NASA TECHNICAL MEMORANDUM

NASA TM X-3325

NASA TM X-3325

A COMPUTER PROGRAM FOR ANISOTROPIC SHALLOW-SHELL FINITE ELEMENTS USING SYMBOLIC INTEGRATION

C. M. Andersen and John T. Bowen

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1976

1. Report No. NASA TM X-3325	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A COMPUTER PROGRAM FOR ANISOTROPIC SHALLOW-SHELL FINITE ELEMENTS USING SYMBOLIC INTEGRATION		5. Report Date June 1976 6. Performing Organization Code
7. Author(s) C. M. Andersen and John T. Bowen		8. Performing Organization Report No. L-10483
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23665		10. Work Unit No. 506-25-99-06 11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Technical Memorandum 14. Sponsoring Agency Code

15. Supplementary Notes

C. M. Andersen: Senior Research Associate in Mathematics and Computer Science, College of William and Mary, Williamsburg, Virginia.

John T. Bowen: Langley Research Center.

16. Abstract

A FORTRAN computer program for anisotropic shallow-shell finite elements with variable curvature is described. A listing of the program is presented together with printed output for a sample case. Computation times and central memory requirements are given for several different elements.

The program is based on a stiffness (displacement) finite-element model in which the fundamental unknowns consist of both the displacement and the rotation components of the reference surface of the shell. Two triangular and four quadrilateral elements are implemented in the program. The triangular elements have 6 or 10 nodes, and the quadrilateral elements have 4 or 8 nodes. Two of the quadrilateral elements have internal degrees of freedom associated with displacement modes which vanish along the edges of the elements (bubble modes). The triangular elements and the remaining two quadrilateral elements do not have bubble modes.

The output from the program consists of arrays corresponding to the stiffness, the geometric stiffness, the consistent mass, and the consistent load matrices for individual elements. The integrals required for the generation of these arrays are evaluated by using symbolic (or analytic) integration in conjunction with certain group-theoretic techniques. The analytic expressions for the integrals are exact and were developed using the symbolic and algebraic manipulation language MACSYMA.

17. Key Words (Suggested by Author(s))		18. Distribution Statement
Finite elements Composite materials Shells Symbolic integration Computers Anisotropy	Group theory MACSYMA Stiffness	Unclassified – Unlimited Subject Category 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	126	$5.75

*For sale by the National Technical Information Service, Springfield, Virginia 22161

A COMPUTER PROGRAM FOR ANISOTROPIC SHALLOW-SHELL FINITE ELEMENTS USING SYMBOLIC INTEGRATION

C. M. Andersen* and John T. Bowen
Langley Research Center

SUMMARY

A FORTRAN computer program for anisotropic shallow-shell finite elements with variable curvature is described. A listing of the program is presented together with printed output for a sample case. Computation times and central memory requirements are given for several different elements.

The program is based on a stiffness (displacement) finite-element model in which the fundamental unknowns consist of both the displacement and the rotation components of the reference surface of the shell. Two triangular and four quadrilateral elements are implemented in the program. The triangular elements have 6 or 10 nodes, and the quadrilateral elements have 4 or 8 nodes. Two of the quadrilateral elements have internal degrees of freedom associated with displacement modes which vanish along the edges of the elements (bubble modes). The triangular elements and the remaining two quadrilateral elements do not have bubble modes.

The output from the program consists of arrays corresponding to the stiffness, the geometric stiffness, the consistent mass, and the consistent load matrices for individual elements. The integrals required for the generation of these arrays are evaluated by using symbolic (or analytic) integration in conjunction with certain group-theoretic techniques. The analytic expressions for the integrals are exact and were developed using the symbolic and algebraic manipulation language MACSYMA.

INTRODUCTION

This paper contains a description and listing of SYMINSE (symbolically integrated shell elements), a FORTRAN computer program for computing the characteristic arrays (stiffness, geometric stiffness, consistent mass, and consistent load) associated with anisotropic shallow-shell finite elements with variable curvature. The SYMINSE program is based on a stiffness (displacement) finite-element model having five fundamental

*Senior Research Associate in Mathematics and Computer Science, College of William and Mary, Williamsburg, Virginia.

unknowns (dependent variables) and on a form of shallow-shell theory which includes the effects of shear deformation, rotary inertia, and bending-extensional coupling.

Two triangular and four quadrilateral elements are implemented in SYMINSE. The two triangular elements are ST6 with 6 nodes and ST10 with 10 nodes. These elements have 30 and 50 degrees of freedom per element, respectively. Two of the four quadrilateral elements have bubble modes. These are SQ5 and SQ9 with 4 and 8 nodes, respectively, and 25 and 50 degrees of freedom, respectively. The two remaining elements are SQ4 and SQ8 which also have 4 and 8 nodes, respectively, but since they have no bubble modes, they have only 20 and 40 degrees of freedom, respectively.

The SYMINSE program is intended to be used as part of a finite-element system whose capabilities would include analysis of laminated composite (and therefore anisotropic) shells. SYMINSE deals only with individual elements, and other modules in the system must be relied upon for such operations as (1) determining the positions of the nodes and the connectivities of the elements, (2) determining the shell stiffnesses from the thicknesses and material properties of the lamina, (3) assembling the total stiffness matrix from the elemental stiffness matrices generated by SYMINSE, (4) solving the assembled system of equations, and (5) displaying the solutions.

Whereas the conventional approach for evaluating the element characteristic arrays depends on numerical quadrature, the SYMINSE program relies entirely on symbolic (or analytic) integration implemented with the aid of group-theoretic techniques. The symbolic expressions for the integrals to be numerically evaluated by the SYMINSE program were computed using the algebraic and symbolic manipulation language MACSYMA.[1] SYMINSE itself has been run on the CONTROL DATA 6000, CYBER 70, and CYBER 170 series computer systems.

SYMBOLS

$A^{ijk\ell}, B_{\alpha}^{ijk}, C_{\alpha\beta}^{ij}$ basic integrals

$[K]$ element stiffness matrix

K_{IJ}^{ij} stiffness coefficients of shell element

$[\overline{K}]$ geometric stiffness matrix

---

[1]The MACSYMA system is being developed by the Mathlab group at Massachusetts Institute of Technology under the support of the Advanced Research Projects Agency of the U.S. Department of Defense (work order 2095) through Office of Naval Research Contract No. N00014-75-C-0661.

$\bar{K}_{IJ}^{ij}$ geometric stiffness coefficients of shell element

[M] consistent mass matrix

M_{IJ}^{ij} consistent mass coefficients of shell element

$m, \bar{m}, \bar{\bar{m}}$ superscript index designating particular representative A-, B-, or C-integrals, respectively

$n, \bar{n}, \tilde{\bar{n}}$ superscript indices designating particular group transformations

$\{P\}$ consistent load vector

P_J^j consistent load coefficients

R's, S's, T's coefficients associated with A-, B-, and C-integrals, respectively

$\mathcal{R}$'s, $\mathcal{S}$'s, $\mathcal{T}$'s integers associated with representative A-, B-, and C-integrals, respectively

r number of shape functions associated with a finite element

$s, \tilde{s}, t, \tilde{t}$ functions of nodal coordinates (see eqs. (34) and (39) of ref. 1)

x_α^i coordinates of ith corner node

STATEMENT OF THE PROBLEM AND TECHNIQUES FOR SOLUTION

The analytic formulation of the shallow-shell theory implemented in SYMINSE and the major techniques employed to gain computational speed are described in reference 1. In particular, reference 1 describes (1) how the components of the characteristic arrays are formed as linear combinations over certain sets of integrals referred to as the A-, B-, and C-integrals, (2) the forms of the analytic expressions for these sets of integrals, and (3) how group-theoretic techniques reduce the number of symbolic computations to be performed in the course of developing a program such as SYMINSE. Reference 1 also includes a demonstration of the efficiency of the SYMINSE program by giving a comparison

of the number of floating-point arithmetic operations required by SYMINSE with the number required by a conventional numerical quadrature approach.

All the equation numbers in this report refer to equations in reference 1.

PROGRAM DESCRIPTION

Major Features and Capabilities

The element characteristic arrays generated by the SYMINSE program are the stiffness SS, the geometric stiffness SG, the consistent load SP, and the consistent mass SM. The symbols SS, SG, SP, and SM are the FORTRAN names for arrays which correspond to the matrices $[K]$, $[\overline{K}]$, $\{P\}$, and $[M]$, respectively, of equation (13) (for details of this correspondence, see the subsection "Program Output"). (Recall that all referenced equations are given in ref. 1.)

The six "types" of elements implemented in the SYMINSE program (see table I and fig. 2 both of ref. 1) are characterized by different values of the FORTRAN variable NSF, which represents the number r of shape functions per element (cf., the description of eq. (6)). The user selects the type of element simply by setting NSF equal to 4, 5, 6, 8, 9, or 10 in the program which calls SYMINSE. The FORTRAN variable NNE, which represents the number of nodes per element, is set by the SYMINSE program equal to NSF - 1 when there is a bubble mode and equal to NSF otherwise.

The SYMINSE program, unlike conventional finite-element programs, does not employ numerical quadrature for evaluation of any integrals. Instead, exact analytic expressions for the integrals are used throughout. Some of the expressions for the integrals involve logarithmic functions, and for these functions, high-accuracy truncated power series expansions are employed. Apart from this, the only inaccuracies in the evaluation of integrals are due to roundoff error.

For quadrilateral elements, different portions of the SYMINSE code are employed for evaluating the C-integrals (see ref. 1) depending on whether the element is a parallelogram (including a rectangle), a trapezoid, or a trapezium (a quadrilateral which has no two sides which are parallel). Separate code is used for each of these three cases because the parallelogram code (based on eq. (43)) is faster than the trapezoid code (based on eqs. (41) and (42)), which in turn is faster than the trapezium code (based on eqs. (37) and (38) for NNE = 4 and on eq. (30) for NNE = 8).

By testing functions of the coordinates of the corner nodes, the SYMINSE program automatically determines whether a quadrilateral element is a parallelogram, a trapezoid, or a trapezium. The FORTRAN variables RR and SS stand for the quantities $\tilde{s}$ and $\tilde{t}$,

respectively, defined in equation (39). SYMINSE deems the element to be a parallelogram (PARA = TRUE) if the absolute values of RR and SS are each less than EPS; otherwise it sets PARA equal to FALSE. It deems the element to be a trapezoid (TRAP = TRUE) if the absolute value of RR or the absolute value of SS is less than EPS; otherwise the element is deemed a trapezium (TRAP = FALSE). The value of EPS has been set equal to 10^{-3}.

There are five logical variables – SGFLAG, SPFLAG, SMFLAG, CURVE, and PRFLAG – set by the calling program which affect the flow through the SYMINSE program. The stiffness SS is computed each time SYMINSE is called, but the arrays SG, SP, and SM are computed only if the corresponding flags SGFLAG, SPFLAG, and SMFLAG have been set equal to TRUE. The flag CURVE should be set to TRUE if the particular shell element has curvature and should be set to FALSE for a flat-plate element. Finally, PRFLAG governs whether the characteristic arrays are to be printed. However, the computed arrays are stored on sequential files on disk whether PRFLAG is TRUE or not. Setting one or more of these five flags to FALSE will save computer time but will not affect the field-length requirements.

The loading forces P, P1, P2 and curvature components Q1, Q2, Q12, which are to be specified at each node, are inputs to SYMINSE. Internally they are approximated by using the same shape functions as the fundamental unknowns. All other inputs such as the thickness H, the density RHO, the prestress coefficients EN1, EN2, EN12 and the material stiffness coefficients are assumed to be constant throughout the element.

The first time SYMINSE is called, it enters an initialization phase in which coefficients and indices for computing integrals are evaluated. The coefficients are the R's, S's, and T's of equations (23) to (25), (28), (29), and (45), and the indices are the $\bar{\bar{m}}$'s and $\bar{\bar{n}}$'s of equation (62). The array NRECORD determines for which values of NSF these coefficients are to be evaluated. As sets of these coefficients and indices are evaluated, they are written out on a random access disk file for subsequent use.

Overlay structures have been employed to reduce the amount of central memory (field length) required. The program has a main overlay, which resides in core at all times, 11 primary overlays, and no secondary overlays. The primary overlay which contains the initializing routines SETUP, SETA, SETB, and SETC is called only once during each computer run. Another primary overlay, containing the routine PRINT, is called only if printed results are desired. A third, containing the routine SGPM, is called once for each element and forms the components of the element characteristic arrays as linear combinations of the A-, B-, and C-integrals.

The SYMINSE program can readily be interfaced with other modules of a finite-element system. Unlabeled common is not used, and all but two arrays in labeled common can be shared with other program modules. The two arrays are NRECORD with dimension 7 in labeled common SPACE and IX with dimension 31 in labeled common STORE.

The normal input file TAPE5 is not used by any of the SYMINSE routines in order to allow the calling program to use TAPE5 without interference from SYMINSE. The read statements within SYMINSE reference either the disk file TAPE1, which contains a fixed block of data not to be changed by the user, or random access records previously written by SYMINSE on the temporary disk file TAPE2. The arrays written on the random access file TAPE2 correspond to the R's, S's, and T's of equations (23) to (25), (28), (29), and (45); the indices m, n, $\overline{m}$, $\overline{n}$, $\overline{\overline{m}}$, and $\overline{\overline{n}}$ of equation (62); and a small number of other quantities. The characteristic arrays SS, SG, SP, and SM generated by SYMINSE are written out on the disk files TAPE3, TAPE4, TAPE8, and TAPE9, respectively.

A listing of SYMINSE is presented in appendix A. The programing style adopted for writing this program was strongly influenced by reference 2. It is hoped that the use of ELSE and THEN comment cards as well as the system of indentation adopted will improve the readability of the program.

Program Input

The input to SYMINSE consists of three blocks of data. The first block contains six fixed sets of integer arrays which are read in from TAPE1 by the routine SETUP in the form of card images. These six sets of data correspond to the six allowed values of NSF (viz, 4, 5, 6, 8, 9, and 10) and are given in appendix B. Each set contains arrays called KA, LA, QA, KB, LB, QB, KC, LC, and QC except that LA is missing from the third and sixth sets.

The arrays QA, QB, and QC, which correspond to the $\mathcal{R}$'s, $\mathcal{S}$'s, and $\mathcal{T}$'s of equations (53) to (55) and (57) to (59) of reference 1, were generated by MACSYMA programs; whereas the arrays KA, LA, KB, LB, KC, and LC, which contain values of the indices m, n, $\overline{m}$, $\overline{n}$, $\overline{\overline{m}}$, and $\overline{\overline{n}}$, were generated by FORTRAN programs.

The $A^{ijk\ell}$ integrals to be evaluated and stored in the array XA may be thought of as having $i \geq j \geq k \geq \ell$. The four indices subject to this restriction can be replaced by a single index given by

$$I1 = \frac{i(i+1)(i+2)(i+3)}{24} + \frac{j(j+1)(j+2)}{6} + \frac{k(k+1)}{2} + \ell + 1$$

which runs from 1 to IXA = (r+1)(r+2)(r+3)(r+4)/24. The arrays KA and LA contain the values of $m(I1) = m(i,j,k,\ell)$ and $n(I1) = n(i,j,k,\ell)$. Similarly, the arrays KB and LB contain the values of $\overline{m}(i,j,k)$ and $\overline{n}(i,j,k)$ for $i \geq j$, and the arrays KC and LC contain the values of $\overline{\overline{m}}(i,j)$ and $\overline{\overline{n}}(i,j)$ for $i \geq j$.

The second block of data consists of variables defined in the calling program and stored by it in the first 100 words of the labeled common SPACE. These variables are

not modified by the SYMINSE program and may be changed by the calling program between calls to SYMINSE. They are listed in table I in the order in which they appear in the common block. Some of these variables do not need to be defined under certain circumstances. These variables are

P, P1, P2	if SPFLAG = FALSE
Q1, Q2, Q12	if CURVE = FALSE
RHO, H	if SMFLAG = FALSE
EN1, EN2, EN12	if SGFLAG = FALSE
X(4), Y(4)	if NSF = 6 or 10

In addition, only the first NNE components of Q1, Q2, Q12, P, P1, and P2 need to be defined even when the curvatures and/or load components are needed. The quantities SPFLAG, SMFLAG, SGFLAG, and CURVE are discussed in more detail subsequently.

The third block of data consists of the vector NRECORD which has dimension 7. This vector, like the variables of the second block of data, is initially defined by the calling program and is stored in the labeled common SPACE. However, NRECORD is modified by the routine SETUP on the first call to SYMINSE and must not be modified by the calling program between calls on SYMINSE. If the ith component of NRECORD, as initially defined, is nonzero, then from the first block of data the set of integer arrays corresponding to NSF = i + 3 will be read in and processed by the routine SETUP. For example, in a computer run involving only finite elements with NSF = 5, NRECORD may be set to (0,1,0,0,0,0,0). On the other hand, elements with all six allowed values of NSF may be processed during the same computer run by setting NRECORD to (1,1,1,0,1,1,1). Since NSF = 7 is not allowed, the value of the fourth component of NRECORD is ignored; but to simplify programing, one blank card is situated between the third and fourth sets of input data. This card is skipped when the fourth component of NRECORD is referenced.

Program Output

The output from the SYMINSE program consists primarily of the arrays SS, SG, SP, SM, and SMASS. The first four of these are stored in the binary sequential disk files TAPE3, TAPE4, TAPE8, and TAPE9, respectively, so that the system equations can be assembled by some other module of the finite-element system. These four arrays correspond to the stiffness K_{IJ}^{ij}, the geometric stiffness $\bar{K}_{IJ}^{ij}$, the consistent load P_J^j, and the

consistent mass M_{IJ}^{ij}, respectively, of equation (12) of reference 1. The fifth array SMASS is constructed only if the routine PRINT is called. It corresponds more directly to M_{IJ}^{ij} than SM does, but it is much larger than SM and contains many zeros. It is expected that the full consistent mass matrix will be constructed by another program module from the SM arrays stored in TAPE8.

The computed arrays are stored as if they had the following DIMENSION statement (were four-dimensional arrays to be legal in FORTRAN):

```
DIMENSION SS(5,NSF,5,NSF),SG(NSF,NSF),SP(5,NNE),SM(NSF,NSF),SMASS(5,NSF,5,NSF)
```

The explicit relations between the computed arrays and the characteristic matrices of equation (12) are

$$K_{IJ}^{ij} = SS(I,i,J,j)$$

$$\overline{K}_{IJ}^{ij} = \begin{cases} SG(i,j) & \text{for } I = J = 3 \\ 0 & \text{otherwise} \end{cases}$$

$$P_J^j = \begin{cases} SP(J,j) & \text{for } j \leq NNE \\ 0 & \text{for } j = NNE + 1 \end{cases}$$

$$M_{IJ}^{ij} = \begin{cases} SM(i,j) & \text{for } I = J = 1, 2, \text{ or } 3 \\ SM(i,j) * H^2/12. & \text{for } I = J = 4 \text{ or } 5 \\ 0 & \text{for } I \neq J \end{cases}$$

$$M_{IJ}^{ij} = SMASS(I,i,J,j)$$

The printed output from SYMINSE consists of

(1) The integration arrays KA, LA, QA, KB, LB, QB, KC, LC, and QC (printed by the routine SETUP)

(2) The required length of the labeled common SPACE as dependent on NSFM (printed by the routine SETUP)

(3) The arrays SS, SG, SP, SM, and SMASS which have been evaluated (printed by the program PRINT only if PRFLAG = TRUE)

Program Organization

The computational processes implemented in the SYMINSE program are outlined in figure 1(a) for triangular and parallelogram elements and in figure 1(b) for trapezoidal and trapezium elements. They differ only in the method of computation of the C-integrals.

The routines in the SYMINSE program, their field lengths, the files they reference, and brief descriptions are listed in table II, and the subroutine linkages of the SYMINSE program are given in figure 2. The subroutine ELEMENT and the program interfacing SYMINSE with the other modules of the finite-element system are to be contained in the main overlay. The large boxes in figure 2 represent main programs of different primary overlays. Each of the small boxes below one of these large boxes represents a subroutine located in the same overlay as the main program above it. There are several different subroutines named XDNDN which appear in table II and figure 2. None of these are alike except that they all evaluate C-integrals. There should be no confusion among them since each appears in a different overlay.

As indicated in figure 2, the SYMINSE program is entered by a call to the subroutine ELEMENT. The flow chart for ELEMENT, given in figure 3, indicates that ELEMENT makes calls on five basic routines, namely, SETUP, INTGRAL, SGPM, STORE, and PRINT. It also indicates that the initializing program SETUP is called only once during each computer run and that the routine PRINT is bypassed if the logical variable PRFLAG has been set to FALSE by the calling program.

The function of the routine SETUP and its subroutines SETA, SETB, and SETC is to set up the "integration" arrays used in the evaluation of A-, B-, and C-integrals. The inputs to these initializing routines consist of the array NRECORD in common SPACE and the six sets of data listed in appendix B, which are read in from TAPE1. The initializing routines affect the execution of the rest of the program only through the integration arrays they store in binary on the random access disk file TAPE2. These routines also produce a printed record of these integration arrays.

The subroutine INTGRAL manages the evaluation of the A-, B-, and C-integrals. Its flow chart is given in figure 4. The only parameters specified by the calling program which can affect the flow through INTGRAL are NSF and the coordinates of the corner nodes. For a triangular element (NSF = 6 or 10) the flow is independent of the coordinates of the corner nodes, but for a quadrilateral element (NSF = 4, 5, 6, or 9) the program QUAD sets PARA to TRUE if the element is deemed a parallelogram and to FALSE otherwise. If PARA has been set to FALSE, then the program QUAD sets TRAP to TRUE if the element is deemed a trapezoid and to FALSE if it is deemed a trapezium. The effects of

PARA and TRAP on the program flow are shown in figure 4. Two of the many routines called by INTGRAL are TRI and QUAD. These are the routines which read the integration arrays stored by the initializing routines.

The program SGPM manages the evaluation of the characteristic arrays (SS, SG, SP, and SM), which are formed from linear combinations of the A-, B-, and C-integrals. There are two flags set by the calling program which affect the flow through SGPM. They are SPFLAG and SMFLAG. The consistent load SP is evaluated only if SPFLAG is TRUE, and the consistent mass SM is evaluated only if SMFLAG is TRUE. Two other flags, CURVE and SGFLAG, set by the calling program affect the flow in LINSTF, the subroutine called by SGPM to compute the stiffness SS and the geometric stiffness SG. The curvature-dependent terms in SS are omitted if CURVE is FALSE, and SG is evaluated only if SGFLAG is TRUE.

The subroutine STORE takes the characteristic arrays which have been computed and writes them out as binary sequential files on disk storage. The fifth and last routine called by ELEMENT is the program PRINT, which provides a printed record of the characteristic arrays computed by SGPM. First, PRINT displays the stiffness array SS. It then displays SG, SP, and SM if they have been evaluated. Finally, it reconstructs the full consistent mass matrix SMASS and displays it. However, in doing so it writes SMASS, which has dimension $(5*NSF)^2$, over that portion of memory previously occupied by SS.

Storage of Data

The FORTRAN variables employed in the SYMINSE program fall into the following categories: (1) the input variables stored in the first 100 words of labeled common SPACE, (2) other variables stored in fixed positions in SPACE, (3) dynamically allocated arrays in common SPACE, (4) variables stored in fixed positions in labeled common TEMP, (5) the array IX in labeled common STORE, and (6) certain DO loop indices and the variables defined by DATA statements.

The FORTRAN variables in category (6) are the only ones which were not placed in a labeled common. The lengths of the arrays in category (3) depend on the value of the FORTRAN variable NSF (number of shape functions per element). By not assigning these arrays to fixed positions in memory, the size of the labeled common SPACE can be considerably reduced for computer runs in which the larger elements are not to be computed (see the subsection "Operating Instructions"). The variables and arrays in categories (1), (2), and (3) are listed in tables I, III, and IV, respectively. Each of the variables in categories (2) and (3) is referenced by routines in more than one overlay, whereas those in category (4) are referenced within one overlay only. A list of the more important variables of category (4) is given in table V. The array IX from category (5) contains the

indices needed by the mass-storage subroutines. It is important that IX, like the array NRECORD, not be modified between calls to the SYMINSE program.

Use of Computerized Algebraic Manipulation

The symbolic and algebraic manipulation language MACSYMA (see refs. 3, 4, and 5) played a central role in the development of the SYMINSE program. In addition to many exploratory calculations, symbolic manipulation was used to compute

(1) The input arrays of $\mathcal{R}$'s, $\mathcal{S}$'s, and $\mathcal{T}$'s which were derived from the evaluation of the logarithm-free representative integrals

(2) The FORTRAN expressions for the representative C-integrals over trapezoids and trapeziums, found in overlays numbered 6 through 11_8 (see table II)

(3) The truncated power series expansions for the logarithmic functions evaluated in the subroutines BLOG, ELOG, WLOG1, and WLOG2

Some of the symbolic expressions computed by MACSYMA were converted into FORTRAN expressions by commands within the MACSYMA programs. These FORTRAN expressions were edited in TECO (a text editing language for the DEC PDP-10 computer) to add decimal points and to format continuation lines appropriately. They were then punched on cards and incorporated into the SYMINSE program. FORTRAN expressions generated in this way can be found in overlays 7 through 11_8.

The other symbolic expressions were computed by MACSYMA but then were hand coded in FORTRAN. Some of these (for example, the FORTRAN statements in the function subroutine XDNDN in overlay 6) were then checked by using symbolic manipulation. This was done by giving as input to MACSYMA the sequence of FORTRAN statements needed to numerically evaluate (in closed form) one of the integrals. MACSYMA was then used to carry out the substitutions indicated in order to form a single large analytic expression for the integral in question. This expression was then subtracted from the quantity derived by symbolically integrating the corresponding integrand. The FORTRAN expressions are accurate if this difference evaluates to zero.

PROGRAM USAGE

Operating Instructions

The program which calls the SYMINSE program should have the following statements or their equivalent:

```
OVERLAY (MAIN,0,0)
PROGRAM MAIN (INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5=INPUT,
  TAPE6=OUTPUT,TAPE8,TAPE9)
                .
                .
                .
COMMON/TEMP/TEMP(60)
COMMON/SPACE/SPACE(5261)
COMMON/STORE/STORE(31)
                .
                .
                .
CALL ELEMENT
                .
                .
                .
END
```

In addition, the user's control cards should equivalence the file TAPE1 to a file which contains the card images listed in appendix B. The length of the labeled common SPACE depends on the highest values of NSF referenced in the variable NRECORD. The required lengths are given in the following table:

Highest value of NSF	Length of common SPACE	Highest value of NSF	Length of common SPACE
4	780	8	3157
5	1180	9	4125
6	1686	10	5261

For any practical production run it is expected that no more than two of the three characteristic arrays SG, SP, and SM will be calculated. Thus the header card of the calling program can be modified to delete reference to the input-output files which will not be called upon.

Some savings of computer resources can often be effected by loading only those primary overlays which are needed for a given computer run. The main overlay and primary overlays 1 and 12_8 are always required. The primary overlay 13_8 is needed for test runs in which printed output of the characteristic arrays is desired but should not be needed for

production runs. The primary overlays required for computing A-, B-, and C-integrals for the various elements are given in the following table:

Element shape	NSF	Overlays required
Triangle	6 or 10	2
Parallelogram	4, 5, 8, or 9	3
Trapezoid	4 or 5 8 or 9	3 and 4 3 and 5
Trapezium	4 or 5 8 9	3 and 6 3, 7, and 10_8 3, 7, 10_8, and 11_8

The field lengths required to run the SYMINSE program vary depending on the space allotted to labeled common, the files referenced, and the overlays selected. The field lengths required for several test cases are given in table VI.

No library subroutines are referenced by the SYMINSE program.

Sample Calling Program and Sample Program Output

A sample program which calls SYMINSE to generate the characteristic arrays for 14 different shell elements is given in appendix C. Table VI gives the central processing unit (CPU) times required for each of these finite elements as well as for a corresponding set of plate elements. Each of these plate elements differs from its corresponding shell element only in that the parameter CURVE has been set to FALSE. Effects of bending-extensional coupling are still included. The timings given do not include the time taken to write the computed arrays on either the OUTPUT file or disk storage. The CPU time in the SETUP overlay is 1.04 seconds when all six element types are set up. The timing routine is referenced by the subroutine ELEMENT.

Sample output from SYMINSE for finite elements with NSF = 5 and 6 is given in appendix D. The appropriate modifications to the calling program are also given in this appendix.

POSSIBLE FURTHER IMPROVEMENTS AND DEVELOPMENTS

There are several ways in which the SYMINSE program could be modified to improve its performance or could be extended to cover additional elements:

(1) The program can be made more efficient for constant or zero curvatures since in these special cases the only A- and B-integrals required have the forms A^{ij00} and

B_{α}^{i0k}. For this purpose, the computed A- and B-integrals should have a different ordering in the arrays XA and XB, and a new flag should be introduced to signal constant curvature.

(2) The routines QUAD81, QUAD82, and QUAD9, which evaluate C-integrals for trapezium (nontrapezoidal quadrilateral) elements with NNE = 8, are based on equation (30). (Recall that referenced equations are given in ref. 1.) Consequently, roundoff errors can be severe when these routines are used for small values of RR or SS. This difficulty can be avoided by reformulating the routines so that they are based on equations (37) and (38) instead of equation (30). This reformulation has successfully been carried out for the routine QUAD5.

(3) The extension of the SYMINSE program to add triangular elements with more than 10 nodes and parallelogram and trapezoidal elements with more than 8 nodes is a straightforward task. However, for such trapezoidal elements, further "subtractions" on the function $\overline{L}$ of equation (42) may be necessary. Without resorting to numerical quadrature, the addition of trapezium elements with more nodes would be more difficult.

(4) The previous comment also applies for an extension which would add quadrilateral elements with additional "bubble modes." In this case, however, the group-theoretic techniques described in the present study need a slight generalization which is described in reference 6.

(5) For trapezium elements a hybrid approach which combines numerical quadrature with symbolic integration appears to have advantages over a purely numerical quadrature or a purely symbolic integration approach. In the hybrid approach the A- and B-integrals would be evaluated by symbolic integration and the C-integrals by numerical quadrature. This approach would retain the major advantages resulting from the symbolic integration of the A- and B-integrals and eliminate the difficulties associated with the symbolic integration of the C-integrals. The hybrid approach may be particularly advantageous for higher order elements for which the symbolic expressions for the C-integrals become both numerous and highly complicated and for which roundoff errors can become severe unless several "subtractions" on the logarithmic functions L_1 and L_2 of equation (33) are performed. A count of floating-point arithmetic operations suggests that, even for a purely numerical quadrature approach, evaluation of A-, B-, and C-integrals followed by forming the stiffness as a linear combination of these integrals is faster than the conventional approach discussed in the section on program performance of reference 1. This suggests that symbolic integration and numerical quadrature can be readily combined in one program.

CONCLUDING REMARKS

Triangular and quadrilateral shear-flexible finite elements for shallow anisotropic shells with variable curvatures have been implemented in a FORTRAN computer program called SYMINSE. A listing is given of this program, together with a sample calling program and some sample output. A stiffness (displacement) formulation is used with the fundamental unknowns consisting of both the displacement and the rotation components of the reference surface of the shell. The triangular elements implemented have 30 or 50 degrees of freedom per element, and the quadrilateral elements have 20, 25, 40, and 45 degrees of freedom per element.

The SYMINSE program does not use numerical quadrature but instead uses exact analytic expressions for all the integrals needed. These expressions are obtained by symbolically integrating certain selected representative integrals and using group-theoretic techniques to relate the other required integrals to these representative integrals.

Both the theortical ideas used in the SYMINSE program and the performance of the program were discussed in a companion paper in which evidence was given to indicate that SYMINSE would be faster than any equivalent program based on conventional techniques. It is believed that some of the new techniques implemented in SYMINSE will prove valuable for other finite-element applications as well.

Langley Research Center
National Aeronautics and Space Administration
Hampton, Va. 23665
April 12, 1976

APPENDIX A

LISTING OF THE SYMINSE COMPUTER PROGRAM

```
      SUBROUTINE ELEMENT                                                    159
C*****************************************************************************160
C                                                                           161
C     THIS SUBROUTINE IS THE TOP-LEVEL ROUTINE OF THE  S Y M I N S E        162
C        (SYMBOLICALLY INTEGRATED SHELL ELEMENTS) PROGRAM                   163
C        FOR EVALUATION OF THE LINEAR STIFFNESS SS, THE GEOMETRIC           164
C        STIFFNESS SG, THE CONSISTENT LOAD SP AND THE CONSISTENT MASS SM    165
C        FOR A DOUBLY-CURVED ANISOTROPIC SHALLOW-SHELL FINITE ELEMENT.      166
C                                                                           167
C                                                                           168
C                                                                           169
C     THIS PROGRAM WAS WRITTEN BY                                           170
C                  CARL M. ANDERSEN                                         171
C                  SENIOR RESEARCH ASSOCIATE IN MATHEMATICS                 172
C                       AND COMPUTER SCIENCE                                173
C                  DEPARTMENT OF MATHEMATICS                                174
C                  COLLEGE OF WILLIAM AND MARY                              175
C                  WILLIAMSBURG, VA. 23185                                  176
C     WITH THE ASSISTANCE OF                                                177
C                  JOHN T. BOWEN                                            178
C                  N.A.S.A. LANGLEY RESEARCH CENTER                         179
C                  HAMPTON, VA. 23665                                       180
C                                                                           181
C                                                                           182
C     THE RESULTS (SS,SG,SP AND SM) ARE STORED IN BINARY SEQUENTIAL         183
C        FILES ON DISC LOGICAL UNITS  3,4,8, AND 9, RESPECTIVELY.           184
C        FOR FUTURE ASSEMBLY OF THE SYSTEM EQUATIONS.                       185
C           EACH  SS  RECORD CONTAINS  25*NSF*NSF  WORDS                    186
C           EACH  SG  RECORD CONTAINS    NSF*NSF   WORDS                    187
C           EACH  SP  RECORD CONTAINS     5*NNE    WORDS                    188
C           EACH  SM  RECORD CONTAINS    NSF*NSF   WORDS                    189
C                                                                           190
C     NSF  IS THE NUMBER OF SHAPE FUNCTIONS ASSOCIATED WITH THE ELEMENT.    191
C        FOR A TRIANGULAR ELEMENT  NSF  MAY TAKE THE VALUES  6  OR 10.      192
C        FOR A QUADRILATERAL ELEMENT NSF MAY TAKE THE VALUES 4,5,8 OR 9.    193
C     NNE  IS THE NUMBER OF NODES ASSOCIATED WITH THE ELEMENT.              194
C        NNE = NSF     IF  NSF=4,6,8 OR 10.                                 195
C        NNE = NSF-1   IF  NSF=5 OR 9.                                      196
C                                                                           197
C     THE MAIN BLOCK OF INPUT COMPRISES THE FIRST ONE HUNDRED WORDS OF      198
C        COMMON/SPACE/.  NONE OF THE WORDS IN THIS BLOCK ARE CHANGED BY     199
C        THE SYMINSE PROGRAM, BUT ANY OF THEM MAY BE CHANGED,IF DESIRED,    200
C        BY OTHER PROGRAMS BETWEEN CALLS TO ELEMENT.                        201
C     THE VARIABLES CONTAINED IN THIS BLOCK ARE AS FOLLOWS.                 202
C        THE FIRST  21 WORDS ARE THE MATERIAL STIFFNESS PROPERTIES ---      203
C          C11,C12,C16,C22,C26,C66,F11,F12,F16,F22,F26,F66,                 204
C          D11,D12,D16,D22,D26,D66,C55,C44,C54.                             205
C        THE NEXT  3  WORDS ARE THE PRESTRESS COEFFICIENTS EN1,EN2,EN12.    206
C        THE NEXT WORD CONTAINS  NSF.                                       207
C        THE NEXT  5  WORDS CONTAIN LOGICAL VARIABLES                       208
C           CURVE -- IF FALSE, ALL CURVATURE DEPENDENT CONTRIBUTIONS        209
C              TO SS ARE BYPASSED IN THE SUBROUTINE  LINSTF. THIS SPEEDS    210
C              UP THE COMPUTATION FOR THE CASE OF ZERO CURVATURE.           211
C           SGFLAG -- IF TRUE THEN THE GEOMETRIC STIFFNESS SG IS            212
C              COMPUTED. IF FALSE SG WILL CONTAIN GARBAGE.                  213
C           SMFLAG -- IF TRUE THEN THE CONSISTENT MASS SM IS COMPUTED.      214
C              IF FALSE THE SUBROUTINE MASS WILL NOT BE CALLED AND          215
C              SM WILL CONTAIN GARBAGE.                                     216
C           SPFLAG -- IF TRUE THEN THE CONSISTENT LOAD SP IS COMPUTED.      217
C              IF FALSE THE SUBROUTINE LODVEC WILL NOT BE CALLED AND        218
C              SP WILL CONTAIN GARBAGE.                                     219
C           PRFLAG -- IF TRUE THEN THE PROGRAM OUTPUT (OVERLAY 13,0) IS     220
C              CALLED TO PRINT THE RESULTS  SS,SG,SP AND SM.                221
```

```
C         THE NEXT WORD CONTAINS THE DENSITY,RHO, OF THE MATERIAL.                222
C         THE NEXT WORD CONTAINS THE THICKNESS,H, OF THE MATERIAL.                223
C         THE NEXT  4  WORDS CONTAIN THE X-COORDINATES OF THE CORNER              224
C            NODES.  FOR TRIANGLES, ONLY THE FIRST THREE OF THESE WORDS           225
C            ARE USED.                                                            226
C         THE NEXT  4  WORDS CONTAIN THE Y-COORDINATES OF THE CORNER              227
C            NODES.  FOR TRIANGLES, ONLY THE FIRST THREE OF THESE WORDS           228
C            ARE USED.                                                            229
C         THE NEXT  30  WORDS ARE RESERVED FOR NODAL VALUES OF THE                230
C            CURVATURES Q1,Q2 AND Q12.                                            231
C         FINALLY THE LAST  30  WORDS ARE RESERVED FOR THE NODAL VALUES           232
C            OF THE PRESSURES  P1,P2 AND P12.                                     233
C                                                                                 234
C      IN ADDITION TO THE ABOVE ONE HUNDRED WORDS OF INPUT DATA, THE              235
C         CALLING PROGRAM MUST ON THE FIRST CALL TO ELEMENT PROVIDE THE           236
C         VECTOR NRECORD(7).  NRECORD IS SUBSEQUENTLY MODIFIED BY THE             237
C         PROGRAM SETUP.                                                          238
C                                                                                 239
C      THE VARIABLES  NNE,ISS,IXC,IXA  ARE DEFINED BY THE PROGRAMS               240
C         TRI  OR  QUAD.                                                          241
C                                                                                 242
C      EACH OF THE VARIABLES (OTHER THAN THE INPUT VARIABLES) STORED IN           243
C         COMMON/SPACE/  IS REFERENCED BY PROGRAMS IN MORE THAN ONE               244
C         OVERLAY OF THE SYMINSE MODULE.  BY CONTRAST, THE VARIABLES              245
C         STORED IN  COMMON/TEMP/  ARE REFERENCED BY PROGRAMS WITHIN ONE          246
C         OVERLAY ONLY.                                                           247
C                                                                                 248
C      IT IS IMPORTANT TO REMARK THAT THE VARIABLE  NRECORD                       249
C         MUST NOT BE MODIFIED BY THE CALLING PROGAM BETWEEN CALLS TO             250
C         ELEMENT.  ALL OTHER WORDS IN  COMMON/SPACE/  AND ALL WORDS IN           251
C         COMMON/TEMP/  MAY BE FREELY CHANGED BETWEEN CALLS TO  ELEMENT.          252
C                                                                                 253
C      THE VARIABLE  FIRST  IS INITIALLY .TRUE.  BUT IS SET TO .FALSE.            254
C         ON THE FIRST PASS THROUGH ELEMENT.                                      255
C                                                                                 256
C*****************************************************************************  257
C                                                                                 258
      LOGICAL FIRST,PRFLAG                                                        259
      COMMON/SPACE/SPACE(29),PRFLAG,OTHERS(1)                                     260
      COMMON/TEMP/TEMP(61)                                                        261
C                                                                                 262
      DATA FIRST/.T./                                                             263
C                                                                                 264
C                                                                                 265
      TIME = TIMING(DUMMY)                                                        266
      IF (FIRST)                                  GO TO 1                         267
                                                  GO TO 3                         268
C     THEN                                                                        269
C        . CALL THE PROGRAM CALLED  SETUP  WHICH READS IN THE REQUIRED            270
C           ARRAYS FROM TAPE1 AND STORES THEM ON DISK FOR FUTURE RECALL           271
C           BY THE PROGRAMS  QUAD  AND  TRI.                                      272
 1       CALL OVERLAY(4HMAIN,1,0)                                                 273
         FIRST = .F.                                                              274
         TIME = TIMING(DUMMY)                                                     275
         WRITE(6,2) TIME                                                          276
 2          FORMAT(////////* THE SETUP TIME WAS*,F7.3,* SECONDS.*)                277
C     CONTINUE                                                                    278
C                                                                                 279
C     . CALL THE SUBROUTINE INTGRAL WHICH EVALUATES THE A-, B- AND                280
C        C-INTEGRALS AND STORES THEM IN POSITIONS  IXC+1  THRU  IXA+IA            281
C        OF COMMON.                                                               282
 3    CALL INTGRAL                                                               283
C                                                                                 284
C     . CALL THE PROGRAM  SGPM  WHICH COMPUTES THE STIFFNESS MATRIX  SS,          285
C        THE GEOMETRIC STIFFNESS MATRIX  SG  IF  SGFLAG=.TRUE.,                   286
C        THE LOADVECTOR  SP  IF  SPFLAG=.TRUE.,                                   287
C        AND THE CONSISTENT MASS MATRIX  SM  IF  SMFLAG=.TRUE.                    288
      CALL OVERLAY(4HMAIN,12B,0,6HRECALL)                                         289
      TIME = TIMING(DUMMY)                                                        290
```

```
C                                                                       291
C     . STORE THE RESULTS ON DISK FOR FUTURE ASSEMBLY OF THE SYSTEM     292
C       EQUATIONS.                                                      293
      CALL STORE                                                        294
C                                                                       295
      IF (PRFLAG)                                                       296
C     THEN                                                              297
C       . CALL THE PROGRAM  OUTPUT  WHICH DISPLAYS THE RESULTS.         298
     *    CALL OVERLAY(4HMAIN,13B,0,6HRECALL)                          299
C     CONTINUE                                                          300
C                                                                       301
      WRITE(6,4) TIME                                                   302
 4      FORMAT(////////* THE COMPUTATION TIME FOR THIS ELEMENT WAS*     303
     *       ,F6.3,* SECONDS.*)                                         304
      RETURN                                                            305
      END                                                               306
```

APPENDIX A

```
      SUBROUTINE INTGRAL                                                    307
C*************************************************************************  308
C                                                                           309
C     THIS SUBROUTINE CALLS VARIOUS SUBPROGRAMS TO COMPUTE THE A-, B-       310
C        AND C-INTEGRALS NEEDED IN THE SUBROUTINES LINSTF,LODVEC AND        311
C        AND MASS.                                                          312
C                                                                           313
C     PARA  IS SET TO  .TRUE.  BY THE SUBROUTINE  QUAD  WHEN THE ELEMENT    314
C        IS DEEMED TO BE A PARALLELOGRAM.                                   315
C     TRAP  IS SET TO  .TRUE.  BY THE SUBROUTINE  QUAD  WHEN THE ELEMENT    316
C        IS DEEMED TO BE A TRAPEZOID.                                       317
C     THE VARIABLE  LIMIT  SPECIFIES THE UPPER LIMIT TO THE RANGE OF        318
C        C-INTEGRALS TO BE EVALUATED WITHIN A GIVEN OVERLAY CALL.           319
C                                                                           320
C*************************************************************************  321
C                                                                           322
      LOGICAL PARA,TRAP                                                     323
      COMMON/SPACE/SPACE(24),NSF,SKIP(75),SKP(7),LIMIT,SKIPP(27),PARA,      324
     *    TRAP,OTHERS(1)                                                    325
C                                                                           326
C                                                                           327
C                                                                           328
      IF (NSF.EQ.6 .OR. NSF.EQ.10)                  GO TO 2                 329
                                                    GO TO 3                 330
C     THEN  TRIANGULAR CASE                                                 331
C        . CALL THE PROGRAM  TRI  WHICH EVALUATES THE A-, B- AND C-TYPE     332
C           INTEGRALS FOR TRIANGULAR ELEMENTS.                              333
 2       CALL OVERLAY(4HMAIN,2,0,6HRECALL)                                  334
                                                    GO TO 20                335
C     ELSE                                                                  336
 3    IF (NSF.EQ.4 .OR. NSF.EQ.5 .OR. NSF.EQ.8 .OR. NSF.EQ.9) GO TO 4       337
                                                    GO TO 18                338
C     THEN  QUADRILATERAL CASE                                              339
C        . CALL THE PROGRAM  QUAD  WHICH EVALUATES THE A-, B- AND C-TYPE    340
C           INTEGRALS FOR PARALLELOGRAM ELEMENTS AND THE A- AND B-TYPE      341
C           INTEGRALS FOR THE NONPARALLELOGRAM QUADRILATERAL CASE.          342
 4       CALL OVERLAY(4HMAIN,3,0,6HRECALL)                                  343
         IF (.NOT.PARA)                             GO TO 5                 344
                                                    GO TO 20                345
C        THEN THE ELEMENT IS NOT A PARALLELOGRAM.                           346
 5          IF (TRAP)                              GO TO 6                 347
                                                    GO TO 13                348
C           THEN THE ELEMENT IS A TRAPEZOID.                                349
 6             IF (NSF.EQ.4)                       GO TO 7                 350
                                                    GO TO 8                 351
C              THEN                                                         352
 7                LIMIT = 10                                                353
                  CALL OVERLAY(4HMAIN,4,0)                                  354
                                                    GO TO 20                355
C              ELSE                                                         356
 8             IF (NSF.EQ.5)                       GO TO 9                 357
                                                    GO TO 10                358
C              THEN                                                         359
 9                LIMIT = 15                                                360
                  CALL OVERLAY(4HMAIN,4,0)                                  361
                                                    GO TO 20                362
C              ELSE                                                         363
 10            IF (NSF.EQ.8)                       GO TO 11                364
                                                    GO TO 12                365
C              THEN                                                         366
 11               LIMIT = 36                                                367
                  CALL OVERLAY(4HMAIN,5,0)                                  368
                                                    GO TO 20                369
C              ELSE  NSF=9                                                  370
 12               LIMIT = 45                                                371
                  CALL OVERLAY(4HMAIN,5,0)                                  372
                                                    GO TO 20                373
C           CONTINUE                                                        374
```

```
C            ELSE  NON-TRAPEZOIDAL CASE                                          375
 13             IF (NSF.EQ.4 .OR. NSF.EQ.5)        GO TO 14                     376
                                                   GO TO 16                     377
C               THEN                                                            378
C                  . LIMIT IS 10 IF NSF.EQ.4 BUT LIMIT IS 15 IF NSF.EQ.5        379
 14                LIMIT = 10                                                    380
                   IF (NSF.EQ.5)                                                381
C                  THEN                                                         382
     *                LIMIT = 15                                                383
C                  CONTINUE                                                     384
                   CALL OVERLAY(4HMAIN,6,0)                                     385
                                                   GO TO 20                     386
C                  CONTINUE                                                     387
C               ELSE  NSF=8 OR NSF=9                                            388
 16                CALL OVERLAY(4HMAIN,7,0)                                     389
                   CALL OVERLAY(4HMAIN,10B,0)                                   390
                   IF (NSF.EQ.9)                   GO TO 17                     391
                                                   GO TO 20                     392
C                  THEN                                                         393
 17                   CALL OVERLAY(4HMAIN,11B,0)                               394
                                                   GO TO 20                     395
C                  CONTINUE                                                     396
C               CONTINUE                                                        397
C            CONTINUE                                                           398
C         CONTINUE                                                              399
C      ELSE  ERROR                                                              400
 18       WRITE (6,19) NSF                                                      401
 19          FORMAT(* NSF =*,I5,* IS ILLEGAL.*)                                 402
          STOP                                                                  403
 20    CONTINUE                                                                 404
C      . AT THIS POINT ALL INTEGRALS HAVE BEEN COMPUTED.                        405
       RETURN                                                                   406
       END                                                                      407
```

APPENDIX A

```
      SUBROUTINE STORE                                                  408
C                                                                       409
      LOGICAL SGFLAG,SMFLAG,SPFLAG                                      410
      COMMON/TEMP/I,IL,IU                                               411
      COMMON/SPACE/SPACE(24),NSF,SKIP,SGFLAG,SMFLAG,SPFLAG,SKP(71),     412
     *    SKIPP(14),NNE,ISS,ISG,IXC,OTHERS(1)                          413
C                                                                       414
C                                                                       415
C     . STORE  SS  ON UNIT 3.                                           416
      IL = ISS + 1                                                      417
      WRITE (3) (SPACE(I),I=IL,ISG)                                     418
C                                                                       419
      IL = ISG + 1                                                      420
      IF (SGFLAG)                                                       421
C     THEN                                                              422
C     . STORE  SG  ON UNIT 4.                                           423
     *    WRITE (4) (SPACE(I),I=IL,IXC)                                424
C     CONTINUE                                                          425
C                                                                       426
      IL = IXC + 1                                                      427
      IU = IXC + 5*NNE                                                  428
      IF (SPFLAG)                                                       429
C     THEN                                                              430
C     . STORE  SP  ON UNIT 8.                                           431
     *    WRITE (8) (SPACE(I),I=IL,IU)                                 432
C     CONTINUE                                                          433
C                                                                       434
      IL = IU + 1                                                       435
      IU = IL + NSF*NSF                                                 436
      IF (SMFLAG)                                                       437
C     THEN                                                              438
C     . STORE  SM  ON UNIT 9.                                           439
     *    WRITE (9) (SPACE(I),I=IL,IU)                                 440
C     CONTINUE                                                          441
      RETURN                                                            442
      END                                                               443
```

```
      OVERLAY(MAIN,1,0)                                                    444
      PROGRAM SETUP                                                        445
C*************************************************************************446
C                                                                          447
C     THE PURPOSE OF THIS SUBROUTINE IS TO SET UP THE PARAMETERS AND       448
C        ARRAYS REQUIRED BY THE SUBROUTINES WHICH EVALUATE AND STORE       449
C        INTEGRALS.                                                        450
C     ALL READ STATEMENTS IN THIS PROGRAM REFER TO  TAPE1.                 451
C     THE WRITE STATEMENTS IN THIS PROGRAM REFER TO THE OUTPUT FILE OR     452
C        TO SCRATCH DISC (UNIT 2).                                         453
C                                                                          454
C     THE LENGTH OF COMMON MUST BE THE MAXIMUM OF THE VALUES OF *LENGTH*   455
C        FOR THE VALUES OF NSF EMPLOYED.                                   456
C     NNE  IS THE NUMBER OF TRUE NODES PER ELEMENT.                        457
C     IA  IS THE NUMBER OF DISTINCT A-INTEGRALS.                           458
C        IA = (NSF+1)*(NSF+2)*(NSF+3)*(NSF+4)/24                           459
C     2*IB  IS THE NUMBER OF DISTINCT B-INTEGRALS.                         460
C        IB = NSF*(NSF+1)*(NSF+2)/2                                        461
C     4*IC  IS THE NUMBER OF COMPUTED C-INTEGRALS.                         462
C        IC = NSF*(NSF+1)/2                                                463
C     JA  IS THE NUMBER OF REPRESENTATIVE A-INTEGRALS.                     464
C     JB  IS THE NUMBER OF REPRESENTATIVE B-INTEGRALS.                     465
C     JC  IS THE NUMBER OF REPRESENTATIVE C-INTEGRALS.                     466
C     THE ARRAY  SS  IS STORED BEGINNING IN ISS+1.                         467
C     THE ARRAY  SG  IS STORED BEGINNING IN ISG+1.                         468
C     THE C-INTEGRALS AND LATER  SP  ARE STORED BEGINNING IN IXC+1.        469
C     THE ARRAY  SM  IS STORED BEGINNING IN IXC+5*NNE+1.                   470
C     THE B-INTEGRALS ARE STORED BEGINNING IN IXB+1.                       471
C     THE A-INTEGRALS ARE STORED BEGINNING IN  IXA+1.                      472
C                                                                          473
C*************************************************************************474
C                                                                          475
      LOGICAL TRI                                                          476
      DIMENSION INDICES(8,7),NCARDS(7),INDX(12)                            477
      COMMON/SPACE/SPACE(100),NRECORD(7),SKIP,IA,IB,IC,JA,JB,JC,NNE,ISS,   478
     *   ISG,IXC,IXB,IXA,QC(1)                                             479
      COMMON/TEMP/  IL,IU,   I2,I3,I4,  JL,JU,LENGTH,LEVEN,LODD,  N,       480
     *   NSFM,TRI                                                          481
      COMMON/STORE/IX(31)                                                  482
      EQUIVALENCE (IA,INDX(1))                                             483
C                                                                          484
      DATA (NCARDS(I),I=1,7)/42,82,119,1,204,305,427/                      485
      DATA (INDICES(I),I=1,56)/                                            486
     *    70,   60,  10,   17,   11,   3,   4,  137,                       487
     *   126,  105,  15,   34,   24,   5,   4,  137,                       488
     *   210,  168,  21,   51,   36,   6,   6,  120,                       489
     *     0,    0,   0,    0,    0,   0,   0,    0,                       490
     *   495,  360,  36,   84,   54,   8,   8,  137,                       491
     *   715,  495,  45,  130,   83,  11,   8,  137,                       492
     *  1001,  660,  55,  195,  121,  14,  10,  120/                       493
C                                                                          494
C                                                                          495
      NRECORD(4) = 0                                                       496
      CALL OPENMS(2,IX(1),31,0)                                            497
      N = 0                                                                498
      DO 8 M=1,7                                                           499
         NSFM = M + 3                                                      500
         IF (NRECORD(M).EQ.0)                        GO TO 1               501
                                                     GO TO 4               502
C        THEN  THE INTEGRATION ARRAYS FOR NSF EQUAL TO NSFM ARE NOT TO     503
C              BE SET UP.  NCARDS(M) IS THE NUMBER OF INPUT CARDS TO BE    504
C              SKIPPED OVER.                                               505
 1          IU = NCARDS(M)                                                 506
            DO 3 I=1,IU                                                    507
               READ(1,2)                                                   508
 2                FORMAT(I1)                                              509
 3          CONTINUE                                                       510
                                                     GO TO 7               511
```

```
C        ELSE  THE INTEGRATION ARRAYS FOR THIS  VALUE OF NSFM ARE TO BE          512
C              SET UP.                                                          513
 4          N = N + 1                                                          514
C           . THE (M)TH POSITION OF NRECORD IS TO BE REPLACED BY AN           515
C              INTEGER WHICH INDICATES WHERE ON UNIT (2) THE INTEGRATION       516
C              ARRAYS FOR  NSF=M+3  ARE TO BE STORED.  SEE SUBROUTINES         517
C              TRI AND QUAD.                                                   518
            NRECORD(M) = N                                                     519
            TRI = NSFM.EQ.6 .OR. NSFM.EQ.10                                    520
            DO 5 I=1,8                                                         521
               INDX(I) = INDICES(I,M)                                          522
 5          CONTINUE                                                           523
            ISG = ISS + 25*NSFM*NSFM                                           524
            IXC = ISG + NSFM*NSFM                                              525
            IXB = IXC + 4*IC                                                   526
            IXA = IXB + 2*IB                                                   527
            LENGTH = IXA + IA                                                  528
            CALL WRITMS(2,INDX(1),12,5*N-4)                                    529
            WRITE(6,6) NSFM,LENGTH                                             530
 6             FORMAT(*1LENGTH OF COMMON/SPACE/ REQUIRED FOR NSF = *,I5,       531
     *             * IS *,I6,*.*/)                                             532
C                                                                              533
C           . SET UP THE INTEGRATION ARRAYS FOR THE A-INTEGRALS .              534
            CALL SETA                                                          535
C                                                                              536
C           . SET UP THE INTEGRATION ARRAYS FOR THE B-INTEGRALS .              537
            CALL SETB                                                          538
C                                                                              539
C           . SET UP THE INTEGRATION ARRAYS FOR THE C-INTEGRALS .              540
            CALL SETC                                                          541
C                                                                              542
 7       CONTINUE                                                              543
 8    CONTINUE                                                                 544
      WRITE(6,9)                                                               545
 9       FORMAT(1H1)                                                           546
      END                                                                      547
```

