N)
39600          F1(N1)=F1(N1)+F (M1)
39700          DO 1 M2=1,6
39800          N2=IT(M2,N)
39900      1  AK(N1,N2)=AK(N1,N2)+S (M1,M2)
40000          DET=AK(9,9)*AK(10,10)-AK(9,10)**2
40100      1005 FORMAT(5HODET=,E20.8)
40200          AK(9,10)=AK(9,9)/DET
40300          AK(9,9)=AK(10,10)/DET
40400          AK(10,10)=AK(9,10)
40500          AK(10,9)=-AK(10,9)/DET
40600          AK(9,10)=AK(10,9)
40700          DO 3 N=9,10
40800          F2(N-8)=0.
40900          DO 2 M=1,8
41000          AK(N,M)=0.
41100          DO 2 N1=9,10
41200      2  AK(N,M)=AK(N,M)+AK(N,N1)*AK(M,N1)
41300          DO 3 M=1,2
41400      3  F2(N-8)=F2(N-8)+AK(N,M+8)*F1(M+8)
41500          DO 5 N=1,8
41600          DO 401 M=1,8

```

```

41700      DET=AK(N,M)
41800      DO 4 N1=9,10
41900      4 DET=DET-AK(N,N1)*AK(N1,M)
42000      401 S(N,M)=DET
42100      DO 402 M=1,2
42200      402 F1(N)=F1(N)-AK(N,M+8)*F2(M)
42300      5 F(N)=F1(N)
42400      WRITE(44) C,ET,RK,ZK,F2,((AK(I1,I2),I1=9,10),I2=1,8)
42500      PI=0.
42600      PT=0.
42700      IPT=0
42800      1006 FORMAT(9HORESTIFFQ)
42900      RETURN
43000      END
43100      SUBROUTINE INTERP(C,ET,BFR,BFZ,TEM,MN)
43200      C*** INTERPERTING IN MATERIAL PROPERTY TABLE
43300      IMPLICIT REAL*8(A-H,O-Z)
43400      REAL*8 TITLE,WORD1,WORD2,WORD3
43500      DIMENSION C(4,4),ET(4),TABLE(12,10),X(10)
43600      COMMON/VARPRO/XTABLE(12,10,5)
43700      EQUIVALENCE (X,E1),(X(2),E2),(X(3),E3),(X(4),XNU21),(X(5),XNU31),(
43800      1 X(6),XNU32),(X(7),G13),(X(8),ALPHA1),(X(9),ALPHA2),(X(10),ALPHA3
43900      2 )
44000      DO 8 II=1,12
44100      DO 8 JJ=1,10
44200      8* TABLE(II,JJ)=XTABLE(II,JJ,MN-15)
44300      1 NTEMP=TABLE(12,1)+0.5
44400      TEMP=TEM +TABLE(12,5)
44500      IF(TEMP.LE.TABLE(1,1))GO TO 5
44600      IF(TEMP.GE.TABLE(1,NTEMP))GO TO 6
44700      DO 2 N=2,NTEMP
44800      N1=N
44900      IF(TEMP.LE.TABLE(1,N).AND.TEMP.GE.TABLE(1,N-1))GO TO 3
45000      2 CONTINUE
45100      3 RATIO=(TEMP-TABLE(1,N1-1))/(TABLE(1,N1)-TABLE(1,N1-1))
45200      7 CONTINUE
45300      DO 4 II=2,11
45400      4 X(II-1)=TABLE(II,N1-1)+RATIO*(TABLE(II,N1)-TABLE(II,N1-1))
45500      PHI=TABLE(12,6)
45600      BFR=TABLE(12,2)
45700      BFZ=TABLE(12,3)
45800      CALL PROP(E1,E2,E3,XNU21,XNU31,XNU32,G13,PHI,ALPHA1,ALPHA2,ALPHA3,
45900      1 C,ET)
46000      RETURN
46100      5 N1=2
46200      RATIO=0.
46300      GO TO 7
46400      6 N1=NTEMP
46500      RATIO=1.0
46600      GO TO 7
46700      END
46800      SUBROUTINE MESH

```

```

46900      IMPLICIT REAL*8(A-H,O-Z)
47000      REAL*4 R1,Z1,W1,W2,W3
47100      REAL*4 POLAR, SAME
47200      REAL*8 TITLE,WORD1,WORD2,WORD3
47300      REAL LINE
47400      INTEGER CT, CODE, TYPE, ERR, PCODE, BW
47500      DIMENSION R1(30),Z1(30)
47600      DIMENSION R(30,50),Z(30,50),CODE(30,50),TITLE(13),UF(100),WF(100)
47700      1 ,TANF(100),IMAX(50),IMIN(50),BC(100,3),CONPR(16,15),IP(200)
47800      2 ,JP(200),P(200),TAU(200),PCODE(200),NEQ(50),          S(8,8)
47900      3 ,F(8),RR(5),ZZ(5)
48000      COMMON UF,WF,TANF,JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
48100      1      CONPR,IP,JP,P,TAU,PCODE,BW,NEQ,JRAN,          S,F,NPCARD
48200      2      ,ITOPT,NP,MN
48300      EQUIVALENCE (BC(1),UF)
48400      DATA POLAR,LINE,SAME/1HP,1HL,1HS/
48500      DATA WORD1/6HH ONLY/
48600      C***** READ NODAL POINT DATA
48700      ABS(X)=DABS(X)
48800      SIN(X)=DSIN(X)
48900      COS(X)=DCOS(X)
49000      WRITE(41,204)
49100      ICOUNT=0
49200      IC=1
49300      IF(ITOPT.NE.4.AND. ITOPT .NE. 5) GO TO 1
49400      JRAN=JMAX+1-JMIN
49500      READ(42) (IMAX(J),IMIN(J),NEQ(J),J=1,JRAN),BW
49600      BW=2*BW
49700      DO 1000 J=1,JRAN
49800      IRAN=NEQ(J)
49900      NEQ(J) = 2*NEQ(J)
50000      READ(42)(R1(I),Z1(I),CODE(I,J),I=1,IRAN)
50100      DO 1100 I=1,IRAN
50200      R(I,J)=R1(I)
50300      1100 Z(I,J)=Z1(I)
50400      1000 CONTINUE
50500      1001 READ(40,104) W2,I,J,TYPE,I1,I2,I3,I4,BC1,BC2,BC3,BC4
50600      IF(I.EQ.0) GO TO 9
50700      IF(J.LE.JMAX ) GO TO 1002
50800      ERR=ERR+1
50900      WRITE(41,200) I,J
51000      1002 CT=I4+10*I3+100*I2+1000*I1
51100      IF(I1+I2+I3+I4.NE.0) WRITE(41,203) I,J,I1,BC1,I2,BC2,I4,BC4
51200      IF(ABS(BC1)+ABS(BC2)+ABS(BC3)+ABS(BC4).EQ.0.) GO TO 1003
51300      UF(IC)=BC1
51400      WF(IC)=BC2
51500      TANF(IC)=BC4
51600      CT=CT+IC*10000
51700      IC=IC+1
51800      IF(IC.LE.100) GO TO 1003
51900      ERR=ERR+1
52000      WRITE(41,201)

```

