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)) Finite elements Composite materials Shells Symbolic integration Computers Anisotropy Group theory MACSYMA Stiffness		18. Distribution Statement Unclassified – Unlimited Subject Category 39	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 126	22. Price* $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

$A^{ijk\ell}, B^{ijk}_{\alpha}, C^{ij}_{\alpha\beta}$	basic integrals
$[K]$	element stiffness matrix
K^{ij}_{IJ}	stiffness coefficients of shell element
$[\bar{K}]$	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.

$\overline{K}_{IJ}^{ij}$	geometric stiffness coefficients of shell element
$[M]$	consistent mass matrix
M_{IJ}^{ij}	consistent mass coefficients of shell element
$m,\overline{m},\overline{\overline{m}}$	superscript index designating particular representative A-, B-, or C-integrals, respectively
$n,\overline{n},\overline{\overline{n}}$	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 <u>n</u>umber r of <u>s</u>hape <u>f</u>unctions 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 <u>n</u>umber of <u>n</u>odes per <u>e</u>lement, is set by the SYMINSE program equal to $\text{NSF} - 1$ 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, <u>exact</u> 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 $\text{NNE} = 4$ and on eq. (30) for $\text{NNE} = 8$).

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, $\bar{m}$, $\bar{n}$, $\bar{\bar{m}}$, and $\bar{\bar{n}}$ 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, $\bar{m}$, $\bar{n}$, $\bar{\bar{m}}$, and $\bar{\bar{n}}$, 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 $i \geqq j \geqq k \geqq \ell$. 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 $\bar{m}(i,j,k)$ and $\bar{n}(i,j,k)$ for $i \geqq j$, and the arrays KC and LC contain the values of $\bar{\bar{m}}(i,j)$ and $\bar{\bar{n}}(i,j)$ for $i \geqq j$.

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 $\overline{K}_{IJ}^{ij}$, 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

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

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

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

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

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

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 SGFLAC 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	3 and 4
	8 or 9	3 and 5
Trapezium	4 or 5	3 and 6
	8	3, 7, and 10_8
	9	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
C***********************************************************************
C
C     THIS SUBROUTINE IS THE TOP-LEVEL ROUTINE OF THE  S Y M I N S E
C        (SYMBOLICALLY INTEGRATED SHELL ELEMENTS) PROGRAM
C        FOR EVALUATION OF THE LINEAR STIFFNESS SS, THE GEOMETRIC
C        STIFFNESS SG, THE CONSISTENT LOAD SP AND THE CONSISTENT MASS SM
C        FOR A DOUBLY-CURVED ANISOTROPIC SHALLOW-SHELL FINITE ELEMENT.
C
C
C
C     THIS PROGRAM WAS WRITTEN BY
C                  CARL M. ANDERSEN
C                  SENIOR RESEARCH ASSOCIATE IN MATHEMATICS
C                       AND COMPUTER SCIENCE
C                  DEPARTMENT OF MATHEMATICS
C                  COLLEGE OF WILLIAM AND MARY
C                  WILLIAMSBURG, VA. 23185
C     WITH THE ASSISTANCE OF
C                  JOHN T. BOWEN
C                  N.A.S.A. LANGLEY RESEARCH CENTER
C                  HAMPTON, VA. 23665
C
C
C     THE RESULTS (SS,SG,SP AND SM) ARE STORED IN BINARY SEQUENTIAL
C        FILES ON DISC LOGICAL UNITS  3,4,8, AND 9, RESPECTIVELY,
C        FOR FUTURE ASSEMBLY OF THE SYSTEM EQUATIONS.
C           EACH  SS  RECORD CONTAINS  25*NSF*NSF  WORDS
C           EACH  SG  RECORD CONTAINS     NSF*NSF  WORDS
C           EACH  SP  RECORD CONTAINS      5*NNE   WORDS
C           EACH  SM  RECORD CONTAINS     NSF*NSF  WORDS
C
C     NSF  IS THE NUMBER OF SHAPE FUNCTIONS ASSOCIATED WITH THE ELEMENT.
C        FOR A TRIANGULAR ELEMENT  NSF  MAY TAKE THE VALUES  6  OR 10.
C        FOR A QUADRILATERAL ELEMENT NSF MAY TAKE THE VALUES 4,5,8 OR 9.
C     NNE  IS THE NUMBER OF NODES ASSOCIATED WITH THE ELEMENT.
C        NNE = NSF     IF  NSF=4,6,8 OR 10.
C        NNE = NSF-1   IF  NSF=5 OR 9.
C
C     THE MAIN BLOCK OF INPUT COMPRISES THE FIRST ONE HUNDRED WORDS OF
C        COMMON/SPACE/.  NONE OF THE WORDS IN THIS BLOCK ARE CHANGED BY
C        THE SYMINSE PROGRAM, BUT ANY OF THEM MAY BE CHANGED,IF DESIRED,
C        BY OTHER PROGRAMS BETWEEN CALLS TO ELEMENT.
C     THE VARIABLES CONTAINED IN THIS BLOCK ARE AS FOLLOWS.
C        THE FIRST  21 WORDS ARE THE MATERIAL STIFFNESS PROPERTIES ---
C          C11,C12,C16,C22,C26,C66,F11,F12,F16,F22,F26,F66,
C          D11,D12,D16,D22,D26,D66,C55,C44,C54.
C        THE NEXT  3  WORDS ARE THE PRESTRESS COEFFICIENTS EN1,EN2,EN12.
C        THE NEXT WORD CONTAINS  NSF.
C        THE NEXT  5  WORDS CONTAIN LOGICAL VARIABLES
C           CURVE -- IF FALSE, ALL CURVATURE DEPENDENT CONTRIBUTIONS
C              TO SS ARE BYPASSED IN THE SUBROUTINE  LINSTF. THIS SPEEDS
C              UP THE COMPUTATION FOR THE CASE OF ZERO CURVATURE.
C           SGFLAG -- IF TRUE THEN THE GEOMETRIC STIFFNESS SG IS
C              COMPUTED. IF FALSE SG WILL CONTAIN GARBAGE.
C           SMFLAG -- IF TRUE THEN THE CONSISTENT MASS SM IS COMPUTED.
C              IF FALSE THE SUBROUTINE MASS WILL NOT BE CALLED AND
C              SM WILL CONTAIN GARBAGE.
C           SPFLAG -- IF TRUE THEN THE CONSISTENT LOAD SP IS COMPUTED.
C              IF FALSE THE SUBROUTINE LOOVEC WILL NOT BE CALLED AND
C              SP WILL CONTAIN GARBAGE.
C           PRFLAG -- IF TRUE THEN THE PROGRAM OUTPUT (OVERLAY 13,0) IS
C              CALLED TO PRINT THE RESULTS  SS,SG,SP AND SM.
```

```
C         THE NEXT WORD CONTAINS THE DENSITY,RHO, OF THE MATERIAL.
C         THE NEXT WORD CONTAINS THE THICKNESS,H, OF THE MATERIAL.
C         THE NEXT  4  WORDS CONTAIN THE X-COORDINATES OF THE CORNER
C            NODES.  FOR TRIANGLES, ONLY THE FIRST THREE OF THESE WORDS
C            ARE USED.
C         THE NEXT  4  WORDS CONTAIN THE Y-COORDINATES OF THE CORNER
C            NODES.  FOR TRIANGLES, ONLY THE FIRST THREE OF THESE WORDS
C            ARE USED.
C         THE NEXT  30  WORDS ARE RESERVED FOR NODAL VALUES OF THE
C            CURVATURES Q1,Q2 AND Q12.
C         FINALLY THE LAST  30  WORDS ARE RESERVED FOR THE NODAL VALUES
C            OF THE PRESSURES  P1,P2 AND P12.
C
C      IN ADDITION TO THE ABOVE ONE HUNDRED WORDS OF INPUT DATA, THE
C         CALLING PROGRAM MUST ON THE FIRST CALL TO ELEMENT PROVIDE THE
C         VECTOR NRECORD(7).  NRECORD IS SUBSEQUENTLY MODIFIED BY THE
C         PROGRAM SETUP.
C
C      THE VARIABLES  NNE,ISS,IXC,IXA  ARE DEFINED BY THE PROGRAMS
C         TRI  OR  QUAD.
C
C      EACH OF THE VARIABLES (OTHER THAN THE INPUT VARIABLES) STORED IN
C         COMMON/SPACE/  IS REFERENCED BY PROGRAMS IN MORE THAN ONE
C         OVERLAY OF THE SYMINSE MODULE.  BY CONTRAST, THE VARIABLES
C         STORED IN  COMMON/TEMP/  ARE REFERENCED BY PROGRAMS WITHIN ONE
C         OVERLAY ONLY.
C
C      IT IS IMPORTANT TO REMARK THAT THE VARIABLE  NRECORD
C         MUST NOT BE MODIFIED BY THE CALLING PROGAM BETWEEN CALLS TO
C         ELEMENT.  ALL OTHER WORDS IN  COMMON/SPACE/  AND ALL WORDS IN
C         COMMON/TEMP/  MAY BE FREELY CHANGED BETWEEN CALLS TO  ELEMENT.
C
C      THE VARIABLE  FIRST  IS INITIALLY .TRUE.  BUT IS SET TO .FALSE.
C         ON THE FIRST PASS THROUGH ELEMENT.
C
C*********************************************************************
C
      LOGICAL FIRST,PRFLAG
      COMMON/SPACE/SPACE(29),PRFLAG,OTHERS(1)
      COMMON/TEMP/TEMP(61)
C
      DATA FIRST/.T./
C
C
      TIME = TIMING(DUMMY)
      IF (FIRST)                              GO TO 1
                                              GO TO 3
C     THEN
C         . CALL THE PROGRAM CALLED  SETUP  WHICH READS IN THE REQUIRED
C            ARRAYS FROM TAPE1 AND STORES THEM ON DISK FOR FUTURE RECALL
C            BY THE PROGRAMS  QUAD  AND  TRI.
 1        CALL OVERLAY(4HMAIN,1,0)
          FIRST = .F.
          TIME = TIMING(DUMMY)
          WRITE(6,2) TIME
 2           FORMAT(///////* THE SETUP TIME WAS*,F7.3,* SECONDS.*)
C     CONTINUE
C
C     . CALL THE SUBROUTINE INTGRAL WHICH EVALUATES THE A-, B- AND
C         C-INTEGRALS AND STORES THEM IN POSITIONS  IXC+1  THRU  IXA+IA
C         OF COMMON.
 3    CALL INTGRAL
C
C     . CALL THE PROGRAM  SGPM  WHICH COMPUTES THE STIFFNESS MATRIX  SS,
C         THE GEOMETRIC STIFFNESS MATRIX  SG  IF  SGFLAG=.TRUE.,
C         THE LOADVECTOR  SP  IF  SPFLAG=.TRUE.,
C         AND THE CONSISTENT MASS MATRIX  SM  IF  SMFLAG=.TRUE.
      CALL OVERLAY(4HMAIN,12B,0,6HRECALL)
      TIME = TIMING(DUMMY)
```

```
C
C     . STORE THE RESULTS ON DISK FOR FUTURE ASSEMBLY OF THE SYSTEM
C        EQUATIONS.
      CALL STORE
C
      IF (PRFLAG)
C     THEN
C        . CALL THE PROGRAM  OUTPUT  WHICH DISPLAYS THE RESULTS.
     *   CALL OVERLAY(4HMAIN,13B,0,6HRECALL)
C     CONTINUE
C
      WRITE(6,4) TIME
 4       FORMAT(///////* THE COMPUTATION TIME FOR THIS ELEMENT WAS*
     *       ,F6.3,* SECONDS.*)
      RETURN
      END
```

APPENDIX A

```
      SUBROUTINE INTGRAL
C***********************************************************************
C
C     THIS SUBROUTINE CALLS VARIOUS SUBPROGRAMS TO COMPUTE THE A-, B-
C        AND C-INTEGRALS NEEDED IN THE SUBROUTINES LINSTF,LODVEC AND
C        AND MASS.
C
C     PARA  IS SET TO  .TRUE.  BY THE SUBROUTINE  QUAD  WHEN THE ELEMENT
C        IS DEEMED TO BE A PARALLELOGRAM.
C     TRAP  IS SET TO  .TRUE.  BY THE SUBROUTINE  QUAD  WHEN THE ELEMENT
C        IS DEEMED TO BE A TRAPEZOID.
C     THE VARIABLE  LIMIT  SPECIFIES THE UPPER LIMIT TO THE RANGE OF
C        C-INTEGRALS TO BE EVALUATED WITHIN A GIVEN OVERLAY CALL.
C
C***********************************************************************
C
      LOGICAL PARA,TRAP
      COMMON/SPACE/SPACE(24),NSF,SKIP(75),SKP(7),LIMIT,SKIPP(27),PARA,
     *   TRAP,OTHERS(1)
C
C
C
      IF (NSF.EQ.6 .OR. NSF.EQ.10)                GO TO 2
                                                  GO TO 3
C     THEN  TRIANGULAR CASE
C        . CALL THE PROGRAM  TRI  WHICH EVALUATES THE A-, B- AND C-TYPE
C           INTEGRALS FOR TRIANGULAR ELEMENTS.
 2       CALL OVERLAY(4HMAIN,2,0,6HRECALL)
                                                  GO TO 20
C     ELSE
 3    IF (NSF.EQ.4 .OR. NSF.EQ.5 .OR. NSF.EQ.8 .OR. NSF.EQ.9) GO TO 4
                                                  GO TO 18
C     THEN  QUADRILATERAL CASE
C        . CALL THE PROGRAM  QUAD  WHICH EVALUATES THE A-, B- AND C-TYPE
C           INTEGRALS FOR PARALLELOGRAM ELEMENTS AND THE A- AND B-TYPE
C           INTEGRALS FOR THE NONPARALLELOGRAM QUADRILATERAL CASE.
 4       CALL OVERLAY(4HMAIN,3,0,6HRECALL)
         IF (.NOT.PARA)                           GO TO 5
                                                  GO TO 20
C        THEN THE ELEMENT IS NOT A PARALLELOGRAM.
 5          IF (TRAP)                             GO TO 6
                                                  GO TO 13
C           THEN THE ELEMENT IS A TRAPEZOID.
 6             IF (NSF.EQ.4)                      GO TO 7
                                                  GO TO 8
C              THEN
 7                LIMIT = 10
                  CALL OVERLAY(4HMAIN,4,0)
                                                  GO TO 20
C              ELSE
 8             IF (NSF.EQ.5)                      GO TO 9
                                                  GO TO 10
C              THEN
 9                LIMIT = 15
                  CALL OVERLAY(4HMAIN,4,0)
                                                  GO TO 20
C              ELSE
 10            IF (NSF.EQ.8)                      GO TO 11
                                                  GO TO 12
C              THEN
 11               LIMIT = 36
                  CALL OVERLAY(4HMAIN,5,0)
                                                  GO TO 20
C              ELSE  NSF=9
 12               LIMIT = 45
                  CALL OVERLAY(4HMAIN,5,0)
                                                  GO TO 20
C              CONTINUE
```

```
C           ELSE  NON-TRAPEZOIDAL CASE
 13            IF (NSF.EQ.4 .OR. NSF.EQ.5)        GO TO 14
                                                  GO TO 16
C              THEN
C                 . LIMIT IS 10 IF NSF.EQ.4 BUT LIMIT IS 15 IF NSF.EQ.5
 14               LIMIT = 10
                  IF (NSF.EQ.5)
C                 THEN
     *               LIMIT = 15
C                 CONTINUE
                  CALL OVERLAY(4HMAIN,6,0)
                                                  GO TO 20
C                 CONTINUE
C              ELSE  NSF=8 OR NSF=9
 16               CALL OVERLAY(4HMAIN,7,0)
                  CALL OVERLAY(4HMAIN,10B,0)
                  IF (NSF.EQ.9)                   GO TO 17
                                                  GO TO 20
C                 THEN
 17                  CALL OVERLAY(4HMAIN,11B,0)
                                                  GO TO 20
C                 CONTINUE
C              CONTINUE
C           CONTINUE
C        CONTINUE
C     ELSE  ERROR
 18      WRITE (6,19) NSF
 19         FORMAT(* NSF =*,I5,* IS ILLEGAL.*)
         STOP
 20   CONTINUE
C     . AT THIS POINT ALL INTEGRALS HAVE BEEN COMPUTED.
      RETURN
      END
```

APPENDIX A

```
      SUBROUTINE STORE
C
      LOGICAL SGFLAG,SMFLAG,SPFLAG
      COMMON/TEMP/I,IL,IU
      COMMON/SPACE/SPACE(24),NSF,SKIP,SGFLAG,SMFLAG,SPFLAG,SKP(71),
     *   SKIPP(14),NNE,ISS,ISG,IXC,OTHERS(1)
C
C
C     . STORE  SS  ON UNIT 3.
      IL = ISS + 1
      WRITE (3) (SPACE(I),I=IL,ISG)
C
      IL = ISG + 1
      IF (SGFLAG)
C     THEN
C     . STORE  SG  ON UNIT 4.
     *   WRITE (4) (SPACE(I),I=IL,IXC)
C     CONTINUE
C
      IL = IXC + 1
      IU = IXC + 5*NNE
      IF (SPFLAG)
C     THEN
C     . STORE  SP  ON UNIT 8.
     *   WRITE (8) (SPACE(I),I=IL,IU)
C     CONTINUE
C
      IL = IU + 1
      IU = IL + NSF*NSF
      IF (SMFLAG)
C     THEN
C     . STORE  SM  ON UNIT 9.
     *   WRITE (9) (SPACE(I),I=IL,IU)
C     CONTINUE
      RETURN
      END
```

APPENDIX A

```
      OVERLAY(MAIN,1,0)
      PROGRAM SETUP
C***********************************************************************
C
C     THE PURPOSE OF THIS SUBROUTINE IS TO SET UP THE PARAMETERS AND
C        ARRAYS REQUIRED BY THE SUBROUTINES WHICH EVALUATE AND STORE
C        INTEGRALS.
C     ALL READ STATEMENTS IN THIS PROGRAM REFER TO  TAPE1.
C     THE WRITE STATEMENTS IN THIS PROGRAM REFER TO THE OUTPUT FILE OR
C        TO SCRATCH DISC (UNIT 2).
C
C     THE LENGTH OF COMMON MUST BE THE MAXIMUM OF THE VALUES OF *LENGTH*
C        FOR THE VALUES OF NSF EMPLOYED.
C     NNE  IS THE NUMBER OF TRUE NODES PER ELEMENT.
C     IA  IS THE NUMBER OF DISTINCT A-INTEGRALS.
C        IA = (NSF+1)*(NSF+2)*(NSF+3)*(NSF+4)/24
C     2*IB  IS THE NUMBER OF DISTINCT B-INTEGRALS.
C        IB = NSF*(NSF+1)*(NSF+2)/2
C     4*IC  IS THE NUMBER OF COMPUTED C-INTEGRALS.
C        IC = NSF*(NSF+1)/2
C     JA  IS THE NUMBER OF REPRESENTATIVE A-INTEGRALS.
C     JB  IS THE NUMBER OF REPRESENTATIVE B-INTEGRALS.
C     JC  IS THE NUMBER OF REPRESENTATIVE C-INTEGRALS.
C     THE ARRAY  SS  IS STORED BEGINNING IN ISS+1.
C     THE ARRAY  SG  IS STORED BEGINNING IN ISG+1.
C     THE C-INTEGRALS AND LATER  SP  ARE STORED BEGINNING IN IXC+1.
C     THE ARRAY  SM  IS STORED BEGINNING IN IXC+5*NNE+1.
C     THE B-INTEGRALS ARE STORED BEGINNING IN IXB+1.
C     THE A-INTEGRALS ARE STORED BEGINNING IN  IXA+1.
C
C***********************************************************************
C
      LOGICAL TRI
      DIMENSION INDICES(8,7),NCARDS(7),INDX(12)
      COMMON/SPACE/SPACE(100),NRECORD(7),SKIP,IA,IB,IC,JA,JB,JC,NNE,ISS,
     *   ISG,IXC,IXB,IXA,QC(1)
      COMMON/TEMP/  IL,IU,   I2,I3,I4,  JL,JU,LENGTH,LEVEN,LODD,  N,
     *   NSFM,TRI
      COMMON/STORE/IX(31)
      EQUIVALENCE (IA,INDX(1))
C
      DATA (NCARDS(I),I=1,7)/42,82,119,1,204,305,427/
      DATA (INDICES(I),I=1,56)/
     *   70,   60,  10,   17,   11,   3,   4,  137,
     *  126,  105,  15,   34,   24,   5,   4,  137,
     *  210,  168,  21,   51,   36,   6,   6,  120,
     *    0,    0,   0,    0,    0,   0,   0,    0,
     *  495,  360,  36,   84,   54,   8,   8,  137,
     *  715,  495,  45,  130,   83,  11,   8,  137,
     * 1001,  660,  55,  195,  121,  14,  10,  120/
C
C
      NRECORD(4) = 0
      CALL OPENMS(2,IX(1),31,0)
      N = 0
      DO 8 M=1,7
         NSFM = M + 3
         IF (NRECORD(M).EQ.0)                    GO TO 1
                                                 GO TO 4
C        THEN  THE INTEGRATION ARRAYS FOR NSF EQUAL TO NSFM ARE NOT TO
C              BE SET UP.  NCARDS(M) IS THE NUMBER OF INPUT CARDS TO BE
C              SKIPPED OVER.
 1          IU = NCARDS(M)
            DO 3 I=1,IU
               READ(1,2)
 2                FORMAT(I1)
 3          CONTINUE
                                                 GO TO 7
```

```
C        ELSE  THE INTEGRATION ARRAYS FOR THIS  VALUE OF NSFM ARE TO BE
C              SET UP.
 4          N = N + 1
C           . THE (M)TH POSITION OF NRECORD IS TO BE REPLACED BY AN
C             INTEGER WHICH INDICATES WHERE ON UNIT (2) THE INTEGRATION
C             ARRAYS FOR  NSF=M+3  ARE TO BE STORED.  SEE SUBROUTINES
C             TRI AND QUAD.
            NRECORD(M) = N
            TRI = NSFM.EQ.6 .OR. NSFM.EQ.10
            DO 5 I=1,8
               INDX(I) = INDICES(I,M)
 5          CONTINUE
            ISG = ISS + 25*NSFM*NSFM
            IXC = ISG + NSFM*NSFM
            IXB = IXC + 4*IC
            IXA = IXB + 2*IB
            LENGTH = IXA + IA
            CALL WRITMS(2,INDX(1),12,5*N-4)
            WRITE(6,6) NSFM,LENGTH
 6             FORMAT(*1LENGTH OF COMMON/SPACE/ REQUIRED FOR NSF = *,I5,
     *            * IS *,I6,*.*/)
C
C           . SET UP THE INTEGRATION ARRAYS FOR THE A-INTEGRALS .
            CALL SETA
C
C           . SET UP THE INTEGRATION ARRAYS FOR THE B-INTEGRALS .
            CALL SETB
C
C           . SET UP THE INTEGRATION ARRAYS FOR THE C-INTEGRALS .
            CALL SETC
C
 7        CONTINUE
 8     CONTINUE
       WRITE(6,9)
 9        FORMAT(1H1)
       END
```

APPENDIX A

```
      SUBROUTINE SETA
C***********************************************************************
C
C     THIS SUBROUTINE SETS UP THE INTEGRATION ARRAYS FOR EVALUATING
C        A-INTEGRALS.
C
C     THE NUMBERS  KA  SELECT  THE REPRESENTATIVE INTEGRALS.
C     THE NUMBERS  LA  SELECT THE GROUP TRANSFORMATIONS.
C
C***********************************************************************
C
      LOGICAL TRI
      DIMENSION KA(1),LA(1),QA(4,1),QA1(1),QA2(1),QA3(1)
      COMMON/SPACE/SPACE(108),IA,SKIP(2),JA,SKP(7),IXA,QQ(1)
      COMMON/TEMP/  IL,IU,    I2,I3,I4,   JL,JU,LENGTH,LEVEN,LODD,  N,
     *   NSFM,TRI,K,L,Q1,Q2
      EQUIVALENCE (KA(1),SPACE(1)),(LA(1),SPACE(1)),(QA(1,1),QQ(1)),
     *   (QA1(1),SPACE(1)),(QA2(1),SPACE(1)),(QA3(1),SPACE(1))
C
C
      IL = IXA + 1
      IU = IXA + IA
      READ(1,1) (KA(I),I=IL,IU)
 1       FORMAT(20I4)
      WRITE(6,1) (KA(I),I=IL,IU)
      IF (TRI)                                 GO TO 2
                                               GO TO 7
C     THEN  TRIANGULAR CASE
 2       READ(1,3) ((QA(I,J),I=1,2),J=1,JA)
 3          FORMAT(10X,2F10.0)
         DO 4 J=1,JA
            QA(1,J) = QA(1,J)/QA(2,J)
 4       CONTINUE
         WRITE(6,5) (QA(1,J),J=1,JA)
 5          FORMAT(5X,12E10.3)
         DO 6 I=IL,IU
            K = KA(I)
            QA1(I) = QA(1,K)
 6       CONTINUE
         CALL WRITMS(2,QA1(IL),IU-IL+1,5*N-3)
                                               GO TO 12
C     ELSE  QUADRILATERAL CASE
 7       JL = IL - IA
         JU = IU - IA
         READ(1,8) (LA(I),I=JL,JU)
 8          FORMAT(80I1)
         WRITE(6,1) (LA(I),I=JL,JU)
         READ(1,9) ((QA(I,J),I=1,4),J=1,JA)
 9          FORMAT(10X,4F10.0)
         DO 10 I=1,3
            DO 10 J=1,JA
               QA(I,J) = QA(I,J)/QA(4,J)
C           CONTINUE
 10      CONTINUE
         WRITE(6,5) ((QA(I,J),J=1,JA),I=1,3)
         DO 11 I1=IL,IU
            I2 = I1 - IA
            K = KA(I1)
            L = LA(I2)
            LODD = L - (L/2)*2
            LEVEN = 1 - LODD
            Q1 = QA(1,K)
            Q2 = QA(2,K)
            QA1(I1) = QA(3,K)
            QA2(I2) = -(-1)**(L/2)*(LODD*Q2+LEVEN*Q1)
            QA3(I2-IA) = -(-1)**((L+1)/4)*(LODD*Q1+LEVEN*Q2)
 11      CONTINUE
```

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

APPENDIX A

```
      SUBROUTINE SETB
C***********************************************************************
C
C     THIS SUBROUTINE SETS UP THE INTEGPATION ARRAYS FOR EVALUATING
C        B-INTEGRALS.
C
C     THE NUMBERS  KB  SELECT  THE REPRESENTATIVE INTEGRALS.
C     THE NUMBERS  LB  SELECT THE GROUP TRANSFORMATIONS.
C
C***********************************************************************
C
      LOGICAL TRI
      DIMENSION KB(1),LB(1),QB(4,1),QB1(1),QB2(1),QB3(1)
      COMMON/SPACE/SPACE(109),IB,SKIP(2),JB,SKP(5),IXB,SKIPP,QQ(1)
      COMMON/TEMP/  IL,IU,    I2,I3,I4,   JL,JU,LENGTH,LEVEN,LODD,   N,
     *   NSFM,TRI,K,L,Q2,Q3
      EQUIVALENCE (KB(1),SPACE(1)),(LB(1),SPACE(1)),(QB(1,1),QQ(1)),
     *   (QB1(1),SPACE(1)),(QB2(1),SPACE(1)),(QB3(1),SPACE(1))
C
C
      IL = IXB + 1
      IU = IXB + IB
      READ(1,1) (KB(I),I=IL,IU)
 1       FORMAT(20I4)
      JL = IL + IB
      JU = IU + IB
      READ(1,2) (LB(I),I=JL,JU)
 2       FORMAT(80I1)
      WRITE(6,1) (KB(I),I=IL,IU),(LB(I),I=JL,JU)
      IF (TRI)                                    GO TO 3
                                                  GO TO 13
C     THEN   TRIANGULAR CASE
 3       READ(1,4) ((QB(I,J),I=1,3),J=1,JB)
 4          FORMAT(10X,3F10.0)
         DO 5 I=1,2
            DO 5 J=1,JB
               QB(I,J) = QB(I,J)/QB(3,J)
C           CONTINUE
 5       CONTINUE
         DO 12 I1=JL,JU
            I2 = I1 - IB
            K = KB(I2)
            L = LB(I1)
            GO TO (6,7,8,9,10,11),L
 6             QB1(I1) =  QB(1,K)
               QB2(I2) =  QB(2,K)
                  GO TO 12
 7             QB1(I1) =  QB(2,K)
               QB2(I2) = -QB(1,K) - QB(2,K)
                  GO TO 12
 8             QB1(I1) = -QB(1,K) - QB(2,K)
               QB2(I2) =  QB(1,K)
                  GO TO 12
 9             QB1(I1) = -QB(2,K)
               QB2(I2) = -QB(1,K)
                  GO TO 12
 10            QB1(I1) =  QB(1,K) + QB(2,K)
               QB2(I2) = -QB(2,K)
                  GO TO 12
 11            QB1(I1) = -QB(1,K)
               QB2(I2) =  QB(1,K) + QB(2,K)
C           . END COMPUTED GO TO .
 12      CONTINUE
C        . STORE  QB2  AND  QB1 .
         CALL WRITMS(2,SPACE(IL),IU-IL+1,5*N-2)
                                                  GO TO 18
C     ELSE  QUADRILATERAL CASE
 13      READ(1,14) ((QB(I,J),I=1,4),J=1,JB)
 14         FORMAT(10X,4F10.0)
```