APPENDIX A

```
      SUBROUTINE SETA                                                    548
C*********************************************************************  549
C                                                                        550
C     THIS SUBROUTINE SETS UP THE INTEGRATION ARRAYS FOR EVALUATING      551
C        A-INTEGRALS.                                                    552
C                                                                        553
C     THE NUMBERS  KA  SELECT  THE REPRESENTATIVE INTEGRALS.             554
C     THE NUMBERS  LA  SELECT THE GROUP TRANSFORMATIONS.                 555
C                                                                        556
C*********************************************************************  557
C                                                                        558
      LOGICAL TRI                                                        559
      DIMENSION KA(1),LA(1),QA(4,1),QA1(1),QA2(1),QA3(1)                 560
      COMMON/SPACE/SPACE(108),IA,SKIP(2),JA,SKP(7),IXA,QQ(1)             561
      COMMON/TEMP/  IL,IU,   I2,I3,I4,  JL,JU,LENGTH,LEVEN,LODD,  N,     562
     *   NSFM,TRI,K,L,Q1,Q2                                              563
      EQUIVALENCE (KA(1),SPACE(1)),(LA(1),SPACE(1)),(QA(1,1),QQ(1)),     564
     *   (QA1(1),SPACE(1)),(QA2(1),SPACE(1)),(QA3(1),SPACE(1))           565
C                                                                        566
C                                                                        567
      IL = IXA + 1                                                       568
      IU = IXA + IA                                                      569
      READ(1,1) (KA(I),I=IL,IU)                                          570
 1       FORMAT(20I4)                                                    571
      WRITE(6,1) (KA(I),I=IL,IU)                                         572
      IF (TRI)                                  GO TO 2                  573
                                                GO TO 7                  574
C     THEN  TRIANGULAR CASE                                              575
 2       READ(1,3) ((QA(I,J),I=1,2),J=1,JA)                              576
 3          FORMAT(10X,2F10.0)                                           577
         DO 4 J=1,JA                                                     578
            QA(1,J) = QA(1,J)/QA(2,J)                                    579
 4       CONTINUE                                                        580
         WRITE(6,5) (QA(1,J),J=1,JA)                                     581
 5          FORMAT(5X,12E10.3)                                           582
         DO 6 I=IL,IU                                                    583
            K = KA(I)                                                    584
            QA1(I) = QA(1,K)                                             585
 6       CONTINUE                                                        586
         CALL WRITMS(2,QA1(IL),IU-IL+1,5*N-3)                            587
                                                GO TO 12                 588
C     ELSE  QUADRILATERAL CASE                                           589
 7       JL = IL - IA                                                    590
         JU = IU - IA                                                    591
         READ(1,8) (LA(I),I=JL,JU)                                       592
 8          FORMAT(80I1)                                                 593
         WRITE(6,1) (LA(I),I=JL,JU)                                      594
         READ(1,9) ((QA(I,J),I=1,4),J=1,JA)                              595
 9          FORMAT(10X,4F10.0)                                           596
         DO 10 I=1,3                                                     597
            DO 10 J=1,JA                                                 598
               QA(I,J) = QA(I,J)/QA(4,J)                                 599
C           CONTINUE                                                     600
 10      CONTINUE                                                        601
         WRITE(6,5) ((QA(I,J),J=1,JA),I=1,3)                             602
         DO 11 I1=IL,IU                                                  603
            I2 = I1 - IA                                                 604
            K = KA(I1)                                                   605
            L = LA(I2)                                                   606
            LODD = L - (L/2)*2                                           607
            LEVEN = 1 - LODD                                             608
            Q1 = QA(1,K)                                                 609
            Q2 = QA(2,K)                                                 610
            QA1(I1) = QA(3,K)                                            611
            QA2(I2) = -(-1)**(L/2)*(LODD*Q2+LEVEN*Q1)                    612
            QA3(I2-IA) = -(-1)**((L+1)/4)*(LODD*Q1+LEVEN*Q2)             613
 11      CONTINUE                                                        614
```

```
          IL = JL - IA                                                    615
C         . STORE QA3, QA2 AND QA1  ON DISC .                             616
          CALL WRITMS(2,SPACE(IL),IU-IL+1,5*N-3)                          617
 12     CONTINUE                                                         618
        RETURN                                                           619
        END                                                              620
```

```
      SUBROUTINE SETB                                                    621
C*************************************************************************622
C                                                                        623
C     THIS SUBROUTINE SETS UP THE INTEGRATION ARRAYS FOR EVALUATING      624
C        B-INTEGRALS.                                                    625
C                                                                        626
C     THE NUMBERS  KB  SELECT  THE REPRESENTATIVE INTEGRALS.             627
C     THE NUMBERS  LB  SELECT THE GROUP TRANSFORMATIONS.                 628
C                                                                        629
C*************************************************************************630
C                                                                        631
      LOGICAL TRI                                                        632
      DIMENSION KB(1),LB(1),QB(4,1),QB1(1),QB2(1),QB3(1)                 633
      COMMON/SPACE/SPACE(109),IB,SKIP(2),JB,SKP(5),IXB,SKIPP,QQ(1)       634
      COMMON/TEMP/  IL,IU,   I2,I3,I4,  JL,JU,LENGTH,LEVEN,LODD,  N,     635
     *   NSFM,TRI,K,L,Q2,Q3                                              636
      EQUIVALENCE (KB(1),SPACE(1)),(LB(1),SPACE(1)),(QB(1,1),QQ(1)),     637
     *   (QB1(1),SPACE(1)),(QB2(1),SPACE(1)),(QB3(1),SPACE(1))           638
C                                                                        639
C                                                                        640
      IL = IXB + 1                                                       641
      IU = IXB + IB                                                      642
      READ(1,1) (KB(I),I=IL,IU)                                          643
 1       FORMAT(20I4)                                                    644
      JL = IL + IB                                                       645
      JU = IU + IB                                                       646
      READ(1,2) (LB(I),I=JL,JU)                                          647
 2       FORMAT(80I1)                                                    648
      WRITE(6,1) (KB(I),I=IL,IU),(LB(I),I=JL,JU)                         649
      IF (TRI)                                          GO TO 3          650
                                                        GO TO 13         651
C     THEN   TRIANGULAR CASE                                             652
 3       READ(1,4) ((QB(I,J),I=1,3),J=1,JB)                              653
 4          FORMAT(10X,3F10.0)                                           654
           DO 5 I=1,2                                                   655
              DO 5 J=1,JB                                               656
                 QB(I,J) = QB(I,J)/QB(3,J)                              657
C             CONTINUE                                                   658
 5        CONTINUE                                                       659
          DO 12 I1=JL,JU                                                 660
             I2 = I1 - IB                                                661
             K = KB(I2)                                                  662
             L = LB(I1)                                                  663
             GO TO (6,7,8,9,10,11),L                                     664
 6              QB1(I1) =  QB(1,K)                                       665
                QB2(I2) =  QB(2,K)                                       666
                   GO TO 12                                              667
 7              QB1(I1) =  QB(2,K)                                       668
                QB2(I2) = -QB(1,K) - QB(2,K)                             669
                   GO TO 12                                              670
 8              QB1(I1) = -QB(1,K) - QB(2,K)                             671
                QB2(I2) =  QB(1,K)                                       672
                   GO TO 12                                              673
 9              QB1(I1) = -QB(2,K)                                       674
                QB2(I2) = -QB(1,K)                                       675
                   GO TO 12                                              676
 10             QB1(I1) =  QB(1,K) + QB(2,K)                             677
                QB2(I2) = -QB(2,K)                                       678
                   GO TO 12                                              679
 11             QB1(I1) = -QB(1,K)                                       680
                QB2(I2) =  QB(1,K) + QB(2,K)                             681
C            . END COMPUTED GO TO .                                      682
 12       CONTINUE                                                       683
C         . STORE  QB2  AND  QB1 .                                       684
          CALL WRITMS(2,SPACE(IL),IU-IL+1,5*N-2)                         685
                                                        GO TO 18         686
C     ELSE  QUADRILATERAL CASE                                           687
 13       READ(1,14) ((QB(I,J),I=1,4),J=1,JB)                            688
 14          FORMAT(10X,4F10.0)                                          689
```

```
      DO 15 I=1,3                                                        690
         DO 15 J=1,JB                                                    691
            QB(I,J) = QB(I,J)/QB(4,J)                                    692
C        CONTINUE                                                        693
 15    CONTINUE                                                          694
       WRITE(6,16) ((QB(I,J),J=1,JB),I=1,3)                              695
 16       FORMAT(5X,12E10.3)                                             696
       DO 17 I1=JL,JU                                                    697
          I2 = I1 - IB                                                   698
          K = KB(I2)                                                     699
          L = LB(I1)                                                     700
          LODD = L - (L/2)*2                                             701
          LEVEN = 1 - LODD                                               702
          QB1(I1) = (LODD-LEVEN)*CB(1,K)                                 703
          Q2 = QB(2,K)                                                   704
          Q3 = QB(3,K)                                                   705
          QB2(I2) = (-1)**(L/2)*(LODD*Q2-LEVEN*Q3)                       706
          QB3(I2-IB) = (-1)**((L+1)/4)*(LODD*Q3-LEVEN*Q2)                707
 17    CONTINUE                                                          708
       IL = IL - IB                                                      709
C      . STORE  QB3, QB2  AND  QB1 .                                     710
       CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N-2)                            711
 18  CONTINUE                                                            712
     RETURN                                                              713
     END                                                                 714
```

```
      SUBROUTINE SETC                                                    715
C*********************************************************************  716
C                                                                        717
C     THIS SUBROUTINE SETS UP THE INTEGRATION ARRAYS FOR EVALUATING      718
C        C-INTEGRALS.                                                    719
C                                                                        720
C     THE NUMBERS  KC  SELECT  THE REPRESENTATIVE INTEGRALS.             721
C     THE NUMBERS  LC  SELECT THE GROUP TRANSFORMATIONS.                 722
C                                                                        723
C*********************************************************************  724
C                                                                        725
      LOGICAL TRI                                                        726
      DIMENSION KC(1),LC(1),QC(5,1),QC1(1),QC2(1),QC3(1),QC4(1)          727
      COMMON/SPACE/SPACE(110),IC,SKIP(2),JC,SKP(3),IXC,SKIPP(2),QQ(1)    728
      COMMON/TEMP/  IL,IU,   I2,I3,I4,  JL,JU,LENGTH,LEVEN,LODD,  N,     729
     *   NSFM,TRI,K,L,Q1,Q2,Q3,Q4                                        730
      EQUIVALENCE (KC(1),SPACE(1)),(LC(1),SPACE(1)),(QC(1,1),QQ(1)),     731
     *   (QC1(1),SPACE(1)),(QC2(1),SPACE(1)),(QC3(1),SPACE(1)),          732
     *   (QC4(1),SPACE(1))                                               733
C                                                                        734
C                                                                        735
      IL = IXC + 1                                                       736
      IU = IXC + IC                                                      737
      READ(1,1) (KC(I),I=IL,IU)                                          738
 1       FORMAT(20I4)                                                    739
      JL = IL + IC                                                       740
      JU = IU + IC                                                       741
      READ(1,2) (LC(I),I=JL,JU)                                          742
 2       FORMAT(80I1)                                                    743
      WRITE(6,1) (KC(I),I=IL,IU),(LC(I),I=JL,JU)                         744
      READ(1,3) ((QC(I,J),I=1,5),J=1,JC)                                 745
 3       FORMAT(10X,5F10.0)                                              746
      DO 4 I=1,4                                                         747
         DO 4 J=1,JC                                                     748
            QC(I,J) = QC(I,J)/QC(5,J)                                    749
C        CONTINUE                                                        750
 4    CONTINUE                                                           751
      WRITE(6,5) ((QC(I,J),J=1,JC),I=1,4)                                752
 5       FORMAT(5X,12E10.3)                                              753
      IF (TRI)                                    GO TO 6                754
                                                  GO TO 15               755
C     THEN  TRIANGULAR CASE                                              756
 6       JL = JL + 2*IC                                                  757
         JU = JU + 2*IC                                                  758
         DO 14 I1=JL,JU                                                  759
            I2 = I1 - IC                                                 760
            I3 = I2 - IC                                                 761
            I4 = I3 - IC                                                 762
            K  = KC(I4)                                                  763
            L  = LC(I3)                                                  764
            Q1 = QC(1,K)                                                 765
            Q2 = QC(2,K)                                                 766
            Q3 = QC(3,K)                                                 767
            Q4 = QC(4,K)                                                 768
            GO TO (7,8,9,10,11,12),L                                     769
 7             QC1(I1) =  Q1                                             770
               QC2(I2) =  Q2                                             771
               QC3(I3) =  Q3                                             772
               QC4(I4) =  Q4                                             773
                  GO TO 13                                               774
 8             QC1(I1) =  Q4                                             775
               QC2(I2) = -(Q3+Q4)                                        776
               QC3(I3) = -(Q2+Q4)                                        777
               QC4(I4) =  Q1+Q2+Q3+Q4                                    778
                  GO TO 13                                               779
 9             QC1(I1) =  Q1+Q2+Q3+Q4                                    780
               QC2(I2) = -(Q1+Q3)                                        781
               QC3(I3) = -(Q1+Q2)                                        782
               QC4(I4) =  Q1                                             783
                  GO TO 13                                               784
```

```
 10            QC1(I1) =  Q4                                          785
               QC2(I2) =  Q3                                          786
               QC3(I3) =  Q2                                          787
               QC4(I4) =  Q1                                          788
                  GO TO 13                                            789
 11            QC1(I1) =  Q1+Q2+Q3+Q4                                 790
               QC2(I2) = -(Q2+Q4)                                     791
               QC3(I3) = -(Q3+Q4)                                     792
               QC4(I4) =  Q4                                          793
                  GO TO 13                                            794
 12            QC1(I1) =  Q1                                          795
               QC2(I2) = -(Q1+Q2)                                     796
               QC3(I3) = -(Q1+Q3)                                     797
               QC4(I4) =  Q1+Q2+Q3+Q4                                 798
C           . END COMPUTED GO TO .                                    799
 13         CONTINUE                                                  800
 14      CONTINUE                                                     801
C        . STORE QC4, QC3, QC2 AND QC1 .                              802
        CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N-1)                        803
                                                  GO TO 22            804
C     THEN  QUADRILATERAL CASE                                        805
C        . STORE  KC  AND  LC .                                       806
 15     CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N-1)                        807
        JL = JL + 2*IC                                                808
        JU = JU + 2*IC                                                809
        DO 21 I1=JL,JU                                                810
           I2 = I1 - IC                                               811
           I3 = I2 - IC                                               812
           I4 = I3 - IC                                               813
           K  = KC(I4)                                                814
           L  = LC(I3)                                                815
           GO TO (16,17,16,17,18,19,18,19),L                          816
 16           QC1(I1) =  QC(1,K)                                      817
              QC2(I2) =  QC(2,K)                                      818
              QC3(I3) =  QC(3,K)                                      819
              QC4(I4) =  QC(4,K)                                      820
                 GO TO 20                                             821
 17           QC1(I1) =  QC(4,K)                                      822
              QC2(I2) = -QC(3,K)                                      823
              QC3(I3) = -QC(2,K)                                      824
              QC4(I4) =  QC(1,K)                                      825
                 GO TO 20                                             826
 18           QC1(I1) =  QC(1,K)                                      827
              QC2(I2) = -QC(2,K)                                      828
              QC3(I3) = -QC(3,K)                                      829
              QC4(I4) =  QC(4,K)                                      830
                 GO TO 20                                             831
 19           QC1(I1) =  QC(4,K)                                      832
              QC2(I2) =  QC(3,K)                                      833
              QC3(I3) =  QC(2,K)                                      834
              QC4(I4) =  QC(1,K)                                      835
C          . END COMPUTED GO TO .                                     836
 20        CONTINUE                                                  837
 21     CONTINUE                                                     838
C       . STORE QC4, QC3, QC2 AND QC1 .                              839
        CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N)                          840
 22  CONTINUE                                                         841
     RETURN                                                           842
     END                                                              843
```

```
      OVERLAY(MAIN,2,0)                                                    844
      PROGRAM TRI                                                          845
C*************************************************************************846
C                                                                          847
C     THIS SUBROUTINE NUMERICALLY EVALUATES FOR TRIANGULAR FINITE          848
C        ELEMENTS ALL THREE TYPES OF INTEGRALS AND STORES THEM IN COMMON    849
C        WITH ALIASES XA,XB,XC.                                            850
C                                                                          851
C     THE NODES OF THE SIX- AND TEN-NODE TRIANGULAR FINITE ELEMENTS        852
C        ARE NUMBERED AS FOLLOWS.                                          853
C                                                                          854
C                                              3                           855
C                    3                                                     856
C                                          6       8                       857
C               6        5                                                 858
C                                       9      10       5                  859
C          1        4        2                                             860
C                                  1       4       7       2               861
C                                                                          862
C     EACH TIME THIS PROGRAM IS CALLED IT READS SEVEN RECORDS FROM         863
C        SCRATCH DISC (UNIT 2).  THESE RECORDS WERE WRITTEN BY THE         864
C        PROGRAM SETUP.                                                    865
C                                                                          866
C*************************************************************************867
C                                                                          868
C                                                                          869
      DIMENSION XA(1),XB(1),XC(1),QA1(1),QB1(1),QB2(1),                    870
     *   QC1(1),QC2(1),QC3(1),QC4(1),INDX(12)                              871
      COMMON/SPACE/SPACE(24),NSF,SKIP(7),X1,X2,X3,X4,Y1,Y2,Y3,Y4,ZZ(60),   872
     *   NRECORD(7),SKIPP,IA,IB,IC,JA,JB,JC,NNE,ISS,ISG,IXC,IXB,IXA,       873
     *   OTHERS(1)                                                         874
      COMMON/TEMP/AREA,AREAINV,CA,CB,CC,CD,C1,C2,IL,IU,I2,I3,I4,           875
     *   K,L,NREC,TEMP,X12,X23,X31,Y12,Y23,Y31                             876
      EQUIVALENCE (XA(1),SPACE(1)),(XB(1),SPACE(1)),(XC(1),SPACE(1)),      877
     *   (QA1(1),SPACE(1)),(QB1(1),SPACE(1)),(QB2(1),SPACE(1)),            878
     *   (QC1(1),SPACE(1)),(QC2(1),SPACE(1)),(QC3(1),SPACE(1)),            879
     *   (QC4(1),SPACE(1)),(INDX(1),IA)                                    880
C                                                                          881
C                                                                          882
      AREA = ((X1-X2)*(Y1-Y3) - (X1-X3)*(Y1-Y2))/2.                        883
      AREAINV = 1./AREA                                                    884
      X12 = X1 - X2                                                        885
      X23 = X2 - X3                                                        886
      X31 = X3 - X1                                                        887
      Y12 = Y1 - Y2                                                        888
      Y23 = Y2 - Y3                                                        889
      Y31 = Y3 - Y1                                                        890
      NREC = 5*NRECORD(NSF-3) - 4                                          891
      IF (NREC.LT.0)                                                       892
C  ...THEN ERROR TERMINATION                                               893
     *   STOP 234                                                          894
C     CONTINUE                                                             895
      CALL READMS(2,INDX(1),12,NREC)                                       896
      IL = IXA + 1                                                         897
      IU = IXA + IA                                                        898
      CALL READMS(2,QA1(IL),IU-IL+1,NREC+1)                                899
      DO 2 I=IL,IU                                                         900
         XA(I) = QA1(I)*AREA                                               901
 2    CONTINUE                                                             902
C     . EVALUATION OF XA IS NOW COMPLETE.                                  903
C                                                                          904
C                                                                          905
      IL = IXB + 1                                                         906
      IU = IXB + 2*IB                                                      907
C     . READ IN  QB2  AND  QB1 .                                           908
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+2)                              909
      IU = IXB + IB                                                        910
```

```
      DO 3 I1=IL,IU                                                    911
         I2 = I1 + IB                                                  912
         CA = QB1(I2)                                                  913
         CB = QB2(I1)                                                  914
         XB(I1) = CB*Y23 - CA*Y31                                      915
         XB(I2) = CA*X31 - CB*X23                                      916
 3    CONTINUE                                                         917
C     . EVALUATION OF XB IS NOW COMPLETE.                              918
C                                                                      919
      IL = IXC + 1                                                     920
      IU = IXC + 4*IC                                                  921
C     . READ IN  QC4, QC3, QC2  AND  QC1 .                             922
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+3)                          923
      IU = IXC + IC                                                    924
      DO 4 I1=IL,IU                                                    925
         I2 = I1 + IC                                                  926
         I3 = I2 + IC                                                  927
         I4 = I3 + IC                                                  928
         CA = QC1(I4)                                                  929
         CB = QC2(I3)                                                  930
         CC = QC3(I2)                                                  931
         CD = QC4(I1)                                                  932
         C1 = CB*Y23 - CA*Y31                                          933
         C2 = CD*Y23 - CC*Y31                                          934
         XC(I1) = (C2*Y23 - C1*Y31)*AREAINV                            935
         XC(I2) = (C1*X31 - C2*X23)*AREAINV                            936
         C1 = CB*X23 - CA*X31                                          937
         C2 = CD*X23 - CC*X31                                          938
         XC(I3) = (C1*Y31 - C2*Y23)*AREAINV                            939
         XC(I4) = (C2*X23 - C1*X31)*AREAINV                            940
 4    CONTINUE                                                         941
C     . EVALUATION OF XC IS NOW COMPLETE.                              942
      END                                                              943
```

APPENDIX A

```
      OVERLAY(MAIN,3,0)                                                    944
      PROGRAM QUAD                                                         945
C*************************************************************************946
C                                                                          947
C     FOR PARALLELOGRAM FINITE ELEMENTS THIS PROGRAM NUMERICALLY EVALU-    948
C        ATES ALL 3 TYPES OF INTEGRALS AND STORES THEM IN COMMON/SPACE/    949
C        WITH ALIASES  XA,XB,XC.  IT THEN SETS  PARA  TO  .TRUE.           950
C     FOR QUADRILATERAL ELEMENTS WHICH ARE NOT PARALLELOGRAMS ONLY  XA     951
C        AND  XB  ARE COMPUTED,  PARA  IS SET TO .FALSE.,  TRAP  IS SET   952
C        TO .TRUE. OR .FALSE. ACCORDING AS THE ELEMENT IS OR IS NOT A      953
C        TRAPEZOID, AND LOGARITHM TERMS ARE EVALUATED.                     954
C                                                                          955
C     THE NODES FOR THE FOUR-,FIVE-, EIGHT-, AND NINE-NODE QUADRILATERAL   956
C        FINITE ELEMENTS ARE LABELED AS FOLLOWS                            957
C                                                                          958
C                  4        3            4   7   3                       959
C                                                                          960
C                      5                 8   9   6                       961
C                                                                          962
C                  1        2            1   5   2                       963
C                                                                          964
C     THE SHAPE FUNCTION ASSOCIATED WITH THE BUBBLE MODE IS                965
C                                                                          966
C                              2       2                                   967
C        N(KSE,ETA) = (1-KSI )*(1-ETA )                                    968
C                                                                          969
C     EACH TIME THIS PROGRAM IS CALLED IT READS RECORDS FROM SCRATCH       970
C        DISC (UNIT 2).  THESE RECORDS WERE WRITTEN BY THE PROGRAM         971
C        CALLED SETUP.                                                     972
C                                                                          973
C*************************************************************************974
C                                                                          975
      LOGICAL PARA,TRAP,RRR,SSS                                            976
      DIMENSION XA(1),XB(1),XC(1),KC(1),LC(1),                             977
     *   QA1(1),QA2(1),QA3(1),QB1(1),QB2(1),QB3(1),                        978
     *   QC1(1),QC2(1),QC3(1),QC4(1),INDX(12)                              979
      COMMON/SPACE/SPACE(24),NSF,DUM(7),X(4),Y(4),DUMMY(60),               980
     *   NRECORD(7),SKIP,IA,IB,IC,JA,JB,JC,NNE,ISS,ISG,IXC,IXB,IXA,        981
     *   X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,ALG1,ALG2,ALG3,ULG1,ULG2,ULG3,          982
     *   PARA,TRAP,OTHERS(1)                                               983
      COMMON/TEMP/CA,CB,CC,CD,C1,C2,IL,IQA,IU,I2,I3,I4,K,L,                984
     *   NREC,TEMP,Z1INV,R,S,RR,SS,RRR,SSS,RD,SD,R2,S2,RR2,SS2             985
      EQUIVALENCE (XA(1),SPACE(1)),(XB(1),SPACE(1)),(XC(1),SPACE(1)),      986
     *   (KC(1),SPACE(1)),(LC(1),SPACE(1)),                                987
     *   (QA1(1),SPACE(1)),(QA2(1),SPACE(1)),(QA3(1),SPACE(1)),            988
     *   (QB1(1),SPACE(1)),(QB2(1),SPACE(1)),(QB3(1),SPACE(1)),            989
     *   (QC1(1),SPACE(1)),(QC2(1),SPACE(1)),(QC3(1),SPACE(1)),            990
     *   (QC4(1),SPACE(1)),(INDX(1),IA),(DLG,ALG1),(RS2,ALG2),(CLG,ALG3)   991
     *   ,(VLG1,ALG1),(VLG2,ALG2),(VLG3,ALG3)                              992
C                                                                          993
      DATA EPS/1.E-4/                                                      994
C                                                                          995
C                                                                          996
      X1 = (X(1)+X(3)-X(2)-X(4))/4.                                        997
      X2 = (X(2)+X(3)-X(4)-X(1))/4.                                        998
      X3 = (X(3)+X(4)-X(1)-X(2))/4.                                        999
      Y1 = (Y(1)+Y(3)-Y(2)-Y(4))/4.                                       1000
      Y2 = (Y(2)+Y(3)-Y(4)-Y(1))/4.                                       1001
      Y3 = (Y(3)+Y(4)-Y(1)-Y(2))/4.                                       1002
C     . Z1  IS ONE-FOURTH THE AREA OF THE QUADRILATERAL ELEMENT.           1003
      Z1 = X2*Y3 - X3*Y2                                                  1004
      Z2 = X3*Y1 - X1*Y3                                                  1005
      Z3 = X1*Y2 - X2*Y1                                                  1006
      Z1INV = 1./Z1                                                       1007
      R = Z2/Z1                                                           1008
      S = Z3/Z1                                                           1009
      R2 = R*R                                                            1010
      S2 = S*S                                                            1011
```

```
      RD = 1./(1.-R2)                                                    1012
      SD = 1./(1.-S2)                                                    1013
      RR = R*SD                                                          1014
      SS = S*RD                                                          1015
      RRR = ABS(RR).LT.EPS                                               1016
      SSS = ABS(SS).LT.EPS                                               1017
      PARA = RRR.AND.SSS                                                 1018
      TRAP = RRR.OR.SSS                                                  1019
      NREC = 5*NRECORD(NSF-3) - 4                                        1020
      IF (NREC.LT.0)                                                     1021
C  ...THEN ERROR TERMINATION                                             1022
     *    STOP 123                                                       1023
C     CONTINUE                                                          1024
      CALL READMS(2,INDX(1),12,NREC)                                     1025
      IL = IXA + 1 - 2*IA                                                1026
      IU = IXA + IA                                                      1027
C     . READ IN  QA3, QA2  AND  QA1 .                                    1028
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+1)                            1029
      IL = IXA + 1                                                       1030
      IF (PARA)                                    GO TO 1               1031
                                                   GO TO 3               1032
C     THEN  PARALLELOGRAM CASE                                           1033
 1       DO 2 I=IL,IU                                                    1034
            XA(I) = QA1(I)*Z1                                            1035
 2       CONTINUE                                                        1036
                                                   GO TO 5               1037
C     ELSE  NONPARALLELOGRAM CASE                                        1038
 3       DO 4 I1=IL,IU                                                   1039
            I2 = I1 - IA                                                 1040
            XA(I1) = QA1(I1)*Z1 + QA2(I2)*Z2 + QA3(I2-IA)*Z3             1041
 4       CONTINUE                                                        1042
 5    CONTINUE                                                          1043
C     . EVALUATION OF XA IS NOW COMPLETE.                                1044
C     ***********************************************                    1045
C                                                                        1046
      IL = IXB + 1 - IB                                                  1047
      IU = IXB + 2*IB                                                    1048
C     . READ IN  QB3, QB2  AND  QB1 .                                    1049
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+2)                            1050
      IL = IXB + 1                                                       1051
      IU = IXB + IB                                                      1052
      IF (PARA)                                    GO TO 6               1053
                                                   GO TO 8               1054
C     THEN  PARALLELOGRAM CASE                                           1055
 6       DO 7 I1=IL,IU                                                   1056
            CB = QB2(I1)                                                 1057
            CC = QB3(I1-IB)                                              1058
            XB(I1) =     CB*Y2 + CC*Y3                                   1059
            XB(I1+IB) = -CB*X2 - CC*X3                                   1060
 7       CONTINUE                                                        1061
                                                   GO TO 10              1062
C     ELSE  NONPARALLELOGRAM CASE                                        1063
 8       DO 9 I1=IL,IU                                                   1064
            I2 = I1 + IB                                                 1065
            CA = QB1(I2)                                                 1066
            CB = QB2(I1)                                                 1067
            CC = QB3(I1-IB)                                              1068
            XB(I1) =  CA*Y1 + CB*Y2 + CC*Y3                              1069
            XB(I2) = -CA*X1 - CB*X2 - CC*X3                              1070
 9       CONTINUE                                                        1071
 10   CONTINUE                                                          1072
C     . EVALUATION OF XB IS NOW COMPLETE.                                1073
C     ***********************************************                    1074
C                                                                        1075
      IF (PARA)                                    GO TO 11              1076
                                                   GO TO 13              1077
C     THEN  PARALLELOGRAM CASE                                           1078
 11      IL = IXC + 1                                                    1079
         IU = IXC + 4*IC                                                 1080
```

```
C        . READ IN  QC4, QC3, QC2  AND  QC1 .                          1081
         CALL READMS(2,SPACE(IL),IU-IL+1,NREC+4)                       1082
         IU = IXC + IC                                                 1083
         DO 12 I1=IL,IU                                                1084
            I2 = I1 + IC                                               1085
            I3 = I2 + IC                                               1086
            I4 = I3 + IC                                               1087
            CA = QC1(I4)                                               1088
            CB = QC2(I3)                                               1089
            CC = QC3(I2)                                               1090
            CD = QC4(I1)                                               1091
            C1 = CB*Y3 + CD*Y2                                         1092
            C2 = CA*Y3 + CC*Y2                                         1093
            XC(I1) = -(C1*Y2+C2*Y3)*Z1INV                              1094
            XC(I2) =  (C1*X2+C2*X3)*Z1INV                              1095
            C1 = CB*X3 + CD*X2                                         1096
            C2 = CA*X3 + CC*X2                                         1097
            XC(I3) =  (C1*Y2+C2*Y3)*Z1INV                              1098
            XC(I4) = -(C1*X2+C2*X3)*Z1INV                              1099
 12      CONTINUE                                                      1100
C        . EVALUATION OF XC IS NOW COMPLETE.                           1101
                                                    GO TO 22           1102
C     ********************************************                     1103
C     ELSE  NONPARALLELOGRAM CASE                                      1104
 13      IL = IXC + 1                                                  1105
         IU = IXC + 2*IC                                               1106
C        . READ IN  KC  AND  LC .                                      1107
         CALL READMS(2,SPACE(IL),IU-IL+1,NREC+3)                       1108
         IF (SSS)                                   GO TO 14           1109
                                                    GO TO 15           1110
C        THEN  FIRST TRAPEZOIDAL CASE                                  1111
 14         DLG = ELOG(R)                                              1112
            RS2 = R*R                                                  1113
            CLG = 2./3. + RS2*DLG                                      1114
                                                    GO TO 21           1115
C        ELSE                                                          1116
 15      IF (RRR)                                   GO TO 16           1117
                                                    GO TO 17           1118
C        THEN  SECOND TRAPEZOIDAL CASE                                 1119
 16         DLG = ELOG(S)                                              1120
            RS2 = S*S                                                  1121
            CLG = 2./3. + RS2*DLG                                      1122
                                                    GO TO 21           1123
C        ELSE  TRAPEZIUM CASE                                          1124
 17         IF (NNE.EQ.4)                          GO TO 18           1125
                                                    GO TO 19           1126
C           THEN                                                       1127
 18            RR2 = RR*RR                                           1128
               SS2 = SS*SS                                           1129
               VLG1 = WLOG1(R,S)                                     1130
               VLG2 = WLOG2(R,S)                                     1131
               VLG3 = WLOG2(S,R)                                     1132
               ULG1 = -14./3. + 2.*(R2+S2) + RR2*SS2*VLG1            1133
               ULG2 =    2./3. + 2.*R2 + SS2*VLG2                    1134
               ULG3 =    2./3. + 2.*S2 + RR2*VLG3                    1135
C              . ALG1 = RR*SS*(-2.+RR2*SS2*ULG1)                     1136
C              . ALG2 = SS*(2. + SS2*ULG2)                           1137
C              . ALG3 = RR*(2. + RR2*ULG3)                           1138
               VLG1 = VLG1*(RD*SD)**5                                1139
               VLG2 = VLG2*RD**5                                     1140
               VLG3 = VLG3*SD**5                                     1141
               ULG1 = ULG1*(RD*SD)**3                                1142
               ULG2 = ULG2*RD**3                                     1143
               ULG3 = ULG3*SD**3                                     1144
                                                    GO TO 20           1145
C           ELSE  NNE EQUALS 8                                         1146
 19            ALG1 = BLOG(Z1,Z2,Z3)                                 1147
               ALG2 = BLOG(Z2,Z3,Z1)                                 1148
               ALG3 = BLOG(Z3,Z1,Z2)                                 1149
 20         CONTINUE                                                  1150
```

```
 21      CONTINUE                                                        1151
C        . EVALUATION OF  XC  IS DEFERRED TO ANOTHER OVERLAY.            1152
 22   CONTINUE                                                           1153
C  ...ONLY EXIT.                                                        1154
      END                                                               1155



      FUNCTION BLOG(C,A,B)                                              1156
C*************************************************************************1157
C                                                                       1158
C                                                      1+X              1159
C     THIS SUBROUTINE COMPUTES X AND THEN  BLOG = LOG(---)/2.           1160
C                                                      1-X              1161
C     FOR SMALL X (X .LE. 0.1)  THE FOLLOWING EXPANSION IS USED IN ORDER1162
C        TO GIVE GREATER ACCURACY                                       1163
C                                                                       1164
C                      3    5    7    9    11   13   15   17           1165
C                     X    X    X    X    X    X    X    X            1166
C        BLOG = X + -- + -- + -- + -- + --- + --- + --- + ---           1167
C                     3    5    7    9    11   13   15   17            1168
C                                                                       1169
C     NOTE THAT  BLOG(C,A,B)  IS SYMMETRIC UNDER INTERCHANGE OF A AND B.1170
C                                                                       1171
C*************************************************************************1172
C                                                                       1173
      COMMON/TEMP/X,X2,N,OTHERS(1)                                      1174
C                                                                       1175
C                                                                       1176
      X = 2.*A*B/(A*A+B*B-C*C)                                          1177
      IF (ABS(X) .LT. 0.1)                    GO TO 1                   1178
                                              GO TO 3                   1179
C     THEN                                                              1180
 1       N = 8                                                          1181
         X2 = X*X                                                       1182
         BLOG = 1./(2.*N+1.)                                            1183
         DO 2 I=1,N                                                     1184
 2          BLOG = 1./(2.*(N-I)+1.) + X2*BLOG                           1185
C        CONTINUE                                                       1186
         BLOG = X*BLOG                                                  1187
                                              GO TO 4                   1188
C     ELSE                                                              1189
 3       BLOG = ALOG((1.+X)/(1.-X))/2.                                  1190
 4    CONTINUE                                                          1191
      RETURN                                                            1192
      END                                                               1193
```

```
      FUNCTION ELOG(R)                                                  1194
C*************************************************************************1195
C                                                                       1196
C                                                 1+R          2  3   5 1197
C     THIS SUBROUTINE COMPUTES  ELOG = (LOG(---) - 2*R - -*R )/R .      1198
C                                                 1-R          3       1199
C                                                                       1200
C     FOR SMALL R (R .LE. 0.1)  A TAYLOR SERIES EXPANSION IS USED FOR   1201
C        GREATER ACCURACY.                                              1202
C                                                                       1203
C                        2       4       6       8       10      12      14 1204
C              2   2*R     2*R     2*R     2*R     2*R     2*R     2*R     1205
C        ELOG = - + ---- + ---- + ---- + ---- + ----- + ----- + -----    1206
C              5    7       9       11      13      15      17      19     1207
C                                                                    16 1208
C                                                                 2*R   1209
C                                                              + -----. 1210
C                                                                21     1211
C                                                                       1212
C     THE FUNCTION  BLOG  IS RELATED TO  ELOG  BY                       1213
C                                                 1+R          2  3      5 1214
C        BLOG(0,1,R) = BLOG(0,R,1) = LOG(---) = 2*R + -*R  + ELOG(R)*R . 1215
C                                                 1-R          3         1216
C                                                                       1217
C*************************************************************************1218
C                                                                       1219
      COMMON/TEMP/R2,N,Y                                                1220
C                                                                       1221
C                                                                       1222
      R2 = R*R                                                          1223
      IF (ABS(R) .LT. 0.25)                        GO TO 1              1224
                                                   GO TO 3              1225
C     THEN                                                              1226
 1       N = 8                                                          1227
         ELOG = 2./(2.*N+5.)                                            1228
         DO 2 I=1,N                                                     1229
 2          ELOG = 2./(2.*(N-I)+5.) + R2*ELOG                           1230
C        CONTINUE                                                       1231
                                                   GO TO 4              1232
C     ELSE                                                              1233
 3       ELOG = (ALOG((1.+R)/(1.-R))-2.*R*(1.+R2/3.))/(R*R2*R2)         1234
 4    CONTINUE                                                          1235
      RETURN                                                            1236
      END                                                               1237
```

```
      FUNCTION WLOG1(R1,S1)                                                  1238
C*************************************************************************  1239
C                                                                            1240
C     THIS SUBROUTINE EVALUATES                                              1241
C                                                                            1242
C                                    2                                       1243
C                  1     1 - (S+R)                14        2  2     3      1244
C                  -*LOG(----------) + 2*TT + (-- - 2*(R +S ))*TT )          1245
C                  2               2             3                          1246
C                        1 - (S-R)                                           1247
C     WLOG1(R,S) =  ---------------------------------------------           1248
C                                          5                                 1249
C                                        TT                                  1250
C                        R*S                                                 1251
C        WHERE  TT = ------------ .                                          1252
C                      2      2                                             1253
C                   (1-S )*(1-R )                                            1254
C                                                                            1255
C     TAYLOR SERIES EXPANSIONS ARE USED WHEN  R  OR  S  IS SMALL.            1256
C                                                                            1257
C*************************************************************************  1258
C                                                                            1259
      COMMON/TEMP/ALOG1,C0,C1,C2,C3,C4,R,RR,R2,S,SS,SS2,S2,T,TT              1260
C                                                                            1261
C                                                                            1262
      IF (ABS(S1) .LE. ABS(R1))                    GO TO 1                   1263
                                                   GO TO 2                   1264
C     THEN                                                                   1265
 1       R = R1                                                              1266
         S = S1                                                              1267
                                                   GO TO 3                   1268
C     ELSE                                                                   1269
 2       R = S1                                                              1270
         S = R1                                                              1271
C     CONTINUE  (R  IS GREATER THAN OR EQUAL TO S IN MAGNITUDE)              1272
 3    R2 = R*R                                                              1273
      SS = S/(1.-R2)                                                         1274
      SS2=SS*SS                                                              1275
      IF (SS2 .LT. 0.01)                           GO TO 4                   1276
                                                   GO TO 5                   1277
C     THEN                                                                   1278
 4       C0 = -46. + R2*(40.-10.*R2)                                         1279
         C1 = 56. + R2*(-435.+R2*(497.+R2*(-217.+35.*R2)))                   1280
         C2 = -126.+R2*(-1341.+R2*(-6628.+R2*(11826.+R2*(-7434.              1281
     *      +R2*(2163.-252.*R2)))))                                          1282
         C3 = R2*(-30096.+R2*(-156200.+R2*(132751.+R2*(41305.                1283
     *      +R2*(-83226.+R2*(38346.+R2*(-8085.+693.*R2)))))))                1284
         C4 = R2*(-473616.+R2*(-4453592.+R2*(-982033.+R2*(5808786.           1285
     *      +R2*(-3402451+R2*(651508.+R2*(39897.+R2*(-30030.                 1286
     *      +3003.*R2)))))))                                                 1287
         WLOG1 = 2.*(C0*0.2+SS2*(C1/7.+SS2*(C2/63.+SS2*(C3/693.              1288
     *      +SS2*C4/9009.))))                                                1289
C  ......EXIT                                                               1290
                                                   RETURN                    1291
C     ELSE                                                                   1292
 5       ALOG1 = 0.5*ALOG((1.-(R+S)**2)/(1.-(R-S)**2))                       1293
         S2 = S*S                                                            1294
         RR = R/(1.-S2)                                                      1295
         TT = RR*SS                                                          1296
         WLOG1 = (ALOG1+2.*TT+(14./3.-2.*(R2+S2))*TT**3)/TT**5               1297
C  ......EXIT                                                               1298
                                                   RETURN                    1299
      END                                                                    1300
```

```
      FUNCTION WLOG2(R,S)                                                1301
C*********************************************************************  1302
C                                                                        1303
C     THIS SUBROUTINE EVALUATES                                          1304
C                                                                        1305
C                                                  2 1   3               1306
C                              2    2            2*(R +-)*R       2 5     1307
C                    1     (1+S)  - R      2*R            3      (1-S )  1308
C     WLOG2(R,S) = (-*LOG(-----------) - ------ - ----------)*------- .  1309
C                    2           2    2         2          2 3       5    1310
C                         (1-S)  - R     (1-S )     (1-S )       R       1311
C                                                                        1312
C     TAYLOR SERIES EXPANSIONS ARE USED WHEN  R  OR  S  IS SMALL.        1313
C                                                                        1314
C*********************************************************************  1315
C                                                                        1316
      COMMON/TEMP/ALOG2,C0,C1,C2,C3,C4,C5,C6,RR,RR2,R2,SS,SS2,S2,THIRD   1317
C                                                                        1318
C                                                                        1319
      R2 = R*R                                                           1320
      SS = S/(1.-R2)                                                     1321
      SS2 = SS*SS                                                        1322
      IF (SS2 .LT. 0.01)                          GO TO 1                1323
                                                  GO TO 2                1324
C     THEN                                                               1325
 1       C0 = 1.+R2*(10.+5.*R2)                                          1326
         C1 = 1.+R2*(21.+R2*(35.+7.*R2))                                 1327
         C2 = 1.+R2*(36.+R2*(126.+R2*(84.+9.*R2)))                       1328
         C4 = 1.+R2*(78.+R2*(715.+R2*(1716.+R2*(1287.+R2*(286.+13.*R2))))1329
     *      ))                                                           1330
         WLOG2 = 2.*(C0*0.2+SS2*(C1/7.+SS2*(C2/9.+SS2*(C3/11.+SS2*C4/13.1331
     *      ))))                                                         1332
C  ......EXIT                                                           1333
                                                  RETURN                 1334
C     ELSE                                                               1335
 2     S2 = S*S                                                          1336
      RR = R/(1.-S2)                                                     1337
      RR2 = RR*RR                                                        1338
      IF (RR2 .LT. 0.01)                          GO TO 3                1339
                                                  GO TO 4                1340
C     THEN                                                               1341
 3       THIRD = 1./3.                                                   1342
         C1 = 6. - 4.*S2                                                 1343
         C2 = -2. + S2*(28.+S2*(-32.+S2*10.))                            1344
         C3 = 4. + S2*(60.+S2*(40.+S2*(-174.+S2*(120.-26.*S2))))*THIRD   1345
         C4 = -2. + S2*(222.+S2*(10.+S2*(-224.+S2*(52.+S2*(88.+S2*       1346
     *      (-56.+S2*10.))))))*THIRD                                     1347
         C5 = 0.4 + S2*(222.+S2*(856.+S2*(-1198.+S2*(444.+S2*(-30.+S2*   1348
     *      (14.-2.*S2))))))*THIRD                                       1349
         C6 = S2*(3.-S2)*(12.+S2*(9.+S2*(-6.+S2)))                       1350
     *      *(8.+S2*(54.+S2*(-36.+S2*6.)))*THIRD                        1351
C        . ELOG(S) = (ALOG((1+S)/(1-S))-2*S-2/3*S**3)/S**5 .             1352
         WLOG2 = ELOG(S)*(1.-R2)**5 + RR2*(C1+RR2*(C2+RR2*(C3+RR2*(C4    1353
     *      +RR2*(C5+RR2*C6)))))                                         1354
C        . THIS EXPANSION IS ACCURATE FOR  S=0  AS WELL AS FOR  R=0 .    1355
C  ......EXIT                                                           1356
                                                  RETURN                 1357
C     ELSE  (RR AND SS ARE BOTH GREATER THAN 0.1)                        1358
 4       ALOG2 = 0.5*ALOG(((1.+S)*(1.+S)-R2)/((1.-S)*(1.-S)-R2))         1359
         WLOG2 = (ALOG2 - 2.*SS*(1.+(1./3.+R2)*SS2))/SS*SS2*SS2          1360
C  ......EXIT                                                           1361
                                                  RETURN                 1362
      END
```

```
      OVERLAY(MAIN,4,0)                                                    1363
      PROGRAM TRAP5                                                        1364
C*************************************************************************1365
C                                                                          1366
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR TRAPEZOIDAL ELEMENTS          1367
C        WITH  NNE = 4.                                                    1368
C                                                                          1369
C     THE  DO  LOOPS ENDING AT STATEMENTS NUMBERED  3 AND 4  CARRY OUT     1370
C        GROUP TRANSFORMATIONS.                                            1371
C                                                                          1372
C*************************************************************************1373
C                                                                          1374
      DIMENSION LC(1),KC(1)                                                1375
      COMMON/SPACE/XC(107),LIMIT,SKIP(2),IC,SKP(6),IXC,SKIPP(2),           1376
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,DLG,RS2,CLG,OTHERS(1)         1377
      COMMON/TEMP/I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,R,S,                   1378
     *   IL,IU,LL,KI,XJ,TEMP                                               1379
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))                              1380
C                                                                          1381
C                                                                          1382
      X1 = XX1                                                             1383
      X2 = XX2                                                             1384
      X3 = XX3                                                             1385
      Y1 = YY1                                                             1386
      Y2 = YY2                                                             1387
      Y3 = YY3                                                             1388
      Z1 = ZZ1                                                             1389
      R  = ZZ2/ZZ1                                                         1390
      S  = ZZ3/ZZ1                                                         1391
      IL = IXC + 1                                                         1392
      IU = IXC + LIMIT                                                     1393
      LL = 0                                                               1394
      DO 4 I=1,2                                                           1395
         DO 3 J=1,4                                                        1396
            LL = LL + 1                                                    1397
            KI = 0                                                         1398
            IF (ABS(R).LT.ABS(S))                                          1399
C           THEN  SECOND TYPE OF TRAPEZOID                                 1400
     *         KI = 5                                                      1401
C           CONTINUE                                                       1402
            DO 2 I1=IL,IU                                                  1403
               I2 = I1 + IC                                                1404
               I3 = I2 + IC                                                1405
               I4 = I3 + IC                                                1406
               K = KC(I1) + KI                                             1407
               L = LC(I2)                                                  1408
C              IF (L.EQ.LL)                                                1409
                                                  IF(L.NE.LL) GO TO 1      1410
C              THEN                                                        1411
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)                       1412
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)                       1413
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)                       1414
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)                       1415
 1             CONTINUE                                                    1416
 2          CONTINUE                                                      1417
C           . TRANSFORMATION OF TYPE ONE.                                  1418
            X1 = -X1                                                       1419
            XJ =  X2                                                       1420
            X2 = -X3                                                       1421
            X3 =  XJ                                                       1422
            Y1 = -Y1                                                       1423
            XJ =  Y2                                                       1424
            Y2 = -Y3                                                       1425
            Y3 =  XJ                                                       1426
            XJ =  R                                                        1427
            R  =  S                                                        1428
            S  = -XJ                                                       1429
 3       CONTINUE                                                          1430
```

```
C         . TRANSFORMATION OF TYPE TWO.                          1431
          X3 = -X3                                               1432
          Y3 = -Y3                                               1433
          S  = -S                                                1434
 4    CONTINUE                                                   1435
      END                                                        1436
```