```

52100      GO TO 19
52200      1003 CT=CT+1000000*TYPE
52300          J1=J-JMIN+1
52400          I8=I-IMIN(J1)+1
52500          CODE(I8,J1)=(CODE(I8,J1)/100000000)*100000000+CT
52600          IF(W2.NE.LINE) GO TO 1001
52700          READ(40,104) W2,II,JJ
52800          NSTEPS=MAXO(IABS(II-I),IABS(JJ-J))
52900          ISTEP=(II-I)/NSTEPS
53000          JSTEP=(JJ-J)/NSTEPS
53100          DO 1004 N=1,NSTEPS
53200              I=I+ISTEP
53300              J=J+JSTEP
53400              IF(I1+I2+I3+I4.NE.0) WRITE(41,203) I,J,I1,BC1,I2,BC2,I4,BC4
53500              J1=J+1-JMIN
53600              I8=I-IMIN(J1)+1
53700      1004 CODE(I8,J1)=(CODE(I8,J1)/100000000)*100000000+CT
53800          GO TO 1001
53900      104 FORMAT(1X,A1,I3,2I5,4I1,2I1,4F10.0)
54000          1 READ(40,100) W1,W2,I,J,TYPE,I1,I2,I3,I4,RT,ZT,BC1,BC2,BC3,BC4
54100          IF(I.EQ.0) GO TO 5
54200          KK=0
54300      101 IF(J.LE.JMAX) GO TO 2
54400          ERR= ERR+1
54500          WRITE(41,200)I,J
54600          2 CT=I4+10*I3+100*I2+1000*I1
54700          IF(I1+I2+I3+I4.NE.0)WRITE(41,203)I,J,I1,BC1,I2,BC2,          I4,BC4
54800          IF(ABS(BC1)+ABS(BC2)+ABS(BC3)+ABS(BC4).EQ.0.) GO TO 3
54900          UF(IC) = BC1
55000          WF(IC) = BC2
55100          TANF(IC)=BC4
55200          CT=CT+IC*10000
55300          IC = IC+1
55400          IF(IC.LE.100) GO TO 3
55500          ERR = ERR+1
55600          WRITE(41,201)
55700          GO TO 19
55800          3 CT = CT+1000000*TYPE
55900          IF(ABS(RT)+ABS(ZT).NE.0.) CT=CT+100000000
56000          J1=J-JMIN+1
56100          IMIN(J1) = MINO(IMIN(J1),I)
56200          IMAX(J1) = MAXO(IMAX(J1),I)
56300          IF(W1.NE.POLAR) GO TO 4
56400          RAD = RT
56500          ANGLE=ZT/57.2957795
56600          RT= RAD*COS(ANGLE)
56700          ZT= RAD*SIN(ANGLE)
56800          4 WRITE(45) I,J1,RT,ZT,CT
56900          ICOUNT=ICOUNT+1
57000          IF(KK.EQ.1) GO TO 404
57100          IF(W2.NE.LINE) GO TO 401
57200      C**** CALCULATION OF POINTS ON A STRAIGHT LINE

```

```

57300      READ(40,103) W1,W3,II,JJ,RT1,ZT1
57400      IF(W1.NE.POLAR) GO TO 402
57500      RAD = RT1
57600      ANGLE = ZT1/57.2957795
57700      RT1= RAD*COS(ANGLE)
57800      ZT1=RAD*SIN(ANGLE)
57900      W1 = 0.
58000      402 NSTEPS=MAX0(IABS(II-I),IABS(JJ-J))
58100      ISTEP=(II-I)/NSTEPS
58200      JSTEP=(JJ-J)/NSTEPS
58300      DD=NSTEPS
58400      DR = (RT1-RT)/DJ
58500      DZ= (ZT1-ZT)/DD
58600      DO 404 N=1,NSTEPS
58700      IJK = N
58800      I=I+ISTEP
58900      J=J+JSTEP
59000      J1=J-JMIN+1
59100      IMAX(J1)=MAX0(IMAX(J1),I)
59200      IMIN(J1)=MIN0(IMIN(J1),I)
59300      RT=RT+DR
59400      ZT=ZT+DZ
59500      IF(W3.EQ.SAME) GO TO 403
59600      READ(40,102) I,J,TYPE,I1,I2,I3,I4,BC1,BC2,BC3,BC4
59700      KK=1
59800      GO TO 101
59900      403 CT=CT
60000      IF(J.EQ.JMAX) CT=MOD(CT,1000000)+130000000
60100      WRITE(45) I,J1,RT,ZT,CT
60200      ICOUNT =ICOUNT+1
60300      IF(I1+I2+I3+I4.NE.0) WRITE(41,203) I,J,I1,BC1,I2,BC2,I4,BC4
60400      404 CONTINUE
60500      401 GO TO 1
60600      C*** STORE COORDINATES AND CODE IN I,J ARRAYS
60700      5 REWIND 45
60800      51 READ(45) I,J1,RT,ZT,CT
60900      ICOUNT=ICOUNT-1
61000      I1=I-IMIN(J1)+1
61100      R(I1,J1)=RT
61200      Z(I1,J1)=ZT
61300      IF (I .EQ. IMAX(J1)) CT = MOD(CT,1000000) + 130000000
61400      CODE(I1,J1)=CT
61500      IF(ICOUNT.GT.0)GO TO 51
61600      JRAN = JMAX-JMIN+1
61700      C*** DETERMINATION OF BAND WIDTH
61800      MAXBAN = 80
61900      J2= JRAN-1
62000      BW= 0
62100      DO 52 J=1,J2
62200      NEQ(J)= 2*(IMAX(J)+1-IMIN(J))
62300      NBAND= 2*(IMAX(J)+3-IMIN(J+1))
62400      IF(NBAND.LE.MAXBAN) GO TO 52

```

```

62500      WRITE(41,207) J1,MAXBAN
62600      52 BW= MAXO(BW,NBAND)
62700      NEQ(JRAN)=2*(IMAX(JRAN)+1-IMIN(JRAN))
62800      DO 6 J=1,JRAN
62900      IF(IMAX(J)-IMIN(J).LE.MAX-1) GO TO 6
63000      ERR= ERR+1
63100      J1 = J+JMIN-1
63200      WRITE(41,202) J1
63300      6 CONTINUE
63400      IF(ERR.NE.0) GO TO 19
63500      C**** CALCULATE COORDINATES OF INTERIOR GRID POINTS
63600      IF(JRAN.LE.2) GO TO 9
63700      J2= JRAN-1
63800      DO 8 N=1,500
63900      IJK = N
64000      RESID= 0.
64100      DO 7 J=2,J2
64200      IRAN = IMAX(J)-IMIN(J)
64300      IN = IMIN(J)-IMIN(J+1)+1
64400      IM = IMIN(J)-IMIN(J-1)+1
64500      DO 7 I=2,IRAN
64600      IN = IN+1
64700      IM = IM+1
64800      IF(CODE(I,J).GE.100000000) GO TO 7
64900      DR = (R(IN,J+1)+R(IM,J-1)+R(I+1,J)+R(I-1,J))/4.-R(I,J)
65000      DZ= (Z(IM,J-1)+Z(IN,J+1)+Z(I+1,J)+Z(I-1,J))/4.-Z(I,J)
65100      R(I,J) = R(I,J)+1.8*DR
65200      Z(I,J) = Z(I,J)+1.8*DZ
65300      RESID = RESID + ABS(DR)+ABS(DZ)
65400      7 CONTINUE
65500      IF(N.EQ.1) RES1= RESID
65600      IF(RESID/RES1.LT.1.E-5) GO TO 9
65700      8 CONTINUE
65800      C**** OUTPUT OF NODAL POINT COORDINATES
65900      WRITE(43,799)
66000      799 FORMAT(6X,'I°',3X,'J°',12X,'MAT NO.°',12X,'AREA°',/)
66100      9 WRITE(41,205)N
66200      DO 16 J=1,JRAN
66300      I1=IMIN(J)-1
66400      J1= JMIN-1+J
66500      IRAN = IMAX(J)-I1
66600      DO 16 I=1,IRAN
66700      I1= I1+1
66800      C*** FIX RADIAL DISPLACEMENT IF R EQUALS ZERO
66900      IF(R(I,J).EQ.0.)CODE(I,J)=(CODE(I,J)/10000)*10000+1000+MOD(CODE(
67000      1 I,J),1000)
67100      C*** SET TYPE EQUAL 30 IF I EQUAL IRAN OR J EQUALS JMAX
67200      IF(I.EQ.IRAN.OR.J.EQ.JRAN)CODE(I,J)=CODE(I,J)/100000000*100000000
67300      U +30000000 +MOD(CODE(I,J),1000000)
67400      WRITE(41,206) I1,J1,R(I,J),Z(I,J),CODE(I,J)
67500      IF(CODE(I,J).EQ.0) GO TO 225
67600      TYPE=(CODE(I,J)-100000000 )/1000000

```