```
          DO 15 I=1,3
             DO 15 J=1,JB
                QB(I,J) = QB(I,J)/QB(4,J)
C            CONTINUE
 15       CONTINUE
          WRITE(6,16) ((QB(I,J),J=1,JB),I=1,3)
 16          FORMAT(5X,12E10.3)
          DO 17 I1=JL,JU
             I2 = I1 - IB
             K = KB(I2)
             L = LB(I1)
             LODD = L - (L/2)*2
             LEVEN = 1 - LODD
             QB1(I1) = (LODD-LEVEN)*QB(1,K)
             Q2 = QB(2,K)
             Q3 = QB(3,K)
             QB2(I2) = (-1)**(L/2)*(LODD*Q2-LEVEN*Q3)
             QB3(I2-IB) = (-1)**((L+1)/4)*(LODD*Q3-LEVEN*Q2)
 17       CONTINUE
          IL = IL - IB
C         . STORE  QB3, QB2  AND  QB1 .
          CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N-2)
 18    CONTINUE
       RETURN
       END
```

```
      SUBROUTINE SETC
C***********************************************************************
C
C     THIS SUBROUTINE SETS UP THE INTEGRATION ARRAYS FOR EVALUATING
C        C-INTEGRALS.
C
C     THE NUMBERS  KC  SELECT  THE REPRESENTATIVE INTEGRALS.
C     THE NUMBERS  LC  SELECT THE GROUP TRANSFORMATIONS.
C
C***********************************************************************
C
      LOGICAL TRI
      DIMENSION KC(1),LC(1),QC(5,1),QC1(1),QC2(1),QC3(1),QC4(1)
      COMMON/SPACE/SPACE(110),IC,SKIP(2),JC,SKP(3),IXC,SKIPP(2),QQ(1)
      COMMON/TEMP/  IL,IU,    I2,I3,I4,   JL,JU,LENGTH,LEVEN,LODD,   N,
     *   NSFM,TRI,K,L,Q1,Q2,Q3,Q4
      EQUIVALENCE (KC(1),SPACE(1)),(LC(1),SPACE(1)),(QC(1,1),QQ(1)),
     *   (QC1(1),SPACE(1)),(QC2(1),SPACE(1)),(QC3(1),SPACE(1)),
     *   (QC4(1),SPACE(1))
C
C
      IL = IXC + 1
      IU = IXC + IC
      READ(1,1) (KC(I),I=IL,IU)
 1       FORMAT(20I4)
      JL = IL + IC
      JU = IU + IC
      READ(1,2) (LC(I),I=JL,JU)
 2       FORMAT(80I1)
      WRITE(6,1) (KC(I),I=IL,IU),(LC(I),I=JL,JU)
      READ(1,3) ((QC(I,J),I=1,5),J=1,JC)
 3       FORMAT(10X,5F10.0)
      DO 4 I=1,4
         DO 4 J=1,JC
            QC(I,J) = QC(I,J)/QC(5,J)
C        CONTINUE
 4    CONTINUE
      WRITE(6,5) ((QC(I,J),J=1,JC),I=1,4)
 5       FORMAT(5X,12E10.3)
      IF (TRI)                                   GO TO 6
                                                 GO TO 15
C     THEN  TRIANGULAR CASE
 6       JL = JL + 2*IC
         JU = JU + 2*IC
         DO 14 I1=JL,JU
            I2 = I1 - IC
            I3 = I2 - IC
            I4 = I3 - IC
            K  = KC(I4)
            L  = LC(I3)
            Q1 = QC(1,K)
            Q2 = QC(2,K)
            Q3 = QC(3,K)
            Q4 = QC(4,K)
            GO TO (7,8,9,10,11,12),L
 7             QC1(I1) =  Q1
               QC2(I2) =  Q2
               QC3(I3) =  Q3
               QC4(I4) =  Q4
                  GO TO 13
 8             QC1(I1) =  Q4
               QC2(I2) = -(Q3+Q4)
               QC3(I3) = -(Q2+Q4)
               QC4(I4) =  Q1+Q2+Q3+Q4
                  GO TO 13
 9             QC1(I1) =  Q1+Q2+Q3+Q4
               QC2(I2) = -(Q1+Q3)
               QC3(I3) = -(Q1+Q2)
               QC4(I4) =  Q1
                  GO TO 13
 10           QC1(I1) =  Q4
              QC2(I2) =  Q3
              QC3(I3) =  Q2
              QC4(I4) =  Q1
                GO TO 13
 11           QC1(I1) =  Q1+Q2+Q3+Q4
              QC2(I2) = -(Q2+Q4)
              QC3(I3) = -(Q3+Q4)
              QC4(I4) =  Q4
                GO TO 13
 12           QC1(I1) =  Q1
              QC2(I2) = -(Q1+Q2)
              QC3(I3) = -(Q1+Q3)
              QC4(I4) =  Q1+Q2+Q3+Q4
C           . END COMPUTED GO TO .
 13         CONTINUE
 14       CONTINUE
C         . STORE QC4, QC3, QC2 AND QC1 .
          CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N-1)
                                                     GO TO 22
C       THEN  QUADRILATERAL CASE
C         . STORE  KC   AND   LC .
 15       CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N-1)
          JL = JL + 2*IC
          JU = JU + 2*IC
          DO 21 I1=JL,JU
            I2 = I1 - IC
            I3 = I2 - IC
            I4 = I3 - IC
            K   = KC(I4)
            L   = LC(I3)
            GO TO (16,17,16,17,18,19,18,19),L
 16           QC1(I1) =  QC(1,K)
              QC2(I2) =  QC(2,K)
              QC3(I3) =  QC(3,K)
              QC4(I4) =  QC(4,K)
                GO TO 20
 17           QC1(I1) =  QC(4,K)
              QC2(I2) = -QC(3,K)
              QC3(I3) = -QC(2,K)
              QC4(I4) =  QC(1,K)
                GO TO 20
 18           QC1(I1) =  QC(1,K)
              QC2(I2) = -QC(2,K)
              QC3(I3) = -QC(3,K)
              QC4(I4) =  QC(4,K)
                GO TO 20
 19           QC1(I1) =  QC(4,K)
              QC2(I2) =  QC(3,K)
              QC3(I3) =  QC(2,K)
              QC4(I4) =  QC(1,K)
C           . END COMPUTED GO TO .
 20         CONTINUE
 21       CONTINUE
C         . STORE QC4, QC3, QC2 AND QC1 .
          CALL WRITMS(2,SPACE(IL),JU-IL+1,5*N)
 22     CONTINUE
        RETURN
        END
```

```
      OVERLAY(MAIN,2,0)
      PROGRAM TRI
C*********************************************************************
C
C     THIS SUBROUTINE NUMERICALLY EVALUATES FOR TRIANGULAR FINITE
C        ELEMENTS ALL THREE TYPES OF INTEGRALS AND STORES THEM IN COMMON
C        WITH ALIASES XA,XB,XC.
C
C     THE NODES OF THE SIX- AND TEN-NODE TRIANGULAR FINITE ELEMENTS
C        ARE NUMBERED AS FOLLOWS.
C
C                                                 3
C                   3
C                                             6       8
C               6       5
C                                         9       10      5
C           1       4       2
C                                     1       4       7       2
C
C     EACH TIME THIS PROGRAM IS CALLED IT READS SEVEN RECORDS FROM
C        SCRATCH DISC (UNIT 2).  THESE RECORDS WERE WRITTEN BY THE
C        PROGRAM SETUP.
C
C*********************************************************************
C
C
      DIMENSION XA(1),XB(1),XC(1),QA1(1),QB1(1),QB2(1),
     *   QC1(1),QC2(1),QC3(1),QC4(1),INDX(12)
      COMMON/SPACE/SPACE(24),NSF,SKIP(7),X1,X2,X3,X4,Y1,Y2,Y3,Y4,ZZ(60),
     *   NRECORD(7),SKIPP,IA,IB,IC,JA,JB,JC,NNE,ISS,ISG,IXC,IXB,IXA,
     *   OTHERS(1)
      COMMON/TEMP/AREA,AREAINV,CA,CB,CC,CD,C1,C2,IL,IU,I2,I3,I4,
     *   K,L,NREC,TEMP,X12,X23,X31,Y12,Y23,Y31
      EQUIVALENCE (XA(1),SPACE(1)),(XB(1),SPACE(1)),(XC(1),SPACE(1)),
     *   (QA1(1),SPACE(1)),(QB1(1),SPACE(1)),(QB2(1),SPACE(1)),
     *   (QC1(1),SPACE(1)),(QC2(1),SPACE(1)),(QC3(1),SPACE(1)),
     *   (QC4(1),SPACE(1)),(INDX(1),IA)
C
C
      AREA = ((X1-X2)*(Y1-Y3) - (X1-X3)*(Y1-Y2))/2.
      AREAINV = 1./AREA
      X12 = X1 - X2
      X23 = X2 - X3
      X31 = X3 - X1
      Y12 = Y1 - Y2
      Y23 = Y2 - Y3
      Y31 = Y3 - Y1
      NREC = 5*NRECORD(NSF-3) - 4
      IF (NREC.LT.0)
C   ...THEN ERROR TERMINATION
     *   STOP 234
C     CONTINUE
      CALL READMS(2,INDX(1),12,NREC)
      IL = IXA + 1
      IU = IXA + IA
      CALL READMS(2,QA1(IL),IU-IL+1,NREC+1)
      DO 2 I=IL,IU
         XA(I) = QA1(I)*AREA
 2    CONTINUE
C     . EVALUATION OF XA IS NOW COMPLETE.
C
C
      IL = IXB + 1
      IU = IXB + 2*IB
C     . READ IN  QB2  AND  QB1 .
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+2)
      IU = IXB + IB
      DO 3 I1=IL,IU
         I2 = I1 + IB
         CA = QB1(I2)
         CB = QB2(I1)
         XB(I1) = CB*Y23 - CA*Y31
         XB(I2) = CA*X31 - CB*X23
 3    CONTINUE
C     . EVALUATION OF XB IS NOW COMPLETE.
C
      IL = IXC + 1
      IU = IXC + 4*IC
C     . READ IN  QC4, QC3, QC2  AND  QC1 .
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+3)
      IU = IXC + IC
      DO 4 I1=IL,IU
         I2 = I1 + IC
         I3 = I2 + IC
         I4 = I3 + IC
         CA = QC1(I4)
         CB = QC2(I3)
         CC = QC3(I2)
         CD = QC4(I1)
         C1 = CB*Y23 - CA*Y31
         C2 = CD*Y23 - CC*Y31
         XC(I1) = (C2*Y23 - C1*Y31)*AREAINV
         XC(I2) = (C1*X31 - C2*X23)*AREAINV
         C1 = CB*X23 - CA*X31
         C2 = CD*X23 - CC*X31
         XC(I3) = (C1*Y31 - C2*Y23)*AREAINV
         XC(I4) = (C2*X23 - C1*X31)*AREAINV
 4    CONTINUE
C     . EVALUATION OF XC IS NOW COMPLETE.
      END
      OVERLAY(MAIN,3,0)
      PROGRAM QUAD
C**********************************************************************
C
C     FOR PARALLELOGRAM FINITE ELEMENTS THIS PROGRAM NUMERICALLY EVALU-
C        ATES ALL 3 TYPES OF INTEGRALS AND STORES THEM IN COMMON/SPACE/
C        WITH ALIASES  XA,XB,XC.  IT THEN SETS  PARA  TO  .TRUE.
C     FOR QUADRILATERAL ELEMENTS WHICH ARE NOT PARALLELOGRAMS ONLY  XA
C        AND  XB  ARE COMPUTED,  PARA  IS SET TO .FALSE.,  TRAP  IS SET
C        TO .TRUE. OR .FALSE. ACCORDING AS THE ELEMENT IS OR IS NOT A
C        TRAPEZOID, AND LOGARITHM TERMS ARE EVALUATED.
C
C     THE NODES FOR THE FOUR-,FIVE-, EIGHT-, AND NINE-NODE QUADRILATERAL
C        FINITE ELEMENTS ARE LABELED AS FOLLOWS
C
C                   4       3         4   7   3
C
C                       5             8   9   6
C
C                   1       2         1   5   2
C
C     THE SHAPE FUNCTION ASSOCIATED WITH THE BUBBLE MODE IS
C
C                             2         2
C        N(KSE,ETA) = (1-KSI )*(1-ETA )
C
C     EACH TIME THIS PROGRAM IS CALLED IT READS RECORDS FROM SCRATCH
C        DISC (UNIT 2).  THESE RECORDS WERE WRITTEN BY THE PROGRAM
C        CALLED SETUP.
C
C**********************************************************************
C
      LOGICAL PARA,TRAP,RRR,SSS
      DIMENSION XA(1),XB(1),XC(1),KC(1),LC(1),
     *   QA1(1),QA2(1),QA3(1),QB1(1),QB2(1),QB3(1),
     *   QC1(1),QC2(1),QC3(1),QC4(1),INDX(12)
      COMMON/SPACE/SPACE(24),NSF,DUM(7),X(4),Y(4),DUMMY(60),
     *   NRECORD(7),SKIP,IA,IB,IC,JA,JB,JC,NNE,ISS,ISG,IXC,IXB,IXA,
     *   X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,ALG1,ALG2,ALG3,ULG1,ULG2,ULG3,
     *   PARA,TRAP,OTHERS(1)
      COMMON/TEMP/CA,CB,CC,CD,C1,C2,IL,IQA,IU,I2,I3,I4,K,L,
     *   NREC,TEMP,Z1INV,R,S,RR,SS,RRR,SSS,RD,SD,R2,S2,RR2,SS2
      EQUIVALENCE (XA(1),SPACE(1)),(XB(1),SPACE(1)),(XC(1),SPACE(1)),
     *   (KC(1),SPACE(1)),(LC(1),SPACE(1)),
     *   (QA1(1),SPACE(1)),(QA2(1),SPACE(1)),(QA3(1),SPACE(1)),
     *   (QB1(1),SPACE(1)),(QB2(1),SPACE(1)),(QB3(1),SPACE(1)),
     *   (QC1(1),SPACE(1)),(QC2(1),SPACE(1)),(QC3(1),SPACE(1)),
     *   (QC4(1),SPACE(1)),(INDX(1),IA),(DLG,ALG1),(RS2,ALG2),(CLG,ALG3)
     *   ,(VLG1,ALG1),(VLG2,ALG2),(VLG3,ALG3)
C
      DATA EPS/1.E-4/
C
C
      X1 = (X(1)+X(3)-X(2)-X(4))/4.
      X2 = (X(2)+X(3)-X(4)-X(1))/4.
      X3 = (X(3)+X(4)-X(1)-X(2))/4.
      Y1 = (Y(1)+Y(3)-Y(2)-Y(4))/4.
      Y2 = (Y(2)+Y(3)-Y(4)-Y(1))/4.
      Y3 = (Y(3)+Y(4)-Y(1)-Y(2))/4.
C     . Z1  IS ONE-FOURTH THE AREA OF THE QUADRILATERAL ELEMENT.
      Z1 = X2*Y3 - X3*Y2
      Z2 = X3*Y1 - X1*Y3
      Z3 = X1*Y2 - X2*Y1
      Z1INV = 1./Z1
      R = Z2/Z1
      S = Z3/Z1
      R2 = R*R
      S2 = S*S
      RD = 1./(1.-R2)
      SD = 1./(1.-S2)
      RR = R*SD
      SS = S*RD
      RRR = ABS(RR).LT.EPS
      SSS = ABS(SS).LT.EPS
      PARA = RRR.AND.SSS
      TRAP = RRR.OR.SSS
      NREC = 5*NRECORD(NSF-3) - 4
      IF (NREC.LT.0)
C  ...THEN ERROR TERMINATION
     *   STOP 123
C     CONTINUE
      CALL READMS(2,INDX(1),12,NREC)
      IL = IXA + 1 - 2*IA
      IU = IXA + IA
C     . READ IN  QA3, QA2  AND  QA1 .
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+1)
      IL = IXA + 1
      IF (PARA)                                    GO TO 1
                                                   GO TO 3
C     THEN  PARALLELOGRAM CASE
 1       DO 2 I=IL,IU
            XA(I) = QA1(I)*Z1
 2       CONTINUE
                                                   GO TO 5
C     ELSE  NONPARALLELOGRAM CASE
 3       DO 4 I1=IL,IU
            I2 = I1 - IA
            XA(I1) = QA1(I1)*Z1 + QA2(I2)*Z2 + QA3(I2-IA)*Z3
 4       CONTINUE
 5    CONTINUE
C     . EVALUATION OF XA IS NOW COMPLETE.
C     *******************************************
C
      IL = IXB + 1 - IB
      IU = IXB + 2*IB
C     . READ IN  QB3, QB2  AND  QB1 .
      CALL READMS(2,SPACE(IL),IU-IL+1,NREC+2)
      IL = IXB + 1
      IU = IXB + IB
      IF (PARA)                                    GO TO 6
                                                   GO TO 8
C     THEN  PARALLELOGRAM CASE
 6       DO 7 I1=IL,IU
            CB = QB2(I1)
            CC = QB3(I1-IB)
            XB(I1) =     CB*Y2 + CC*Y3
            XB(I1+IB) = -CB*X2 - CC*X3
 7       CONTINUE
                                                   GO TO 10
C     ELSE  NONPARALLELOGRAM CASE
 8       DO 9 I1=IL,IU
            I2 = I1 + IB
            CA = QB1(I2)
            CB = QB2(I1)
            CC = QB3(I1-IB)
            XB(I1) =  CA*Y1 + CB*Y2 + CC*Y3
            XB(I2) = -CA*X1 - CB*X2 - CC*X3
 9       CONTINUE
 10   CONTINUE
C     . EVALUATION OF XB IS NOW COMPLETE.
C     *******************************************
C
      IF (PARA)                                    GO TO 11
                                                   GO TO 13
C     THEN  PARALLELOGRAM CASE
 11      IL = IXC + 1
         IU = IXC + 4*IC
```

```
C       . READ IN  QC4, QC3, QC2  AND  QC1 .
        CALL READMS(2,SPACE(IL),IU-IL+1,NREC+4)
        IU = IXC + IC
        DO 12 I1=IL,IU
           I2 = I1 + IC
           I3 = I2 + IC
           I4 = I3 + IC
           CA = QC1(I4)
           CB = QC2(I3)
           CC = QC3(I2)
           CD = QC4(I1)
           C1 = CB*Y3 + CD*Y2
           C2 = CA*Y3 + CC*Y2
           XC(I1) = -(C1*Y2+C2*Y3)*Z1INV
           XC(I2) =  (C1*X2+C2*X3)*Z1INV
           C1 = CB*X3 + CD*X2
           C2 = CA*X3 + CC*X2
           XC(I3) =  (C1*Y2+C2*Y3)*Z1INV
           XC(I4) = -(C1*X2+C2*X3)*Z1INV
 12     CONTINUE
C       . EVALUATION OF XC IS NOW COMPLETE.
                                                 GO TO 22
C    ********************************************
C    ELSE  NONPARALLELOGRAM CASE
 13     IL = IXC + 1
        IU = IXC + 2*IC
C       . READ IN  KC  AND  LC .
        CALL READMS(2,SPACE(IL),IU-IL+1,NREC+3)
        IF (SSS)                                 GO TO 14
                                                 GO TO 15
C       THEN  FIRST TRAPEZOIDAL CASE
 14        DLG = ELOG(R)
           RS2 = R*R
           CLG = 2./3. + RS2*DLG
                                                 GO TO 21
C       ELSE
 15     IF (RRR)                                 GO TO 16
                                                 GO TO 17
C       THEN  SECOND TRAPEZOIDAL CASE
 16        DLG = ELOG(S)
           RS2 = S*S
           CLG = 2./3. + RS2*DLG
                                                 GO TO 21
C       ELSE  TRAPEZIUM CASE
 17        IF (NNE.EQ.4)                         GO TO 18
                                                 GO TO 19
C          THEN
 18           RR2 = RR*RR
              SS2 = SS*SS
              VLG1 = WLOG1(R,S)
              VLG2 = WLOG2(R,S)
              VLG3 = WLOG2(S,R)
              ULG1 = -14./3. + 2.*(R2+S2) + RR2*SS2*VLG1
              ULG2 =   2./3. + 2.*R2 + SS2*VLG2
              ULG3 =   2./3. + 2.*S2 + RR2*VLG3
C             . ALG1 = RR*SS*(-2.+RR2*SS2*ULG1)
C             . ALG2 = SS*(2. + SS2*ULG2)
C             . ALG3 = RR*(2. + RR2*ULG3)
              VLG1 = VLG1*(RD*SD)**5
              VLG2 = VLG2*RD**5
              VLG3 = VLG3*SD**5
              ULG1 = ULG1*(RD*SD)**3
              ULG2 = ULG2*RD**3
              ULG3 = ULG3*SD**3
                                                 GO TO 20
C          ELSE  NNE EQUALS 8
 19           ALG1 = BLOG(Z1,Z2,Z3)
              ALG2 = BLOG(Z2,Z3,Z1)
              ALG3 = BLOG(Z3,Z1,Z2)
 20        CONTINUE
 21        CONTINUE
C          . EVALUATION OF  XC  IS DEFERRED TO ANOTHER OVERLAY.
 22    CONTINUE
C   ...ONLY EXIT.
       END




      FUNCTION BLOG(C,A,B)
C*************************************************************************
C
C                                                      1+X
C     THIS SUBROUTINE COMPUTES X AND THEN  BLOG = LOG(---)/2.
C                                                      1-X
C     FOR SMALL X (X .LE. 0.1)  THE FOLLOWING EXPANSION IS USED IN ORDER
C        TO GIVE GREATER ACCURACY
C
C                      3    5    7    9    11     13     15     17
C                     X    X    X    X    X      X      X      X
C        BLOG = X + -- + -- + -- + -- + --- + --- + --- + ---
C                     3    5    7    9    11     13     15     17
C
C     NOTE THAT  BLOG(C,A,B)  IS SYMMETRIC UNDER INTERCHANGE OF A AND B.
C
C*************************************************************************
C
      COMMON/TEMP/X,X2,N,OTHERS(1)
C
C
      X = 2.*A*B/(A*A+B*B-C*C)
      IF (ABS(X) .LT. 0.1)                          GO TO 1
                                                    GO TO 3
C     THEN
 1       N = 8
         X2 = X*X
         BLOG = 1./(2.*N+1.)
         DO 2 I=1,N
 2          BLOG = 1./(2.*(N-I)+1.) + X2*BLOG
C        CONTINUE
         BLOG = X*BLOG
                                                    GO TO 4
C     ELSE
 3       BLOG = ALOG((1.+X)/(1.-X))/2.
 4    CONTINUE
      RETURN
      END
```

APPENDIX A

```
      FUNCTION ELOG(R)
C***********************************************************************
C
C                                         1+R          2  3   5
C     THIS SUBROUTINE COMPUTES  ELOG = (LOG(---) - 2*R - -*R )/R .
C                                         1-R          3
C
C     FOR SMALL R (R .LE. 0.1)  A TAYLOR SERIES EXPANSION IS USED FOR
C        GREATER ACCURACY.
C
C                         2      4      6      8      10      12      14
C                 2    2*R    2*R    2*R    2*R    2*R     2*R     2*R
C        ELOG = - + ---- + ---- + ---- + ---- + ----- + ----- + -----
C               5     7      9      11     13     15      17      19
C                                                                    16
C                                                                 2*R
C                                                               + -----.
C                                                                  21
C
C     THE FUNCTION  BLOG  IS RELATED TO  ELOG  BY
C                                      1+R          2  3            5
C        BLOG(0,1,R) = BLOG(0,R,1) = LOG(---) = 2*R + -*R  + ELOG(R)*R .
C                                      1-R          3
C
C***********************************************************************
C
      COMMON/TEMP/R2,N,Y
C
C
      R2 = R*R
      IF (ABS(R) .LT. 0.25)                        GO TO 1
                                                   GO TO 3
C     THEN
 1       N = 8
         ELOG = 2./(2.*N+5.)
         DO 2 I=1,N
 2          ELOG = 2./(2.*(N-I)+5.) + R2*ELOG
C        CONTINUE
                                                   GO TO 4
C     ELSE
 3       ELOG = (ALOG((1.+R)/(1.-R))-2.*R*(1.+R2/3.))/(R*R2*R2)
 4    CONTINUE
      RETURN
      END
```

APPENDIX A

```
      FUNCTION WLOG1(R1,S1)
C***********************************************************************
C
C     THIS SUBROUTINE EVALUATES
C
C                                    2
C                    1     1 - (S+R)            14       2  2      3
C                    -*LOG(----------) + 2*TT + (-- - 2*(R +S ))*TT )
C                    2               2           3
C                          1 - (S-R)
C     WLOG1(R,S) =  --------------------------------------------------
C                                               5
C                                             TT
C                          R*S
C        WHERE  TT = ------------- .
C                        2      2
C                    (1-S )*(1-R )
C
C     TAYLOR SERIES EXPANSIONS ARE USED WHEN  R  OR  S  IS SMALL.
C
C***********************************************************************
C
      COMMON/TEMP/ALOG1,C0,C1,C2,C3,C4,R,RR,R2,S,SS,SS2,S2,T,TT
C
C
      IF (ABS(S1) .LE. ABS(R1))                       GO TO 1
                                                      GO TO 2
C     THEN
 1       R = R1
         S = S1
                                                      GO TO 3
C     ELSE
 2       R = S1
         S = R1
C     CONTINUE  (R  IS GREATER THAN OR EQUAL TO S IN MAGNITUDE)
 3    R2 = R*R
      SS = S/(1.-R2)
      SS2=SS*SS
      IF (SS2 .LT. 0.01)                              GO TO 4
                                                      GO TO 5
C     THEN
 4       C0 = -46. + R2*(40.-10.*R2)
         C1 = 56. + R2*(-435.+R2*(497.+R2*(-217.+35.*R2)))
         C2 = -126.+R2*(-1341.+R2*(-6628.+R2*(11826.+R2*(-7434.
     *      +R2*(2163.-252.*R2)))))
         C3 = R2*(-30096.+R2*(-156200.+R2*(132751.+R2*(41305.
     *      +R2*(-83226.+R2*(38346.+R2*(-8085.+693.*R2)))))))
         C4 = R2*(-473616.+R2*(-4453592.+R2*(-982033.+R2*(5808786.
     *      +R2*(-3402451+R2*(651508.+R2*(39897.+R2*(-30030.
     *      +3003.*R2)))))))))
         WLOG1 = 2.*(C0*0.2+SS2*(C1/7.+SS2*(C2/63.+SS2*(C3/693.
     *      +SS2*C4/9009.))))
C  ......EXIT
                                                      RETURN
C     ELSE
 5       ALOG1 = 0.5*ALOG((1.-(R+S)**2)/(1.-(R-S)**2))
         S2 = S*S
         RR = R/(1.-S2)
         TT = RR*SS
         WLOG1 = (ALOG1+2.*TT+(14./3.-2.*(R2+S2))*TT**3)/TT**5
C  ......EXIT
                                                      RETURN
      END
```

APPENDIX A