APPENDIX A

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)                                  1437
C*************************************************************************1438
C                                                                         1439
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM TRAP5 TO EVALUATE          1440
C        C-INTEGRALS.                                                     1441
C                                                                         1442
C     RS2  IS THE LARGER OF THE TWO QUANTITIES  R*R  AND S*S.             1443
C                                                                         1444
C*************************************************************************1445
C                                                                         1446
C                                                                         1447
      COMMON/SPACE/SKIP(129),DLOG,RS2,CLOG,OTHERS(1)                      1448
      COMMON/TEMP/I2,I3,I4,K,L,XY(6),Z1,R,S,                              1449
     *   IL,IU,LL,KI,XJ,TEMP,XY11,XY22,XY33,XY12,XY31,X23,Y12,Y23,        1450
     *   Y31,X1X2,X3X1,Y1Y2,Y2Y3,Y3Y1                                     1451
C                                                                         1452
C                                                                         1453
      GO TO (1,2,3,4,5,11,12,13,14,15),K                                  1454
C        . 1,1 .                                                          1455
 1       X23 = X2 - X3                                                    1456
         Y23 = Y2 - Y3                                                    1457
         X3X1 = X3 + X1                                                   1458
         Y3Y1 = Y3 + Y1                                                   1459
         TEMP = (X1+X2)*(Y1+Y2) + 3.*X23*Y23                              1460
         XDNDN = -(2.*TEMP + CLOG*(3.*X3X1*Y3Y1+3.*R*(X3X1*Y23+X23*Y3Y1)  1461
     *      +RS2*TEMP))/(24.*Z1)                                          1462
            RETURN                                                        1463
C        . 2,1 .                                                          1464
 2       X23 = X2 - X3                                                    1465
         Y31 = Y3 - Y1                                                    1466
         Y2Y3 = Y2 + Y3                                                   1467
         X3X1 = X3 + X1                                                   1468
         TEMP = (X1+X2)*(Y1+Y2) - 3.*X23*Y2Y3                             1469
         XDNDN = (2.*TEMP + CLOG*(3.*X3X1*Y31+3.*R*(X23*Y31-X3X1*Y2Y3)    1470
     *      +RS2*TEMP))/(24.*Z1)                                          1471
         RETURN                                                           1472
C        . 3,1 .                                                          1473
 3       X23 = X2 - X3                                                    1474
         Y23 = Y2 - Y3                                                    1475
         Y31 = Y3 - Y1                                                    1476
         X3X1 = X3 + X1                                                   1477
         TEMP = (X1+X2)*(Y1-Y2) + 3.*X23*Y23                              1478
         XDNDN = (2.*TEMP + CLOG*(-3.*X3X1*Y31+3.*R*(X3X1*Y23-X23*Y31)    1479
     *      +RS2*TEMP))/(24.*Z1)                                          1480
         RETURN                                                           1481
C        . 5,1 .                                                          1482
 4       X23 = X2 - X3                                                    1483
         X1X2 = X1 + X2                                                   1484
         X3X1 = X3 + X1                                                   1485
         TEMP1 = -X23*Y1 + X1X2*Y3                                        1486
         TEMP = 2.*(X3X1*Y2+X1X2*Y3) - R*(X3X1*Y1-2.*X23*Y2) + RS2*TEMP1  1487
         XDNDN = -(2.*TEMP + 3.*DLOG*(3.*R*X3X1*Y1+RS2*(TEMP-3.*TEMP1)))  1488
     *      /(9.*Z1)                                                      1489
         RETURN                                                           1490
C        . 5,5 .                                                          1491
 5       XY12 = X1*Y2 + X2*Y1                                             1492
         XY11 = X1*Y1                                                     1493
         XY22 = X2*Y2                                                     1494
         XY33 = X3*Y3                                                     1495
         TEMP1 = -3.*XY11 - 5.*XY33                                       1496
         TEMP = XY11-8.*XY22-5.*XY33 + 2.*R*XY12 + RS2*TEMP1              1497
         XDNDN = 8.*(2.*TEMP + 3.*DLOG*(-15.*XY11-5.*XY33 - 10.*R*XY12    1498
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)                               1499
         RETURN                                                           1500
C        . 1,1 .                                                          1501
 11      X23 = X2 - X3                                                    1502
         Y23 = Y2 - Y3                                                    1503
         X1X2 = X1 + X2                                                   1504
         Y1Y2 = Y1 + Y2                                                   1505
         TEMP = (X3+X1)*(Y3+Y1) + 3.*X23*Y23                              1506
```

```
      XDNDN = -(2.*TEMP + CLOG*(3.*X1X2*Y1Y2-3.*S*(X23*Y1Y2+X1X2*Y23)
     *    +RS2*TEMP))/(24.*Z1)
      RETURN
C     . 2,1 .
 12   Y31 = Y3 - Y1
      X1X2 = X1 + X2
      Y1Y2 = Y1 + Y2
      X3X1 = X3 + X1
      TEMP = X3X1*Y31 - 3.*(X2-X3)*(Y2+Y3)
      XDNDN = (2.*TEMP + CLOG*(3.*X1X2*Y1Y2+3.*S*(X1X2*Y31+X3X1*Y1Y2)
     *    +RS2*TEMP))/(24.*Z1)
      RETURN
C     . 3,1 .
 13   Y12 = Y1 - Y2
      Y31 = Y3 - Y1
      X1X2 = X1 + X2
      X3X1 = X3 + X1
      TEMP = -X3X1*Y31 + 3.*(X2-X3)*(Y2-Y3)
      XDNDN = (2.*TEMP + CLOG*(3.*X1X2*Y12+3.*S*(X1X2*Y31+X3X1*Y12)
     *    +RS2*TEMP))/(24.*Z1)
      RETURN
C     . 5,1 .
 14   X23 = X2 - X3
      X1X2 = X1 + X2
      X3X1 = X3 + X1
      TEMP1 = X23*Y1 + X3X1*Y2
      TEMP = 2.*(X1X2*Y3+X3X1*Y2) - S*(X1X2*Y1+2.*X23*Y3) + RS2*TEMP1
      XDNDN = -(2.*TEMP + 3.*DLOG*(3.*S*X1X2*Y1+RS2*(TEMP-3.*TEMP1)))
     *    /(9.*Z1)
      RETURN
C     . 5,5 .
 15   XY31 = X1*Y3 + X3*Y1
      XY11 = X1*Y1
      XY22 = X2*Y2
      XY33 = X3*Y3
      TEMP1 = -3.*XY11 - 5.*XY22
      TEMP = XY11-5.*XY22-8.*XY33 + 2.*S*XY31 + RS2*TEMP1
      XDNDN = 8.*(2.*TEMP + 3.*DLOG*(-15.*XY11-5.*XY22 - 10.*S*XY31
     *    +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)
      RETURN
      END
```

APPENDIX A

```
      OVERLAY(MAIN,5,0)                                                    1548
      PROGRAM TRAP9                                                        1549
C*************************************************************************1550
C                                                                          1551
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR TRAPEZOIDAL ELEMENTS          1552
C        WITH  NNE = 8.                                                    1553
C                                                                          1554
C     THE  DO  LOOPS ENDING AT STATEMENTS NUMBERED  3 AND 4  CARRY OUT     1555
C        GROUP TRANSFORMATIONS.                                            1556
C                                                                          1557
C*************************************************************************1558
C                                                                          1559
      DIMENSION LC(1),KC(1)                                                1560
      COMMON/SPACE/XC(107),LIMIT,SKIP(2),IC,SKP(6),IXC,SKIPP(2),           1561
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,DLG,RS2,CLG,OTHERS(1)         1562
      COMMON/TEMP/I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,R,S,                   1563
     *   IL,IU,LL,KI,XJ,TEMP                                               1564
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))                              1565
C                                                                          1566
C                                                                          1567
      X1 = XX1                                                             1568
      X2 = XX2                                                             1569
      X3 = XX3                                                             1570
      Y1 = YY1                                                             1571
      Y2 = YY2                                                             1572
      Y3 = YY3                                                             1573
      Z1 = ZZ1                                                             1574
      R  = ZZ2/ZZ1                                                         1575
      S  = ZZ3/ZZ1                                                         1576
      IL = IXC + 1                                                         1577
      IU = IXC + LIMIT                                                     1578
      LL = 0                                                               1579
      DO 4 I=1,2                                                           1580
         DO 3 J=1,4                                                        1581
            LL = LL + 1                                                    1582
            KI = 0                                                         1583
            IF (ABS(R).LT.ABS(S))                                          1584
C           THEN  SECOND TYPE OF TRAPEZOID                                 1585
     *         KI = 11                                                     1586
C           CONTINUE                                                       1587
            DO 2 I1=IL,IU                                                  1588
               I2 = I1 + IC                                                1589
               I3 = I2 + IC                                                1590
               I4 = I3 + IC                                                1591
               K = KC(I1) + KI                                             1592
               L = LC(I2)                                                  1593
C              IF (L.EQ.LL)                                                1594
                                                  IF(L.NE.LL) GO TO 1      1595
C              THEN                                                        1596
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)                       1597
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)                       1598
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)                       1599
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)                       1600
 1             CONTINUE                                                    1601
 2          CONTINUE                                                       1602
C           . TRANSFORMATION OF TYPE ONE.                                  1603
            X1 = -X1                                                       1604
            XJ =  X2                                                       1605
            X2 = -X3                                                       1606
            X3 =  XJ                                                       1607
            Y1 = -Y1                                                       1608
            XJ =  Y2                                                       1609
            Y2 = -Y3                                                       1610
            Y3 =  XJ                                                       1611
            XJ =  R                                                        1612
            R  =  S                                                        1613
            S  = -XJ                                                       1614
 3       CONTINUE                                                          1615
```

```
C         . TRANSFORMATION OF TYPE TWO.                                  1616
          X3 = -X3                                                       1617
          Y3 = -Y3                                                       1618
          S  = -S                                                        1619
 4    CONTINUE                                                          1620
      END                                                               1621
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)                                 1622
C*************************************************************************1623
C                                                                         1624
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM TRAP9 TO EVALUATE          1625
C        C-INTEGRALS.                                                     1626
C                                                                         1627
C     RS2  IS THE LARGER OF THE TWO QUANTITIES  R*R  AND S*S.             1628
C                                                                         1629
C*************************************************************************1630
C                                                                         1631
      COMMON/SPACE/SKIP(129),DLOG,RS2,CLOG,OTHERS(1)                      1632
      COMMON/TEMP/I2,I3,I4,K,L,XY(6),Z1,R,S,                              1633
     *   IL,IU,LL,KI,XJ,TEMP,TEMP1,XY11,XY22,XY33,XY12,XY23,XY31,         1634
     *   YX21,YX32,YX13                                                   1635
C                                                                         1636
C                                                                         1637
      XY11 = X1*Y1                                                        1638
      XY22 = X2*Y2                                                        1639
      XY33 = X3*Y3                                                        1640
      XY12 = X1*Y2 + X2*Y1                                                1641
      XY23 = X2*Y3 + X3*Y2                                                1642
      XY31 = X3*Y1 + X1*Y3                                                1643
      YX21 = X1*Y2 - X2*Y1                                                1644
      YX32 = X2*Y3 - X3*Y2                                                1645
      YX13 = X3*Y1 - X1*Y3                                                1646
      GO TO (101,102,103,104,105,106,107,108,109,110,111,                 1647
     *   201,202,203,204,205,206,207,208,209,210,211),K                   1648
C        . 1,1 .                                                          1649
 101     TEMP1 = -12.*XY11-8.*XY22-20.*XY33-6.*XY12+10.*XY23               1650
         TEMP = XY11-104.*XY22-95.*XY33-13.*XY12+85.*XY23+5.*XY31          1651
     *      +R*(12.*XY11+40.*(XY22+XY33)+23.*XY12-35.*XY23-10.*XY31)       1652
     *      +RS2*TEMP1                                                     1653
         XDNDN = (2.*TEMP + 3.*DLOG*(-65.*XY11-15.*XY33-30.*XY31           1654
     *      +R*(-20.*XY11+30.*XY33-70.*XY12-30.*XY23+35.*XY31)             1655
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)                               1656
            RETURN                                                        1657
C        . 2,1 .                                                          1658
 102     TEMP1 = -12.*XY11+2.*XY22-20.*XY33-6.*XY12+10.*YX32               1659
         TEMP = XY11-34.*XY22-65.*XY33-23.*XY12+25.*YX13+15.*YX32          1660
     *      +R*(12.*XY11+40.*XY33+23.*XY12-10.*YX13-25.*YX32)              1661
     *      +RS2*TEMP1                                                     1662
         XDNDN = (2.*TEMP + 3.*DLOG*(-55.*XY11+15.*XY33-30.*YX13           1663
     *      +R*(-20.*XY11-30.*XY33-50.*XY12+15.*YX13+30.*YX32)             1664
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)                               1665
            RETURN                                                        1666
C        . 3,1 .                                                          1667
 103     TEMP1 = -12.*XY11-2.*XY22-20.*XY33+6.*YX21+10.*XY23               1668
          TEMP = 7.*XY11-46.*XY22-55.*XY33+35.*XY23+15.*YX13+3.*YX21       1669
     *       +R*(17.*XY12-10.*XY31-15.*YX32) + RS2*TEMP1                   1670
          XDNDN = (2.*TEMP + 3.*DLOG*(-55.*XY11+15.*XY33-30.*YX13          1671
     *       +R*(-50.*XY12+25.*XY31+30.*YX32)                              1672
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)                               1673
            RETURN                                                        1674
C        . 5,1 .                                                          1675
 104     TEMP1 = 6.*XY11-XY22+10.*XY33+0.5*XY12-2.5*YX21-5.*X2*Y3          1676
         TEMP = 12.*XY11-3.*XY22+40.*XY33+16.5*XY12-12.5*YX21-10.*XY23     1677
     *      -15.*YX32-5.*XY31-10.*YX13                                     1678
     *      +R*(-11.*XY11+10.*XY22-20.*XY33-6.5*XY12+7.5*YX21+5.*XY23      1679
     *      +10.*YX32+5.*X3*Y1) + RS2*TEMP1                                1680
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(20.*XY11+2.5*XY31+7.5*YX13)       1681
     *      +RS2*(TEMP-3.*TEMP1)))/(90.*Z1)                                1682
            RETURN                                                        1683
C        . 5,3 .                                                          1684
 105     TEMP1 = 6.*XY11+XY22+10.*XY33-2.5*XY12+0.5*YX21-5.*X2*Y3          1685
         TEMP =4.*XY11+3.*XY22+20.*XY33+7.5*XY12-3.5*YX21-5.*(XY23+XY31)   1686
     *      +R*(-5.*XY11+10.*XY22-3.5*XY12+2.5*YX21-5.*XY23+5.*X3*Y1)      1687
     *      +RS2*TEMP1                                                     1688
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(20.*XY11-2.5*XY31-7.5*YX13)       1689
     *      +RS2*(TEMP-3.*TEMP1)))/(90.*Z1)                                1690
            RETURN                                                        1691
```

```
C        . 5,5 .                                                          1692
 106     TEMP1 = -6.*XY11-4.*XY22-10.*XY33+2.*XY12                        1693
         TEMP = -32.*XY11-12.*XY22-40.*XY33+6.*XY12                       1694
     *      +R*(16.*XY11+20.*XY33-6.*XY12) + RS2*TEMP1                    1695
         XDNDN = (2.*TEMP + 3.*DLOG*RS2*(TEMP+18.*XY11+12.*XY22+30.*XY33  1696
     *      -6.*XY12))/(45.*Z1)                                           1697
                RETURN                                                   1698
C        . 6,5 .                                                          1699
 107     TEMP1 = XY31-X2*Y3                                               1700
         TEMP = 2.*XY11-3.*XY12-YX21-2.*XY23+3.*XY31+YX13                 1701
     *      +R*(-2.*XY22+2.*(X1+X3)*Y2-1.5*XY31+0.5*YX13) + RS2*TEMP1     1702
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(-4.*XY11+0.5*XY31-1.5*YX13)       1703
     *      +RS2*(TEMP-3.*TEMP1)))/(9.*Z1)                                1704
                RETURN                                                   1705
C        . 7,5 .                                                          1706
 108     TEMP1 = -6.*XY11+4.*XY22-10.*XY33-2.*YX21                        1707
         TEMP = -4.*XY11+12.*XY22-20.*XY33-6.*YX21 +6.*R*XY12 +RS2*TEMP1  1708
         XDNDN = (2.*TEMP + 3.*DLOG*RS2*(TEMP-3.*TEMP1))/(45.*Z1)         1709
                RETURN                                                   1710
C        . 9,1 .                                                          1711
 109     TEMP1 = 6.*XY11+10.*XY33+X2*(3.*Y1-5.*Y3)                        1712
         TEMP = -2.*XY11+20.*(XY22+XY33)+3.*XY12-YX21-10.*XY23-5.*YX13    1713
     *      +R*(-3.*XY11-2.*XY22-10.*XY33-7.*XY12+3.*YX21+10.*X2*Y3       1714
     *      +5.*X3*Y1) + RS2*TEMP1                                        1715
         XDNDN = (2.*TEMP + 3.*DLOG*(30.*XY11+5.*XY31+10.*YX13            1716
     *      +R*(5.*XY11+10.*XY33+25.*XY12-5.*YX21-10.*YX32-15.*X3*Y1)     1717
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)                               1718
                RETURN                                                   1719
C        . 9,5 .                                                          1720
 110     TEMP1 = -12.*XY11-20.*XY33+4.*X2*Y1                              1721
         TEMP = -16.*XY11-40.*XY33-16.*XY12-8.*YX21                       1722
     *      +R*(16.*XY11-16.*XY22+20.*XY33+8.*X1*Y2) + RS2*TEMP1          1723
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(-40.*XY11-20.*XY33)               1724
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)                               1725
                RETURN                                                   1726
C        . 9,9 .                                                          1727
 111     TEMP1 = -3.*XY11-5.*XY33                                         1728
         TEMP = XY11-8.*XY22-5.*XY33 + 2.*R*XY12 + RS2*TEMP1              1729
         XDNDN =8.*(2.*TEMP + 3.*DLOG*(-15.*XY11-5.*XY33 -10.*R*XY12      1730
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)                               1731
                RETURN                                                   1732
C        . 1,1 .                                                          1733
 201     TEMP1 = -12.*XY11-20.*XY22-8.*XY33+10.*XY23-6.*XY31              1734
         TEMP = XY11-95.*XY22-104.*XY33+5.*XY12+85.*XY23-13.*XY31         1735
     *      +S*(12.*XY11+40.*(XY22+XY33)-10.*XY12-35.*XY23+23.*XY31)      1736
     *      +RS2*TEMP1                                                    1737
         XDNDN = (2.*TEMP + 3.*DLOG*(-65.*XY11-15.*XY22-30.*XY12          1738
     *      +S*(-20.*XY11+30.*XY22+35.*XY12-30.*XY23-70.*XY31)            1739
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)                              1740
                RETURN                                                   1741
C        . 2,1 .                                                          1742
 202     TEMP1 = -12.*XY11-20.*XY22+8.*XY33-10.*YX32-6.*YX13              1743
         TEMP = 7.*XY11-25.*XY22-56.*XY33-5.*XY12+7.*YX13+15.*YX32        1744
     *      +S*(-5.*XY23+17.*XY31+10.*YX21) + RS2*TEMP1                   1745
         XDNDN = (2.*TEMP + 3.*DLOG*(-65.*XY11-15.*XY22-30.*XY12          1746
     *      +S*(-30.*XY23-70.*XY31-45.*YX21)                              1747
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)                              1748
                RETURN                                                   1749
C        . 3,1 .                                                          1750
 203     TEMP1 = -12.*XY11-20.*XY22-2.*XY33+10.*XY23-6.*YX13              1751
        TEMP = 7.*XY11-55.*XY22-46.*XY33+35.*XY23-3.*YX13-15.*YX21        1752
     *      +S*(-10.*XY12+17.*XY31+15.*YX32) + RS2*TEMP1                  1753
         XDNDN = (2.*TEMP + 3.*DLOG*(-55.*XY11+15.*XY22+30.*YX21          1754
     *      +S*(25.*XY12-50.*XY31-30.*YX32)                               1755
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)                              1756
                RETURN                                                   1757
C        . 5,1 .                                                          1758
 204     TEMP1 = 2.*XY12-X3*(Y1-Y2)                                       1759
```

```
      TEMP =-3.*XY11-3.*XY22+16.*XY33+3.*XY12-5.*YX21-4.*XY23-6.*YX32          1760
     *    -2.*XY31+4.*YX13                                                    1761
     *    +S*(3.*XY22-4.*XY33-1.5*XY12+0.5*YX21-XY23+5.*YX32-XY31             1762
     *    +3.*YX13) + RS2*TEMP1                                               1763
      XDNDN = (2.*TEMP + 3.*DLOG*(13.*XY11+3.*XY22+6.*XY12                    1764
     *    +S*(2.*XY11-3.*XY22-3.5*XY12+4.5*YX21+6.*XY23+14.*XY31)             1765
     *    +RS2*(TEMP-3.*TEMP1)))/(36.*Z1)                                     1766
          RETURN                                                              1767
C     . 5,3 .                                                                 1768
 205  TEMP1 = 2.*XY12-X3*(Y1+Y2)                                              1769
      TEMP = -5.*XY11+3.*XY22+8.*XY33+3.*XY12-5.*YX21-2.*XY23-2.*XY31         1770
     *    +S*(3.*XY22+1.5*XY12-0.5*YX21-XY23+5.*YX32-3.*XY31+YX13)            1771
     *    +RS2*TEMP1                                                          1772
      XDNDN = (2.*TEMP + 3.*DLOG*(11.*XY11-3.*XY22+6.*YX21                    1773
     *    +S*(-2.*XY11-3.*XY22-2.5*XY12+1.5*YX21+10.*XY31-6.*YX32)            1774
     *    +RS2*(TEMP-3.*TEMP1)))/(36.*Z1)                                     1775
          RETURN                                                              1776
C     . 5,5 .                                                                 1777
 206  TEMP1 = -XY11-3.*XY22                                                   1778
      TEMP = -5.*XY11-3.*XY22-16.*XY33+6.*XY12 +S*(6.*XY23-2.*XY31)           1779
     *    +RS2*TEMP1                                                          1780
      XDNDN = (2.*TEMP + 3.*DLOG*(-13.*XY11-3.*XY22-6.*XY12                   1781
     *    +S*(-6.*XY23-14.*XY31) + RS2*(TEMP-3.*TEMP1)))/(18.*Z1)             1782
          RETURN                                                              1783
C     . 6,5 .                                                                 1784
 207  TEMP1 = -XY12-X2*Y3                                                     1785
      TEMP = 2.*XY11-3.*XY12-YX21-2.*XY23+3.*XY31+YX13                        1786
     *    +S*(2.*XY33-1.5*XY12+0.5*YX21+2.*X3*(Y1-Y2)) + RS2*TEMP1            1787
      XDNDN = (2.*TEMP + 3.*DLOG*(S*(4.*XY11+0.5*XY12-1.5*YX21)               1788
     *    +RS2*(TEMP-3.*TEMP1)))/(9.*Z1)                                      1789
          RETURN                                                              1790
C     . 7,5 .                                                                 1791
 208  TEMP1 = XY11+3.*XY22                                                    1792
      TEMP = 5.*XY11+3.*XY22-8.*XY33-6.*YX21                                  1793
     *    +S*(2.*XY31+6.*YX32) + RS2*TEMP1                                    1794
      XDNDN = (2.*TEMP + 3.*DLOG*(-11.*XY11+3.*XY22+6.*YX21                   1795
     *    +S*(-10.*XY31-6.*YX32) + RS2*(TEMP-3.*TEMP1)))/(18.*Z1)             1796
          RETURN                                                              1797
C     . 9,1 .                                                                 1798
 209  TEMP1 = 6.*XY11+10.*XY22+X3*(3.*Y1-5.*Y2)                               1799
      TEMP = -2.*XY11+20.*(XY22+XY33)+5.*YX21-10.*XY23+3.*XY31+YX13           1800
     *    +S*(-3.*XY11-10.*XY22-2.*XY33+5.*X2*Y1+10.*X3*Y2-7.*XY31            1801
     *    -3.*YX13) + RS2*TEMP1                                               1802
      XDNDN = (2.*TEMP + 3.*DLOG*(30.*XY11+5.*XY12-10.*YX21                   1803
     *    +S*(5.*XY11+10.*XY22-15.*X2*Y1+10.*YX32+25.*XY31+5.*YX13)           1804
     *    +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)                                     1805
          RETURN                                                              1806
C     . 9,5 .                                                                 1807
 210  TEMP = 4.*XY11-8.*XY33-2.*XY12-4.*YX21 + 4.*S*(YX32+X3*Y1)              1808
     *    - 2.*RS2*XY12                                                       1809
      XDNDN = (2.*TEMP + 3.*DLOG*(-12.*XY11-2.*XY12+4.*YX21                   1810
     *    +S*(-4.*YX32-10.*XY31-2.*YX13) +RS2*(TEMP+6.*XY12)))/(9.*Z1)        1811
          RETURN                                                              1812
C     . 9,9 .                                                                 1813
 211  TEMP1 = -3.*XY11-5.*XY22                                                1814
      TEMP = XY11-5.*XY22-8.*XY33 + 2.*S*XY31 + RS2*TEMP1                     1815
      XDNDN = 8.*(2.*TEMP + 3.*DLOG*(-15.*XY11-5.*XY22 - 10.*S*XY31           1816
     *    +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)                                     1817
          RETURN                                                              1818
       END                                                                    1819
```

```
      OVERLAY(MAIN,6,0)                                                    1820
      PROGRAM QUAD5                                                        1821
C*************************************************************************1822
C                                                                          1823
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE         1824
C        ELEMENTS WITH NNE = 4.                                            1825
C                                                                          1826
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT        1827
C        GROUP TRANSFORMATIONS.                                            1828
C                                                                          1829
C*************************************************************************1830
C                                                                          1831
      DIMENSION LC(1),KC(1)                                                1832
      COMMON/SPACE/XC(107),LIMIT,SKIP(2),IC,SKP(6),IXC,SKIPP(2),           1833
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,VLG1,VLG2,VLG3,               1834
     *   ULG1,ULG2,ULG3,OTHERS(1)                                          1835
      COMMON/TEMP/CL0,CL1,CL2,CL3,I,IL,IU,I1,I2,I3,I4,J,K,L,LL,            1836
     *   R,S,R2,S2,RS,RD,SD,                                               1837
     *   T1,T2,T3,T4,T5,T6,ULOG1,ULOG2,ULOG3,U1,U2,U3,U4,U5,U6,            1838
     *   VLOG1,VLOG2,VLOG3,V2,V3,V4,V5,V6,XJ,X1,X2,X3,Y1,Y2,Y3,            1839
     *   X2X1,X3X1,X32,Y2Y1,Y3Y1,Y3Y2,Y31,Y32,Z1                           1840
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))                              1841
C                                                                          1842
C                                                                          1843
      X1 = XX1                                                             1844
      X2 = XX2                                                             1845
      X3 = XX3                                                             1846
      Y1 = YY1                                                             1847
      Y2 = YY2                                                             1848
      Y3 = YY3                                                             1849
      Z1 = ZZ1                                                             1850
      R  = ZZ2/ZZ1                                                         1851
      S  = ZZ3/ZZ1                                                         1852
      R2 = R*R                                                             1853
      S2 = S*S                                                             1854
      RS = R*S                                                             1855
      RD = 1./(1.-R2)                                                      1856
      SD = 1./(1.-S2)                                                      1857
      ULOG1 = ULG1                                                         1858
      ULOG2 = ULG2                                                         1859
      ULOG3 = ULG3                                                         1860
      VLOG1 = VLG1                                                         1861
      VLOG2 = VLG2                                                         1862
      VLOG3 = VLG3                                                         1863
      IL = IXC + 1                                                         1864
      IU = IXC + LIMIT                                                     1865
      LL = 0                                                               1866
      DO 4 I=1,2                                                           1867
         DO 3 J=1,4                                                        1868
            LL = LL + 1                                                    1869
            DO 2 I1=IL,IU                                                  1870
               I2 = I1 + IC                                                1871
               I3 = I2 + IC                                                1872
               I4 = I3 + IC                                                1873
               K  = KC(I1)                                                 1874
               L  = LC(I2)                                                 1875
C              IF (L.EQ.LL)                                                1876
                                                  IF(L.NE.LL) GO TO 1      1877
C              THEN                                                        1878
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)                       1879
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)                       1880
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)                       1881
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)                       1882
 1             CONTINUE                                                    1883
 2          CONTINUE                                                      1884
C           . TRANSFORMATION OF TYPE ONE.                                  1885
            X1 = -X1                                                       1886
            XJ =  X2                                                       1887
            X2 = -X3                                                       1888
            X3 =  XJ                                                       1889
```

```
            Y1 = -Y1                                                    1890
            XJ =  Y2                                                    1891
            Y2 = -Y3                                                    1892
            Y3 =  XJ                                                    1893
            XJ =  R                                                     1894
            R  =  S                                                     1895
            S  = -XJ                                                    1896
            XJ =  R2                                                    1897
            R2 =  S2                                                    1898
            S2 =  XJ                                                    1899
            RS = -RS                                                    1900
            XJ =  RD                                                    1901
            RD =  SD                                                    1902
            SD =  XJ                                                    1903
            XJ    = ULOG2                                               1904
            ULOG2 = ULOG3                                               1905
            ULOG3 = XJ                                                  1906
            XJ    = VLOG2                                               1907
            VLOG2 = VLOG3                                               1908
            VLOG3 = XJ                                                  1909
 3        CONTINUE                                                      1910
C           . TRANSFORMATION OF TYPE TWO.                               1911
            X3 = -X3                                                    1912
            Y3 = -Y3                                                    1913
            S = -S                                                      1914
            RS = -RS                                                    1915
 4      CONTINUE                                                        1916
        END                                                             1917
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)                                 1918
C*************************************************************************1919
C                                                                         1920
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD5 TO EVALUATE          1921
C        C-INTEGRALS.                                                     1922
C                                                                         1923
C*************************************************************************1924
C                                                                         1925
C                                                                         1926
C                                                                         1927
      COMMON/TEMP/CL0,CL1,CL2,CL3,I,IL,IU,I1,I2,I3,I4,J,K,L,LL,           1928
     *   R,S,R2,S2,RS,RD,SD,                                            1929
     *   T1,T2,T3,T4,T5,T6,ULOG1,ULOG2,ULOG3,U1,U2,U3,U4,U5,U6,          1930
     *   VLOG1,VLOG2,VLOG3,V2,V3,V4,V5,V6,XJ,XY(6),                     1931
     *   X2X1,X3X1,X32,Y2Y1,Y3Y1,Y3Y2,Y31,Y32,Z1                        1932
C                                                                         1933
C                                                                         1934
      CL04(S,SSS3,R2,S2,T1,T3,T5,U1,U2) = 2.*(RD*SD)**3*                  1935
     *   (T1*S*(4.*SSS3+R2*(108.+S2*(-114.+S2*(108.-30.*S2))             1936
     *      +R2*(186.+S2*(-264.+S2*(33.+9.*S2))+R2*(-64.+S2*(69.-6.*S2)  1937
     *      +R2*(-18.+9.*S2)))))                                         1938
     *   +T3*S*(-20.*SSS3+R2*(-546.+S2*(1140.+S2*(-582+60.*S2))          1939
     *      +R2*(-360.+S2*(15.+69.*S2)+R2*(278.-111.*S2))))              1940
     *   +T5*(-20.*SSS3+R2*(50.+S2*(-48.+S2*(-300.+154.*S2))             1941
     *      +R2*(-30.+S2*(450.+S2*(-45.-33.*S2))+R2*(-10.+S2*(-276.      1942
     *      +57.*S2)+R2*(10.+30.*S2)))))                                 1943
     *   +U1*S2*(SSS3+R2*S2*(-75.+S2*(43.+3.*S2)-9.*R2*S2))              1944
     *   +U2*(-4.+8.*SSS3+R2*S2*(-45.+S2*(48.-6.*S2)+R2*S2*(10.5-9.*S2)  1945
     *      )))                                                          1946
      CL14(R,S,R2,S2,T1,T3,T5,U1,U2) = -3.*R2*S*                         1947
     *   (T1*(S2*(-1.+S2*(6.+3.*S2))+R2*(3.-3.*S2*S2+3.*R2*(6.-S2+R2)))  1948
     *   +T3*(3.+R2*(-30.-45.*R2)+S2*(-10.+30.*R2+15.*S2))               1949
     *   +T5*S*(-1.+R2*(10.+15.*R2)-15.*S2*S2)                           1950
     *   +U1*R2*S*(1.+3.*R2)                                             1951
     *   +U2*S*(0.5+R2*(-6.+3.*S2-3.*R2)))                               1952
      CL24(R,S,R2,S2,T1,T2,T3,T4,T5,T6,U1,U2) = 3.*S*                    1953
     *   (T1*(-12.*(1.+R2)+4.*S2)  -T2*8.*R*S                            1954
     *   +T3*(60.+12.*R2-20.*S2)   -T4*8.*R*S                            1955
     *   -T5*S*(20.+4.*R2)         +T6*S*(10.+6.*R2-10.*S2)              1956
     *   -U1*S*(3.+R2-S2)          +U2*8.*S)                             1957
      CL05(R2,S2,T1,T2,T4,T6) = (RD*SD)**3*                             1958
     *   (T1*(20.+R2*(480.-1176.*S2+R2*(1596.+S2*(-1080.+762.*S2)        1959
     *      +R2*(-40.+S2*(-144.-156.*S2)+R2*(-348.+216.*S2)))))          1960
     *   +T2*(-70.+S2*(140.+S2*S2*(-140.+70.*S2))                        1961
     *      +R2*(936.+S2*(-1842.+S2*(3822.+S2*(-1974+210.*S2)))          1962
     *      +R2*(2610.+S2*(-5940.+S2*(1626.+168.*S2))                    1963
     *      +R2*(532.+S2*(982.-442.*S2) + R2*(-840.+420.*S2)))))         1964
     *   +T4*(-80.+S2*(1044.+S2*(540.+S2*(-292.-60.*S2)))                1965
     *      +R2*(-1368.+S2*(1740.+S2*(-2706.+S2*(624.+30.*S2)))          1966
     *      +R2*(-1800.+S2*(2208.+S2*(-204.-20.*S2))                     1967
     *      +R2*(824.+S2*(-708.+130.*S2) + R2*(120.-60.*S2)))))          1968
     *   +T6*(-148.+R2*(-6168.+11730.*S2+R2*(-3192.+S2*(-5472.+612.*S2)  1969
     *      +R2*(2744.-588.*S2)))))                                      1970
      CL15(R2,S2,T1,T2,T4,T6) = -3.*                                    1971
     *   (T1*R2*(-S2+R2*(6.+R2*(60.+30.*(R2-S2))))                       1972
     *   +T2*R2*(1.+S2*(-7.+35.*S2*(1.+S2))+R2*R2*(105.*(1.-S2)+70.*R2))  1973
     *   +T4*(S2*(-1.+S2*(15.+S2*(45.+5.*S2)))+R2*(2.-5.*S2*(1.+S2*S2)   1974
     *      +R2*(-30.+S2*(45.-15.*S2)+R2*(-90.+25.*S2-10.*R2))))         1975
     *   +T6*(-3.+R2*(42.-70.*S2+210.*R2*(-1.+S2-R2))))                  1976
      CL25(R2,S2,T1,T2,T3,T4,T5,T6) = -3.*                              1977
     *   (T1*(30.+S2*(-20.+6.*S2)+R2*(60.-12.*S2+6.*R2))                 1978
     *   +T2*(35.+S2*(-70.+35.*S2)+R2*(126.-42.*S2+15.*R2))              1979
     *   +T3*S2*(56.+8.*R2)                                              1980
     *   +T4*(-80.+64.*S2-48.*R2)                                        1981
     *   +T5*(40.-8.*S2+24.*R2)                                          1982
     *   +T6*(-336.+112.*S2-48.*R2))                                     1983
```

```
C                                                                       1984
C                                                                       1985
      IF (K.LE.3)                                  GO TO 1              1986
                                                   GO TO 6              1987
C     THEN                                                              1988
 1       X32 = X3 - X2                                                  1989
         X2X1 = X2 + X1                                                 1990
         GO TO (2,3,4),K                                                1991
C                                                                       1992
C        . 1,1 .                                                        1993
 2          Y32 = Y3 - Y2                                               1994
            Y2Y1 = Y2 + Y1                                              1995
            X3X1 = X3 + X1                                              1996
            Y3Y1 = Y3 + Y1                                              1997
            T1 = -X32*Y32                                               1998
            T2 = X2X1*Y2Y1                                              1999
            T3 = X3X1*Y3Y1                                              2000
            T4 = X2X1*Y3Y1 + X3X1*Y2Y1                                  2001
            T5 = X2X1*Y32 + X32*Y2Y1                                    2002
            T6 = -X3X1*Y32 - X32*Y3Y1                                   2003
                                                   GO TO 5              2004
C        . 2,1 .                                                        2005
 3          Y31 = Y3 - Y1                                               2006
            Y2Y1 = Y2 + Y1                                              2007
            X3X1 = X3 + X1                                              2008
            Y3Y2 = Y3 + Y2                                              2009
            T1 =  X32*Y3Y2                                              2010
            T2 = -X2X1*Y2Y1                                             2011
            T3 = -X3X1*Y31                                              2012
            T4 = -X2X1*Y3 - X3X1*Y2 - X32*Y1                            2013
            T5 =  T4                                                    2014
            T6 =  X32*Y31 + X3X1*Y3Y2                                   2015
                                                   GO TO 5              2016
C        . 3,1 .                                                        2017
 4          Y21 = Y2 - Y1                                               2018
            Y31 = Y3 - Y1                                               2019
            X3X1 = X3 + X1                                              2020
            T1 = X32*(Y3-Y2)                                            2021
            T2 = X2X1*Y21                                               2022
            T3 = X3X1*Y31                                               2023
            T4 = X2X1*Y31 + X3X1*Y21                                    2024
            T5 = -X2X1*Y3 + X3X1*Y2 - X32*Y1                            2025
            T6 = -T5                                                    2026
 5       CONTINUE                                                       2027
         U1 = 6.*T1                                                     2028
         U2 = T2 + T2                                                   2029
         V2 = U2*R2                                                     2030
         U3 = T3 + T3                                                   2031
         V3 = U3*S2                                                     2032
         U4 = T4*RS                                                     2033
         V4 = U4 + U4                                                   2034
         U5 = 3.*S*T5                                                   2035
         V5 = U5*R2                                                     2036
         U6 = 3.*R*T6                                                   2037
         V6 = U6*S2                                                     2038
         CL0 = 2.*RD*SD*(U1*(1.-R2-S2)+4.*(V2+V3+U4)+U4+V5+V6)          2039
         CL1 = U1*R2*S2 - V2*(1.+3.*R2) - V3*(1.+3.*S2)                 2040
     *       + U4*(1.-3.*(R2+S2)) - V5*(1.+R2-S2) - V6*(1.-R2+S2)       2041
         CL2 = U1*S2 - U2*(3.+R2) - V3 - V4 - U5 - U5                   2042
         CL3 = U1*R2 - V2 - U3*(3.+S2) - V4 - U6 - U6                   2043
         XDNDN = (CL0 + CL1*ULOG1 + CL2*ULOG2 + CL3*ULOG3)/(48.*Z1)     2044
C  ......EXIT                                                          2045
                                                   RETURN               2046
C     ELSE                                                              2047
 6    IF (K.EQ.4)                                  GO TO 7              2048
                                                   GO TO 8              2049
```

```
C      THEN                                                                    2050
C         . 5,1 .                                                              2051
 7        X32 = X3 - X2                                                        2052
          X2X1 = X2 + X1                                                       2053
          X3X1 = X3 + X1                                                       2054
          T1 = 5.*X2X1*Y1                                                      2055
          T2 = 5.*X3X1*Y1                                                      2056
          T3 = R*X2X1*Y2                                                       2057
          T4 = S*X3X1*Y3                                                       2058
          T5 = X2X1*Y3                                                         2059
          T5 = T5 + T5                                                         2060
          T6 = X3X1*Y2                                                         2061
          T6 = T6 + T6                                                         2062
          U1 = 20.*X32*Y1                                                      2063
          U2 = 5.*X32*(R*Y2-S*Y3)                                              2064
          RRR3 = (1.-R2)**3                                                    2065
          SSS3 = (1.-S2)**3                                                    2066
          TEMP1 =  CL04(S,SSS3,R2,S2,T1,T3,T5,U1,U2)                           2067
          TEMP2 =  CL04(R,RRR3,S2,R2,T2,T4,T6,-U1,U2)                          2068
          TEMP3 = CL14(R,S,R2,S2,T1,T3,T5,U1,U2)                               2069
          TEMP4 = CL14(S,R,S2,R2,T2,T4,T6,-U1,U2)                              2070
          TEMP5 = CL24(R,S,R2,S2,T1,T2,T3,T4,T5,T6,U1,U2)                      2071
          TEMP6 = CL24(S,R,S2,R2,T2,T1,T4,T3,T6,T5,-U1,U2)                     2072
          XDNDN = (TEMP1+TEMP2+(TEMP3+TEMP4)*VLOG1                             2073
     *       +TEMP5*VLOG2+TEMP6*VLOG3)/(180.*Z1)                              2074
C  ......EXIT                                                                 2075
                                                        RETURN                 2076
C      ELSE  (K  MUST EQUAL  5)                                                2077
C         . 5,5 .                                                              2078
 8        T1 = 28.*X1*Y1                                                       2079
          T2 = 8.*X2*Y2                                                        2080
          T3 = 8.*X3*Y3                                                        2081
          T4 = 14.*R*(X1*Y2+X2*Y1)                                             2082
          T5 = 14.*S*(X1*Y3+X3*Y1)                                             2083
          T6 = RS*(X2*Y3+X3*Y2)                                                2084
          TEMP1 =  CL05(R2,S2,T1,T2,T4,T6)                                     2085
          TEMP2 = CL05(S2,R2,T1,T3,T5,T6)                                      2086
          TEMP3 = CL15(R2,S2,T1,T2,T4,T6)                                      2087
          TEMP4 = CL15(S2,R2,T1,T3,T5,T6)                                      2088
          TEMP5 = CL25(R2,S2,T1,T2,T3,T4,T5,T6)                                2089
          TEMP6 = CL25(S2,R2,T1,T3,T2,T5,T4,T6)                                2090
          XDNDN = (TEMP1+TEMP2+(TEMP3+TEMP4)*VLOG1                             2091
     *       +TEMP5*VLOG2+TEMP6*VLOG3)/(315.*Z1)                              2092
C  ......EXIT                                                                 2093
                                                        RETURN                 2094
C      CONTINUE                                                                2095
       END                                                                     2096
```

APPENDIX A

```
      OVERLAY(MAIN,7,0)                                                    2097
      PROGRAM QUAD81                                                       2098
C*********************************************************************   2099
C                                                                          2100
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE         2101
C        ELEMENTS WITH NNE = 8.                                            2102
C                                                                          2103
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT        2104
C        GROUP TRANSFORMATIONS.                                            2105
C                                                                          2106
C*********************************************************************   2107
C                                                                          2108
      DIMENSION LC(1),KC(1)                                                2109
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,SKP(2),                          2110
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,ALG1,ALG2,ALG3,OTHERS(1)      2111
      COMMON/TEMP/I1,I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,              2112
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,                   2113
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,                           2114
     *   T1,T2,T3,T4,T5,T6,                                                2115
     *   DL1,EL1,DL2,FL3,XJ                                                2116
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))                              2117
C                                                                          2118
C                                                                          2119
      X1 = XX1                                                             2120
      X2 = XX2                                                             2121
      X3 = XX3                                                             2122
      Y1 = YY1                                                             2123
      Y2 = YY2                                                             2124
      Y3 = YY3                                                             2125
      Z1 = ZZ1                                                             2126
      Z2 = ZZ2                                                             2127
      Z3 = ZZ3                                                             2128
      ALOG1 = ALG1                                                         2129
      ALOG2 = ALG2                                                         2130
      ALOG3 = ALG3                                                         2131
      IL = IXC + 1                                                         2132
      IU = IXC + 10                                                        2133
      LL = 0                                                               2134
      DO 4 I=1,2                                                           2135
         DO 3 J=1,4                                                        2136
            LL = LL + 1                                                    2137
            D = Z2/Z3                                                      2138
            DP1 = D + 1.                                                   2139
            DM1 = D - 1.                                                   2140
            D2 = D*D                                                       2141
            D3 = D*D2                                                      2142
            E = -Z1/Z3                                                     2143
            EP1 = E + 1.                                                   2144
            EM1 = E - 1.                                                   2145
            E2 = E*E                                                       2146
            E3 = E*E2                                                      2147
            FF = Z1/Z2                                                     2148
            F2 = FF*FF                                                     2149
            F4 = F2*F2                                                     2150
            F6 = F2*F4                                                     2151
            G = Z3/Z2                                                      2152
            G2 = G*G                                                       2153
            G4 = G2*G2                                                     2154
            G6 = G2*G4                                                     2155
C           . T'L' = (Z2/Z3)*(((Z1+Z3)/Z2)**L - ((Z1-Z3)/Z2)**L)           2156
            T1 = 2.                                                        2157
            T2 = 4.*FF                                                     2158
            T3 = 2.*G2 + 6.*F2                                             2159
            T4 = 8.*FF*(G2+F2)                                             2160
            T5 = 2.*G4 + 20.*G2*F2 + 10.*F4                                2161
            T6 = 2.*FF*T3*(3.*G2+F2)                                       2162
            DL1 = D*ALOG1                                                  2163
            EL1 = -E*ALOG1                                                 2164
            DL2 = D*ALOG2                                                  2165
            FL3 = FF*ALOG3                                                 2166
```

```
            DO 2 I1=IL,IU                                                    2167
               I2 = I1 + IC                                                  2168
               I3 = I2 + IC                                                  2169
               I4 = I3 + IC                                                  2170
               K = KC(I1)                                                    2171
               L = LC(I2)                                                    2172
C              IF (L.EQ.LL)                                                  2173
                                                    IF(L.NE.LL) GO TO 1      2174
C              THEN                                                          2175
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)                         2176
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)                         2177
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)                         2178
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)                         2179
 1             CONTINUE                                                      2180
 2          CONTINUE                                                         2181
C           . TRANSFORMATION OF TYPE ONE.                                    2182
            X1 = -X1                                                         2183
            XJ =  X2                                                         2184
            X2 = -X3                                                         2185
            X3 =  XJ                                                         2186
            Y1 = -Y1                                                         2187
            XJ =  Y2                                                         2188
            Y2 = -Y3                                                         2189
            Y3 =  XJ                                                         2190
            XJ =  Z2                                                         2191
            Z2 =  Z3                                                         2192
            Z3 = -XJ                                                         2193
            ALOG1 = -ALOG1                                                   2194
            XJ    =  ALOG2                                                   2195
            ALOG2 = -ALOG3                                                   2196
            ALOG3 =  XJ                                                      2197
 3       CONTINUE                                                            2198
C        . TRANSFORMATION OF TYPE TWO.                                       2199
         X3 = -X3                                                            2200
         Y3 = -Y3                                                            2201
         Z3 = -Z3                                                            2202
         ALOG1 = -ALOG1                                                      2203
         ALOG2 = -ALOG2                                                      2204
 4    CONTINUE                                                              2205
      END                                                                   2206
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)                                 2207
C*********************************************************************  2208
C                                                                        2209
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD81 TO EVALUATE        2210
C        C-INTEGRALS.                                                    2211
C                                                                        2212
C*********************************************************************  2213
C                                                                        2214
C                                                                        2215
      COMMON/TEMP/I1,I2,I3,I4,K,L,XY(6),Z1,Z2,Z3,                       2216
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,                 2217
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,                         2218
     *   T1,T2,T3,T4,T5,T6,                                              2219
     *   DL1,EL1,DL2,FL3,SUML,CF(7)                                      2220
C                                                                        2221
      D2 = D*D                                                           2222
      D3 = D*D2                                                          2223
      D4 = D*D3                                                          2224
      E2 = E*E                                                           2225
      E3 = E*E2                                                          2226
      E4 = E*E3                                                          2227
      SUML = 0.                                                          2228
      GO TO (301,302,303),K                                              2229
C            . 1,1                                                       2230
 301        XDNDN = (((240.*E-120.*D+120.)*X3+(((-90.*E2 ))+(60.*D-165.) 2231
     *         *E-30.*D2 +105.*D-135.)*X2+((((-210.*E2 ))+(180.*D-15.)*E 2232
     *         -78.*D2 +21.*D-1.)*X1))*Y3+((((-90.*E2 ))+(60.*D-165.)*E- 2233
     *         30.*D2 +105.*D-135.)*X3+(45.*E3  +90.*E2 +(45.*D2 -120.*D 2234
     *         +120.)*E+30.*D2 -120.*D+150.)*X2+((90.*E3  +(((-45.*D))+1 2235
     *         05.)*E2 +(90.*D2 -150.*D+15.)*E-9.*D3  +39.*D2 -47.*D+5.) 2236
     *         *X1))*Y2+(((((-210.*E2 ))+(180.*D-15.)*E-78.*D2 +21.*D-1. 2237
     *         )*X3+(90.*E3  +(((-45.*D))+105.)*E2 +(90.*D2 -150.*D+15.) 2238
     *         *E-9.*D3  +39.*D2 -47.*D+5.)*X2+((195.*E3  +(((-180.*D))- 2239
     *         60.)*E2 +(207.*D2 +36.*D-46.)*E-36.*D3  -36.*D2 +16.*D+16 2240
     *         .)*X1))*Y1))/180.                                          2241
           CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1 2242
     *         ))*Y3+((((-2.*E3  ))-(2.*E2 ))*X3+(E4  +2.*E3  +E2 )*X2+( 2243
     *         (2.*E4  +(2.*E3  ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2 2244
     *         .*E3  ))*X2+(4.*E4  *X1))*Y1))/16.)                        2245
           CF(2) = (((8.*E2 +((((-8.*D))+4.)*E))*X3+(((-2.*E3  ))+(6.*D 2246
     *         -7.)*E2 +((4.*D-5.)*E))*X2+((((-6.*E3  ))+((12.*D-2.)*E2  2247
     *         ))*X1))*Y3+((((-2.*E3  ))+(6.*D-7.)*E2 +((4.*D-5.)*E))*X3 2248
     *         +((((-4.*D))+4.)*E3  +(((-6.*D))+8.)*E2 +((((-2.*D))+4.)* 2249
     *         E))*X2+((E4  +(((-8.*D))+5.)*E3  +((((-6.*D))+4.)*E2 ))*X 2250
     *         1))*Y2+(((((-6.*E3  ))+((12.*D-2.)*E2 ))*X3+(E4  +(((-8.* 2251
     *         D))+5.)*E3  +((((-6.*D))+4.)*E2 ))*X2+((4.*E4  -(16.*D*E* 2252
     *         *3))*X1))*Y1))/16.                                         2253
           CF(3) = (-(((4.*E2 +(((-16.*D))+8.)*E+4.*D2 -4.*D+1.)*X3+((6 2254
     *         .*D-5.)*E2 +(((-6.*D2 ))+14.*D-7.)*E-2.*D2 +5.*D-2.)*X2+( 2255
     *         (((-2.*E3  ))+(18.*D-1.)*E2 +((((-12.*D2 ))+4.*D+4.)*E))* 2256
     *         X1))*Y3+(((6.*D-5.)*E2 +(((-6.*D2 ))+14.*D-7.)*E-2.*D2 +5 2257
     *         .*D-2.)*X3+((6.*D2 -12.*D+4.)*E2 +(6.*D2 -16.*D+8.)*E+D2  2258
     *         -4.*D+4.)*X2+(((((-4.*D))+1.)*E3  +(12.*D2 -15.*D-1.)*E2  2259
     *         +((6.*D2 -8.*D-2.)*E))*X1))*Y2+(((((-2.*E3  ))+(18.*D-1.) 2260
     *         *E2 +((((-12.*D2 ))+4.*D+4.)*E))*X3+((((-4.*D))+1.)*E3  + 2261
     *         (12.*D2 -15.*D-1.)*E2 +((6.*D2 -8.*D-2.)*E))*X2+((E4  +(( 2262
     *         (-16.*D))-4.)*E3  +((24.*D2 -8.)*E2 ))*X1))*Y1))/16.)      2263
           CF(4) = (-((((8.*D-4.)*E-8.*D2 +8.*D-2.)*X3+((6.*D2 -10.*D+2 2264
     *         .)*E-2.*D3  +7.*D2 -7.*D+2.)*X2+(((((-6.*D))-1.)*E2 +(18. 2265
     *         *D2 -2.*D-5.)*E-4.*D3  +2.*D2 +4.*D-2.)*X1))*Y3+(((6.*D2  2266
     *         -10.*D+2.)*E-2.*D3  +7.*D2 -7.*D+2.)*X3+((4.*D3  -12.*D2  2267
     *         +(8.*D))*E+2.*D3  -8.*D2 +(8.*D))*X2+(((((-6.*D2 ))+3.*D+ 2268
     *         2.)*E2 +(8.*D3  -15.*D2 -2.*D+6.)*E+2.*D3  -4.*D2 -2.*D+4 2269
     *         .)*X1))*Y2+((((((-6.*D))-1.)*E2 +(18.*D2 -2.*D-5.)*E-4.*D 2270
     *         3  +2.*D2 +4.*D-2.)*X3+((((-6.*D2 ))+3.*D+2.)*E2 +(8.*D** 2271
     *         3-15.*D2 -2.*D+6.)*E+2.*D3  -4.*D2 -2.*D+4.)*X2+(((4.*D+2 2272
     *         .)*E3  +(((-24.*D2 ))-12.*D+4.)*E2 +((16.*D3  -(16.*D))*E 2273
     *         ))*X1))*Y1))/16.)                                          2274
```