```

67700          GO TO 226
67800          225 TYPE=0
67900          226 WRITE(46) I,J,TYPE
68000          16 CODE(I,J) = MOD(CODE(I,J),100000000)
68100          I=0
68200          J=0
68300          TYPE=0
68400          WRITE(46) I,J,TYPE
68500 C*** CALCULATION OF AREAS OF TRIANGLES
68600          DO 18 J=1,JRAN
68700          I1= IMIN(J)
68800          I2= IMAX(J)
68900          J1= JMIN-1+J
69000          DO 18 I= I1,I2
69100          I3= I-I1+1
69200          IF(CODE(I3,J)/1000000.GT.25)GO TO 18
69300          IN= I-IMIN(J+1)+1
69400          RR(1)= R(I3,J)
69500          RR(2)= R(I3+1,J)
69600          RR(3)= R(IN+1,J+1)
69700          RR(4)= R(IN,J+1)
69800          RR(5)= RR(1)
69900          ZZ(1)= Z(I3,J)
70000          ZZ(2)= Z(I3+1,J)
70100          ZZ(3)= Z(IN+1,J+1)
70200          ZZ(4)= Z(IN,J+1)
70300          ZZ(5)= ZZ(1)
70400          RK= (RR(1)+RR(2)+RR(3)+RR(4))/4.
70500          ZK= (ZZ(1)+ZZ(2)+ZZ(3)+ZZ(4))/4.
70600          AREAT=0.
70700          ARM=0.
70800          DO 17 N=1,4
70900          AREA = (RR(N+1)-RR(N))*(ZK-ZZ(N))-(RK-RR(N))*(ZZ(N+1)-ZZ(N))
71000          RA=(RR(N)+RR(N+1)+RK)/3.
71100          ARM=RA*AREA+ARM
71200          IF(AREA.GT.0.) GO TO 17
71300          ERR= ERR+1
71400          WRITE(41,220) N,I,J1
71500 17      AREAT=AREAT+AREA
71600          NTT=(CODE(I+1-IMIN(J),J)/1000000)
71700          IF(NTT.NE.18) GO TO 18
71800          VOL=3.1415926536*ARM
71900          WRITE(43,999) I,J1,VOL,NTT
72000 999    FORMAT(2X,2I5,5X,°VOL=°,E10.4,I15)
72100          18 CONTINUE
72200          IF(TITLE(13).NE.WORD1)WRITE(46) IMAX,IMIN,JRAN
72300          19 RETURN
72400          100 FORMAT(2A1,I3,2I5,4I1,1X,6F10.5)
72500          102 FORMAT(3I5,4I1,21X,4F10.5)
72600          103 FORMAT(2A1,I3,I5,10X,2F10.5)
72700          200 FORMAT(26HOJ EXCEEDS JMAX ON CARD I=I5,4H J=I5)
72800          201 FORMAT(53HOMORE THAN 99 NODES HAVE NON ZERO BOUNDARY CONDITIONS)

```

```

72900      202 FORMAT(36HOMORE THAN 30 NODAL POINTS ON ROW J=15 )
73000      203 FORMAT(2I5,3(I8,1PE17.7))
73100      204 FORMAT(25HBOUNDARY CONDITION ARRAY/10H0 NODAL PT10X6HRADIAL19X5HA
73200          1XIAL          15X7HSLIDING /1H 3X1HI4X1HJ5X4HCODE7X5HVALUE9X
73300          24HCODE7X5HVALUE9X4HCODE7X5HVALUE)
73400      205 FORMAT(30H1COORDINATES CALCULATED AFTER I3,11H ITERATIONS /
73500          14X1HI4X1HJ10X1HR14X1HZ14X4HCODE /)
73600      206 FORMAT(2I5,2F15.4,I15)
73700      207 FORMAT(21H BAND WIDTH OF ROW J=13, 8HEXCEEDS 15)
73800      220 FORMAT(49H ZERO OR NEGATIVE AREA IN TRIANGULAR ELEMENT NO. I1,
73900          126H OF QUADRILATERAL ELEMENT I2,2H, I2 )
74000      END
74100      SUBROUTINE TRIAN(A,B,NEQJ,NEQJ1,BW)
74200      C***** THIS SUBROUTINE TRIANGLIZES A BLOCK OF BANDED EQUATIONS
74300      IMPLICIT REAL*8(A-H,O-Z)
74400      REAL*8 TITLE,WORD1,WORD2,WORD3
74500      INTEGER BW,BW1
74600      DIMENSION A(120,80),B(120)
74700      DOUBLE PRECISION RATIO
74800      N3= NEQJ+NEQJ1
74900      DO 1 N=1,NEQJ
75000          N1= N+1
75100          N2= MINO(BW+N-1,N3)
75200          IF(A(N,1).EQ.0.) GO TO 2
75300          DO 1 I=N1,N2
75400              I1= I+1-N
75500              RATIO= A(N,I1)/A(N,1)
75600              IF(RATIO.EQ.0.)GO TO 1
75700              B(I)= B(I)-RATIO*B(N)
75800              J1= BW-I1+1
75900              DO 3 J=1,J1
76000                  I2= I-N+J
76100                  3 A(I,J)= A(I,J)-A(N,I2)*RATIO
76200          1 CONTINUE
76300          RETURN
76400          2 WRITE(41,201)
76500      201 FORMAT(49H ZERO TERM ON MAJOR DIAGONAL EXECUTION TERMINATED)
76600      STOP
76700      END
76800      SUBROUTINE BACSUB(A,B,NEQ,BW,JRAN)
76900      IMPLICIT REAL*8(A-H,O-Z)
77000      REAL*8 TITLE,WORD1,WORD2,WORD3
77100      INTEGER BW
77200      DIMENSION A(120,80),B(120),NEQ(50),DISP(120)
77300      DOUBLE PRECISION DP
77400      DATA JR1/121/
77500      DO 5 J=1,JRAN
77600          J1= JRAN+1-J
77700          NEQJ= NEQ(J1)
77800          NEQJ1= NEQ(J1+1)
77900          IF(J1.EQ.JRAN) NEQJ1= 0
78000          N1= NEQJ+NEQJ1

```

