STUDIES IN FINITE ELEMENT ANALYSIS OF COMPOSITE MATERIAL STRUCTURES

By
Dale O. Douglas
Donna E. Holzmacher
Zoa C. Lane
and
Earl A. Thornton

Final Technical Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

Under
Grant NSG 1043
June 1, 1974 - May 31, 1975

September 1975
NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
STUDIES IN FINITE ELEMENT ANALYSIS OF COMPOSITE MATERIAL STRUCTURES

By
Dale O. Douglas
Donna E. Holzmacher
Zoa C. Lane
and
Earl A. Thornton

Final Technical Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Grant NSG 1043
June 1, 1974 - May 31, 1975
Dr. J.G. Davis, Jr., Technical Monitor
Materials Division
Composites Section

Submitted by the
Old Dominion University Research Foundation
Norfolk, Virginia

September 1975
FOREWORD

This report presents four papers resulting from research conducted under a grant from NASA to the Old Dominion University Research Foundation entitled: "A Research Participation Program for Minority Engineering Students". The three undergraduate engineering students, Dale O. Douglas, Donna E. Holzmacher, and Zoa C. Lane, worked under the direction of Dr. Earl A. Thornton, Associate Professor of Mechanical Engineering and Mechanics.

The student-faculty team began their research in analysis of composite materials at Langley Research Center during a ten-week period in the summer of 1974. The work was continued during the academic year 1974-1975 at Old Dominion University.

Dr. John G. Davis, Jr., of the Composites Section, Materials Application Branch of the Materials Division served as technical monitor for the program. For his cooperation, encouragement, and counsel the authors express their deepest appreciation.
<table>
<thead>
<tr>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEFINITE ELEMENT ANALYSIS OF A PICTURE FRAME SHEAR TEST</td>
</tr>
<tr>
<td>Dale O. Douglas</td>
</tr>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>Description of Test</td>
</tr>
<tr>
<td>Test Specimen</td>
</tr>
<tr>
<td>Analysis of Shear Test</td>
</tr>
<tr>
<td>Results and Discussion</td>
</tr>
<tr>
<td>Concluding Remarks</td>
</tr>
<tr>
<td>References</td>
</tr>
<tr>
<td>Figures</td>
</tr>
<tr>
<td>BANSAP, A BANDWIDTH REDUCTION PROGRAM FOR SAP IV</td>
</tr>
<tr>
<td>Donna E. Holzmacher</td>
</tr>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>Collins Bandwidth Reduction Algorithm</td>
</tr>
<tr>
<td>SAP IV Preprocessing Program</td>
</tr>
<tr>
<td>Applications of BANSAP</td>
</tr>
<tr>
<td>Concluding Remarks</td>
</tr>
<tr>
<td>Appendix A: User Instructions</td>
</tr>
<tr>
<td>Appendix B: BANSAP Source Listing</td>
</tr>
<tr>
<td>Appendix C: Sample BANSAP Output</td>
</tr>
<tr>
<td>References</td>
</tr>
<tr>
<td>Tables</td>
</tr>
<tr>
<td>Figures</td>
</tr>
<tr>
<td>FEMESH, A FINITE ELEMENT MESH GENERATION PROGRAM BASED ON ISOPARAMETRIC ZONES</td>
</tr>
<tr>
<td>Zoa C. Lane</td>
</tr>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>Interpolation Function Techniques for Finite Element Generation</td>
</tr>
<tr>
<td>Program FEMESH</td>
</tr>
<tr>
<td>Applications</td>
</tr>
<tr>
<td>Concluding Remarks</td>
</tr>
<tr>
<td>Appendix A: User Instructions for FEMESH</td>
</tr>
<tr>
<td>Appendix B: FORTRAN Listing of Mesh Generation Program, FEMESH</td>
</tr>
<tr>
<td>References</td>
</tr>
<tr>
<td>Tables</td>
</tr>
<tr>
<td>Figures</td>
</tr>
</tbody>
</table>

(cont'd.)
# FINITE ELEMENT ANALYSIS OF A COMPOSITE BOLTED JOINT SPECIMEN

Earl A. Thornton

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>109</td>
</tr>
<tr>
<td>Bolted Joint Specimen</td>
<td>110</td>
</tr>
<tr>
<td>Analytical Procedures</td>
<td>110</td>
</tr>
<tr>
<td>Finite Element Model</td>
<td>110</td>
</tr>
<tr>
<td>Bolt Loads</td>
<td>112</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>113</td>
</tr>
<tr>
<td>Concluding Remarks</td>
<td>115</td>
</tr>
<tr>
<td>References</td>
<td>117</td>
</tr>
<tr>
<td>Table</td>
<td>118</td>
</tr>
<tr>
<td>Figures</td>
<td>119</td>
</tr>
</tbody>
</table>
FINITE ELEMENT ANALYSIS OF A PICTURE FRAME SHEAR TEST

By:

Dale O. Douglas
FINITE ELEMENT ANALYSIS OF A PICTURE FRAME SHEAR TEST

By

Dale O. Douglas

INTRODUCTION

Shear testing of composite materials is generally concerned with two principal areas of interest: (1) to determine the in-plane shear properties, or (2) to determine the interlaminar or normal shear properties. In-plane shear properties of a laminate are among the most difficult to determine because of problems in applying a state of uniform shearing stress. Concepts for determining in-plane shear properties include torsion tube tests, rail shear tests, and picture frame shear tests.

The most direct method of applying pure shear is by torsion of a tube. This test method has proven to be a reliable means of determining in-plane shear properties (ref. 1). However, fabrication techniques for high quality ± 45° metal matrix composite tubes have not yet been established. The difficulty of fabricating high quality tubes has stimulated research in other methods of shear testing.