```
      CF(5) = (-(((4.*D2 -4.*D+1.)*X3+(2.*D3  -5.*D2 +(2.*D))*X2+(          2275
     *   ((((-6.*D2 ))-2.*D+1.)*E+6.*D3  -D2 -5.*D+2.)*X1))*Y3+((2          2276
     *   .*D3  -5.*D2 +(2.*D))*X3+(D4  -4.*D3  +(4.*D2 ))*X2+(((((          2277
     *   -4.*D3  ))+3.*D2 +(4.*D))*E+2.*D4  -5.*D3  -D2 +(6.*D))*X          2278
     *   1))*Y2+((((((-6.*D2 ))-2.*D+1.)*E+6.*D3  -D2 -5.*D+2.)*X3          2279
     *   +((((-4.*D3  ))+3.*D2 +(4.*D))*E+2.*D4  -5.*D3  -D2 +(6.*          2280
     *   D))*X2+(((6.*D2 +6.*D+1.)*E2 +(((-16.*D3  ))-12.*D2 +8.*D          2281
     *   +4.)*E+4.*D4  -8.*D2 +4.)*X1))*Y1))/16.)                           2282
      CF(6) = (D*(D+1.)*((((2.*D)-1.)*X1*Y3)+(D2 -(2.*D))*X1*Y2+((          2283
     *   (2.*D-1.)*X3+(D2 -(2.*D))*X2+(((((-4.*D))-2.)*E+4.*D2 -4.          2284
     *   )*X1))*Y1))))/16.                                                  2285
      CF(7) =         (-(D2 *(D+1.)**2*X1*Y1)/16.)                        2286
      GO TO 350                                                           2287
C     . 2,1                                                               2288
 302  XDNDN = (((240.*E-(120.*D))*X3+(((-90.*E2 ))+(60.*D-135.)*E-          2289
     *   30.*D2 +75.*D-45.)*X2+((((-210.*E2 ))+(180.*D+75.)*E-78.*          2290
     *   D2 -57.*D-19.)*X1))*Y3+((((-90.*E2 ))+(60.*D+135.)*E-30.*          2291
     *   D2 -75.*D-45.)*X3+(45.*E3  +((45.*D2 -90.)*E))*X2+((90.*E          2292
     *   3  +(((-45.*D))-135.)*E2 +(90.*D2 +150.*D+45.)*E-9.*D3  -          2293
     *   57.*D2 -83.*D-15.)*X1))*Y2+(((((-210.*E2 ))+(180.*D-75.)*          2294
     *   E-78.*D2 +57.*D-19.)*X3+(90.*E3  +(((-45.*D))+135.)*E2 +(          2295
     *   90.*D2 -150.*D+45.)*E-9.*D3  +57.*D2 -83.*D+15.)*X2+((195          2296
     *   .*E3  -180.*D*E2 +(207.*D2 -64.)*E-36.*D3  +(16.*D))*X1))          2297
     *   *Y1))/180.                                                         2298
      CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1          2299
     *   ))*Y3+((((-2.*E3  ))+(2.*E2 ))*X3+(E4  -E2 )*X2+((2.*E4           2300
     *   -(2.*E3  ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2.*E3  ))         2301
     *   *X2+(4.*E4  *X1))*Y1))/16.)                                        2302
      CF(2) = (((8.*E2 -(8.*D*E))*X3+(((-2.*E3  ))+(6.*D-5.)*E2 +(          2303
     *   (4.*D-3.)*E))*X2+((((-6.*E3  ))+((12.*D+2.)*E2 ))*X1))*Y3          2304
     *   +((((-2.*E3  ))+(6.*D+5.)*E2 +((((-4.*D))-3.)*E))*X3+(((-          2305
     *   4.*D*E3  ))+(2.*D*E))*X2+((E4  +(((-8.*D))-5.)*E3  +((6.*          2306
     *   D+4.)*E2 ))*X1))*Y2+(((((-6.*E3  ))+((12.*D-2.)*E2 ))*X3+          2307
     *   (E4  +(((-8.*D))+5.)*E3  +((((-6.*D))+4.)*E2 ))*X2+((4.*E          2308
     *   4  -(16.*D*E3  ))*X1))*Y1))/16.                                    2309
      CF(3) = (-(((4.*E2 -16.*D*E+4.*D2 -1.)*X3+((6.*D-3.)*E2 +(((          2310
     *   -6.*D2 ))+10.*D-1.)*E-2.*D2 +3.*D+2.)*X2+((((-2.*E3  ))+(          2311
     *   18.*D+5.)*E2 +((((-12.*D2 ))-4.*D+4.)*E))*X1))*Y3+(((6.*D          2312
     *   +3.)*E2 +(((-6.*D2 ))-10.*D-1.)*E+2.*D2 +3.*D-2.)*X3+((6.          2313
     *   *D2 -4.)*E2 -D2 +4.)*X2+(((((-4.*D))-3.)*E3  +(12.*D2 +15          2314
     *   .*D+1.)*E2 +((((-6.*D2 ))-8.*D+2.)*E))*X1))*Y2+(((((-2.*E          2315
     *   3  ))+(18.*D-5.)*E2 +((((-12.*D2 ))+4.*D+4.)*E))*X3+((((-          2316
     *   4.*D))+3.)*E3  +(12.*D2 -15.*D+1.)*E2 +((6.*D2 -8.*D-2.)*          2317
     *   E))*X2+((E4  -16.*D*E3  +((24.*D2 -8.)*E2 ))*X1))*Y1))/16          2318
     *   .)                                                                 2319
      CF(4) = (-(((8.*D*E-8.*D2 +2.)*X3+((6.*D2 -6.*D-2.)*E-2.*D**          2320
     *   3+5.*D2 -D-2.)*X2+(((((-6.*D))-3.)*E2 +(18.*D2 +10.*D-3.)          2321
     *   *E-4.*D3  -2.*D2 +4.*D+2.)*X1))*Y3+(((6.*D2 +6.*D-2.)*E-2          2322
     *   .*D3  -5.*D2 -D+2.)*X3+(4.*D3  -(8.*D))*E*X2+(((((-6.*D2           2323
     *   ))-9.*D-2.)*E2 +(8.*D3  +15.*D2 +2.*D-2.)*E-2.*D3  -4.*D*          2324
     *   D+2.*D+4.)*X1))*Y2+((((((-6.*D))+3.)*E2 +(18.*D2 -10.*D-3          2325
     *   .)*E-4.*D3  +2.*D2 +4.*D-2.)*X3+((((-6.*D2 ))+9.*D-2.)*E*          2326
     *   E+(8.*D3  -15.*D2 +2.*D+2.)*E+2.*D3  -4.*D2 -2.*D+4.)*X2+          2327
     *   ((4.*D*E3  +(((-24.*D2 ))+4.)*E2 +((16.*D3  -(16.*D))*E))          2328
     *   *X1))*Y1))/16.)                                                    2329
      CF(5) = (-(((4.*D2 -1.)*X3+(2.*D3  -3.*D2 -(2.*D))*X2+(((((-          2330
     *   6.*D2 ))-6.*D-1.)*E+6.*D3  +5.*D2 -3.*D-2.)*X1))*Y3+((2.*          2331
     *   D3  +3.*D2 -(2.*D))*X3+(D4  -(4.*D2 ))*X2+(((((-4.*D3  ))          2332
     *   -9.*D2 -(4.*D))*E+2.*D4  +5.*D3  +D2 -(2.*D))*X1))*Y2+(((          2333
     *   (((-6.*D2 ))+6.*D-1.)*E+6.*D3  -5.*D2 -3.*D+2.)*X3+((((-4          2334
     *   .*D3  ))+9.*D2 -(4.*D))*E+2.*D4  -5.*D3  +D2 +(2.*D))*X2+          2335
     *   (((6.*D2 -1.)*E2 +(8.*D-(16.*D3  ))*E+4.*D4  -8.*D2 +4.)*          2336
     *   X1))*Y1))/16.)                                                     2337
      CF(6) = (D*((((2.*D2 )+3.*D+1.)*X1*Y3)+(D3  +3.*D2 +(2.*D))*          2338
     *   X1*Y2+(((2.*D2 -3.*D+1.)*X3+(D3  -3.*D2 +(2.*D))*X2+(((2-          2339
     *   (4.*D2 ))*E+4.*D3  -(4.*D))*X1))*Y1)))/16.                         2340
      CF(7) = (-((D4  -D2 )*X1*Y1)/16.)                                   2341
      GO TO 350                                                           2342
```

```
C           . 3,1                                                          2343
 303        XDNDN = ((120.*E*X3+(((-90.*E2 ))+(((-60.*D))-15.)*E-30.*D2    2344
     *         +45.*D+15.)*X2+((((-150.*E2 ))+(((-60.*D))+45.)*E-42.*D2    2345
     *         +39.*D+1.)*X1))*Y3+((90.*E2 +(((-60.*D))-15.)*E+30.*D2 -4    2346
     *         5.*D-15.)*X3+(((-45.*E3  ))+((((-45.*D2 ))+120.*D-30.)*E)    2347
     *         )*X2+((((-90.*E3  ))+(45.*D+75.)*E2 +(((-90.*D2 ))+90.*D+    2348
     *         15.)*E+9.*D3  +21.*D2 -73.*D-5.)*X1))*Y2+(((((-150.*E2 ))    2349
     *         +(60.*D-45.)*E-42.*D2 +39.*D+1.)*X3+(90.*E3  +(45.*D+75.)    2350
     *         *E2 +(90.*D2 -90.*D-15.)*E+9.*D3  +21.*D2 -73.*D-5.)*X2+(    2351
     *         (165.*E3  +((153.*D2 -36.*D-74.)*E))*X1))*Y1))/180.          2352
           CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1    2353
     *         ))*Y3+((2.*E3  -(2.*E2 ))*X3+(((-E4  ))+E2 )*X2+((((-2.*E    2354
     *         4  ))+(2.*E3  ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2.*E    2355
     *         3  ))*X2+(4.*E4  *X1))*Y1))/16.)                             2356
           CF(2) = (-(((8.*D-4.)*E*X3+(((-2.*E3  ))+(((-6.*D))+3.)*E2 +    2357
     *         ((((-4.*D))+5.)*E))*X2+((((-2.*E3  ))+((((-12.*D))+2.)*E*    2358
     *         E))*X1))*Y3+((((-2.*E3  ))+(6.*D-3.)*E2 +((((-4.*D))+5.)*    2359
     *         E))*X3+((((-4.*D))+4.)*E3  +((2.*D-4.)*E))*X2+((E4  +(((-    2360
     *         8.*D))+3.)*E3  +((6.*D-4.)*E2 ))*X1))*Y2+(((2.*E3  +((((-    2361
     *         12.*D))+2.)*E2 ))*X3+(E4  +(8.*D-3.)*E3  +((6.*D-4.)*E2 )    2362
     *         )*X2+(16.*D*E3  *X1))*Y1))/16.)                              2363
           CF(3) = (((4.*E2 -4.*D2 +4.*D-1.)*X3+((6.*D-5.)*E2 +(6.*D2 -    2364
     *         6.*D-3.)*E+2.*D2 -5.*D+2.)*X2+((((-2.*E3  ))+(6.*D-3.)*E*    2365
     *         E+((12.*D2 -4.*D-4.)*E))*X1))*Y3+(((6.*D-5.)*E2 +(((-6.*D    2366
     *         *D))+6.*D+3.)*E+2.*D2 -5.*D+2.)*X3+((6.*D2 -12.*D+4.)*E2     2367
     *         -D2 +4.*D-4.)*X2+(((((-4.*D))+1.)*E3  +(12.*D2 -9.*D-3.)*    2368
     *         E2 +((((-6.*D2 ))+8.*D+2.)*E))*X1))*Y2+(((((-2.*E3  ))+((    2369
     *         (-6.*D))+3.)*E2 +((12.*D2 -4.*D-4.)*E))*X3+((((-4.*D))+1.    2370
     *         )*E3  +(((-12.*D2 ))+9.*D+3.)*E2 +((((-6.*D2 ))+8.*D+2.)*    2371
     *         E))*X2+((E4  +((((-24.*D2 ))+8.)*E2 ))*X1))*Y1))/16.         2372
           CF(4) = (((8.*D-4.)*E*X3+((6.*D2 -10.*D+2.)*E+2.*D3  -3.*D2    2373
     *         -3.*D+2.)*X2+(((((-6.*D))-1.)*E2 +(6.*D2 -6.*D-3.)*E+4.*D    2374
     *         3  -2.*D2 -4.*D+2.)*X1))*Y3+(((6.*D2 -10.*D+2.)*E-2.*D3      2375
     *         +3.*D2 +3.*D-2.)*X3+(4.*D3  -12.*D2 +(8.*D))*E*X2+(((((-6    2376
     *         .*D2 ))+3.*D+2.)*E2 +(8.*D3  -9.*D2 -6.*D+2.)*E-2.*D3  +4    2377
     *         .*D2 +2.*D-4.)*X1))*Y2+((((((-6.*D))-1.)*E2 +(((-6.*D2 ))    2378
     *         +6.*D+3.)*E+4.*D3  -2.*D2 -4.*D+2.)*X3+((((-6.*D2 ))+3.*D    2379
     *         +2.)*E2 +(((-8.*D3  ))+9.*D2 +6.*D-2.)*E-2.*D3  +4.*D2 +2    2380
     *         .*D-4.)*X2+(((4.*D+2.)*E3  +((((-16.*D3  ))+(16.*D))*E))*    2381
     *         X1))*Y1))/16.                                                2382
           CF(5) = (((4.*D2 -4.*D+1.)*X3+(2.*D3  -5.*D2 +(2.*D))*X2+(((    2383
     *         ((-6.*D2 ))-2.*D+1.)*E+2.*D3  -3.*D2 -3.*D+2.)*X1))*Y3+((    2384
     *         2.*D3  -5.*D2 +(2.*D))*X3+(D4  -4.*D3  +(4.*D2 ))*X2+((((    2385
     *         (-4.*D3  ))+3.*D2 +(4.*D))*E+2.*D4  -3.*D3  -3.*D2 +(2.*D    2386
     *         ))*X1))*Y2+((((((-6.*D2 ))-2.*D+1.)*E-2.*D3  +3.*D2 +3.*D    2387
     *         -2.)*X3+((((-4.*D3  ))+3.*D2 +(4.*D))*E-2.*D4  +3.*D3  +3    2388
     *         .*D2 -(2.*D))*X2+(((6.*D2 +6.*D+1.)*E2 -4.*D4  +8.*D2 -4.    2389
     *         )*X1))*Y1))/16.                                              2390
           CF(6) = (-(D*(D+1.)*((((2.*D)-1.)*X1*Y3)+(D2 -(2.*D))*X1*Y2+    2391
     *         (((2.*D-1.)*X3+(D2 -(2.*D))*X2+((((-4.*D))-2.)*E*X1))*Y1)    2392
     *         ))/16.)                                                      2393
           CF(7) = (D2 *(D+1.)**2*X1*Y1)/16.                              2394
C     . END OF COMPUTED GO TO.                                             2395
 350   CONTINUE                                                           2396
       SUML = CF(7)*(2.*(G6+5.*G4*F2+3.*G2*F4+F6/7.)*EL1 + 2./7.*DL2      2397
     *       +2.*(F6+5.*F4*G2+3.*F2*G4+G6/7.)*ALOG3                       2398
     *       -(30.*T6+10.*T4+6.*T2)/105.)                                 2399
     *    + CF(6)*(((F6+G6-1.)/3.+5.*F2*G2*(F2+G2))*DL1                   2400
     *       +2.*(F4+10.*F2*G2/3.+G4)*FL3 -(15.*T5+5.*T3+3.*T1)/45.)      2401
     *    + CF(5)*(2.*(G4+2.*F2*G2+F4/5.)*EL1 +2./5.*DL2                  2402
     *       +2.*(F4+2.*F2*G2+G4/5.)*ALOG3 -(6.*T4+2.*T2)/15.)            2403
     *    + CF(4)*(((F4+G4-1.)/2.+3.*F2*G2)*DL1 +2.*(F2+G2)*FL3           2404
     *       -(3.*T3+T1)/6.)                                              2405
     *    + CF(3)*(2./3.)*((F2+3.*G2)*EL1 +DL2 +(3.*F2+G2)*ALOG3 -T2)    2406
     *    + CF(2)*((F2+G2-1.)*DL1 +2.*FL3 -T1)                           2407
     *    + CF(1)*2.*(EL1+DL2+ALOG3)                                     2408
       XDNDN = -XDNDN/Z3 + SUML/Z2                                        2409
       RETURN                                                           2410
       END                                                              2411
```

```
      OVERLAY(MAIN,10,0)                                                    2412
      PROGRAM QUAD82                                                        2413
C*************************************************************************2414
C                                                                           2415
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE          2416
C        ELEMENTS WITH NNE = 8.                                             2417
C                                                                           2418
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT         2419
C        GROUP TRANSFORMATIONS.                                             2420
C                                                                           2421
C*************************************************************************2422
C                                                                           2423
      DIMENSION LC(1),KC(1)                                                 2424
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,SKP(2),                           2425
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,ALG1,ALG2,ALG3,OTHERS(1)       2426
      COMMON/TEMP/I1,I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,               2427
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,                    2428
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,                            2429
     *   T1,T2,T3,T4,T5,T6,                                                 2430
     *   DL1,EL1,DL2,FL3,XJ                                                 2431
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))                               2432
C                                                                           2433
C                                                                           2434
      X1 = XX1                                                              2435
      X2 = XX2                                                              2436
      X3 = XX3                                                              2437
      Y1 = YY1                                                              2438
      Y2 = YY2                                                              2439
      Y3 = YY3                                                              2440
      Z1 = ZZ1                                                              2441
      Z2 = ZZ2                                                              2442
      Z3 = ZZ3                                                              2443
      ALOG1 = ALG1                                                          2444
      ALOG2 = ALG2                                                          2445
      ALOG3 = ALG3                                                          2446
      IL = IXC + 11                                                         2447
      IU = IXC + 36                                                         2448
      LL = 0                                                                2449
      DO 4 I=1,2                                                            2450
         DO 3 J=1,4                                                         2451
            LL = LL + 1                                                     2452
            D = Z2/Z3                                                       2453
            DP1 = D + 1.                                                    2454
            DM1 = D - 1.                                                    2455
            D2 = D*D                                                        2456
            D3 = D*D2                                                       2457
            E = -Z1/Z3                                                      2458
            EP1 = E + 1.                                                    2459
            EM1 = E - 1.                                                    2460
            E2 = E*E                                                        2461
            E3 = E*E2                                                       2462
            FF = Z1/Z2                                                      2463
            F2 = FF*FF                                                      2464
            F4 = F2*F2                                                      2465
            F6 = F2*F4                                                      2466
            G = Z3/Z2                                                       2467
            G2 = G*G                                                        2468
            G4 = G2*G2                                                      2469
            G6 = G2*G4                                                      2470
C           . T'L' = (Z2/Z3)*(((Z1+Z3)/Z2)**L - ((Z1-Z3)/Z2)**L)            2471
            T1 = 2.                                                         2472
            T2 = 4.*FF                                                      2473
            T3 = 2.*G2 + 6.*F2                                              2474
            T4 = 8.*FF*(G2+F2)                                              2475
            T5 = 2.*G4 + 20.*G2*F2 + 10.*F4                                 2476
            T6 = 2.*FF*T3*(3.*G2+F2)                                        2477
            DL1 = D*ALOG1                                                   2478
            EL1 = -E*ALOG1                                                  2479
            DL2 = D*ALOG2                                                   2480
            FL3 = FF*ALOG3                                                  2481
```

```
            DO 2 I1=IL,IU                                                    2482
               I2 = I1 + IC                                                  2483
               I3 = I2 + IC                                                  2484
               I4 = I3 + IC                                                  2485
               K = KC(I1) - 3                                                2486
               L = LC(I2)                                                    2487
C              IF (L.EQ.LL)                                                  2488
                                                  IF(L.NE.LL) GO TO 1        2489
C              THEN                                                          2490
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)                         2491
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)                         2492
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)                         2493
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)                         2494
 1             CONTINUE                                                      2495
 2          CONTINUE                                                         2496
C           . TRANSFORMATION OF TYPE ONE.                                    2497
            X1 = -X1                                                         2498
            XJ =  X2                                                         2499
            X2 = -X3                                                         2500
            X3 =  XJ                                                         2501
            Y1 = -Y1                                                         2502
            XJ =  Y2                                                         2503
            Y2 = -Y3                                                         2504
            Y3 =  XJ                                                         2505
            XJ =  Z2                                                         2506
            Z2 =  Z3                                                         2507
            Z3 = -XJ                                                         2508
            ALOG1 = -ALOG1                                                   2509
            XJ    =  ALOG2                                                   2510
            ALOG2 = -ALOG3                                                   2511
            ALOG3 =  XJ                                                      2512
 3       CONTINUE                                                            2513
C        . TRANSFORMATION OF TYPE TWO.                                       2514
         X3 = -X3                                                            2515
         Y3 = -Y3                                                            2516
         Z3 = -Z3                                                            2517
         ALOG1 = -ALOG1                                                      2518
         ALOG2 = -ALOG2                                                      2519
 4    CONTINUE                                                              2520
      END                                                                    2521
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)                                 2522
C*************************************************************************2523
C                                                                         2524
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD82 TO EVALUATE         2525
C        C-INTEGRALS.                                                     2526
C                                                                         2527
C*************************************************************************2528
C                                                                         2529
C                                                                         2530
      COMMON/TEMP/I1,I2,I3,I4,K,L,XY(6),Z1,Z2,Z3,                        2531
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,                  2532
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,                          2533
     *   T1,T2,T3,T4,T5,T6,                                               2534
     *   DL1,EL1,DL2,FL3,SUML,CF(7)                                       2535
C                                                                         2536
      D2 = D*D                                                           2537
      D3 = D*D2                                                          2538
      D4 = D*D3                                                          2539
      E2 = E*E                                                           2540
      E3 = E*E2                                                          2541
      E4 = E*E3                                                          2542
      SUML = 0.                                                          2543
      GO TO (304,305,306,307,308),K                                      2544
C           .5,1                                                        2545
 304      XDNDN = (-(((240.*E-120.*D+60.)*X3+(((-90.*E2 ))+(60.*D-150.  2546
     *        )*E-30.*D2 +90.*D-90.)*X2+((((-210.*E2 ))+(180.*D+30.)*E-  2547
     *        78.*D2 -18.*D-10.)*X1))*Y3+((((-90.*E2 ))+(60.*D-15.)*E-3  2548
     *        0.*D2 +15.*D+60.)*X3+(45.*E3  +45.*E2 +(45.*D2 -60.*D-30.  2549
     *        )*E+15.*D2 -30.*D-30.)*X2+((90.*E3  +(((-45.*D))-15.)*E2   2550
     *        +(90.*D2 -90.)*E-9.*D3  -9.*D2 +10.*D+10.)*X1))*Y2+(((((-  2551
     *        210.*E2 ))+(180.*D-45.)*E-78.*D2 +39.*D-100.)*X3+(90.*E**  2552
     *        3+(((-45.*D))+120.)*E2 +(90.*D2 -150.*D+60.)*E-9.*D3  +48  2553
     *        .*D2 -80.*D+70.)*X2+((195.*E3  +(((-180.*D))-30.)*E2 +(20  2554
     *        7.*D2 +18.*D+20.)*E-36.*D3  -18.*D2 -32.*D-10.)*X1))*Y1))  2555
     *        /90.)                                                      2556
          CF(1) = ((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1))  2557
     *        *Y3+((((-2.*E3  ))+(2.*E))*X3+(E4  +E3  -E2 -E)*X2+((2.*E  2558
     *        4  -(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2.*E3   2559
     *        ))*X2+(4.*E4  *X1)))*Y1))/8.                               2560
          CF(2) = (-(((8.*E2 +((((-8.*D))+2.)*E))*X3+(((-2.*E3  ))+(6.  2561
     *        *D-6.)*E2 +((4.*D-4.)*E))*X2+((((-6.*E3  ))+(12.*D*E2 ))*  2562
     *        X1))*Y3+((((-2.*E3  ))+(6.*D-1.)*E2 +2.*E-2.*D+1.)*X3+(((  2563
     *        (-4.*D))+2.)*E3  +(((-3.*D))+2.)*E2 +(2.*D-2.)*E+D-2.)*X2  2564
     *        +((E4  -8.*D*E3  -E2 +(4.*D*E))*X1))*Y2+(((((-6.*E3  ))+(  2565
     *        12.*D-2.)*E2 -(2.*E))*X3+(E4  +(((-8.*D))+5.)*E3  +(((-6.  2566
     *        *D))+5.)*E2 +E)*X2+((4.*E4  -16.*D*E3  +(2.*E2 ))*X1))*Y1  2567
     *        ))/8.)                                                     2568
          CF(3) = (((4.*E2 +(((-16.*D))+4.)*E+4.*D2 -(2.*D))*X3+((6.*D  2569
     *        -4.)*E2 +(((-6.*D2 ))+12.*D-4.)*E-2.*D2 +(4.*D))*X2+((((-  2570
     *        2.*E3  ))+(18.*D+2.)*E2 +((((-12.*D2 ))+4.)*E))*X1))*Y3+(  2571
     *        ((6.*D-1.)*E2 +(((-6.*D2 ))+(2.*D))*E-2.*D+1.)*X3+((6.*D*  2572
     *        D-(6.*D))*E2 +(3.*D2 -(4.*D))*E-D2 +(2.*D))*X2+(((((-4.*D  2573
     *        ))-1.)*E3  +(12.*D2 -2.)*E2 +(2.*D+1.)*E-2.*D2 +2.)*X1))*  2574
     *        Y2+((((-2.*E3  ))+(18.*D-3.)*E2 +(((-12.*D2 ))+4.*D-2.)*  2575
     *        E+2.*D-1.)*X3+((((-4.*D))+2.)*E3  +(12.*D2 -15.*D+2.)*E2   2576
     *        +(6.*D2 -10.*D+2.)*E-D+2.)*X2+((E4  +(((-16.*D))-2.)*E3   2577
     *        +(24.*D2 -3.)*E2 -(4.*D*E))*X1))*Y1))/8.                   2578
          CF(4) = ((((8.*D-2.)*E-8.*D2 +(4.*D))*X3+((6.*D2 -(8.*D))*E-  2579
     *        2.*D3  +6.*D2 -(4.*D))*X2+(((((-6.*D))-2.)*E2 +(18.*D2 +4  2580
     *        .*D-4.)*E-4.*D3  +(4.*D))*X1))*Y3+(((6.*D2 -(2.*D))*E-2.*  2581
     *        D3  +D2 )*X3+((4.*D3  -(6.*D2 ))*E+D3  -(2.*D2 ))*X2+((((  2582
     *        (-6.*D2 ))-(3.*D))*E2 +(8.*D3  -(4.*D))*E+D2 +D)*X1))*Y2+  2583
     *        ((((((-6.*D))+1.)*E2 +(18.*D2 -(6.*D))*E-4.*D3  +2.*D2 -2  2584
     *        .*D+1.)*X3+((((-6.*D2 ))+(6.*D))*E2 +(8.*D3  -15.*D2 +(4.  2585
     *       *D))*E+2.*D3  -5.*D2 +(2.*D))*X2+(((4.*D+1.)*E3  +(((-24.   2586
     *        *D2 ))-6.*D+2.)*E2 +(16.*D3  -6.*D+1.)*E-2.*D2 +2.)*X1))*  2587
     *        Y1))/8.                                                    2588
```

```
      CF(5) = (D*((((((4.*D)-2.)*X3)+(2.*D2 -(4.*D))*X2+(((((-6.*D          2589
     *     ))-4.)*E+6.*D2 +2.*D-4.)*X1))*Y3)+((2.*D2 -D)*X3+(D3  -(2       2590
     *     .*D2 ))*X2+(((((-4.*D2 ))-(3.*D))*E+2.*D3  -(2.*D))*X1))*       2591
     *     Y2+((((2-(6.*D))*E+6.*D2 -(3.*D))*X3+((6.*D-(4.*D2 ))*E+2       2592
     *     .*D3  -5.*D2 +(2.*D))*X2+(((6.*D+3.)*E2 +(((-16.*D2 ))-6.       2593
     *     *D+4.)*E+4.*D3  -3.*D+1.)*X1))*Y1))))/8.                         2594
      CF(6) = (-(D2 *((((2.*D)+2.)*X1*Y3)+(D2 +D)*X1*Y2+(((2.*D-1.          2595
     *     )*X3+(D2 -(2.*D))*X2+(((((-4.*D))-3.)*E+4.*D2 +2.*D-2.)*X       2596
     *     1))*Y1))))/8.)                                                   2597
      CF(7) =   ((D+1.)*D3  *X1*Y1)/8.                                      2598
      GO TO 350                                                             2599
C     . 5,3                                                                 2600
 305  XDNDN = (-(((120.*E*X3+(90.*E2 +(((-60.*D))-30.)*E+30.*D2 -30         2601
     *     .*D-30.)*X2+((((-150.*E2 ))+(60.*D-30.)*E-42.*D2 +18.*D+1       2602
     *     0.)*X1))*Y3+((((-90.*E2 ))+(((-60.*D))+15.)*E-30.*D2 +15.       2603
     *     *D+60.)*X3+(((-45.*E3  ))+45.*E2 +(((-45.*D2 ))+60.*D+30.       2604
     *     )*E+15.*D2 -30.*D-30.)*X2+((90.*E3  +(45.*D+15.)*E2 +(90.       2605
     *     *D2 -90.)*E+9.*D3  +9.*D2 -10.*D-10.)*X1))*Y2+(((((-150.*       2606
     *     E2 ))+(((-60.*D))+15.)*E-42.*D2 +21.*D-20.)*X3+(((-90.*E*       2607
     *     *3))+(45.*D+60.)*E2 +(((-90.*D2 ))+(90.*D))*E+9.*D3  +12.       2608
     *     *D2 -40.*D-10.)*X2+((165.*E3  +30.*E2 +(153.*D2 -18.*D-20       2609
     *     .)*E+18.*D2 -12.*D+10.)*X1))*Y1))/90.)                           2610
      CF(1) = ((4.*E2 *X3+(2.*E3  -(2.*E2 ))*X2-(4.*E3  *X1))*Y3+(          2611
     *     (((-2.*E3  ))+(2.*E))*X3+(((-E4  ))+E3  +E2 -E)*X2+((2.*E       2612
     *     4  -(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(((-2.*E4  ))+(2.       2613
     *     *E3  ))*X2+(4.*E4  *X1))*Y1))/8.                                 2614
      CF(2) = (((8.*D-2.)*E*X3+(((-2.*E3  ))+(6.*D-2.)*E2 +((((-4.          2615
     *     *D))+4.)*E))*X2+((2.*E3  -(12.*D*E2 ))*X1))*Y3+((((-2.*E*       2616
     *     *3))+(((-6.*D))+1.)*E2 +2.*E+2.*D-1.)*X3+((((-4.*D))+2.)*       2617
     *     E3  +(3.*D-2.)*E2 +(2.*D-2.)*E-D+2.)*X2+((E4  +8.*D*E3  -       2618
     *     E2 -(4.*D*E))*X1))*Y2+(((((-2.*E3  ))+(((-12.*D))+2.)*E2        2619
     *     +(2.*E))*X3+(E4  +(((-8.*D))+3.)*E3  +(6.*D-3.)*E2 -E)*X2       2620
     *     +((16.*D*E3  -(2.*E2 ))*X1))*Y1))/8.                             2621
      CF(3) = (-(((4.*E2 -4.*D2 +(2.*D))*X3+((6.*D-4.)*E2 +(((-6.*          2622
     *     D2 ))+4.*D+4.)*E+2.*D2 -(4.*D))*X2+((((-2.*E3  ))+(((-6.*       2623
     *     D))+2.)*E2 +((12.*D2 -4.)*E))*X1))*Y3+(((6.*D-1.)*E2 +(6.       2624
     *     *D2 -(2.*D))*E-2.*D+1.)*X3+((6.*D2 -(6.*D))*E2 +(((-3.*D*       2625
     *     D))+(4.*D))*E-D2 +(2.*D))*X2+(((((-4.*D))-1.)*E3  +(((-12       2626
     *     .*D2 ))+2.)*E2 +(2.*D+1.)*E+2.*D2 -2.)*X1))*Y2+(((((-2.*E       2627
     *     3  ))+(6.*D-1.)*E2 +(12.*D2 -4.*D-2.)*E-2.*D+1.)*X3+((((-       2628
     *     4.*D))+2.)*E3  +(12.*D2 -9.*D-2.)*E2 +(((-6.*D2 ))+6.*D+2       2629
     *     .)*E+D-2.)*X2+((E4  -2.*E3  +(((-24.*D2 ))+5.)*E2 +(4.*D*       2630
     *     E))*X1))*Y1))/8.)                                                2631
      CF(4) = (-(((8.*D-2.)*E*X3+((6.*D2 -(8.*D))*E-2.*D3  +2.*D2           2632
     *     +(4.*D))*X2+(((((-6.*D))-2.)*E2 +(((-6.*D2 ))+4.*D+4.)*E+       2633
     *     4.*D3  -(4.*D))*X1))*Y3+(((6.*D2 -(2.*D))*E+2.*D3  -D2 )*       2634
     *     X3+((4.*D3  -(6.*D2 ))*E-D3  +(2.*D2 ))*X2+(((((-6.*D2 ))       2635
     *     -(3.*D))*E2 +(((-8.*D3  ))+(4.*D))*E+D2 +D)*X1))*Y2+((((       2636
     *     (-6.*D))+1.)*E2 +(6.*D2 -(2.*D))*E+4.*D3  -2.*D2 -2.*D+1.       2637
     *     )*X3+((((-6.*D2 ))+(6.*D))*E2 +(8.*D3  -9.*D2 -(4.*D))*E-       2638
     *     2.*D3  +3.*D2 +(2.*D))*X2+(((4.*D+1.)*E3  +(((-6.*D))-2.)       2639
     *     *E2 +(((-16.*D3  ))+10.*D+1.)*E+2.*D2 -2.)*X1))*Y1))/8.)         2640
      CF(5) = (-(D*((((((4.*D)-2.)*X3)+(2.*D2 -(4.*D))*X2+(((((-6.          2641
     *     *D))-4.)*E-2.*D2 +2.*D+4.)*X1))*Y3)+((2.*D2 -D)*X3+(D3  -       2642
     *     (2.*D2 ))*X2+(((((-4.*D2 ))-(3.*D))*E-2.*D3  +(2.*D))*X1)       2643
     *     )*Y2+((((2-(6.*D))*E+2.*D2 -D)*X3+((6.*D-(4.*D2 ))*E+2.*D       2644
     *     3  -3.*D2 -(2.*D))*X2+(((6.*D+3.)*E2 +(((-6.*D))-4.)*E-4.       2645
     *     *D3  +5.*D+1.)*X1))*Y1))))/8.)                                   2646
      CF(6) = (D2 *((((2.*D)+2.)*X1*Y3)+(D2 +D)*X1*Y2+(((2.*D-1.)*          2647
     *     X3+(D2 -(2.*D))*X2+(((((-4.*D))-3.)*E+2.*D+2.)*X1))*Y1)))       2648
     *     /8.                                                              2649
      CF(7) =  (-((D+1.)*D3  *X1*Y1)/8.)                                    2650
      GO TO 350                                                             2651
C     . 5,5                                                                 2652
 306  XDNDN = (((240.*E-(120.*D))*X3+(((-90.*E2 ))+60.*D*E-30.*D2           2653
     *     +60.)*X2+((((-210.*E2 ))+180.*D*E-78.*D2 -100.)*X1))*Y3+(       2654
     *     (((-90.*E2 ))+60.*D*E-30.*D2 +60.)*X3+(45.*E3  +((45.*D2        2655
     *     -75.)*E))*X2+((90.*E3  -45.*D*E2 +(90.*D2 -60.)*E-9.*D3          2656
     *     -(5.*D))*X1))*Y2+(((((-210.*E2 ))+180.*D*E-78.*D2 -100.)*        2657
```

```
     *          X3+(90.*E3  -45.*D*E2 +(90.*D2 -60.)*E-9.*D3  -(5.*D))*X2      2658
     *          +((195.*E3  -180.*D*E2 +(207.*D2 +95.)*E-36.*D3  -(80.*D)      2659
     *          )*X1))*Y1))/45.                                                2660
               CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))+(2.*E))*X2-(4.*E3  *X1))   2661
     *          *Y3+((((-2.*E3  ))+(2.*E))*X3+(E4  -2.*E2 +1.)*X2+((2.*E*     2662
     *          *4-(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  -(2.*E2 ))     2663
     *          *X2+(4.*E4  *X1))*Y1))/4.)                                     2664
               CF(2) = (((8.*E2 -(8.*D*E))*X3+(((-2.*E3  ))+6.*D*E2 +2.*E-(   2665
     *          2.*D))*X2+((((-6.*E3  ))+12.*D*E2 -(2.*E))*X1))*Y3+((((-2      2666
     *          .*E3  ))+6.*D*E2 +2.*E-(2.*D))*X3+(((-4.*D*E3  ))+(4.*D*E      2667
     *          ))*X2+((E4  -8.*D*E3  +4.*D*E-1.)*X1))*Y2+(((((-6.*E3  ))      2668
     *          +12.*D*E2 -(2.*E))*X3+(E4  -8.*D*E3  +4.*D*E-1.)*X2+((4.*      2669
     *          E4  -16.*D*E3  +(4.*E2 ))*X1))*Y1))/4.                         2670
               CF(3) = (-(((4.*E2 -16.*D*E+(4.*D2 ))*X3+(6.*D*E2 -6.*D2 *E-   2671
     *          (2.*D))*X2+((((-2.*E3  ))+18.*D*E2 +(((-12.*D2 ))-2.)*E+(      2672
     *          2.*D))*X1))*Y3+((6.*D*E2 -6.*D2 *E-(2.*D))*X3+(6.*D2 *E2       2673
     *          -(2.*D2 ))*X2+((((-4.*D*E3  ))+12.*D2 *E2 -(2.*D2 ))*X1))      2674
     *          *Y2+(((((-2.*E3  ))+18.*D*E2 +(((-12.*D2 ))-2.)*E+(2.*D))      2675
     *          *X3+(((-4.*D*E3  ))+12.*D2 *E2 -(2.*D2 ))*X2+((E4  -16.*D      2676
     *          *E3  +(24.*D2 +2.)*E2 -8.*D*E+1.)*X1))*Y1))/4.)                2677
               CF(4) = (-(((4.*D*E-(4.*D2 ))*X3+(3.*D2 *E-D3  )*X2+((((-3.*   2678
     *          D*E2 ))+9.*D2 *E-2.*D3  -D)*X1))*Y3+((3.*D2 *E-03  )*X3+2      2679
     *          .*D3  *E*X2+((((-3.*D2 *E2 ))+(4.*D3  *E))*X1))*Y2+(((((-      2680
     *          3.*D*E2 ))+9.*D2 *E-2.*D3  -D)*X3+(((-3.*D2 *E2 ))+(4.*D*      2681
     *          *3*E))*X2+((2.*D*E3  -12.*D2 *E2 +(8.*D3  +(2.*D))*E-(2.*      2682
     *          D2 ))*X1))*Y1))/2.)                                            2683
               CF(5) = (-(D2 *((((4.*X3)+2.*D*X2+((6.*D-(6.*E))*X1))*Y3)+(2   2684
     *          .*D*X3+D2 *X2+((2.*D2 -(4.*D*E))*X1))*Y2+(((6.*D-(6.*E))*      2685
     *          X3+(2.*D2 -(4.*D*E))*X2+((6.*E2 -16.*D*E+4.*D2 +2.)*X1))*      2686
     *          Y1))))/4.)                                                     2687
               CF(6) =      (D3  *((2.*X1*Y3)+D*X1*Y2+((2.*X3+D*X2+((4.*D-(   2688
     *          4.*E))*X1))*Y1)))/4.                                           2689
               CF(7) =     (-(D4  *X1*Y1)/4.)                                 2690
               GO TO 350                                                      2691
C              . 6,5                                                          2692
 307           XDNDN = ((60.*X3-30.*E*X2+((((-60.*E))+(6.*D))*X1))*Y3+((60.   2693
     *          *E-60.*D-60.)*X3+(60.*D*E-(30.*D))*X2+((((-30.*E2 ))+(120      2694
     *          .*D+30.)*E-18.*D2 -60.*D-40.)*X1))*Y2+(((((-120.*E))+48.*      2695
     *          D+60.)*X3+(60.*E2 -30.*E+24.*D2 -40.)*X2+((120.*E2 +(((-4      2696
     *          8.*D))-60.)*E+48.*D2 +18.*D+40.)*X1))*Y1))/45.                 2697
               CF(1) = (-((2.*E*X3+(((-E2 ))+1.)*X2-(2.*E2 *X1))*Y3+((((-2.   2698
     *          *E2 *X3))+(E3  -E)*X2+(2.*E3  *X1))*Y1))/4.)                   2699
               CF(2) = (((2.*E-(2.*D))*X3+2.*D*E*X2+((((-E2 ))+4.*D*E-1.)*X   2700
     *          1))*Y3+((4.*E2 -(4.*E))*X3+(((-2.*E3  ))+2.*E2 +2.*E-2.)*      2701
     *          X2+((((-4.*E3  ))+(4.*E2 ))*X1))*Y2+(((((-2.*E2 ))+(4.*D*      2702
     *          E))*X3+(((-3.*D*E2 ))+D)*X2+((E3  -6.*D*E2 +E)*X1))*Y1))/      2703
     *          4.                                                             2704
               CF(3) = (((2.*E+(2.*D))*X3+(((-E2 ))+D2 +1.)*X2+((((-2.*E2 )   2705
     *          )-2.*D*E+(2.*D2 ))*X1))*Y3+((((-4.*E2 ))+(8.*D+4.)*E-(4.*      2706
     *          D))*X3+(((-6.*D*E2 ))+4.*D*E+(2.*D))*X2+((2.*E3  +(((-12.      2707
     *          *D))-2.)*E2 +(8.*D+2.)*E-2.)*X1))*Y2+(((2.*E2 +(((-4.*D))      2708
     *          -4.)*E+(2.*D2 ))*X3+(((-E3  ))+2.*E2 +(((-3.*D2 ))+1.)*E-      2709
     *          2.)*X2+((((-2.*E3  ))+(3.*D+4.)*E2 -6.*D2 *E+D)*X1))*Y1))      2710
     *          /4.                                                            2711
               CF(4) = (-(((2.*E-(2.*D))*X3+2.*D*E*X2+((((-E2 ))+4.*D*E+D2    2712
     *          -1.)*X1))*Y3+((8.*D*E-4.*D2 -(4.*D))*X3+(6.*D2 *E-(2.*D2       2713
     *          ))*X2+((((-6.*D*E2 ))+(12.*D2 +(4.*D))*E-4.*D2 -(2.*D))*X      2714
     *          1))*Y2+(((2.*E2 +(((-4.*D))-4.)*E+2.*D2 +(4.*D))*X3+(3.*D      2715
     *          *E2 -4.*D*E+D3  -D)*X2+((((-E3  ))+(6.*D+2.)*E2 +(((-3.*D      2716
     *          *D))-8.*D-1.)*E+2.*D3  +2.)*X1))*Y1))/4.)                      2717
               CF(5) = (-(D*((((2.*X3)+D*X2+((2.*D-(2.*E))*X1))*Y3)+(4.*D*X   2718
     *          3+2.*D2 *X2+((((-6.*D*E))+4.*D2 +(2.*D))*X1))*Y2+(((4.*E-      2719
     *          2.*D-4.)*X3+(3.*D*E-(2.*D))*X2+((((-3.*E2 ))+(6.*D+4.)*E-      2720
     *          D2 -4.*D-1.)*X1))*Y1))))/4.)                                   2721
               CF(6) =    (D2 *((X1*Y3)+2.*D*X1*Y2+((((-2.*X3))-D*X2+((3.*E   2722
     *          -2.*D-2.)*X1))*Y1))))/4.                                       2723
               CF(7) =      (D3  *X1*Y1)/4.                                   2724
               GO TO 350                                                      2725
```

```
C           . 7,5                                                          2726
 308        XDNDN = ((120.*E*X3+(((-90.*E2 ))-60.*D*E-30.*D2 +60.)*X2+((   2727
     *         ((-150.*E2 ))-60.*D*E-42.*D2 -20.)*X1))*Y3+((90.*E2 -60.*   2728
     *         D*E+30.*D2 -60.)*X3+(((-45.*E3  ))+((((-45.*D2 ))+75.)*E)   2729
     *         )*X2+((((-90.*E3  ))+45.*D*E2 +(((-90.*D2 ))+60.)*E+9.*D*   2730
     *         *3+(5.*D))*X1))*Y2+(((((-150.*E2 ))+60.*D*E-42.*D2 -20.)*   2731
     *         X3+(90.*E3  +45.*D*E2 +(90.*D2 -60.)*E+9.*D3  +(5.*D))*X2   2732
     *         +((165.*E3  +((153.*D2 +25.)*E))*X1))*Y1))/45.              2733
           CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))+(2.*E))*X2-(4.*E3  *X1))   2734
     *         *Y3+((2.*E3  -(2.*E))*X3+(((-E4  ))+2.*E2 -1.)*X2+((((-2.   2735
     *         *E4  ))+(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  -(2.*   2736
     *         E2 ))*X2+(4.*E4  *X1))*Y1))/4.)                             2737
           CF(2) = (-((8.*D*E*X3+(((-2.*E3  ))-6.*D*E2 +2.*E+(2.*D))*X2   2738
     *         +((((-2.*E3  ))-12.*D*E2 +(2.*E))*X1))*Y3+((((-2.*E3  ))+   2739
     *         6.*D*E2 +2.*E-(2.*D))*X3+(((-4.*D*E3  ))+(4.*D*E))*X2+((E   2740
     *         4  -8.*D*E3  +4.*D*E-1.)*X1))*Y2+(((2.*E3  -12.*D*E2 -(2.   2741
     *         *E))*X3+(E4  +8.*D*E3  -4.*D*E-1.)*X2+(16.*D*E3  *X1))*Y1   2742
     *         ))/4.)                                                      2743
           CF(3) = (((4.*E2 -(4.*D2 ))*X3+(6.*D*E2 +6.*D2 *E-(2.*D))*X2   2744
     *         +((((-2.*E3  ))+6.*D*E2 +(12.*D2 -2.)*E-(2.*D))*X1))*Y3+(   2745
     *         (6.*D*E2 -6.*D2 *E-(2.*D))*X3+(6.*D2 *E2 -(2.*D2 ))*X2+((   2746
     *         ((-4.*D*E3  ))+12.*D2 *E2 -(2.*D2 ))*X1))*Y2+(((((-2.*E**   2747
     *         3))-6.*D*E2 +(12.*D2 -2.)*E+(2.*D))*X3+(((-4.*D*E3  ))-12   2748
     *         .*D2 *E2 +(2.*D2 ))*X2+((E4  +(((-24.*D2 ))+2.)*E2 +1.)*X   2749
     *         1))*Y1))/4.                                                 2750
           CF(4) = ((4.*D*E*X3+(3.*D2 *E+D3  )*X2+((((-3.*D*E2 ))+3.*D*   2751
     *         D*E+2.*D3  -D)*X1))*Y3+((3.*D2 *E-D3  )*X3+2.*D3  *E*X2+(   2752
     *         (((-3.*D2 *E2 ))+(4.*D3  *E))*X1))*Y2+(((((-3.*D*E2 ))-3.   2753
     *         *D2 *E+2.*D3  -D)*X3+(((-3.*D2 *E2 ))-(4.*D3  *E))*X2+((2   2754
     *         .*D*E3  +((((-8.*D3  ))+(2.*D))*E))*X1))*Y1))/2.            2755
           CF(5) = (D2 *((((4.*X3)+2.*D*X2+((2.*D-(6.*E))*X1))*Y3)+(2.*   2756
     *         D*X3+D2 *X2+((2.*D2 -(4.*D*E))*X1))*Y2+(((((-6.*E))-(2.*D   2757
     *         ))*X3+(((-4.*D*E))-(2.*D2 ))*X2+((6.*E2 -4.*D2 +2.)*X1))*   2758
     *         Y1))))/4.                                                   2759
           CF(6) = (-(D3  *((2.*X1*Y3)+D*X1*Y2+((2.*X3+D*X2-(4.*E*X1))*   2760
     *         Y1))))/4.)                                                  2761
           CF(7) =      (D4  *X1*Y1)/4.                                    2762
C     . END OF COMPUTED GO TO.                                             2763
 350   CONTINUE                                                            2764
       SUML = CF(7)*(2.*(G6+5.*G4*F2+3.*G2*F4+F6/7.)*EL1 + 2./7.*DL2       2765
     *       +2.*(F6+5.*F4*G2+3.*F2*G4+G6/7.)*ALOG3                        2766
     *       -(30.*T6+10.*T4+6.*T2)/105.)                                  2767
     *    + CF(6)*(((F6+G6-1.)/3.+5.*F2*G2*(F2+G2))*DL1                    2768
     *       +2.*(F4+10.*F2*G2/3.+G4)*FL3 -(15.*T5+5.*T3+3.*T1)/45.)       2769
     *    + CF(5)*(2.*(G4+2.*F2*G2+F4/5.)*EL1 +2./5.*DL2                   2770
     *       +2.*(F4+2.*F2*G2+G4/5.)*ALOG3 -(6.*T4+2.*T2)/15.)             2771
     *    + CF(4)*(((F4+G4-1.)/2.+3.*F2*G2)*DL1 +2.*(F2+G2)*FL3            2772
     *       -(3.*T3+T1)/6.)                                               2773
     *    + CF(3)*(2./3.)*((F2+3.*G2)*EL1 +DL2 +(3.*F2+G2)*ALOG3 -T2)     2774
     *    + CF(2)*((F2+G2-1.)*DL1 +2.*FL3 -T1)                            2775
     *    + CF(1)*2.*(EL1+DL2+ALOG3)                                      2776
       XDNDN = -XDNDN/Z3 + SUML/Z2                                       2777
       RETURN                                                            2778
       END                                                               2779
```

```
      OVERLAY(MAIN,11,0)                                                    2780
      PROGRAM QUAD9                                                         2781
C*************************************************************************2782
C                                                                           2783
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE          2784
C        ELEMENTS WITH NSF = 9.                                             2785
C                                                                           2786
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT         2787
C        GROUP TRANSFORMATIONS.                                             2788
C                                                                           2789
C*************************************************************************2790
C                                                                           2791
      DIMENSION LC(1),KC(1)                                                 2792
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,SKP(2),                           2793
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,ALG1,ALG2,ALG3,OTHERS(1)       2794
      COMMON/TEMP/I1,I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,               2795
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,                    2796
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,                            2797
     *   T1,T2,T3,T4,T5,T6,                                                 2798
     *   DL1,EL1,DL2,FL3,XJ                                                 2799
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))                               2800
C                                                                           2801
C                                                                           2802
      X1 = XX1                                                              2803
      X2 = XX2                                                              2804
      X3 = XX3                                                              2805
      Y1 = YY1                                                              2806
      Y2 = YY2                                                              2807
      Y3 = YY3                                                              2808
      Z1 = ZZ1                                                              2809
      Z2 = ZZ2                                                              2810
      Z3 = ZZ3                                                              2811
      ALOG1 = ALG1                                                          2812
      ALOG2 = ALG2                                                          2813
      ALOG3 = ALG3                                                          2814
      IL = IXC + 37                                                         2815
      IU = IXC + 45                                                         2816
      LL = 0                                                                2817
      DO 4 I=1,2                                                            2818
         DO 3 J=1,4                                                         2819
            LL = LL + 1                                                     2820
            D = Z2/Z3                                                       2821
            DP1 = D + 1.                                                    2822
            DM1 = D - 1.                                                    2823
            D2 = D*D                                                        2824
            D3 = D*D2                                                       2825
            E = -Z1/Z3                                                      2826
            EP1 = E + 1.                                                    2827
            EM1 = E - 1.                                                    2828
            E2 = E*E                                                        2829
            E3 = E*E2                                                       2830
            FF = Z1/Z2                                                      2831
            F2 = FF*FF                                                      2832
            F4 = F2*F2                                                      2833
            F6 = F2*F4                                                      2834
            G = Z3/Z2                                                       2835
            G2 = G*G                                                        2836
            G4 = G2*G2                                                      2837
            G6 = G2*G4                                                      2838
C           . T'L' = (Z2/Z3)*(((Z1+Z3)/Z2)**L - ((Z1-Z3)/Z2)**L)            2839
            T1 = 2.                                                         2840
            T2 = 4.*FF                                                      2841
            T3 = 2.*G2 + 6.*F2                                              2842
            T4 = 8.*FF*(G2+F2)                                              2843
            T5 = 2.*G4 + 20.*G2*F2 + 10.*F4                                 2844
            T6 = 2.*FF*T3*(3.*G2+F2)                                        2845
            DL1 = D*ALOG1                                                   2846
            EL1 = -E*ALOG1                                                  2847
```