```
      FUNCTION WLOG2(R,S)
C***********************************************************************
C
C     THIS SUBROUTINE EVALUATES
C                                                     2 1    3
C                               2    2            2*(R +-)*R       2 5
C                    1    (1+S)  - R      2*R            3     (1-S )
C     WLOG2(R,S) = (-*LOG(-----------) - ------ - -----------)*-------- .
C                    2          2    2        2          2 3       5
C                         (1-S)  - R     (1-S )      (1-S )       R
C
C     TAYLOR SERIES EXPANSIONS ARE USED WHEN  R  OR  S  IS SMALL.
C
C***********************************************************************
C
      COMMON/TEMP/ALOG2,C0,C1,C2,C3,C4,C5,C6,RR,RR2,R2,SS,SS2,S2,THIRD
C
C
      R2 = R*R
      SS = S/(1.-R2)
      SS2 = SS*SS
      IF (SS2 .LT. 0.01)                        GO TO 1
                                                GO TO 2
C     THEN
 1       C0 = 1.+R2*(10.+5.*R2)
         C1 = 1.+R2*(21.+R2*(35.+7.*R2))
         C2 = 1.+R2*(36.+R2*(126.+R2*(84.+9.*R2)))
         C4 = 1.+R2*(78.+R2*(715.+R2*(1716.+R2*(1287.+R2*(286.+13.*R2)))
     *      ))
         WLOG2 = 2.*(C0*0.2+SS2*(C1/7.+SS2*(C2/9.+SS2*(C3/11.+SS2*C4/13.
     *      ))))
C  ......EXIT
                                                RETURN
C     ELSE
 2    S2 = S*S
      RR = R/(1.-S2)
      RR2 = RR*RR
      IF (RR2 .LT. 0.01)                        GO TO 3
                                                GO TO 4
C     THEN
 3       THIRD = 1./3.
         C1 = 6. - 4.*S2
         C2 = -2. + S2*(28.+S2*(-32.+S2*10.))
         C3 = 4. + S2*(60.+S2*(40.+S2*(-174.+S2*(120.-26.*S2))))*THIRD
         C4 = -2. + S2*(222.+S2*(10.+S2*(-224.+S2*(52.+S2*(88.+S2*
     *      (-56.+S2*10.))))))*THIRD
         C5 = 0.4 + S2*(222.+S2*(856.+S2*(-1198.+S2*(444.+S2*(-30.+S2*
     *      (14.-2.*S2))))))*THIRD
         C6 = S2*(3.-S2)*(12.+S2*(9.+S2*(-6.+S2)))
     *      *(8.+S2*(54.+S2*(-36.+S2*6.)))*THIRD
C        . ELOG(S) = (ALOG((1+S)/(1-S))-2*S-2/3*S**3)/S**5 .
         WLOG2 = ELOG(S)*(1.-R2)**5 + RR2*(C1+RR2*(C2+RR2*(C3+RR2*(C4
     *      +RR2*(C5+RR2*C6)))))
C        . THIS EXPANSION IS ACCURATE FOR  S=0  AS WELL AS FOR  R=0 .
C  ......EXIT
                                                RETURN
C     ELSE  (RR AND SS ARE BOTH GREATER THAN 0.1)
 4       ALOG2 = 0.5*ALOG(((1.+S)*(1.+S)-R2)/((1.-S)*(1.-S)-R2))
         WLOG2 = (ALOG2 - 2.*SS*(1.+(1./3.+R2)*SS2))/SS*SS2*SS2
C  ......EXIT
                                                RETURN
      END
```

APPENDIX A

```
      OVERLAY(MAIN,4,0)
      PROGRAM TRAP5
C*********************************************************************
C
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR TRAPEZOIDAL ELEMENTS
C        WITH  NNE = 4.
C
C     THE  DO  LOOPS ENDING AT STATEMENTS NUMBERED  3 AND 4  CARRY OUT
C        GROUP TRANSFORMATIONS.
C
C*********************************************************************
C
      DIMENSION LC(1),KC(1)
      COMMON/SPACE/XC(107),LIMIT,SKIP(2),IC,SKP(6),IXC,SKIPP(2),
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,DLG,RS2,CLG,OTHERS(1)
      COMMON/TEMP/I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,R,S,
     *   IL,IU,LL,KI,XJ,TEMP
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))
C
C
      X1 = XX1
      X2 = XX2
      X3 = XX3
      Y1 = YY1
      Y2 = YY2
      Y3 = YY3
      Z1 = ZZ1
      R  = ZZ2/ZZ1
      S  = ZZ3/ZZ1
      IL = IXC + 1
      IU = IXC + LIMIT
      LL = 0
      DO 4 I=1,2
         DO 3 J=1,4
            LL = LL + 1
            KI = 0
            IF (ABS(R).LT.ABS(S))
C           THEN  SECOND TYPE OF TRAPEZOID
     *         KI = 5
C           CONTINUE
            DO 2 I1=IL,IU
               I2 = I1 + IC
               I3 = I2 + IC
               I4 = I3 + IC
               K = KC(I1) + KI
               L = LC(I2)
C              IF (L.EQ.LL)
                                                    IF(L.NE.LL) GO TO 1
C              THEN
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)
 1             CONTINUE
 2          CONTINUE
C           . TRANSFORMATION OF TYPE ONE.
            X1 = -X1
            XJ =  X2
            X2 = -X3
            X3 =  XJ
            Y1 = -Y1
            XJ =  Y2
            Y2 = -Y3
            Y3 =  XJ
            XJ =  R
            R  =  S
            S  = -XJ
 3       CONTINUE
```

```
C         . TRANSFORMATION OF TYPE TWO.
          X3 = -X3
          Y3 = -Y3
          S  = -S
 4     CONTINUE
       END
```

APPENDIX A

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)
C**********************************************************************
C
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM TRAP5 TO EVALUATE
C        C-INTEGRALS.
C
C     RS2  IS THE LARGER OF THE TWO QUANTITIES  R*R  AND S*S.
C
C**********************************************************************
C
C
      COMMON/SPACE/SKIP(129),DLOG,RS2,CLOG,OTHERS(1)
      COMMON/TEMP/I2,I3,I4,K,L,XY(6),Z1,R,S,
     *   IL,IU,LL,KI,XJ,TEMP,XY11,XY22,XY33,XY12,XY31,X23,Y12,Y23,
     *   Y31,X1X2,X3X1,Y1Y2,Y2Y3,Y3Y1
C
C
      GO TO (1,2,3,4,5,11,12,13,14,15),K
C        . 1,1 .
 1       X23 = X2 - X3
         Y23 = Y2 - Y3
         X3X1 = X3 + X1
         Y3Y1 = Y3 + Y1
         TEMP = (X1+X2)*(Y1+Y2) + 3.*X23*Y23
         XDNDN = -(2.*TEMP + CLOG*(3.*X3X1*Y3Y1+3.*R*(X3X1*Y23+X23*Y3Y1)
     *      +RS2*TEMP))/(24.*Z1)
            RETURN
C        . 2,1 .
 2       X23 = X2 - X3
         Y31 = Y3 - Y1
         Y2Y3 = Y2 + Y3
         X3X1 = X3 + X1
         TEMP = (X1+X2)*(Y1+Y2) - 3.*X23*Y2Y3
         XDNDN = (2.*TEMP + CLOG*(3.*X3X1*Y31+3.*R*(X23*Y31-X3X1*Y2Y3)
     *      +RS2*TEMP))/(24.*Z1)
         RETURN
C        . 3,1 .
 3       X23 = X2 - X3
         Y23 = Y2 - Y3
         Y31 = Y3 - Y1
         X3X1 = X3 + X1
         TEMP = (X1+X2)*(Y1-Y2) + 3.*X23*Y23
         XDNDN = (2.*TEMP + CLOG*(-3.*X3X1*Y31+3.*R*(X3X1*Y23-X23*Y31)
     *      +RS2*TEMP))/(24.*Z1)
         RETURN
C        . 5,1 .
 4       X23 = X2 - X3
         X1X2 = X1 + X2
         X3X1 = X3 + X1
         TEMP1 = -X23*Y1 + X1X2*Y3
         TEMP = 2.*(X3X1*Y2+X1X2*Y3) - R*(X3X1*Y1-2.*X23*Y2) + RS2*TEMP1
         XDNDN = -(2.*TEMP + 3.*DLOG*(3.*R*X3X1*Y1+RS2*(TEMP-3.*TEMP1)))
     *      /(9.*Z1)
         RETURN
C        . 5,5 .
 5       XY12 = X1*Y2 + X2*Y1
         XY11 = X1*Y1
         XY22 = X2*Y2
         XY33 = X3*Y3
         TEMP1 = -3.*XY11 - 5.*XY33
         TEMP = XY11-8.*XY22-5.*XY33 + 2.*R*XY12 + RS2*TEMP1
         XDNDN = 8.*(2.*TEMP + 3.*DLOG*(-15.*XY11-5.*XY33 - 10.*R*XY12
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)
         RETURN
C        . 1,1 .
 11      X23 = X2 - X3
         Y23 = Y2 - Y3
         X1X2 = X1 + X2
         Y1Y2 = Y1 + Y2
         TEMP = (X3+X1)*(Y3+Y1) + 3.*X23*Y23
      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
      OVERLAY(MAIN,5,0)
      PROGRAM TRAP9
C*********************************************************************
C
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR TRAPEZOIDAL ELEMENTS
C        WITH  NNE = 8.
C
C     THE  DO  LOOPS ENDING AT STATEMENTS NUMBERED  3 AND 4  CARRY OUT
C        GROUP TRANSFORMATIONS.
C
C*********************************************************************
C
      DIMENSION LC(1),KC(1)
      COMMON/SPACE/XC(107),LIMIT,SKIP(2),IC,SKP(6),IXC,SKIPP(2),
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,DLG,RS2,CLG,OTHERS(1)
      COMMON/TEMP/I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,R,S,
     *   IL,IU,LL,KI,XJ,TEMP
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))
C
C
      X1 = XX1
      X2 = XX2
      X3 = XX3
      Y1 = YY1
      Y2 = YY2
      Y3 = YY3
      Z1 = ZZ1
      R  = ZZ2/ZZ1
      S  = ZZ3/ZZ1
      IL = IXC + 1
      IU = IXC + LIMIT
      LL = 0
      DO 4 I=1,2
         DO 3 J=1,4
            LL = LL + 1
            KI = 0
            IF (ABS(R).LT.ABS(S))
C           THEN  SECOND TYPE OF TRAPEZOID
     *         KI = 11
C           CONTINUE
            DO 2 I1=IL,IU
               I2 = I1 + IC
               I3 = I2 + IC
               I4 = I3 + IC
               K = KC(I1) + KI
               L = LC(I2)
C              IF (L.EC.LL)
                                                  IF(L.NE.LL) GO TO 1
C              THEN
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)
 1             CONTINUE
 2          CONTINUE
C           . TRANSFORMATION OF TYPE ONE.
            X1 = -X1
            XJ =  X2
            X2 = -X3
            X3 =  XJ
            Y1 = -Y1
            XJ =  Y2
            Y2 = -Y3
            Y3 =  XJ
            XJ =  R
            R  =  S
            S  = -XJ
 3       CONTINUE
```

```
C         . TRANSFORMATION OF TYPE TWO.
          X3 = -X3
          Y3 = -Y3
          S  = -S
  4    CONTINUE
       END
```

APPENDIX A

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)
C*********************************************************************
C
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM TRAP9 TO EVALUATE
C        C-INTEGRALS.
C
C     RS2  IS THE LARGER OF THE TWO QUANTITIES  R*R  AND S*S.
C
C*********************************************************************
C
      COMMON/SPACE/SKIP(129),DLOG,RS2,CLOG,OTHERS(1)
      COMMON/TEMP/I2,I3,I4,K,L,XY(6),Z1,R,S,
     *   IL,IU,LL,KI,XJ,TEMP,TEMP1,XY11,XY22,XY33,XY12,XY23,XY31,
     *   YX21,YX32,YX13
C
C
      XY11 = X1*Y1
      XY22 = X2*Y2
      XY33 = X3*Y3
      XY12 = X1*Y2 + X2*Y1
      XY23 = X2*Y3 + X3*Y2
      XY31 = X3*Y1 + X1*Y3
      YX21 = X1*Y2 - X2*Y1
      YX32 = X2*Y3 - X3*Y2
      YX13 = X3*Y1 - X1*Y3
      GO TO (101,102,103,104,105,106,107,108,109,110,111,
     *   201,202,203,204,205,206,207,208,209,210,211),K
C        . 1,1 .
 101     TEMP1 = -12.*XY11-8.*XY22-20.*XY33-6.*XY12+10.*XY23
         TEMP = XY11-104.*XY22-95.*XY33-13.*XY12+85.*XY23+5.*XY31
     *      +R*(12.*XY11+40.*(XY22+XY33)+23.*XY12-35.*XY23-10.*XY31)
     *      +RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(-65.*XY11-15.*XY33-30.*XY31
     *      +R*(-20.*XY11+30.*XY33-70.*XY12-30.*XY23+35.*XY31)
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)
            RETURN
C        . 2,1 .
 102     TEMP1 = -12.*XY11+2.*XY22-20.*XY33-6.*XY12+10.*YX32
         TEMP = XY11-34.*XY22-65.*XY33-23.*XY12+25.*YX13+15.*YX32
     *      +R*(12.*XY11+40.*XY33+23.*XY12-10.*YX13-25.*YX32)
     *      +RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(-55.*XY11+15.*XY33-30.*YX13
     *      +R*(-20.*XY11-30.*XY33-50.*XY12+15.*YX13+30.*YX32)
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)
            RETURN
C        . 3,1 .
 103     TEMP1 = -12.*XY11-2.*XY22-20.*XY33+6.*YX21+10.*XY23
          TEMP = 7.*XY11-46.*XY22-55.*XY33+35.*XY23+15.*YX13+3.*YX21
     *       +R*(17.*XY12-10.*XY31-15.*YX32) + RS2*TEMP1
          XDNDN = (2.*TEMP + 3.*DLOG*(-55.*XY11+15.*XY33-30.*YX13
     *      +R*(-50.*XY12+25.*XY31+30.*YX32)
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)
            RETURN
C        . 5,1 .
 104     TEMP1 = 6.*XY11-XY22+10.*XY33+0.5*XY12-2.5*YX21-5.*X2*Y3
         TEMP = 12.*XY11-3.*XY22+40.*XY33+16.5*XY12-12.5*YX21-10.*XY23
     *      -15.*YX32-5.*XY31-10.*YX13
     *      +R*(-11.*XY11+10.*XY22-20.*XY33-6.5*XY12+7.5*YX21+5.*XY23
     *      +10.*YX32+5.*X3*Y1) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(20.*XY11+2.5*XY31+7.5*YX13)
     *      +RS2*(TEMP-3.*TEMP1)))/(90.*Z1)
            RETURN
C        . 5,3 .
 105     TEMP1 = 6.*XY11+XY22+10.*XY33-2.5*XY12+0.5*YX21-5.*X2*Y3
         TEMP =4.*XY11+3.*XY22+20.*XY33+7.5*XY12-3.5*YX21-5.*(XY23+XY31)
     *      +R*(-5.*XY11+10.*XY22-3.5*XY12+2.5*YX21-5.*XY23+5.*X3*Y1)
     *      +RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(20.*XY11-2.5*XY31-7.5*YX13)
     *      +RS2*(TEMP-3.*TEMP1)))/(90.*Z1)
            RETURN
```

```
C        . 5,5 .
 106     TEMP1 = -6.*XY11-4.*XY22-10.*XY33+2.*XY12
         TEMP = -32.*XY11-12.*XY22-40.*XY33+6.*XY12
     *      +R*(16.*XY11+20.*XY33-6.*XY12) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*RS2*(TEMP+18.*XY11+12.*XY22+30.*XY33
     *      -6.*XY12))/(45.*Z1)
            RETURN
C        . 6,5 .
 107     TEMP1 = XY31-X2*Y3
         TEMP = 2.*XY11-3.*XY12-YX21-2.*XY23+3.*XY31+YX13
     *      +R*(-2.*XY22+2.*(X1+X3)*Y2-1.5*XY31+0.5*YX13) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(-4.*XY11+0.5*XY31-1.5*YX13)
     *      +RS2*(TEMP-3.*TEMP1)))/(9.*Z1)
            RETURN
C        . 7,5 .
 108     TEMP1 = -6.*XY11+4.*XY22-10.*XY33-2.*YX21
         TEMP = -4.*XY11+12.*XY22-20.*XY33-6.*YX21 +6.*R*XY12 +RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*RS2*(TEMP-3.*TEMP1))/(45.*Z1)
            RETURN
C        . 9,1 .
 109     TEMP1 = 6.*XY11+10.*XY33+X2*(3.*Y1-5.*Y3)
         TEMP = -2.*XY11+20.*(XY22+XY33)+3.*XY12-YX21-10.*XY23-5.*YX13
     *      +R*(-3.*XY11-2.*XY22-10.*XY33-7.*XY12+3.*YX21+10.*X2*Y3
     *      +5.*X3*Y1) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(30.*XY11+5.*XY31+10.*YX13
     *      +R*(5.*XY11+10.*XY33+25.*XY12-5.*YX21-10.*YX32-15.*X3*Y1)
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)
            RETURN
C        . 9,5 .
 110     TEMP1 = -12.*XY11-20.*XY33+4.*X2*Y1
         TEMP = -16.*XY11-40.*XY33-16.*XY12-8.*YX21
     *      +R*(16.*XY11-16.*XY22+20.*XY33+8.*X1*Y2) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(R*(-40.*XY11-20.*XY33)
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)
            RETURN
C        . 9,9 .
 111     TEMP1 = -3.*XY11-5.*XY33
         TEMP = XY11-8.*XY22-5.*XY33 + 2.*R*XY12 + RS2*TEMP1
         XDNDN =8.*(2.*TEMP + 3.*DLOG*(-15.*XY11-5.*XY33 -10.*R*XY12
     *      +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)
            RETURN
C        . 1,1 .
 201     TEMP1 = -12.*XY11-20.*XY22-8.*XY33+10.*XY23-6.*XY31
         TEMP = XY11-95.*XY22-104.*XY33+5.*XY12+85.*XY23-13.*XY31
     *      +S*(12.*XY11+40.*(XY22+XY33)-10.*XY12-35.*XY23+23.*XY31)
     *      +RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(-65.*XY11-15.*XY22-30.*XY12
     *      +S*(-20.*XY11+30.*XY22+35.*XY12-30.*XY23-70.*XY31)
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)
            RETURN
C        . 2,1 .
 202     TEMP1 = -12.*XY11-20.*XY22+8.*XY33-10.*YX32-6.*YX13
         TEMP = 7.*XY11-25.*XY22-56.*XY33-5.*XY12+7.*YX13+15.*YX32
     *      +S*(-5.*XY23+17.*XY31+10.*YX21) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(-65.*XY11-15.*XY22-30.*XY12
     *      +S*(-30.*XY23-70.*XY31-45.*YX21)
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)
            RETURN
C        . 3,1 .
 203     TEMP1 = -12.*XY11-20.*XY22-2.*XY33+10.*XY23-6.*YX13
        TEMP = 7.*XY11-55.*XY22-46.*XY33+35.*XY23-3.*YX13-15.*YX21
     *      +S*(-10.*XY12+17.*XY31+15.*YX32) + RS2*TEMP1
         XDNDN = (2.*TEMP + 3.*DLOG*(-55.*XY11+15.*XY22+30.*YX21
     *      +S*(25.*XY12-50.*XY31-30.*YX32)
     *      +RS2*(TEMP-3.*TEMP1)))/(360.*Z1)
            RETURN
C        . 5,1 .
 204     TEMP1 = 2.*XY12-X3*(Y1-Y2)
```

```
      TEMP =-3.*XY11-3.*XY22+16.*XY33+3.*XY12-5.*YX21-4.*XY23-6.*YX32
     *   -2.*XY31+4.*YX13
     *   +S*(3.*XY22-4.*XY33-1.5*XY12+0.5*YX21-XY23+5.*YX32-XY31
     *   +3.*YX13) + RS2*TEMP1
      XDNDN = (2.*TEMP + 3.*DLOG*(13.*XY11+3.*XY22+6.*XY12
     *   +S*(2.*XY11-3.*XY22-3.5*XY12+4.5*YX21+6.*XY23+14.*XY31)
     *   +RS2*(TEMP-3.*TEMP1)))/(36.*Z1)
         RETURN
C     . 5,3 .
 205  TEMP1 = 2.*XY12-X3*(Y1+Y2)
      TEMP = -5.*XY11+3.*XY22+8.*XY33+3.*XY12-5.*YX21-2.*XY23-2.*XY31
     *   +S*(3.*XY22+1.5*XY12-0.5*YX21-XY23+5.*YX32-3.*XY31+YX13)
     *   +RS2*TEMP1
      XDNDN = (2.*TEMP + 3.*DLOG*(11.*XY11-3.*XY22+6.*YX21
     *   +S*(-2.*XY11-3.*XY22-2.5*XY12+1.5*YX21+10.*XY31-6.*YX32)
     *   +RS2*(TEMP-3.*TEMP1)))/(36.*Z1)
         RETURN
C     . 5,5 .
 206  TEMP1 = -XY11-3.*XY22
      TEMP = -5.*XY11-3.*XY22-16.*XY33+6.*XY12 +S*(6.*XY23-2.*XY31)
     *   +RS2*TEMP1
      XDNDN = (2.*TEMP + 3.*DLOG*(-13.*XY11-3.*XY22-6.*XY12
     *   +S*(-6.*XY23-14.*XY31) + RS2*(TEMP-3.*TEMP1)))/(18.*Z1)
         RETURN
C     . 6,5 .
 207  TEMP1 = -XY12-X2*Y3
      TEMP = 2.*XY11-3.*XY12-YX21-2.*XY23+3.*XY31+YX13
     *   +S*(2.*XY33-1.5*XY12+0.5*YX21+2.*X3*(Y1-Y2)) + RS2*TEMP1
      XDNDN = (2.*TEMP + 3.*DLOG*(S*(4.*XY11+0.5*XY12-1.5*YX21)
     *   +RS2*(TEMP-3.*TEMP1)))/(9.*Z1)
         RETURN
C     . 7,5 .
 208  TEMP1 = XY11+3.*XY22
      TEMP = 5.*XY11+3.*XY22-8.*XY33-6.*YX21
     *   +S*(2.*XY31+6.*YX32) + RS2*TEMP1
      XDNDN = (2.*TEMP + 3.*DLOG*(-11.*XY11+3.*XY22+6.*YX21
     *   +S*(-10.*XY31-6.*YX32) + RS2*(TEMP-3.*TEMP1)))/(18.*Z1)
         RETURN
C     . 9,1 .
 209  TEMP1 = 6.*XY11+10.*XY22+X3*(3.*Y1-5.*Y2)
      TEMP = -2.*XY11+20.*(XY22+XY33)+5.*YX21-10.*XY23+3.*XY31+YX13
     *   +S*(-3.*XY11-10.*XY22-2.*XY33+5.*X2*Y1+10.*X3*Y2-7.*XY31
     *   -3.*YX13) + RS2*TEMP1
      XDNDN = (2.*TEMP + 3.*DLOG*(30.*XY11+5.*XY12-10.*YX21
     *   +S*(5.*XY11+10.*XY22-15.*X2*Y1+10.*YX32+25.*XY31+5.*YX13)
     *   +RS2*(TEMP-3.*TEMP1)))/(45.*Z1)
         RETURN
C     . 9,5 .
 210  TEMP = 4.*XY11-8.*XY33-2.*XY12-4.*YX21 + 4.*S*(YX32+X3*Y1)
     *   - 2.*RS2*XY12
      XDNDN = (2.*TEMP + 3.*DLOG*(-12.*XY11-2.*XY12+4.*YX21
     *   +S*(-4.*YX32-10.*XY31-2.*YX13) +RS2*(TEMP+6.*XY12)))/(9.*Z1)
         RETURN
C     . 9,9 .
 211  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
```