Another type of shear test is the rail shear test. It uses a thin laminate, loaded along its length by two pairs of rails, leaving an unsupported central test section.

In the present study an analysis of a picture frame shear test performed at Langley Research Center is presented. The purposes of the study were to determine the stress distributions in the picture frame shear test specimen and to determine the effect of local reinforcements on the stress distributions.

DESCRIPTION OF TEST

The experimental setup for a picture frame shear test is shown in figure 1. The picture frame shear test was used to
produce in-plane shear stress in the test panel. The shear panel was bonded to a frame constructed from four 1 in. x 1 in. steel edge bars designed to simulate fully clamped edge conditions. The panel specimen was bolted to a test frame by 0.375-in.-diameter bolts, seven per side. At each corner of the test frame, loads were applied to the pin joints by the testing machine. Tensile loads were applied to the vertical pins, and compressive loads were applied to the horizontal pins to produce shear loading in the test specimen.

TEST SPECIMEN

The test specimens were made using 7 in. x 7 in. borsic aluminum sandwich shear panels. With the addition of 1 in. x 1 in. steel edge bars, the overall dimensions of the shear panel specimen were 9 in. x 9 in. with a nominal thickness of 1 in. To permit installation of the pins on the test frame, a portion of the shear panel was cut away at each corner. Each corner had a radius of 0.25 in. The test specimen is shown schematically in figure 2.

The sandwich panel consisted of two face sheets separated by a honeycomb core. On each face sheet there were four plies (0.0285 in. thick) at a ± 45° layup. The panel face sheets were cut from 10-in.-square laminates. The filaments of the laminate were parallel to the applied loads. Some of the test specimens were reinforced with titanium doublers (0.060 in. thick) in the vicinity of the corner radii.

ANALYSIS OF SHEAR TEST

Finite element analyses were made to determine the in-plane stress distributions in the shear panel. The finite models represented the shear panel specimens using orthotropic, two-dimensional plane stress elements. Two general purpose finite element computer programs were utilized in the analysis of the shear panel. The first was NASTRAN (NASA Structural Analysis Program) which was
used on the CDC-6600 computer at Langley Research Center. NASTRAN (ref. 2) is a general purpose digital computer program for the analysis of large complex structures. The second program, SAP IV (Structural Analysis Program), was executed on an IBM-370, Model 158 computer at Virginia Polytechnic Institute & State University through the computer center at Old Dominion University. SAP IV (ref. 3) is a structural analysis program for static and dynamic response of linear systems. Symmetry of loading, geometry, and material properties made the analysis of only one quarter of the specimen sufficient.

NASTRAN embodied a finite element approach, wherein the distributed physical properties of the shear specimen were represented by a model (fig. 3) consisting of 490 membrane elements that were interconnected at 529 grid points. The grid point definition formed the basic framework for the structural model. All other parts of the structural model were referenced either directly or indirectly to the grid points. Each grid point had two degrees of freedom, the in-plane displacements. The elements used in the analysis were the quadrilateral membrane element CQDMEM and the triangular membrane element CTRMEM.

The steel edge bars of the test specimen were represented in NASTRAN as rigid boundaries. The rigid boundaries were modeled using multipoint constraints in the NASTRAN program. The constraints were applied to grid points on the test frame edge of the finite element model so that these points deformed as a straight line. Static loads were applied to the structural model through nodes constrained to the rigid boundary.

The loads were from Langley Research Center Test 560, Run 7; a horizontal load of 5004.9 lb and a vertical load of 5039.4 lb are shown in figure 3 at the points of application.

SAP embodied a finite element approach where the shear specimen was represented by a model (fig. 4) consisting of 554 membrane elements that were interconnected at 595 nodal points.
The steel edge bars of the test specimen were represented in SAP as deformable boundaries. The deformable boundaries were simulated by the addition of 64 plane stress membrane elements to the NASTRAN model. The horizontal and vertical applied loads were represented by statically equivalent loads applied along the simulated boundary. Nine colinear loads were applied at nodal points nearest the center of each bolt hole. These loads were applied at an angle of 45 degrees. The magnitudes of these applied loads are given in figure 4. Stresses were computed at the centroid of each element using the stress print option available in SAP.

The titanium doublers used for local reinforcement at corner radii were modeled with an addition of 21 finite elements on existing elements at the extreme corner of the sandwich panel. The material elasticity matrix for titanium and Borsic aluminum is given in table 1.

### Table 1. Material elasticity matrix.

<table>
<thead>
<tr>
<th></th>
<th>$G_{11}$</th>
<th>$G_{12}$</th>
<th>$G_{13}$</th>
<th>$G_{21}$</th>
<th>$G_{22}$</th>
<th>$G_{23}$</th>
<th>$G_{33}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Borsic Aluminum</td>
<td>2.81E+7</td>
<td>5.65E+6</td>
<td>0</td>
<td>2.81E+7</td>
<td>0</td>
<td>9.5E+6</td>
<td></td>
</tr>
<tr>
<td>Titanium</td>
<td>1.81E+7</td>
<td>6.15E+6</td>
<td>0</td>
<td>1.81E+7</td>
<td>0</td>
<td>6.15E+7</td>
<td></td>
</tr>
</tbody>
</table>

The NASTRAN finite element model of the shear panel, simulating a rigid boundary, had 1000 degrees of freedom. Using a CDC-6400 computer, it took 945 CPU seconds for the program to execute. In contrast to the NASTRAN model, the SAP finite model
had 1132 degrees of freedom with a bandwidth of 1106. Due to the excessive storage required by the large bandwidth, the SAP finite element program was unable to execute. To optimize the bandwidth, the nodes were renumbered using a computer program, BANSAP. With this renumbering, the SAP program had a final bandwidth of 69. It then completed execution in 160 CPU seconds.