```
      DL2 = D*ALOG2                                                    2848
      FL3 = FF*ALOG3                                                   2849
      DO 2 I1=IL,IU                                                    2850
         I2 = I1 + IC                                                  2851
         I3 = I2 + IC                                                  2852
         I4 = I3 + IC                                                  2853
         K = KC(I1) - 8                                                2854
         L = LC(I2)                                                    2855
C        IF (L.EQ.LL)                                                  2856
                                              IF(L.NE.LL) GO TO 1      2857
C        THEN                                                          2858
            XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)                         2859
            XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)                         2860
            XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)                         2861
            XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)                         2862
 1       CONTINUE                                                      2863
 2    CONTINUE                                                         2864
C     . TRANSFORMATION OF TYPE ONE.                                    2865
      X1 = -X1                                                         2866
      XJ =  X2                                                         2867
      X2 = -X3                                                         2868
      X3 =  XJ                                                         2869
      Y1 = -Y1                                                         2870
      XJ =  Y2                                                         2871
      Y2 = -Y3                                                         2872
      Y3 =  XJ                                                         2873
      XJ =  Z2                                                         2874
      Z2 =  Z3                                                         2875
      Z3 = -XJ                                                         2876
      ALOG1 = -ALOG1                                                   2877
      XJ    =  ALOG2                                                   2878
      ALOG2 = -ALOG3                                                   2879
      ALOG3 =  XJ                                                      2880
 3    CONTINUE                                                         2881
C     . TRANSFORMATION OF TYPE TWO.                                    2882
      X3 = -X3                                                         2883
      Y3 = -Y3                                                         2884
      Z3 = -Z3                                                         2885
      ALOG1 = -ALOG1                                                   2886
      ALOG2 = -ALOG2                                                   2887
 4  CONTINUE                                                          2888
    END                                                               2889
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)                                 2890
C*********************************************************************  2891
C                                                                        2892
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD9 TO EVALUATE         2893
C        C-INTEGRALS.                                                    2894
C                                                                        2895
C*********************************************************************  2896
C                                                                        2897
      COMMON/TEMP/I1,I2,I3,I4,K,L,XY(6),Z1,Z2,Z3,                       2898
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,                 2899
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,                         2900
     *   T1,T2,T3,T4,T5,T6,                                              2901
     *   DL1,EL1,DL2,FL3,SUML,CF(7)                                      2902
C                                                                        2903
      D2 = D*D                                                           2904
      D3 = D*D2                                                          2905
      D4 = D*D3                                                          2906
      E2 = E*E                                                           2907
      E3 = E*E2                                                          2908
      E4 = E*E3                                                          2909
      SUML = 0.                                                          2910
      GO TO (309,310,311),K                                              2911
C           . 9,1                                                        2912
 309      XDNDN = (-(((120.*E-24.*D+12.)*X3+(((-60.*E2 ))-60.*E-12.*D*    2913
     *        D+24.*D-20.)*X2+((((-120.*E2 ))+(24.*D+12.)*E-24.*D2 -16.  2914
     *        )*X1))*Y3+((60.*E2 +(((-120.*D))+30.)*E+36.*D2 -18.*D-40.  2915
     *        )*X3+((90.*D-60.)*E2 +(60.*D-60.)*E+18.*D3  -36.*D2 -20.*  2916
     *        D+40.)*X2+((((-30.*E3  ))+180.*D*E2 +(((-54.*D2 ))-36.*D+  2917
     *        20.)*E+36.*D3  -(76.*D))*X1))*Y2+(((((-180.*E2 ))+(120.*D  2918
     *        -30.)*E-60.*D2 +(30.*D))*X3+(90.*E3  +90.*E2 +(90.*D2 -(1  2919
     *        20.*D))*E+30.*D2 -(60.*D))*X2+((180.*E3  +(((-90.*D))-30.  2920
     *        )*E2 +(180.*D2 -60.)*E-18.*D3  -18.*D2 +8.*D+8.)*X1))*Y1)  2921
     *        )/45.)                                                     2922
          CF(1) = ((2.*E2 *X3+(((-E3  ))-E2 )*X2-(2.*E3  *X1))*Y3+((((   2923
     *        -2.*E3  *X3))+(E4  +E3  )*X2+(2.*E4  *X1))*Y1))/2.         2924
          CF(2) = (-(((2.*E2 +((((-4.*D))+1.)*E))*X3+((3.*D-2.)*E2 +((   2925
     *        2.*D-2.)*E))*X2+((((-E3  ))+(6.*D*E2 ))*X1))*Y3+((2.*E3    2926
     *        -(2.*E))*X3+(((-E4  ))-E3  +E2 +E)*X2+((((-2.*E4  ))+(2.*  2927
     *        E2 ))*X1))*Y2+(((((-2.*E3  ))+((6.*D-1.)*E2 ))*X3+((((-4.  2928
     *        *D))+2.)*E3  +((((-3.*D))+2.)*E2 ))*X2+((E4  -(8.*D*E3  )  2929
     *        )*X1))*Y1))/2.)                                            2930
          CF(3) = (-(((2.*E2 +(4.*D-1.)*E-2.*D2 +D)*X3+(((-E3  ))-E2 +   2931
     *        (3.*D2 -(4.*D))*E+D2 -(2.*D))*X2+((((-2.*E3  ))+(((-3.*D)  2932
     *        )-1.)*E2 +((6.*D2 -2.)*E))*X1))*Y3+((((-2.*E3  ))+(6.*D-1  2933
     *        .)*E2 +2.*E-2.*D+1.)*X3+((((-4.*D))+2.)*E3  +(((-3.*D))+2  2934
     *        .)*E2 +(2.*D-2.)*E+D-2.)*X2+((E4  -8.*D*E3  -E2 +(4.*D*E)  2935
     *        )*X1))*Y2+((((((-6.*D))+1.)*E2 +((6.*D2 -2.*D-2.)*E))*X3+  2936
     *        ((((-6.*D2 ))+6.*D+1.)*E2 +((((-3.*D2 ))+4.*D+1.)*E))*X2+  2937
     *        (((4.*D+1.)*E3  +((((-12.*D2 ))+4.)*E2 ))*X1))*Y1))/2.)    2938
          CF(4) = (((2.*E2 +(((-4.*D))+1.)*E-2.*D2 +D)*X3+((3.*D-2.)*E   2939
     *        *E+(2.*D-2.)*E-D3  +(2.*D2 ))*X2+((((-E3  ))+6.*D*E2 +(3.  2940
     *        *D2 +(2.*D))*E-2.*D3  +(2.*D))*X1))*Y3+(((6.*D-1.)*E2 +((  2941
     *        (-6.*D2 ))+(2.*D))*E-2.*D+1.)*X3+((6.*D2 -(6.*D))*E2 +(3.  2942
     *        *D2 -(4.*D))*E-D2 +(2.*D))*X2+(((((-4.*D))-1.)*E3  +(12.*  2943
     *        D2 -2.)*E2 +(2.*D+1.)*E-2.*D2 +2.)*X1))*Y2+((((6.*D2 -2.*  2944
     *        D-2.)*E-2.*D3  +D2 +2.*D-1.)*X3+((4.*D3  -6.*D2 -2.*D+2.)  2945
     *        *E+D3  -2.*D2 -D+2.)*X2+(((((-6.*D2 ))-3.*D+1.)*E2 +((8.*  2946
     *        D3  -(8.*D))*E))*X1))*Y1))/2.                              2947
          CF(5) = ((((4.*D-1.)*E-2.*D2 +D)*X3+((3.*D2 -(4.*D))*E+D2 -(   2948
     *        2.*D))*X2+(((((-3.*D))-1.)*E2 +(6.*D2 -2.)*E+D3  +D2 )*X1  2949
     *        ))*Y3+(((6.*D2 -(2.*D))*E-2.*D3  +D2 )*X3+((4.*D3  -(6.*D  2950
     *        *D))*E+D3  -(2.*D2 ))*X2+(((((-6.*D2 ))-(3.*D))*E2 +(8.*D  2951
     *        3  -(4.*D))*E+D2 +D)*X1))*Y2+(((2.*D3  -D2 -2.*D+1.)*X3+(  2952
     *        D4  -2.*D3  -D2 +(2.*D))*X2+(((((-4.*D3  ))-3.*D2 +2.*D+1  2953
     *        .)*E+2.*D4  -4.*D2 +2.)*X1))*Y1))/2.                       2954
          CF(6) = (D*((((((2.*D)-1.)*X3)+(D2 -(2.*D))*X2+(((((-3.*D))-   2955
     *        2.)*E+2.*D2 -2.)*X1))*Y3)+((2.*D2 -D)*X3+(D3  -(2.*D2 ))*  2956
     *        X2+(((((-4.*D2 ))-(3.*D))*E+2.*D3  -(2.*D))*X1))*Y2+((((-  2957
     *        D3  ))-D2 +D+1.)*X1*Y1))))/2.                              2958
          CF(7) =      (-((D+1.)*D2 *X1*(Y3+(D*Y2)))/2.)                 2959
          GO TO 350                                                      2960
```

```
C           . 9,5                                                          2961
 310        XDNDN = (((240.*E-(48.*D))*X3+(((-120.*E2 ))-24.*D2 +80.)*X2   2962
     *         +((((-240.*E2 ))+48.*D*E-48.*D2 -80.)*X1))*Y3+((120.*E2 -   2963
     *         240.*D*E+72.*D2 -80.)*X3+(180.*D*E2 +36.*D3  -(100.*D))*X   2964
     *         2+((((-60.*E3  ))+360.*D*E2 +(((-108.*D2 ))-20.)*E+72.*D*   2965
     *         *3-(80.*D))*X1))*Y2+(((((-360.*E2 ))+240.*D*E-(120.*D2 ))   2966
     *         *X3+(180.*E3  +((180.*D2 -180.)*E))*X2+((360.*E3  -180.*D   2967
     *         *E2 +360.*D2 *E-36.*D3  -(44.*D))*X1))*Y1))/45.             2968
           CF(1) = (((-2.*E2 *X3))+(E3  -E)*X2+(2.*E3  *X1))*Y3+(2.*E**   2969
     *         3*X3+(((-E4  ))+E2 )*X2-(2.*E4  *X1))*Y1                    2970
           CF(2) = ((2.*E2 -(4.*D*E))*X3+(3.*D*E2 -D)*X2+((((-E3  ))+6.   2971
     *         *D*E2 -E)*X1))*Y3+((2.*E3  -(2.*E))*X3+(((-E4  ))+2.*E2 -   2972
     *         1.)*X2+((((-2.*E4  ))+(2.*E2 ))*X1))*Y2+((((-2.*E3  ))+(6   2973
     *         .*D*E2 ))*X3+(((-4.*D*E3  ))+(2.*D*E))*X2+((E4  -8.*D*E**   2974
     *         3+E2 )*X1))*Y1                                              2975
           CF(3) = ((2.*E2 +4.*D*E-(2.*D2 ))*X3+(((-E3  ))+((3.*D2 +1.)   2976
     *         *E))*X2+((((-2.*E3  ))-3.*D*E2 +6.*D2 *E-D)*X1))*Y3+((((-   2977
     *         2.*E3  ))+6.*D*E2 +2.*E-(2.*D))*X3+(((-4.*D*E3  ))+(4.*D*   2978
     *         E))*X2+((E4  -8.*D*E3  +4.*D*E-1.)*X1))*Y2+((((-6.*D*E2 )   2979
     *         )+((6.*D2 -2.)*E))*X3+((((-6.*D2 ))+1.)*E2 +D2 -1.)*X2+((   2980
     *         4.*D*E3  +(((-12.*D2 ))+2.)*E2 +(2.*D*E))*X1))*Y1           2981
           CF(4) = ((((-2.*E2 ))+4.*D*E+(2.*D2 ))*X3+(((-3.*D*E2 ))+D**   2982
     *         3+D)*X2+((E3  -6.*D*E2 +(((-3.*D2 ))+1.)*E+(2.*D3  ))*X1)   2983
     *         )*Y3+((((-6.*D*E2 ))+6.*D2 *E+(2.*D))*X3+(((-6.*D2 *E2 ))   2984
     *         +(2.*D2 ))*X2+((4.*D*E3  -12.*D2 *E2 +(2.*D2 ))*X1))*Y2+(   2985
     *         ((((-6.*D2 ))+2.)*E+2.*D3  -(2.*D))*X3+(((-4.*D3  ))+(2.*   2986
     *         D))*E*X2+(((6.*D2 -1.)*E2 +(((-8.*D3  ))+(4.*D))*E+D2 -1.   2987
     *         )*X1))*Y1                                                   2988
           CF(5) = (-D*((((((4.*E)-(2.*D))*X3)+3.*D*E*X2+((((-3.*E2 ))+   2989
     *         6.*D*E+D2 -1.)*X1))*Y3)+((6.*D*E-(2.*D2 ))*X3+4.*D2 *E*X2   2990
     *         +((8.*D2 *E-(6.*D*E2 ))*X1))*Y2+((2.*D2 -2.)*X3+(D3  -D)*   2991
     *         X2+(((2-(4.*D2 ))*E+2.*D3  -(2.*D))*X1))*Y1))               2992
           CF(6) = (-D2 *((((2.*X3)+D*X2+((2.*D-(3.*E))*X1))*Y3)+(2.*D*   2993
     *         X3+D2 *X2+((2.*D2 -(4.*D*E))*X1))*Y2+(1-D2 )*X1*Y1))        2994
           CF(7) =      D3  *X1*(Y3+D*Y2)                                  2995
           GO TO 350                                                       2996
C          . 9,9                                                          2997
 311       XDNDN = ((384.*E*X3-192.*D*E*X2+((((-480.*E2 ))-96.*D2 -64.)   2998
     *         *X1))*Y3+(((-192.*D*E*X3))+(240.*E3  +((432.*D2 -400.)*E)   2999
     *         )*X2+((720.*D*E2 +144.*D3  -(304.*D))*X1))*Y2+(((((-480.*   3000
     *         E2 ))-96.*D2 -64.)*X3+(720.*D*E2 +144.*D3  -(304.*D))*X2+   3001
     *         ((720.*E3  +((720.*D2 -240.)*E))*X1))*Y1))/45.              3002
           CF(1) = (((-4.*E2 *X3))+(4.*E3  *X1))*Y3+(4.*E3  *X3-(4.*E**   3003
     *         4*X1))*Y1                                                   3004
           CF(2) = (((-8.*D*E*X3))+(4.*E3  -(4.*E))*X2+(12.*D*E2 *X1))*   3005
     *         Y3+((4.*E3  -(4.*E))*X3+((((-4.*E4  ))+(4.*E2 ))*X1))*Y2+   3006
     *         (12.*D*E2 *X3+(((-4.*E4  ))+(4.*E2 ))*X2-(16.*D*E3  *X1))   3007
     *         *Y1                                                         3008
           CF(3) = ((8.*E2 -(4.*D2 ))*X3+(12.*D*E2 -(4.*D))*X2+((((-4.*   3009
     *         E3  ))+((12.*D2 -4.)*E))*X1))*Y3+((12.*D*E2 -(4.*D))*X3+(   3010
     *         ((-4.*E4  ))+8.*E2 -4.)*X2+((((-16.*D*E3  ))+(8.*D*E))*X1   3011
     *         ))*Y2+((((-4.*E3  ))+((12.*D2 -4.)*E))*X3+(((-16.*D*E3  )   3012
     *         )+(8.*D*E))*X2+((((-24.*D2 ))+8.)*E2 *X1))*Y1               3013
           CF(4) = (16.*D*E*X3+(((-4.*E3  ))+((12.*D2 +4.)*E))*X2+((((-   3014
     *         12.*D*E2 ))+4.*D3  -(4.*D))*X1))*Y3+((((-4.*E3  ))+((12.*   3015
     *         D2 +4.)*E))*X3+(((-16.*D*E3  ))+(16.*D*E))*X2+(((((-24.*D   3016
     *         *D))+4.)*E2 +4.*D2 -4.)*X1))*Y2+((((-12.*D*E2 ))+4.*D3  -   3017
     *         (4.*D))*X3+((((-24.*D2 ))+4.)*E2 +4.*D2 -4.)*X2+((((-16.*   3018
     *         D3  ))+(16.*D))*E*X1))*Y1                                   3019
           CF(5) = (-4.*((((((E2 -(2.*D2 ))*X3)+(3.*D*E2 -D3  -D)*X2+((3   3020
     *         .*D2 -1.)*E*X1))*Y3)+((3.*D*E2 -D3  -D)*X3+(6.*D2 *E2 -(2   3021
     *         .*D2 ))*X2+((4.*D3  -(2.*D))*E*X1))*Y2+((3.*D2 -1.)*E*X3+   3022
     *         (4.*D3  -(2.*D))*E*X2+((D4  -2.*D2 +1.)*X1))*Y1))          3023
           CF(6) = (-4.*D*((((2.*E*X3)+3.*D*E*X2+((D2 -1.)*X1))*Y3)+(3.   3024
     *         *D*E*X3+4.*D2 *E*X2+((D3  -D)*X1))*Y2+((D2 -1.)*X3+((D3     3025
     *         -D)*X2))*Y1))                                               3026
           CF(7) =      (-4.*D2 *(X3+(D*X2))*(Y3+D*Y2))                    3027
C     . END OF COMPUTED GO TO.                                             3028
```

```
350  CONTINUE                                                          3029
     SUML = CF(7)*(2.*(G6+5.*G4*F2+3.*G2*F4+F6/7.)*EL1 + 2./7.*DL2     3030
    *       +2.*(F6+5.*F4*G2+3.*F2*G4+G6/7.)*ALOG3                    3031
    *       -(30.*T6+10.*T4+6.*T2)/105.)                              3032
    *    + CF(6)*(((F6+G6-1.)/3.+5.*F2*G2*(F2+G2))*DL1                3033
    *       +2.*(F4+10.*F2*G2/3.+G4)*FL3 -(15.*T5+5.*T3+3.*T1)/45.)   3034
    *    + CF(5)*(2.*(G4+2.*F2*G2+F4/5.)*EL1 +2./5.*DL2               3035
    *       +2.*(F4+2.*F2*G2+G4/5.)*ALOG3 -(6.*T4+2.*T2)/15.)         3036
    *    + CF(4)*(((F4+G4-1.)/2.+3.*F2*G2)*DL1 +2.*(F2+G2)*FL3        3037
    *       -(3.*T3+T1)/6.)                                           3038
    *    + CF(3)*(2./3.)*((F2+3.*G2)*EL1 +DL2 +(3.*F2+G2)*ALOG3 -T2)  3039
    *    + CF(2)*((F2+G2-1.)*DL1 +2.*FL3 -T1)                         3040
    *    + CF(1)*2.*(EL1+DL2+ALOG3)                                   3041
     XDNDN = -XDNDN/Z3 + SUML/Z2                                      3042
     RETURN                                                           3043
     END                                                              3044
```

```
      OVERLAY(MAIN,12,0)                                                3045
      PROGRAM SGPM                                                      3046
      LOGICAL SMFLAG,SPFLAG                                             3047
      COMMON/SPACE/SPACE(27),SMFLAG,SPFLAG,OTHERS(1)                    3048
C                                                                       3049
C                                                                       3050
      CALL LINSTF                                                       3051
      IF (SPFLAG)                                                       3052
C     THEN                                                              3053
     *   CALL LODVEC                                                    3054
C     CONTINUE                                                          3055
      IF (SMFLAG)                                                       3056
C     THEN                                                              3057
     *   CALL MASS                                                      3058
C     CONTINUE                                                          3059
      END                                                               3060
```

```
      SUBROUTINE LINSTF                                                 3061
C*********************************************************************  3062
C                                                                       3063
C     THIS SUBROUTINE CALCULATES THE LINEAR STIFFNESS MATRIX  SS        3064
C        AND THE GEOMETRIC STIFFNESS MATRIX  SG  FOR AN ISOLATED        3065
C        DOUBLY-CURVED SHALLOW SHELL ELEMENT.                           3066
C                                                                       3067
C     THE NUMBER OF DEGREES OF FREEDOM PER NODE IS FIVE.                3068
C     NSF  IS THE NUMBER OF SHAPE FUNCTIONS PER ELEMENT.                3069
C     NNE  IS THE NUMBER OF *NODAL* SHAPE FUNCTIONS PER ELEMENT.        3070
C     THE QUANTITIES STORED IN POSITIONS 1 THRU 21 OF COMMON ARE THE    3071
C        MATERIAL STIFFNESS COEFFICIENTS.  THE FIRST SIX OF THESE       3072
C        QUANTITIES ARE THE EXTENSIONAL STIFFNESSES OF THE SHELL, THE   3073
C        NEXT SIX ARE THE STIFFNESS INTERACTION COEFFICIENTS, THE NEXT  3074
C        SIX ARE THE BENDING STIFFNESSES, AND THE LAST THREE ARE THE    3075
C        TRANSVERSE SHEAR STIFFNESSES.                                  3076
C     EN1,EN2,EN12  ARE THE PRESTRESS COEFFICIENTS.                     3077
C     USE  CURVE = .TRUE.  IF THE CURVATURE IS NONZERO.                 3078
C     USE  CURVE = .FALSE. IF THE CURVATURE TERMS ARE TO BE IGNORED.    3079
C     USE SGFLAG = .TRUE.  IF THE GEOMETRIC STIFFNESS MATRIX  SG  IS TO 3080
C        BE COMPUTED.  IN THIS CASE THE DIMENSIONS OF  SG  MUST BE      3081
C     USE  SGFLAG = .FALSE. IF SG IS NOT TO BE COMPUTED.                3082
C     Q1,Q2,Q12  ARE THE CURVATURES AT THE NODES.                       3083
C     NDFE IS THE NUMBER OF DEGREES OF FREEDOM PER ELEMENT  (NDFE=5*NSF) 3084
C     THE ARRAY  SS  IS TO BE STORED IN POSITIONS  ISS+1  THRU          3085
C        ISS+NDFE*NDFE  OF COMMON.                                      3086
C     THE ARRAY  SG  IS TO BE STORED IN POSITIONS  ISG+1  THRU          3087
C        ISG+NSF*NSF  OF COMMON.                                        3088
C     THE INTEGRALS XDNDN ARE STORED IN POSITIONS  IXC+1  THRU  IXC+4*IC 3089
C        OF COMMON.                                                     3090
C     THE INTEGRALS XNNDN ARE STORED IN POSITIONS  IXB+1  THRU  IXB+2*IB 3091
C        OF COMMON.                                                     3092
C     THE INTEGRALS XNNNN ARE STORED IN POSITIONS  IXA+1  THRU  IXA+IA  3093
C        OF COMMON.                                                     3094
C     SS  SHOULD BE THOUGHT OF AS HAVING FOUR INDICES,                  3095
C        I.E. SS = SS(J1,K1,J2,K2), (J1,J2=1 THRU 5), (K1,K2=1 THRU NSF) 3096
C                                                                       3097
C*********************************************************************  3098
C                                                                       3099
      LOGICAL SGFLAG,CURVE                                              3100
      DIMENSION SG(1),SS(1)                                             3101
      COMMON/SPACE/C11,C12,C16,C22,C26,C66,                             3102
     *   F11,F12,F16,F22,F26,F66,D11,D12,D16,D22,D26,D66,C55,C44,C54,   3103
     *   EN1,EN2,EN12,NSF,CURVE,SGFLAG,DUM(13),                        3104
     *   Q1(10),Q2(10),Q12(10),PRESS(30),                               3105
     *   DUMMY(14),NNE,ISS,ISG,OTHERS(1)                                3106
      COMMON/TEMP/SKIP(6),                                              3107
     *            CURC1,CURC2,CURC6,CURF1,CURF2,CURF6,J,J1,J2,JON,J1N,  3108
     *   J2N,J3N,J4N,J5N,K1,K1N,K1M,K1P1,K2,K2G,K2M,K2N,K3,K3M1,K4,NDFE,  3109
     *   NDFE5,NSFM1,                                                   3110
     *   QU1,QU2,QU12,XNN,XN1N,XN2N,XNN1N,XNN2N,X1N1N,X1N2N,X2N1N,X2N2N  3111
     *   ,DIAG,OFFD,Q21(10),QU21                                        3112
      EQUIVALENCE (SS(1),C11),(SG(1),C11)                               3113
C                                                                       3114
      CDNDN(A,B,C,D) = A*X1N1N + B*X1N2N + C*X2N1N + D*X2N2N            3115
      CURV(A,B,C) = A*QU1 + B*QU2 + C*QU21                              3116
C                                                                       3117
C                                                                       3118
C     IF (CURVE)                                                        3119
                                                  IF(.NOT.CURVE)GOTO 10 3120
C     THEN                                                              3121
         DO 1 J=1,NNE                                                   3122
            Q21(J) = 2.*Q12(J)                                          3123
 1       CONTINUE                                                       3124
 10   CONTINUE                                                          3125
      NDFE = 5*NSF                                                      3126
      NDFE5 = 5*NDFE                                                    3127
      JON = ISS - 5*(NDFE+1)                                            3128
      DO 8 K1=1,NSF                                                     3129
         JON = JON + 5                                                  3130
```

```
      J1N = JON                                                          3131
      K2G = ISG - NSF                                                    3132
      DO 8 K2=1,K1                                                       3133
         J1N = J1N + NDFE5                                               3134
         J2N = J1N + NDFE                                                3135
         J3N = J2N + NDFE                                                3136
         J4N = J3N + NDFE                                                3137
         J5N = J4N + NDFE                                                3138
         K2G = K2G + NSF                                                 3139
         X1N1N = XDNDN(1,K1,1,K2)                                        3140
         X1N2N = XDNDN(1,K1,2,K2)                                        3141
         X2N1N = XDNDN(2,K1,1,K2)                                        3142
         X2N2N = XDNDN(2,K1,2,K2)                                        3143
         SS(1+J1N) = CDNDN(C11,C16,C16,C66)                              3144
         SS(1+J2N) = CDNDN(C16,C12,C66,C26)                              3145
         SS(2+J1N) = CDNDN(C16,C66,C12,C26)                              3146
         SS(2+J2N) = CDNDN(C66,C26,C26,C22)                              3147
         SS(1+J4N) = CDNDN(F11,F16,F16,F66)                              3148
         SS(1+J5N) = CDNDN(F16,F12,F66,F26)                              3149
         SS(2+J4N) = CDNDN(F16,F66,F12,F26)                              3150
         SS(2+J5N) = CDNDN(F66,F26,F26,F22)                              3151
         SS(4+J1N) = CDNDN(F11,F16,F16,F66)                              3152
         SS(4+J2N) = CDNDN(F16,F12,F66,F26)                              3153
         SS(5+J1N) = CDNDN(F16,F66,F12,F26)                              3154
         SS(5+J2N) = CDNDN(F66,F26,F26,F22)                              3155
         SS(4+J4N) = CDNDN(D11,D16,D16,D66)                              3156
         SS(4+J5N) = CDNDN(D16,D12,D66,D26)                              3157
         SS(5+J4N) = CDNDN(D16,D66,D12,D26)                              3158
         SS(5+J5N) = CDNDN(D66,D26,D26,D22)                              3159
         SS(1+J3N) = 0.                                                  3160
         SS(2+J3N) = 0.                                                  3161
         SS(3+J1N) = 0.                                                  3162
         SS(3+J2N) = 0.                                                  3163
C        DO  (ADD ON THE TERMS INVOLVING  C55,C54,C44)                   3164
            SS(3+J3N) = CDNDN(C55,C54,C54,C44)                           3165
            XN1N = XNNDN(K2,0,1,K1)                                      3166
            XN2N = XNNDN(K2,0,2,K1)                                      3167
            SS(3+J4N) = C55*XN1N + C54*XN2N                              3168
            SS(3+J5N) = C54*XN1N + C44*XN2N                              3169
            XN1N = XNNDN(K1,0,1,K2)                                      3170
            XN2N = XNNDN(K1,0,2,K2)                                      3171
            SS(4+J3N) = C55*XN1N + C54*XN2N                              3172
            SS(5+J3N) = C54*XN1N + C44*XN2N                              3173
            XNN = XNNNN(K1,K2,0,0)                                       3174
            SS(4+J4N) = SS(4+J4N) + C55*XNN                              3175
            SS(4+J5N) = SS(4+J5N) + C54*XNN                              3176
            SS(5+J4N) = SS(5+J4N) + C54*XNN                              3177
            SS(5+J5N) = SS(5+J5N) + C44*XNN                              3178
C        CONTINUE                                                        3179
         IF (CURVE)                                  GO TO 2             3180
                                                     GO TO 7             3181
C        THEN  ADD ON THE CURVATURE TERMS                                3182
 2          DIAG = 0                                                    3183
            OFFD = 0                                                    3184
            DO 6 K3=1,NNE                                               3185
               QU1  = Q1(K3)                                            3186
               QU2  = Q2(K3)                                            3187
               QU21 = Q21(K3)                                           3188
               CURC1 = CURV(C11,C12,C16)                                3189
               CURC2 = CURV(C12,C22,C26)                                3190
               CURC6 = CURV(C16,C26,C66)                                3191
               CURF1 = CURV(F11,F12,F16)                                3192
               CURF2 = CURV(F12,F22,F26)                                3193
               CURF6 = CURV(F16,F26,F66)                                3194
               XNN1N = XNNDN(K2,K3,1,K1)                                3195
               XNN2N = XNNDN(K2,K3,2,K1)                                3196
               SS(1+J3N) = SS(1+J3N) + CURC1*XNN1N + CURC6*XNN2N        3197
               SS(2+J3N) = SS(2+J3N) + CURC6*XNN1N + CURC2*XNN2N        3198
```

```
                  SS(4+J3N) = SS(4+J3N) + CURF1*XNN1N + CURF6*XNN2N                3199
                  SS(5+J3N) = SS(5+J3N) + CURF6*XNN1N + CURF2*XNN2N                3200
                  XNN1N = XNNDN(K1,K3,1,K2)                                       3201
                  XNN2N = XNNDN(K1,K3,2,K2)                                       3202
                  SS(3+J1N) = SS(3+J1N) + CURC1*XNN1N + CURC6*XNN2N                3203
                  SS(3+J2N) = SS(3+J2N) + CURC6*XNN1N + CURC2*XNN2N                3204
                  SS(3+J4N) = SS(3+J4N) + CURF1*XNN1N + CURF6*XNN2N                3205
                  SS(3+J5N) = SS(3+J5N) + CURF6*XNN1N + CURF2*XNN2N                3206
C                 . THE FOLLOWING CODE IS EQUIVALENT TO SUMMING                   3207
C                    BOTH K3 AND K4 FROM 1 TO NNE.                                3208
C                 . OFF-DIAGONAL TERMS HAVE A COMPENSATING FACTOR OF 2.           3209
                  IF(K3.NE.1)                        GO TO 3                      3210
                                                     GO TO 5                      3211
C                 THEN     (TERMS WITH K3 NOT EQUAL K4)                           3212
 3                   K3M1 = K3 - 1                                                3213
                     DO 4 K4=1,K3M1                                               3214
                        OFFD = OFFD + XNNNN(K1,K2,K3,K4)*                         3215
     *                     (Q1(K4)*CURC1 +Q2(K4)*CURC2 +Q21(K4)*CURC6)            3216
 4                   CONTINUE                                                     3217
C                 CONTINUE       (TERM WITH K3 = K4)                            3218
 5                DIAG = DIAG + XNNNN(K1,K2,K3,K3)*                              3219
     *                (QU1*CURC1 +QU2*CURC2 +QU21*CURC6)                         3220
 6             CONTINUE                                                         3221
               SS(3+J3N) = SS(3+J3N) + DIAG + OFFD + OFFD                       3222
 7          CONTINUE                                                            3223
            IF (SGFLAG)                                                         3224
C           THEN                                                                3225
     *         SG(K1+K2G) = EN1*X1N1N + EN2*X2N2N + EN12*(X1N2N+X2N1N)           3226
C           CONTINUE                                                            3227
 8       CONTINUE                                                               3228
C     CONTINUE                                                                  3229
C*****************************************************************************3230
C     . SYMMETRIZE SS AND SG                                                    3231
      K1N = ISS - NDFE - 5                                                      3232
      K1M = -NDFE5                                                              3233
      NSFM1 = NSF - 1                                                           3234
      DO 9 K1=1,NSFM1                                                           3235
         K1N = K1N + 5                                                          3236
         K1M = K1M + NDFE5                                                      3237
         K2N = K1M + K1N                                                        3238
         K2M = K2N                                                              3239
         K1P1 = K1 + 1                                                          3240
         DO 9 K2=K1P1,NSF                                                       3241
            K2N = K2N + NDFE5                                                   3242
            K2M = K2M + 5                                                       3243
            IF (SGFLAG)                                                         3244
C           THEN                                                                3245
     *         SG(ISG+K1+K2*NSF-NSF) = SG(ISG+K2+K1*NSF-NSF)                     3246
C           CONTINUE                                                            3247
            DO 9 J1=1,5                                                         3248
               DO 9 J2=1,5                                                      3249
                  SS(J1+J2*NDFE+K2N) = SS(J2+J1*NDFE+K2M)                       3250
C           CONTINUE                                                            3251
C        CONTINUE                                                               3252
C     CONTINUE                                                                  3253
 9    CONTINUE                                                                  3254
C  ...ONLY EXIT                                                                 3255
      RETURN                                                                    3256
      END                                                                       3257
```

```
      SUBROUTINE LODVEC                                                 3258
C*********************************************************************** 3259
C                                                                       3260
C     THIS SUBROUTINE EVALUATES THE CONSISTENT LOAD SP.                 3261
C     THE NUMBER OF DEGREES OF FREEDOM PER NODE IS FIVE                 3262
C     NSF  IS THE NUMBER OF SHAPE FUNCTIONS PER ELEMENT                 3263
C     NNE  IS THE NUMBER OF *NODAL* SHAPE FUNCTIONS PER ELEMENT         3264
C     P  CONTAINS THE NODAL VALUES OF THE NORMAL(TRANSVERSE) LOADS      3265
C     P1,P2  CONTAIN THE NODAL VALUES OF THE IN-PLANE LOADS             3266
C     SP IS TO BE STORED BETWEEN POSITIONS  IXC+1  AND  IXC+5*NNE OF    3267
C        COMMON.                                                        3268
C                                                                       3269
C*********************************************************************** 3270
C                                                                       3271
      COMMON/SPACE/SP(70),P(10),P1(10),P2(10),                          3272
     *   DUMMY(14),NNE,SKP(2),IXC,OTHERS(1)                            3273
      COMMON/TEMP/SKIP(6),I,IL,IU,K,K1,K2,XNN                           3274
C                                                                       3275
C                                                                       3276
      IL = IXC + 1                                                      3277
      IU = IXC + 5*NNE                                                  3278
      DO 1 I=IL,IU                                                      3279
         SP(I) = 0                                                      3280
 1    CONTINUE                                                          3281
      K = IXC - 5                                                       3282
      DO 2 K1=1,NNE                                                     3283
         K = K + 5                                                      3284
         DO 2 K2=1,NNE                                                  3285
            XNN = XNNNN(K1,K2,0,0)                                      3286
            SP(1+K) = SP(1+K) + P1(K2)*XNN                              3287
            SP(2+K) = SP(2+K) + P2(K2)*XNN                              3288
            SP(3+K) = SP(3+K) -  P(K2)*XNN                              3289
C        CONTINUE                                                       3290
 2    CONTINUE                                                          3291
      RETURN                                                            3292
      END                                                               3293
```

```
      SUBROUTINE MASS                                                   3294
C*********************************************************************** 3295
C                                                                       3296
C     THIS SUBROUTINE CALCULATES THE CONSISTENT MASS MATRIX  SM  FOR AN  3297
C        ISOLATED DOUBLY-CURVED SHALLOW SHELL FINITE ELEMENT.           3298
C     RHO  IS THE DENSITY OF THE SHELL MATERIAL                         3299
C     H    IS THE SHELL THICKNESS                                       3300
C     SM  IS TO BE STORED BETWEEN POSITIONS  IXC+5*NNE+1  AND           3301
C        IXC+5*NNE+NSF*NSF  OF COMMON.                                  3302
C                                                                       3303
C*********************************************************************** 3304
C                                                                       3305
      COMMON/SPACE/SM(24),NSF,DUMMY(5),RHO,H,DUM(68),                   3306
     *   SKIPP(14),NNE,SKP(2),IXC,OTHERS(1)                            3307
      COMMON/TEMP/SKIP(6),IXM,K1,K1N,K2,TEMP                            3308
C                                                                       3309
C                                                                       3310
      TEMP = RHO*H                                                      3311
      IXM = IXC + 5*NNE                                                 3312
      DO 2 K1=1,NSF                                                     3313
         K1N = IXM + K1 - NSF                                           3314
         DO 2 K2=1,NSF                                                  3315
            SM(K1N+K2*NSF) = TEMP*XNNNN(K1,K2,0,0)                      3316
C        CONTINUE                                                       3317
 2    CONTINUE                                                          3318
      RETURN                                                            3319
      END                                                               3320
```

```
      FUNCTION XDNDN(L1,K1,L2,K2)                                        3321
C*************************************************************************3322
C                                                                        3323
C     THIS SUBROUTINE RETRIEVES XDNDN(L1,K1,L2,K2) FROM WHERE IT WAS     3324
C        PREVIOUSLY STORED.  IT IS LOCATED BETWEEN XC(IXC+1) AND         3325
C        XC(IXC+IC).                                                     3326
C                                                                        3327
C*************************************************************************3328
C                                                                        3329
      DIMENSION M(10)                                                    3330
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,OTHERS(1)                      3331
      COMMON/TEMP/INDX                                                   3332
      DATA (M(I),I=1,10)/0,1,3,6,10,15,21,28,36,45/                      3333
C                                                                        3334
C                                                                        3335
C     IF (K1.GE.K2)                                                      3336
                                                IF(K1.LT.K2) GO TO 1     3337
C     THEN                                                               3338
        INDX = K2 + M(K1) + IC*(L1+L1+L2-3)                              3339
                                                GO TO 2                  3340
C     ELSE                                                               3341
 1      INDX = K1 + M(K2) + IC*(L1+L2+L2-3)                              3342
 2    CONTINUE                                                           3343
      XDNDN = XC(IXC+INDX)                                               3344
      RETURN                                                             3345
      END                                                                3346
```

```
      FUNCTION XNNDN(K1,K2,L,K3)                                         3347
C*************************************************************************3348
C                                                                        3349
C     THIS SUBROUTINE RETRIEVES XNNDN(K1,K2,L,K3) FROM WHERE IT WAS      3350
C        PREVIOUSLY STORED.  IT IS LOCATED BETWEEN XB(IBX+1) AND         3351
C        XB(IBX+IB).                                                     3352
C                                                                        3353
C*************************************************************************3354
C                                                                        3355
      DIMENSION M(11)                                                    3356
      COMMON/SPACE/XB(24),NSF,SKIP(75),SKP(9),IB,SKIPP(8),IXB,OTHERS(1)  3357
      COMMON/TEMP/INDX                                                   3358
      DATA (M(I),I=1,11)/0,1,3,6,10,15,21,28,36,45,55/                   3359
C                                                                        3360
C                                                                        3361
C     IF (K1.GE.K2)                                                      3362
                                                IF(K1.LT.K2) GO TO 1     3363
C     THEN                                                               3364
        INDX = NSF*(K2+M(K1+1))                                          3365
                                                GO TO 2                  3366
C     ELSE                                                               3367
 1      INDX = NSF*(K1+M(K2+1))                                          3368
 2    CONTINUE                                                           3369
      XNNDN = XB(IXB+INDX+IB*(L-1)+K3)                                   3370
      RETURN                                                             3371
      END                                                                3372
```

```
      FUNCTION XNNNN(KK1,KK2,KK3,KK4)                                    3373
C*************************************************************************3374
C                                                                         3375
C     THIS SUBROUTINE RETRIEVES XNNNN(K1,K2,K3,K4) FROM WHERE IT WAS      3376
C        PREVIOUSLY STORED.  IT IS LOCATED BETWEEN XA(IXA+1) AND          3377
C        XA(IXA+IA).                                                      3378
C                                                                         3379
C*************************************************************************3380
C                                                                         3381
      DIMENSION M1(11),M2(11),M3(11)                                      3382
      COMMON/SPACE/XA(119),IXA,OTHERS(1)                                  3383
      COMMON/TEMP/KJ,INDX,K1,K2,K3,K4                                     3384
      DATA (M1(I),I=1,11)/0,1,5,15,35,70,126,210,330,495,715/             3385
      DATA (M2(I),I=1,11)/0,1,4,10,20,35,56,84,120,165,220/               3386
      DATA (M3(I),I=1,11)/0,1,3,6,10,15,21,28,36,45,55/                   3387
C                                                                         3388
C                                                                         3389
      K1 = KK1 + 1                                                        3390
      K2 = KK2 + 1                                                        3391
      K3 = KK3 + 1                                                        3392
      K4 = KK4 + 1                                                        3393
C     . PLACE  K1,K2,K3,K4  INTO DESCENDING ORDER .                       3394
      IF (K1.GE.K3) GO TO 1                                               3395
         KJ = K1                                                          3396
         K1 = K3                                                          3397
         K3 = KJ                                                          3398
 1    IF (K2.GE.K4) GO TO 2                                               3399
         KJ = K2                                                          3400
         K2 = K4                                                          3401
         K4 = KJ                                                          3402
 2    IF (K1.GE.K2) GO TO 3                                               3403
         KJ = K1                                                          3404
         K1 = K2                                                          3405
         K2 = KJ                                                          3406
 3    IF (K3.GE.K4) GO TO 4                                               3407
         KJ = K3                                                          3408
         K3 = K4                                                          3409
         K4 = KJ                                                          3410
 4    IF (K2.GE.K3) GO TO 5                                               3411
         KJ = K2                                                          3412
         K2 = K3                                                          3413
         K3 = KJ                                                          3414
 5    CONTINUE                                                            3415
C     . K1,K2,K3,K4  ARE NOW IN DESCENDING ORDER .                        3416
      INDX = M1(K1) + M2(K2) + M3(K3) + K4                                3417
      XNNNN = XA(IXA+INDX)                                                3418
      RETURN                                                              3419
      END                                                                 3420
```

```
      OVERLAY(MAIN,13,0)                                                3421
      PROGRAM PRINT                                                     3422
C*********************************************************************3423
C                                                                       3424
C     THIS PROGRAM PRINTS SS, SG, SP AND SM IF THEY HAVE BEEN EVALUATED.3425
C        IT ALSO RECONSTRUCTS AND PRINTS THE FULL CONSISTENT MASS MATRIX 3426
C        FROM  SM,  BUT IN DOING SO IT CLOBBERS  SS  IN CORE.           3427
C                                                                       3428
C*********************************************************************3429
C                                                                       3430
      LOGICAL SGFLAG,SMFLAG,SPFLAG,SWM                                  3431
      DIMENSION LABEL1(4),LABEL2(4),LABEL3(4),LABEL4(4),LABEL5(4)       3432
      DIMENSION SS(1),SG(1),SP(1),SM(1),SMASS(1)                        3433
      COMMON/SPACE/SPACE(24),NSF,CURVE,SGFLAG,SMFLAG,SPFLAG,PRFLAG,     3434
     *    RHO,H,DUMMY(68),                                              3435
     *    SKIP(14),NNE,ISS,ISG,IXC,OTHERS(1)                            3436
      COMMON/TEMP/I,IL,IU,K1,K1N,K12,K2,K2N,NDFE,SWM,TEMP,TEMP1         3437
      EQUIVALENCE (SS(1),SPACE(1)),(SG(1),SPACE(1)),(SP(1),SPACE(1)),   3438
     *    (SM(1),SPACE(1)),(SMASS(1),SPACE(1))                          3439
C                                                                       3440
      DATA LABEL1/40HSTIFFNESS MATRIX --- SS                  /         3441
      DATA LABEL2/40HGEOMETRIC STIFFNESS ARRAY --- SG         /         3442
      DATA LABEL3/40HLOAD VECTOR --- SP                       /         3443
      DATA LABEL4/40HCONSISTENT MASS ARRAY --- SM             /         3444
      DATA LABEL5/40HFULL CONSISTENT MASS MATRIX --- SMASS    /         3445
C                                                                       3446
C                                                                       3447
      SWM = .TRUE.                                                      3448
      NDFE = 5*NSF                                                      3449
      IL = ISS + 1                                                      3450
      IU = ISS + NDFE*NDFE                                              3451
      WRITE (6,1) LABEL1,(SS(I),I=IL,IU)                                3452
 1       FORMAT (1H1,40X,4A10//(10E12.5))                               3453
      IF (SGFLAG)                                  GO TO 2              3454
                                                   GO TO 3              3455
C     THEN                                                              3456
 2       IL = ISG + 1                                                   3457
         IU = ISG + NSF*NSF                                             3458
         WRITE (6,1) LABEL2,(SG(I),I=IL,IU)                             3459
 3    CONTINUE                                                          3460
      IF (SPFLAG)                                  GO TO 4              3461
                                                   GO TO 6              3462
C     THEN                                                              3463
 4       IL = IXC + 1                                                   3464
         IU = IXC + 5*NNE                                               3465
         WRITE (6,5) LABEL3,(SP(I),I=IL,IU)                             3466
 5          FORMAT(////40X,4A10//(10E12.5))                             3467
 6    CONTINUE                                                          3468
      IF (SMFLAG)                                  GO TO 7              3469
                                                   GO TO 8              3470
C     THEN                                                              3471
 7       IL = IXC + 5*NNE + 1                                           3472
         IU = IXC + 5*NNE + NSF*NSF                                     3473
         WRITE (6,5) LABEL4,(SM(I),I=IL,IU)                             3474
 8    CONTINUE                                                          3475
      IF (SWM .AND. SMFLAG)                        GO TO 9              3476
                                                   GO TO 12             3477
C     THEN                                                              3478
 9       IL = ISS + 1                                                   3479
         IU = ISS + NDFE*NDFE                                           3480
         DO 10 I=IL,IU                                                  3481
            SS(I) = 0.                                                  3482
 10      CONTINUE                                                       3483
         TEMP1 = H*H/12.                                                3484
         K1N = ISS - NDFE - 5                                           3485
         DO 11 K1=1,NSF                                                 3486
            K1N = K1N + 5                                               3487
            DO 11 K2=1,NSF                                              3488
               K12 = 5*NDFE*(K2-1) + K1N                                3489
```

```
                TEMP = SM(IXC+5*NNE+K1+NSF*(K2-1))                    3490
                SMASS(1+K12+  NDFE) = TEMP                            3491
                SMASS(2+K12+2*NDFE) = TEMP                            3492
                SMASS(3+K12+3*NDFE) = TEMP                            3493
                SMASS(4+K12+4*NDFE) = TEMP*TEMP1                      3494
                SMASS(5+K12+5*NDFE) = TEMP*TEMP1                      3495
C            CONTINUE                                                 3496
 11       CONTINUE                                                    3497
       WRITE(6,1) LABEL5,(SMASS(I),I=IL,IU)                          3498
 12    CONTINUE                                                       3499
       END                                                            3500
```

APPENDIX B

FIXED INPUT ARRAYS FOR THE SIX ELEMENT TYPES

NSF = 4

```
KA:   1   2   3   4   5   2   6   7   8   3   7   9   4   8   5   2  10  11  12  63501
     13  14   7  15   8   3  11  16   7  14   9   4  12   8   5   2   6   7   8 103502
     13  15  11  14  12   6  13  14  13  17  14   7  14  15   8   3   7   9  11 143503
     16   7  15  14   9   4   3  12   8   5                                      3504
LA:1111141114514543111411414331664336322886222454338414334322222737322272      3505
QA:  0 0 0 0                                        1                           3506
     1 0 0 0          -1          -1          3     3                           3507
     1 1 0 0          -2          -2          4     9                           3508
     1 1 1 0          -3          -3          5    20                           3509
     1 1 1 1          -8          -8         12    75                           3510
     2 1 0 0           0          -1          2     9                           3511
     2 1 1 0          -1          -3          5    60                           3512
     2 1 1 1          -1          -2          3    75                           3513
     2 2 1 1           0          -4          6   225                           3514
     3 1 0 0           0           0          1     9                           3515
     3 1 1 0          -1          -1          5   180                           3516
     3 1 1 1          -1          -1          3   300                           3517
     3 2 1 0           1          -1          5   180                           3518
     3 2 1 1           0          -1          3   450                           3519
     3 2 2 1           1          -1          3   300                           3520
     3 3 1 1           0           0          1   225                           3521
     4 3 2 1           0           0          1   225                           3522
KB:  1   1   1   1   2   3   4   3   5   6   7   6   3   2   3   4   8   8   9 93523
     6   5   6   7   4   3   2   3  10  11  10  11   9   8   8   9   7   6   5 63524
     3   4   3   2   8   9   9   8  11  10  11  10   9   9   8   8   6   7   6 53525
LB:1432111811185444151554443633113364643633227282822442373722 72               3526
QB:    0 0 1           0           1          -1     1                           3527
       1 0 1           0           1          -1     3                           3528
       1 0 2          -1           1           2     6                           3529
       1 0 3           0          -1           1     6                           3530
       1 1 1           0           1          -1     6                           3531
       1 1 2          -2           1           3    18                           3532
       1 1 3           0          -1           1    18                           3533
       2 1 1          -1           2          -3    36                           3534
       2 1 3           1          -2           1    36                           3535
       3 1 1           0           1          -1    36                           3536
       3 1 2           0           1           1    36                           3537
KC:  1   2   1   3   2   1   2   3   2   1                                       3538
LC:1141438432                                                                   3539
QC:      1 1          -4           3           3    -4          12               3540
         2 1           4          -3           3    -2          12               3541
         3 1           2          -3          -3     2          12               3542
```

APPENDIX B

NSF = 5

```
KA:   1   2   3   4   5   2   6   7   5   3   7   9   4   6   5   2  10  11  12   63543
     13  14   7  15   8   3  11  16   7  14   9   4  12   8   5   2   6   7   8  103544
     13  15  11  14  12   6  13  14  13  17  14   7  14  15   8   3   7   9  11  143545
     16   7  15  14   9   4   6  12   8   5  18  19  20  21  19  22  23  20  23  213546
     19  24  25  22  26  23  20  25  23  21  19  22  23  24  26  25  22  26  26  233547
     20  23  25  23  21  27  28  29  28  30  29  28  31  30  29  28  30  31  30  293548
     32  33  33  33  33  34                                                      3549
LA:1111141114514543111411414331664336322882224543384143343222222737322721111411454355O
   3114143363228224334322272111414314322232114321                              3551
QA:  0 0 0 0                                        1                           3552
     1 0 0 0          -1          -1           3     3                           3553
     1 1 0 0          -2          -2           4     9                           3554
     1 1 1 0          -3          -3           5    20                           3555
     1 1 1 1          -3          -6          12    75                           3556
     2 1 0 0           0          -1           2     9                           3557
     2 1 1 0          -1          -3           5    60                           3558
     2 1 1 1          -1          -2           3    75                           3559
     2 2 1 1           0          -4           6   225                           3560
     3 1 0 0           0           0           1     9                           3561
     3 1 1 0          -1          -1           5   180                           3562
     3 1 1 1          -1          -1           3   300                           3563
     3 2 1 0           1          -1           5   180                           3564
     3 2 1 1           0          -1           3   450                           3565
     3 2 2 1           1          -1           3   300                           3566
     3 3 1 1           0           0           1   225                           3567
     4 3 2 1           0           0           1   225                           3568
     5 0 0 0           0           0          16     9                           3569
     5 1 0 0          -4          -4          20    45                           3570
     5 1 1 0          -4          -4          12    75                           3571
     5 1 1 1         -48         -48         112  1575                           3572
     5 2 1 0           0          -8          24   225                           3573
     5 2 1 1          -8         -24          56  1575                           3574
     5 3 1 0           0           0          16   225                           3575
     5 3 1 1          -4          -4          28  1575                           3576
     5 3 2 1           4          -4          28  1575                           3577
     5 5 0 0           0           0         256   225                           3578
     5 5 1 0         -64         -64         448  1575                           3579
     5 5 1 1        -256        -256        1024 11025                           3580
     5 5 2 1           0         -64         256  3675                           3581
     5 5 3 1           0           0          64  1225                           3582
     5 5 5 0           0           0        1024  1225                           3583
     5 5 5 1        -256        -256        2304 11025                           3584
     5 5 5 5           0           0       65536 99225                           3585
KB:   1   1   1   1   2   3   4   5   4   6   7   8   9   8  10   4   3   4   5  63586
     11  11  12  12  13   6   7   8   7  10   5   4   3   4   6  14  15  14  15 163587
     12  11  11  12  13   7   8   7   6  10   4   5   4   3   6  11  12  12  11 133588
     15  14  15  14  16  12  12  11  11  13   8   9   8   7  10  17  17  17  17 183589
     19  20  21  20  22  20  19  20  21  22  21  20  19  20  22  20  21  20  19 223590
     23  23  23  23  24                                                      3591
LB:1432111181111815444415151544443633311331646443633322722828222442237373227221432135 92
   1118154444363332272214321                                                3593
QB:    0 0 1           0           1          -1           1                   3594
       0 0 5           0           0           0           1                   3595
       1 0 1           0           1          -1           3                   3596
       1 0 2          -1           1           2           6                   3597
       1 0 3           0          -1           1           6                   3598
       1 0 5           0          -4           4           9                   3599
       1 1 1           0           1          -1           6                   3600
       1 1 2          -2           1           3          18                   3601
       1 1 3           0          -1           1          18                   3602
       1 1 5           0          -4           4          15                   3603
       2 1 1          -1           2          -3          36                   3604
       2 1 3           1          -2           1          36                   3605
       2 1 5           4          -8           0          45                   3606
       3 1 1           0           1          -1          36                   3607
       3 1 2           0           1           1          36                   3608
```