```
      OVERLAY(MAIN,6,0)
      PROGRAM QUAD5
C**********************************************************************
C
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE
C        ELEMENTS WITH NNE = 4.
C
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT
C        GROUP TRANSFORMATIONS.
C
C**********************************************************************
C
      DIMENSION LC(1),KC(1)
      COMMON/SPACE/XC(107),LIMIT,SKIP(2),IC,SKP(6),IXC,SKIPP(2),
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,VLG1,VLG2,VLG3,
     *   ULG1,ULG2,ULG3,OTHERS(1)
      COMMON/TEMP/CL0,CL1,CL2,CL3,I,IL,IU,I1,I2,I3,I4,J,K,L,LL,
     *   R,S,R2,S2,RS,RD,SD,
     *   T1,T2,T3,T4,T5,T6,ULOG1,ULOG2,ULOG3,U1,U2,U3,U4,U5,U6,
     *   VLOG1,VLOG2,VLOG3,V2,V3,V4,V5,V6,XJ,X1,X2,X3,Y1,Y2,Y3,
     *   X2X1,X3X1,X32,Y2Y1,Y3Y1,Y3Y2,Y31,Y32,Z1
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))
C
C
      X1 = XX1
      X2 = XX2
      X3 = XX3
      Y1 = YY1
      Y2 = YY2
      Y3 = YY3
      Z1 = ZZ1
      R  = ZZ2/ZZ1
      S  = ZZ3/ZZ1
      R2 = R*R
      S2 = S*S
      RS = R*S
      RD = 1./(1.-R2)
      SD = 1./(1.-S2)
      ULOG1 = ULG1
      ULOG2 = ULG2
      ULOG3 = ULG3
      VLOG1 = VLG1
      VLOG2 = VLG2
      VLOG3 = VLG3
      IL = IXC + 1
      IU = IXC + LIMIT
      LL = 0
      DO 4 I=1,2
         DO 3 J=1,4
            LL = LL + 1
            DO 2 I1=IL,IU
               I2 = I1 + IC
               I3 = I2 + IC
               I4 = I3 + IC
               K  = KC(I1)
               L  = LC(I2)
C              IF (L.EQ.LL)
                                                 IF(L.NE.LL) GO TO 1
C              THEN
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)
 1             CONTINUE
 2          CONTINUE
C           . TRANSFORMATION OF TYPE ONE.
            X1 = -X1
            XJ =  X2
            X2 = -X3
            X3 =  XJ
            Y1 = -Y1
            XJ =  Y2
            Y2 = -Y3
            Y3 =  XJ
            XJ =  R
            R  =  S
            S  = -XJ
            XJ =  R2
            R2 =  S2
            S2 =  XJ
            RS = -RS
            XJ =  RD
            RD =  SD
            SD =  XJ
            XJ    = ULOG2
            ULOG2 = ULOG3
            ULOG3 = XJ
            XJ    = VLOG2
            VLOG2 = VLOG3
            VLOG3 = XJ
 3        CONTINUE
C         . TRANSFORMATION OF TYPE TWO.
          X3 = -X3
          Y3 = -Y3
          S = -S
          RS = -RS
 4      CONTINUE
        END
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)
C***********************************************************************
C
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD5 TO EVALUATE
C        C-INTEGRALS.
C
C***********************************************************************
C
C
C
      COMMON/TEMP/CLO,CL1,CL2,CL3,I,IL,IU,I1,I2,I3,I4,J,K,L,LL,
     *   R,S,R2,S2,RS,RD,SD,
     *   T1,T2,T3,T4,T5,T6,ULOG1,ULOG2,ULOG3,U1,U2,U3,U4,U5,U6,
     *   VLOG1,VLOG2,VLOG3,V2,V3,V4,V5,V6,XJ,XY(6),
     *   X2X1,X3X1,X32,Y2Y1,Y3Y1,Y3Y2,Y31,Y32,Z1
C
C
      CL04(S,SSS3,R2,S2,T1,T3,T5,U1,U2) = 2.*(RD*SD)**3*
     *   (T1*S*(4.*SSS3+R2*(108.+S2*(-114.+S2*(108.-30.*S2))
     *       +R2*(186.+S2*(-264.+S2*(33.+9.*S2))+R2*(-64.+S2*(69.-6.*S2)
     *       +R2*(-18.+9.*S2)))))
     *   +T3*S*(-20.*SSS3+R2*(-546.+S2*(1140.+S2*(-582+60.*S2))
     *       +R2*(-360.+S2*(15.+69.*S2)+R2*(278.-111.*S2))))
     *   +T5*(-20.*SSS3+R2*(50.+S2*(-48.+S2*(-300.+154.*S2))
     *       +R2*(-30.+S2*(450.+S2*(-45.-33.*S2))+R2*(-10.+S2*(-276.
     *       +57.*S2)+R2*(10.+30.*S2)))))
     *   +U1*S2*(SSS3+R2*S2*(-75.+S2*(43.+3.*S2)-9.*R2*S2))
     *   +U2*(-4.+8.*SSS3+R2*S2*(-45.+S2*(48.-6.*S2)+R2*S2*(10.5-9.*S2)
     *       )))
      CL14(R,S,R2,S2,T1,T3,T5,U1,U2) = -3.*R2*S*
     *   (T1*(S2*(-1.+S2*(6.+3.*S2))+R2*(3.-3.*S2*S2+3.*R2*(6.-S2+R2)))
     *   +T3*(3.+R2*(-30.-45.*R2)+S2*(-10.+30.*R2+15.*S2))
     *   +T5*S*(-1.+R2*(10.+15.*R2)-15.*S2*S2)
     *   +U1*R2*S*(1.+3.*R2)
     *   +U2*S*(0.5+R2*(-6.+3.*S2-3.*R2)))
      CL24(R,S,R2,S2,T1,T2,T3,T4,T5,T6,U1,U2) = 3.*S*
     *   (T1*(-12.*(1.+R2)+4.*S2)   -T2*8.*R*S
     *   +T3*(60.+12.*R2-20.*S2)    -T4*8.*R*S
     *   -T5*S*(20.+4.*R2)          +T6*S*(10.+6.*R2-10.*S2)
     *   -U1*S*(3.+R2-S2)           +U2*8.*S)
      CL05(R2,S2,T1,T2,T4,T6) = (RD*SD)**3*
     *   (T1*(20.+R2*(480.-1176.*S2+R2*(1596.+S2*(-1080.+762.*S2)
     *       +R2*(-40.+S2*(-144.-156.*S2)+R2*(-348.+216.*S2)))))
     *   +T2*(-70.+S2*(140.+S2*S2*(-140.+70.*S2))
     *       +R2*(936.+S2*(-1842.+S2*(3822.+S2*(-1974+210.*S2)))
     *       +R2*(2610.+S2*(-5940.+S2*(1626.+168.*S2))
     *       +R2*(532.+S2*(982.-442.*S2) + R2*(-840.+420.*S2)))))
     *   +T4*(-80.+S2*(1044.+S2*(540.+S2*(-292.-60.*S2)))
     *       +R2*(-1368.+S2*(1740.+S2*(-2706.+S2*(624.+30.*S2)))
     *       +R2*(-1800.+S2*(2208.+S2*(-204.-20.*S2))
     *       +R2*(824.+S2*(-708.+130.*S2) + R2*(120.-60.*S2)))))
     *   +T6*(-148.+R2*(-6168.+11730.*S2+R2*(-3192.+S2*(-5472.+612.*S2)
     *       +R2*(2744.-588.*S2)))))
      CL15(R2,S2,T1,T2,T4,T6) = -3.*
     *   (T1*R2*(-S2+R2*(6.+R2*(60.+30.*(R2-S2))))
     *   +T2*R2*(1.+S2*(-7.+35.*S2*(1.+S2))+R2*R2*(105.*(1.-S2)+70.*R2))
     *   +T4*(S2*(-1.+S2*(15.+S2*(45.+5.*S2)))+R2*(2.-5.*S2*(1.+S2*S2)
     *       +R2*(-30.+S2*(45.-15.*S2)+R2*(-90.+25.*S2-10.*R2))))
     *   +T6*(-3.+R2*(42.-70.*S2+210.*R2*(-1.+S2-R2))))
      CL25(R2,S2,T1,T2,T3,T4,T5,T6) = -3.*
     *   (T1*(30.+S2*(-20.+6.*S2)+R2*(60.-12.*S2+6.*R2))
     *   +T2*(35.+S2*(-70.+35.*S2)+R2*(126.-42.*S2+15.*R2))
     *   +T3*S2*(56.+8.*R2)
     *   +T4*(-80.+64.*S2-48.*R2)
     *   +T5*(40.-8.*S2+24.*R2)
     *   +T6*(-336.+112.*S2-48.*R2))
```

```
C
C
      IF (K.LE.3)                                 GO TO 1
                                                  GO TO 6
C     THEN
 1       X32 = X3 - X2
         X2X1 = X2 + X1
         GO TO (2,3,4),K
C
C        . 1,1 .
 2          Y32 = Y3 - Y2
            Y2Y1 = Y2 + Y1
            X3X1 = X3 + X1
            Y3Y1 = Y3 + Y1
            T1 = -X32*Y32
            T2 = X2X1*Y2Y1
            T3 = X3X1*Y3Y1
            T4 = X2X1*Y3Y1 + X3X1*Y2Y1
            T5 = X2X1*Y32 + X32*Y2Y1
            T6 = -X3X1*Y32 - X32*Y3Y1
                                                  GO TO 5
C        . 2,1 .
 3          Y31 = Y3 - Y1
            Y2Y1 = Y2 + Y1
            X3X1 = X3 + X1
            Y3Y2 = Y3 + Y2
            T1 =  X32*Y3Y2
            T2 = -X2X1*Y2Y1
            T3 = -X3X1*Y31
            T4 = -X2X1*Y3 - X3X1*Y2 - X32*Y1
            T5 =  T4
            T6 =  X32*Y31 + X3X1*Y3Y2
                                                  GO TO 5
C        . 3,1 .
 4          Y21 = Y2 - Y1
            Y31 = Y3 - Y1
            X3X1 = X3 + X1
            T1 = X32*(Y3-Y2)
            T2 = X2X1*Y21
            T3 = X3X1*Y31
            T4 = X2X1*Y31 + X3X1*Y21
            T5 = -X2X1*Y3 + X3X1*Y2 - X32*Y1
            T6 = -T5
 5       CONTINUE
         U1 = 6.*T1
         U2 = T2 + T2
         V2 = U2*R2
         U3 = T3 + T3
         V3 = U3*S2
         U4 = T4*RS
         V4 = U4 + U4
         U5 = 3.*S*T5
         V5 = U5*R2
         U6 = 3.*R*T6
         V6 = U6*S2
         CL0 = 2.*RD*SD*(U1*(1.-R2-S2)+4.*(V2+V3+U4)+U4+V5+V6)
         CL1 = U1*R2*S2 - V2*(1.+3.*R2) - V3*(1.+3.*S2)
     *      + U4*(1.-3.*(R2+S2)) - V5*(1.+R2-S2) - V6*(1.-R2+S2)
         CL2 = U1*S2 - U2*(3.+R2) - V3 - V4 - U5 - U5
         CL3 = U1*R2 - V2 - U3*(3.+S2) - V4 - U6 - U6
         XDNDN = (CL0 + CL1*ULOG1 + CL2*ULOG2 + CL3*ULOG3)/(48.*Z1)
C  ......EXIT
                                                  RETURN
C     ELSE
 6    IF (K.EQ.4)                                 GO TO 7
                                                  GO TO 8
```

```
C      THEN
C         . 5,1 .
 7        X32 = X3 - X2
          X2X1 = X2 + X1
          X3X1 = X3 + X1
          T1 = 5.*X2X1*Y1
          T2 = 5.*X3X1*Y1
          T3 = R*X2X1*Y2
          T4 = S*X3X1*Y3
          T5 = X2X1*Y3
          T5 = T5 + T5
          T6 = X3X1*Y2
          T6 = T6 + T6
          U1 = 20.*X32*Y1
          U2 = 5.*X32*(R*Y2-S*Y3)
          RRR3 = (1.-R2)**3
          SSS3 = (1.-S2)**3
          TEMP1 =  CL04(S,SSS3,R2,S2,T1,T3,T5,U1,U2)
          TEMP2 =  CL04(R,RRR3,S2,R2,T2,T4,T6,-U1,U2)
          TEMP3 = CL14(R,S,R2,S2,T1,T3,T5,U1,U2)
          TEMP4 = CL14(S,R,S2,R2,T2,T4,T6,-U1,U2)
          TEMP5 = CL24(R,S,R2,S2,T1,T2,T3,T4,T5,T6,U1,U2)
          TEMP6 = CL24(S,R,S2,R2,T2,T1,T4,T3,T6,T5,-U1,U2)
          XDNDN = (TEMP1+TEMP2+(TEMP3+TEMP4)*VLOG1
     *         +TEMP5*VLOG2+TEMP6*VLOG3)/(180.*Z1)
C  ......EXIT
                                                     RETURN
C      ELSE  (K  MUST EQUAL  5)
C         . 5,5 .
 8        T1 = 28.*X1*Y1
          T2 = 8.*X2*Y2
          T3 = 8.*X3*Y3
          T4 = 14.*R*(X1*Y2+X2*Y1)
          T5 = 14.*S*(X1*Y3+X3*Y1)
          T6 = RS*(X2*Y3+X3*Y2)
          TEMP1 =  CL05(R2,S2,T1,T2,T4,T6)
          TEMP2 = CL05(S2,R2,T1,T3,T5,T6)
          TEMP3 = CL15(R2,S2,T1,T2,T4,T6)
          TEMP4 = CL15(S2,R2,T1,T3,T5,T6)
          TEMP5 = CL25(R2,S2,T1,T2,T3,T4,T5,T6)
          TEMP6 = CL25(S2,R2,T1,T3,T2,T5,T4,T6)
          XDNDN = (TEMP1+TEMP2+(TEMP3+TEMP4)*VLOG1
     *         +TEMP5*VLOG2+TEMP6*VLOG3)/(315.*Z1)
C  ......EXIT
                                                     RETURN
C      CONTINUE
       END
```

APPENDIX A

```
      OVERLAY(MAIN,7,0)
      PROGRAM QUAD81
C***********************************************************************
C
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE
C        ELEMENTS WITH NNE = 8.
C
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT
C        GROUP TRANSFORMATIONS.
C
C***********************************************************************
C
      DIMENSION LC(1),KC(1)
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,SKP(2),
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,ALG1,ALG2,ALG3,OTHERS(1)
      COMMON/TEMP/I1,I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,
     *   T1,T2,T3,T4,T5,T6,
     *   DL1,EL1,DL2,FL3,XJ
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))
C
C
      X1 = XX1
      X2 = XX2
      X3 = XX3
      Y1 = YY1
      Y2 = YY2
      Y3 = YY3
      Z1 = ZZ1
      Z2 = ZZ2
      Z3 = ZZ3
      ALOG1 = ALG1
      ALOG2 = ALG2
      ALOG3 = ALG3
      IL = IXC + 1
      IU = IXC + 10
      LL = 0
      DO 4 I=1,2
         DO 3 J=1,4
            LL = LL + 1
            D = Z2/Z3
            DP1 = D + 1.
            DM1 = D - 1.
            D2 = D*D
            D3 = D*D2
            E = -Z1/Z3
            EP1 = E + 1.
            EM1 = E - 1.
            E2 = E*E
            E3 = E*E2
            FF = Z1/Z2
            F2 = FF*FF
            F4 = F2*F2
            F6 = F2*F4
            G = Z3/Z2
            G2 = G*G
            G4 = G2*G2
            G6 = G2*G4
C           . T'L' = (Z2/Z3)*(((Z1+Z3)/Z2)**L - ((Z1-Z3)/Z2)**L)
            T1 = 2.
            T2 = 4.*FF
            T3 = 2.*G2 + 6.*F2
            T4 = 8.*FF*(G2+F2)
            T5 = 2.*G4 + 20.*G2*F2 + 10.*F4
            T6 = 2.*FF*T3*(3.*G2+F2)
            DL1 = D*ALOG1
            EL1 = -E*ALOG1
            DL2 = D*ALOG2
            FL3 = FF*ALOG3
            DO 2 I1=IL,IU
               I2 = I1 + IC
               I3 = I2 + IC
               I4 = I3 + IC
               K = KC(I1)
               L = LC(I2)
C              IF (L.EQ.LL)
                                                  IF(L.NE.LL) GO TO 1
C              THEN
                  XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)
                  XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)
                  XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)
                  XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)
 1             CONTINUE
 2          CONTINUE
C           . TRANSFORMATION OF TYPE ONE.
            X1 = -X1
            XJ =  X2
            X2 = -X3
            X3 =  XJ
            Y1 = -Y1
            XJ =  Y2
            Y2 = -Y3
            Y3 =  XJ
            XJ =  Z2
            Z2 =  Z3
            Z3 = -XJ
            ALOG1 = -ALOG1
            XJ    =  ALOG2
            ALOG2 = -ALOG3
            ALOG3 =  XJ
 3       CONTINUE
C        . TRANSFORMATION OF TYPE TWO.
         X3 = -X3
         Y3 = -Y3
         Z3 = -Z3
         ALOG1 = -ALOG1
         ALOG2 = -ALOG2
 4    CONTINUE
      END
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)
C***********************************************************************
C
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD81 TO EVALUATE
C        C-INTEGRALS.
C
C***********************************************************************
C
C
      COMMON/TEMP/I1,I2,I3,I4,K,L,XY(6),Z1,Z2,Z3,
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,
     *   T1,T2,T3,T4,T5,T6,
     *   DL1,EL1,DL2,FL3,SUML,CF(7)
C
      D2 = D*D
      D3 = D*D2
      D4 = D*D3
      E2 = E*E
      E3 = E*E2
      E4 = E*E3
      SUML = 0.
      GO TO (301,302,303),K
C           . 1,1
 301        XDNDN = (((240.*E-120.*D+120.)*X3+(((-90.*E2 ))+(60.*D-165.)
     *         *E-30.*D2 +105.*D-135.)*X2+((((-210.*E2 ))+(180.*D-15.)*E
     *         -78.*D2 +21.*D-1.)*X1))*Y3+((((-90.*E2 ))+(60.*D-165.)*E-
     *         30.*D2 +105.*D-135.)*X3+(45.*E3  +90.*E2 +(45.*D2 -120.*D
     *         +120.)*E+30.*D2 -120.*D+150.)*X2+((90.*E3  +(((-45.*D))+1
     *         05.)*E2 +(90.*D2 -150.*D+15.)*E-9.*D3  +39.*D2 -47.*D+5.)
     *         *X1))*Y2+(((((-210.*E2 ))+(180.*D-15.)*E-78.*D2 +21.*D-1.
     *         )*X3+(90.*E3  +(((-45.*D))+105.)*E2 +(90.*D2 -150.*D+15.)
     *         *E-9.*D3  +39.*D2 -47.*D+5.)*X2+((195.*E3  +(((-180.*D))-
     *         60.)*E2 +(207.*D2 +36.*D-46.)*E-36.*D3  -36.*D2 +16.*D+16
     *         .)*X1))*Y1))/180.
            CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1
     *         ))*Y3+((((-2.*E3  ))-(2.*E2 ))*X3+(E4  +2.*E3  +E2 )*X2+(
     *         (2.*E4  +(2.*E3  ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2
     *         .*E3  ))*X2+(4.*E4  *X1))*Y1))/16.)
            CF(2) = (((8.*E2 +((((-8.*D))+4.)*E))*X3+(((-2.*E3  ))+(6.*D
     *         -7.)*E2 +((4.*D-5.)*E))*X2+((((-6.*E3  ))+((12.*D-2.)*E2
     *         ))*X1))*Y3+((((-2.*E3  ))+(6.*D-7.)*E2 +((4.*D-5.)*E))*X3
     *         +((((-4.*D))+4.)*E3  +(((-6.*D))+8.)*E2 +((((-2.*D))+4.)*
     *         E))*X2+((E4  +(((-8.*D))+5.)*E3  +((((-6.*D))+4.)*E2 ))*X
     *         1))*Y2+(((((-6.*E3  ))+((12.*D-2.)*E2 ))*X3+(E4  +(((-8.*
     *         D))+5.)*E3  +((((-6.*D))+4.)*E2 ))*X2+((4.*E4  -(16.*D*E*
     *         *3))*X1))*Y1))/16.
            CF(3) = (-(((4.*E2 +(((-16.*D))+8.)*E+4.*D2 -4.*D+1.)*X3+((6
     *         .*D-5.)*E2 +(((-6.*D2 ))+14.*D-7.)*E-2.*D2 +5.*D-2.)*X2+(
     *         (((-2.*E3  ))+(18.*D-1.)*E2 +((((-12.*D2 ))+4.*D+4.)*E))*
     *         X1))*Y3+(((6.*D-5.)*E2 +(((-6.*D2 ))+14.*D-7.)*E-2.*D2 +5
     *         .*D-2.)*X3+((6.*D2 -12.*D+4.)*E2 +(6.*D2 -16.*D+8.)*E+D2
     *         -4.*D+4.)*X2+(((((-4.*D))+1.)*E3  +(12.*D2 -15.*D-1.)*E2
     *         +((6.*D2 -8.*D-2.)*E))*X1))*Y2+(((((-2.*E3  ))+(18.*D-1.)
     *         *E2 +((((-12.*D2 ))+4.*D+4.)*E))*X3+(((((-4.*D))+1.)*E3  +
     *         (12.*D2 -15.*D-1.)*E2 +((6.*D2 -8.*D-2.)*E))*X2+((E4  +((
     *         (-16.*D))-4.)*E3  +((24.*D2 -8.)*E2 ))*X1))*Y1))/16.)
            CF(4) = (-((((8.*D-4.)*E-8.*D2 +8.*D-2.)*X3+((6.*D2 -10.*D+2
     *         .)*E-2.*D3  +7.*D2 -7.*D+2.)*X2+(((((-6.*D))-1.)*E2 +(18.
     *         *D2 -2.*D-5.)*E-4.*D3  +2.*D2 +4.*D-2.)*X1))*Y3+(((6.*D2
     *         -10.*D+2.)*E-2.*D3  +7.*D2 -7.*D+2.)*X3+((4.*D3  -12.*D2
     *         +(8.*D))*E+2.*D3  -8.*D2 +(8.*D))*X2+(((((-6.*D2 ))+3.*D+
     *         2.)*E2 +(8.*D3  -15.*D2 -2.*D+6.)*E+2.*D3  -4.*D2 -2.*D+4
     *         .)*X1))*Y2+((((((-6.*D))-1.)*E2 +(18.*D2 -2.*D-5.)*E-4.*D
     *         3  +2.*D2 +4.*D-2.)*X3+((((-6.*D2 ))+3.*D+2.)*E2 +(8.*D**
     *         3-15.*D2 -2.*D+6.)*E+2.*D3  -4.*D2 -2.*D+4.)*X2+(((4.*D+2
     *         .)*E3  +(((-24.*D2 ))-12.*D+4.)*E2 +((16.*D3  -(16.*D))*E
     *         ))*X1))*Y1))/16.)
```

```
           CF(5) = (-(((4.*D2 -4.*D+1.)*X3+(2.*D3  -5.*D2 +(2.*D))*X2+(
     *        ((((-6.*D2 ))-2.*D+1.)*E+6.*D3  -D2 -5.*D+2.)*X1))*Y3+((2
     *        .*D3  -5.*D2 +(2.*D))*X3+(D4  -4.*D3  +(4.*D2 ))*X2+((((
     *        -4.*D3  ))+3.*D2 +(4.*D))*E+2.*D4  -5.*D3  -D2 +(6.*D))*X
     *        1))*Y2+((((((-6.*D2 ))-2.*D+1.)*E+6.*D3  -D2 -5.*D+2.)*X3
     *        +((((-4.*D3  ))+3.*D2 +(4.*D))*E+2.*D4  -5.*D3  -D2 +(6.*
     *        D))*X2+(((6.*D2 +6.*D+1.)*E2 +(((-16.*D3  ))-12.*D2 +8.*D
     *        +4.)*E+4.*D4  -8.*D2 +4.)*X1))*Y1))/16.)
           CF(6) = (D*(D+1.)*((((2.*D)-1.)*X1*Y3)+(D2 -(2.*D))*X1*Y2+((
     *        (2.*D-1.)*X3+(D2 -(2.*D))*X2+(((((-4.*D))-2.)*E+4.*D2 -4.
     *        )*X1))*Y1))))/16.
           CF(7) =          (-(D2 *(D+1.)**2*X1*Y1)/16.)
           GO TO 350
C          . 2,1
 302       XDNDN = (((240.*E-(120.*D))*X3+(((-90.*E2 ))+(60.*D-135.)*E-
     *        30.*D2 +75.*D-45.)*X2+((((-210.*E2 ))+(180.*D+75.)*E-78.*
     *        D2 -57.*D-19.)*X1))*Y3+((((-90.*E2 ))+(60.*D+135.)*E-30.*
     *        D2 -75.*D-45.)*X3+(45.*E3  +((45.*D2 -90.)*E))*X2+((90.*E
     *        3  +(((-45.*D))-135.)*E2 +(90.*D2 +150.*D+45.)*E-9.*D3  -
     *        57.*D2 -83.*D-15.)*X1))*Y2+(((((-210.*E2 ))+(180.*D-75.)*
     *        E-78.*D2 +57.*D-19.)*X3+(90.*E3  +(((-45.*D))+135.)*E2 +(
     *        90.*D2 -150.*D+45.)*E-9.*D3  +57.*D2 -83.*D+15.)*X2+((195
     *        .*E3  -180.*D*E2 +(207.*D2 -64.)*E-36.*D3  +(16.*D))*X1))
     *        *Y1))/180.
           CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1
     *        ))*Y3+((((-2.*E3  ))+(2.*E2 ))*X3+(E4  -E2 )*X2+((2.*E4
     *        -(2.*E3  ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2.*E3  ))
     *        *X2+(4.*E4  *X1))*Y1))/16.)
           CF(2) = (((8.*E2 -(8.*D*E))*X3+(((-2.*E3  ))+(6.*D-5.)*E2 +(
     *        (4.*D-3.)*E))*X2+((((-6.*E3  ))+((12.*D+2.)*E2 ))*X1))*Y3
     *        +((((-2.*E3  ))+(6.*D+5.)*E2 +((((-4.*D))-3.)*E))*X3+(((-
     *        4.*D*E3  ))+(2.*D*E))*X2+((E4  +(((-8.*D))-5.)*E3  +((6.*
     *        D+4.)*E2 ))*X1))*Y2+(((((-6.*E3  ))+((12.*D-2.)*E2 ))*X3+
     *        (E4  +(((-8.*D))+5.)*E3  +((((-6.*D))+4.)*E2 ))*X2+((4.*E
     *        4  -(16.*D*E3  ))*X1))*Y1))/16.
           CF(3) = (-(((4.*E2 -16.*D*E+4.*D2 -1.)*X3+((6.*D-3.)*E2 +(((
     *        -6.*D2 ))+10.*D-1.)*E-2.*D2 +3.*D+2.)*X2+((((-2.*E3  ))+(
     *        18.*D+5.)*E2 +((((-12.*D2 ))-4.*D+4.)*E))*X1))*Y3+(((6.*D
     *        +3.)*E2 +(((-6.*D2 ))-10.*D-1.)*E+2.*D2 +3.*D-2.)*X3+((6.
     *        *D2 -4.)*E2 -D2 +4.)*X2+(((((-4.*D))-3.)*E3  +(12.*D2 +15
     *        .*D+1.)*E2 +((((-6.*D2 ))-8.*D+2.)*E))*X1))*Y2+(((((-2.*E
     *        3  ))+(18.*D-5.)*E2 +((((-12.*D2 ))+4.*D+4.)*E))*X3+((((-
     *        4.*D))+3.)*E3  +(12.*D2 -15.*D+1.)*E2 +((6.*D2 -8.*D-2.)*
     *        E))*X2+((E4  -16.*D*E3  +((24.*D2 -8.)*E2 ))*X1))*Y1))/16
     *        .)
           CF(4) = (-(((8.*D*E-8.*D2 +2.)*X3+((6.*D2 -6.*D-2.)*E-2.*D**
     *        3+5.*D2 -D-2.)*X2+(((((-6.*D))-3.)*E2 +(18.*D2 +10.*D-3.)
     *        *E-4.*D3  -2.*D2 +4.*D+2.)*X1))*Y3+(((6.*D2 +6.*D-2.)*E-2
     *        .*D3  -5.*D2 -D+2.)*X3+(4.*D3  -(8.*D))*E*X2+(((((-6.*D2
     *        ))-9.*D-2.)*E2 +(8.*D3  +15.*D2 +2.*D-2.)*E-2.*D3  -4.*D*
     *        D+2.*D+4.)*X1))*Y2+((((((-6.*D))+3.)*E2 +(18.*D2 -10.*D-3
     *        .)*E-4.*D3  +2.*D2 +4.*D-2.)*X3+((((-6.*D2 ))+9.*D-2.)*E*
     *        E+(8.*D3  -15.*D2 +2.*D+2.)*E+2.*D3  -4.*D2 -2.*D+4.)*X2+
     *        ((4.*D*E3  +(((-24.*D2 ))+4.)*E2 +((16.*D3  -(16.*D))*E))
     *        *X1))*Y1))/16.)
           CF(5) = (-(((4.*D2 -1.)*X3+(2.*D3  -3.*D2 -(2.*D))*X2+(((((-
     *        6.*D2 ))-6.*D-1.)*E+6.*D3  +5.*D2 -3.*D-2.)*X1))*Y3+((2.*
     *        D3  +3.*D2 -(2.*D))*X3+(D4  -(4.*D2 ))*X2+(((((-4.*D3  ))
     *        -9.*D2 -(4.*D))*E+2.*D4  +5.*D3  +D2 -(2.*D))*X1))*Y2+(((
     *        (((-6.*D2 ))+6.*D-1.)*E+6.*D3  -5.*D2 -3.*D+2.)*X3+((((-4
     *        .*D3  ))+9.*D2 -(4.*D))*E+2.*D4  -5.*D3  +D2 +(2.*D))*X2+
     *        (((6.*D2 -1.)*E2 +(8.*D-(16.*D3  ))*E+4.*D4  -8.*D2 +4.)*
     *        X1))*Y1))/16.)
           CF(6) = (D*((((2.*D2 )+3.*D+1.)*X1*Y3)+(D3  +3.*D2 +(2.*D))*
     *        X1*Y2+(((2.*D2 -3.*D+1.)*X3+(D3  -3.*D2 +(2.*D))*X2+(((2-
     *        (4.*D2 ))*E+4.*D3  -(4.*D))*X1))*Y1))))/16.
           CF(7) = (-((D4  -D2 )*X1*Y1)/16.)
           GO TO 350
```