RESULTS AND DISCUSSION

The stresses computed in the shear panel for the loads applied to the rigid (NASTRAN) and deformable (SAP) boundary models are given in figures 5 through 8. Normal stresses $\sigma_x$ and $\sigma_y$ are plotted as ordinates with the horizontal and vertical coordinates from the center of the shear panel as abscissae.

The stress distributions along the horizontal and vertical axes of both the rigid and deformable boundary models are uniform near the center of the specimen. The uniform stress values differ considerably between the two models; the uniform normal stresses predicted by the rigid boundary model are nearly three times the stresses predicted assuming a deformable boundary. These results indicate that the assumption of a rigid boundary should not be made.

There is an appreciable stress concentration at the corner fillets. The stress components perpendicular to the lines of symmetry rise sharply at the corners. For example, figure 7 shows that in the deformable boundary model the stress component $\sigma_y$ increases from 10 ksi to about 105 ksi indicating a stress concentration factor of over 10.

Contour plots of the principal shearing stresses for the rigid and deformable boundary models are shown in figures 9 and 10. The shearing stresses are uniform only over a small portion of the specimen. Figure 10 shows that the shearing stress may vary by as much as 25 percent over the center one-half of the specimen.
The effect of the reinforcing titanium doubler on the normal stresses $\sigma_x$ and $\sigma_y$ is shown in figures 11 through 14. These results indicate that the doubler significantly reduced the stresses along the $x$-axis near the fillet. The critical stress $\sigma_y$ on this axis was reduced by about one-half. However, stress distributions along the vertical axis and in the center of the shear panel show no reinforcing effects of the titanium doubler.

The contours of the principal shearing stress in the specimen with the titanium doubler are shown in figure 15. By comparing this figure with figure 10 it can be seen that the doubler tended to reduce the region of nearly uniform shearing stress since the contours in figure 15 are closer to the center of the panel. As expected there is also an appreciable local disturbance in the shearing stresses in the vicinity of the doubler.

CONCLUDING REMARKS

Two finite element analyses of a picture frame shear test of a borsic aluminum test specimen have been performed. Two methods for modeling the specimen test frame have been investigated. Results for nominal stresses and principal shear stress have been presented for Test 560, Run 7 conducted at Langley Research Center.

There were striking differences in the stress distributions predicted by the rigid (NASTRAN) and deformable (SAP) boundary models. It was found that it is not realistic to assume the test fixture to be a rigid frame. In the regions of nearly uniform stress, the stresses predicted by the deformable boundary models were approximately one third of the stresses predicted by the rigid boundary model. In the vicinity of the corner, the stresses predicted by the two models nearly coincided.

The constant principal shear stress, $\tau_{\text{max}}$, was uniform over only a very small region in the center of the shear panel specimen. Moreover, at the corners near the fillets, there were steep gradients with stresses being highly concentrated.
The effect of a local reinforcing titanium doubler has been evaluated. It was found that the doubler reduced the maximum nominal stress in the vicinity of the fillet by about 50 percent.
REFERENCES


Figure 1. Picture Frame Shear Test Experimental Setup at Langley Research Center.
Figure 2. Schematic of Test Specimen.
\[ P_1 = 5039.4 \text{ lb} \]

Node for load application constrained to rigid boundary

490 membrane elements

Typical boundary node constrained to represent rigid test frame

\[ P_2 = 5004.9 \text{ lb} \]

Figure 3. NASTRAN Finite Element Model of Shear Panel with Rigid Boundary.
Figure 4. SAP Finite Element Model of Shear Panel (Deformable Boundary).
Figure 5. Normal Stress $\sigma_x$ as a Function of Horizontal Coordinate Along Center Line of Shear Panel
Figure 6. Normal Stress $\sigma_x$ as a Function of Vertical Coordinate Along Center Line of Shear Panel Specimen.
Figure 7. Normal Stress $\sigma_y$ as a Function of Horizontal Coordinate Along Center Line of Shear Panel Specimen.
Figure 8. Normal Stress $\sigma_y$ as a Function of Vertical Coordinate Along Center Line of Shear Panel Specimen.
Figure 9. Contours of Constant Principal Shear Stress, $\tau_{max}$ Predicted by NASTRAN (Rigid Boundary).
Figure 10. Contours of Constant Principal Shear Stress, $T_{\text{max}}$. Predicted by SAP (Deformable Boundary).
Figure 11. Normal Stress $\sigma_x$ as a Function of Horizontal Coordinate Along Center Line of Shear Panel Specimen.
Figure 12. Normal Stress $\sigma_x$ as a Function of Vertical Coordinate Along Center Line of Shear Panel Specimen.
the centroidal coordinate of the finite elements bordering the hole. Comparison of figures 6 through 8 and figure 10 shows that qualitatively the finite element analysis of the anistropic composite and the isotropic elasticity solution are in close agreement. This agreement serves to validate the finite element solution.

The variation of the longitudinal membrane force in an isotropic infinite medium is shown in figure 11 in terms of the $x$ coordinate of the composite specimen to facilitate comparison with the finite element solution given in figure 9. The elasticity solution shows an extremely sharp gradient for the membrane force in the vicinity of the hole. This sharp variation raises questions about the accuracy of the finite element solution in this region. Since the NASTRAN finite element assumed constant stress within the element, it is possible that the peak stress was underestimated because not enough elements were used to accurately represent the stress gradient. The variation of the stress away from the hole according to the isotropic solution shows that in a distance of about five radii ($5a = 0.48$ in.) away from the hole the force has decreased to one-tenth of its maximum value. This result supports the findings of figures 6 through 8 in which the membrane force distributions in the center and outside holes were very nearly the same. This occurred because there were no hole interaction effects since the holes were more than five radii apart. Only very small edge effects were present for the same reason.