```
          3 1 5          0          0          0          1                    3609
          5 0 1          0          4         -4          9                    3610
          5 0 5          0          0          0          1                    3611
          5 1 1          0          2         -2         15                    3612
          5 1 2         -2          4          6         45                    3613
          5 1 3          0         -4          4         45                    3614
          5 1 5          0        -32         32        225                    3615
          5 5 1          0         64        -64        225                    3616
          5 5 5          0          0          0          1                    3617
KC:   1   2   1   3   2   1   2   3   2   1   4   4   4   4   5               3618
LC:1141438432143 21                                                          3619
QC:       1 1         -4          3          3         -4         12          3620
          2 1          4         -3          3         -2         12          3621
          3 1          2         -3         -3          2         12          3622
          5 1          0         -4         -4          0          9          3623
          5 5       -128          0          0       -128         45          3624
```

APPENDIX B

NSF = 6

```
KA:   1   2   3   4   5   2   6   7   8   3   7   9   4   8   5   2   6   7   8   63625
     10  11   7  11   8   3   7   9   7  11   9   4   8   8   5  12  13  14  15  133626
     16  17  14  17  15  18  19  20  19  21  20  22  23  23  24  25  26  27  26  283627
     27  29  30  30  31  32  33  33  34  35  12  16  22  24  13  19  23  14  20  153628
     13  19  23  16  21  17  14  20  17  15  36  37  38  39  40  41  37  42  40  383629
     43  44  45  46  47  25  29  31  26  30  27  26  30  28  27  43  46  45  44  483630
     32  34  33  33  47  35  12  13  14  15  18  19  20  22  23  24  13  16  17  193631
     21  23  14  17  20  15  36  39  41  37  40  38  37  40  42  38  43  45  44  463632
     47  36  37  38  37  42  38  39  40  40  41  49  50  50  50  51  43  46  44  453633
     51  47  25  26  27  29  30  31  26  28  30  27  43  45  46  44  48  43  44  463634
     45  51  46  32  33  34  33  47  47  35                                      3635
QA:  0 0 0 0       1         1                                                  3636
     1 0 0 0       0         1                                                  3637
     1 1 0 0       1        30                                                  3638
     1 1 1 0       1        70                                                  3639
     1 1 1 1       1       105                                                  3640
     2 1 0 0      -1       180                                                  3641
     2 1 1 0      -1       630                                                  3642
     2 1 1 1      -1      1260                                                  3643
     2 2 1 1       1      3150                                                  3644
     3 2 1 0       1      1260                                                  3645
     3 2 1 1       1     18900                                                  3646
     4 0 0 0       1         3                                                  3647
     4 1 0 0       0         1                                                  3648
     4 1 1 0       1       105                                                  3649
     4 1 1 1       1       315                                                  3650
     4 2 1 0      -1       315                                                  3651
     4 2 1 1      -1      1890                                                  3652
     4 3 0 0      -1        45                                                  3653
     4 3 1 0       0         1                                                  3654
     4 3 1 1      -1      1890                                                  3655
     4 3 2 1       1      4725                                                  3656
     4 3 3 0       1       315                                                  3657
     4 3 3 1      -1      9450                                                  3658
     4 3 3 3       0         1                                                  3659
     4 4 0 0       3        45                                                  3660
     4 4 1 0       0         1                                                  3661
     4 4 1 1       4       945                                                  3662
     4 4 2 1      -1       525                                                  3663
     4 4 3 0      -4       315                                                  3664
     4 4 3 1       1      4725                                                  3665
     4 4 3 3       2      1575                                                  3666
     4 4 4 0       4        35                                                  3667
     4 4 4 1       3         1                                                  3668
     4 4 4 3      -4       525                                                  3669
     4 4 4 4     128      1575                                                  3670
     5 4 0 0       4        45                                                  3671
     5 4 1 0      -2       315                                                  3672
     5 4 1 1       1       945                                                  3673
     5 4 2 0       0         1                                                  3674
     5 4 2 1      -1      4725                                                  3675
     5 4 2 2       2       945                                                  3676
     5 4 3 1       2      4725                                                  3677
     5 4 4 0       4       105                                                  3678
     5 4 4 1      -4      1575                                                  3679
     5 4 4 2       0         1                                                  3680
     5 4 4 3     -16      4725                                                  3681
     5 4 4 4      32      1575                                                  3682
     5 5 4 4      64      4725                                                  3683
     6 5 4 0       8       315                                                  3684
     6 5 4 1      -8      4725                                                  3685
     6 5 4 4      16      1575                                                  3686
KB:  1   1   1   2   2   2   3   4   4   5   6   5   7   8   8   9  10   9   4  33687
     4   5   5   6  11  11  12  13  14  14   8   7   8   9   9  10   4   4   3  63668
     5   5  11  12  11  14  14  13  12  11  11  14  13  14   8   8   7  10   9  93689
    15  15  16  17  18  18  19  20  21  22  23  24  20  19  21  22  24  23  25 253690
```

```
     26  27  28  28  29  29  30  31  32  32  16  15  15  18  17  18  26  25  25  283691
     27  28  21  19  20  24  22  23  21  20  19  23  22  24  33  34  33  35  35  363692
     30  29  29  32  31  32  15  16  19  18  18  17  19  21  20  24  23  22  25  263693
     25  28  28  27  20  21  19  23  24  22  34  33  33  35  36  35  33  33  34  363694
     35  35  29  30  29  32  32  31                                                3695
LB:132132116116116116433433141114433433252252622262335533252252141114111111444444143696
   111414111433553333553333333335555551151513355336222626666666222622222222424226333697
   36622262                                                                      3698
QC:     0 0 1          0          1          6                                   3699
        0 0 4         -2          2          3                                   3700
        1 0 1          0          1         15                                   3701
        1 0 2          1          0         30                                   3702
        1 0 4         -2         -1         30                                   3703
        1 0 5          0          1         30                                   3704
        1 1 1          0         13        420                                   3705
        1 1 2          1          0        140                                   3706
        1 1 4         -5          1        105                                   3707
        1 1 5          0         -1        105                                   3708
        2 1 1          0         -1        280                                   3709
        2 1 3         11        -11       2520                                   3710
        2 1 4          2         -2        315                                   3711
        2 1 5         -5          4        630                                   3712
        4 0 1          0          1         10                                   3713
        4 0 3         -1          1         30                                   3714
        4 0 4         -4          4         15                                   3715
        4 0 5          2         -4         15                                   3716
        4 1 1          0          2        105                                   3717
        4 1 2          4          0        315                                   3718
        4 1 3         -2          2        315                                   3719
        4 1 4         -6         -4        315                                   3720
        4 1 5         -2          4        315                                   3721
        4 1 6          6         -8        315                                   3722
        4 3 1          0         -1        126                                   3723
        4 3 3          1         -1        210                                   3724
        4 3 4          2         -2        105                                   3725
        4 3 5         -4          6        315                                   3726
        4 4 1          0          4         63                                   3727
        4 4 3         -4          4        105                                   3728
        4 4 4        -16         16        105                                   3729
        4 4 5         32        -48        315                                   3730
        5 4 1          0          2        315                                   3731
        5 4 2         -2          0         63                                   3732
        5 4 4        -16         24        315                                   3733
        5 4 6         16          0        315                                   3734
KC:  1   2   1   2   2   1   3   3   4   5   4   3   3   6   5   3   4   3   6  63735
     5                                                                          3736
LC:113632141133513622432                                                        3737
QC:     1 1          0          0          0          1          4               3738
        2 1          0          0          1          0         12               3739
        4 1          0          0         -1          0          3               3740
        4 3          0          0          0          0          1               3741
        4 4          2         -1         -1          2          3               3742
        5 4          0          1          1         -2          3               3743
```

Blank card for NSF = 7

3744

APPENDIX B

NSF = 8

```
KA:   1   2   3   4   5   2   6   7   5   3   7   9   4   8   5   2  10  11  12  63745
     13  14   7  15   8   3  11  16   7  14   9   4  12   8   5   2   6   7   8 103746
     13  15  11  14  12   6  13  14  13  17  14   7  14  15   8   3   7   9  11 143747
     16   7  15  14   9   4   8  12   8   5  18  19  20  21  19  22  23  20  23 213748
     24  25  26  27  28  29  30  31  32  33  24  27  29  25  28  26  34  35  35 363749
     30  32  31  36  33  37  38  39  35  40  39  41  42  43  44  41  43  42  45 443750
     46  47  47  48  48  49  18  24  30  33  19  27  32  20  29  21  19  25  31 223751
     28  23  20  26  23  21  24  34  36  25  35  26  27  35  28  29  30  36  31 323752
     33  50  51  52  53  54  55  51  56  54  52  57  58  59  58  60  61  62  63 643753
     65  66  37  41  44  38  43  39  38  42  40  39  41  45  42  43  44  61  64 633754
     62  65  67  46  48  47  47  48  66  49  18  24  30  33  24  34  36  30  36 333755
     19  25  31  27  35  32  20  26  29  21  19  27  32  25  35  31  22  28  28 233756
     20  29  26  23  21  68  69  70  69  71  70  69  72  73  70  69  73  72  71 703757
     74  75  75  76  76  77  50  57  60  51  58  52  53  59  54  55  51  58  56 543758
     52  78  79  80  80  79  81  61  65  62  63  64  82  66  37  41  44  41  45 443759
     38  42  43  39  38  43  42  40  39  74  76  76  75  75  83  61  65  64  63 623760
     81  67  46  48  48  47  47  77  66  49  18  19  20  21  24  27  29  30  32 333761
     24  25  26  34  35  36  30  31  36  33  19  22  23  25  28  31  27  28  35 323762
     20  23  26  29  21  50  53  55  51  54  52  57  59  58  60  51  54  56  58 523763
     61  63  62  65  64  66  63  69  70  69  73  70  69  72  71  70  69  71  72 733764
     70  78  80  80  79  79  82  74  76  75  75  76  81  77  50  51  52  57  58 603765
     51  56  58  52  53  54  59  54  55  78  80  79  79  80  81  76  79  79  80 803766
     84  81  61  64  65  62  63  81  82  66  37  38  39  41  43  44  41  42  45 443767
     38  40  42  43  39  61  63  64  65  62  67  74  75  76  76  75  81  83  61 623768
     65  64  63  82  81  67  46  47  48  48  47  66  77  66  49                  3769
LA:111114111451454311141141433166433632288222454338414334322222273732227211115115553770
   111111111155555511515555511151511115551511515146664664646664646666446444464444443771
   411111161661116111111114664646664644445666614646464333373377733373733737337733733772
   737771115153113735371151514444444447747711177144444143337373373737373373717777737773
   743373737328662222223882628888228222882822282222555225222222555555288426646622243774
   222662246464643863833333333333355335484481433333343288222882822282222222228282228833775
   88838328282222                                                                      3776
QA:   0 0 0 0                                          1                              3777
      1 0 0 0                -1          -1          -3           9                   3778
      1 1 0 0                -2          -2           6          45                   3779
      1 1 1 0               -55         -55          33        2100                   3780
      1 1 1 1              -560        -560         732       33075                   3781
      2 1 0 0                 0           1           2          45                   3782
      2 1 1 0                23           9         -95        6300                   3783
      2 1 1 1                41          58          21       33075                   3784
      2 2 1 1                 0         -16         106       33075                   3785
      3 1 0 0                 0           0           1          15                   3786
      3 1 1 0                 5           5         -93        6300                   3787
      3 1 1 1                35          35         327      132300                   3788
      3 2 1 0                 7          -7         -65        6300                   3789
      3 2 1 1               -14           5         161       66150                   3790
      3 2 2 1                17         -17         339      132300                   3791
      3 3 1 1                 0           0          19        6615                   3792
      4 3 2 1                 0           0          71       33075                   3793
      5 0 0 0                 0          -4          12           9                   3794
      5 1 0 0                -2           0          -6          45                   3795
      5 1 1 0               -13         -29          75        1575                   3796
      5 1 1 1              -192        -176          24       33075                   3797
      5 2 1 0                 0          10          26        1575                   3798
      5 2 1 1                44          36        -204       33075                   3799
      5 3 0 0                 0           2          -8          45                   3800
      5 3 1 0                 4          -6          46        1575                   3801
      5 3 1 1                -3          35        -213       33075                   3802
      5 3 2 0                -7          -3          39        1575                   3803
      5 3 2 1                 7          -1        -153       33075                   3804
      5 3 2 2               -10          48        -216       33075                   3805
      5 3 3 0                 3          -7          53        1575                   3806
      5 3 3 1               -12          22        -200       33075                   3807
      5 3 3 2                12           8        -176       33075                   3808
      5 3 3 3                 6          40        -192       33075                   3809
      5 4 3 0                 0         -10          46        1575                   3810
```

	5	4	3	1		-12		22		-176		33075									3811
	5	4	3	3		-7		27		-195		33075									3812
	5	5	0	0		0		-16		32		45									3813
	5	5	1	0		-36		20		-108		1575									3814
	5	5	1	1		-80		-352		768		33075									3815
	5	5	2	1		0		16		48		6615									3816
	5	5	3	0		-4		52		-148		1575									3817
	5	5	3	1		56		-116		492		33075									3818
	5	5	3	2		-60		-80		432		33075									3819
	5	5	3	3		24		-144		544		33075									3820
	5	5	4	3		0		-144		520		33075									3821
	5	5	5	0		0		-48		80		175									3822
	5	5	5	1		-448		416		-1344		33075									3823
	5	5	5	3		-56		808		-1848		33075									3824
	5	5	5	5		0		-1024		1536		4725									3825
	6	5	0	0		4		-4		20		45									3826
	6	5	1	0		-24		16		-112		1575									3827
	6	5	1	1		64		-80		480		33075									3828
	6	5	2	0		4		-4		-84		1575									3829
	6	5	2	1		16		16		336		33075									3830
	6	5	2	2		112		-112		528		33075									3831
	6	5	3	1		68		-68		444		33075									3832
	6	5	4	0		-16		16		-112		1575									3833
	6	5	4	1		68		-44		420		33075									3834
	6	5	4	2		20		-20		396		33075									3835
	6	5	4	4		44		-44		444		33075									3836
	6	5	5	0		48		-112		336		1575									3837
	6	5	5	1		-224		288		-1152		33075									3838
	6	5	5	2		64		96		-864		33075									3839
	6	5	5	3		-160		336		-1200		33075									3840
	6	5	5	4		-112		288		-1152		33075									3841
	6	5	5	5		448		-1728		4032		33075									3842
	6	6	5	5		256		-256		1024		11025									3843
	7	5	0	0		0		0		16		45									3844
	7	5	1	0		-4		-4		-92		1575									3845
	7	5	1	1		-4		-8		408		33075									3846
	7	5	2	1		0		28		348		33075									3847
	7	5	3	1		0		0		376		33075									3848
	7	5	3	2		-24		0		352		33075									3849
	7	5	5	0		0		-16		80		525									3850
	7	5	5	1		-56		88		-840		33075									3851
	7	5	5	3		0		144		-896		33075									3852
	7	5	5	5		0		-128		384		4725									3853
	7	6	5	0		32		0		224		1575									3854
	7	6	5	1		-112		0		-864		33075									3855
	7	6	5	2		-64		-48		-816		33075									3856
	7	6	5	5		224		-288		2016		33075									3857
	7	6	6	5		64		0		256		3675									3858
	7	7	5	5		0		0		256		4725									3859
	8	7	6	5		0		0		64		1225									3860
KB:	1	1	1	1	2	2	2	2	3	4	5	4	6	7	7	6	8	9	10		93861
	11	12	12	11	4	3	4	5	6	6	7	7	13	13	14	14	15	16	17		163862
	9	8	9	10	11	11	12	12	5	4	3	4	7	6	6	7	13	19	18		193863
	20	20	20	20	14	13	13	14	16	15	16	17	10	9	8	9	12	11	11		123864
	4	5	4	3	7	7	6	6	13	14	14	13	16	17	16	15	19	18	19		183865
	20	20	20	20	14	14	13	13	17	16	15	16	9	10	9	8	12	12	11		113866
	21	21	22	22	23	24	25	24	26	27	28	29	30	31	32	33	27	26	29		263867
	30	33	32	31	34	35	36	37	38	39	40	41	35	34	37	36	38	41	40		393868
	42	42	43	43	44	45	46	45	22	21	21	22	24	23	24	25	36	35	34		373869
	39	38	41	40	29	26	27	28	33	30	31	32	28	27	26	29	31	30	33		323870
	37	34	35	36	41	38	39	40	47	46	47	49	50	50	51	51	43	42	42		433871
	45	44	45	46	22	22	21	21	25	24	23	24	36	37	34	35	40	41	38		393872
	37	36	35	34	40	39	38	41	28	29	26	27	32	33	30	31	29	28	27		283873
	32	31	30	33	52	52	52	52	53	54	53	54	49	47	48	47	51	50	50		513874
	43	43	42	42	45	45	44	45	21	22	22	21	24	25	24	23	26	29	28		273875
	33	32	31	30	35	36	37	34	39	40	41	38	34	37	36	35	41	40	39		383876
	27	28	29	26	31	32	33	30	48	47	49	47	50	51	51	50	52	52	52		523877
	54	53	54	53	47	49	47	48	51	51	50	50	42	43	43	42	45	46	45		443878

APPENDIX B

```
LB:1432143211181188111811885444544515151115544454453633663311331638664646444363366333879
   2272277282822282244254723737373322722772151511151111111155555555111111115555555553880
   1515111564646446666666664444444466666666444444441161161664646444373737333333333333881
   7777777733333333777777771537113344477474373737338282228288888882222222288888883882
   2222222225225252646224428333838822822823883
QB:      0 0 1          0          1         -1          3                                3884
         0 0 5         -4          4          0          3                                3885
         1 0 1          0          2         -2         15                                3886
         1 0 2          3         -3         -8         90                                3887
         1 0 3          0          1         -1         30                                3888
         1 0 5         -3         -7         10         45                                3889
         1 0 6         -5          0         -7         45                                3890
         1 1 1          0         13        -13        210                                3891
         1 1 2         48         27        -43       3150                                3892
         1 1 3          0          1         -1        210                                3893
         1 1 5       -141         64        119       1575                                3894
         1 1 6         23        -21         64       1575                                3895
         2 1 1         -3        -72         43       6300                                3896
         2 1 3         27        -68         27       6300                                3897
         2 1 5          3          4          0        175                                3898
         2 1 6         -3         35         39       1575                                3899
         2 1 7        -23        -36          0       1575                                3900
         3 1 1          0         -1          1        420                                3901
         3 1 2          0         41         41       6300                                3902
         3 1 5        -13         46        -14       1575                                3903
         5 0 1         -3         13        -10         45                                3904
         5 0 3         -5          7          0         45                                3905
         5 0 5         -3          3          0         15                                3906
         5 0 6          4         -4          4          9                                3907
         5 0 7          0         -3          0         15                                3908
         5 1 1         -9         86       -119       3150                                3909
         5 1 2         33        -96       -133       3150                                3910
         5 1 3         17        -16        -21       3150                                3911
         5 1 4        -25         26         -7       3150                                3912
         5 1 5          2       -100        126       1575                                3913
         5 1 6        -10          8        -14        225                                3914
         5 1 7         18        100         14       1575                                3915
         5 1 8          6         -8         14        225                                3916
         5 3 1          9        -76         49       3150                                3917
         5 3 2          9       -104        -77       3150                                3918
         5 3 3         23        -64         21       3150                                3919
         5 3 4         23        -36          7       3150                                3920
         5 3 5         82       -120         14       1575                                3921
         5 3 6         -3         12        -16        225                                3922
         5 3 7         -2        120        -14       1575                                3923
         5 3 8         -3          8         16        225                                3924
         5 5 1       -124        320       -252       1575                                3925
         5 5 3       -164        240        -28       1575                                3926
         5 5 5        -32         32          0        105                                3927
         5 5 6         64        -80         48        225                                3928
         5 5 7         32        -96          0        315                                3929
         6 5 1          2          8          2        225                                3930
         6 5 2          0         22         22        225                                3931
         6 5 4          0          8          8        225                                3932
         6 5 5        -32         40        -24        225                                3933
         6 5 7         -8        -40        -16        225                                3934
         7 5 1        -16         20        -28       1575                                3935
         7 5 5        -16         48          0        315                                3936
         7 5 6         16          0         32        225                                3937
KC:   1   2   1   3   2   1   2   3   2   1   4   4   5   5   6   5   4   4   5   73938
      6   5   5   4   4   8   7   6   4   5   5   4   7   8   7   6                    3939
LC:1141438432151516464143737143828254 32                                             3940
QC:      1 1        -104         85         85        -104        180                 3941
         2 1         -56         15        -15         -34        180                 3942
         3 1         -46         35         35         -46        180                 3943
         5 1          40        -25          5          -3         45                 3944
         5 3          20         -5         -5           3         45                 3945
         5 5         -80          0          0         -24         45                 3946
         6 5           0         -4         -4           0          9                 3947
         7 5         -40          0          0          24         45                 3948
```

APPENDIX B

NSF = 9

KA:																			
	1	2	3	4	5	2	6	7	8	3	7	9	4	8	5	2	10	11	12 63949
	13	14	7	15	8	3	11	16	7	14	9	4	12	8	5	2	6	7	6 103950
	13	15	11	14	12	6	13	14	13	17	14	7	14	15	8	3	7	9	11 143951
	16	7	15	14	9	4	8	12	8	5	18	19	20	21	19	22	23	20	23 213952
	24	25	26	27	28	29	30	31	32	33	24	27	29	25	28	26	34	35	35 363953
	30	32	31	36	33	37	38	39	38	40	39	41	42	43	44	41	43	42	45 443954
	46	47	47	48	48	49	18	24	30	33	19	27	32	20	29	21	19	25	31 223955
	28	23	20	26	23	21	24	34	36	25	35	26	27	35	28	29	30	36	31 323956
	33	50	51	52	53	54	55	51	56	54	52	57	58	59	58	60	61	62	63 643957
	65	66	37	41	44	38	43	39	33	42	40	39	41	45	42	43	44	61	64 633958
	62	65	67	46	45	47	47	48	66	49	18	24	30	33	24	34	36	30	36 333959
	19	25	31	27	35	32	20	26	29	21	19	27	32	25	35	31	22	28	28 233960
	20	29	26	23	21	65	69	70	69	71	70	69	72	73	70	69	73	72	71 703961
	74	75	75	76	76	77	50	57	60	51	58	52	53	59	54	55	51	58	56 543962
	52	78	79	80	80	79	81	61	65	62	63	64	82	66	37	41	44	41	45 443963
	38	42	43	39	33	43	42	40	39	74	76	76	75	75	83	61	65	64	63 623964
	81	67	46	48	48	47	47	77	66	49	18	19	20	21	24	27	29	30	32 333965
	24	25	26	34	35	36	30	31	36	33	19	22	23	25	28	31	27	28	35 323966
	20	23	26	29	21	50	53	55	51	54	52	57	59	58	60	51	54	56	58 523967
	61	63	62	65	64	66	68	69	70	69	73	70	69	72	71	70	69	71	72 733968
	70	78	80	80	79	79	82	74	76	75	75	76	81	77	50	51	52	57	58 603969
	51	56	58	52	53	54	59	54	55	78	80	79	79	80	81	78	79	79	80 803970
	84	81	61	64	65	62	63	81	32	66	37	38	39	41	43	44	41	42	45 443971
	38	40	42	43	39	61	63	64	65	62	67	74	75	76	76	75	81	83	61 623972
	65	64	63	82	61	67	45	47	46	48	47	66	77	66	49	85	86	87	88 863973
	89	90	87	90	88	86	91	92	89	93	90	87	92	90	88	86	89	90	91 933974
	92	89	93	93	90	87	90	92	90	88	94	95	96	95	97	96	98	99	100 1013975
	98	100	99	102	101	103	104	104	105	105	106	94	98	101	95	100	96	95	99 973976
	96	98	102	99	100	101	107	108	109	108	110	111	103	105	104	104	105	111	106 943977
	98	101	98	102	101	95	99	100	96	95	100	99	97	96	112	113	113	113	113 1143978
	107	110	106	109	108	115	111	103	105	105	104	104	114	111	106	94	95	96	98 1003979
	101	98	99	102	101	95	97	99	100	96	107	109	108	110	108	111	112	113	113 1133980
	113	115	114	107	108	110	103	109	115	115	111	103	104	105	105	104	111	114	111 1063981
	116	117	118	117	119	118	117	120	119	118	117	119	120	119	118	121	122	122	123 1233982
	124	121	123	122	122	123	125	124	121	123	123	122	122	126	125	124	121	122	123 1233983
	122	125	126	125	124	127	128	128	128	128	129	129	129	129	130				3984

LA:
1111411114514543111411414331664336322862224543384143343222222737322272111151115553985
1111111111555555511515555511151511115551511515146664664646666464666644644446444443986
4111111516611116111111146646466464444466666146464643333733777333737337373377733733987
7377711151531137353711515144444444444774771117714444414333737337373737337371777773968
74337373737888222222288623288882282228828222822225552252222225555555286424664622243989
2226622464646643886383333333833333553354844814333333343288622288282228222222228282222883990
8883832828222821111411454311414336322822433432227211151511115551511515146646466643991
644444111611464646433373733737373711537144447143373737323822283282228222522528463992
2243833334328282228211141431432223211515146466414337371432828222321143214321 3993

QA:									
	0 0 0 0				1	3994
	1 0 0 0	-1	-1	-3	9	3995
	1 1 0 0	-2	-2	6	45	3996
	1 1 1 0	-55	-55	33	2100	3997
	1 1 1 1	-560	-560	732	33075	3998
	2 1 0 0	0	1	2	45	3999
	2 1 1 0	23	9	-95	6300	4000
	2 1 1 1	41	58	21	33075	4001
	2 2 1 1	0	-16	106	33075	4002
	3 1 0 0	0	0	1	15	4003
	3 1 1 0	5	5	-93	6300	4004
	3 1 1 1	35	35	327	132300	4005
	3 2 1 0	7	-7	-65	6300	4006
	3 2 1 1	-14	5	161	66150	4007
	3 2 2 1	17	-17	339	132300	4008
	3 3 1 1	0	0	19	6615	4009
	4 3 2 1	0	0	71	33075	4010
	5 0 0 0	0	-4	12	9	4011
	5 1 0 0	-2	0	-6	45	4012
	5 1 1 0	-13	-29	75	1575	4013
	5 1 1 1	-192	-176	24	33075	4014

5	2	1	0	0	10	26	1575	4015
5	2	1	1	44	36	-204	33075	4016
5	3	0	0	0	2	-8	45	4017
5	3	1	0	4	-6	46	1575	4018
5	3	1	1	-3	35	-213	33075	4019
5	3	2	0	-7	-3	39	1575	4020
5	3	2	1	7	-1	-153	33075	4021
5	3	2	2	-10	48	-216	33075	4022
5	3	3	0	3	-7	53	1575	4023
5	3	3	1	-12	22	-200	33075	4024
5	3	3	2	12	8	-176	33075	4025
5	3	3	3	6	40	-192	33075	4026
5	4	3	0	0	-10	46	1575	4027
5	4	3	1	-12	22	-176	33075	4028
5	4	3	3	-7	27	-195	33075	4029
5	5	0	0	0	-16	32	45	4030
5	5	1	0	-36	20	-108	1575	4031
5	5	1	1	-80	-352	768	33075	4032
5	5	2	1	0	16	48	6615	4033
5	5	3	0	-4	52	-148	1575	4034
5	5	3	1	56	-116	492	33075	4035
5	5	3	2	-60	-80	432	33075	4036
5	5	3	3	24	-144	544	33075	4037
5	5	4	3	0	-144	520	33075	4038
5	5	5	0	0	-48	80	175	4039
5	5	5	1	-448	416	-1344	33075	4040
5	5	5	3	-56	808	-1848	33075	4041
5	5	5	5	0	-1024	1536	4725	4042
6	5	0	0	4	-4	20	45	4043
6	5	1	0	-24	16	-112	1575	4044
6	5	1	1	64	-80	480	33075	4045
6	5	2	0	4	-4	-84	1575	4046
6	5	2	1	16	16	336	33075	4047
6	5	2	2	112	-112	528	33075	4048
6	5	3	1	68	-68	444	33075	4049
6	5	4	0	-16	16	-112	1575	4050
6	5	4	1	68	-44	420	33075	4051
6	5	4	2	20	-20	396	33075	4052
6	5	4	4	44	-44	444	33075	4053
6	5	5	0	48	-112	336	1575	4054
6	5	5	1	-224	288	-1152	33075	4055
6	5	5	2	64	96	-864	33075	4056
6	5	5	3	-160	336	-1200	33075	4057
6	5	5	4	-112	288	-1152	33075	4058
6	5	5	5	448	-1728	4032	33075	4059
6	6	5	5	256	-256	1024	11025	4060
7	5	0	0	0	0	16	45	4061
7	5	1	0	-4	-4	-92	1575	4062
7	5	1	1	-4	-8	408	33075	4063
7	5	2	1	0	28	348	33075	4064
7	5	3	1	0	0	376	33075	4065
7	5	3	2	-24	0	352	33075	4066
7	5	5	0	0	-16	80	525	4067
7	5	5	1	-56	88	-840	33075	4068
7	5	5	3	0	144	-896	33075	4069
7	5	5	5	0	-128	384	4725	4070
7	6	5	0	32	0	224	1575	4071
7	6	5	1	-112	0	-864	33075	4072
7	6	5	2	-64	-48	-816	33075	4073
7	6	5	5	224	-288	2016	33075	4074
7	6	6	5	64	0	256	3675	4075
7	7	5	5	0	0	256	4725	4076
8	7	6	5	0	0	64	1225	4077
9	0	0	0	0	0	16	9	4078
9	1	0	0	-4	-4	-60	225	4079
9	1	1	0	-4	-4	92	1575	4080
9	1	1	1	-48	-48	-304	33075	4081
9	2	1	0	0	8	72	1575	4082
9	2	1	1	16	-8	-336	33075	4083

APPENDIX B

	9	3	1	0			0		0		16		315								4084
	9	3	1	1			4		4		-356		33075								4085
	9	3	2	1			12		-12		-308		33075								4086
	9	5	0	0			0		-32		160		225								4087
	9	5	1	0			-16		16		-176		1575								4088
	9	5	1	1			0		-128		832		33075								4089
	9	5	2	1			0		0		128		6615								4090
	9	5	3	0			0		32		-192		1575								4091
	9	5	3	1			32		-80		752		33075								4092
	9	5	3	2			-48		-64		704		33075								4093
	9	5	3	3			16		-96		800		33075								4094
	9	5	4	3			0		-112		752		33075								4095
	9	5	5	0			0		-64		192		525								4096
	9	5	5	1			-192		384		-1856		33075								4097
	9	5	5	3			-32		544		-2080		33075								4098
	9	5	5	5			0		-3072		7168		33075								4099
	9	6	5	0			64		-64		448		1575								4100
	9	6	5	1			-256		192		-1728		33075								4101
	9	6	5	2			-64		64		-1536		33075								4102
	9	6	5	4			-64		64		-576		11025								4103
	9	6	5	5			512		-1152		4608		33075								4104
	9	7	5	0			0		0		128		525								4105
	9	7	5	1			-32		-32		-1504		33075								4106
	9	7	5	5			0		-512		3584		33075								4107
	9	7	6	5			128		0		1152		11025								4108
	9	9	0	0			0		0		256		225								4109
	9	9	1	0			-64		-64		-2240		11025								4110
	9	9	1	1			0		0		1408		33075								4111
	9	9	2	1			0		64		1280		33075								4112
	9	9	3	1			0		0		64		1575								4113
	9	9	5	0			0		-256		1792		3675								4114
	9	9	5	1			-128		256		-2944		33075								4115
	9	9	5	3			0		128		-1024		11025								4116
	9	9	5	5			0		-2048		8192		33075								4117
	9	9	6	5			256		-256		2304		11025								4118
	9	9	7	5			0		0		2048		11025								4119
	9	9	9	0			0		0		1024		1225								4120
	9	9	9	1			-256		-256		-16128		99225								4121
	9	9	9	5			0		-4096		36864		99225								4122
	9	9	9	9			0		0		65536		99225								4123
KB:	1	1	1	1	2	2	2	2	3	4	5	6	5	7	8	8	7	9	10		114124
	12	11	13	14	14	13	15	5	4	5	6	7	7	8	8	9	16	16	17		174125
	18	19	20	19	21	11	10	11	12	13	13	14	14	15	6	5	4	5	8		74126
	7	8	9	22	23	22	23	24	24	24	24	25	17	16	16	17	19	18	19		204127
	21	12	11	10	11	14	13	13	14	15	5	6	5	4	8	8	7	7	9		164128
	17	17	16	19	20	19	18	21	23	22	23	22	24	24	24	24	25	17	17		164129
	16	20	19	18	19	21	11	12	11	10	14	14	13	13	15	26	26	27	27		284130
	29	30	29	31	32	33	34	35	36	37	38	39	40	33	32	35	34	36	39		384131
	37	40	41	42	43	44	45	46	47	48	49	42	41	44	43	45	48	47	46		494132
	50	50	51	51	52	53	54	53	55	27	26	26	27	29	28	29	30	31	43		424133
	41	44	46	45	48	47	49	35	32	33	34	39	36	37	38	40	34	33	32		354134
	37	36	39	38	40	44	41	42	43	48	45	46	47	49	56	57	56	58	59		594135
	60	60	61	51	50	50	51	53	52	53	54	55	27	27	26	26	30	29	28		294136
	31	43	44	41	42	47	48	45	46	49	44	43	42	41	47	46	45	48	49		344137
	35	32	33	38	39	36	37	40	35	34	33	32	38	37	36	39	40	62	62		624138
	62	63	64	63	64	65	58	56	57	56	60	59	59	60	61	51	51	50	50		544139
	53	52	53	55	26	27	27	26	29	30	29	28	31	32	35	34	33	39	38		374140
	36	40	42	43	44	41	46	47	48	45	49	41	44	43	42	48	47	46	45		494141
	33	34	35	32	37	38	39	36	40	57	56	58	56	59	60	60	59	61	62		624142
	62	62	64	63	64	63	65	56	58	56	57	60	60	59	59	61	50	51	51		504143
	53	54	53	52	55	66	66	66	66	67	67	67	67	68	69	70	71	70	72		734144
	73	72	74	70	69	70	71	72	72	73	73	74	71	70	69	70	73	72	72		734145
	74	70	71	70	69	73	73	72	72	74	75	75	76	76	77	78	79	78	80		764146
	75	75	76	78	77	78	79	80	76	76	75	75	79	78	77	78	80	75	76		764147
	75	78	79	78	77	80	81	81	81	81	82	82	82	82	83						4148
LB:	1432143211111811381111811881544454454151511151544454454543633663331133163816464644441																				49
	4363366333227227722828222822244254722373737333227227722151511151111111111555555415																				50
	5511111111115555555551515111516464644446666666664444444446666666664444444441161164151																				
	1616466464444373737333333333333777777777333333333777777777153711331444774744373734152																				

APPENDIX B

7333828222622886666888882222222228368888888222222222225225252284622442283333838382824153
2282214321432111181158154445445436336633322722277221515111516464644443737373338284154
222822143214321 4155

QB:	0 0 1	0	1	-1	3		4156
	0 0 5	-4	4	0	3		4157
	0 0 9	0	0	0	1		4158
	1 0 1	0	2	-2	15		4159
	1 0 2	3	-3	-8	90		4160
	1 0 3	0	1	-1	30		4161
	1 0 5	-3	-7	10	45		4162
	1 0 6	-5	0	-7	45		4163
	1 0 9	0	-4	4	45		4164
	1 1 1	0	13	-13	210		4165
	1 1 2	48	27	-43	3150		4166
	1 1 3	0	1	-1	210		4167
	1 1 5	-141	64	119	1575		4168
	1 1 6	23	-21	64	1575		4169
	1 1 9	0	-44	44	1575		4170
	2 1 1	-3	-72	43	6300		4171
	2 1 3	27	-68	27	6300		4172
	2 1 5	3	4	0	175		4173
	2 1 6	-8	35	39	1575		4174
	2 1 7	-23	-36	0	1575		4175
	2 1 9	-12	40	0	1575		4176
	3 1 1	0	-1	1	420		4177
	3 1 2	0	41	41	6300		4178
	3 1 5	-13	46	-14	1575		4179
	3 1 9	0	0	0	1		4180
	5 0 1	-3	13	-10	45		4181
	5 0 3	-5	7	0	45		4182
	5 0 5	-8	8	0	15		4183
	5 0 6	4	-4	4	9		4184
	5 0 7	0	-8	0	15		4185
	5 0 9	15	-32	0	45		4186
	5 1 1	-9	86	-119	3150		4187
	5 1 2	33	-96	-133	3150		4188
	5 1 3	17	-16	-21	3150		4189
	5 1 4	-25	26	-7	3150		4190
	5 1 5	2	-100	126	1575		4191
	5 1 6	-10	8	-14	225		4192
	5 1 7	18	100	14	1575		4193
	5 1 8	5	-8	14	225		4194
	5 1 9	-8	32	56	1575		4195
	5 3 1	9	-76	49	3150		4196
	5 3 2	9	-104	-77	3150		4197
	5 3 3	23	-64	21	3150		4198
	5 3 4	23	-36	7	3150		4199
	5 3 5	82	-120	14	1575		4200
	5 3 6	-8	12	-16	225		4201
	5 3 7	-2	120	-14	1575		4202
	5 3 8	-8	8	16	225		4203
	5 3 9	-32	112	0	1575		4204
	5 5 1	-124	320	-252	1575		4205
	5 5 3	-164	240	-23	1575		4206
	5 5 5	-32	32	0	105		4207
	5 5 6	64	-80	48	225		4208
	5 5 7	32	-96	0	315		4209
	5 5 9	192	-320	0	525		4210
	6 5 1	2	6	2	225		4211
	6 5 2	0	22	22	225		4212
	6 5 4	0	8	8	225		4213
	6 5 5	-32	40	-24	225		4214
	6 5 7	-8	-40	-16	225		4215
	6 5 9	0	-32	-32	225		4216
	7 5 1	-16	20	-28	1575		4217
	7 5 5	-16	48	0	315		4218
	7 5 6	16	0	32	225		4219
	7 5 9	64	0	0	1575		4220
	9 0 1	0	4	-4	45		4221

```
                9 0 5          -16          32           0          45                    4222
                9 0 9            0           0           0           1                    4223
                9 1 1            0          22         -22        1575                    4224
                9 1 2            6         -20         -34        1575                    4225
                9 1 3            0           4          -4         315                    4226
                9 1 5           32        -184          56        1575                    4227
                9 1 6          -72           0        -184        1575                    4228
                9 1 9            0         -32          32        1575                    4229
                9 5 1          -24         152        -112        1575                    4230
                9 5 3          -40          72           0        1575                    4231
                9 5 5          -96         160           0         525                    4232
                9 5 6           32         -32          64         225                    4233
                9 5 7          -32        -480           0        1575                    4234
                9 5 9          126        -384           0        1575                    4235
                9 9 1            0          64         -64        1575                    4236
                9 9 5         -256         768           0        1575                    4237
                9 9 9            0           0           0           1                    4238
KC:     1   2   1   3   2   1   2   3   2   1   4   4   5   5   6   5   4   4   5       74239
      6   5   5   4   4   8   7   6   4   5   5   4   7   8   7   6   9   9   9       94240
    10  10  10  10  11                                                                   4241
LC:1141438432151516464143737143828254321432143214321                                    4242
QC:             1 1          -104          85          85        -104         180       4243
                2 1           -56          15         -15         -34         180       4244
                3 1           -46          35          35         -46         180       4245
                5 1            40         -25           5          -3          45       4246
                5 3            20          -5          -5           3          45       4247
                5 5           -80           0           0         -24          45       4248
                6 5             0          -4          -4           0           9       4249
                7 5           -40           0           0          24          45       4250
                9 1            40         -20         -20          40          45       4251
                9 5           -80           0           0           0          45       4252
                9 9          -128           0           0        -128          45       4253
```

APPENDIX B

NSF = 10

KA:

1	2	3	4	5	2	6	7	8	3	7	9	4	8	5	2	6	7	8	64254
10	11	7	11	8	3	7	9	7	11	9	4	8	8	5	12	13	14	15	164255
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	364256
37	38	39	40	41	42	43	44	45	46	12	22	28	31	13	23	29	14	24	154257
16	25	30	17	26	18	19	27	20	21	47	48	49	50	51	52	53	54	55	564258
57	58	59	60	61	32	38	41	33	39	34	35	40	36	37	62	63	64	65	664259
42	45	43	44	67	46	12	16	19	21	22	25	27	28	30	31	13	17	20	234260
26	29	14	18	24	15	47	50	52	53	55	56	48	51	54	49	62	64	65	634261
67	47	53	56	48	54	49	50	59	51	52	68	69	69	69	70	57	60	58	594262
70	61	32	35	37	38	40	41	33	36	39	34	57	59	60	56	66	62	65	634263
64	70	66	42	44	45	43	61	67	46	12	16	19	21	13	17	20	14	18	154264
22	25	27	23	26	24	28	30	29	31	71	72	73	72	74	73	75	76	76	774265
78	79	80	81	82	83	84	85	86	87	88	84	89	87	85	90	91	92	93	944266
95	96	97	98	99	100	101	102	103	104	105	106	104	105	107	106	108	109	110	1114267
112	113	114	115	116	117	118	119	120	121	122	123	124	125	32	35	37	33	36	344268
38	40	39	41	73	80	79	81	126	95	98	97	96	127	128	119	120	122	121	1294269
130	131	42	44	43	45	82	100	125	46	12	22	28	31	16	25	30	19	27	214270
13	23	29	17	26	20	14	24	18	15	101	104	106	102	105	103	104	107	105	1064271
119	122	120	121	125	71	75	77	72	76	73	72	76	74	73	108	111	109	110	1234272
78	81	79	80	112	82	83	84	85	84	89	85	86	87	87	88	113	116	114	1154273
124	90	93	91	92	117	94	95	98	96	97	118	99	100	47	53	56	50	55	524274
48	54	51	49	103	110	109	111	129	90	93	92	91	132	127	113	114	116	115	1334275
134	130	62	65	64	63	112	99	124	67	32	38	41	35	40	37	33	39	36	344276
119	121	120	122	131	78	81	80	79	129	126	95	96	98	97	130	127	128	57	604277
59	58	123	94	118	66	42	45	44	43	125	82	100	61	46	12	13	14	15	224278
23	24	28	29	31	16	17	18	25	26	30	19	20	27	21	83	86	88	84	874279
85	84	87	89	85	95	97	98	96	100	101	104	106	104	107	106	102	105	105	1034280
113	115	116	114	118	119	121	122	120	124	125	71	72	73	75	76	77	72	74	764281
73	90	92	93	91	99	108	110	111	109	117	123	78	80	81	79	94	112	82	474282
50	52	48	51	49	53	55	54	56	90	92	91	93	127	113	116	115	114	134	1304283
108	109	111	110	132	133	129	97	59	58	60	94	118	123	61	47	48	49	53	544284
56	50	51	55	52	113	115	114	116	130	108	111	110	109	133	129	90	91	93	924285
134	132	127	68	69	69	69	117	117	117	70	62	63	65	64	124	112	99	70	674286
32	33	34	38	39	41	35	36	40	37	95	97	96	98	128	119	122	121	120	1304287
131	78	79	81	80	127	129	126	62	64	63	65	99	124	112	66	57	58	60	594288
118	123	94	70	66	42	43	45	44	100	125	82	67	61	46	135	136	137	138	1364289
139	140	137	140	138	136	139	140	139	141	140	137	140	140	138	142	143	144	145	1464290
147	148	149	150	151	152	153	154	155	156	142	148	151	143	149	144	145	150	146	1474291
157	158	159	160	161	152	155	153	154	162	156	142	145	147	148	150	151	143	146	1494292
144	157	159	160	158	162	157	160	155	159	163	161	152	154	155	153	161	162	156	1424293
145	147	143	146	144	148	150	149	151	164	165	165	166	167	168	169	170	169	171	1724294
173	174	175	175	176	177	178	152	154	153	155	167	172	178	156	142	148	151	145	1504295
147	143	149	146	144	173	175	174	175	178	164	166	165	165	176	167	168	169	169	1704296
177	171	172	157	160	159	158	176	171	177	162	152	155	154	153	178	167	172	161	1564297
142	143	144	148	149	151	145	146	150	147	168	170	169	169	172	173	175	175	174	1774298
178	164	165	166	165	171	176	167	157	159	158	160	171	177	176	161	157	158	160	1594299
177	176	171	163	162	152	153	155	154	172	178	167	162	161	156	179	180	181	180	1824300
181	180	182	182	181	183	184	185	186	187	183	166	184	185	188	187	183	185	186	1844301
188	188	187	183	185	184	186	189	190	191	187	183	186	185	184	191	189	190	188	1874302
183	184	186	185	190	191	189	188	188	187	192	193	193	193	194	194	194	194	194	1944303
195																			4304

QA:

0	0	0	0	1	1	4305
1	0	0	0	1	30	4306
1	1	0	0	19	1680	4307
1	1	1	0	31	6160	4308
1	1	1	1	743	240240	4309
2	1	0	0	11	6720	4310
2	1	1	0	1	2640	4311
2	1	1	1	569	3843840	4312
2	2	1	1	3293	100900800	4313
3	2	1	0	1	9240	4314
3	2	1	1	19	1411200	4315
4	0	0	0	3	40	4316
4	1	0	0	3	1120	4317
4	1	1	0	39	12320	4318
4	1	1	1	177	128128	4319

4	2	0	0	0	1	4320
4	2	1	0	73	246400	4321
4	2	1	1	129	915200	4322
4	2	2	0	-3	2464	4323
4	2	2	1	-321	6406400	4324
4	2	2	2	-411	640640	4325
4	3	0	0	9	2240	4326
4	3	1	0	1	7700	4327
4	3	1	1	57	582400	4328
4	3	2	0	-1	35200	4329
4	3	2	1	401	44844800	4330
4	3	2	2	-57	1601600	4331
4	3	3	0	1	12320	4332
4	3	3	1	-109	44844800	4333
4	3	3	2	-23	8968960	4334
4	3	3	3	-3	133040	4335
4	4	0	0	9	112	4336
4	4	1	0	27	6160	4337
4	4	1	1	549	320320	4338
4	4	2	0	81	30800	4339
4	4	2	1	1107	3203200	4340
4	4	2	2	729	1601600	4341
4	4	3	0	369	123200	4342
4	4	3	1	369	3203200	4343
4	4	3	2	4941	44844800	4344
4	4	3	3	711	4484480	4345
4	4	4	0	2187	49280	4346
4	4	4	1	81	29120	4347
4	4	4	2	12231	11211200	4348
4	4	4	3	81	50960	4349
4	4	4	4	13851	400400	4350
5	4	0	0	-9	448	4351
5	4	1	0	-117	123200	4352
5	4	1	1	-549	6406400	4353
5	4	2	0	-27	24640	4354
5	4	2	1	-1629	44844800	4355
5	4	2	2	-549	1281280	4356
5	4	3	0	-153	246400	4357
5	4	3	1	-1611	44844800	4358
5	4	3	2	-243	3449600	4359
5	4	3	3	-9	320320	4360
5	4	4	0	81	35200	4361
5	4	4	1	81	640640	4362
5	4	4	2	27	256256	4363
5	4	4	3	1809	22422400	4364
5	4	4	4	-729	560560	4365
5	5	4	0	-2349	246400	4366
5	5	4	1	-4509	8968960	4367
5	5	4	2	-3051	6406400	4368
5	5	4	3	-2997	8968960	4369
5	5	4	4	3159	1724800	4370
5	5	5	4	-729	128128	4371
6	5	4	0	81	61600	4372
6	5	4	1	1809	22422400	4373
6	5	4	4	729	8968960	4374
7	4	0	0	-9	320	4375
7	4	1	0	-81	24640	4376
7	4	1	1	-927	1281280	4377
7	4	2	1	-27	140140	4378
7	4	3	0	-9	7700	4379
7	4	3	1	-261	2242240	4380
7	4	3	3	-1413	22422400	4381
7	4	4	0	-729	246400	4382
7	4	4	1	-8181	6406400	4383
7	4	4	2	-1377	6406400	4384
7	4	4	3	-6723	44844800	4385
7	4	4	4	-729	114400	4386
7	5	0	0	9	224	4387
7	5	1	0	387	246400	4388

7	5	1	1	729	6406400	4389
7	5	2	0	27	12320	4390
7	5	2	1	171	1601600	4391
7	5	2	2	549	640640	4392
7	5	3	1	1611	22422400	4393
7	5	4	0	-81	15400	4394
7	5	4	1	-15471	44844800	4395
7	5	4	2	-2889	6406400	4396
7	5	4	3	-837	3449600	4397
7	5	4	4	79461	44844800	4398
7	5	5	0	729	49280	4399
7	5	5	1	16713	22422400	4400
7	5	5	2	27	29120	4401
7	5	5	3	3807	5605600	4402
7	5	5	4	-4617	1601600	4403
7	5	5	5	13851	1601600	4404
7	6	0	0	-9	1120	4405
7	6	1	0	27	61600	4406
7	6	1	1	279	1601600	4407
7	6	2	0	9	246400	4408
7	6	2	1	549	44844800	4409
7	6	2	2	-27	492800	4410
7	6	3	2	261	8968960	4411
7	6	4	0	729	246400	4412
7	6	4	1	1107	3203200	4413
7	6	4	2	891	4076800	4414
7	6	4	3	7263	44844800	4415
7	6	4	4	53217	44844800	4416
7	6	5	0	-81	123200	4417
7	6	5	1	-1809	44844800	4418
7	6	5	2	27	800800	4419
7	6	5	3	-513	4484480	4420
7	6	5	4	729	8968960	4421
7	6	5	5	-16767	22422400	4422
7	6	6	0	-1053	246400	4423
7	6	6	1	-4077	22422400	4424
7	6	6	2	-2403	8968960	4425
7	6	6	3	-81	3203200	4426
7	6	6	4	-23571	44844800	4427
7	6	6	5	34263	44844800	4428
7	6	6	6	-21141	11211200	4429
7	7	4	4	70713	11211200	4430
7	7	5	4	-729	246400	4431
7	7	5	5	4617	800800	4432
7	7	6	4	-21141	44844800	4433
7	7	6	5	-729	802400	4434
7	7	6	6	4617	5605600	4435
8	7	5	4	23571	22422400	4436
8	7	6	4	-729	4484480	4437
8	7	6	5	29889	44844800	4438
10	0	0	0	9	20	4439
10	1	0	0	3	560	4440
10	1	1	0	9	6160	4441
10	1	1	1	9	45760	4442
10	2	1	0	-3	123200	4443
10	2	1	1	9	246400	4444
10	3	2	1	159	11211200	4445
10	4	0	0	27	1120	4446
10	4	1	0	-27	12320	4447
10	4	1	1	171	640640	4448
10	4	2	0	81	61600	4449
10	4	2	1	27	862400	4450
10	4	2	2	-171	3203200	4451
10	4	3	0	243	123200	4452
10	4	3	1	-207	5605600	4453
10	4	3	2	657	22422400	4454
10	4	3	3	9	145600	4455
10	4	4	0	3159	123200	4456
10	4	4	1	243	1601600	4457

APPENDIX B