```

78100      BACKSPACE 45
78200      READ(45) ((A(I1,I2),I2=1,BW),B(I1),I1=1,NEQJ)
78300      BACKSPACE 45
78400      DO 2 N=1,NEQJ
78500      N2= NEQJ+1-N
78600      DP= B(N2)
78700      DO 1 K=2,BW
78800      K1= N2-1+K
78900      IF(K1.GT.N1) GO TO 2
79000      1 DP= DP-A(N2,K)*DISP(K1)
79100      2 DISP(N2) = DP/A(N2,1)
79200      JR = JR1-J
79300      DO 3 K=1,NEQJ
79400      3 A(JR,K)= DISP(K)
79500      IF(J.EQ.JRAN)GO TO 5
79600      DO 4 K=1,NEQJ
79700      K1= NEQ(J1-1)+K
79800      4 DISP(K1) = A(JR,K)
79900      5 CONTINUE
80000      RETURN
80100      END
80200      SUBROUTINE  INVRT(A,ACT,DIM)
80300      C      INVERSION OF SYMMETRIC MATRIX
80400      IMPLICIT REAL*8(A-H,O-Z)
80500      REAL*8 TITLE,WORD1,WORD2,WORD3
80600      INTEGER ACT,DIM
80700      DIMENSION A(DIM,DIM),LOC(61)
80800      DOUBLE PRECISION DP
80900      ABS(X)=DABS(X)
81000      DO 1 N=1,ACT
81100      1 LOC(N)=N
81200      DO 6 N1=1,ACT
81300      M=0
81400      PIVOT=0.
81500      DO 2 N2=N1,ACT
81600      NN=LOC(N2)
81700      IF (ABS(A(NN,NN)).LE.ABS(PIVOT)) GO TO 2
81800      M=N2
81900      PIVOT=A(NN,NN)
82000      2 CONTINUE
82100      IF (M.EQ.0) GO TO 8
82200      N=LOC(M)
82300      LOC(M)=LOC(N1)
82400      LOC(N1)=N
82500      A(N,N)=-1.
82600      DO 3 J=1,ACT
82700      3 A(N,J)=A(N,J)/PIVOT
82800      DO 5 I1=1,ACT
82900      I=LOC(I1)
83000      DP=A(I,N)
83100      IF (N.EQ.I.OR.A(I,N).EQ.0.) GO TO 5
83200      DO 4 J1=I1,ACT

```

```

83300      J=LOC(J1)
83400      IF (N.EQ.J) GO TO 4
83500      A(I,J)=A(I,J)-      A(N,J)*DP
83600      A(J,I)=A(I,J)
83700      4 CONTINUE
83800      5 CONTINUE
83900      DO 6 I=1,ACT
84000      6 A(I,N)=A(N,I)
84100      DO 7 I=1,ACT
84200      DO 7 J=1,ACT
84300      7 A(I,J)=-A(I,J)
84400      RETURN
84500      8 WRITE(41,9)
84600      9 FORMAT (42HOMATRIX IS SINGULAR - EXECUTION TERMINATED )
84700      CALL EXIT
84800      RETURN
84900      END
85000      SUBROUTINE CLEAN
85100      RETURN
85200      END
85300      SUBROUTINE TEMPT(TOPT)
85400      IMPLICIT REAL*8(A-H,O-Z)
85500      REAL*4 RI,ZI,TIMP, TXME, TXEMP
85600      REAL*8 TITLE,WORD1,WORD2,WORD3 ,ISTUFF
85700      INTEGER TOPT, CODE,ERR
85800      LOGICAL L1,L2
85900      COMMON ISTUFF,JMIN,JMAX,ERR,MAX,R,Z, CODE, TITLE, IMAX, IMIN
86000      DIMENSION RI(2000),ZI(2000),TIMP(2000)
86100      DIMENSION TEMP(2000),RT(2000),ZT(2000),D(5),RR(5),ZZ(5),T(5)
86200      1 ,ISTUFF(300),R(30,50),Z(30,50),CODE(30,50),TITLE(13)
86300      2 ,IMAX(50),IMIN(50),DT2(30)
86400      3 ,TTEMP(30,50),IPRINT(30)
86500      COMMON/TEM/A(120,80)
86600      EQUIVALENCE (A(3001),RT),(A(5001),ZT),(A(7001),TEMP,TTEMP)
86700      DIMENSION TXEMP(30,50)
86800      SIN(X)=DSIN(X)
86900      COS(X)=DCOS(X)
87000      ABS(X)=DABS(X)
87100      ATAN(X)=DATAN(X)
87200      READ(40,100) TO
87300      WRITE(41,201)TO
87400      ITOPT=TOPT+1
87500      GO TO(20,1,3,4,41),ITOPT
87600      1 READ(40,100) T1
87700      DT= T1-TO
87800      WRITE(41,202)DT
87900      JRN1= JMAX-JMIN
88000      DO 2 J=1,JRN1
88100      I1= IMIN(J)
88200      I2=IMAX(J)-1
88300      DO 2 I=I1,I2
88400      I3= I+1-I1

```

```

88500      IF(CODE(I3,J)/1000000.GT.25) GO TO 2
88600      WRITE(42) DT
88700      2 CONTINUE
88800      RETURN
88900      3 READ(40,101) NOTEMP,Z0,D4
89000      READ(42) (RI(N),ZI(N),TIMP(N),N=1,NOTEMP)
89100      DO 999 N=1,NOTEMP
89200      RT(N)=RI(N)
89300      ZT(N)=ZI(N)
89400      999 TEMP(N)=TIMP(N)
89500      GO TO 5
89600      4 READ(40,101) NOTEMP,Z0,D4
89700      READ(40,102) (RT(N),ZT(N),TEMP(N),N=1,NOTEMP)
89800      GO TO 5
89900      41 READ(40,101)NDIST
90000      DO 42 N1=1,NDIST
90100      READ(42) TXME,TXEMP
90200      TIME=TXME
90300      DO 442 L3=1,30
90400      DO 442 L4=1,50
90500      442 TTEMP(L3,L4)= TXEMP(L3,L4)
90600      42 CONTINUE
90700      GO TO 602
90800      5 DO 6 N=1,NOTEMP
90900      6 ZT(N)= ZT(N)-Z0
91000      602 JRAN1= JMAX-JMIN
91100      D4=D4*D4
91200      IF(TOPT.NE.4)WRITE(41,203)
91300      IF(TOPT.EQ.4)WRITE(41,205)TIME
91400      DO 16 J=1,JRAN1
91500      I1= IMIN(J)
91600      I2= IMAX(J)-1
91700      J1=JMIN-1+J
91800      DO 1601 I=I1,I2
91900      I3= I+1-I1
92000      IF(CODE(I3,J)/1000000.GT.25) GO TO 1601
92100      IN= I-IMIN(J+1)+1
92200      RK= (R(I3,J)+R(I3+1,J)+R(IN,J+1)+R(IN+1,J+1))/4.
92300      ZK= (Z(I3,J)+Z(I3+1,J)+Z(IN,J+1)+Z(IN+1,J+1))/4.
92400      IF(ITOPT.NE.5)GO TO 1602
92500      RR(1)=R(I3,J)
92600      RR(2)=R(I3+1,J)
92700      RR(3)=R(IN+1,J+1)
92800      RR(4)=R(IN,J+1)
92900      ZZ(1)=Z(I3,J)
93000      ZZ(2)=Z(I3+1,J)
93100      ZZ(3)=Z(IN+1,J+1)
93200      ZZ(4)=Z(IN,J+1)
93300      T(1)=TTEMP(I3,J)
93400      T(2)=TTEMP(I3+1,J)
93500      T(3)=TTEMP(IN+1,J+1)
93600      T(4)=TTEMP(IN,J+1)

```