```
C          . 3,1
 303       XDNDN = ((120.*E*X3+(((-90.*E2 ))+(((-60.*D))-15.)*E-30.*D2
     *        +45.*D+15.)*X2+((((-150.*E2 ))+(((-60.*D))+45.)*E-42.*D2
     *        +39.*D+1.)*X1))*Y3+((90.*E2 +(((-60.*D))-15.)*E+30.*D2 -4
     *        5.*D-15.)*X3+(((-45.*E3  ))+((((-45.*D2 ))+120.*D-30.)*E)
     *        )*X2+((((-90.*E3  ))+(45.*D+75.)*E2 +(((-90.*D2 ))+90.*D+
     *        15.)*E+9.*D3  +21.*D2 -73.*D-5.)*X1))*Y2+(((((-150.*E2 ))
     *        +(60.*D-45.)*E-42.*D2 +39.*D+1.)*X3+(90.*E3  +(45.*D+75.)
     *        *E2 +(90.*D2 -90.*D-15.)*E+9.*D3  +21.*D2 -73.*D-5.)*X2+(
     *        (165.*E3  +((153.*D2 -36.*D-74.)*E))*X1))*Y1))/180.
           CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1
     *        ))*Y3+((2.*E3  -(2.*E2 ))*X3+(((-E4  ))+E2 )*X2+((((-2.*E
     *        4  ))+(2.*E3  ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2.*E
     *        3  ))*X2+(4.*E4  *X1))*Y1))/16.)
           CF(2) = (-(((8.*D-4.)*E*X3+(((-2.*E3  ))+(((-6.*D))+3.)*E2 +
     *        ((((-4.*D))+5.)*E))*X2+((((-2.*E3  ))+((((-12.*D))+2.)*E*
     *        E))*X1))*Y3+((((-2.*E3  ))+(6.*D-3.)*E2 +((((-4.*D))+5.)*
     *        E))*X3+((((-4.*D))+4.)*E3  +((2.*D-4.)*E))*X2+((E4  +(((-
     *        8.*D))+3.)*E3  +((6.*D-4.)*E2 ))*X1))*Y2+(((2.*E3  +((((-
     *        12.*D))+2.)*E2 ))*X3+(E4  +(8.*D-3.)*E3  +((6.*D-4.)*E2 )
     *        )*X2+(16.*D*E3  *X1))*Y1))/16.)
           CF(3) = (((4.*E2 -4.*D2 +4.*D-1.)*X3+((6.*D-5.)*E2 +(6.*D2 -
     *        6.*D-3.)*E+2.*D2 -5.*D+2.)*X2+((((-2.*E3  ))+(6.*D-3.)*E*
     *        E+((12.*D2 -4.*D-4.)*E))*X1))*Y3+(((6.*D-5.)*E2 +(((-6.*D
     *        *D))+6.*D+3.)*E+2.*D2 -5.*D+2.)*X3+((6.*D2 -12.*D+4.)*E2
     *        -D2 +4.*D-4.)*X2+(((((-4.*D))+1.)*E3  +(12.*D2 -9.*D-3.)*
     *        E2 +((((-6.*D2 ))+8.*D+2.)*E))*X1))*Y2+(((((-2.*E3  ))+((
     *        (-6.*D))+3.)*E2 +((12.*D2 -4.*D-4.)*E))*X3+((((-4.*D))+1.
     *        )*E3  +(((-12.*D2 ))+9.*D+3.)*E2 +((((-6.*D2 ))+8.*D+2.)*
     *        E))*X2+((E4  +((((-24.*D2 ))+8.)*E2 ))*X1))*Y1))/16.
           CF(4) = (((8.*D-4.)*E*X3+((6.*D2 -10.*D+2.)*E+2.*D3  -3.*D2
     *        -3.*D+2.)*X2+(((((-6.*D))-1.)*E2 +(6.*D2 -6.*D-3.)*E+4.*D
     *        3  -2.*D2 -4.*D+2.)*X1))*Y3+(((6.*D2 -10.*D+2.)*E-2.*D3
     *        +3.*D2 +3.*D-2.)*X3+(4.*D3  -12.*D2 +(8.*D))*E*X2+(((((-6
     *        .*D2 ))+3.*D+2.)*E2 +(8.*D3  -9.*D2 -6.*D+2.)*E-2.*D3  +4
     *        .*D2 +2.*D-4.)*X1))*Y2+((((((-6.*D))-1.)*E2 +(((-6.*D2 ))
     *        +6.*D+3.)*E+4.*D3  -2.*D2 -4.*D+2.)*X3+((((-6.*D2 ))+3.*D
     *        +2.)*E2 +(((-8.*D3  ))+9.*D2 +6.*D-2.)*E-2.*D3  +4.*D2 +2
     *        .*D-4.)*X2+(((4.*D+2.)*E3  +((((-16.*D3  ))+(16.*D))*E))*
     *        X1))*Y1))/16.
           CF(5) = (((4.*D2 -4.*D+1.)*X3+(2.*D3  -5.*D2 +(2.*D))*X2+(((
     *        ((-6.*D2 ))-2.*D+1.)*E+2.*D3  -3.*D2 -3.*D+2.)*X1))*Y3+((
     *        2.*D3  -5.*D2 +(2.*D))*X3+(D4  -4.*D3  +(4.*D2 ))*X2+((((
     *        (-4.*D3  ))+3.*D2 +(4.*D))*E+2.*D4  -3.*D3  -3.*D2 +(2.*D
     *        ))*X1))*Y2+((((((-6.*D2 ))-2.*D+1.)*E-2.*D3  +3.*D2 +3.*D
     *        -2.)*X3+((((-4.*D3  ))+3.*D2 +(4.*D))*E-2.*D4  +3.*D3  +3
     *        .*D2 -(2.*D))*X2+(((6.*D2 +6.*D+1.)*E2 -4.*D4  +8.*D2 -4.
     *        )*X1))*Y1))/16.
           CF(6) = (-(D*(D+1.)*((((2.*D)-1.)*X1*Y3)+(D2 -(2.*D))*X1*Y2+
     *        (((2.*D-1.)*X3+(D2 -(2.*D))*X2+((((-4.*D))-2.)*E*X1))*Y1)
     *        ))/16.)
           CF(7) = (D2 *(D+1.)**2*X1*Y1)/16.
C     . END OF COMPUTED GO TO.
 350  CONTINUE
      SUML = CF(7)*(2.*(G6+5.*G4*F2+3.*G2*F4+F6/7.)*EL1 + 2./7.*DL2
     *        +2.*(F6+5.*F4*G2+3.*F2*G4+G6/7.)*ALOG3
     *        -(30.*T6+10.*T4+6.*T2)/105.)
     *    + CF(6)*(((F6+G6-1.)/3.+5.*F2*G2*(F2+G2))*DL1
     *        +2.*(F4+10.*F2*G2/3.+G4)*FL3 -(15.*T5+5.*T3+3.*T1)/45.)
     *    + CF(5)*(2.*(G4+2.*F2*G2+F4/5.)*EL1 +2./5.*DL2
     *        +2.*(F4+2.*F2*G2+G4/5.)*ALOG3 -(6.*T4+2.*T2)/15.)
     *    + CF(4)*(((F4+G4-1.)/2.+3.*F2*G2)*DL1 +2.*(F2+G2)*FL3
     *        -(3.*T3+T1)/6.)
     *    + CF(3)*(2./3.)*((F2+3.*G2)*EL1 +DL2 +(3.*F2+G2)*ALOG3 -T2)
     *    + CF(2)*((F2+G2-1.)*DL1 +2.*FL3 -T1)
     *    + CF(1)*2.*(EL1+DL2+ALOG3)
      XDNDN = -XDNDN/Z3 + SUML/Z2
      RETURN
      END
      OVERLAY(MAIN,10,0)
      PROGRAM QUAD82
C***********************************************************************
C
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE
C        ELEMENTS WITH NNE = 8.
C
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT
C        GROUP TRANSFORMATIONS.
C
C***********************************************************************
C
      DIMENSION LC(1),KC(1)
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,SKP(2),
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,ALG1,ALG2,ALG3,OTHERS(1)
      COMMON/TEMP/I1,I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,
     *   T1,T2,T3,T4,T5,T6,
     *   DL1,EL1,DL2,FL3,XJ
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))
C
C
      X1 = XX1
      X2 = XX2
      X3 = XX3
      Y1 = YY1
      Y2 = YY2
      Y3 = YY3
      Z1 = ZZ1
      Z2 = ZZ2
      Z3 = ZZ3
      ALOG1 = ALG1
      ALOG2 = ALG2
      ALOG3 = ALG3
      IL = IXC + 11
      IU = IXC + 36
      LL = 0
      DO 4 I=1,2
         DO 3 J=1,4
            LL = LL + 1
            D = Z2/Z3
            DP1 = D + 1.
            DM1 = D - 1.
            D2 = D*D
            D3 = D*D2
            E = -Z1/Z3
            EP1 = E + 1.
            EM1 = E - 1.
            E2 = E*E
            E3 = E*E2
            FF = Z1/Z2
            F2 = FF*FF
            F4 = F2*F2
            F6 = F2*F4
            G = Z3/Z2
            G2 = G*G
            G4 = G2*G2
            G6 = G2*G4
C           . T'L' = (Z2/Z3)*(((Z1+Z3)/Z2)**L - ((Z1-Z3)/Z2)**L)
            T1 = 2.
            T2 = 4.*FF
            T3 = 2.*G2 + 6.*F2
            T4 = 8.*FF*(G2+F2)
            T5 = 2.*G4 + 20.*G2*F2 + 10.*F4
            T6 = 2.*FF*T3*(3.*G2+F2)
            DL1 = D*ALOG1
            EL1 = -E*ALOG1
            DL2 = D*ALOG2
            FL3 = FF*ALOG3
```

```
                DO 2 I1=IL,IU
                   I2 = I1 + IC
                   I3 = I2 + IC
                   I4 = I3 + IC
                   K = KC(I1) - 3
                   L = LC(I2)
C                  IF (L.EQ.LL)
                                                IF(L.NE.LL) GO TO 1
C                  THEN
                      XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)
                      XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)
                      XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)
                      XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)
 1                 CONTINUE
 2              CONTINUE
C               . TRANSFORMATION OF TYPE ONE.
                X1 = -X1
                XJ =  X2
                X2 = -X3
                X3 =  XJ
                Y1 = -Y1
                XJ =  Y2
                Y2 = -Y3
                Y3 =  XJ
                XJ =  Z2
                Z2 =  Z3
                Z3 = -XJ
                ALOG1 = -ALOG1
                XJ    =  ALOG2
                ALOG2 = -ALOG3
                ALOG3 =  XJ
 3           CONTINUE
C            . TRANSFORMATION OF TYPE TWO.
             X3 = -X3
             Y3 = -Y3
             Z3 = -Z3
             ALOG1 = -ALOG1
             ALOG2 = -ALOG2
 4        CONTINUE
          END
```

```
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)
C**********************************************************************
C
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD82 TO EVALUATE
C        C-INTEGRALS.
C
C**********************************************************************
C
C
      COMMON/TEMP/I1,I2,I3,I4,K,L,XY(6),Z1,Z2,Z3,
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,
     *   T1,T2,T3,T4,T5,T6,
     *   DL1,EL1,DL2,FL3,SUML,CF(7)
C
      D2 = D*D
      D3 = D*D2
      D4 = D*D3
      E2 = E*E
      E3 = E*E2
      E4 = E*E3
      SUML = 0.
      GO TO (304,305,306,307,308),K
C           . 5,1
 304        XDNDN = (-(((240.*E-120.*D+60.)*X3+(((-90.*E2 ))+(60.*D-150.
     *         )*E-30.*D2 +90.*D-90.)*X2+((((-210.*E2 ))+(180.*D+30.)*E-
     *         78.*D2 -18.*D-10.)*X1))*Y3+((((-90.*E2 ))+(60.*D-15.)*E-3
     *         0.*D2 +15.*D+60.)*X3+(45.*E3  +45.*E2 +(45.*D2 -60.*D-30.
     *         )*E+15.*D2 -30.*D-30.)*X2+((90.*E3  +(((-45.*D))-15.)*E2
     *         +(90.*D2 -90.)*E-9.*D3  -9.*D2 +10.*D+10.)*X1))*Y2+(((((-
     *         210.*E2 ))+(180.*D-45.)*E-78.*D2 +39.*D-100.)*X3+(90.*E**
     *         3+(((-45.*D))+120.)*E2 +(90.*D2 -150.*D+60.)*E-9.*D3  +48
     *         .*D2 -80.*D+70.)*X2+((195.*E3  +(((-180.*D))-30.)*E2 +(20
     *         7.*D2 +18.*D+20.)*E-36.*D3  -18.*D2 -32.*D-10.)*X1))*Y1))
     *         /90.)
            CF(1) = ((4.*E2 *X3+(((-2.*E3  ))-(2.*E2 ))*X2-(4.*E3  *X1))
     *         *Y3+((((-2.*E3  ))+(2.*E))*X3+(E4  +E3  -E2 -E)*X2+((2.*E
     *         4  -(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  +(2.*E3
     *         ))*X2+(4.*E4  *X1))*Y1))/8.
            CF(2) = (-(((8.*E2 +((((-8.*D))+2.)*E))*X3+(((-2.*E3  ))+(6.
     *         *D-6.)*E2 +((4.*D-4.)*E))*X2+((((-6.*E3  ))+(12.*D*E2 ))*
     *         X1))*Y3+((((-2.*E3  ))+(6.*D-1.)*E2 +2.*E-2.*D+1.)*X3+(((
     *         (-4.*D))+2.)*E3  +(((-3.*D))+2.)*E2 +(2.*D-2.)*E+D-2.)*X2
     *         +((E4  -8.*D*E3  -E2 +(4.*D*E))*X1))*Y2+(((((-6.*E3  ))+(
     *         12.*D-2.)*E2 -(2.*E))*X3+(E4  +(((-8.*D))+5.)*E3  +(((-6.
     *         *D))+5.)*E2 +E)*X2+((4.*E4  -16.*D*E3  +(2.*E2 ))*X1))*Y1
     *         ))/8.)
            CF(3) = (((4.*E2 +(((-16.*D))+4.)*E+4.*D2 -(2.*D))*X3+((6.*D
     *         -4.)*E2 +(((-6.*D2 ))+12.*D-4.)*E-2.*D2 +(4.*D))*X2+((((-
     *         2.*E3  ))+(18.*D+2.)*E2 +((((-12.*D2 ))+4.)*E))*X1))*Y3+(
     *         ((6.*D-1.)*E2 +(((-6.*D2 ))+(2.*D))*E-2.*D+1.)*X3+((6.*D*
     *         D-(6.*D))*E2 +(3.*D2 -(4.*D))*E-D2 +(2.*D))*X2+(((((-4.*D
     *         ))-1.)*E3  +(12.*D2 -2.)*E2 +(2.*D+1.)*E-2.*D2 +2.)*X1))*
     *         Y2+(((((-2.*E3  ))+(18.*D-3.)*E2 +(((-12.*D2 ))+4.*D-2.)*
     *         E+2.*D-1.)*X3+((((-4.*D))+2.)*E3  +(12.*D2 -15.*D+2.)*E2
     *         +(6.*D2 -10.*D+2.)*E-D+2.)*X2+((E4  +(((-16.*D))-2.)*E3
     *         +(24.*D2 -3.)*E2 -(4.*D*E))*X1))*Y1))/8.
            CF(4) = ((((8.*D-2.)*E-8.*D2 +(4.*D))*X3+((6.*D2 -(8.*D))*E-
     *         2.*D3  +6.*D2 -(4.*D))*X2+(((((-6.*D))-2.)*E2 +(18.*D2 +4
     *         .*D-4.)*E-4.*D3  +(4.*D))*X1))*Y3+(((6.*D2 -(2.*D))*E-2.*
     *         D3  +D2 )*X3+((4.*D3  -(6.*D2 ))*E+D3  -(2.*D2 ))*X2+((((
     *         (-6.*D2 ))-(3.*D))*E2 +(8.*D3  -(4.*D))*E+D2 +D)*X1))*Y2+
     *         ((((((-6.*D))+1.)*E2 +(18.*D2 -(6.*D))*E-4.*D3  +2.*D2 -2
     *         .*D+1.)*X3+((((-6.*D2 ))+(6.*D))*E2 +(8.*D3  -15.*D2 +(4.
     *        *D))*E+2.*D3  -5.*D2 +(2.*D))*X2+(((4.*D+1.)*E3  +(((-24.
     *         *D2 ))-6.*D+2.)*E2 +(16.*D3  -6.*D+1.)*E-2.*D2 +2.)*X1))*
     *         Y1))/8.
```

```
      CF(5) = (D*((((((4.*D)-2.)*X3)+(2.*D2 -(4.*D))*X2+(((((-6.*D
     *   ))-4.)*E+6.*D2 +2.*D-4.)*X1))*Y3)+((2.*D2 -D)*X3+(D3  -(2
     *   .*D2 ))*X2+(((((-4.*D2 ))-(3.*D))*E+2.*D3  -(2.*D))*X1))*
     *   Y2+((((2-(6.*D))*E+6.*D2 -(3.*D))*X3+((6.*D-(4.*D2 ))*E+2
     *   .*D3  -5.*D2 +(2.*D))*X2+(((6.*D+3.)*E2 +(((-16.*D2 ))-6.
     *   *D+4.)*E+4.*D3  -3.*D+1.)*X1))*Y1)))/8.
      CF(6) = (-(D2 *((((2.*D)+2.)*X1*Y3)+(D2 +D)*X1*Y2+(((2.*D-1.
     *   )*X3+(D2 -(2.*D))*X2+(((((-4.*D))-3.)*E+4.*D2 +2.*D-2.)*X
     *   1))*Y1)))/8.)
      CF(7) =   ((D+1.)*D3  *X1*Y1)/8.
      GO TO 350
C     . 5,3
 305  XDNDN = (-((120.*E*X3+(90.*E2 +(((-60.*D))-30.)*E+30.*D2 -30
     *   .*D-30.)*X2+((((-150.*E2 ))+(60.*D-30.)*E-42.*D2 +18.*D+1
     *   0.)*X1))*Y3+((((-90.*E2 ))+(((-60.*D))+15.)*E-30.*D2 +15.
     *   *D+60.)*X3+(((-45.*E3  ))+45.*E2 +(((-45.*D2 ))+60.*D+30.
     *   )*E+15.*D2 -30.*D-30.)*X2+((90.*E3  +(45.*D+15.)*E2 +(90.
     *   *D2 -90.)*E+9.*D3  +9.*D2 -10.*D-10.)*X1))*Y2+(((((-150.*
     *   E2 ))+(((-60.*D))+15.)*E-42.*D2 +21.*D-20.)*X3+(((-90.*E*
     *   *3))+(45.*D+60.)*E2 +(((-90.*D2 ))+(90.*D))*E+9.*D3  +12.
     *   *D2 -40.*D-10.)*X2+((165.*E3  +30.*E2 +(153.*D2 -18.*D-20
     *   .)*E+18.*D2 -12.*D+10.)*X1))*Y1))/90.)
      CF(1) = ((4.*E2 *X3+(2.*E3  -(2.*E2 ))*X2-(4.*E3  *X1))*Y3+(
     *   (((-2.*E3  ))+(2.*E))*X3+(((-E4  ))+E3  +E2 -E)*X2+((2.*E
     *   4  -(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(((-2.*E4  ))+(2.
     *   *E3  ))*X2+(4.*E4  *X1))*Y1))/8.
      CF(2) = (((8.*D-2.)*E*X3+(((-2.*E3  ))+(6.*D-2.)*E2 +((((-4.
     *   *D))+4.)*E))*X2+((2.*E3  -(12.*D*E2 ))*X1))*Y3+((((-2.*E*
     *   *3))+(((-6.*D))+1.)*E2 +2.*E+2.*D-1.)*X3+((((-4.*D))+2.)*
     *   E3  +(3.*D-2.)*E2 +(2.*D-2.)*E-D+2.)*X2+((E4  +8.*D*E3  -
     *   E2 -(4.*D*E))*X1))*Y2+(((((-2.*E3  ))+(((-12.*D))+2.)*E2
     *   +(2.*E))*X3+(E4  +(((-8.*D))+3.)*E3  +(6.*D-3.)*E2 -E)*X2
     *   +((16.*D*E3  -(2.*E2 ))*X1))*Y1))/8.
      CF(3) = (-(((4.*E2 -4.*D2 +(2.*D))*X3+((6.*D-4.)*E2 +(((-6.*
     *   D2 ))+4.*D+4.)*E+2.*D2 -(4.*D))*X2+((((-2.*E3  ))+(((-6.*
     *   D))+2.)*E2 +((12.*D2 -4.)*E))*X1))*Y3+(((6.*D-1.)*E2 +(6.
     *   *D2 -(2.*D))*E-2.*D+1.)*X3+((6.*D2 -(6.*D))*E2 +(((-3.*D*
     *   D))+(4.*D))*E-D2 +(2.*D))*X2+(((((-4.*D))-1.)*E3  +(((-12
     *   .*D2 ))+2.)*E2 +(2.*D+1.)*E+2.*D2 -2.)*X1))*Y2+(((((-2.*E
     *   3  ))+(6.*D-1.)*E2 +(12.*D2 -4.*D-2.)*E-2.*D+1.)*X3+((((-
     *   4.*D))+2.)*E3  +(12.*D2 -9.*D-2.)*E2 +(((-6.*D2 ))+6.*D+2
     *   .)*E+D-2.)*X2+((E4  -2.*E3  +(((-24.*D2 ))+5.)*E2 +(4.*D*
     *   E))*X1))*Y1))/8.)
      CF(4) = (-(((8.*D-2.)*E*X3+((6.*D2 -(8.*D))*E-2.*D3  +2.*D2
     *   +(4.*D))*X2+(((((-6.*D))-2.)*E2 +(((-6.*D2 ))+4.*D+4.)*E+
     *   4.*D3  -(4.*D))*X1))*Y3+(((6.*D2 -(2.*D))*E+2.*D3  -D2 )*
     *   X3+((4.*D3  -(6.*D2 ))*E-D3  +(2.*D2 ))*X2+(((((-6.*D2 ))
     *   -(3.*D))*E2 +(((-8.*D3  ))+(4.*D))*E+D2 +D)*X1))*Y2+(((((
     *   (-6.*D))+1.)*E2 +(6.*D2 -(2.*D))*E+4.*D3  -2.*D2 -2.*D+1.
     *   )*X3+((((-6.*D2 ))+(6.*D))*E2 +(8.*D3  -9.*D2 -(4.*D))*E-
     *   2.*D3  +3.*D2 +(2.*D))*X2+(((4.*D+1.)*E3  +(((-6.*D))-2.)
     *   *E2 +(((-16.*D3  ))+10.*D+1.)*E+2.*D2 -2.)*X1))*Y1))/8.)
      CF(5) = (-(D*((((((4.*D)-2.)*X3)+(2.*D2 -(4.*D))*X2+(((((-6.
     *   *D))-4.)*E-2.*D2 +2.*D+4.)*X1))*Y3)+((2.*D2 -D)*X3+(D3  -
     *   (2.*D2 ))*X2+(((((-4.*D2 ))-(3.*D))*E-2.*D3  +(2.*D))*X1)
     *   )*Y2+((((2-(6.*D))*E+2.*D2 -D)*X3+((6.*D-(4.*D2 ))*E+2.*D
     *   3  -3.*D2 -(2.*D))*X2+(((6.*D+3.)*E2 +(((-6.*D))-4.)*E-4.
     *   *D3  +5.*D+1.)*X1))*Y1)))/8.)
      CF(6) = (D2 *((((2.*D)+2.)*X1*Y3)+(D2 +D)*X1*Y2+(((2.*D-1.)*
     *   X3+(D2 -(2.*D))*X2+(((((-4.*D))-3.)*E+2.*D+2.)*X1))*Y1)))
     *   /8.
      CF(7) =  (-((D+1.)*D3  *X1*Y1)/8.)
      GO TO 350
C     . 5,5
 306  XDNDN = (((240.*E-(120.*D))*X3+(((-90.*E2 ))+60.*D*E-30.*D2
     *   +60.)*X2+((((-210.*E2 ))+180.*D*E-78.*D2 -100.)*X1))*Y3+(
     *   (((-90.*E2 ))+60.*D*E-30.*D2 +60.)*X3+(45.*E3  +((45.*D2
     *   -75.)*E))*X2+((90.*E3  -45.*D*E2 +(90.*D2 -60.)*E-9.*D3
     *   -(5.*D))*X1))*Y2+(((((-210.*E2 ))+180.*D*E-78.*D2 -100.)*
```

```
    *      X3+(90.*E3  -45.*D*E2 +(90.*D2 -60.)*E-9.*D3  -(5.*D))*X2
    *      +((195.*E3  -180.*D*E2 +(207.*D2 +95.)*E-36.*D3  -(80.*D)
    *      )*X1))*Y1))/45.
        CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))+(2.*E))*X2-(4.*E3  *X1))
    *      *Y3+((((-2.*E3  ))+(2.*E))*X3+(E4  -2.*E2 +1.)*X2+((2.*E*
    *      *4-(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  -(2.*E2 ))
    *      *X2+(4.*E4  *X1))*Y1))/4.)
        CF(2) = (((8.*E2 -(8.*D*E))*X3+(((-2.*E3  ))+6.*D*E2 +2.*E-(
    *      2.*D))*X2+((((-6.*E3  ))+12.*D*E2 -(2.*E))*X1))*Y3+((((-2
    *      .*E3  ))+6.*D*E2 +2.*E-(2.*D))*X3+(((-4.*D*E3  ))+(4.*D*E
    *      ))*X2+((E4  -8.*D*E3  +4.*D*E-1.)*X1))*Y2+(((((-6.*E3  ))
    *      +12.*D*E2 -(2.*E))*X3+(E4  -8.*D*E3  +4.*D*E-1.)*X2+((4.*
    *      E4  -16.*D*E3  +(4.*E2 ))*X1))*Y1))/4.
        CF(3) = (-(((4.*E2 -16.*D*E+(4.*D2 ))*X3+(6.*D*E2 -6.*D2 *E-
    *      (2.*D))*X2+((((-2.*E3  ))+18.*D*E2 +(((-12.*D2 ))-2.)*E+(
    *      2.*D))*X1))*Y3+((6.*D*E2 -6.*D2 *E-(2.*D))*X3+(6.*D2 *E2
    *      -(2.*D2 ))*X2+((((-4.*D*E3  ))+12.*D2 *E2 -(2.*D2 ))*X1))
    *      *Y2+(((((-2.*E3  ))+18.*D*E2 +(((-12.*D2 ))-2.)*E+(2.*D))
    *      *X3+(((-4.*D*E3  ))+12.*D2 *E2 -(2.*D2 ))*X2+((E4  -16.*D
    *      *E3  +(24.*D2 +2.)*E2 -8.*D*E+1.)*X1))*Y1))/4.)
        CF(4) = (-(((4.*D*E-(4.*D2 ))*X3+(3.*D2 *E-D3  )*X2+((((-3.*
    *      D*E2 ))+9.*D2 *E-2.*D3  -D)*X1))*Y3+((3.*D2 *E-D3  )*X3+2
    *      .*D3  *E*X2+((((-3.*D2 *E2 ))+(4.*D3  *E))*X1))*Y2+(((((-
    *      3.*D*E2 ))+9.*D2 *E-2.*D3  -D)*X3+(((-3.*D2 *E2 ))+(4.*D*
    *      *3*E))*X2+((2.*D*E3  -12.*D2 *E2 +(8.*D3  +(2.*D))*E-(2.*
    *      D2 ))*X1))*Y1))/2.)
        CF(5) = (-(D2 *((((4.*X3)+2.*D*X2+((6.*D-(6.*E))*X1))*Y3)+(2
    *      .*D*X3+D2 *X2+((2.*D2 -(4.*D*E))*X1))*Y2+(((6.*D-(6.*E))*
    *      X3+(2.*D2 -(4.*D*E))*X2+((6.*E2 -16.*D*E+4.*D2 +2.)*X1))*
    *      Y1)))/4.)
        CF(6) =      (D3  *((2.*X1*Y3)+D*X1*Y2+((2.*X3+D*X2+((4.*D-(
    *      4.*E))*X1))*Y1)))/4.
        CF(7) =      (-(D4  *X1*Y1)/4.)
        GO TO 350
C       . 6,5
 307    XDNDN = ((60.*X3-30.*E*X2+((((-60.*E))+(6.*D))*X1))*Y3+((60.
    *      *E-60.*D-60.)*X3+(60.*D*E-(30.*D))*X2+((((-30.*E2 ))+(120
    *      .*D+30.)*E-18.*D2 -60.*D-40.)*X1))*Y2+(((((-120.*E))+48.*
    *      D+60.)*X3+(60.*E2 -30.*E+24.*D2 -40.)*X2+((120.*E2 +(((-4
    *      8.*D))-60.)*E+48.*D2 +18.*D+40.)*X1))*Y1))/45.
        CF(1) = (-((2.*E*X3+(((-E2 ))+1.)*X2-(2.*E2 *X1))*Y3+((((-2.
    *      *E2 *X3))+(E3  -E)*X2+(2.*E3  *X1))*Y1))/4.)
        CF(2) = (((2.*E-(2.*D))*X3+2.*D*E*X2+((((-E2 ))+4.*D*E-1.)*X
    *      1))*Y3+((4.*E2 -(4.*E))*X3+(((-2.*E3  ))+2.*E2 +2.*E-2.)*
    *      X2+((((-4.*E3  ))+(4.*E2 ))*X1))*Y2+(((((-2.*E2 ))+(4.*D*
    *      E))*X3+(((-3.*D*E2 ))+D)*X2+((E3  -6.*D*E2 +E)*X1))*Y1))/
    *      4.
        CF(3) = (((2.*E+(2.*D))*X3+(((-E2 ))+D2 +1.)*X2+((((-2.*E2 )
    *      )-2.*D*E+(2.*D2 ))*X1))*Y3+((((-4.*E2 ))+(8.*D+4.)*E-(4.*
    *      D))*X3+(((-6.*D*E2 ))+4.*D*E+(2.*D))*X2+((2.*E3  +(((-12.
    *      *D))-2.)*E2 +(8.*D+2.)*E-2.)*X1))*Y2+(((2.*E2 +(((-4.*D))
    *      -4.)*E+(2.*D2 ))*X3+(((-E3  ))+2.*E2 +(((-3.*D2 ))+1.)*E-
    *      2.)*X2+((((-2.*E3  ))+(3.*D+4.)*E2 -6.*D2 *E+D)*X1))*Y1))
    *      /4.
        CF(4) = (-(((2.*E-(2.*D))*X3+2.*D*E*X2+((((-E2 ))+4.*D*E+D2
    *      -1.)*X1))*Y3+((8.*D*E-4.*D2 -(4.*D))*X3+(6.*D2 *E-(2.*D2
    *      ))*X2+((((-6.*D*E2 ))+(12.*D2 +(4.*D))*E-4.*D2 -(2.*D))*X
    *      1))*Y2+(((2.*E2 +(((-4.*D))-4.)*E+2.*D2 +(4.*D))*X3+(3.*D
    *      *E2 -4.*D*E+D3  -D)*X2+((((-E3  ))+(6.*D+2.)*E2 +(((-3.*D
    *      *D))-8.*D-1.)*E+2.*D3  +2.)*X1))*Y1))/4.)
        CF(5) = (-(D*((((2.*X3)+D*X2+((2.*D-(2.*E))*X1))*Y3)+(4.*D*X
    *      3+2.*D2 *X2+((((-6.*D*E))+4.*D2 +(2.*D))*X1))*Y2+(((4.*E-
    *      2.*D-4.)*X3+(3.*D*E-(2.*D))*X2+((((-3.*E2 ))+(6.*D+4.)*E-
    *      D2 -4.*D-1.)*X1))*Y1)))/4.)
        CF(6) =      (D2 *((X1*Y3)+2.*D*X1*Y2+((((-2.*X3))-D*X2+((3.*E
    *      -2.*D-2.)*X1))*Y1)))/4.
        CF(7) =      (D3  *X1*Y1)/4.
        GO TO 350
```