```
10 4 4 2      12379  22422400                                          4458
10 4 4 3      25029  22422400                                          4459
10 4 4 4       2187    200200                                          4460
10 5 4 0      -1053    123200                                          4461
10 5 4 1       -243    700700                                          4462
10 5 4 2        -81   1601600                                          4463
10 5 4 3      -5589  22422400                                          4464
10 5 4 4      19683  22422400                                          4465
10 5 5 4     -19683   5605600                                          4466
10 6 5 4       6561  11211200                                          4467
10 7 4 0       -243     24640                                          4468
10 7 4 1       -243    457600                                          4469
10 7 4 3      -4617  11211200                                          4470
10 7 4 4      -2187   2802800                                          4471
10 7 5 0       1053     61600                                          4472
10 7 5 1        243    407680                                          4473
10 7 5 2         81    800800                                          4474
10 7 5 4      -2187   1121120                                          4475
10 7 5 5       2187    400400                                          4476
10 7 6 0        -81     17600                                          4477
10 7 6 1        -81  11211200                                          4478
10 7 6 2        243   5605600                                          4479
10 7 6 4       2187   2038400                                          4480
10 7 6 5      -6561  22422400                                          4481
10 7 6 6     -37179  22422400                                          4482
1010 0 0         81       280                                          4483
1010 1 0         27     15400                                          4484
1010 1 1        513    800800                                          4485
1010 2 1        -81    509600                                          4486
1010 4 0        729     61600                                          4487
1010 4 1       -729    400400                                          4488
1010 4 2       2187   2242240                                          4489
1010 4 3       6561   5605600                                          4490
1010 4 4       2187    175175                                          4491
1010 5 4      -6561   1401400                                          4492
1010 7 4      -2187    431200                                          4493
1010 7 5       6561    700700                                          4494
1010 7 6      -6561   2242240                                          4495
101010 0       6561     30800                                          4496
101010 1       2187   2802800                                          4497
101010 4      19683   2802800                                          4498
101010 10     59049    350350                                          4499
KB:  1  1  1  2  2  2  2  2  2  3  4  5  5  6  7  8  8  7  6   94500
    10 11 11 12 13 14 14 13 12 15  5  4  5  8  6  7  6  8  7   94501
    16 16 17 18 19 20 18 20 19 21 11 10 11 14 12 13 12 14 13  154502
     5  5  4  7  8  6  7  6  8  9 16 17 16 19 20 18 20 19 18  214503
    17 16 16 20 18 19 19 18 20 21 11 11 10 13 14 12 13 12 14  154504
    22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40  414505
    42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60  614506
    62 63 64 65 66 67 68 69 70 71 24 22 23 27 25 26 30 28 29  314507
    54 52 53 57 55 56 60 58 59 61 34 32 33 37 35 36 40 38 39  414508
    44 42 43 47 45 46 50 48 49 51 72 73 74 75 76 77 78 79 80  814509
    64 62 63 67 65 66 70 68 69 71 23 24 22 26 27 25 29 30 28  314510
    43 44 42 46 47 45 49 50 48 51 53 54 52 56 57 55 59 60 58  614511
    33 34 32 36 37 35 39 40 38 41 73 74 72 76 77 75 79 80 78  814512
    74 72 73 77 75 76 80 78 79 81 63 64 62 66 67 65 69 70 68  714513
    23 22 24 28 30 29 25 27 26 31 43 42 44 48 50 49 45 47 46  514514
    33 32 34 38 40 39 35 37 36 41 53 52 54 58 60 59 55 57 56  614515
    82 82 83 84 85 86 84 86 85 87 88 89 88 90 91 92 91 90 92  934516
    94 95 95 96 97 98 98 97 96 99 63 62 64 68 70 69 65 67 66  714517
    24 23 22 29 28 30 26 25 27 31 54 53 52 59 58 60 56 55 57  614518
    44 43 42 49 48 50 46 45 47 51 34 33 32 39 38 40 36 35 37  414519
    95 94 95 98 96 97 96 98 97 99 83 82 82 86 84 85 85 84 86  874520
    88 88 89 92 90 91 92 91 90 93 74 73 72 79 78 80 76 75 77  814521
    64 63 62 69 68 70 66 65 67 71 22 24 23 30 29 28 27 26 25  314522
    32 34 33 40 39 38 37 36 35 41 52 54 53 60 59 58 57 56 55  614523
    42 44 43 50 49 48 47 46 45 51 89 88 88 91 92 90 90 92 91  934524
    95 95 94 97 98 96 97 96 98 99 82 83 82 85 86 84 86 85 84  874525
    73 72 74 78 80 79 75 77 76 81 72 74 73 80 79 78 77 76 75  814526
```

APPENDIX B

```
     62  64  63  70  69  68  67  66  65  71 100 100 100 101 101 101 101 101 101 101 1024527
    103 104 104 105 106 107 107 106 105 108 104 103 104 107 105 106 105 107 106 106 1084528
    104 104 103 106 107 105 106 105 107 108 109 110 111 112 113 114 115 116 117 117 1184529
    111 109 110 114 112 113 117 115 116 118 110 111 109 113 114 112 116 117 115 115 1184530
    110 109 111 115 117 116 112 114 113 118 111 110 109 116 115 117 113 112 114 114 1184531
    109 111 110 117 116 115 114 113 112 118 119 119 119 120 120 120 120 120 120 120 1214532
LB: 1321324561116111666111611166614333334443141111444143333344432522225552622222666245333
    339333555325222255211111111111111111111111111111111111111111111111111111111111333333333345334
    333333333333333333333333333333311111111113333333333222222222222222222222222222224535
    222222222222222222223333333333222222222224444444444444444444444444444444444444444536
    141111444111511155511161116661444444444455555555555555555555555555555555555555554537
    433333444333533355536333336663555555555555555555556666666666666666666666666666664538
    666666666624222244422522225552622222666244444444446666666666666666666613213245614539
    116111666143333344432522225552111111111133333333332222222222444444444455555555554540
    66666666661321324561                                                                4541
QB:      0 0 1          0          1          8                                        4542
         0 0 4         -3          3          8                                        4543
         0 010          0          0          1                                        4544
         1 0 1          0          4        105                                        4545
         1 0 2        -19          0       1680                                        4546
         1 0 4        -66         -3       1120                                        4547
         1 0 5          0         -3        224                                        4548
         1 0 6        -24         39       1120                                        4549
         1 010          0          9        560                                        4550
         1 1 1          0         17        960                                        4551
         1 1 2         -1          0        336                                        4552
         1 1 4        -18          3        560                                        4553
         1 1 5          0          0          1                                        4554
         1 1 6        -27         27       2240                                        4555
         1 110          0         -3        560                                        4556
         2 1 1          0          1        672                                        4557
         2 1 3         17        -17      19200                                        4558
         2 1 4       -105          3      44800                                        4559
         2 1 5        105       -105      44800                                        4560
         2 1 6        -63         81      44800                                        4561
         2 110         33        -33      22400                                        4562
         4 0 1          0         69       1120                                        4563
         4 0 2         39          0       1120                                        4564
         4 0 3          3         -3        224                                        4565
         4 0 4        -27         27        140                                        4566
         4 0 5        -27         81       1120                                        4567
         4 0 6        -81         54       1120                                        4568
         4 0 7        -27        -81       1120                                        4569
         4 0 8        -27         54       1120                                        4570
         4 0 9         54        -27        280                                        4571
         4 010         54       -135        560                                        4572
         4 1 1          0          3        224                                        4573
         4 1 2         -3          0       8960                                        4574
         4 1 3          3         -3       1280                                        4575
         4 1 4        -54          9       2240                                        4576
         4 1 5         -9         45       6960                                        4577
         4 1 6       -108         99       8960                                        4578
         4 1 7        108        -45       6960                                        4579
         4 1 8         -9         18       8960                                        4580
         4 1 9        108        -99       4480                                        4581
         4 110          0        -27       4480                                        4582
         4 2 1          0        117      44800                                        4583
         4 2 2         27          0       4480                                        4584
         4 2 3         -9          9       6400                                        4585
         4 2 4       -171        135      22400                                        4586
         4 2 5        -99        108       8960                                        4587
         4 2 6         -9         -9      44800                                        4588
         4 2 7         45       -108       8960                                        4589
         4 2 8        144       -135      22400                                        4590
         4 2 9        342       -171      44800                                        4591
         4 210        -27          0       5600                                        4592
         4 3 1          0         27      11200                                        4593
         4 3 2         81          0      44800                                        4594
         4 3 3          0          0          1                                        4595
```

4 3 4	-333	333	44800	4596
4 3 5	-135	126	44800	4597
4 3 6	-99	54	44800	4598
4 3 7	-9	-18	6400	4599
4 3 8	54	-9	44800	4600
4 3 9	333	-162	44800	4601
4 310	81	-162	22400	4602
4 4 1	0	9	224	4603
4 4 2	9	0	2800	4604
4 4 3	333	-333	22400	4605
4 4 4	-243	243	2240	4606
4 4 5	-91	-243	22400	4607
4 4 6	-1215	1053	22400	4608
4 4 7	-1215	243	22400	4609
4 4 8	-162	243	11200	4610
4 4 9	243	-162	2240	4611
4 410	1215	-1458	11200	4612
5 4 1	0	-9	3200	4613
5 4 2	9	0	396	4614
5 4 3	117	-117	44800	4615
5 4 4	81	243	44800	4616
5 4 5	-162	1215	44800	4617
5 4 6	-81	243	44800	4618
5 4 7	567	-1215	44800	4619
5 4 8	-405	243	44800	4620
5 4 9	-81	0	44800	4621
5 410	-243	-243	22400	4622
7 4 1	0	-9	640	4623
7 4 3	-9	9	6400	4624
7 4 4	1215	-243	44800	4625
7 4 5	-648	1215	44800	4626
7 4 6	243	-81	44800	4627
7 410	-243	243	22400	4628
7 5 1	0	9	44800	4629
7 5 2	-9	0	448	4630
7 5 4	81	0	44800	4631
7 5 5	81	-243	4480	4632
7 5 6	162	-243	44800	4633
7 510	729	0	22400	4634
7 6 1	0	153	22400	4635
7 6 2	99	0	44800	4636
7 6 4	-486	-81	44800	4637
7 6 5	-243	324	44800	4638
7 6 6	-243	81	22400	4639
7 610	0	0	1	4640
10 0 1	0	-9	560	4641
10 0 4	-54	135	560	4642
10 010	0	0	1	4643
10 1 1	0	3	1120	4644
10 1 2	-33	0	22400	4645
10 1 4	0	-27	4480	4646
10 1 5	-81	-108	22400	4647
10 1 6	0	27	5600	4648
10 110	0	81	5600	4649
10 4 1	0	27	2240	4650
10 4 2	27	0	2800	4651
10 4 3	27	-27	22400	4652
10 4 4	-1215	1458	22400	4653
10 4 5	0	243	11200	4654
10 4 6	-243	0	22400	4655
10 4 7	-243	-486	22400	4656
10 4 8	-243	486	22400	4657
10 4 9	1215	-243	22400	4658
10 410	243	-972	11200	4659
1010 1	0	-81	2800	4660
1010 4	-243	972	5600	4661
101010	0	0	1	4662

APPENDIX B

KC:	1	2	1	2	2	1	3	4	5	6	5	3	4	7	6	4	5	3	8	74663
	6	4	3	5	9	10	11	6	5	4	3	11	9	10	8	6	3	5	4	104664
	11	9	7	8	6	12	12	12	13	13	13	13	13	13	14					4665
LC:	1	1	3	6	3	2	1	1	1	3	3	3	1	3	2	2	2	1	3	4666
QC:			1	1			0			0			0			17				80	4667
			2	1			0			0			-7			0				160	4668
			4	1			0			0			-57			3				160	4669
			4	2			3			24			0			0				160	4670
			4	3			-3			3			3			-3				160	4671
			4	4			54			-27			-27			54				64	4672
			5	4			0			-27			-27			54				320	4673
			6	4			54			-27			-27			0				320	4674
			7	4			-54			27			-135			-54				320	4675
			7	5			0			27			27			-54				64	4676
			7	6			54			-27			-27			0				320	4677
			10	1			0			0			0			0				1	4678
			10	4			0			81			81			-162				160	4679
			10	10			162			-81			-81			162				80	4680

APPENDIX C

SAMPLE CALLING PROGRAM

```
      OVERLAY(MAIN,0,0)                                                    1
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,      2
     *    TAPE3,TAPE4,TAPE8,TAPE9)                                         3
      DIMENSION CFD(21)                                                    4
      COMMON/TEMP/I,TEMP(60)                                               5
      COMMON/STORE/STORE(40)                                               6
      COMMON/SPACE/SPACE(24),NSF,CURVE,SGFLAG,SMFLAG,SPFLAG,PRFLAG,        7
     *    RHO,H,X(4),Y(4),SKIP(60),NRECORD(7),                             8
     *    OTHERS(5200)                                                     9
      EQUIVALENCE(SPACE(1),CFD(1))                                        10
      LOGICAL CURVE,SGFLAG,SMFLAG,SPFLAG,PRFLAG                           11
C                                                                         12
C                                                                         13
C     . SET  NRECORD(I) = 1  TO ALLOW FINITE ELEMENTS WITH  NSF = I + 3   14
C        TO BE COMPUTED.  I  MAY TAKE ON VALUES  1,2,3,5,6, OR 7.         15
C        THE REMAINING COMPONENTS OF  NRECORD  SHOULD BE SET TO ZERO.      16
      NRECORD(1) = 1                                                      17
      NRECORD(2) = 1                                                      18
      NRECORD(3) = 1                                                      19
      NRECORD(5) = 1                                                      20
      NRECORD(6) = 1                                                      21
      NRECORD(7) = 1                                                      22
      NSF = 6                                                             23
C     . SET FIVE FLAGS.                                                   24
      CURVE = .T.                                                         25
      SGFLAG = .T.                                                        26
      SMFLAG = .T.                                                        27
      SPFLAG = .T.                                                        28
      PRFLAG = .T.                                                        29
C     . DEFINE DENSITY AND THICKNESS.                                     30
      RHO = .000001                                                       31
      H = .04                                                             32
C     . DEFINE COORDINATES OF CORNER NODES FOR TRIANGULAR ELEMENTS.       33
      X(1) = .2                                                           34
      Y(1) = .1                                                           35
      X(2) = .6                                                           36
      Y(2) = .3                                                           37
      X(3) = .4                                                           38
      Y(3) = .5                                                           39
      X(4) = 0.                                                           40
      Y(4) = 0.                                                           41
C     . DEFINE EXTENSIONAL AND TRANSVERSE SHEAR STIFFNESSES OF SHELL.     42
      CFD(1) = 1.23970                                                    43
      CFD(2) = .165820                                                    44
      CFD(3) = -.147991                                                   45
      CFD(4) = .0912631                                                   46
      CFD(5) = .0385601                                                   47
      CFD(6) = .179804                                                    48
      CFD(7) = .00244105                                                  49
      CFD(8) = -.000905935                                                50
      CFD(9) = -.00318850                                                 51
      CFD(10) = -.000629179                                               52
      CFD(11) = -.00102919                                                53
      CFD(12) = -.000905935                                               54
      CFD(13) = .000149019                                                55
      CFD(14) = .0000281489                                               56
      CFD(15) = .00000152451                                              57
      CFD(16) = .0000163629                                               58
      CFD(17) = .0000120026                                               59
      CFD(18) = .0000300134                                               60
      CFD(19) = .0234700                                                  61
      CFD(20) = .0205300                                                  62
      CFD(21) = -.000280154                                               63
```

```
C      . DEFINE PRESTRESS COEFFICIENTS.                                  64
       SPACE(22) = 1.                                                    65
       SPACE(23) = 1.                                                    66
       SPACE(24) = 1.                                                    67
C      . DEFINE CURVATURE AND LOAD COMPONENTS.                           68
       SPACE(41) = .5                                                    69
       SPACE(51) = .5                                                    70
       SPACE(61) = .1                                                    71
       SPACE(71) = 1.                                                    72
       SPACE(81) = 0.                                                    73
       SPACE(91) = 0.                                                    74
       DO 7 I=41,91,10                                                   75
          DO 7 J=1,9                                                     76
             SPACE(I+J) = SPACE(I)                                       77
C         CONTINUE                                                       78
 7     CONTINUE                                                          79
C      . DISPLAY FIRST  100  WORDS IN (LABELED COMMON) SPACE.            80
       WRITE(6,4)                                                        81
 4        FORMAT(1H1)                                                    82
       WRITE(6,1) (SPACE(I),I=1,100)                                     83
 1        FORMAT(* THE CONTENTS OF THE FIRST 100 WORDS OF LABELED*       84
     *        * COMMON /SPACE/ ARE AS FOLLOWS*//                         85
     *        1X,12E11.4/1X,12E11.4/18,5L11/7(1X,10E11.4/))              86
        NSF = 6                                                          87
        CALL ELEMENT                                                     88
        NSF = 10                                                         89
        CALL ELEMENT                                                     90
C      . DEFINE COORDINATES OF CORNER NODES FOR PARALLELOGRAM ELEMENTS.  91
        X(1) = .15                                                       92
        Y(1) = .2                                                        93
        X(2) = .55                                                       94
        Y(2) = .1                                                        95
        X(3) = .7                                                        96
        Y(3) = .4                                                        97
        X(4) = .3                                                        98
        Y(4) = .5                                                        99
        NSF = 4                                                         100
        WRITE(6,4)                                                      101
        WRITE(6,1) (SPACE(I),I=1,100)                                   102
        CALL ELEMENT                                                    103
        NSF = 5                                                         104
        CALL ELEMENT                                                    105
        NSF = 8                                                         106
        CALL ELEMENT                                                    107
        NSF = 9                                                         108
        CALL ELEMENT                                                    109
C      . DEFINE COORDINATES OF CORNER NODES FOR TRAPEZOIDAL ELEMENTS.   110
        X(1) = .1                                                       111
        Y(1) = .2                                                       112
        X(2) = .6                                                       113
        Y(2) = .2                                                       114
        X(3) = .5                                                       115
        Y(3) = .5                                                       116
        X(4) = .25                                                      117
        Y(4) = .5                                                       118
        NSF = 4                                                         119
        WRITE(6,4)                                                      120
        WRITE(6,1) (SPACE(I),I=1,100)                                   121
        CALL ELEMENT                                                    122
        NSF = 5                                                         123
        CALL ELEMENT                                                    124
        NSF = 8                                                         125
        CALL ELEMENT                                                    126
        NSF = 9                                                         127
        CALL ELEMENT                                                    128
```

APPENDIX C

```
C       . DEFINE COORDINATES OF CORNER NODES FOR TRAPEZIUM ELEMENTS.
        X(1) = .2
        Y(1) = .1
        X(2) = .6
        Y(2) = .1
        X(3) = .5
        Y(3) = .5
        X(4) = .15
        Y(4) = .35
        NSF = 4
        WRITE(6,4)
        WRITE(6,1) (SPACE(I),I=1,100)
        CALL ELEMENT
        NSF = 5
        CALL ELEMENT
        NSF = 8
        CALL ELEMENT
        NSF = 9
        CALL ELEMENT
```

APPENDIX D

OUTPUT FROM MODIFIED SAMPLE PROGRAM

This appendix presents output from the programs listed in appendixes A and C with the input data given in appendix B. To limit the number of pages of output, the program MAIN listed in appendix C was modified to calculate the characteristic arrays for only two finite elements, one with NSF = 5 and the other with NSF = 6. To accomplish this, the array NRECORD was defined as (0,1,1,0,0,0,0) instead of (1,1,1,0,1,1,1) (see statement lines 17 to 22 in appendix C), and the statements appearing at lines 89 to 109, 122, and 125 to 147 were deleted. The following output resulted:

```
THE CONTENTS OF THE FIRST 100 WORDS OF LABELED COMMON /SPACE/ ARE AS FOLLOWS

 .1240E+01  .1658E+00 -.1460E+00  .9126E-01  .3856E-01  .1798E+00  .2441E-02 -.9059E-03 -.3189E-02 -.6292E-03 -.1029E-02 -.9059E-03
 .1490E-03  .2815E-04  .1525E-05  .1636E-04  .1200E-04  .3001E-04  .2347E-01  .2053E-01 -.2802E-03  .1000E+01  .1000E+01  .1000E+01
     6          T          T          T          T          T
 .1000E-05  .4000E-01  .2000E+00  .6000E+00  .4000E+00 0.          .1000E+00  .3000E+00  .5000E+00 0.
 .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00
 .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00  .5000E+00
 .1000E+00  .1000E+00  .1000E+00  .1000E+00  .1000E+00  .1000E+00  .1000E+00  .1000E+00  .1000E+00  .1000E+00
 .1000E+01  .1000E+01  .1000E+01  .1000E+01  .1000E+01  .1000E+01  .1000E+01  .1000E+01  .1000E+01  .1000E+01
0.         0.         0.         0.         0.         0.         0.         0.         0.         0.
0.         0.         0.         0.         0.         0.         0.         0.         0.         0.
```

```
LENGTH OF COMMON/SPACE/ REQUIRED FOR NSF =     5 IS    1183.

  1   2   3   4   5   2   6   7   8   3   7   9   4   8   5   2  10  11  12   6
 13  14   7  15   8   3  11  16   7  14   9   4  12   8   5   2   6   7   8  10
 13  15  11  14  12   6  13  14  13  17  14   7  14  15   8   3   7   9  11  14
 16   7  15  14   9   4   8  12   8   5  18  19  20  21  19  22  23  20  23  21
 19  24  25  22  26  23  20  25  23  21  19  22  23  24  26  25  22  26  26  23
 20  23  25  23  21  27  28  29  28  30  29  28  31  30  29  28  30  31  30  29
 32  33  33  33  33  34
  1   1   1   1   1   4   1   1   1   4   5   1   4   5   4   3   1   1   1   4
  1   1   4   1   4   3   3   1   6   6   4   3   3   6   3   2   2   8   8   2
  2   2   4   5   4   3   3   8   4   1   4   3   3   4   3   2   2   2   2   2
  2   7   3   7   3   2   2   2   7   2   1   1   1   1   4   1   1   4   5   4
  3   1   1   4   1   4   3   3   6   3   2   2   2   8   2   2   4   3   3   4   3
  2   2   2   7   2   1   1   1   4   1   4   3   1   4   3   2   2   2   3   2
  1   1   4   3   2   1
    0.          -.333E+00 -.222E+00 -.150E+00 -.107E+00  0.          -.167E-01 -.133E-01  0.          0.          -.556E-02 -.333E-02
     .556E-02  0.           .333E-02  0.          0.          0.          -.889E-01 -.533E-01 -.305E-01  0.          -.508E-02  0.
    -.254E-02   .254E-02  0.          -.406E-01 -.232E-01  0.          0.          0.          -.232E-01  0.          0.          -.333E+00
    -.222E+00 -.150E+00 -.107E+00 -.111E+00 -.500E-01 -.267E-01 -.178E-01  0.          -.556E-02 -.333E-02 -.556E-02 -.222E-02
    -.333E-02  0.          0.          0.          -.889E-01 -.533E-01 -.305E-01 -.356E-01 -.152E-01  0.          -.254E-02 -.254E-02
    0.          -.406E-01 -.232E-01 -.174E-01  0.          0.          -.232E-01  0.          0.           .100E+01  .444E+00  .250E+00
     .160E+00   .222E+00   .833E-01   .406E-01   .267E-01   .111E+00   .278E-01   .100E-01   .278E-01   .667E-02  .100E-01  .444E-02
     .444E-02   .178E+01   .444E+00   .160E+00   .711E-01   .107E+00   .356E-01   .711E-01   .178E-01   .178E-01  .114E+01  .284E+00
     .929E-01   .697E-01   .522E-01   .836E+00   .209E+00   .660E+00
  1   1   1   1   2   3   4   5   4   6   7   8   9   8  10   4   3   4   5   6
 11  11  12  12  13   8   7   8   9  10   5   4   3   4   6  14  15  14  15  16
 12  11  11  12  13   9   8   7   8  10   4   5   4   3   6  11  12  12  11  13
 15  14  15  14  16  12  12  11  11  13   8   9   8   7  10  17  17  17  17  18
 19  20  21  20  22  20  19  20  21  22  21  20  19  20  22  20  21  20  19  22
 23  23  23  23  24   1   4   3   2   1   1   1   1   8   1   1   1   1   8   1
  5   4   4   4   4   1   5   1   5   1   5   4   4   4   4   3   6   3   3   3
  1   1   3   3   1   6   4   6   4   4   3   6   3   3   3   2   2   2   2   2
  8   2   8   2   2   2   4   4   2   2   3   7   3   7   3   2   2   7   2   2
  1   4   3   2   1   1   1   1   8   1   5   4   4   4   4   3   6   3   3   3
  2   2   7   2   2   1   4   3   2   1
    0.           0.          0.          -.167E+00  0.          0.          0.          -.111E+00  0.          0.          -.278E-01   .278E-01
     .889E-01   0.          0.          0.          0.          0.          0.          -.444E-01  0.          0.          0.          0.
     .100E+01   0.           .333E+00   .167E+00 -.167E+00 -.444E+00   .167E+00   .556E-01 -.556E-01 -.267E+00   .556E-01 -.556E-01
    -.178E+00   .278E-01   .278E-01   0.           .444E+00  0.           .133E+00   .889E-01 -.889E-01 -.142E+00   .284E+00  0.
    -.100E+01   0.          -.333E+00   .333E+00   .167E+00   .444E+00 -.167E+00   .167E+00   .556E-01   .267E+00 -.833E-01   .278E-01
    0.          -.278E-01   .278E-01   0.          -.444E+00  0.          -.133E+00   .133E+00   .889E-01   .142E+00 -.284E+00  0.
  1   2   1   3   2   1   2   3   2   1   4   4   4   4   5   1   1   4   1   4
  3   8   4   3   2   1   4   3   2   1
    -.333E+00   .333E+00   .167E+00  0.          -.284E+01   .250E+00 -.250E+00 -.250E+00 -.444E+00  0.           .250E+00   .250E+00
    -.250E+00 -.444E+00  0.          -.333E+00 -.167E+00   .167E+00  0.          -.284E+01
```

```
LENGTH OF COMMON/SPACE/ REQUIRED FOR NSF =     6 IS   1686.

 1   2   3   4   5   2   6   7   8   3   7   9   4   8   5   2   6   7   8   6
10  11   7  11   8   3   7   9   7  11   9   4   8   8   5  12  13  14  15  13
16  17  14  17  15  18  19  20  19  21  20  22  23  23  24  25  26  27  26  28
27  29  30  30  31  32  33  33  34  35  12  18  22  24  13  19  23  14  20  15
13  19  23  16  21  17  14  20  17  15  36  37  38  39  40  41  37  42  40  38
43  44  45  46  47  25  29  31  26  30  27  26  30  28  27  43  46  45  44  48
32  34  33  33  47  35  12  13  14  15  18  19  20  22  23  24  13  16  17  19
21  23  14  17  20  15  36  39  41  37  40  38  37  40  42  38  43  45  44  46
47  36  37  38  37  42  38  39  40  40  41  49  50  50  50  51  43  46  44  45
51  47  25  26  27  29  30  31  26  28  30  27  43  45  46  44  48  43  44  46
45  51  48  32  33  34  33  47  47  35
    .100E+01 0.          .333E-01  .143E-01  .952E-02 -.556E-02 -.159E-02 -.794E-03  .317E-03  .794E-03  .529E-04  .333E+00
    0.          .952E-02  .317E-02 -.317E-02 -.529E-03 -.222E-01 0.          -.529E-03  .212E-03  .317E-02 -.106E-03 0.
    .178E+00 0.          .423E-02 -.190E-02 -.127E-01  .212E-03  .127E-02  .114E+00 0.          -.762E-02  .813E-01  .889E-01
    -.635E-02  .106E-02 0.          -.212E-03  .212E-02  .423E-03  .381E-01 -.254E-02 0.          -.339E-02  .203E-01  .135E-01
    .254E-01 -.169E-02  .102E-01
 1   1   1   2   2   2   3   4   4   5   6   5   7   8   6   9  10   9   4   3
 4   5   5   6  11  11  12  13  14  14   8   7   8   9   9  10   4   4   3   6
 5   5  11  12  11  14  14  13  12  11  11  14  13  14   8   8   7  10   9   9
15  15  16  17  18  18  19  20  21  22  23  24  20  19  21  22  24  23  25  25
26  27  28  28  29  29  30  31  32  32  16  15  15  18  17  18  26  25  25  28
27  28  21  19  20  24  22  23  21  20  19  23  22  24  33  34  33  35  35  36
30  29  29  32  31  32  15  16  15  18  16  17  19  21  20  24  23  22  25  26
25  28  28  27  20  21  19  23  24  22  34  33  33  35  36  35  33  33  34  36
35  35  29  30  29  32  32  31   1   3   2   1   3   2   1   1   6   1   1   6
 1   1   6   1   1   6   4   3   3   4   3   3   1   4   1   1   1   4   4   3
 3   4   3   3   2   5   2   2   5   2   6   2   2   2   6   2   3   3   5   5
 3   3   2   5   2   2   5   2   1   4   1   1   1   4   1   1   1   1   1   1
 4   4   4   4   4   4   1   4   1   1   1   4   1   4   1   1   1   4   3   3
 5   5   3   3   3   3   5   5   3   3   3   3   3   3   3   3   5   5   5   5
 5   5   1   1   5   1   5   1   3   3   5   5   3   3   6   2   2   2   6   2
 6   6   6   6   6   6   6   2   2   2   6   2   2   2   2   2   2   2   2   4
 2   4   2   2   6   3   3   3   3   6   6   2   2   2   6   2
 1   2   1   2   2   1   3   3   4   5   4   3   3   6   5   3   4   3   6   6
 5   1   1   3   6   3   2   1   4   1   1   3   3   5   1   3   6   2   2   4
 3   2
    0.          0.          0.          0.          .667E+00 0.          0.          0.          0.          0.          -.333E+00  .333E+00
    0.          .833E-01 -.333E+00 0.          -.333E+00  .333E+00  .250E+00 0.          0.          0.          .667E+00 -.667E+00

THE SETUP TIME WAS   .427 SECONDS.
```

STIFFNESS MATRIX --- SS

.18725E+00	.39366E-01	-.87254E-02	-.80698E-03	-.10049E-02	.11953E+00	-.78197E-02	.43627E-02	.14442E-03	-.34743E-03
-.57116E-01	.20942E-01	.43627E-02	-.41341E-03	.12455E-04	-.47813E+00	.31279E-01	-.13088E-01	-.57767E-03	.13897E-02
0.	0.	.43627E-02	0.	0.	.22846E+00	-.83766E-01	-.13088E-01	.16536E-02	-.49819E-04
.39366E-01	.58031E-01	-.15667E-02	-.10049E-02	-.59892E-03	-.10150E-01	.17050E-01	.78333E-03	-.34743E-03	-.12288E-03
.23272E-01	.22935E-02	.78333E-03	.12455E-04	-.76756E-04	.40601E-01	-.68201E-01	-.23500E-02	.13897E-02	.49153E-03
0.	0.	.78333E-03	0.	0.	-.93089E-01	-.91738E-02	-.23500E-02	-.49819E-04	.30703E-03
-.87254E-02	-.15667E-02	.80419E-02	-.28040E-03	-.22649E-03	-.91005E-02	.11584E-02	.13180E-02	.13760E-03	.15904E-03
.47378E-02	-.19417E-02	.52601E-03	.18600E-03	.13271E-03	.22564E-01	-.15335E-02	-.58066E-02	-.44420E-03	-.47484E-03
-.43627E-02	-.78333E-03	-.53401E-03	.16900E-03	.15676E-03	-.51129E-02	.46668E-02	-.38466E-02	-.54100E-03	-.42218E-03
-.80698E-03	-.10049E-02	-.28040E-03	.77287E-04	.11388E-04	.14442E-03	-.34743E-03	-.32920E-03	.71516E-05	.13670E-05
-.41341E-03	.12455E-04	.14580E-03	-.12683E-04	.28026E-05	-.57767E-03	.13897E-02	.82740E-03	-.59900E-04	-.50943E-05
0.	0.	-.16900E-03	-.31293E-04	.37354E-06	.16536E-02	-.49619E-04	-.12260E-03	.19437E-04	-.10837E-04
-.10049E-02	-.59892E-03	-.22649E-03	.11388E-04	.52790E-04	-.34743E-03	-.12288E-03	.11885E-03	.10562E-05	-.37507E-05
.12455E-04	-.76756E-04	-.29736E-03	.31133E-05	-.60258E-05	.13897E-02	.49153E-03	-.80937E-04	-.38513E-05	-.12370E-04
0.	0.	-.15676E-03	.37354E-06	-.27373E-04	-.49819E-04	.30703E-03	.75147E-03	-.12080E-04	-.32700E-05
.11953E+00	-.10150E-01	-.91005E-02	.14442E-03	-.34743E-03	.95509E+00	-.20744E+00	.18201E-01	.36020E-02	-.16932E-02
.19863E+00	-.58997E-01	-.91005E-02	.10563E-02	-.21698E-03	-.47813E+00	.40601E-01	.27302E-01	-.57767E-03	.13897E-02
-.79533E+00	.23599E+00	.27302E-01	-.42251E-02	.86793E-03	0.	0.	-.91005E-02	0.	0.
-.78197E-02	.17050E-01	.11584E-02	-.34743E-03	-.12288E-03	-.20744E+00	.10937E+00	-.23168E-02	-.16932E-02	-.22693E-04
-.61328E-01	.19407E-01	.11584E-02	-.21698E-03	.11532E-03	.31279E-01	-.68201E-01	-.34753E-02	.13897E-02	.49153E-03
.24531E+00	-.77630E-01	-.34753E-02	.86793E-03	-.46127E-03	0.	0.	.11584E-02	0.	0.
.43627E-02	.78333E-03	.13180E-02	-.32920E-03	.11885E-03	.18201E-01	-.23168E-02	.20057E-01	.66360E-03	-.32929E-03
.47378E-02	-.19417E-02	.48331E-02	-.28340E-03	.13831E-03	-.17826E-01	-.40823E-03	-.58066E-02	.95621E-03	-.35425E-03
-.18576E-01	.50419E-02	-.19867E-01	.96401E-03	-.39319E-03	.91005E-02	-.11584E-02	-.53461E-03	-.29780E-03	.11656E-03
.14442E-03	-.34743E-03	.13760E-03	.71516E-05	.10562E-05	.36020E-02	-.16932E-02	.66360E-03	.15027E-03	-.16931E-04
.10563E-02	-.21698E-03	.14920E-03	.11646E-04	-.60258E-05	-.57767E-03	.13897E-02	-.57300E-03	-.59900E-04	-.38513E-05
-.42251E-02	.86793E-03	-.58424E-03	-.77876E-04	.25679E-04	0.	0.	.29780E-03	-.31293E-04	.37354E-06
-.34743E-03	-.12288E-03	.15904E-03	.13670E-05	-.37507E-05	-.16932E-02	-.22693E-04	-.32929E-03	-.16931E-04	.55794E-04
-.21698E-03	.11532E-03	-.25156E-03	-.66371E-05	-.50245E-05	.13697E-02	.49153E-03	-.20153E-03	-.50943E-05	-.12370E-04
.86793E-03	-.46127E-03	.61967E-03	.26922E-04	-.72755E-05	0.	0.	-.11656E-03	.37354E-06	-.27373E-04
-.57116E-01	.23272E-01	.47378E-02	-.41341E-03	.12455E-04	.19883E+00	-.61328E-01	.47378E-02	.10563E-02	-.21698E-03
.42515E+00	-.11417E+00	-.94756E-02	.19286E-02	-.61359E-03	0.	0.	.47378E-02	0.	0.
-.79533E+00	.24531E+00	-.14213E-01	.42251E-02	.86793E-03	.22846E+00	-.93089E-01	-.14213E-01	.16536E-02	-.49819E-04
.20942E-01	.22935E-02	-.19417E-02	.12455E-04	-.76756E-04	-.58997E-01	.19407E-01	-.19417E-02	-.21698E-03	.11532E-03
-.11417E+00	.65103E-01	.38835E-02	-.61359E-03	.11568E-03	0.	0.	-.19417E-02	0.	0.
.23599E+00	-.77630E-01	.58252E-02	.86793E-03	-.46127E-03	-.83766E-01	-.91738E-02	.58252E-02	-.49819E-04	.30703E-03
.43627E-02	.78333E-03	.52601E-03	.14580E-03	-.29736E-03	-.91005E-02	.11584E-02	.48331E-02	.14320E-03	-.25156E-03
-.94756E-02	.38835E-02	.18587E-01	-.38320E-03	.55578E-03	-.47378E-02	.19417E-02	-.53461E-03	.12880E-03	-.27331E-03
.22939E-01	-.42566E-02	-.19867E-01	-.41521E-03	.77643E-03	-.39876E-02	-.35084E-02	-.38466E-02	-.42040E-03	.86803E-03
-.41341E-03	.12455E-04	.18600E-03	-.12683E-04	.31133E-05	.10563E-02	-.21698E-03	-.28340E-03	.11646E-04	-.66371E-05
.19286E-02	-.61359E-03	-.38320E-03	.90769E-04	-.11692E-04	0.	0.	-.12880E-03	-.31293E-04	.37354E-06
-.42251E-02	.86793E-03	.69561E-03	-.77876E-04	.26922E-04	.16536E-02	-.49819E-04	-.24319E-03	.19437E-04	-.12080E-04
.12455E-04	-.76756E-04	.13271E-03	.28026E-05	-.60258E-05	-.21698E-03	.11532E-03	.13831E-03	-.63263E-05	-.50245E-05
-.61359E-03	.11568E-03	.55578E-03	-.11692E-04	.48969E-04	0.	0.	.27331E-03	.37354E-06	-.27373E-04
.86793E-03	-.46127E-03	-.54994E-03	.25679E-04	-.72755E-05	-.49819E-04	.30703E-03	-.53873E-03	-.10837E-04	-.32700E-05
-.47813E+00	.40601E-01	.22564E-01	-.57767E-03	.13697E-02	-.47813E+00	.31279E-01	-.17826E-01	-.57767E-03	.13897E-02
0.	0.	-.47378E-02	0.	0.	.20900E+01	-.37632E+00	.37902E-01	.62982E-02	-.44157E-02

.45593E+00	-.17685E+00	.15004E-02	.33073E-02	-.99638E-04	-.15907E+01	.48130E+00	.55353E-01	-.84501E-02	.17359E-02
.31279E-01	-.68201E-01	-.15335E-02	.13897E-02	.49153E-03	.40601E-01	-.68201E-01	-.40823E-03	.13897E-02	.49153E-03
0.	0.	.19417E-02	0.	0.	-.37632E+00	.31001E+00	-.15534E-01	-.44157E-02	-.67457E-03
-.17685E+00	-.18348E-01	-.10900E-01	-.99638E-04	.61405E-03	.48130E+00	-.15526E+00	-.12401E-01	.17359E-02	-.92254E-03
-.13088E-01	-.23500E-02	-.58066E-02	.82740E-03	-.80937E-04	.27302E-01	-.34753E-02	-.58066E-02	-.57300E-03	-.20153E-03
.47378E-02	-.19417E-02	-.53461E-03	-.12380E-03	.27331E-03	.37902E-01	-.15534E-01	.63317E-01	.15328E-02	-.22231E-02
-.15004E-02	.10900E-01	-.55548E-02	-.16079E-03	-.17203E-02	-.55353E-01	.12401E-01	-.37595E-01	-.17064E-02	-.15595E-02
-.57767E-03	.13897E-02	-.44420E-03	-.59900E-04	-.38513E-05	-.57767E-03	.13897E-02	.95621E-03	-.59900E-04	-.50943E-05
0.	0.	.12880E-03	-.31293E-04	.37354E-06	.62982E-02	-.44157E-02	.15328E-02	.48702E-03	-.23727E-04
.33073E-02	-.99638E-04	.16079E-03	.16405E-03	-.24411E-04	-.84501E-02	.17359E-02	-.17064E-02	-.30579E-04	.51107E-04
.13897E-02	.49153E-03	-.47404E-03	-.50943E-05	-.12370E-04	.13897E-02	.49153E-03	-.35425E-03	-.38513E-05	-.12370E-04
0.	0.	-.27331E-03	.37354E-06	-.27373E-04	-.44157E-02	-.67457E-03	-.22231E-02	-.23727E-04	.26482E-03
-.99638E-04	.61405E-03	.17203E-02	-.24411E-04	.10295E-03	.17359E-02	-.92254E-03	.15595E-02	.51107E-04	.94942E-04
0.	0.	-.43627E-02	0.	0.	-.79533E+00	.24531E+00	-.18576E-01	-.42251E-02	.86793E-03
-.79533E+00	.23599E+00	.22939E-01	-.42251E-02	.86793E-03	.45693E+00	-.17685E+00	-.15004E-02	.33073E-02	-.99638E-04
.20900E+01	-.37632E+00	.34902E-01	.62982E-02	-.44157E-02	-.95627E+00	.71880E-01	.53853E-01	-.11553E-02	.27794E-02
0.	0.	-.78333E-03	0.	0.	.23599E+00	-.77630E-01	.50419E-02	.86793E-03	-.46127E-03
.24531E+00	-.77630E-01	-.42586E-02	.86793E-03	-.46127E-03	-.17685E+00	-.18348E-01	.10900E-01	-.99638E-04	.61405E-03
-.37632E+00	.31001E+00	.62666E-02	-.44157E-02	-.67457E-03	.71880E-01	-.13640E+00	-.15004E-02	.27794E-02	.98306E-03
.43627E-02	.78333E-03	-.53461E-03	-.16900E-03	-.15676E-03	.27302E-01	-.34753E-02	-.19867E-01	-.58421E-03	.61967E-03
-.14213E-01	.58252E-02	-.19867E-01	.69561E-03	-.54994E-03	.15004E-02	-.10900E-01	-.55548E-02	.16079E-03	.17203E-02
.34902E-01	.62666E-02	.63317E-01	.11216E-02	.90594E-03	-.53853E-01	.15004E-02	-.94748E-02	.18672E-02	.16079E-03
0.	0.	.16900E-03	-.31293E-04	.37354E-06	-.42251E-02	.86793E-03	.86461E-03	-.77876E-04	.26922E-04
-.42251E-02	.86793E-03	-.41521E-03	-.77876E-04	.25679E-04	.33073E-02	-.99638E-04	-.16079E-03	.16405E-03	-.24411E-04
.62982E-02	-.44157E-02	.11216E-02	.48702E-03	-.23727E-04	-.11053E-02	.27794E-02	-.18672E-02	.53737E-05	-.10440E-04
0.	0.	.15676E-03	.37354E-06	-.27373E-04	.86793E-03	-.46127E-03	-.39319E-03	.25679E-04	-.72755E-05
.86793E-03	-.46127E-03	.77643E-03	.26922E-04	-.72755E-05	-.99638E-04	.61405E-03	-.17203E-02	-.24411E-04	.10295E-03
-.44157E-02	-.67457E-03	.90594E-03	-.23727E-04	.26482E-03	.27794E-02	.98306E-03	-.16079E-03	-.10440E-04	.84753E-04
.22846E+00	-.93089E-01	-.51129E-02	.16536E-02	-.49819E-04	0.	0.	.91005E-02	0.	0.
.22846E+00	-.83766E-01	-.39676E-02	.16536E-02	-.49819E-04	-.15907E+01	.48130E+00	-.55353E-01	-.84501E-02	.17359E-02
-.95627E+00	.71880E-01	-.53853E-01	-.11553E-02	.27794E-02	.20900E+01	-.37632E+00	-.72804E-01	.62982E-02	-.44157E-02
-.83766E-01	-.91738E-02	.46666E-02	-.49819E-04	.30703E-03	0.	0.	-.11584E-02	0.	0.
-.93089E-01	-.91738E-02	-.35064E-02	-.49819E-04	.30703E-03	.48130E+00	-.15526E+00	.12401E-01	.17359E-02	-.92254E-03
.71880E-01	-.13640E+00	.15004E-02	.27794E-02	.98306E-03	-.37632E+00	.31001E+00	.92674E-02	-.44157E-02	-.67457E-03
-.13088E-01	-.23500E-02	-.58066E-02	-.12250E-03	.75147E-03	-.91005E-02	.11584E-02	-.53461E-03	.29780E-03	-.11656E-03
-.14213E-01	.58252E-02	-.58466E-02	-.24319E-03	-.53873E-03	.55353E-01	-.12401E-01	-.37595E-01	-.17064E-02	.15595E-02
.53053E-01	-.15004E-02	-.94746E-02	-.18672E-02	-.16079E-03	-.72804E-01	.92674E-02	.63317E-01	-.26544E-02	.13172E-02
.16536E-02	-.49819E-04	-.54100E-03	.19437E-04	-.12086E-04	0.	0.	-.29780E-03	-.31293E-04	.37354E-06
.16536E-02	-.49819E-04	-.42646E-03	.19437E-04	-.10837E-04	-.84501E-02	.17359E-02	.17064E-02	-.30579E-04	.51107E-04
-.11553E-02	.27794E-02	.18672E-02	.53737E-05	-.10440E-04	.62982E-02	-.44157E-02	-.26544E-02	.48702E-03	-.23727E-04
-.49819E-04	.30703E-03	-.42216E-03	-.10837E-04	-.32700E-05	0.	0.	.11656E-03	.37354E-06	-.27373E-04
-.49819E-04	.30703E-03	.86663E-03	-.12080E-04	-.32700E-05	.17359E-02	-.92254E-03	-.15595E-02	.51107E-04	.94942E-04
.27794E-02	.98306E-03	.16079E-03	-.10440E-04	.84753E-04	-.44157E-02	-.67457E-03	.13172E-02	-.23727E-04	.26482E-03

GEOMETRIC STIFFNESS ARRAY --- SG

.66667E+00	.11111E+00	.11111E+00	-.44444E+00	0.	-.44444E+00	.11111E+00	.16667E+00	-.55556E-01	-.44444E+00
.22222E+00	0.	.11111E+00	-.55556E-01	.16667E+00	0.	.22222E+00	-.44444E+00	-.44444E+00	-.44444E+00
0.	.13333E+01	-.88889E+00	.44444E+00	0.	.22222E+00	.22222E+00	-.88889E+00	.13333E+01	-.88889E+00
-.44444E+00	0.	-.44444E+00	.44444E+00	-.88889E+00	.13333E+01				

LOAD VECTOR --- SP

0.	0.	.13878E-16	0.	0.	0.	0.	.13878E-16	0.	0.
0.	0.	.69389E-17	0.	0.	0.	0.	-.20000E-01	0.	0.
0.	0.	-.20000E-01	0.	0.	0.	0.	-.20000E-01	0.	0.

CONSISTENT MASS ARRAY --- SM

.80000E-10	-.13333E-10	-.13333E-10	0.	-.53333E-10	0.	-.13333E-10	.80000E-10	-.13333E-10	0.
0.	-.53333E-10	-.13333E-10	-.13333E-10	.80000E-10	-.53333E-10	0.	0.	0.	0.
-.53333E-10	.42667E-09	.21333E-09	.21333E-09	-.53333E-10	0.	0.	.21333E-09	.42667E-09	.21333E-09
0.	-.53333E-10	0.	.21333E-09	.21333E-09	.42667E-09				

FULL CONSISTENT MASS MATRIX --- SMASS

.80000E-10	0.	0.	0.	0.	-.13333E-10	0.	0.	0.	0.
-.13333E-10	0.	0.	0.	0.	0.	0.	0.	0.	0.
-.53333E-10	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	.80000E-10	0.	0.	0.	0.	-.13333E-10	0.	0.	0.
0.	-.13333E-10	0.	0.	0.	0.	0.	0.	0.	0.
0.	-.53333E-10	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	.80000E-10	0.	0.	0.	0.	-.13333E-10	0.	0.
0.	0.	-.13333E-10	0.	0.	0.	0.	0.	0.	0.
0.	0.	-.53333E-10	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	.10667E-13	0.	0.	0.	0.	-.17778E-14	0.
0.	0.	0.	-.17778E-14	0.	0.	0.	0.	0.	0.
0.	0.	0.	-.71111E-14	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	.10667E-13	0.	0.	0.	0.	-.17778E-14
0.	0.	0.	0.	-.17778E-14	0.	0.	0.	0.	0.
0.	0.	0.	0.	-.71111E-14	0.	0.	0.	0.	0.
-.13333E-10	0.	0.	0.	0.	.80000E-10	0.	0.	0.	0.
-.13333E-10	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	-.53333E-10	0.	0.	0.	0.
0.	-.13333E-10	0.	0.	0.	0.	.80000E-10	0.	0.	0.
0.	-.13333E-10	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	-.53333E-10	0.	0.	0.
0.	0.	-.13333E-10	0.	0.	0.	0.	.80000E-10	0.	0.
0.	0.	-.13333E-10	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	-.53333E-10	0.	0.
0.	0.	0.	-.17778E-14	0.	0.	0.	0.	.10667E-13	0.
0.	0.	0.	-.17778E-14	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	-.71111E-14	0.
0.	0.	0.	0.	-.17778E-14	0.	0.	0.	0.	.10667E-13
0.	0.	0.	0.	-.17778E-14	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	-.71111E-14
-.13333E-10	0.	0.	0.	0.	-.13333E-10	0.	0.	0.	0.
.80000E-10	0.	0.	0.	0.	-.53333E-10	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	-.13333E-10	0.	0.	0.	0.	-.13333E-10	0.	0.	0.
0.	.80000E-10	0.	0.	0.	0.	-.53333E-10	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	-.13333E-10	0.	0.	0.	0.	-.13333E-10	0.	0.
0.	0.	.80000E-10	0.	0.	0.	0.	-.53333E-10	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	-.17778E-14	0.	0.	0.	0.	-.17778E-14	0.
0.	0.	0.	.10667E-13	0.	0.	0.	0.	-.71111E-14	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	-.17778E-14	0.	0.	0.	0.	-.17778E-14
0.	0.	0.	0.	.10667E-13	0.	0.	0.	0.	-.71111E-14
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
-.53333E-10	0.	0.	0.	0.	.42667E-09	0.	0.	0.	0.

.21333E-09	0.	0.	0.	0.	.21333E-09	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	-.53333E-10	0.	0.	0.	0.	.42667E-09	0.	0.	0.
0.	.21333E-09	0.	0.	0.	0.	.21333E-09	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	-.53333E-10	0.	0.	0.	0.	.42667E-09	0.	0.
0.	0.	.21333E-09	0.	0.	0.	0.	.21333E-09	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	-.71111E-14	0.	0.	0.	0.	.56889E-13	0.
0.	0.	0.	.28444E-13	0.	0.	0.	0.	.28444E-13	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	-.71111E-14	0.	0.	0.	0.	.56889E-13
0.	0.	0.	0.	.28444E-13	0.	0.	0.	0.	.28444E-13
-.53333E-10	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	.21333E-09	0.	0.	0.	0.
.42667E-09	0.	0.	0.	0.	.21333E-09	0.	0.	0.	0.
0.	-.53333E-10	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	.21333E-09	0.	0.	0.
0.	.42667E-09	0.	0.	0.	0.	.21333E-09	0.	0.	0.
0.	0.	-.53333E-10	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	.21333E-09	0.	0.	0.
0.	0.	.42667E-09	0.	0.	0.	.21333E-09	0.	0.	0.
0.	0.	0.	-.71111E-14	0.	0.	0.	0.	0.	0.
0.	0.	0.	.56889E-13	0.	0.	0.	.28444E-13	0.	0.
0.	0.	0.	0.	-.71111E-14	0.	0.	.28444E-13	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	.28444E-13
0.	0.	0.	0.	.56889E-13	0.	0.	0.	0.	.28444E-13
0.	0.	0.	0.	0.	-.53333E-10	0.	0.	0.	0.
0.	0.	0.	0.	0.	.21333E-09	0.	0.	0.	0.
.21333E-09	0.	0.	0.	0.	.42667E-09	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	-.53333E-10	0.	0.	0.
0.	0.	0.	0.	0.	0.	.21333E-09	0.	0.	0.
0.	.21333E-09	0.	0.	0.	0.	.42667E-09	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	-.53333E-10	0.	0.
0.	0.	0.	0.	0.	0.	0.	.21333E-09	0.	0.
0.	0.	.21333E-09	0.	0.	0.	0.	.42667E-09	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	-.71111E-14	0.
0.	0.	0.	0.	0.	0.	0.	0.	.28444E-13	0.
0.	0.	0.	.28444E-13	0.	0.	0.	0.	.56889E-13	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	-.71111E-14
0.	0.	0.	0.	0.	0.	0.	0.	0.	.28444E-13
0.	0.	0.	0.	.28444E-13	0.	0.	0.	0.	.56889E-13

THE COMPUTATION TIME FOR THIS ELEMENT WAS .030 SECONDS.

THE CONTENTS OF THE FIRST 100 WORDS OF LABELED COMMON /SPACE/ ARE AS FOLLOWS

.1240E+01	.1656E+00	-.1480E+00	.9126E-01	.3856E-01	.1798E+00	.2441E-02	-.9059E-03	-.3189E-02	-.6292E-03	-.1029E-02	-.9059E-03
.1490E-03	.2815E-04	.1525E-05	.1636E-04	.1200E-04	.3001E-04	.2347E-01	.2053E-01	-.2802E-03	.1000E+01	.1000E+01	.1000E+01
4	T	T	T	T	T						
.1000E-05	.4000E-01	.1000E+00	.6000E+00	.5000E+00	.2500E+00	.2000E+00	.2000E+00	.5000E+00	.5000E+00		
.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00		
.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00	.5000E+00		
.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00	.1000E+00		
.1000E+01	.1000E+01	.1000E+01	.1000E+01	.1000E+01	.1000E+01	.1000E+01	.1000E+01	.1000E+01	.1000E+01		
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.		
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.		

STIFFNESS MATRIX --- SS

.27670E+00	.57791E-01	-.32564E-01	-.11957E-02	-.14869E-02	-.24581E+00	.47612E-01	-.32955E-01	-.92699E-03	.43360E-03
-.13008E+00	-.69051E-01	-.16282E-01	.13477E-02	.10481E-02	.99190E-01	-.36353E-01	-.15891E-01	.77504E-03	.52687E-05
.98833E-01	-.10285E+00	-.43419E-01	.18878E-02	.49738E-03	.57791E-01	.88722E-01	-.70104E-02	-.14869E-02	-.88651E-03
.54604E-01	-.16205E-01	-.41716E-02	.43360E-03	.48975E-05	-.69051E-01	-.58954E-01	-.35052E-02	.10481E-02	.72523E-03
-.43345E-01	-.13562E-01	-.63436E-02	.52687E-05	.15639E-03	-.10285E+00	-.18772E-01	-.93472E-02	.49738E-03	.59234E-03
-.32564E-01	-.70104E-02	.17739E-01	-.10301E-02	-.10123E-02	.34518E-01	-.71827E-02	.28674E-02	-.10515E-02	-.82575E-03
.16282E-01	.35052E-02	-.54189E-02	-.64213E-03	-.67743E-03	-.18236E-01	.10688E-01	-.26578E-02	-.74774E-03	-.10352E-02
.43419E-01	.93472E-02	.65535E-02	-.17123E-02	-.18065E-02	-.11957E-02	-.14869E-02	-.10301E-02	.38713E-03	.13533E-04
-.92699E-03	.43360E-03	.12657E-02	.14454E-03	.14149E-05	.13477E-02	.10481E-02	.64213E-03	.47015E-04	-.17069E-04
.77504E-03	.52687E-05	-.49651E-03	.15475E-03	-.66331E-05	.18878E-02	.49738E-03	.17123E-02	.31353E-03	-.20765E-04
-.14869E-02	-.88651E-03	-.10123E-02	.13533E-04	.31714E-03	.43360E-03	.48975E-05	-.80396E-03	.23471E-05	.14750E-03
.10481E-02	.72523E-03	.67743E-03	-.17069E-04	.51461E-04	.52687E-05	.15639E-03	.16527E-02	-.75653E-05	.12546E-03
.49738E-03	.59234E-03	.18065E-02	-.20765E-04	.26728E-03	-.24581E+00	.54604E-01	.34518E-01	-.92699E-03	.43360E-03
.43967E+00	-.11181E+00	.34908E-01	.19173E-02	-.66676E-03	.99190E-01	-.43345E-01	.17845E-01	.77504E-03	.52687E-05
-.29306E+00	.10055E+00	.17454E-01	-.17653E-02	.22789E-03	-.98833E-01	.10285E+00	.46545E-01	-.18878E-02	-.49738E-03
.47612E-01	-.16205E-01	-.71827E-02	.43360E-03	.48975E-05	-.11181E+00	.57767E-01	-.10021E-01	-.66676E-03	.90245E-04
-.36353E-01	-.13562E-01	-.78493E-02	.52687E-05	.15639E-03	.10055E+00	-.28000E-01	-.50107E-02	.22789E-03	-.25153E-03
.10285E+00	.18772E-01	-.13362E-01	-.49738E-03	-.59234E-03	-.32955E-01	-.41718E-02	.28674E-02	.12657E-02	-.80396E-03
.34908E-01	-.10021E-01	.19730E-01	.13513E-02	-.14323E-02	.15501E-01	.91824E-02	-.26578E-02	.44296E-03	-.12452E-02
-.17454E-01	.50107E-02	-.74099E-02	.51651E-03	-.66653E-03	-.46545E-01	.13362E-01	.41387E-02	.13774E-02	-.17774E-02
-.92699E-03	.43360E-03	-.10515E-02	.14454E-03	.23471E-05	.19173E-02	-.66676E-03	.13513E-02	.38811E-03	-.14548E-04
.77504E-03	.52687E-05	.69420E-03	.15475E-03	-.75653E-05	-.17653E-02	.22789E-03	-.51651E-03	.46039E-04	.11012E-04
-.18878E-02	-.49738E-03	-.13774E-02	.31234E-03	.13294E-04	.43360E-03	.48975E-05	-.82575E-03	.14149E-05	.14750E-03
-.66676E-03	.90245E-04	-.14323E-02	-.14548E-04	.30650E-03	.52687E-05	.15639E-03	.14427E-02	-.66331E-05	.12546E-03
.22789E-03	-.25153E-03	.66653E-03	.11012E-04	.62100E-04	-.49738E-03	-.59234E-03	.17774E-02	.13294E-04	.28019E-03
-.13008E+00	-.69051E-01	.16282E-01	.13477E-02	.10481E-02	.99190E-01	-.36353E-01	.15501E-01	.77504E-03	.52687E-05
.42331E+00	.46531E-01	.32564E-01	-.10436E-02	-.19258E-02	-.39242E+00	.58872E-01	.33346E-01	-.10789E-02	.87247E-03
.19767E+00	-.20570E+00	.43419E-01	.37756E-02	.99475E-03	-.69051E-01	-.58954E-01	.35052E-02	.10481E-02	.72523E-03
-.43345E-01	-.13562E-01	.91824E-02	.52687E-05	.15639E-03	.46531E-01	.11849E+00	.70104E-02	-.19258E-02	-.10478E-02
.65864E-01	-.45972E-01	.13332E-02	.87247E-03	.16618E-03	-.20570E+00	-.37544E-01	.93472E-02	.99475E-03	.11847E-02
-.16282E-01	-.35052E-02	-.54189E-02	.64213E-03	.67743E-03	.17845E-01	-.78493E-02	-.26578E-02	.69420E-03	.14427E-02
.32564E-01	.70104E-02	.21289E-01	.10301E-02	.10123E-02	-.34127E-01	.43441E-02	-.31880E-02	.11051E-02	.41833E-03
-.43419E-01	-.93472E-02	.70926E-02	.17123E-02	.18065E-02	.13477E-02	.10481E-02	-.64213E-03	.47015E-04	-.17069E-04
.77504E-03	.52687E-05	.44296E-03	.15475E-03	-.66331E-05	-.10438E-02	-.19258E-02	.10301E-02	.30788E-03	.16125E-04
-.10789E-02	.87247E-03	-.12122E-02	.77110E-04	.57406E-06	.37756E-02	.99475E-03	-.17123E-02	.27500E-03	-.37327E-04
.10481E-02	.72523E-03	-.67743E-03	-.17069E-04	.51461E-04	.52687E-05	.15639E-03	-.12452E-02	-.75653E-05	.12546E-03
-.19258E-02	-.10478E-02	.10123E-02	.16125E-04	.23664E-03	.87247E-03	.16618E-03	.39653E-03	.15063E-05	.99690E-04