```

93700      GO TO 1201
93800      1602 I4=I3
93900      I5 = IN
94000      IF (I .LT. (I1 + I2)/2) GO TO 601
94100      I4 = I4 + 1
94200      I5 = I5 + 1
94300      601 THETA=1.570795
94400      IF(ABS(R(I5,J+1)-R(I4,J)).GT..001)THETA=ATAN((Z(I5,J+1)-Z(I4,J))/(
94500      1 R(I5,J+1)-R(I4,J)))
94600      D(1)=D4
94700      D(2)=D4
94800      D(3)=D4
94900      D(4)=D4
95000      DO 12 N=1,NOTEMP
95100      AA=RT(N)-RK
95200      BB=ZT(N)-ZK
95300      DD=AA*AA+BB*BB
95400      IF(DD.GE.D4)GO TO 12
95500      L1=.TRUE.
95600      L2=.TRUE.
95700      IF(AA*SIN(THETA)-BB*COS(THETA).LT.0.)L1=.FALSE.
95800      IF(AA*COS(THETA)+BB*SIN(THETA).LT.0.)L2=.FALSE.
95900      DO 11 L=1,4
96000      GO TO(7,8,9,10),L
96100      7 IF(.NOT.L1.OR..NOT.L2) GO TO 11
96200      GO TO 10
96300      8 IF(L1.OR..NOT.L2) GO TO 11
96400      GO TO 10
96500      9 IF(L1.OR.L2) GO TO 11
96600      GO TO 10
96700      10 IF(DD.GE.D(L))GO TO 12
96800      D(L)=DD
96900      T(L)=TEMP(N)
97000      RR(L)= RT(N)
97100      ZZ(L)= ZT(N)
97200      GO TO 12
97300      11 CONTINUE
97400      12 CONTINUE
97500      1201 T(5)=T(1)
97600      CC= 0.
97700      DT1= 0.
97800      RR(5)= RR(1)
97900      ZZ(5)= ZZ(1)
98000      D(5)=D(1)
98100      DO 17 N=1,4
98200      IF(ITOPT.EQ.5)GO TO 1701
98300      IF(D(N).GE.D4.OR.D(N+1).GE.D4) GO TO 17
98400      1701 AJ= RR(N+1)-RR(N)
98500      BJ= ZZ(N+1)-ZZ(N)
98600      AK= RK-RR(N)
98700      BK= ZK-ZZ(N)
98800      AREA=AJ*BK-AK*BJ

```

```

98900      WRITE(41,599) AJ,BK,AK,BJ,AREA
99000      599  FORMAT(2X,'AJ,BK,AK,BJ,AREA=',5E13.6)
99100      IF(AREA.EQ.0.) GO TO 17
99200      C= ZZ(N+1)-ZK
99300      DX= RK-RR(N+1)

00100      COMM= (RR(N)+RR(N+1)+RK)/6./AREA
00200      DT1= COMM*(BK*BJ+AJ*AK)*T(N+1)+COMM*(C*BJ-DX*AJ)*T(N)+DT1

00100      CC=CC+COMM*(BJ*BJ+AJ*AJ)
00200      17  CONTINUE

00100      DT1= DT1/CC-T0
00200      WRITE(42) DT1
00300      IPRINT(I3)=I
00400      1601 DT2(I3)=DT1

00100      IRAN=I2+1-I1

00100      16  WRITE(41,204) J1,(IPRINT(I),DT2(I),I=1,IRAN)
00200      20  RETURN

00100      100  FORMAT(F10.5)
00200      101  FORMAT(I10, 2F10.5)
00300      102  FORMAT(2F10.5,E15.6)
00400      201  FORMAT(1H1,20X,48HT E M P E R A T U R E   D I S T R I B U T I O N
00500      1 //10X23HREFERENCE TEMPERATURE IS F10.3)
00600      202  FORMAT(65H0THE TEMPERATURE DIFFERENCE IS UNIFORM THROUGHOUT THE BO
00700      1DY AND IS F10.3 )
00800      203  FORMAT(64H0THE TEMPERATURE DIFFERENCES WERE DETERMINED FROM AN INP
00900      1UT TABLE)
01000      204  FORMAT(7HOROW J=I3/(3H I=I3,2X3HDT=F8.2,

00100      1  4X2HI=I3,2X3HDT=F8.2,
00200      1  4X2HI=I3,2X3HDT=F8.2,
00300      1  4X2HI=I3,2X3HDT=F8.2,
00400      1  4X2HI=I3,2X3HDT=F8.2))
00500      205  FORMAT(72H0THE TEMPERATURE DIFFERENCES WERE DETERMINED FROM THE DI
00600      1STRIBUTION AT T=,1PE12.5,19H AS FOUND BY AMG065)
00700      END
00800      SUBROUTINE STRESS(U,W,R,Z,I,J,I3,J3,CODE,IC,IN,SSMAX,SSMIN,IJSS)
00900      C*****  CALCULATION OF STRESSES AND STRAINS IN A QUADRILATERAL

```

```

01000      C      DISPLACEMENT FIELD IS FITTED BY A FOUR TERM PARABALOID
01100      C      TO THE FOUR CORNER AND CENTER POINT DISPLACEMENTS BY
01200      C      LEAST SQUARES
01300      IMPLICIT REAL*8(A-H,O-Z)
01400      REAL*8 TITLE,WORD1,WORD2,WORD3
01500      DOUBLE PRECISION F2,S28
01600      INTEGER CODE
01700      LOGICAL EL1,EL2
01800      EQUIVALENCE (EP,EPR),(EP(2),EPT),(EP(3),EPZ),(EP(4),GAMRZ),(SIG,
01900      1SIGR),(SIG(2),SIGT),(SIG(3),SIGZ),(SIG(4),TAURZ)
02000      EQUIVALENCE (SS,SIG),(SS(5),SIGMAX),(SS(6),SIGMIN),(SS(7),TAUMAX),
02100      1 (SS(8),EP),(SS(12),EPMAX),(SS(13),EPMIN),(SS(14),GAMMAX)
02200      DIMENSION PHI(4,5),PTP(4,4),PU(4),PW(4),F2(2),S28(2,8)
02300      1 ,U1(5),W1(5),S4(4,4),ALF(4),BET(4),U(30,50),W(30,50)
02400      2 ,C(4,4),EP(4),ET(4),SIG(4),CODE(30,50),R(30,50),Z(30,50)
02500      3 ,SSMAX(14),SSMIN(14),SS(14),IJSS(14,4)
02600      DATA PHI(1,1),PHI(1,2),PHI(1,3),PHI(1,4),PHI(1,5) /5*1./,
02700      1PHI(2,1),PHI(3,1),PHI(4,1)/3*0.0 /
02800      COS(X)=DCOS(X)
02900      SIN(X)=DSIN(X)
03000      ABS(X)=DABS(X)
03100      SQRT(X)=DSQRT(X)
03200      ATAN(X)=DATAN(X)
03300      SIGN(X,Y)=DSIGN(X,Y)
03400      READ(44) C,ET,RR,ZZ,F2,S28
03500      U1(2)=U(I,J)
03600      U1(3)=U(I+1,J)
03700      U1(4)=U(IN,J+1)
03800      U1(5)=U(IN+1,J+1)
03900      W1(2)=W(I,J)
04000      W1(3)=W(I+1,J)
04100      W1(4)=W(IN,J+1)
04200      W1(5)=W(IN+1,J+1)
04300      PHI(2,2)=R(I,J)-RR
04400      PHI(2,3)=R(I+1,J)-RR
04500      PHI(2,4)=R(IN,J+1)-RR
04600      PHI(2,5)=R(IN+1,J+1)-RR
04700      PHI(3,2)=Z(I,J)-ZZ
04800      PHI(3,3)=Z(I+1,J)-ZZ
04900      PHI(3,4)=Z(IN,J+1)-ZZ
05000      PHI(3,5)=Z(IN+1,J+1)-ZZ
05100      DO 1 M=2,5
05200      1 PHI(4,M)=PHI(2,M)**2 + PHI(3,M)**2
05300      DO 2 M=1,2
05400      DO 2 MM=2,5
05500      2 F2(M)=F2(M)-S28(M,2*MM-3)*U1(MM)-S28(M,2*MM-2)*W1(MM)
05600      U1(1)=F2(1)
05700      W1(1)=F2(2)
05800      DO 4 N=1,4
05900      DO 3 M=1,4
06000      S4(N,M)=0.
06100      DO 3 MM=1,5

```