CONCLUDING REMARKS

A finite element analysis of an extra graph bolted joint specimen has been performed. Two methods were used to represent bolt transfer loads. The first method assumed a perfect fit and modeled the bolt loading as a cosine distribution over one-half of the boundary of the hole. The second method assumed an imperfect fit and used a nonlinear computer analysis to determine the contact area and bolt transfer loads. The
Figure 13. Normal Stress $\sigma_y$ as a Function of Horizontal Coordinate Along $y$-Center Line of Shear Panel Specimen.
Figure 14. Normal Stress $\sigma_y$ as a Function of Vertical Coordinate Along $y$ Center Line of Shear Panel Specimen.
Figure 15. Contours of Constant Principal Shear Stress, $T_{max}$, Predicted by SAP (Deformable Boundary with Reinforcing Titanium Doubler).
BANSAP: A BANDWIDTH REDUCTION PROGRAM FOR SAP IV

By

Donna E. Holzmacher
BANSAP: A BANDWIDTH REDUCTION PROGRAM FOR SAP IV

By

Donna E. Holzmacher

INTRODUCTION

For analysis, a structure may be broken down into parts known as finite elements. The elements of the structure may be one-dimensional such as a rod, two-dimensional such as a triangle, or quadrilateral, or three-dimensional such as a parallelepiped. The elements are positioned and described by nodes which, when connected, describe the structure. Static analysis using finite elements is accomplished by solving simultaneous equations. These equations when written in matrix form are characterized by banded coefficient matrices. Computer time and storage can be saved if the bandwidth of the matrix is a minimum. This occurs with adept numbering of the nodal points of the structure. If the nodes are numbered in an optimum way the non-zero values in the matrix will lie in a band about the diagonal. The bandwidth of a matrix is defined here as the maximum difference between any two connected nodes plus one to take into account the diagonal term.

As a particular example, consider the plane structure shown in figure 1. The displacements of this structure are determined by solving

\[(K) \{U\} = \{P\}\]

where \((K)\) is the stiffness matrix, \(\{U\}\) is the displacement vector, and \(\{P\}\) is the load vector. The \((K)\) matrix is arranged according to the connectivity of the nodes 1 through 9 of the triangular elements. The connectivity matrix for the above structure is represented in figure 2 showing that node 1 is connected to nodes 2, 8, and 9, and node 2 is connected to...
nodes 1, 2, 3, 4, and 9, etc. The actual values in the stiffness matrix, corresponding to the positions of the matrix above, depend on the geometry and material of the structure.

The bandwidth of the connectivity matrix shown in figure 2 is 9. If the nodes are renumbered as in figure 3, the corresponding connectivity matrix as shown in figure 4 has a new, reduced bandwidth of 4.

In order to efficiently renumber the nodes of structures for finite element analysis a number of algorithms have been developed and incorporated into bandwidth reduction programs. Prior to 1969, authors who developed techniques to reduce the bandwidths of matrices included Always and Martin, Tewarson, Rosen, and Akyus and Utku (refs. 1 through 4). In 1969, Cuthill and McKee's (ref. 5) algorithm arranged the rows of the connectivity matrix with regard to the increasing number of non-zero off-diagonal elements. This algorithm was used in a program called BANDIT which serves as a preprocessor for NASTRAN.

H.R. Groom's algorithm for bandwidth reduction was introduced in 1972 (ref. 6). Groom systematically moved closer together rows and columns which were far apart and coupled.

In 1973, Collins (ref. 7) presented the algorithm upon which the program, BANSAP, developed in this study is based. After work on BANSAP had begun, Rodrigues (ref. 8) presented a new algorithm which, for two sample problems presented, showed a smaller bandwidth than the Cuthill and McKee, the Groom, or the Collins algorithms.

The objective of this paper is to describe a study undertaken to incorporate the Collins bandwidth algorithm in a data preprocessing computer program for the finite element program SAP IV (Structural Analysis Program - IV). First to be presented will be Collins' algorithm for bandwidth reduction which contains two subroutines, SETUP and OPTNUM. A description of the SAP IV preprocessing program BANSAP will then be given including its capabilities and limitations. Results from application of the
program to example problems will be presented and discussed. User instructions, the BANSAP program listing, and sample output are presented in appendices.

COLLINS BANDWIDTH REDUCTION ALGORITHM

Collins' algorithm for bandwidth reduction includes two subroutines, SETUP and OPTNUM. His procedure shall be illustrated using the structure in figure 1.

In the first subroutine, SETUP, a list is generated showing the connections between the different nodes shown in figure 1. The relations established by SETUP are displayed in table 1. The information is stored in arrays suitable for use in subroutine OPTNUM. The subroutine SETUP also determines the original bandwidth of the structure.

The subroutine used to renumber related nodes is OPTNUM. OPTNUM locates the origin of the different numbering schemes at each node in turn, making the number of permutations of schemes equal to the number of nodes. In other words, OPTNUM first renumbers the nodes around old node number one making old node number one the origin of the new scheme. OPTNUM then determines the bandwidth of this scheme. Next OPTNUM goes to old node number two and starts its new origin in the position of this node. It renumbers the nodes connecting node two, one at a time, and determines the maximum difference between the new connected nodes. If the maximum difference is less than the lowest maximum difference of the preceding schemes, it continues with the renumbering until the scheme is complete. If not, the current scheme is abandoned. After completion or abandonment of a scheme, OPTNUM proceeds to the next scheme starting with a new origin at the next sequential old node number. The scheme which is retained by OPTNUM is that which exhibits the lowest maximum difference between related nodes. The sequence of renumbering schemes for figure 1 is shown in table 2.
Collins' algorithm is set up to handle the renumbering of nodes for elements containing up to four nodes. Reference 5 indicates that this method has been applied to solid elements but not very successfully.