```
C           . 7,5
 308        XDNDN = ((120.*E*X3+(((-90.*E2 ))-60.*D*E-30.*D2 +60.)*X2+((
     *          ((-150.*E2 ))-60.*D*E-42.*D2 -20.)*X1))*Y3+((90.*E2 -60.*
     *          D*E+30.*D2 -60.)*X3+(((-45.*E3  ))+((((-45.*D2 ))+75.)*E)
     *          )*X2+((((-90.*E3  ))+45.*D*E2 +(((-90.*D2 ))+60.)*E+9.*D*
     *          *3+(5.*D))*X1))*Y2+(((((-150.*E2 ))+60.*D*E-42.*D2 -20.)*
     *          X3+(90.*E3  +45.*D*E2 +(90.*D2 -60.)*E+9.*D3  +(5.*D))*X2
     *          +((165.*E3  +((153.*D2 +25.)*E))*X1))*Y1))/45.
            CF(1) = (-((4.*E2 *X3+(((-2.*E3  ))+(2.*E))*X2-(4.*E3  *X1))
     *          *Y3+((2.*E3  -(2.*E))*X3+(((-E4  ))+2.*E2 -1.)*X2+((((-2.
     *          *E4  ))+(2.*E2 ))*X1))*Y2+((((-4.*E3  *X3))+(2.*E4  -(2.*
     *          E2 ))*X2+(4.*E4  *X1))*Y1))/4.)
            CF(2) = (-((8.*D*E*X3+(((-2.*E3  ))-6.*D*E2 +2.*E+(2.*D))*X2
     *          +((((-2.*E3  ))-12.*D*E2 +(2.*E))*X1))*Y3+((((-2.*E3  ))+
     *          6.*D*E2 +2.*E-(2.*D))*X3+(((-4.*D*E3  ))+(4.*D*E))*X2+((E
     *          4  -8.*D*E3  +4.*D*E-1.)*X1))*Y2+(((2.*E3  -12.*D*E2 -(2.
     *          *E))*X3+(E4  +8.*D*E3  -4.*D*E-1.)*X2+(16.*D*E3  *X1))*Y1
     *          ))/4.)
            CF(3) = (((4.*E2 -(4.*D2 ))*X3+(6.*D*E2 +6.*D2 *E-(2.*D))*X2
     *          +((((-2.*E3  ))+6.*D*E2 +(12.*D2 -2.)*E-(2.*D))*X1))*Y3+(
     *          (6.*D*E2 -6.*D2 *E-(2.*D))*X3+(6.*D2 *E2 -(2.*D2 ))*X2+((
     *          ((-4.*D*E3  ))+12.*D2 *E2 -(2.*D2 ))*X1))*Y2+(((((-2.*E**
     *          3))-6.*D*E2 +(12.*D2 -2.)*E+(2.*D))*X3+(((-4.*D*E3  ))-12
     *          .*D2 *E2 +(2.*D2 ))*X2+((E4  +(((-24.*D2 ))+2.)*E2 +1.)*X
     *          1))*Y1))/4.
            CF(4) = ((4.*D*E*X3+(3.*D2 *E+D3  )*X2+((((-3.*D*E2 ))+3.*D*
     *          D*E+2.*D3  -D)*X1))*Y3+((3.*D2 *E-D3  )*X3+2.*D3  *E*X2+(
     *          (((-3.*D2 *E2 ))+(4.*D3  *E))*X1))*Y2+(((((-3.*D*E2 ))-3.
     *          *D2 *E+2.*D3  -D)*X3+(((-3.*D2 *E2 ))-(4.*D3  *E))*X2+((2
     *          .*D*E3  +((((-8.*D3  ))+(2.*D))*E))*X1))*Y1))/2.
            CF(5) = (D2 *((((4.*X3)+2.*D*X2+((2.*D-(6.*E))*X1))*Y3)+(2.*
     *          D*X3+D2 *X2+((2.*D2 -(4.*D*E))*X1))*Y2+(((((-6.*E))-(2.*D
     *          ))*X3+(((-4.*D*E))-(2.*D2 ))*X2+((6.*E2 -4.*D2 +2.)*X1))*
     *          Y1)))/4.
            CF(6) = (-(D3  *((2.*X1*Y3)+D*X1*Y2+((2.*X3+D*X2-(4.*E*X1))*
     *          Y1)))/4.)
            CF(7) =      (D4  *X1*Y1)/4.
C     . END OF COMPUTED GO TO.
 350  CONTINUE
      SUML = CF(7)*(2.*(G6+5.*G4*F2+3.*G2*F4+F6/7.)*EL1 + 2./7.*DL2
     *      +2.*(F6+5.*F4*G2+3.*F2*G4+G6/7.)*ALOG3
     *      -(30.*T6+10.*T4+6.*T2)/105.)
     *   + CF(6)*(((F6+G6-1.)/3.+5.*F2*G2*(F2+G2))*DL1
     *      +2.*(F4+10.*F2*G2/3.+G4)*FL3 -(15.*T5+5.*T3+3.*T1)/45.)
     *   + CF(5)*(2.*(G4+2.*F2*G2+F4/5.)*EL1 +2./5.*DL2
     *      +2.*(F4+2.*F2*G2+G4/5.)*ALOG3 -(6.*T4+2.*T2)/15.)
     *   + CF(4)*(((F4+G4-1.)/2.+3.*F2*G2)*DL1 +2.*(F2+G2)*FL3
     *      -(3.*T3+T1)/6.)
     *   + CF(3)*(2./3.)*((F2+3.*G2)*EL1 +DL2 +(3.*F2+G2)*ALOG3 -T2)
     *   + CF(2)*((F2+G2-1.)*DL1 +2.*FL3 -T1)
     *   + CF(1)*2.*(EL1+DL2+ALOG3)
      XDNDN = -XDNDN/Z3 + SUML/Z2
      RETURN
      END
      OVERLAY(MAIN,11,0)
      PROGRAM QUAD9
C***********************************************************************
C
C     THIS PROGRAM EVALUATES C-INTEGRALS FOR NONTRAPEZOIDAL FINITE
C        ELEMENTS WITH NSF = 9.
C
C     THE DO LOOPS ENDING AT STATEMENTS NUMBERED 3 AND 4  CARRY OUT
C        GROUP TRANSFORMATIONS.
C
C***********************************************************************
C
      DIMENSION LC(1),KC(1)
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,SKP(2),
     *   XX1,XX2,XX3,YY1,YY2,YY3,ZZ1,ZZ2,ZZ3,ALG1,ALG2,ALG3,OTHERS(1)
      COMMON/TEMP/I1,I2,I3,I4,K,L,X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,
     *   T1,T2,T3,T4,T5,T6,
     *   DL1,EL1,DL2,FL3,XJ
      EQUIVALENCE (KC(1),XC(1)),(LC(1),XC(1))
C
C
      X1 = XX1
      X2 = XX2
      X3 = XX3
      Y1 = YY1
      Y2 = YY2
      Y3 = YY3
      Z1 = ZZ1
      Z2 = ZZ2
      Z3 = ZZ3
      ALOG1 = ALG1
      ALOG2 = ALG2
      ALOG3 = ALG3
      IL = IXC + 37
      IU = IXC + 45
      LL = 0
      DO 4 I=1,2
         DO 3 J=1,4
            LL = LL + 1
            D = Z2/Z3
            DP1 = D + 1.
            DM1 = D - 1.
            D2 = D*D
            D3 = D*D2
            E = -Z1/Z3
            EP1 = E + 1.
            EM1 = E - 1.
            E2 = E*E
            E3 = E*E2
            FF = Z1/Z2
            F2 = FF*FF
            F4 = F2*F2
            F6 = F2*F4
            G = Z3/Z2
            G2 = G*G
            G4 = G2*G2
            G6 = G2*G4
C           . T'L' = (Z2/Z3)*(((Z1+Z3)/Z2)**L - ((Z1-Z3)/Z2)**L)
            T1 = 2.
            T2 = 4.*FF
            T3 = 2.*G2 + 6.*F2
            T4 = 8.*FF*(G2+F2)
            T5 = 2.*G4 + 20.*G2*F2 + 10.*F4
            T6 = 2.*FF*T3*(3.*G2+F2)
            DL1 = D*ALOG1
            EL1 = -E*ALOG1
```

```
        DL2 = D*ALOG2
        FL3 = FF*ALOG3
        DO 2 I1=IL,IU
           I2 = I1 + IC
           I3 = I2 + IC
           I4 = I3 + IC
           K = KC(I1) - 8
           L = LC(I2)
C          IF (L.EQ.LL)
                                          IF(L.NE.LL) GO TO 1
C          THEN
              XC(I1) = -XDNDN(Y1,Y2,Y3,Y1,Y2,Y3)
              XC(I2) =  XDNDN(X1,X2,X3,Y1,Y2,Y3)
              XC(I3) =  XDNDN(Y1,Y2,Y3,X1,X2,X3)
              XC(I4) = -XDNDN(X1,X2,X3,X1,X2,X3)
 1         CONTINUE
 2      CONTINUE
C       . TRANSFORMATION OF TYPE ONE.
        X1 = -X1
        XJ =  X2
        X2 = -X3
        X3 =  XJ
        Y1 = -Y1
        XJ =  Y2
        Y2 = -Y3
        Y3 =  XJ
        XJ =  Z2
        Z2 =  Z3
        Z3 = -XJ
        ALOG1 = -ALOG1
        XJ    =  ALOG2
        ALOG2 = -ALOG3
        ALOG3 =  XJ
 3     CONTINUE
C      . TRANSFORMATION OF TYPE TWO.
       X3 = -X3
       Y3 = -Y3
       Z3 = -Z3
       ALOG1 = -ALOG1
       ALOG2 = -ALOG2
 4    CONTINUE
      END
      FUNCTION XDNDN(X1,X2,X3,Y1,Y2,Y3)
C*********************************************************************
C
C     THIS SUBROUTINE IS CALLED BY THE PROGRAM QUAD9 TO EVALUATE
C        C-INTEGRALS.
C
C*********************************************************************
C
      COMMON/TEMP/I1,I2,I3,I4,K,L,XY(6),Z1,Z2,Z3,
     *   ALOG1,ALOG2,ALOG3,IL,IU,LL,I,J,D,DP1,DM1,D2,D3,
     *   E,EP1,EM1,E2,E3,FF,F2,F4,F6,G,G2,G4,G6,
     *   T1,T2,T3,T4,T5,T6,
     *   DL1,EL1,DL2,FL3,SUML,CF(7)
C
      D2 = D*D
      D3 = D*D2
      D4 = D*D3
      E2 = E*E
      E3 = E*E2
      E4 = E*E3
      SUML = 0.
      GO TO (309,310,311),K
C             . 9,1
 309          XDNDN = (-(((120.*E-24.*D+12.)*X3+(((-60.*E2 ))-60.*E-12.*D*
     *           D+24.*D-20.)*X2+((((-120.*E2 ))+(24.*D+12.)*E-24.*D2 -16.
     *           )*X1))*Y3+((60.*E2 +(((-120.*D))+30.)*E+36.*D2 -18.*D-40.
     *           )*X3+((90.*D-60.)*E2 +(60.*D-60.)*E+18.*D3  -36.*D2 -20.*
     *           D+40.)*X2+((((-30.*E3  ))+180.*D*E2 +(((-54.*D2 ))-36.*D+
     *           20.)*E+36.*D3  -(76.*D))*X1))*Y2+(((((-180.*E2 ))+(120.*D
     *           -30.)*E-60.*D2 +(30.*D))*X3+(90.*E3  +90.*E2 +(90.*D2 -(1
     *           20.*D))*E+30.*D2 -(60.*D))*X2+((180.*E3  +(((-90.*D))-30.
     *           )*E2 +(180.*D2 -60.)*E-18.*D3  -18.*D2 +8.*D+8.)*X1))*Y1)
     *           )/45.)
              CF(1) = ((2.*E2 *X3+(((-E3  ))-E2 )*X2-(2.*E3  *X1))*Y3+((((
     *           -2.*E3  *X3))+(E4  +E3  )*X2+(2.*E4  *X1))*Y1))/2.
              CF(2) = (-(((2.*E2 +((((-4.*D))+1.)*E))*X3+((3.*D-2.)*E2 +((
     *           2.*D-2.)*E))*X2+((((-E3  ))+(6.*D*E2 ))*X1))*Y3+((2.*E3
     *           -(2.*E))*X3+(((-E4  ))-E3  +E2 +E)*X2+((((-2.*E4  ))+(2.*
     *           E2 ))*X1))*Y2+(((((-2.*E3  ))+((6.*D-1.)*E2 ))*X3+((((-4.
     *           *D))+2.)*E3  +((((-3.*D))+2.)*E2 ))*X2+((E4  -(8.*D*E3  )
     *           )*X1))*Y1))/2.)
              CF(3) = (-(((2.*E2 +(4.*D-1.)*E-2.*D2 +D)*X3+(((-E3  ))-E2 +
     *           (3.*D2 -(4.*D))*E+D2 -(2.*D))*X2+((((-2.*E3  ))+(((-3.*D)
     *           )-1.)*E2 +((6.*D2 -2.)*E))*X1))*Y3+((((-2.*E3  ))+(6.*D-1
     *           .)*E2 +2.*E-2.*D+1.)*X3+((((-4.*D))+2.)*E3  +(((-3.*D))+2
     *           .)*E2 +(2.*D-2.)*E+D-2.)*X2+((E4  -8.*D*E3  -E2 +(4.*D*E)
     *           )*X1))*Y2+((((((-6.*D))+1.)*E2 +((6.*D2 -2.*D-2.)*E))*X3+
     *           ((((-6.*D2 ))+6.*D+1.)*E2 +((((-3.*D2 ))+4.*D+1.)*E))*X2+
     *           (((4.*D+1.)*E3  +((((-12.*D2 ))+4.)*E2 ))*X1))*Y1))/2.)
              CF(4) = (((2.*E2 +(((-4.*D))+1.)*E-2.*D2 +D)*X3+((3.*D-2.)*E
     *           *E+(2.*D-2.)*E-D3  +(2.*D2 ))*X2+((((-E3  ))+6.*D*E2 +(3.
     *           *D2 +(2.*D))*E-2.*D3  +(2.*D))*X1))*Y3+(((6.*D-1.)*E2 +((
     *           (-6.*D2 ))+(2.*D))*E-2.*D+1.)*X3+((6.*D2 -(6.*D))*E2 +(3.
     *           *D2 -(4.*D))*E-D2 +(2.*D))*X2+(((((-4.*D))-1.)*E3  +(12.*
     *           D2 -2.)*E2 +(2.*D+1.)*E-2.*D2 +2.)*X1))*Y2+((((6.*D2 -2.*
     *           D-2.)*E-2.*D3  +D2 +2.*D-1.)*X3+((4.*D3  -6.*D2 -2.*D+2.)
     *           *E+D3  -2.*D2 -D+2.)*X2+(((((-6.*D2 ))-3.*D+1.)*E2 +((8.*
     *           D3  -(8.*D))*E))*X1))*Y1))/2.
              CF(5) = ((((4.*D-1.)*E-2.*D2 +D)*X3+((3.*D2 -(4.*D))*E+D2 -(
     *           2.*D))*X2+(((((-3.*D))-1.)*E2 +(6.*D2 -2.)*E+D3  +D2 )*X1
     *           ))*Y3+(((6.*D2 -(2.*D))*E-2.*D3  +D2 )*X3+((4.*D3  -(6.*D
     *           *D))*E+D3  -(2.*D2 ))*X2+(((((-6.*D2 ))-(3.*D))*E2 +(8.*D
     *           3  -(4.*D))*E+D2 +D)*X1))*Y2+(((2.*D3  -D2 -2.*D+1.)*X3+(
     *           D4  -2.*D3  -D2 +(2.*D))*X2+(((((-4.*D3  ))-3.*D2 +2.*D+1
     *           .)*E+2.*D4  -4.*D2 +2.)*X1))*Y1))/2.
              CF(6) = (D*((((((2.*D)-1.)*X3)+(D2 -(2.*D))*X2+(((((-3.*D))-
     *           2.)*E+2.*D2 -2.)*X1))*Y3)+((2.*D2 -D)*X3+(D3  -(2.*D2 ))*
     *           X2+(((((-4.*D2 ))-(3.*D))*E+2.*D3  -(2.*D))*X1))*Y2+((((-
     *           D3  ))-D2 +D+1.)*X1*Y1)))/2.
              CF(7) =     (-((D+1.)*D2 *X1*(Y3+(D*Y2)))/2.)
              GO TO 350
```

```
C           . 9,5
  310       XDNDN = (((240.*E-(48.*D))*X3+(((-120.*E2 ))-24.*D2 +80.)*X2
     *         +((((-240.*E2 ))+48.*D*E-48.*D2 -80.)*X1))*Y3+((120.*E2 -
     *         240.*D*E+72.*D2 -80.)*X3+(180.*D*E2 +36.*D3  -(100.*D))*X
     *         2+((((-60.*E3  ))+360.*D*E2 +(((-108.*D2 ))-20.)*E+72.*D*
     *         *3-(80.*D))*X1))*Y2+(((((-360.*E2 ))+240.*D*E-(120.*D2 ))
     *         *X3+(180.*E3  +((180.*D2 -180.)*E))*X2+((360.*E3  -180.*D
     *         *E2 +360.*D2 *E-36.*D3  -(44.*D))*X1))*Y1))/45.
            CF(1) = (((-2.*E2 *X3))+(E3  -E)*X2+(2.*E3  *X1))*Y3+(2.*E**
     *         3*X3+(((-E4  ))+E2 )*X2-(2.*E4  *X1))*Y1
            CF(2) = ((2.*E2 -(4.*D*E))*X3+(3.*D*E2 -D)*X2+((((-E3  ))+6.
     *         *D*E2 -E)*X1))*Y3+((2.*E3  -(2.*E))*X3+(((-E4  ))+2.*E2 -
     *         1.)*X2+((((-2.*E4  ))+(2.*E2 ))*X1))*Y2+((((-2.*E3  ))+(6
     *         .*D*E2 ))*X3+(((-4.*D*E3  ))+(2.*D*E))*X2+((E4  -8.*D*E**
     *         3+E2 )*X1))*Y1
            CF(3) = ((2.*E2 +4.*D*E-(2.*D2 ))*X3+(((-E3  ))+((3.*D2 +1.)
     *         *E))*X2+((((-2.*E3  ))-3.*D*E2 +6.*D2 *E-D)*X1))*Y3+((((-
     *         2.*E3  ))+6.*D*E2 +2.*E-(2.*D))*X3+(((-4.*D*E3  ))+(4.*D*
     *         E))*X2+((E4  -8.*D*E3  +4.*D*E-1.)*X1))*Y2+((((-6.*D*E2 )
     *         )+((6.*D2 -2.)*E))*X3+((((-6.*D2 ))+1.)*E2 +D2 -1.)*X2+((
     *         4.*D*E3  +(((-12.*D2 ))+2.)*E2 +(2.*D*E))*X1))*Y1
            CF(4) = ((((-2.*E2 ))+4.*D*E+(2.*D2 ))*X3+(((-3.*D*E2 ))+D**
     *         3+D)*X2+((E3  -6.*D*E2 +(((-3.*D2 ))+1.)*E+(2.*D3  ))*X1)
     *         )*Y3+((((-6.*D*E2 ))+6.*D2 *E+(2.*D))*X3+(((-6.*D2 *E2 ))
     *         +(2.*D2 ))*X2+((4.*D*E3  -12.*D2 *E2 +(2.*D2 ))*X1))*Y2+(
     *         ((((-6.*D2 ))+2.)*E+2.*D3  -(2.*D))*X3+(((-4.*D3  ))+(2.*
     *         D))*E*X2+(((6.*D2 -1.)*E2 +(((-8.*D3  ))+(4.*D))*E+D2 -1.
     *         )*X1))*Y1
            CF(5) = (-D*((((((4.*E)-(2.*D))*X3)+3.*D*E*X2+((((-3.*E2 ))+
     *         6.*D*E+D2 -1.)*X1))*Y3)+((6.*D*E-(2.*D2 ))*X3+4.*D2 *E*X2
     *         +((8.*D2 *E-(6.*D*E2 ))*X1))*Y2+((2.*D2 -2.)*X3+(D3  -D)*
     *         X2+(((2-(4.*D2 ))*E+2.*D3  -(2.*D))*X1))*Y1))
            CF(6) = (-D2 *((((2.*X3)+D*X2+((2.*D-(3.*E))*X1))*Y3)+(2.*D*
     *         X3+D2 *X2+((2.*D2 -(4.*D*E))*X1))*Y2+(1-D2 )*X1*Y1))
            CF(7) =      D3  *X1*(Y3+D*Y2)
            GO TO 350
C           . 9,9
  311       XDNDN = ((384.*E*X3-192.*D*E*X2+((((-480.*E2 ))-96.*D2 -64.)
     *         *X1))*Y3+(((-192.*D*E*X3))+(240.*E3  +((432.*D2 -400.)*E)
     *         )*X2+((720.*D*E2 +144.*D3  -(304.*D))*X1))*Y2+(((((-480.*
     *         E2 ))-96.*D2 -64.)*X3+(720.*D*E2 +144.*D3  -(304.*D))*X2+
     *         ((720.*E3  +((720.*D2 -240.)*E))*X1))*Y1))/45.
            CF(1) = (((-4.*E2 *X3))+(4.*E3  *X1))*Y3+(4.*E3  *X3-(4.*E**
     *         4*X1))*Y1
            CF(2) = (((-8.*D*E*X3))+(4.*E3  -(4.*E))*X2+(12.*D*E2 *X1))*
     *         Y3+((4.*E3  -(4.*E))*X3+((((-4.*E4  ))+(4.*E2 ))*X1))*Y2+
     *         (12.*D*E2 *X3+(((-4.*E4  ))+(4.*E2 ))*X2-(16.*D*E3  *X1))
     *         *Y1
            CF(3) = ((8.*E2 -(4.*D2 ))*X3+(12.*D*E2 -(4.*D))*X2+((((-4.*
     *         E3  ))+((12.*D2 -4.)*E))*X1))*Y3+((12.*D*E2 -(4.*D))*X3+(
     *         ((-4.*E4  ))+8.*E2 -4.)*X2+((((-16.*D*E3  ))+(8.*D*E))*X1
     *         ))*Y2+((((-4.*E3  ))+((12.*D2 -4.)*E))*X3+(((-16.*D*E3  )
     *         )+(8.*D*E))*X2+((((-24.*D2 ))+8.)*E2 *X1))*Y1
            CF(4) = (16.*D*E*X3+(((-4.*E3  ))+((12.*D2 +4.)*E))*X2+((((-
     *         12.*D*E2 ))+4.*D3  -(4.*D))*X1))*Y3+((((-4.*E3  ))+((12.*
     *         D2 +4.)*E))*X3+(((-16.*D*E3  ))+(16.*D*E))*X2+(((((-24.*D
     *         *D))+4.)*E2 +4.*D2 -4.)*X1))*Y2+((((-12.*D*E2 ))+4.*D3  -
     *         (4.*D))*X3+((((-24.*D2 ))+4.)*E2 +4.*D2 -4.)*X2+((((-16.*
     *         D3  ))+(16.*D))*E*X1))*Y1
            CF(5) = (-4.*(((((E2 -(2.*D2 ))*X3)+(3.*D*E2 -D3  -D)*X2+((3
     *         .*D2 -1.)*E*X1))*Y3)+((3.*D*E2 -D3  -D)*X3+(6.*D2 *E2 -(2
     *         .*D2 ))*X2+((4.*D3  -(2.*D))*E*X1))*Y2+((3.*D2 -1.)*E*X3+
     *         (4.*D3  -(2.*D))*E*X2+((D4  -2.*D2 +1.)*X1))*Y1))
            CF(6) = (-4.*D*((((2.*E*X3)+3.*D*E*X2+((D2 -1.)*X1))*Y3)+(3.
     *         *D*E*X3+4.*D2 *E*X2+((D3  -D)*X1))*Y2+((D2 -1.)*X3+((D3
     *         -D)*X2))*Y1))
            CF(7) =      (-4.*D2 *(X3+(D*X2))*(Y3+D*Y2))
C     . END OF COMPUTED GO TO.
```