```

06200      3 S4(N,M)=S4(N,M)+PHI(N,MM)*PHI(M,MM)
06300      PU(N)=0.
06400      PW(N)=0.
06500      DO 4 M=1,5
06600      PU(N)=PU(N)+PHI(N,M)*U1(M)
06700      4 PW(N)=PW(N)+PHI(N,M)*W1(M)
06800      CALL INVRT(S4,4,4)
06900      DO 5 N=1,4
07000      ALF(N)=0.
07100      BET(N)=0.
07200      DO 5 M=1,4
07300      ALF(N)=ALF(N)+S4(N,M)*PU(M)
07400      5 BET(N)=BET(N)+S4(N,M)*PW(M)
07500      EPR=ALF(2)
07600      EPT = U1(1)/RR
07700      EPZ=BET(3)
07800      GAMRZ=ALF(3)+BET(2)
07900      ANGLE=.5*ATAN( GAMRZ/( EPR- EPZ))*57.2958
08000      IF (EPR .LT. EPZ) ANGLE = ANGLE + SIGN(90.0DO,GAMRZ)
08100      TEM=(EPR +EPZ )/2.
08200      TEM1 =SQRT((( EPR- EPZ)/2.)**2+GAMRZ**2/4.)
08300      EPMAX=TEM+TEM1
08400      EPMIN=TEM-TEM1
08500      GAMMAX=2.*TEM1
08600      DO 8 N=1,4
08700      SIG(N)=-ET(N)
08800      DO 8 M= 1,4
08900      8 SIG(N)= SIG(N)+C(N,M)*EP(M)
09000      TEM = (SIGR+SIGZ)/2.
09100      TEM1 = SQRT(((SIGR-SIGZ)/2.)**2+TAURZ**2)
09200      SIGMAX= TEM+TEM1
09300      SIGMIN= TEM-TEM1
09400      TAUMAX = TEM1
09500      IF (MOD(IC,19)) 7,6,7
09600      6 WRITE(41,100)
09700      7 WRITE(41,101)I3,J3,RR,ZZ,SIGR,SIGT,SIGZ,TAURZ,SIGMAX,SIGMIN,
09800      1TAUMAX,ANGLE,EPR,EPT,EPZ,GAMRZ,EPMAX,EPMIN,GAMMAX,CODE(I,J)
09900      IF(CODE(I,J).NE.18000000) GO TO 661
10000      WRITE(47,499) I3,J3,SIGR,SIGT,SIGMAX,SIGMIN,CODE(I,J)
10100      499 FORMAT(1H0I3,I4,1P4E13.4,I15)
10200      661 IC=IC+1
10300      DO 11 K=1,14
10400      IF(SS(K).GT.SSMIN(K))GO TO 10
10500      SSMIN(K)=SS(K)
10600      IJSS(K,1)=I3
10700      IJSS(K,2)=J3
10800      10 IF(SS(K).LT.SSMAX(K))GO TO 11
10900      SSMAX(K)=SS(K)
11000      IJSS(K,3)=I3
11100      IJSS(K,4)=J3
11200      11 CONTINUE
11300      EL1=.FALSE.

```

```

11400      EL2=.FALSE.
11500      IF (CODE(IN,J+1)/1000000 .LT. 26) EL1 = .TRUE.
11600      IF (CODE(I+1,J )/1000000 .LT. 26) EL2 = .TRUE.
11700      IF (.NOT.EL1.AND..NOT.EL2) N=1
11800      IF (.NOT.EL1.AND.EL2) N = 2
11900      IF (EL1.AND..NOT.EL2) N=3
12000      IF (EL1.AND.EL2) N = 4
12100      WRITE(46) I,J,RR,ZZ,SIGR,SIGZ,SIGT,TAURZ,SIGMAX,SIGMIN,TAUMAX
12200      1 ,EPR,EPZ,EPT,GAMRZ,EPMAX,EPMIN,GAMMAX,N
12300      100 FORMAT( 9H1 - J/
12400      1 5X 11HCOORDINATES37X33HS T R E S S E S / S T R
12500      1A I N S /8H ANGLE7X1HR8X1HZ4X9HRADIAL R2X10HHOOP THETA5X8HAXIA
12600      2L Z3X10HSHEAR R-Z6X7HMAXIMUM6X7HMINIMUM4X9HMAX SHEAR )
12700      101 FORMAT(1HOI3,I4,OPF8.3,F9.3,1P7E13.4/1H OPF7.2,17X1P7E13.4,I15)
12800      102 FORMAT(1H 1P1E11.3)
12900      RETURN
13000      END
13100      SUBROUTINE PRESBC(ICF,JCF,H,TE,SIDE,LC,ERR)
13200      C*** THIS SUBROUTINE READS THE PRESSURE BOUNDARY CONDITION DATA
13300      C DATA MAY BE READ IN FOR AN ELEMENT SIDE AT A TIME OR FOR A LINE
13400      C WHICH HAS I OR J AS A CONSTANT. THE SUBROUTINE MAY BE CHANGED
13500      C TO FIT PARTICULAR NEEDS.
13600      IMPLICIT REAL*8(A-H,O-Z)
13700      REAL*8 TITLE,WORD1,WORD2,WORD3
13800      INTEGER SIDE, SIDET, ERR
13900      C
14000      DIMENSION ICF(200), JCF(200), H(200),
14100      1 TE(200), SIDE(200)
14200      C
14300      LC=0
14400      1 READ(40,100) I1,J1,I2,J2,HT,TET,SIDET
14500      IF(I1.EQ.0)RETURN
14600      LC=LC+1
14700      IF(LC.GT.200) GO TO 5
14800      ICF(LC)=I1
14900      JCF(LC)=J1
15000      H(LC)=HT
15100      TE(LC)=TET
15200      SIDE(LC)=SIDET
15300      IF(I2.EQ.0) GO TO 1
15400      NSTEPS=MAX0(IABS(I2-I1),IABS(J2-J1))
15500      ISTEP=(I2-I1)/NSTEPS
15600      JSTEP=(J2-J1)/NSTEPS
15700      DO 3 N=1,NSTEPS
15800      LC=LC+1
15900      IF(LC.GT.200) GO TO 5
16000      ICF(LC)=ICF(LC-1)+ISTEP
16100      JCF(LC)=JCF(LC-1)+JSTEP
16200      H(LC)=HT
16300      TE(LC)=TET
16400      3 SIDE(LC)=SIDET
16500      GO TO 1

```