**SAP IV PREPROCESSING PROGRAM**

A program, BANSAP, has been written using the Collins algorithm as a preprocessing program for SAP IV. BANSAP consists of four subroutines: SAPIN, SETUP, OPTNUM, and SAPOUT as shown in figure 5.

The first subroutine, SAPIN, reads the data in the formats stipulated by SAP IV and stores element and node connections according to type. BANSAP is set up to handle two basic types of finite elements: elements connecting two nodes, and elements connecting three or four nodes. The two node elements which can be entered into subroutine SAPIN are either the truss, beam, or boundary. The actual renumbering of a two node element is the same for either element. The only difference in the handling of these elements by BANSAP is in their SAP IV formats. The three or four node elements which may be entered into subroutine SAPIN include membranes, axisymmetric two-dimensional elements and plate bending elements. Again, the only difference in the handling of these three and four node elements is in their SAP IV formats. If more than one type of element comprises the structure, the elements may be grouped according to their type. As is required for SAP IV, nodes must be sequentially numbered from one.

From subroutine SAPIN, BANSAP goes on to subroutines SETUP and OPTNUM. The new bandwidth is printed and a list of old number node numbers and new numbers is generated. As a user option the subroutine SAPOUT will punch the original elements with the new node numbers. Program BANSAP has been dimensioned in this paper to permit up to 1000 nodes and 1000 elements.
APPLICATIONS OF BANSAP

Applications of BANSAP are presented in table 3. The first two problems shown are illustrations of reduction in bandwidth which may be attained for simple problems. Problem 3 is an example taken from structural analysis of a ship radar tower. The last two entries are practical problems encountered in finite analysis of composite material structures.

The first illustration is the sample finite element scheme shown in figure 1. After renumbering by BANSAP the bandwidth was reduced from 9 to 4 and the final scheme is shown in figure 3.

The truss problem shown in figure 6a is a wagonwheel. After processing by BANSAP the bandwidth was reduced from 9 to 6. It has been found, however, that this value is not the optimum bandwidth. Collins has noted that the wagonwheel problem is a special case and the true optimum bandwidth occurs when the node number of the hub of the wheel is set equal to half the number of spokes plus one. The optimum bandwidth of the wagonwheel shown in figure 6 is actually 5.

The third structure is the ship's radar tower shown in figure 7. The original numbering scheme shown is nearly optimum with a bandwidth of 12 since the renumbering scheme only reduces the bandwidth to 9. For such structures there is no appreciable gain by using BANSAP as the structures could easily be numbered by hand to obtain a small bandwidth.

The shear panel of figure 8 is an example of a greatly enlarged bandwidth which can occur from the addition of new finite elements after the original structure has been numbered. With the addition of new elements for the shear panel a bandwidth of 406 was obtained, but after BANSAP, the bandwidth was reduced to 35.

The bolted joint specimen illustrated in figure 9 is a good example of how BANSAP can be used to obtain an optimum bandwidth when the numbering scheme is difficult to select by hand. The
nodes of the bolted joint specimen were originally numbered to permit easier data generation using a FORTRAN program. After the cards had been generated, BANSAP renumbered the nodes to reduce the bandwidth from 168 to 28.

CONCLUDING REMARKS

A FORTRAN program has been written for bandwidth reduction by nodal renumbering. The program is based upon the Collins algorithm and serves as a data preprocessor for the finite element program SAP IV. Applications of the preprocessing program to a number of simple and realistic problems have been presented.

Nodal renumbering for finite element analysis may be required for a variety of reasons. Renumbering may be needed if new elements were to be added onto a previously numbered structure or if a structure is difficult to optimally number by hand. It may also be needed if the element and nodal data were prepared by data generation programs. Such reasons clearly show a need and use for a program such as BANSAP.

BANSAP is an effective preprocessing program for SAP IV. The algorithm used greatly reduces the bandwidth for reduced computer time and storage during the finite element analysis.
APPENDIX A

USER INSTRUCTIONS
USER INSTRUCTIONS

CONTROL CARD (3I5)

Columns  1  -  5  Number of different groups of elements
         6  -  10  Total number of nodes
         11 -  15  The number zero for nopunched output and
                     any number greater than zero for punched output

The following types of elements are permitted in the program.

Type 1  TRUSS

CONTROL CARD (3I5)

Columns  1  -  5  The number 1
         6  -  10  The number of elements in group 1

Element Data Cards (3I5,  2A10)

Columns  1  -  5  Element number
         6  -  10  Node number I
         11 -  15  Node number J

Type 2  BEAM

CONTROL CARD (3I5)

Columns  1  -  5  The number 2
         6  -  10  The number of elements in group 2

Element Data Cards (4I5,  5A10)

Columns  1  -  5  Element number
         6  -  10  Node number I
         11 -  15  Node number J
         16 -  20  Node number K; K is any nodal point which
                     lies in the local 1-2 plane but not on
                     the 1 axis (see ref. 9, page iv.2.2)

Type 3  MEMBRANE

CONTROL CARD (3I5)

Columns  1  -  5  The number 3
         6  -  10  The number of elements in group 3

Element Data Cards (5I5,  5A10)

Columns  1  -  5  Element number
         6  -  10  Node number I
         11 -  15  Node number J
         16 -  20  Node number K
         21 -  25  Node number L
Type 4  TWO D

CONTROL CARD (3I5)

Columns 1 - 5  The number 4
6 - 10  The number of elements in group 4

Element Data Cards (5I5, 5A10)

Columns 1 - 5  Element number
6 - 10  Node number I
11 - 15  Node number J
16 - 20  Node number K
21 - 25  Node number L

Type 6  PLATE

CONTROL CARD (3I5)

Columns 1 - 5  The number 6
6 - 10  Number of plate elements

Element Data Cards (5I5, 5A10)

Columns 1 - 5  Element number
6 - 10  Node number I
11 - 15  Node number J
16 - 20  Node number K
21 - 25  Node number L

Type 7  BOUNDARY (LINEAR SPRING)

CONTROL CARD (3I5)

Columns 1 - 5  The number 7
6 - 10  The number of elements in group 7.