```
  350 CONTINUE
      SUML = CF(7)*(2.*(G6+5.*G4*F2+3.*G2*F4+F6/7.)*EL1 + 2./7.*DL2
     *      +2.*(F6+5.*F4*G2+3.*F2*G4+G6/7.)*ALOG3
     *      -(30.*T6+10.*T4+6.*T2)/105.)
     *   + CF(6)*(((F6+G6-1.)/3.+5.*F2*G2*(F2+G2))*DL1
     *      +2.*(F4+10.*F2*G2/3.+G4)*FL3 -(15.*T5+5.*T3+3.*T1)/45.)
     *   + CF(5)*(2.*(G4+2.*F2*G2+F4/5.)*EL1 +2./5.*DL2
     *      +2.*(F4+2.*F2*G2+G4/5.)*ALOG3 -(6.*T4+2.*T2)/15.)
     *   + CF(4)*(((F4+G4-1.)/2.+3.*F2*G2)*DL1 +2.*(F2+G2)*FL3
     *      -(3.*T3+T1)/6.)
     *   + CF(3)*(2./3.)*((F2+3.*G2)*EL1 +DL2 +(3.*F2+G2)*ALOG3 -T2)
     *   + CF(2)*((F2+G2-1.)*DL1 +2.*FL3 -T1)
     *   + CF(1)*2.*(EL1+DL2+ALOG3)
      XDNDN = -XDNDN/Z3 + SUML/Z2
      RETURN
      END
      OVERLAY(MAIN,12,0)
      PROGRAM SGPM
      LOGICAL SMFLAG,SPFLAG
      COMMON/SPACE/SPACE(27),SMFLAG,SPFLAG,OTHERS(1)
C
C
      CALL LINSTF
      IF (SPFLAG)
C     THEN
     *   CALL LODVEC
C     CONTINUE
      IF (SMFLAG)
C     THEN
     *   CALL MASS
C     CONTINUE
      END
      SUBROUTINE LINSTF
C*********************************************************************
C
C     THIS SUBROUTINE CALCULATES THE LINEAR STIFFNESS MATRIX  SS
C        AND THE GEOMETRIC STIFFNESS MATRIX  SG  FOR AN ISOLATED
C        DOUBLY-CURVED SHALLOW SHELL ELEMENT.
C
C     THE NUMBER OF DEGREES OF FREEDOM PER NODE IS FIVE.
C     NSF  IS THE NUMBER OF SHAPE FUNCTIONS PER ELEMENT.
C     NNE  IS THE NUMBER OF *NODAL* SHAPE FUNCTIONS PER ELEMENT.
C     THE QUANTITIES STORED IN POSITIONS 1 THRU 21 OF COMMON ARE THE
C        MATERIAL STIFFNESS COEFFICIENTS.  THE FIRST SIX OF THESE
C        QUANTITIES ARE THE EXTENSIONAL STIFFNESSES OF THE SHELL, THE
C        NEXT SIX ARE THE STIFFNESS INTERACTION COEFFICIENTS, THE NEXT
C        SIX ARE THE BENDING STIFFNESSES, AND THE LAST THREE ARE THE
C        TRANSVERSE SHEAR STIFFNESSES.
C     EN1,EN2,EN12  ARE THE PRESTRESS COEFFICIENTS.
C     USE  CURVE = .TRUE.  IF THE CURVATURE IS NONZERO.
C     USE  CURVE = .FALSE. IF THE CURVATURE TERMS ARE TO BE IGNORED.
C     USE SGFLAG = .TRUE.  IF THE GEOMETRIC STIFFNESS MATRIX  SG  IS TO
C        BE COMPUTED.  IN THIS CASE THE DIMENSIONS OF  SG  MUST BE
C     USE  SGFLAG = .FALSE. IF SG IS NOT TO BE COMPUTED.
C     Q1,Q2,Q12  ARE THE CURVATURES AT THE NODES.
C     NDFE IS THE NUMBER OF DEGREES OF FREEDOM PER ELEMENT  (NDFE=5*NSF)
C     THE ARRAY  SS  IS TO BE STORED IN POSITIONS  ISS+1  THRU
C        ISS+NDFE*NDFE  OF COMMON.
C     THE ARRAY  SG  IS TO BE STORED IN POSITIONS  ISG+1  THRU
C        ISG+NSF*NSF  OF COMMON.
C     THE INTEGRALS XONDN ARE STORED IN POSITIONS  IXC+1  THRU  IXC+4*IC
C        OF COMMON.
C     THE INTEGRALS XNNDN ARE STORED IN POSITIONS  IXB+1  THRU  IXB+2*IB
C        OF COMMON.
C     THE INTEGRALS XNNNN ARE STORED IN POSITIONS  IXA+1  THRU  IXA+IA
C        OF COMMON.
C     SS  SHOULD BE THOUGHT OF AS HAVING FOUR INDICES,
C        I.E. SS = SS(J1,K1,J2,K2), (J1,J2=1 THRU 5), (K1,K2=1 THRU NSF)
C
C*********************************************************************
C
      LOGICAL SGFLAG,CURVE
      DIMENSION SG(1),SS(1)
      COMMON/SPACE/C11,C12,C16,C22,C26,C66,
     *  F11,F12,F16,F22,F26,F66,D11,D12,D16,D22,D26,D66,C55,C44,C54,
     *  EN1,EN2,EN12,NSF,CURVE,SGFLAG,DUM(13),
     *  Q1(10),Q2(10),Q12(10),PRESS(30),
     *  DUMMY(14),NNE,ISS,ISG,OTHERS(1)
      COMMON/TEMP/SKIP(6),
     *              CURC1,CURC2,CURC6,CURF1,CURF2,CURF6,J,J1,J2,JON,J1N,
     *  J2N,J3N,J4N,J5N,K1,K1N,K1M,K1P1,K2,K2G,K2M,K2N,K3,K3M1,K4,NDFE,
     *  NDFE5,NSFM1,
     *  QU1,QU2,QU12,XNN,XN1N,XN2N,XNN1N,XNN2N,X1N1N,X1N2N,X2N1N,X2N2N
     *  ,DIAG,OFFD,Q21(10),QU21
      EQUIVALENCE (SS(1),C11),(SG(1),C11)
C
      CONDN(A,B,C,D) = A*X1N1N + B*X1N2N + C*X2N1N + D*X2N2N
      CURV(A,B,C) = A*QU1 + B*QU2 + C*QU21
C
C
C     IF (CURVE)
                                                IF(.NOT.CURVE)GOTO 10
C     THEN
         DO 1 J=1,NNE
            Q21(J) = 2.*Q12(J)
 1       CONTINUE
 10   CONTINUE
      NDFE = 5*NSF
      NDFE5 = 5*NDFE
      JON = ISS - 5*(NDFE+1)
      DO 8 K1=1,NSF
         JON = JON + 5
```

```
        J1N = JON
        K2G = ISG - NSF
        DO 8 K2=1,K1
           J1N = J1N + NDFE5
           J2N = J1N + NDFE
           J3N = J2N + NDFE
           J4N = J3N + NDFE
           J5N = J4N + NDFE
           K2G = K2G + NSF
           X1N1N = XDNDN(1,K1,1,K2)
           X1N2N = XDNDN(1,K1,2,K2)
           X2N1N = XDNDN(2,K1,1,K2)
           X2N2N = XDNDN(2,K1,2,K2)
           SS(1+J1N) = CDNDN(C11,C16,C16,C66)
           SS(1+J2N) = CDNDN(C16,C12,C66,C26)
           SS(2+J1N) = CDNDN(C16,C66,C12,C26)
           SS(2+J2N) = CDNDN(C66,C26,C26,C22)
           SS(1+J4N) = CDNDN(F11,F16,F16,F66)
           SS(1+J5N) = CDNDN(F16,F12,F66,F26)
           SS(2+J4N) = CDNDN(F16,F66,F12,F26)
           SS(2+J5N) = CDNDN(F66,F26,F26,F22)
           SS(4+J1N) = CDNDN(F11,F16,F16,F66)
           SS(4+J2N) = CDNDN(F16,F12,F66,F26)
           SS(5+J1N) = CDNDN(F16,F66,F12,F26)
           SS(5+J2N) = CDNDN(F66,F26,F26,F22)
           SS(4+J4N) = CDNDN(D11,D16,D16,D66)
           SS(4+J5N) = CDNDN(D16,D12,D66,D26)
           SS(5+J4N) = CDNDN(D16,D66,D12,D26)
           SS(5+J5N) = CDNDN(D66,D26,D26,D22)
           SS(1+J3N) = 0.
           SS(2+J3N) = 0.
           SS(3+J1N) = 0.
           SS(3+J2N) = 0.
C          DO  (ADD ON THE TERMS INVOLVING  C55,C54,C44)
              SS(3+J3N) = CDNDN(C55,C54,C54,C44)
              XN1N = XNNDN(K2,0,1,K1)
              XN2N = XNNDN(K2,0,2,K1)
              SS(3+J4N) = C55*XN1N + C54*XN2N
              SS(3+J5N) = C54*XN1N + C44*XN2N
              XN1N = XNNDN(K1,0,1,K2)
              XN2N = XNNDN(K1,0,2,K2)
              SS(4+J3N) = C55*XN1N + C54*XN2N
              SS(5+J3N) = C54*XN1N + C44*XN2N
              XNN = XNNNN(K1,K2,0,0)
              SS(4+J4N) = SS(4+J4N) + C55*XNN
              SS(4+J5N) = SS(4+J5N) + C54*XNN
              SS(5+J4N) = SS(5+J4N) + C54*XNN
              SS(5+J5N) = SS(5+J5N) + C44*XNN
C          CONTINUE
           IF (CURVE)                                     GO TO 2
                                                          GO TO 7
C          THEN  ADD ON THE CURVATURE TERMS
 2            DIAG = 0
              OFFD = 0
              DO 6 K3=1,NNE
                 QU1  = Q1(K3)
                 QU2  = Q2(K3)
                 QU21 = Q21(K3)
                 CURC1 = CURV(C11,C12,C16)
                 CURC2 = CURV(C12,C22,C26)
                 CURC6 = CURV(C16,C26,C66)
                 CURF1 = CURV(F11,F12,F16)
                 CURF2 = CURV(F12,F22,F26)
                 CURF6 = CURV(F16,F26,F66)
                 XNN1N = XNNDN(K2,K3,1,K1)
                 XNN2N = XNNDN(K2,K3,2,K1)
                 SS(1+J3N) = SS(1+J3N) + CURC1*XNN1N + CURC6*XNN2N
                 SS(2+J3N) = SS(2+J3N) + CURC6*XNN1N + CURC2*XNN2N
```

```
                    SS(4+J3N) = SS(4+J3N) + CURF1*XNN1N + CURF6*XNN2N
                    SS(5+J3N) = SS(5+J3N) + CURF6*XNN1N + CURF2*XNN2N
                    XNN1N = XNNDN(K1,K3,1,K2)
                    XNN2N = XNNDN(K1,K3,2,K2)
                    SS(3+J1N) = SS(3+J1N) + CURC1*XNN1N + CURC6*XNN2N
                    SS(3+J2N) = SS(3+J2N) + CURC6*XNN1N + CURC2*XNN2N
                    SS(3+J4N) = SS(3+J4N) + CURF1*XNN1N + CURF6*XNN2N
                    SS(3+J5N) = SS(3+J5N) + CURF6*XNN1N + CURF2*XNN2N
C                   . THE FOLLOWING CODE IS EQUIVALENT TO SUMMING
C                      BOTH K3 AND K4 FROM 1 TO NNE.
C                   . OFF-DIAGONAL TERMS HAVE A COMPENSATING FACTOR OF 2.
                    IF(K3.NE.1)                         GO TO 3
                                                        GO TO 5
C                   THEN      (TERMS WITH K3 NOT EQUAL K4)
 3                     K3M1 = K3 - 1
                       DO 4 K4=1,K3M1
                          OFFD = OFFD + XNNNN(K1,K2,K3,K4)*
     *                       (Q1(K4)*CURC1 +Q2(K4)*CURC2 +Q21(K4)*CURC6)
 4                     CONTINUE
C                   CONTINUE      (TERM WITH K3 = K4)
 5                  DIAG = DIAG + XNNNN(K1,K2,K3,K3)*
     *                 (QU1*CURC1 +QU2*CURC2 +QU21*CURC6)
 6               CONTINUE
                 SS(3+J3N) = SS(3+J3N) + DIAG + OFFD + OFFD
 7            CONTINUE
              IF (SGFLAG)
C             THEN
     *           SG(K1+K2G) = EN1*X1N1N + EN2*X2N2N + EN12*(X1N2N+X2N1N)
C             CONTINUE
 8         CONTINUE
C       CONTINUE
C*********************************************************************
C       . SYMMETRIZE SS AND SG
        K1N = ISS - NDFE - 5
        K1M = -NDFE5
        NSFM1 = NSF - 1
        DO 9 K1=1,NSFM1
           K1N = K1N + 5
           K1M = K1M + NDFE5
           K2N = K1M + K1N
           K2M = K2N
           K1P1 = K1 + 1
           DO 9 K2=K1P1,NSF
              K2N = K2N + NDFE5
              K2M = K2M + 5
              IF (SGFLAG)
C             THEN
     *           SG(ISG+K1+K2*NSF-NSF) = SG(ISG+K2+K1*NSF-NSF)
C             CONTINUE
              DO 9 J1=1,5
                 DO 9 J2=1,5
                    SS(J1+J2*NDFE+K2N) = SS(J2+J1*NDFE+K2M)
C                CONTINUE
C             CONTINUE
C          CONTINUE
 9      CONTINUE
C   ...ONLY EXIT
        RETURN
        END
```

```
      SUBROUTINE LODVEC
C*********************************************************************
C
C     THIS SUBROUTINE EVALUATES THE CONSISTENT LOAD SP.
C     THE NUMBER OF DEGREES OF FREEDOM PER NODE IS FIVE
C     NSF  IS THE NUMBER OF SHAPE FUNCTIONS PER ELEMENT
C     NNE  IS THE NUMBER OF *NODAL* SHAPE FUNCTIONS PER ELEMENT
C     P  CONTAINS THE NODAL VALUES OF THE NORMAL(TRANSVERSE) LOADS
C     P1,P2  CONTAIN THE NODAL VALUES OF THE IN-PLANE LOADS
C     SP IS TO BE STORED BETWEEN POSITIONS  IXC+1  AND  IXC+5*NNE OF
C        COMMON.
C
C*********************************************************************
C
      COMMON/SPACE/SP(70),P(10),P1(10),P2(10),
     *   DUMMY(14),NNE,SKP(2),IXC,OTHERS(1)
      COMMON/TEMP/SKIP(6),I,IL,IU,K,K1,K2,XNN
C
C
      IL = IXC + 1
      IU = IXC + 5*NNE
      DO 1 I=IL,IU
         SP(I) = 0
 1    CONTINUE
      K = IXC - 5
      DO 2 K1=1,NNE
         K = K + 5
         DO 2 K2=1,NNE
            XNN = XNNNN(K1,K2,0,0)
            SP(1+K) = SP(1+K) + P1(K2)*XNN
            SP(2+K) = SP(2+K) + P2(K2)*XNN
            SP(3+K) = SP(3+K) -  P(K2)*XNN
C        CONTINUE
 2    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MASS
C*********************************************************************
C
C     THIS SUBROUTINE CALCULATES THE CONSISTENT MASS MATRIX  SM  FOR AN
C        ISOLATED DOUBLY-CURVED SHALLOW SHELL FINITE ELEMENT.
C     RHO  IS THE DENSITY OF THE SHELL MATERIAL
C     H   IS THE SHELL THICKNESS
C     SM  IS TO BE STORED BETWEEN POSITIONS  IXC+5*NNE+1  AND
C        IXC+5*NNE+NSF*NSF  OF COMMON.
C
C*********************************************************************
C
      COMMON/SPACE/SM(24),NSF,DUMMY(5),RHO,H,DUM(68),
     *   SKIPP(14),NNE,SKP(2),IXC,OTHERS(1)
      COMMON/TEMP/SKIP(6),IXM,K1,K1N,K2,TEMP
C
C
      TEMP = RHO*H
      IXM = IXC + 5*NNE
      DO 2 K1=1,NSF
         K1N = IXM + K1 - NSF
         DO 2 K2=1,NSF
            SM(K1N+K2*NSF) = TEMP*XNNNN(K1,K2,0,0)
C        CONTINUE
 2    CONTINUE
      RETURN
      END
      FUNCTION XDNDN(L1,K1,L2,K2)
C***********************************************************************
C
C     THIS SUBROUTINE RETRIEVES XDNDN(L1,K1,L2,K2) FROM WHERE IT WAS
C        PREVIOUSLY STORED.  IT IS LOCATED BETWEEN XC(IXC+1) AND
C        XC(IXC+IC).
C
C***********************************************************************
C
      DIMENSION M(10)
      COMMON/SPACE/XC(110),IC,SKIP(6),IXC,OTHERS(1)
      COMMON/TEMP/INDX
      DATA (M(I),I=1,10)/0,1,3,6,10,15,21,28,36,45/
C
C
C     IF (K1.GE.K2)
                                                 IF(K1.LT.K2) GO TO 1
C     THEN
         INDX = K2 + M(K1) + IC*(L1+L1+L2-3)
                                                 GO TO 2
C     ELSE
 1       INDX = K1 + M(K2) + IC*(L1+L2+L2-3)
 2    CONTINUE
      XDNDN = XC(IXC+INDX)
      RETURN
      END
```

```
      FUNCTION XNNDN(K1,K2,L,K3)
C***********************************************************************
C
C     THIS SUBROUTINE RETRIEVES XNNDN(K1,K2,L,K3) FROM WHERE IT WAS
C        PREVIOUSLY STORED.  IT IS LOCATED BETWEEN XB(IBX+1) AND
C        XB(IBX+IB).
C
C***********************************************************************
C
      DIMENSION M(11)
      COMMON/SPACE/XB(24),NSF,SKIP(75),SKP(9),IB,SKIPP(8),IXB,OTHERS(1)
      COMMON/TEMP/INDX
      DATA (M(I),I=1,11)/0,1,3,6,10,15,21,28,36,45,55/
C
C
C     IF (K1.GE.K2)
                                                 IF(K1.LT.K2) GO TO 1
C     THEN
         INDX = NSF*(K2+M(K1+1))
                                                 GO TO 2
C     ELSE
 1       INDX = NSF*(K1+M(K2+1))
 2    CONTINUE
      XNNDN = XB(IXB+INDX+IB*(L-1)+K3)
      RETURN
      END
      FUNCTION XNNNN(KK1,KK2,KK3,KK4)
C**************************************************************************
C
C     THIS SUBROUTINE RETRIEVES XNNNN(K1,K2,K3,K4) FROM WHERE IT WAS
C        PREVIOUSLY STORED.  IT IS LOCATED BETWEEN XA(IXA+1) AND
C        XA(IXA+IA).
C
C**************************************************************************
C
      DIMENSION M1(11),M2(11),M3(11)
      COMMON/SPACE/XA(119),IXA,OTHERS(1)
      COMMON/TEMP/KJ,INDX,K1,K2,K3,K4
      DATA (M1(I),I=1,11)/0,1,5,15,35,70,126,210,330,495,715/
      DATA (M2(I),I=1,11)/0,1,4,10,20,35,56,84,120,165,220/
      DATA (M3(I),I=1,11)/0,1,3,6,10,15,21,28,36,45,55/
C
C
      K1 = KK1 + 1
      K2 = KK2 + 1
      K3 = KK3 + 1
      K4 = KK4 + 1
C     . PLACE  K1,K2,K3,K4  INTO DESCENDING ORDER .
      IF (K1.GE.K3) GO TO 1
         KJ = K1
         K1 = K3
         K3 = KJ
 1    IF (K2.GE.K4) GO TO 2
         KJ = K2
         K2 = K4
         K4 = KJ
 2    IF (K1.GE.K2) GO TO 3
         KJ = K1
         K1 = K2
         K2 = KJ
 3    IF (K3.GE.K4) GO TO 4
         KJ = K3
         K3 = K4
         K4 = KJ
 4    IF (K2.GE.K3) GO TO 5
         KJ = K2
         K2 = K3
         K3 = KJ
 5    CONTINUE
C     . K1,K2,K3,K4  ARE NOW IN DESCENDING ORDER .
      INDX = M1(K1) + M2(K2) + M3(K3) + K4
      XNNNN = XA(IXA+INDX)
      RETURN
      END
      OVERLAY(MAIN,13,0)
      PROGRAM PRINT
C*********************************************************************
C
C     THIS PROGRAM PRINTS SS, SG, SP AND SM IF THEY HAVE BEEN EVALUATED.
C        IT ALSO RECONSTRUCTS AND PRINTS THE FULL CONSISTENT MASS MATRIX
C        FROM  SM,  BUT IN DOING SO IT CLOBBERS  SS  IN CORE.
C
C*********************************************************************
C
      LOGICAL SGFLAG,SMFLAG,SPFLAG,SWM
      DIMENSION LABEL1(4),LABEL2(4),LABEL3(4),LABEL4(4),LABEL5(4)
      DIMENSION SS(1),SG(1),SP(1),SM(1),SMASS(1)
      COMMON/SPACE/SPACE(24),NSF,CURVE,SGFLAG,SMFLAG,SPFLAG,PRFLAG,
     *   RHO,H,DUMMY(68),
     *   SKIP(14),NNE,ISS,ISG,IXC,OTHERS(1)
      COMMON/TEMP/I,IL,IU,K1,K1N,K12,K2,K2N,NDFE,SWM,TEMP,TEMP1
      EQUIVALENCE (SS(1),SPACE(1)),(SG(1),SPACE(1)),(SP(1),SPACE(1)),
     *   (SM(1),SPACE(1)),(SMASS(1),SPACE(1))
C
      DATA LABEL1/40HSTIFFNESS MATRIX --- SS                 /
      DATA LABEL2/40HGEOMETRIC STIFFNESS ARRAY --- SG        /
      DATA LABEL3/40HLOAD VECTOR --- SP                      /
      DATA LABEL4/40HCONSISTENT MASS ARRAY --- SM            /
      DATA LABEL5/40HFULL CONSISTENT MASS MATRIX --- SMASS   /
C
C
      SWM = .TRUE.
      NDFE = 5*NSF
      IL = ISS + 1
      IU = ISS + NDFE*NDFE
      WRITE (6,1) LABEL1,(SS(I),I=IL,IU)
 1       FORMAT (1H1,40X,4A10//(10E12.5))
      IF (SGFLAG)                                   GO TO 2
                                                    GO TO 3
C     THEN
 2       IL = ISG + 1
         IU = ISG + NSF*NSF
         WRITE (6,1) LABEL2,(SG(I),I=IL,IU)
 3    CONTINUE
      IF (SPFLAG)                                   GO TO 4
                                                    GO TO 6
C     THEN
 4       IL = IXC + 1
         IU = IXC + 5*NNE
         WRITE (6,5) LABEL3,(SP(I),I=IL,IU)
 5          FORMAT(////40X,4A10//(10E12.5))
 6    CONTINUE
      IF (SMFLAG)                                   GO TO 7
                                                    GO TO 8
C     THEN
 7       IL = IXC + 5*NNE + 1
         IU = IXC + 5*NNE + NSF*NSF
         WRITE (6,5) LABEL4,(SM(I),I=IL,IU)
 8    CONTINUE
      IF (SWM .AND. SMFLAG)                         GO TO 9
                                                    GO TO 12
C     THEN
 9       IL = ISS + 1
         IU = ISS + NDFE*NDFE
         DO 10 I=IL,IU
            SS(I) = 0.
 10      CONTINUE
         TEMP1 = H*H/12.
         K1N = ISS - NDFE - 5
         DO 11 K1=1,NSF
            K1N = K1N + 5
            DO 11 K2=1,NSF
               K12 = 5*NDFE*(K2-1) + K1N
```

```
                  TEMP = SM(IXC+5*NNE+K1+NSF*(K2-1))
                  SMASS(1+K12+  NDFE) = TEMP
                  SMASS(2+K12+2*NDFE) = TEMP
                  SMASS(3+K12+3*NDFE) = TEMP
                  SMASS(4+K12+4*NDFE) = TEMP*TEMP1
                  SMASS(5+K12+5*NDFE) = TEMP*TEMP1
C              CONTINUE
 11         CONTINUE
      WRITE(6,1) LABEL5,(SMASS(I),I=IL,IU)
 12   CONTINUE
      END
```

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:111114111145145431111411414331664336322862224543384143343222222737322272         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:14321118111185444151554443633113364643633227282822442373722272                   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   8   5   2  10  11  12    63543
     13  14   7  15   3   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:111114111451454311141141433166433632283822245433841433432222227373222721111411454355 0
   311414336322822433432227211141431432223211432 1                                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        -8       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   9  10   5   4   3   4   6  14  15  14  15   163587
     12  11  11  12  13   9   8   7   8  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:14321111811118154444151515444436333113316464436333227228282224422373732272214321 3592
   11181544443633322722214321                                                      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
```

APPENDIX B

```
          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:11414384321 4321                                                                  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  16  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  18  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  48  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      8       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      0        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   63688
      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   23   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   15   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:1321321161161161164334331411144334332522526222623355332522252141114111111444444143696
   111414111433553333355333333333555551151513355336222626666662226222222224242263333697
   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:11363214113351362 2432                                                                         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   8   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   6   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  59  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  78  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:11111411145145431114114143316643363228822245433841433432222273732227211115115553770
   11111111115555551151555551115151111155515111515146564664646664646666446444464444443771
   41111116166111611111114664646646444445666614646464333373377733373733737337773373 3772
   737771115153113735371151514444444444774771117714444414333737337373737337371777773773
   74337373732886222222338262888822822288282228222255522522222255555528842466462224 3774
   2226622464646438863833333333333355335484481433333343288222882822282222222282822288 3775
   8883832828222282                                                                  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
```

APPENDIX B

```
        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  263873
       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:143214321118118811181188544454445151511155444544453633663311331638646464443633663333879
   22722772828222822442547237373733227222772151511151111111155555555111111115555555553880
   151511156464644446666666644444444466666666444444441161161664646444373737333333333333881
   77777777333333337777777715371133444774743737373382822282888888882222222288888888883882
   2222222225225252846224428333383882822282                                          3883
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         -8        8        0       15                             3906
           5 0 6          4       -4        4        9                             3907
           5 0 7          0       -8        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         62     -120       14     1575                             3921
           5 3 6         -8       12      -16      225                             3922
           5 3 7         -2      120      -14     1575                             3923
           5 3 8         -8        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:11414384321515164641437371438282543 2                                                      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   9  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  48  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  96 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:111114111451454311141141433166433632286222454338414334322222273732227211115115553985
   11111111115555551151555551115151111555151151514666646646466664666644644446444443986
   41111115166111611111114664646646444446666614646464333373377733373733737337773373 3987
   73777111515311373537115151444444444447747711177144444143337373373737373373717777 73968
   74337373737288822222228882328888228222882822282222555225222222255555528864246646222243989
   22266224646464388383333338333355335484481433333343288222882822282222222282822288 3990
   88638328322282111141145431141433632282243343222721115151111555151151514664646664 3991
   64444411161146464643337373373737371153714444714337373732388222832822282225225284 63992
   22438333343282822282111141431432223211515146464143373714328282223211432143 21     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	63	-68	444	33075	4049
6	5	4	0	-16	16	-112	1575	4050
6	5	4	1	63	-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

```
    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     -2060     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  61  82  82  82  82  83                   4148
LB:143214321111811381111311381544454454151511151544454454363366333113316381646464444149
   4363366333227227722828222822244254722373737333227227722151511151111111115555554150
   5511111111115555555551515111516464644446666666666444444444466666666644444444411611 64151
   16164646444437373733333333333337777777773333333333777777777153711331444774744373734152
```

APPENDIX B

```
      7333826222622888666888822222222228386888882222222222522525228462244228333383838282824153
      2282214321432111181158154445445436336633322722277221515111516464644443737373338284154
      2228221432143 21                                                                    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       -28     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       128       -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    8    9    9    9        94240
     10   10   10   10   11                                                                               4241
LC:11414384321515164641437371438282543214321                                                           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   6   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  45  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  55  51  52  68  69  69  69  70  57  60  58  594262
      70  61  32  35  37  35  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  67  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  57  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
     144 157 159 160 158 152 157 160 155 159 163 161 152 154 155 153 161 162 156 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
     181 180 182 182 181 183 184 185 186 187 183 166 184 185 188 187 163 185 186 1844301
     166 168 167 183 185 184 186 189 190 191 187 183 186 185 184 191 189 190 186 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   3343840                                                   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
```

APPENDIX B

```
     <
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    1261280     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    862400                                        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
```

```
    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
    10101010     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  95  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 1024527
       103 104 104 105 106 107 107 106 105 108 104 103 104 107 105 106 105 107 106 1084528
       104 104 103 106 107 105 106 105 107 108 109 110 111 112 113 114 115 116 117 1184529
       111 109 110 114 112 113 117 115 116 118 110 111 109 113 114 112 116 117 115 1184530
       110 109 111 115 117 116 112 114 113 118 111 110 109 116 115 117 113 112 114 1184531
       109 111 110 117 116 115 114 113 112 118 119 119 119 120 120 120 120 120 120 1214532
LB: 132132456111611166611161116661433333444314111144414333334443252222555262222266624533
    335333555325222255521111111111111111111111111111111111111111111111111133333333334534
    333333333333333333333333333333111111111133333333332222222222222222222222222222224535
    222222222222222222223333333333222222222244444444444444444444444444444444444444444536
    141111444111511155511161116661444444444445555555555555555555555555555555555555554537
    433333444333533355536333336663555555555555555555555666666666666666666666666666664538
    666666666624222224442252222555262222266624444444446666666666666666666613213245614539
    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        108       -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       8960                                     4577
         4 1 6       -108         99       8960                                     4578
         4 1 7        108        -45       8960                                     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        -81      -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:11363211133313222132444111455533345666222456132132456 1                        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
          1010      162       -81       -81       162        80                   4680
```

APPENDIX C

SAMPLE CALLING PROGRAM