```

16600      5 ERR=ERR+1
16700      WRITE(41,209)
16800      RETURN
16900      100 FORMAT(4I5,2F10.5,I5)
17000      209 FORMAT(70HNUMBER OF ELEMENT SIDES WITH PRESSURE BOUNDARY CONDITIO
17100      INS EXCEEDS 200)
17200      END
17300      SUBROUTINE SETUP(A,B)
17400      C*** ASSEMBLE STIFFNESS MATRIX OF STRUCTURE IN THE FORM OF A BAND
17500      C EQUATIONS ARE MODIFIED FOR BOUNDARY CONDITIONS AND
17600      C TRIANGLIZED BEFORE BEING WRITTEN ON TAPE
17700      IMPLICIT REAL*8(A-H,O-Z)
17800      REAL*8 TITLE,WORD1,WORD2,WORD3
17900      INTEGER BW,BCCODE,SLCODE,UCODE,WCODE,PCODE,BW1,ERR,PCODE
18000      COMMON BC, JMIN,JMAX,ERR,MAX,R,Z,CODE,TITLE,IMAX,IMIN,
18100      1 CONPR,IP,JP,P,TAU,PCODE,BW,NEQ,JRAN, S,F,NPCARD
18200      2 ,ITOPT,NP,MN
18300      EQUIVALENCE(IFIX(1),UCODE),(IFIX(2),WCODE),(BC(1),UF),(BC(101),WF)
18400      1 ,(BC(201),TANF)
18500      DIMENSION R(30,50),Z(30,50),CODE(30,50),TITLE(13),UF(100),WF(100)
18600      1 ,TANF(100),IMAX(50),IMIN(50),BC(100,3),CONPR(16,15),IP(200)
18700      2 ,JP(200),P(200),TAU(200),PCODE(200),NEQ(50), S(8,8)
18800      3 ,A(120,80),B(120),IFIX(2),F(8)
18900      NP=1
19000      REWIND 45
19100      IPT=0
19200      DO 1 I=1,120
19300      B(I)= 0.
19400      DO 1 J=1,80
19500      1 A(I,J)= 0.
19600      DO 20 J=1,JRAN
19700      IRAN= NEQ(J)/2
19800      IF(J.EQ.JRAN) GO TO 6
19900      DO 501 I=1,IRAN
20000      MN1= CODE(I,J)/1000000
20100      IF(MN1.GT.0.AND.MN1.LT.26)MN=MN1
20200      IF(MN1.GT.25) GO TO 501
20300      C***** CHECK FOR PRESSURE ON THE ELEMENT
20400      I1= IMI.(J)+I-1
20500      J1= JMIN+J-1
20600      IF(NPCARD.LT.NP)GO TO 1002
20700      IS=IP(NP)
20800      JS=JP(NP)
20900      IF(IS.NE.I1.OR.JS.N_.J1)GO TO 1002
21000      PI=P(NP)
21100      PT=TAU(NP)
21200      IPT=PCODE(NP)
21300      NP=NP+1
21400      1002 CALL STIFFQ(I,J,PI,PT,IPT)
21500      C*** ADD ELEMENT STIFFNESS TO TOTAL STIFFNESS
21600      N1= 2*(I-1)
21700      N2= 2*(I+IMIN(J)-1-IMIN(J+1))+NEQ(J)

```

```

21800      DO 5 N=1,4
21900          N3 = N2-N1+1-N
22000      NN1= N1+N
22100      NN2= N2+N
22200      B(NN1)= B(NN1)+F(N)
22300      B(NN2)= B(NN2)+F(N+4)
22400      LL= 0
22500      DO 4 L= N,4
22600          LL= LL+1
22700      A(NN1,LL)= A(NN1,LL)+S(N,L)
22800      4 A(NN2,LL)= A(NN2,LL)+S(N+4,L+4)
22900      DO 5 L=1,4
23000          L1= N3+L
23100      5 A(NN1,L1)= A(NN1,L1)+S(N,L+4)
23200      501 CONTINUE
23300      C***      MODIFY THE BLOCK OF EQUATIONS FOR FIXITY AND APPLIED LOADS
23400      6 DO 11 I=1,IRAN
23500          BCCODE= MOD(CODE(I,J),1000000)
23600          IF(BCCODE.EQ.0) GO TO 11
23700          SLCODE= MOD(BCCODE,10)
23800          WCODE= MOD(BCCODE,1000)/100
23900          UCODE= MOD(BCCODE,10000)/1000
24000          NCODE= MOD(BCCODE,1000000)/10000
24100          IF(NCODE.EQ.0) NCODE=100
24200          IU= 2*(I-1)+1
24300          IW= IU+1
24400          IF(UCODE.EQ.2) B(IU)= B(IU)+UF(NCODE)/6.2831853
24500          IF(WCODE.EQ.2) B(IW)= B(IW)+WF(NCODE)/6.2831853
24600          IF(SLCODE.EQ.0) GO TO 8
24700          ALF= TANF(NCODE)
24800          B(IU)= B(IU)+ALF*B(IW)
24900          B(IW)= 0.
25000          A(IU,1)= A(IU,1)+ALF*(ALF*(A(IW,1)+1.))+2.*A(IU,2))
25100          A(IU,2)= -ALF
25200          A(IW,1)= 1.
25300          BW1= BW-1
25400          DO 7 N=2,BW1
25500              A(IU,N+1)= A(IU,N+1)+ALF*A(IW,N)
25600              A(IW,N)= 0.
25700              II= IU+1-N
25800              IF(II.LT.1) GO TO 7
25900              A(II,N)= A(II,N)+ALF*A(II,N+1)
26000              A(II,N+1)= 0.
26100      7 CONTINUE
26200          A(IW,BW)= 0.
26300      8 NMAX= NEQ(J)+NEQ(J+1)
26400          IF (J .EQ. JRAN) NMAX = NEQ(J)
26500          DO 10 N=1,2
26600              IR=IU+N-1
26700              IF(IFIX(N).NE.1) GO TO 10
26800              DO 9 N1=2,BW
26900                  II= IR+1-N1

```