Element Data Cards (2I5, 6A10)

Columns 1 - 5  Node N, at which the element is placed
6 - 10  Node I
APPENDIX B

`BANSAP SOURCE LISTING`
PROGRAM BANSAP(INPUT, OUTPUT, PUNCH, * TAPES=INPUT, TAP6=OUTPUT, TAP7=PUNCH)

DIMENSION NEWJ(T(1000)), JOINT(1000)

COMMON LMENTS, JT(4000), MEMJT(9000), JMEM(1000), JNT(1000)

COMMON /BAND /IDIFF, MINMAX

COMMON /CONTR/ NELG, I_TYPE(5), NEL(5), NODES, IPUNCH

COMMON /JUNK/A(1000, 6)

COMMON /UNIT/ IN, IT, IP

IN = 5

IT = 6

IP = 7

JMEM(I) = NUMBER OF NODES TO WHICH A SINGLE NODE IS CONNECTED

JT(I) = WORKING ARRAY

-MEMJT(I) = IDENTITIES OF NODES TO WHICH A NODE IS CONNECTED

WRITE(IT, 12)

12 FORMAT(IH1, 9(/),

1 36X, 52HB88BB AAAA N N SSSSS AAAA PPPP/
2 36X, 53HB B A A NN N S A A P P/
3 36X, 53HB B A A NN N S A A P P/
4 36X, 52HB88BB AAAAAAA N N N SSSSS AAAAAA PPPP/
5 36X, 52HB B A A NN N S A A P P/
6 36X, 52HB B A A NN N S A A P P/
7 36X, 52HB88BB A A N N SSSSS A A P P/

WRITE(IT, 16)

16 FORMAT(IH1, 5X, 19H INPUT DATA, //)

DO 10 I = 1, 1000

10 JNT(I) = 0

DO 20 I = 1, 4000

20 JT(I) = 0

DO 30 I = 1, 8000

30 MEMJT(I) = 0

CALL SAPIN

SUBROUTINES SETUP AND OPTNUM FROM:
-BANDWIDTH REDUCTION BY AUTOMATIC RENUMBERING-, R. J. COLLINS,
INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING

CALL SETJP
C

WRITE(IT,32)
DO 40 I= 1,NODES
NO = JMEM(I)
LI = 8*(I-1) + 1
L2 = LI + NO -1
WRITE(IT,34) I,NO,(MEMJT(L),L=L1,L2)
40 CONTINUE
MINMAX = IDIFF + 1
WRITE(IT,36) MINMAX

CALL OPTNUM
MINMAX = MINMAX +1
WRITE(IT,38) MINMAX
WRITE(IT,42)

CALL SAPOUT

32 FORMAT(1H1,12X,4HNODE,3X,4HJMEM,16X,5HMEMST)///
34 FORMAT(11X,2I5,10X,9I6)
36 FORMAT(///,20X,20HORIGINAL BANDWIDTH =,I4 )
38 FORMAT(///,20X,14HNEW BANDWIDTH =,I4)///
42 FORMAT(1H4,10X,33HOLD NODE NUMBER NEW NODE NUMBER,/) STOP
00121 END
SUBROUTINE SAPIN
COMMON LMENTS,JT(4000),MEMJ(8000),JMEM(1000),JNT(1000)
COMMON/BAND/IDIFF,MINMAX
COMMON/CONTR/ NELG,ITYPE(5),NEL(5),NODES,IPUNCH
COMMON/JUNK/A(1000,6)
COMMON/UNIT/ IN,IT,IP

NELG = NUMBER OF DIFFERENT GROUPS OF ELEMENTS (LESS THAN 5
NODES = TOTAL NUMBER OF NODES
IPUNCH = ZERO FOR NO PUNCHED OUTPUT, NUMBER GREATER THAN
ZERO FOR PUNCHER OUTPUT
N = ELEMENT TYPE
NE = NUMBER OF ELEMENTS OF TYPE N
LMNT = TOTAL NUMBER OF ELEMENTS

ITYPE(I) = TYPE OF ELEMENT
NEL(I) = NUMBER OF ELEMENTS IN A GROUP

ELEMENT NUMBER OF NODES
1 TRUSS 2
2 BEAM 2
3 MEMBRANE 3 OR 4
4 TWO D 3 OR 4
5 BRICK 8
6 PLATE 4
7 BOUNDARY 2

READ ELEMENT CARDS AND STORE CONNECTIONS.

READ(IN,12) NELG,NODES,IPUNCH
WRITE(IT,14),NELG,NODES,IPUNCH
DO 200 II= 1,NELG
READ(IN,12) N,NE
WRITE(IT,13) II,NE,N
WRITE(IT,50)
ITYPE(I) = N
NEL(I) = NE
LMENTS = 0

READ ELEMENT CONNECTIONS. FOR TRUSS, BEAM, OR BOUNDARY
ELEMENTS ONLY TWO CONNECTIONS I AND J ARE NEEDED. FOR
ALL OTHER TYPES FOUR CONNECTIONS ARE POSSIBLE— I,J,K,L.