```
      OVERLAY(MAIN,0,0)
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     *   TAPE3,TAPE4,TAPE8,TAPE9)
      DIMENSION CFD(21)
      COMMON/TEMP/I,TEMP(60)
      COMMON/STORE/STORE(40)
      COMMON/SPACE/SPACE(24),NSF,CURVE,SGFLAG,SMFLAG,SPFLAG,PRFLAG,
     *   RHO,H,X(4),Y(4),SKIP(60),NRECORD(7),
     *   OTHERS(5200)
      EQUIVALENCE(SPACE(1),CFD(1))
      LOGICAL CURVE,SGFLAG,SMFLAG,SPFLAG,PRFLAG
C
C
C     . SET  NRECORD(I) = 1  TO ALLOW FINITE ELEMENTS WITH  NSF = I + 3
C        TO BE COMPUTED.  I  MAY TAKE ON VALUES  1,2,3,5,6, OR 7.
C        THE REMAINING COMPONENTS OF  NRECORD  SHOULD BE SET TO ZERO.
      NRECORD(1) = 1
      NRECORD(2) = 1
      NRECORD(3) = 1
      NRECORD(5) = 1
      NRECORD(6) = 1
      NRECORD(7) = 1
      NSF = 6
C     . SET FIVE FLAGS.
      CURVE = .T.
      SGFLAG = .T.
      SMFLAG = .T.
      SPFLAG = .T.
      PRFLAG = .T.
C     . DEFINE DENSITY AND THICKNESS.
      RHO = .000001
      H = .04
C     . DEFINE COORDINATES OF CORNER NODES FOR TRIANGULAR ELEMENTS.
      X(1) = .2
      Y(1) = .1
      X(2) = .6
      Y(2) = .3
      X(3) = .4
      Y(3) = .5
      X(4) = 0.
      Y(4) = 0.
C     . DEFINE EXTENSIONAL AND TRANSVERSE SHEAR STIFFNESSES OF SHELL.
      CFD(1) = 1.23970
      CFD(2) = .165820
      CFD(3) = -.147991
      CFD(4) = .0912631
      CFD(5) = .0385601
      CFD(6) = .179804
      CFD(7) = .00244105
      CFD(8) = -.000905935
      CFD(9) = -.00318850
      CFD(10) = -.000629179
      CFD(11) = -.00102919
      CFD(12) = -.000905935
      CFD(13) = .000149019
      CFD(14) = .0000281489
      CFD(15) = .00000152451
      CFD(16) = .0000163629
      CFD(17) = .0000120026
      CFD(18) = .0000300134
      CFD(19) = .0234700
      CFD(20) = .0205300
      CFD(21) = -.000280154
```

```
C       . DEFINE PRESTRESS COEFFICIENTS.
        SPACE(22) = 1.
        SPACE(23) = 1.
        SPACE(24) = 1.
C       . DEFINE CURVATURE AND LOAD COMPONENTS.
        SPACE(41) = .5
        SPACE(51) = .5
        SPACE(61) = .1
        SPACE(71) = 1.
        SPACE(81) = 0.
        SPACE(91) = 0.
        DO 7 I=41,91,10
           DO 7 J=1,9
              SPACE(I+J) = SPACE(I)
C          CONTINUE
  7     CONTINUE
C       . DISPLAY FIRST  100  WORDS IN (LABELED COMMON) SPACE.
        WRITE(6,4)
  4        FORMAT(1H1)
        WRITE(6,1) (SPACE(I),I=1,100)
  1        FORMAT(* THE CONTENTS OF THE FIRST 100 WORDS OF LABELED*
     *        * COMMON /SPACE/ ARE AS FOLLOWS*//
     *        1X,12E11.4/1X,12E11.4/18,5L11/7(1X,10E11.4/))
        NSF = 6
        CALL ELEMENT
        NSF = 10
        CALL ELEMENT
C       . DEFINE COORDINATES OF CORNER NODES FOR PARALLELOGRAM ELEMENTS.
        X(1) = .15
        Y(1) = .2
        X(2) = .55
        Y(2) = .1
        X(3) = .7
        Y(3) = .4
        X(4) = .3
        Y(4) = .5
        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
C       . DEFINE COORDINATES OF CORNER NODES FOR TRAPEZOIDAL ELEMENTS.
        X(1) = .1
        Y(1) = .2
        X(2) = .6
        Y(2) = .2
        X(3) = .5
        Y(3) = .5
        X(4) = .25
        Y(4) = .5
        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
```

```
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   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  .400E-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   7   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
```

APPENDIX D

```
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   8   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  .52801E-03  .18600E-03  .13271E-03  .22564E-01 -.15335E-02 -.58066E-02 -.44420E-03 -.47484E-03
-.43627E-02 -.78333E-03 -.53461E-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 -.49819E-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
 .19883E+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  .56461E-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  .14320E-03  .11646E-04 -.63263E-05 -.57767E-03  .13897E-02 -.57300E-03 -.59900E-04 -.38513E-05
-.42251E-02  .86793E-03 -.56421E-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  .13897E-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  .32801E-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  .13897E-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
```

APPENDIX D

```
 .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 -.12880E-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 -.47484E-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 -.11553E-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  .46668E-02 -.49819E-04  .30703E-03 0.          0.          -.11584E-02 0.          0.
-.93089E-01 -.91738E-02 -.35084E-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 -.38466E-02 -.12250E-03  .75147E-03 -.91005E-02  .11584E-02 -.53461E-03  .29780E-03 -.11656E-03
-.14213E-01  .58252E-02 -.38466E-02 -.24319E-03 -.53873E-03  .55353E-01 -.12401E-01 -.37595E-01 -.17064E-02  .15595E-02
 .53853E-01 -.15004E-02 -.94748E-02 -.18672E-02 -.16079E-03 -.72804E-01  .92674E-02  .63317E-01 -.26544E-02  .13172E-02
 .16536E-02 -.49819E-04 -.54100E-03  .19437E-04 -.12080E-04 0.          0.          -.29780E-03 -.31293E-04  .37354E-06
 .16536E-02 -.49819E-04 -.42040E-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 -.10337E-04 -.32700E-05 0.          0.           .11656E-03  .37354E-06 -.27373E-04
-.49819E-04  .30703E-03  .86603E-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.            .69369E-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.
```

APPENDIX D

```
 .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.          0.          .21333E-09  0.          0.
 0.          0.          .42667E-09  0.          0.          0.          0.          .21333E-09  0.          0.
 0.          0.          0.         -.71111E-14  0.          0.          0.          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.         -.71111E-14  0.          0.          0.          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.
```

APPENDIX D

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

 .1240E+01  .1658E+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 -.10438E-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  .23280E-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	$\tilde{\mathcal{N}}_{11}$		Prestress coefficients
23	EN2	$\tilde{\mathcal{N}}_{22}$		
24	EN12	$\tilde{\mathcal{N}}_{12}$	↓	

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)*	$k_{11}^i \quad (i = 1 \rightarrow m)$	LINSTF	
51-60	Q2(10)*	$k_{22}^i \quad (i = 1 \rightarrow m)$	↓	Nodal values of curvature components
61-70	Q12(10)*	$k_{12}^i \quad (i = 1 \rightarrow m)$	↓	
71-80	P(10)*	$p^i \quad (i = 1 \rightarrow m)$	LODVEC	Nodal values of transverse load
81-90	P1(10)*	$p_1^i \quad (i = 1 \rightarrow m)$	↓	Nodal values of in-plane loads
91-100	P2(10)*	$p_2^i \quad (i = 1 \rightarrow m)$	↓	

*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, evaluaates 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	Equations of appendix B
	MASS	101		Evaluates the consistent mass SM	Equations of appendix B
	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	CLOG	↓	↓	$\left\{\log[(1+s)/(1-s)] - 2s\right\}/s^3$
130	VLG1	↓	QUAD5	$\bar{L}_1(s,t)$
131	VLG2	↓	↓	$\bar{L}_2(s,t)$ } Logarithmic functions generated by the function subroutines WLOG1 and WLOG2
132	VLG3	↓	↓	$\bar{L}_2(t,s)$
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
	QA(2,JA)*	SETUP	SETUP	$\mathcal{R}_1^m, \mathcal{R}_2^m$
	QA(4,JA)*	↓	↓	$\mathcal{R}_1^m, \mathcal{R}_2^m, \mathcal{R}_3^m, \mathcal{R}_4^m$
121-	QB(3,JB)*	↓	↓	$\mathcal{S}_1^{\bar{m}}, \mathcal{S}_2^{\bar{m}}, \mathcal{S}_3^{\bar{m}}$
	QB(4,JB)*	↓	↓	$\mathcal{S}_1^{\bar{m}}, \mathcal{S}_2^{\bar{m}}, \mathcal{S}_3^{\bar{m}}, \mathcal{S}_4^{\bar{m}}$
	QC(5,JC)*	↓	↓	$\mathcal{T}_1^{\bar{\bar{m}}}, \mathcal{T}_2^{\bar{\bar{m}}}, \mathcal{T}_3^{\bar{\bar{m}}}, \mathcal{T}_4^{\bar{\bar{m}}}, \mathcal{T}_5^{\bar{\bar{m}}}$

*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		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		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		
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		
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		
QC3	IXC + 1 + IC	IXC + 2*IC		
QC4	IXC + 1	IXC + IC		
XA	IXA + 1	IXA + IA	TRI, QUAD, XNNNN	Evaluated A-integrals $A^{ijk\ell}$ with $i \geqq j \geqq k \geqq \ell$
XB	IXB + 1	IXB + 2*IB	TRI, QUAD, XNNDN	Evaluated B-integrals B^{ijk}_{α} with $i \geqq j$
XC	IXC + 1	IXC + 4*IC	Many	Evaluated C-integrals $C^{ij}_{\alpha\beta}$ with $i \geqq j$
SS	ISS + 1	ISS + 25*NSF*NSF	LINSTF, STORE, PRINT	Stiffness array
SG	ISG + 1	ISG + NSF*NSF		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	↓	$L_1\left(s^{\bar{\bar{n}}},t^{\bar{\bar{n}}}\right)$, $L_2\left(s^{\bar{\bar{n}}},t^{\bar{\bar{n}}}\right)$, $L_2\left(t^{\bar{\bar{n}}},s^{\bar{\bar{n}}}\right)$	
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

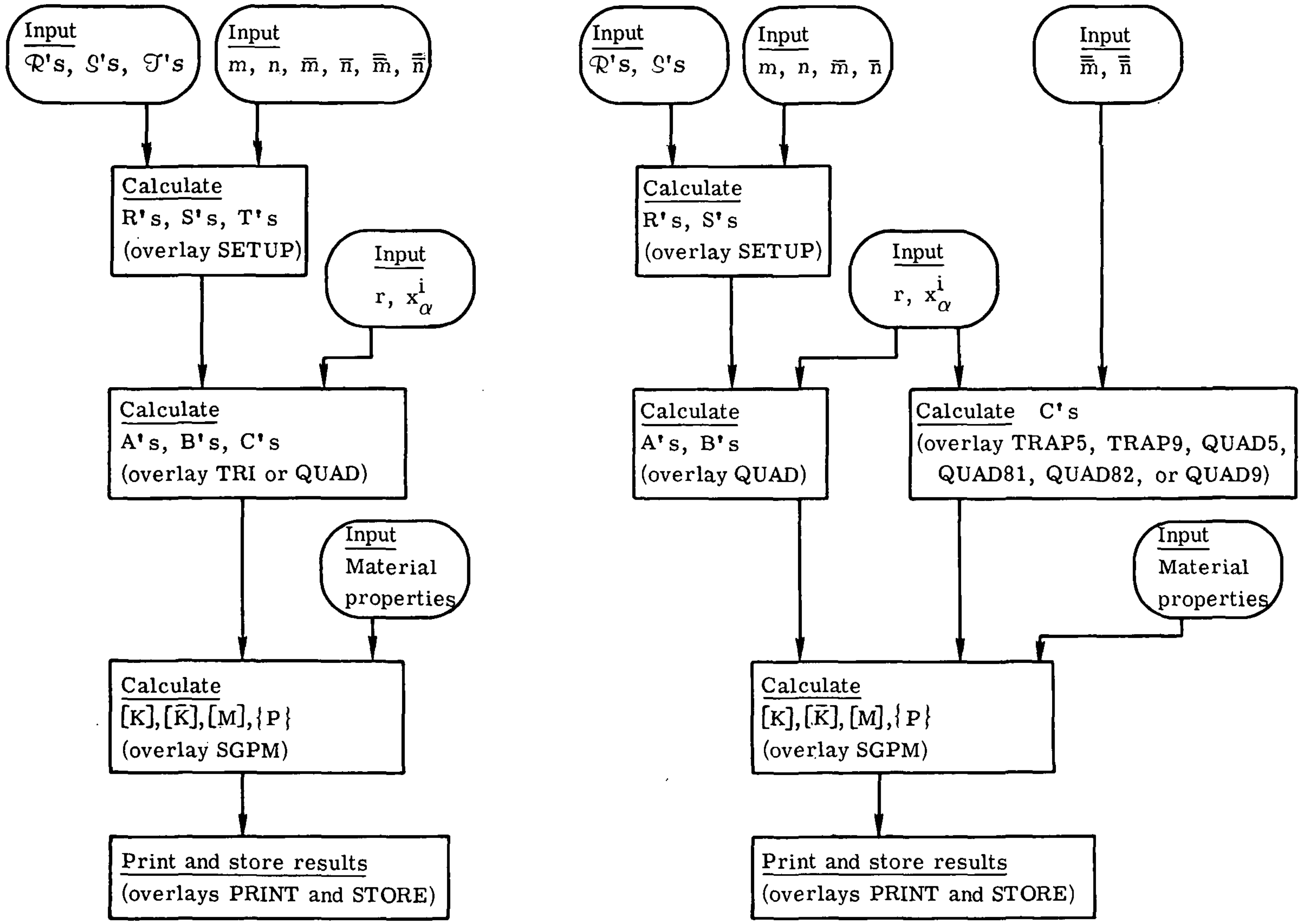
[System used FORTRAN Extended (Version 4) compiler under CONTROL DATA Network Operating System (NOS 1.0) running on CONTROL DATA CYBER 175 computer system]

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.


Input $\mathcal{R}$'s, $\mathcal{S}$'s, $\mathcal{T}$'s
Input m, n, $\bar{m}$, $\bar{n}$, $\bar{\bar{m}}$, $\bar{\bar{n}}$
Calculate R's, S's, T's (overlay SETUP)
Input r, x_α^i
Calculate A's, B's, C's (overlay TRI or QUAD)
Input Material properties
Calculate [K], [$\bar{K}$], [M], {P} (overlay SGPM)
Print and store results (overlays PRINT and STORE)
Input $\mathcal{R}$'s, $\mathcal{S}$'s
Input m, n, $\bar{m}$, $\bar{n}$
Input $\bar{\bar{m}}$, $\bar{\bar{n}}$
Calculate R's, S's (overlay SETUP)
Input r, x_α^i
Calculate A's, B's (overlay QUAD)
Calculate C's (overlay TRAP5, TRAP9, QUAD5, QUAD81, QUAD82, or QUAD9)
Input Material properties
Calculate [K], [$\bar{K}$], [M], {P} (overlay SGPM)
Print and store results (overlays PRINT and STORE)

(a) For triangular and parallelogram elements.

(b) For trapezoidal and trapezium elements.

Figure 1.- Logical organization of SYMINSE program.

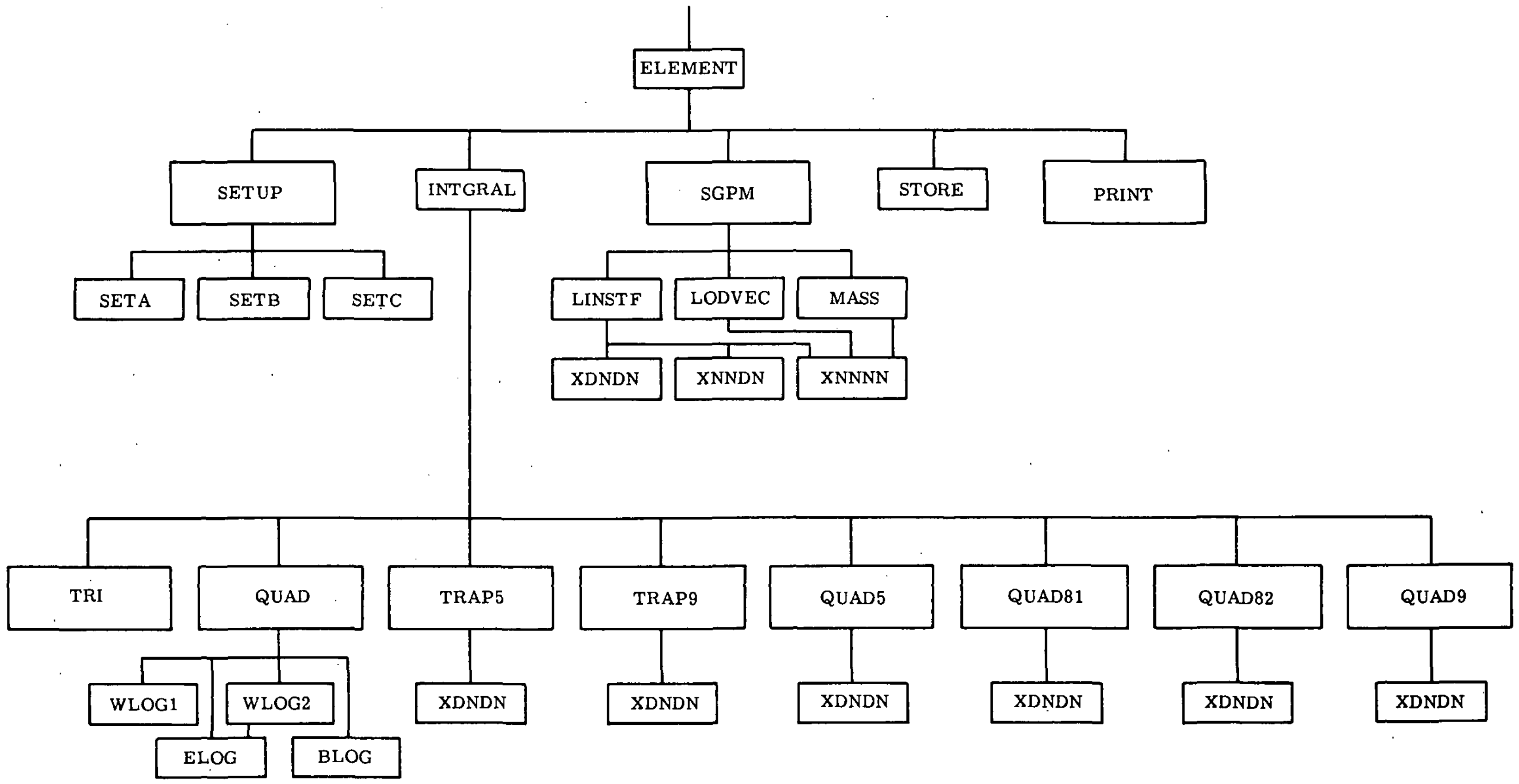


Figure 2.- Subroutine linkages for SYMINSE program. The large boxes represent main programs of primary overlays. The small boxes represent subroutines.

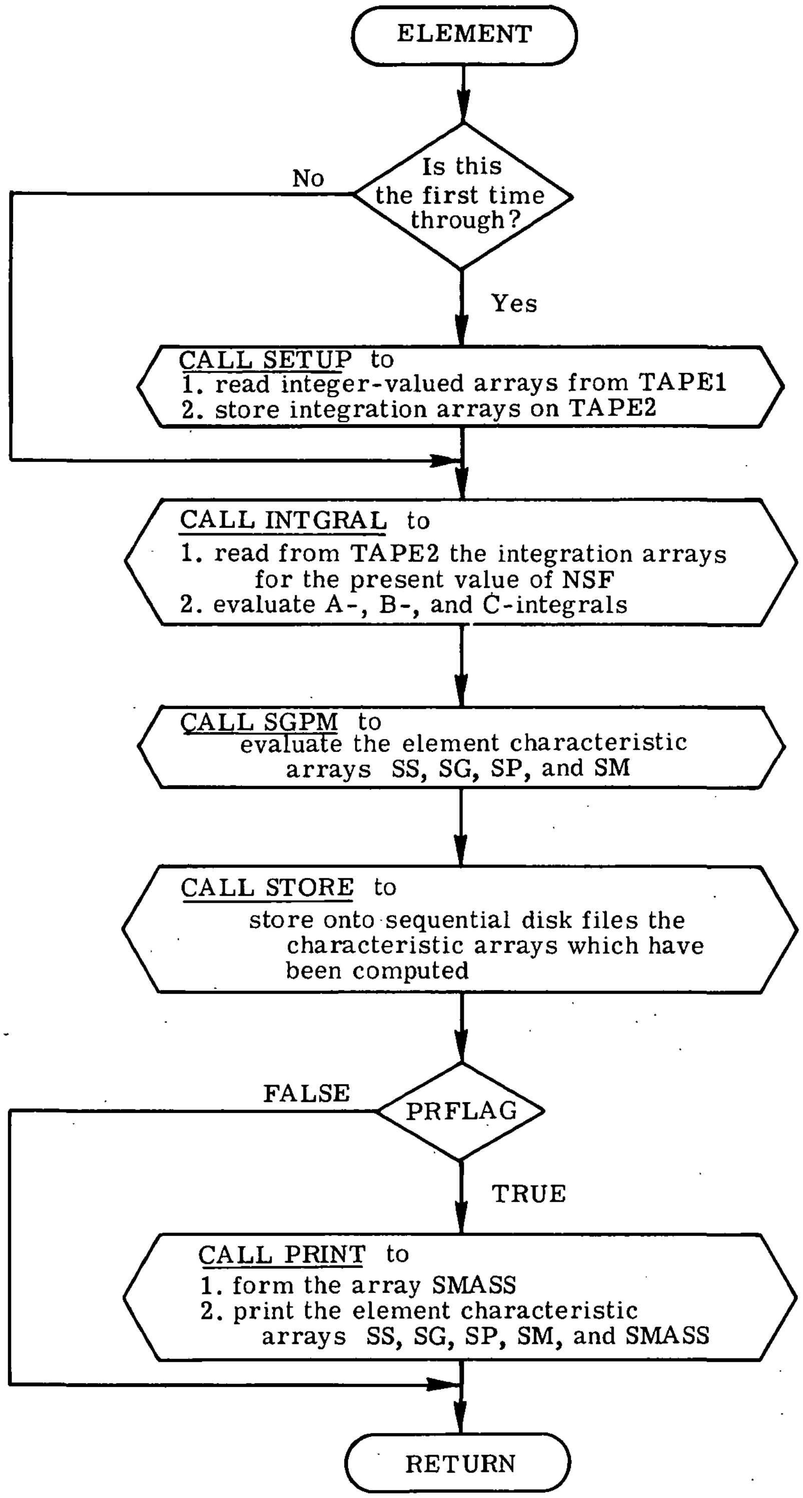

ELEMENT
Is this the first time through?
No
Yes
CALL SETUP to
1. read integer-valued arrays from TAPE1
2. store integration arrays on TAPE2
CALL INTGRAL to
1. read from TAPE2 the integration arrays for the present value of NSF
2. evaluate A-, B-, and C-integrals
CALL SGPM to
evaluate the element characteristic arrays SS, SG, SP, and SM
CALL STORE to
store onto sequential disk files the characteristic arrays which have been computed
PRFLAG
FALSE
TRUE
CALL PRINT to
1. form the array SMASS
2. print the element characteristic arrays SS, SG, SP, SM, and SMASS
RETURN

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

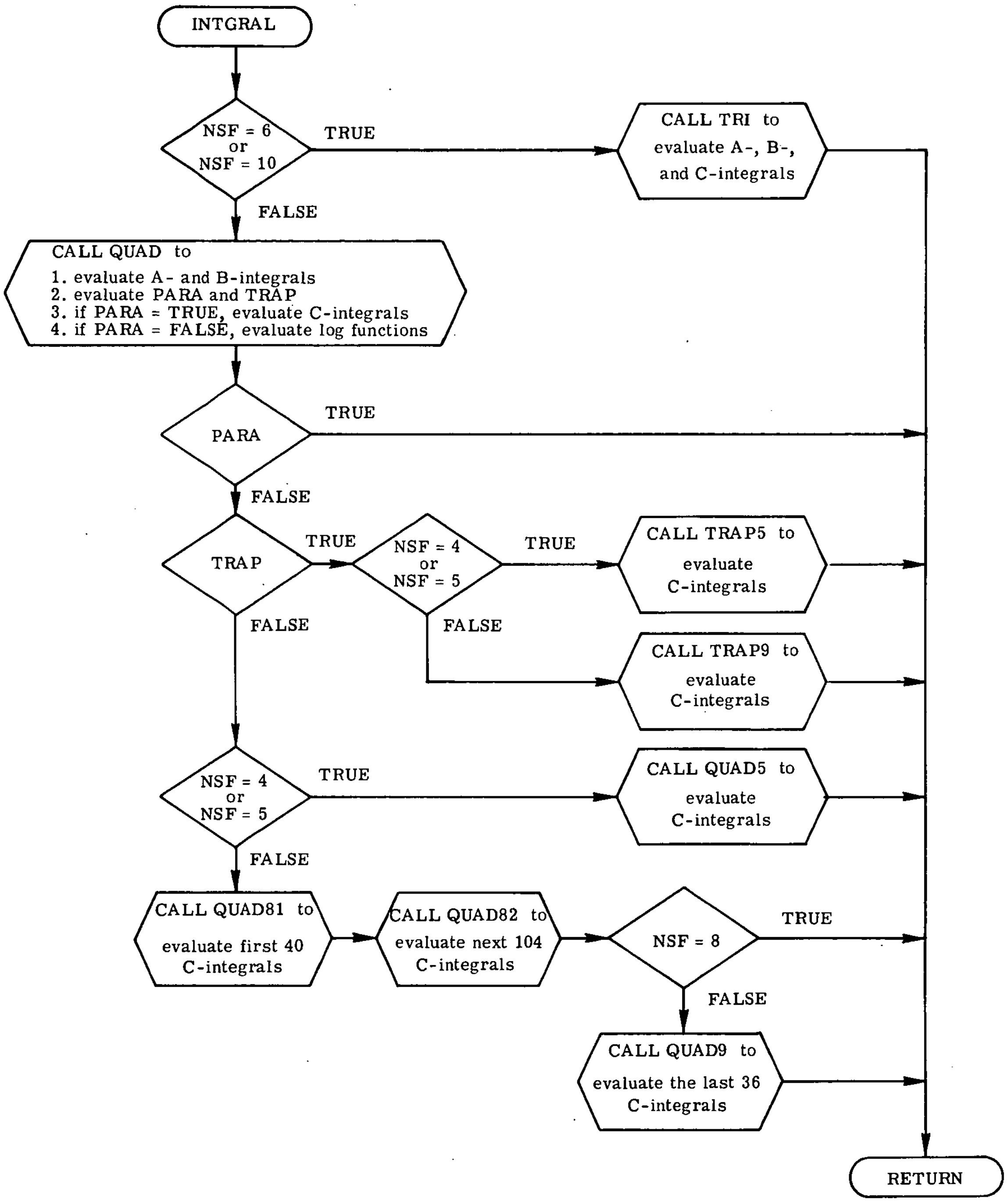

INTGRAL
NSF = 6 or NSF = 10
TRUE
FALSE
CALL TRI to evaluate A-, B-, and C-integrals
CALL QUAD to
1. evaluate A- and B-integrals
2. evaluate PARA and TRAP
3. if PARA = TRUE, evaluate C-integrals
4. if PARA = FALSE, evaluate log functions
PARA
TRUE
FALSE
TRAP
TRUE
FALSE
NSF = 4 or NSF = 5
TRUE
FALSE
CALL TRAP5 to evaluate C-integrals
CALL TRAP9 to evaluate C-integrals
NSF = 4 or NSF = 5
TRUE
FALSE
CALL QUAD5 to evaluate C-integrals
CALL QUAD81 to evaluate first 40 C-integrals
CALL QUAD82 to evaluate next 104 C-integrals
NSF = 8
TRUE
FALSE
CALL QUAD9 to evaluate the last 36 C-integrals
RETURN

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

NASA-Langley, 1976 L-10483

"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