```

27000      IJ= IR+N1-1
27100      IF(II.GT.0)B(II)= B(II)-A(II,N1)*BC(NCODE,N)
27200      IF(IJ.LE.NMAX)B(IJ)= B(IJ)-A(IR,N1)*BC(NCODE,N)
27300      IF(II.GT.0) A(II,N1)= 0.
27400      9 A(IR,N1)= 0.
27500      A(IR,1)= 1.
27600      B(IR)= BC(NCODE,N)
27700      10 CONTINUE
27800      11 CONTINUE
27900      IF(J.EQ.JRAN) GO TO 17
28000      IRAN1= IMAX(J+1)+1-IMIN(J+1)
28100      DO 16 I=1,IRAN1
28200      BCCODE= MOD(CODE(I,J+1),1000000)
28300      IF(BCCODE.EQ.0) GO TO 16
28400      SLCODE= MOD(BCCODE,10)
28500      WCODE= MOD(BCCODE,1000)/100
28600      UCODE= MOD(BCCODE,10000)/1000
28700      NCODE= MOD(BCCODE,1000000)/10000
28800      IF(NCODE.EQ.0) NCODE= 100
28900      IU= NEQ(J)+2*(I-1)+1
29000      IW= IU+1
29100      IF(SLCODE.EQ.0) GO TO 13
29200      ALF= TANF(NCODE)
29300      BW1=BW-1
29400      DO 12 N=2,BW1
29500      II= IU+1-N
29600      IF(II.GT.NEQ(J).OR.II.LT.1) GO TO 12
29700      A(II,N)= A(II,N)+ALF*A(II,N+1)
29800      A(II,N+1)= 0.
29900      12 CONTINUE
30000      13 DO 15 N=1,2
30100      IF(IFIX(N).NE.1) GO TO 15
30200      IR=IU+N-1
30300      DO 14 N1=2,BW
30400      II=IR+1-N1
30500      IF(II.LT.1.OR.II.GT.NEQ(J)) GO TO 14
30600      B(II)= B(II)-A(II,N1)*BC(NCODE,N)
30700      A(II,N1)= 0.
30800      14 CONTINUE
30900      15 CONTINUE
31000      16 CONTINUE
31100      C***      TRIANGLIZE THE J BLOCK OF COEFFICIENTS
31200      17 NEQJ= NEQ(J)
31300      NEQJ1= NEQ(J+1)
31400      IF(J.EQ.JRAN)NEQJ1=0
31500      N4 = NEQJ+NEQJ1
31600      CALL TRIAN(A,B,NEQJ,NEQJ1,BW)
31700      WRITE(45)((A(N1,N2),N2=1,BW),B(N1),N1=1,NEQJ)
31800      IF(J.EQ.JRAN)GO TO 20
31900      DO 18 N1=1,NEQJ1
32000      N3= NEQJ+N1
32100      B(N1)= B(N3)

```

```

32200      DO 18 N2=1,BW
32300      18 A(N1,N2)= A(N3,N2)
32400      NEQJ=NEQJ1+1
32500      J1= JMIN+J-1
32600      ERR= ERR+1
32700      DO 19 N1= NEQJ,N4
32800      B(N1)= 0.
32900      DO 19 N2=1,BW
33000      19 A(N1,N2)= 0.
33100      20 CONTINUE
33200      RETURN
33300      END
33400      SUBROUTINE STIFF3(RI,RJ,RK,ZI,ZJ,ZK,P,TAU,IPT,C,ET,BFR,BFZ)
33500      C      CALCULATION OF STIFFNESS OF AN ANISOTROPIC TRIANGULAR ELEMENT
33600      C      ELEMENT WITH ANISOTROPIC PROPERTIES AND LINEAR DISPLACEMENTS
33700      IMPLICIT REAL*8(A-H,O-Z)
33800      REAL*8 TITLE,WORD1,WORD2,WORD3
33900      COMMON ISTUFF,S,F,NPCARD,ITOPT,NP,MN
34000      DIMENSIONISTUFF(10162),S(8,8),F(8),A(4,6),CA(4,6),C(4,4),
34100      1 ET(4),CODE(30,50)
34200      DATA A(1,1),A(1,2),A(1,3),A(1,4),A(1,5),A(1,6),A(2,1),A(2,2),A(2,3
34300      1),A(2,4),A(2,5),A(2,6),A(3,1),A(3,2),A(3,3),A(3,4),A(3,5),A(3,6),
34400      2A(4,1),A(4,2),A(4,3),A(4,4),A(4,5),A(4,6)/24*0.0/
34500      ABS(X)=DABS(X)
34600      DO 10 I = 1,7
34700      F(I) = 0.
34800      DO 10 J = 1,7
34900      10 S(I,J) = 0.
35000      DEL=(RJ-RI)*(ZK-ZI)-(RK-RI)*(ZJ-ZI)
35100      1 A(1,1)=(ZJ-ZK)/DEL
35200      A(1,3)=(ZK-ZI)/DEL
35300      A(1,5)=-A(1,3)-A(1,1)
35400      A(2,1)=1./3./RI
35500      A(2,3)=1./3./RJ
35600      A(2,5)=1./3./RK
35700      A(3,2)=(RK-RJ)/DEL
35800      A(3,4)=(RI-RK)/DEL
35900      A(3,6)=-A(3,2)-A(3,4)
36000      A(4,1)=A(3,2)
36100      A(4,2)=A(1,1)
36200      A(4,3)=A(3,4)
36300      A(4,4)=A(1,3)
36400      A(4,5)=A(3,6)
36500      A(4,6)=A(1,5)
36600      DO 2 I=1,4
36700      DO 2 J=1,6
36800      CA(I,J)=0.
36900      DO 2 K=1,4
37000      2 CA(I,J)=CA(I,J)+C(I,K)*A(K,J)
37100      DO 4 I=1,6
37200      DO 3 J=I,6
37300      DO 3 K=1,4

```

```

37400      3 S(I,J)=S(I,J)+A(K,I)*CA(K,J)
37500      DO 4 J=1,4
37600      4 F(I)=F(I)+A(J,I)*ET(J)
37700      VOL = DEL*(RI+RJ+RK)/6.
37800      DO 42 I = 1,6
37900      F(I) = F(I)*VOL
38000      DO 42 J = I,6
38100      S(I,J) = S(I,J)*VOL
38200      42 S(J,I) = S(I,J)
38300      IF (ABS(P)+ABS(TAU)+ABS(BFR)+ABS(BFZ).EQ.0.) GO TO 5
38400      AJ= RJ-RI
38500      AK= RK-RI
38600      BJ= ZJ-ZI
38700      BK= ZK-ZI
38800      IF(ABS(BFR)+ABS(BFZ).EQ.0.)GO TO 41
38900      X1=(RI+RJ+RK)/6.
39000      X2=(RI*(RI+RJ+RK)+RJ*(RJ+RK)+RK*RK)/12.
39100      X3=((RI*RI+RJ*RJ+RK*RK)*(RI+RJ+RK)+RI*RJ*RK)/20.
39200      X4=(BK*(RI+2.*RK+RJ)+BJ*(RI+2.*RJ+RK))/24.
39300      X5=(BK*(RK*(2.*(RI+RJ)+3.*RK)+RI*(RI+RJ)+RJ*RJ)+
39400      1      BJ*(RJ*(2.*(RI+RK)+3.*RJ)+RI*(RI+RK)+RK*RK))/60.
39500      41 F(1)= F(1)+(TAU*AJ-P*BJ)*(RI/2.+AJ/6.)+BFR*((RJ*BK-RK*BJ)*X2+
39600      1      (BJ-BK)*X3+(AK-AJ)*X5)
39700      F(2)= F(2)+(TAU*BJ+P*AJ)*(RI/2.+AJ/6.)+BFZ*((RJ*BK-RK*BJ)*X1+
39800      1      (BJ-BK)*X2+(AK-AJ)*X4)
39900      F(3)= F(3)+(TAU*AJ-P*BJ)*(RI/2.+AJ/3.)+BFR*(-RI*BK*X2+BK*X3-AK*X5)
40000      F(4)= F(4)+(TAU*BJ+P*AJ)*(RI/2.+AJ/3.)+BFZ*(-RI*BK*X1+BK*X2-AK*X4)
40100      F(5)= F(5)+BFR*(RI*BJ*X2-BJ*X3+AJ*X5)
40200      F(6)= F(6)+BFZ*(RI*BJ*X1-BJ*X2+AJ*X4)
40300      5 CONTINUE
40400      RETURN
40500      END

```