STORE NODE CONNECTIONS ACCORDING TO TYPE.

DO 210 JJ = 1, NE
  GO TO (1,2,3,3,5,3,7), N

1 CONTINUE
  READ(IN,102) I,J,( A(JJ,L),L=1,2)
  102 FORMAT(5X,2I5,2A10)
  GO TO 300

2 CONTINUE
  READ(IN,104) I,J,( A(JJ,L),L=1,6)
  104 FORMAT(5X,2I5,5A10)
  GO TO 300

3 CONTINUE
  READ(IN,106) I,J,K,L,( A(JJ,L),L=1,5)
  106 FORMAT(5X,4I5,5A10)
  GO TO 300

5 CONTINUE
  WRITE(IT,128)
  128 FORMAT(5X,4I5,2THREE DIMENSIONAL ELEMENTS NOT IMPLEMENTED),/
  GO TO 200

7 CONTINUE
  READ(IN,110) I,J,( A(JJ,L),L=1,6)
  110 FORMAT(2I5,6A10)
  300 CONTINUE
  IF(N.EQ.7) GO TO 20
  20 CONTINUE
  III = JJ + LMENTS
  JJJ = III + 1000
  JT(III) = I
  JT(JJJ) = J
  IF(N.LE. 2) GO TO 205
  KKK = III + 2000
  LLL = III + 3000
  JT(KKK) = K

FOR TRIANGULAR ELEMENT SET REPEATED NODE NUMBER EQUAL TO ZERO.
000234 IF(K.EQ.L) L=0
000236 JT(LLL) = L
000240 205 CONTINUE
000240 WRITE(IT,30) JJ, JT(III), JT(JJJ), JT(KKK), JT(LLL)
000256 210 CONTINUE
000261 LMENTS = LMENTS + NE
000262 200 CONTINUE
000265 50 FORMAT(1X,5HELEMENT,5X,1HI,7X,1HJ,7X,1HK,7X,1HL,/)  
000265 10 FORMAT(/,5X,13HELEMENT GROUP,12,4H HAS,13,17H ELEMENTS OF TYPE,  
1/) 
000265 12 FORMAT(3I5)  
000265 14 FORMAT(2X)  
1 10X,25HNUMBER OF ELEMENT TYPES = ,I5,/  
2 10X,25HNUMBER OF NODAL POINTS =,I5,-/  
3 10X,25HPUNCCHED ELEMENT CARDS =,I5,/)  
4 10X,25H*EQ. 0 NO ;/  
5 10X,25H*EQ. 1 YES ;)  
000265 30 FORMAT(10X,15,4I8) 
000265 RETURN  
000266 END.
SUBROUTINE SETUP

COMMON LMENTS, JT(4000), MEMJT(8000), JMEM(1000), JNT(1000)
COMMON/BAND/IDIFF, MINMAX
COMMON/CONTR/NELG, ITYPE(5), NEL(5), NODES, IPUNCH

C
C NODES = TOTAL NUMBER OF NODES
C JNTI = ELEMENT NODE UNDER CONSIDERATION
C JSUB = LOCATION IN MEMJT(I) OF BEGINNING OF LIST OF
C NOD E S RELATED TO JNTI
C LMENT = TOTAL NUMBER OF ELEMENTS
C JMEM(I) = NUMBER OF NODES TO WHICH A SINGLE NODE IS CONNECTED
C MEMJT(I) = IDENTITIES OF NODES TO WHICH A NODE IS CONNECTED
C IDIFF = BANDWIDTH = IDIFF+1 FOR ORIGINAL SCHEME

000002 IDIFF = 0
000003 DO 10 J= 1, NODES
000005 10 JMEM(J) = 0
000011 DO 60 J= 1, LMENTS
000012 DO 50 I= 1, 4
C NEXT STATEMENT DEPENDS ON THE NUMBER OF NODES FOR WHICH THE
C PROGRAM IS DIMENSIONED. CURRENTLY THE MAXIMUM NUMBER OF NODES
C IS 1000.
000013 JNTI = JT(1000* (I-1) + J)
C IF JNTI EQUALS ZERO ALL NODES OF ELEMENT J HAVE BEEN
C CONSIDERED.
000017 IF(JNTI.EQ.0) GO TO 60
000021 JSUB = (JNTI - 1) * 8
000023 DO 40 II = 1, 4
000025 IF(II.EQ.1) GO TO 40
C NEXT STATEMENT DEPENDS ON THE NUMBER OF NODES FOR WHICH THE
C PROGRAM IS DIMENSIONED. CURRENTLY THE MAXIMUM NUMBER OF NODES
C IS 1000.
000027 JJT = JT(1000* (II-1) + J )
000033 IF(JJT.EQ.0) GO TO 50
DETERMINE WHETHER RELATIONSHIP BETWEEN JNTI AND JJT HAS BEEN ESTABLISHED.

MEM1 = JMEM(JNTI)
IF(MEM1.EQ.0) GO TO 30
DO 20 III =1, MEM1
   IF(MEMJT(III).EQ.JJT) GO TO 40
20 CONTINUE

FIND WIDTH OF ORIGINAL MATRIX BAND.