```
 .99475E-03  .11847E-02 -.18065E-02 -.37327E-04  .22661E-03  .99190E-01 -.43345E-01 -.18236E-01  .77504E-03  .52687E-05
-.29306E+00  .10055E+00 -.17454E-01 -.17653E-02  .22789E-03 -.39242E+00  .65864E-01 -.34127E-01 -.10789E-02  .87247E-03
 .58629E+00 -.12307E+00 -.34908E-01  .20692E-02 -.11056E-02 -.19767E+00  .20570E+00 -.46545E-01 -.37756E-02 -.99475E-03
-.36353E-01 -.13562E-01  .10688E-01  .52687E-05  .15639E-03  .10055E+00 -.28000E-01  .50107E-02  .22789E-03 -.25153E-03
 .58872E-01 -.45972E-01  .43441E-02  .87247E-03  .16618E-03 -.12307E+00  .87534E-01  .10021E-01 -.11056E-02 -.71039E-04
 .20570E+00  .37544E-01  .13362E-01 -.99475E-03 -.11847E-02 -.15891E-01 -.63438E-02 -.26578E-02 -.49651E-03  .16527E-02
 .17454E-01 -.50107E-02 -.74099E-02 -.51651E-03  .66653E-03  .33346E-01  .13332E-02 -.31880E-02 -.12122E-02  .39653E-03
-.34908E-01  .10021E-01  .23260E-01 -.13513E-02  .14323E-02  .46545E-01 -.13362E-01  .22630E-02 -.13774E-02  .17774E-02
 .77504E-03  .52687E-05 -.74774E-03  .15475E-03 -.75653E-05 -.17653E-02  .22789E-03  .51651E-03  .46039E-04  .11012E-04
-.10789E-02  .87247E-03  .11051E-02  .77110E-04  .15063E-05  .20692E-02 -.11056E-02 -.13513E-02  .30886E-03 -.11956E-04
-.37756E-02 -.99475E-03  .13774E-02  .27263E-03  .30790E-04  .52687E-05  .15639E-03 -.10352E-02 -.66331E-05  .12546E-03
 .22789E-03 -.25153E-03 -.66653E-03  .11012E-04  .62100E-04  .87247E-03  .16618E-03  .41833E-03  .57406E-06  .99690E-04
-.11056E-02 -.71039E-04  .14323E-02 -.11956E-04  .22600E-03 -.99475E-03 -.11847E-02 -.17774E-02  .30790E-04  .25242E-03
 .98833E-01 -.10285E+00  .43419E-01  .18878E-02  .49738E-03 -.98833E-01  .10285E+00 -.46545E-01 -.18878E-02 -.49738E-03
 .19767E+00 -.20570E+00 -.43419E-01  .37756E-02  .99475E-03 -.19767E+00  .20570E+00  .46545E-01 -.37756E-02 -.99475E-03
 .36101E-01 -.26192E+00  0.          .34216E-02 -.10950E-01 -.10285E+00 -.18772E-01  .93472E-02  .49738E-03  .59234E-03
 .10285E+00  .18772E-01  .13362E-01 -.49738E-03 -.59234E-03 -.20570E+00 -.37544E-01 -.93472E-02  .99475E-03  .11847E-02
 .20570E+00  .37544E-01 -.13362E-01 -.99475E-03 -.11847E-02 -.26192E+00  .74898E+00 -.46838E-16 -.10950E-01 -.40975E-02
-.43419E-01 -.93472E-02  .65535E-02  .17123E-02  .18065E-02  .46545E-01 -.13362E-01  .41387E-02 -.13774E-02  .17774E-02
 .43419E-01  .93472E-02  .70926E-02 -.17123E-02 -.18065E-02 -.46545E-01  .13362E-01  .22630E-02  .13774E-02 -.17774E-02
 0.         -.46838E-16  .14550E+00  .12333E-17  .21684E-18  .18878E-02  .49738E-03 -.17123E-02  .31353E-03 -.20765E-04
-.18878E-02 -.49738E-03  .13774E-02  .31234E-03  .13294E-04  .37756E-02  .99475E-03  .17123E-02  .27500E-03 -.37327E-04
-.37756E-02 -.99475E-03 -.13774E-02  .27263E-03  .30790E-04  .34216E-02 -.10950E-01  .12333E-17  .12096E-02  .29111E-04
 .49738E-03  .59234E-03 -.18065E-02 -.20765E-04  .26728E-03 -.49738E-03 -.59234E-03 -.17774E-02  .13294E-04  .28019E-03
 .99475E-03  .11847E-02  .18065E-02 -.37327E-04  .22661E-03 -.99475E-03 -.11847E-02  .17774E-02  .30790E-04  .25242E-03
-.10950E-01 -.40975E-02  .21684E-18  .29111E-04  .78414E-03
```

GEOMETRIC STIFFNESS ARRAY --- SG

```
 .10137E+01  .69608E-01 -.78922E+00 -.29412E+00 -.55589E+00  .69608E-01  .97059E-01 -.29412E+00  .12745E+00  .55589E+00
-.78922E+00 -.29412E+00  .12382E+01 -.15490E+00 -.11118E+01 -.29412E+00  .12745E+00 -.15490E+00  .32157E+00  .11118E+01
-.55589E+00  .55589E+00 -.11118E+01  .11118E+01  .57401E+01
```

LOAD VECTOR --- SP

```
0.          0.          -.31250E-01 0.          0.          0.          0.          -.31250E-01 0.          0.
0.          0.          -.25000E-01 0.          0.          0.          0.          -.25000E-01 0.          0.
```

CONSISTENT MASS ARRAY --- SM

```
 .58333E-09  .29167E-09  .12500E-09  .25000E-09  .53333E-09  .29167E-09  .58333E-09  .25000E-09  .12500E-09  .53333E-09
 .12500E-09  .25000E-09  .41667E-09  .20833E-09  .46667E-09  .25000E-09  .12500E-09  .20833E-09  .41667E-09  .46667E-09
 .53333E-09  .53333E-09  .46667E-09  .46667E-09  .12800E-08
```

FULL CONSISTENT MASS MATRIX --- SMASS

.58333E-09	0.	0.	0.	0.	.29167E-09	0.	0.	0.	0.
.12500E-09	0.	0.	0.	0.	.25000E-09	0.	0.	0.	0.
.53333E-09	0.	0.	0.	0.	0.	.58333E-09	0.	0.	0.
0.	.29167E-09	0.	0.	0.	0.	.12500E-09	0.	0.	0.
0.	.25000E-09	0.	0.	0.	0.	.53333E-09	0.	0.	0.
0.	0.	.58333E-09	0.	0.	0.	0.	.29167E-09	0.	0.
0.	0.	.12500E-09	0.	0.	0.	0.	.25000E-09	0.	0.
0.	0.	.53333E-09	0.	0.	0.	0.	0.	.77778E-13	0.
0.	0.	0.	.38889E-13	0.	0.	0.	0.	.16667E-13	0.
0.	0.	0.	.33333E-13	0.	0.	0.	0.	.71111E-13	0.
0.	0.	0.	0.	.77778E-13	0.	0.	0.	0.	.38889E-13
0.	0.	0.	0.	.16667E-13	0.	0.	0.	0.	.33333E-13
0.	0.	0.	0.	.71111E-13	.29167E-09	0.	0.	0.	0.
.58333E-09	0.	0.	0.	0.	.25000E-09	0.	0.	0.	0.
.12500E-09	0.	0.	0.	0.	.53333E-09	0.	0.	0.	0.
0.	.29167E-09	0.	0.	0.	0.	.58333E-09	0.	0.	0.
0.	.25000E-09	0.	0.	0.	0.	.12500E-09	0.	0.	0.
0.	.53333E-09	0.	0.	0.	0.	0.	.29167E-09	0.	0.
0.	0.	.58333E-09	0.	0.	0.	0.	.25000E-09	0.	0.
0.	0.	.12500E-09	0.	0.	0.	0.	.53333E-09	0.	0.
0.	0.	0.	.38889E-13	0.	0.	0.	0.	.77778E-13	0.
0.	0.	0.	.33333E-13	0.	0.	0.	0.	.16667E-13	0.
0.	0.	0.	.71111E-13	0.	0.	0.	0.	0.	.38889E-13
0.	0.	0.	0.	.77778E-13	0.	0.	0.	0.	.33333E-13
0.	0.	0.	0.	.16667E-13	0.	0.	0.	0.	.71111E-13
.12500E-09	0.	0.	0.	0.	.25000E-09	0.	0.	0.	0.
.41667E-09	0.	0.	0.	0.	.20833E-09	0.	0.	0.	0.
.46667E-09	0.	0.	0.	0.	0.	.12500E-09	0.	0.	0.
0.	.25000E-09	0.	0.	0.	0.	.41667E-09	0.	0.	0.
0.	.20833E-09	0.	0.	0.	0.	.46667E-09	0.	0.	0.
0.	0.	.12500E-09	0.	0.	0.	0.	.25000E-09	0.	0.
0.	0.	.41667E-09	0.	0.	0.	0.	.20833E-09	0.	0.
0.	0.	.46667E-09	0.	0.	0.	0.	0.	.16667E-13	0.
0.	0.	0.	.33333E-13	0.	0.	0.	0.	.55556E-13	0.
0.	0.	0.	.27778E-13	0.	0.	0.	0.	.62222E-13	0.
0.	0.	0.	0.	.16667E-13	0.	0.	0.	0.	.33333E-13
0.	0.	0.	0.	.55556E-13	0.	0.	0.	0.	.27778E-13
0.	0.	0.	0.	.62222E-13	.25000E-09	0.	0.	0.	0.
.12500E-09	0.	0.	0.	0.	.20833E-09	0.	0.	0.	0.
.41667E-09	0.	0.	0.	0.	.46667E-09	0.	0.	0.	0.
0.	.25000E-09	0.	0.	0.	0.	.12500E-09	0.	0.	0.
0.	.20833E-09	0.	0.	0.	0.	.41667E-09	0.	0.	0.
0.	.46667E-09	0.	0.	0.	0.	0.	.25000E-09	0.	0.
0.	0.	.12500E-09	0.	0.	0.	0.	.20833E-09	0.	0.
0.	0.	.41667E-09	0.	0.	0.	0.	.46667E-09	0.	0.
0.	0.	0.	.33333E-13	0.	0.	0.	0.	.16667E-13	0.
0.	0.	0.	.27778E-13	0.	0.	0.	0.	.55556E-13	0.

0.	0.	0.	.62222E-13	0.	0.	0.	0.	0.	.33333E-13
0.	0.	0.	0.	.16667E-13	0.	0.	0.	0.	.27778E-13
0.	0.	0.	0.	.55556E-13	0.	0.	0.	0.	.62222E-13
.53333E-09	0.	0.	0.	0.	.53333E-09	0.	0.	0.	0.
.46667E-09	0.	0.	0.	0.	.46667E-09	0.	0.	0.	0.
.12800E-08	0.	0.	0.	0.	0.	.53333E-09	0.	0.	0.
0.	.53333E-09	0.	0.	0.	0.	.46667E-09	0.	0.	0.
0.	.46667E-09	0.	0.	0.	0.	.12800E-08	0.	0.	0.
0.	0.	.53333E-09	0.	0.	0.	0.	.53333E-09	0.	0.
0.	0.	.46667E-09	0.	0.	0.	0.	.46667E-09	0.	0.
0.	0.	.12800E-08	0.	0.	0.	0.	0.	.71111E-13	0.
0.	0.	0.	.71111E-13	0.	0.	0.	0.	.62222E-13	0.
0.	0.	0.	.62222E-13	0.	0.	0.	0.	.17067E-12	0.
0.	0.	0.	0.	.71111E-13	0.	0.	0.	0.	.71111E-13
0.	0.	0.	0.	.62222E-13	0.	0.	0.	0.	.62222E-13
0.	0.	0.	0.	.17067E-12					

THE COMPUTATION TIME FOR THIS ELEMENT WAS .020 SECONDS.

REFERENCES

1. Andersen, C. M.; and Noor, Ahmed K.: A Computerized Symbolic Integration Technique for Development of Triangular and Quadrilateral Composite Shallow-Shell Elements. NASA TN D-8067, 1975.

2. Hull, T. E.: Would You Believe Structured FORTRAN? ACM Signum Newsletter, vol. 8, no. 4, Oct. 1973, pp. 13-16.

3. Martin, W. A.; and Fateman, R. J.: The MACSYMA System. Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, S. R. Petrick, ed., Assoc. Comput. Mach., c.1971, pp. 59-75.

4. Moses, Joel: MACSYMA – The Fifth Year. ACM SIGSAM Bull., vol. 8, no. 3, Aug. 1974, pp. 105-110.

5. Mathlab Group: MACSYMA Reference Manual. Version Eight. Massachusetts Inst. Technol., c.1975.

6. Andersen, C. M.; and Noor, Ahmed K.: Use of Group-Theoretic Methods in the Development of Nonlinear Shell Finite Elements. Symmetry, Similarity and Group Theoretic Methods in Mechanics, P. G. Glockner and M. C. Singh, eds., Univ. Calgary, Aug. 1974, pp. 533-558.

TABLE I.- INPUT VARIABLES CONTAINED IN FIRST 100 WORDS OF LABELED COMMON SPACE

Position in SPACE	FORTRAN name	Variable name	Routine where variable is used	Description
1	C11	C_{1111}	LINSTF	Extensional stiffnesses
2	C12	C_{1122}		
3	C16	C_{1112}		
4	C22	C_{2222}		
5	C26	C_{2212}		
6	C66	C_{1212}		
7	F11	F_{1111}		Stiffness interaction coefficients
8	F12	F_{1112}		
9	F16	F_{1112}		
10	F22	F_{2222}		
11	F26	F_{2212}		
12	F66	F_{1212}		
13	D11	D_{1111}		Bending stiffnesses
14	D12	D_{1122}		
15	D16	D_{1212}		
16	D22	D_{2222}		
17	D26	D_{2212}		
18	D66	D_{1212}		
19	C55	C_{1313}		Transverse shear stiffnesses
20	C44	C_{2323}		
21	C54	C_{1323}		
22	EN1	$\tilde{N}_{11}$		Prestress coefficients
23	EN2	$\tilde{N}_{22}$		
24	EN12	$\tilde{N}_{12}$		

TABLE I. - Concluded

Position in SPACE	FORTRAN name	Variable name	Routine where variable is used	Description
25	NSF		ELEMENT, INTGRAL, TRI, QUAD, LINSTF, MASS, PRINT	Number of shape functions associated with finite element
26	CURVE		LINSTF	To be set to FALSE for flat plate and to TRUE if shell has curvature
27	SGFLAG		LINSTF, STORE, PRINT	To be set to TRUE if SG is to be evaluated
28	SMFLAG		SGPM, STORE, PRINT	To be set to TRUE if SM is to be evaluated
29	SPFLAG		↓	To be set to TRUE if SP is to be evaluated
30	PRFLAG		ELEMENT	To be set to TRUE if the evaluated characteristic arrays are to be printed
31	RHO	ρ	MASS	Density
32	H	h	↓	Thickness of shell
33-36	X(4)*	$x_1^1, x_1^2, x_1^3, x_1^4$	TRI, QUAD	x-coordinates of corner nodes
37-40	Y(4)*	$x_2^1, x_2^2, x_2^3, x_2^4$	TRI, QUAD	y-coordinates of corner nodes
41-50	Q1(10)*	k_{11}^i (i = 1 → m)	LINSTF	Nodal values of curvature components
51-60	Q2(10)*	k_{22}^i (i = 1 → m)	↓	
61-70	Q12(10)*	k_{12}^i (i = 1 → m)	↓	
71-80	P(10)*	p^i (i = 1 → m)	LODVEC	Nodal values of transverse load
81-90	P1(10)*	p_1^i (i = 1 → m)	↓	Nodal values of in-plane loads
91-100	P2(10)*	p_2^i (i = 1 → m)	↓	

*The dimensions of the FORTRAN arrays are given in parentheses.

TABLE II.- LISTING AND DESCRIPTION OF ROUTINES WHICH COMPRISE SYMINSE PROGRAM

Primary overlay	Routine	Field length	Files referenced	Description	Relevant equations from ref. 1
0	ELEMENT	137		*Entering program for the SYMINSE program*	
	INTGRAL	333		Governs evaluation of A-, B-, and C-integrals	
	STORE	141	Write 3 Write 4 Write 8 Write 9	Stores characteristic arrays on disk	
1	SETUP	363	Read 1 Write 2 Write 6	Governs evaluation of integration arrays	
	SETA	515	Read 1 Write 2 Write 6	Sets up integration arrays for A-integrals	(53), (57)
	SETB	556	Read 1 Write 2 Write 6	Sets up integration arrays for B-integrals	(54), (58)
	SETC	665	Read 1 Write 2 Write 6	Sets up integration arrays for C-integrals	(55), (59)
2	TRI	343	Read 2	Evaluates A-, B-, and C-integrals for triangular elements	(23) to (27)
3	QUAD	740	Read 2	Evaluates logical variables PARA and TRAP, evaluates A- and B-integrals, and evaluates C-integrals if PARA = TRUE	(28), (29), (43) to (45)
	BLOG	112		Evaluates a logarithmic function	(33)
	ELOG	105		Evaluates a logarithmic function	(42)
	WLOG1	272		Evaluates a logarithmic function	(38)
	WLOG2	375		Evaluates a logarithmic function	(38)
4	TRAP5	272		Performs group transformations preparatory to evaluation of C-integrals for 4-node trapezoidal elements	(66), (67)
	XDNDN	575		Evaluates C-integrals for 4-node trapezoidal elements	(41)

TABLE II. - Concluded

Primary overlay	Routine	Field length	Files referenced	Description	Relevant equations from ref. 1
5	TRAP9	272		Performs group transformations preparatory to evaluation of C-integrals for 8-node trapezoidal elements	(66), (67)
	XDNDN	2435		Evaluates C-integrals for 8-node trapezoidal elements	(41)
6	QUAD5	343		Performs group transformations preparatory to evaluation of C-integrals for 4-node elements	(66) to (69)
	XDNDN	2215		Evaluates C-integrals for 4-node trapeziums	(37)
7	QUAD81	421		Performs group transformations preparatory to evaluation of first set of C-integrals for 8-node trapeziums	(66), (67)
	XDNDN	3420		Evaluates first set of C-integrals for 8-node trapeziums	(30)
10_8	QUAD82	422		Performs group transformations preparatory to evaluation of second set of C-integrals for 8-node trapeziums	(66), (67)
	XDNDN	4247		Evaluates second set of C-integrals for 8-node trapeziums	(30)
11_8	QUAD9	422		Performs group transformations preparatory to evaluation of third set of C-integrals for trapeziums with NSF = 9	(66), (67)
	XDNDN	2360		Evaluates third set of C-integrals for trapeziums with NSF = 9	(30)
12_8	SGPM	54		Governs the evaluation of the characteristic arrays SS, SG, SP, and SM	
	LINSTF	1225		Evaluates the stiffness SS and the geometric stiffness SG	Equations of appendix B
	LODVEC	121		Evaluates the consistent load SP	
	MASS	101		Evaluates the consistent mass SM	
	XDNDN	75		Retrieves C-integrals	(22)
	XNNDN	74		Retrieves B-integrals	(22)
	XNNNN	163		Retrieves A-integrals	
13_8	PRINT	322	Write 6	Displays the characteristic arrays SS, SG, SP, SM, and SMASS	

TABLE III.- FORTRAN VARIABLES STORED IN FIXED POSITIONS IN COMMON SPACE EXCLUDING THE INPUT VARIABLES LISTED IN TABLE I

Position in SPACE	FORTRAN name	Routine where variable is defined	Routine where variable is used	Description
1-	SPACE( )*			Alias for any position in this common block
101-107	NRECORD(7)*	SETUP	TRI, QUAD	Governs values of m for which the integration arrays are to be set up
108	LIMIT	INTGRAL	TRAP5, TRAP9, QUAD5	Governs the number of integrals to be computed
109-120	INDX(12)*	SETUP		Alias for the next 12 variables
109	IA		SETUP, TRI, QUAD	Number of A-integrals to be evaluated, equal to $(r+1)(r+2)(r+3)(r+4)/24$
110	IB			One-half the number of B-integrals to be evaluated, equal to $r(r+1)(r+2)/2$
111	IC		Many	One-fourth the number of C-integrals to be evaluated, equal to $r(r+1)/2$
112	JA		SETUP	Number of representative A-integrals
113	JB			Number of representative B-integrals
114	JC		TRI, QUAD	Number of representative C-integrals
115	NNE		ELEMENT, LINSTF, LODVEC, MASS, STORE, PRINT	Number of nodes per element
116	ISS		SETUP, TRI, QUAD, LINSTF, STORE, PRINT	Storage of the stiffness array SS begins at ISS + 1
117	ISG		SETUP, LINSTF, STORE, PRINT	Storage of the geometric stiffness array SG begins in ISG + 1
118	IXC		SETUP, TRI, QUAD, TRAP5, TRAP9, QUAD5, QUAD81, QUAD82, QUAD9, LODVEC, MASS, STORE, PRINT	Storage of the C-integrals XC and also of the distributed load array SP begins in IXC + 1
119	IXB		SETUP, TRI, QUAD, XNNDN	Storage of the B-integrals XB begins at IXB + 1
120	IXA		SETUP, TRI, QUAD, XNNNN	Storage of the A-integrals XA begins at IXA + 1

*The dimensions of the FORTRAN arrays are given in parentheses.

TABLE III. - Concluded

Position in SPACE	FORTRAN name	Routine where variable is defined	Routine where variable is used	Description
121	X1 or XX1	QUAD	TRAP5, TRAP9, QUAD5, QUAD81, QUAD82, QUAD9	V_{11}
122	X2 or XX2			V_{12} — Linear combinations of x-coordinates of corner nodes
123	X3 or XX3			V_{13}
124	Y1 or YY1			V_{21}
125	Y2 or YY2			V_{22} — Linear combinations of y-coordinates of corner nodes
126	Y3 or YY3			V_{23}
127	Z1 or ZZ1			U_1
128	Z2 or ZZ2			U_2 — Bilinear combinations of x- and y-coordinates of corner nodes
129	Z3 or ZZ3			U_3
130	ALG1		QUAD81, QUAD82, QUAD9	$L_1(s,t)$
131	ALG2			$L_2(s,t)$ — Logarithmic functions generated by the function subroutine BLOG
132	ALG3			$L_2(t,s)$
130	DLOG		TRAP5, TRAP9	$L(s)$ Logarithmic function generated by the function subroutine ELOG
131	RS2			$\max(s^2,t^2)$
132	C LOG			$\left\{\log[(1+s)/(1-s)] - 2s\right\}/s^3$
130	VLG1		QUAD5	$\overline{L}_1(s,t)$
131	VLG2			$\overline{L}_2(s,t)$ — Logarithmic functions generated by the function subroutines WLOG1 and WLOG2
132	VLG3			$\overline{L}_2(t,s)$
133	ULG1			$\left[L_1(s,t) + 2st\right]/(st)^3$
134	ULG2			$\left[L_2(s,t) - 2t\right]/(t)^3$
135	ULG3			$\left[L_2(t,s) - 2s\right]/(s)^3$
136	PARA		QUAD, INTGRAL	Logical variable which is set to TRUE for parallelograms
137	TRAP			Logical variable which is set to TRUE for trapezoids
121-	QA(2,JA)*	SETUP	SETUP	$\mathcal{R}_1^m$, $\mathcal{R}_2^m$
	QA(4,JA)*			$\mathcal{R}_1^m$, $\mathcal{R}_2^m$, $\mathcal{R}_3^m$, $\mathcal{R}_4^m$
	QB(3,JB)*			$\mathcal{S}_1^{\overline{m}}$, $\mathcal{S}_2^{\overline{m}}$, $\mathcal{S}_3^{\overline{m}}$
	QB(4,JB)*			$\mathcal{S}_1^{\overline{m}}$, $\mathcal{S}_2^{\overline{m}}$, $\mathcal{S}_3^{\overline{m}}$, $\mathcal{S}_4^{\overline{m}}$
	QC(5,JC)*			$\mathcal{T}_1^{\overline{\overline{m}}}$, $\mathcal{T}_2^{\overline{\overline{m}}}$, $\mathcal{T}_3^{\overline{\overline{m}}}$, $\mathcal{T}_4^{\overline{\overline{m}}}$, $\mathcal{T}_5^{\overline{\overline{m}}}$

*The dimensions of the FORTRAN arrays are given in parentheses.

TABLE IV.- LISTING OF DYNAMICALLY ALLOCATED ARRAYS AND THEIR POSITIONS IN LABELED COMMON SPACE

FORTRAN name	Starting position	Terminal position	Routine where array is defined or used	Description
KA	IXA + 1	IXA + IA	SETUP	Superscript m which determines a representative A-integral
KB	IXB + 1	IXB + IB	SETUP	Superscript $\overline{m}$ which determines a representative B-integral
KC	IXC + 1	IXC + IC	Many	Superscript $\overline{\overline{m}}$ which determines a representative C-integral
LA	IXA + 1 - IA	IXA	SETUP	Superscript n which determines a group transformation
LB	IXB + 1 + IB	IXB + 2*IB	SETUP	Superscript $\overline{n}$ which determines a group transformation
LC	IXC + 1 + IC	IXC + 2*IC	Many	Superscript $\overline{\overline{n}}$ which determines a group transformation
QA1	IXA + 1	IXA + IA	SETUP, TRI, QUAD	Coefficients $\mathcal{R}$ in A-integrals
QA2	IXA + 1 - IA	IXA	SETUP, QUAD	
QA3	IXA + 1 - 2*IA	IXA - IA	SETUP, QUAD	
QB1	IXB + 1 + IB	IXB + 2*IB	SETUP, TRI, QUAD	Coefficients $\mathcal{S}$ in B-integrals
QB2	IXB + 1	IXB + IB	SETUP, QUAD	
QB3	IXB + 1 - IB	IXB	SETUP, QUAD	
QC1	IXC + 1 + 3*IC	IXC + 4*IC	SETUP, TRI, QUAD	Coefficients $\mathcal{T}$ in C-integrals
QC2	IXC + 1 + 2*IC	IXC + 3*IC	SETUP, TRI, QUAD	
QC3	IXC + 1 + IC	IXC + 2*IC	SETUP, TRI, QUAD	
QC4	IXC + 1	IXC + IC	SETUP, TRI, QUAD	
XA	IXA + 1	IXA + IA	TRI, QUAD, XNNNN	Evaluated A-integrals $A^{ijk\ell}$ with $i \geq j \geq k \geq \ell$
XB	IXB + 1	IXB + 2*IB	TRI, QUAD, XNNDN	Evaluated B-integrals B_{α}^{ijk} with $i \geq j$
XC	IXC + 1	IXC + 4*IC	Many	Evaluated C-integrals $C_{\alpha\beta}^{ij}$ with $i \geq j$
SS	ISS + 1	ISS + 25*NSF*NSF	LINSTF, STORE, PRINT	Stiffness array
SG	ISG + 1	ISG + NSF*NSF	LINSTF, STORE, PRINT	Abbreviated geometric stiffness array
SP	IXC + 1	IXC + 5*NNE	LODVEC, STORE, PRINT	Distributed load array
SM	IXC + 1 + 5*NNE	IXC + 5*NNE + NSF*NSF	MASS, STORE, PRINT	Abbreviated distributed mass array
SMASS	ISS + 1	ISS + 25*NSF*NSF	PRINT	Full distributed mass array

TABLE V.- THE MORE IMPORTANT FORTRAN VARIABLES IN LABELED COMMON TEMP

FORTRAN name	Routine where variable is defined or used	Variable name and description	Relevant equations from ref. 1
LENGTH	SETUP	Minimum length required for common SPACE as dependent on NSF	
TRI	↓	Logical variable set to TRUE if element is triangular	
AREA	TRI	U_1 Area of a triangular element	(27)
R, S	QUAD	s, t	(34)
RR, SS	↓	$\tilde{s}$, $\tilde{t}$	(39)
XX1, XX2, XX3	TRAP5, TRAP9, QUAD5, QUAD81, QUAD82, QUAD9	$V_{1,1}$, $V_{1,2}$, $V_{1,3}$	(31)
YY1, YY2, YY3	↓	$V_{2,1}$, $V_{2,2}$, $V_{2,3}$ } Linear functions of coordinates of corner nodes	(31)
X1, X2, X3	↓	$V^{\bar{\bar{n}}}_{1,1}$, $V^{\bar{\bar{n}}}_{1,2}$, $V^{\bar{\bar{n}}}_{1,3}$	(66)
Y1, Y2, Y3	↓	$V^{\bar{\bar{n}}}_{2,1}$, $V^{\bar{\bar{n}}}_{2,2}$, $V^{\bar{\bar{n}}}_{2,3}$	(66)
R, S	↓	$s^{\bar{\bar{n}}}$, $t^{\bar{\bar{n}}}$	(67)
VLG1, VLG2, VLG3	QUAD5	$\bar{L}_1(s,t)$, $\bar{L}_2(s,t)$, $\bar{L}_2(t,s)$	(38)
VLOG1, VLOG2, VLOG3	↓	$L^{\bar{\bar{n}}}_1(s,t)$, $L^{\bar{\bar{n}}}_2(s,t)$, $L^{\bar{\bar{n}}}_2(t,s)$	(68)
ALG1, ALG2, ALG3	QUAD81, QUAD82, QUAD9	$L_1(s,t)$, $L_2(s,t)$, $L_2(t,s)$ } Logarithmic functions	(33)
ALOG1, ALOG2, ALOG3	↓	$L_1(s^{\bar{\bar{n}}},t^{\bar{\bar{n}}})$, $L_2(s^{\bar{\bar{n}}},t^{\bar{\bar{n}}})$, $L_2(t^{\bar{\bar{n}}},s^{\bar{\bar{n}}})$	
D, E, FF, G	↓	$s^{\bar{\bar{n}}}/t^{\bar{\bar{n}}}$, $-1/t^{\bar{\bar{n}}}$, $1/s^{\bar{\bar{n}}}$, $t^{\bar{\bar{n}}}/s^{\bar{\bar{n}}}$	
NDFE	LINSTF	Number degrees of freedom per element	

TABLE VI.- FIELD LENGTHS AND AVERAGE CPU TIMES FOR 14 SAMPLE PROBLEMS

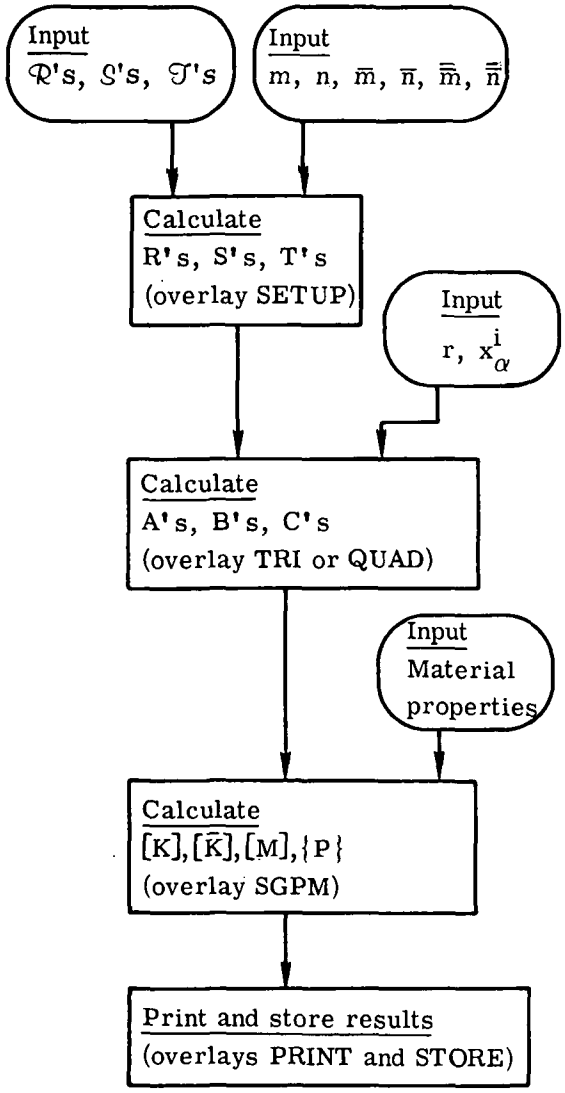
[System used FORTRAN Extended (Version 4) compiler under CONTROL DATA Network Operating System (NOS 1.0) running on CONTROL DATA CYBER 175 computer system]

Element designation (a)	Element shape	Required field lengths (octal) (b)	Total CPU time for plate elements,[c] msec	Total CPU time for shell elements, msec
SQ4	Parallelogram	41 412	6	11
	Trapezoid	41 412	9	13
	Trapezium	41 412	10	16
SQ5	Parallelogram	42 232	8	16
	Trapezoid	42 232	10	20
	Trapezium	42 232	16	23
ST6	Triangle	41 743	11	30
SQ8	Parallelogram	46 123	17	71
	Trapezoid	46 123	28	80
	Trapezium	46 521	59	111
SQ9	Parallelogram	50 033	20	85
	Trapezoid	50 033	32	99
	Trapezium	50 431	74	137
ST10	Triangle	50 732	24	138

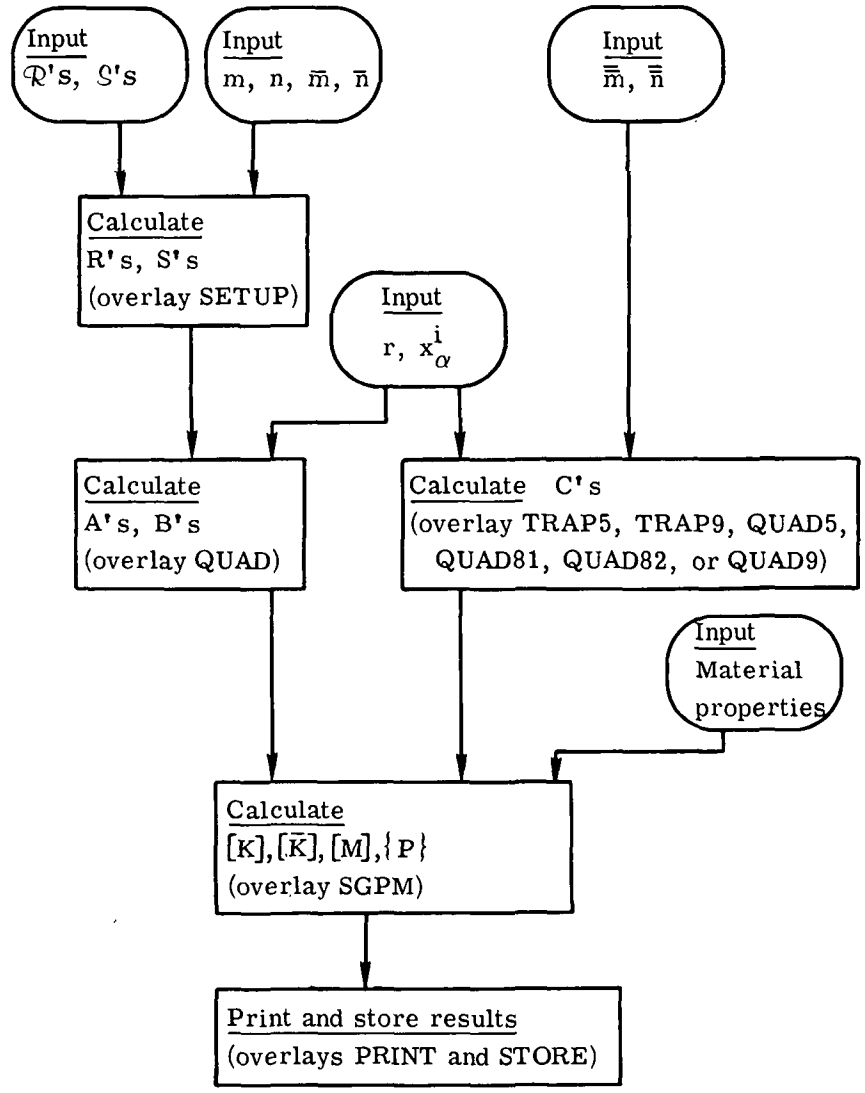
[a]Correspond to designations in reference 1.

[b]After routine SETUP has been executed.

[c]Include bending-extensional coupling.



(a) For triangular and parallelogram elements.



(b) For trapezoidal and trapezium elements.

Figure 1.- Logical organization of SYMINSE program.

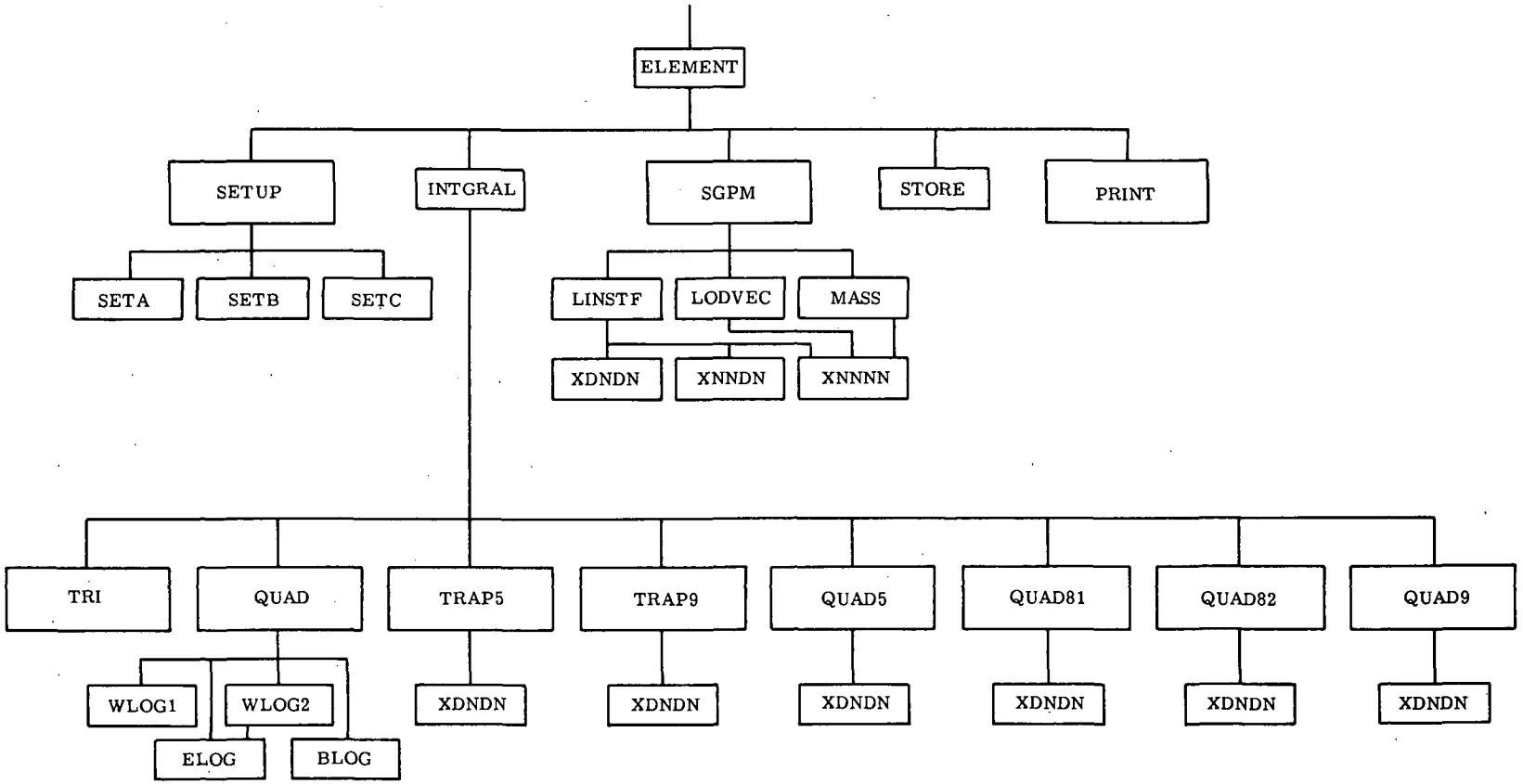


Figure 2.- Subroutine linkages for SYMINSE program. The large boxes represent main programs of primary overlays. The small boxes represent subroutines.

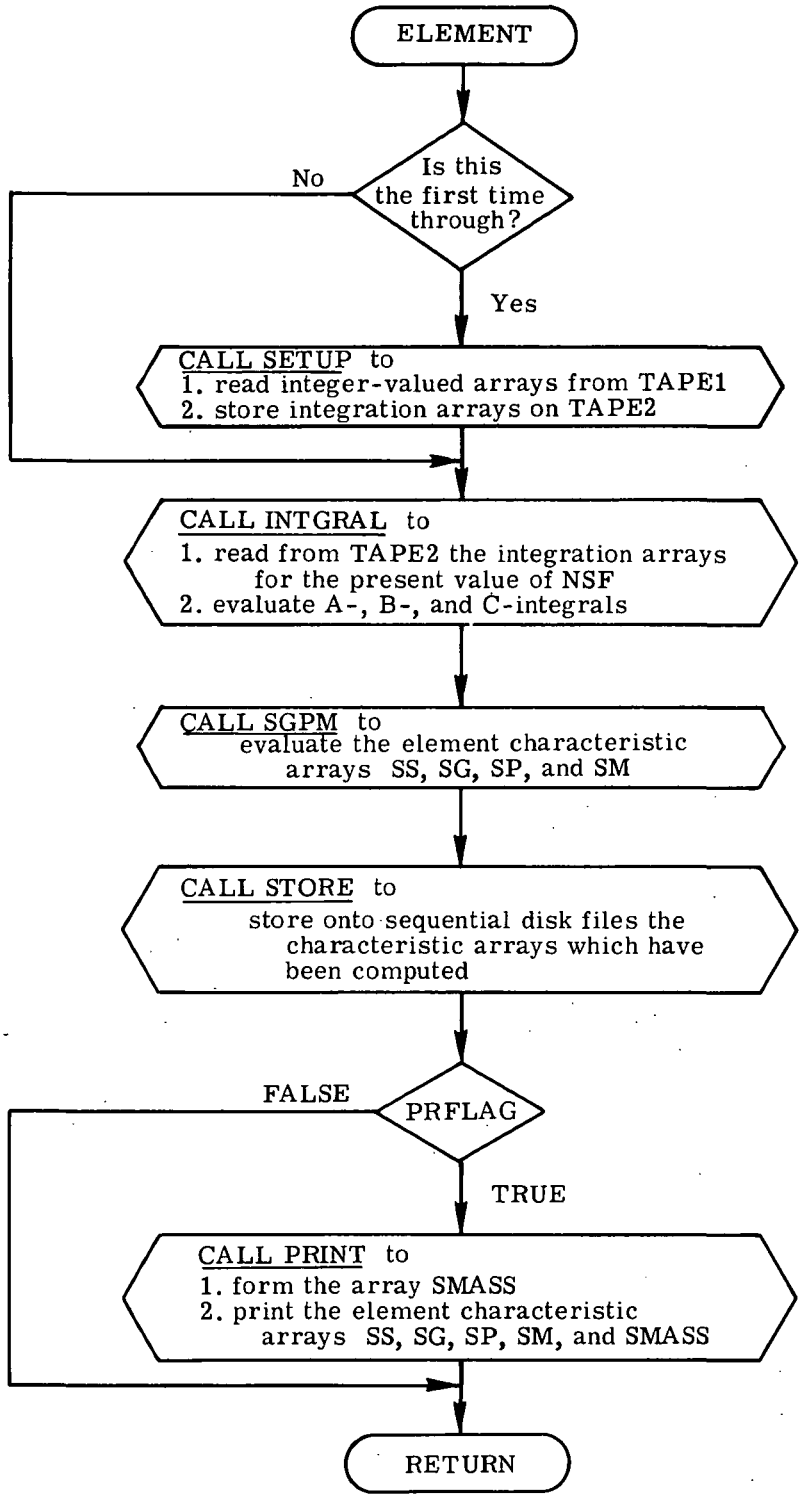


Figure 3.- Flow chart for ELEMENT, the top-level routine in SYMINSE program.

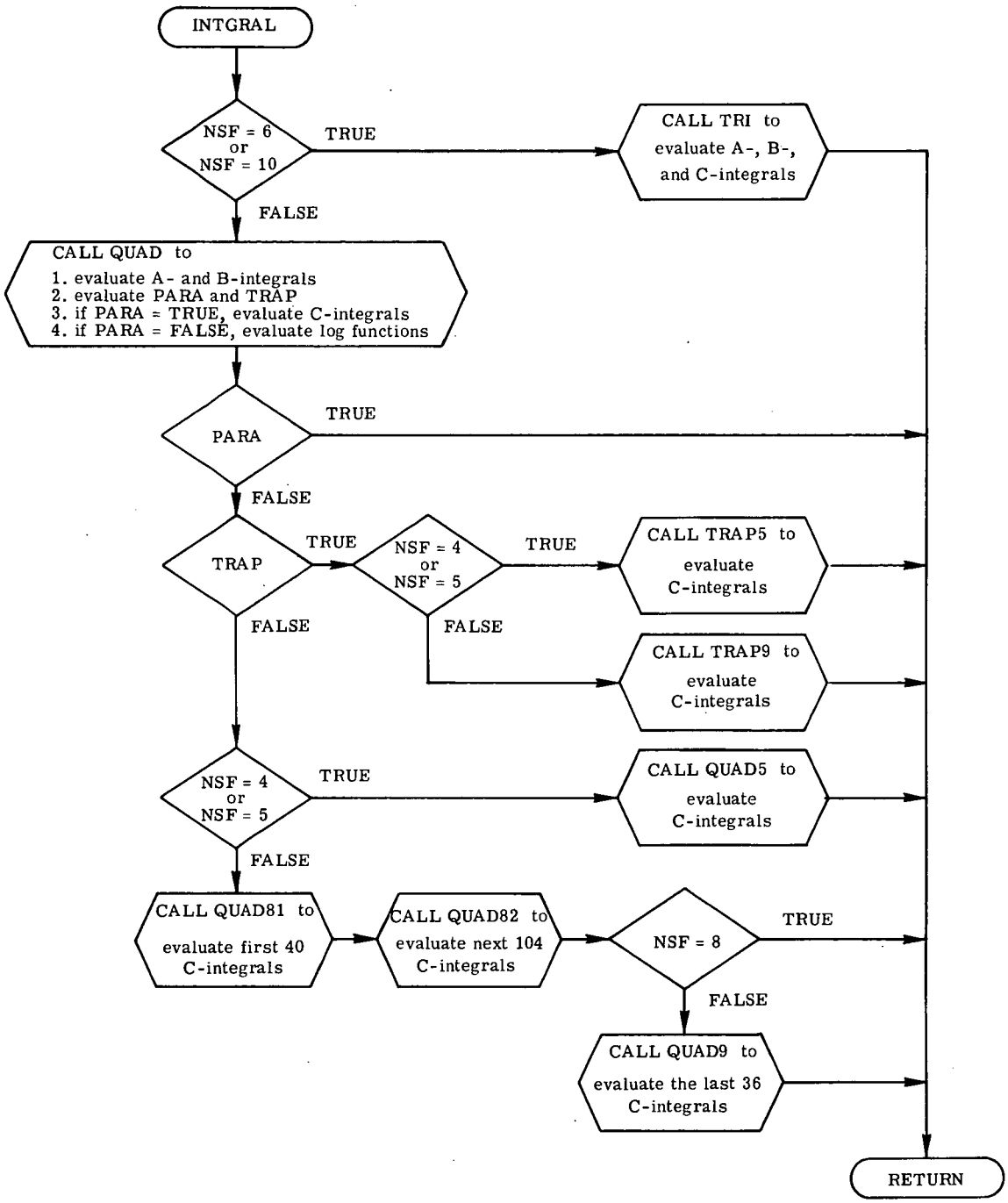


Figure 4.- Flow chart for routine INTGRAL, which governs evaluation of A-, B-, and C-integrals.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

SPECIAL FOURTH-CLASS RATE BOOK

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
451

U.S.MAIL

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546