JMEM(JNTI) = JMEM(JNTI) + 1
IDUM = JSUB + JMEM(JNTI)
MEMJT(IDUM) = JJT
IF(IABS(JNTI-JJT).GT.IDIFF) IDIFF = IABS(JNTI-JJT)
40 CONTINUE

RETURN

END
SUBROUTINE OPTNUM
000002 DIMENSION NEWJT(1000),JOINT(1000)
000002 COMMON LMENTS, JT(4000), MEMJT(8000), JMEM(1000), JNT(1000)
000002 COMMON/BAND/IDIFF,MINMAX
000002 COMMON/CONTR/NELG,ITYPE(5),NEL(5),NODES,IPUNCH
000002 COMMON/UNIT/ IN,IT,IP
C
C JOINT(J) = WORKING ARRAY
C NEWJT(J) = WORKING ARRAY
C JNT(J) = NEW NUMBERING SCHEME
C --MINMAX = BANDWIDTH = MINMAX+1
C
MINMAX IS INITIALIZED.
000002 MINMAX = IDIFF
C
C NEW SCHEME STARTS AT NODE OF OLD NODE NUMBER IK.
000004 DO 60 IK=1,NODES
000005 DO 20 JV=1,NODES
C FOR NEW SCHEME
C JOINT(J) AND NEWJT(J) INITIALIZED TO ZERO FOR EACH NEW NUMBERING SCHEME.
000006 JOINT() = 0
000007 NEWJT(J) = 0
C INITIALIZE FOR NEW NODE NUMBER ONE.
000013 MAX = 0
000013 I = 1
000014 NEWJT(1) = IK
000016 JOINT(IK) = 1
000017 K = 1
000021 CONTINUE
000021 JDUM = NEWJT(I)
000023 K4 = JMEM(JDUM)
000025 IF(K4.EQ.0) GO TO 45
C LOCATE RELATED NODES IN MEMJT(I).
000026 JSUB = (NEWJT(I) -1) # 8

DO 40 JJ= 1,K4

K5 = MEMJT(JSUB +JJ)

IF( JOINT(K5) .GT. 0 ) GO TO 40

K = K+1

NEWJT(K) =K5

IF(JOINT(K5)=K

CHECK DIFFERENCE BETWEEN NEW NUMBERS OF RELATED NODES.

NDIFF = IABS(I-K)

SCHEME ABANDONED IF DIFFERENCE GREATER THAN BANDWIDTH OF
PREVIOUS SCHEME, NEW SCHEME STARTED.

IF(NDIFF.GE.MINMAX) GO TO 60

IF(NDIFF.GT.MAX) MAX =NDIFF

40 CONTINUE

IF(K.EQ.NODES)GO TO 50

45 I=I+1

GO TO 30

50 MINMAX = MAX

55 JNT(J) = JOINT(J)

60 CONTINUE

RETURN

END
SUBROUTINE SAPOUT
COMMON LMENTS, JT(4000), MEMJT(8000), JMEM(1000), JNT(1000)
! COMMON COMMON/CCNTR/ NELG, ITYPE(5), NEL(5), NODES, IPUNCH
! COMMON/JUNK/A(1000,6)
! COMMON/UNIT/IN, IT, IP
DO 10 I = 1, NODES
WRITE(IT,12) I, JNT(I)
10 CONTINUE
LMENTS = 0
WRITE(IT,14)
DO 20 II = 1, NELG
N = ITYPE(II)
NE = NEL(II)
DO 21 JJ = I, NE
IF(NELG.GT.0) WRITE(IP,102) JJ, NI, NJ, (A(JJ,L), L = 1, 2)
21 CONTINUE
20 CONTINUE
GO TO 1, 2, 3, 5, 3, 7), N
C C OUTPUT NODE CONNECTIONS ACCORDING TO TYPE.
C C 1 CONTINUE
1 CONTINUE
WRITE(IT,12) JJ, NI, NJ, (A(JJ,L), L = 1, 2)
IF(IPUNCH.GT.0) WRITE(IP,102) JJ, NI, NJ, (A(JJ,L), L = 1, 2)
102 FORMAT(315,2A10)
GO TO -21
C 2 CONTINUE
2 CONTINUE
K = A(JJ,1)
WRITE(IT,104) JJ, NI, NJ, NK, (A(JJ,L), L = 2, 6)
IF(IPUNCH.GT.0) WRITE(IP,104) JJ, NI, NJ, NK, (A(JJ,L), L = 2, 6)
104 FORMAT(415,5A10)
GO TO 21
C 3 CONTINUE
3 CONTINUE
K = JT(JJ+LMENTS +2000)
L = JT(JJ+LMENTS +3000)
IF(L.EQ.0) L=K
NK= JNT(K)
NL= JNT(L)
WRITE(IT,106) JJ, NI, NJ, NK, NL, (A(JJ,L), L=1,5)

106 FORMAT(5I5,5A10)

IF(IPUNCH.GT.0) WRITE(IP,106) JJ, NI, NJ, NK, NL, (A(JJ,L), L=1,5)
GO TO 21

C
5 CONTINUE
GO TO 21
C
7 CONTINUE
WRITE(IT,108) NI, NJ, (A(JJ,L), L=1,6)
IF(IPUNCH.GT.0) WRITE(IP,108) NI, NJ, (A(JJ,L), L=1,6)

108 FORMAT(2I5,6A10)

21 CONTINUE
LMENTS = LMENTS +NE
20 CONTINUE
12 FORMAT(15X,I5,13X,I5)
14 FORMAT(1H1,4X,17HNEW ELEMENT CARDS,/
30 FORMAT(5I5)
RETURN
END
APPENDIX C

SAMPLE BANSAP OUTPUT
**INPUT DATA**

NUMBER OF ELEMENT TYPES = 1  
NUMBER OF NODAL POINTS = 9  
PUNCHED ELEMENT CARDS = 0  
  EQ. 0 NO  
  EQ. 1 YES  

ELEMENT GROUP 1 HAS 8 ELEMENTS OF TYPE 3  

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>-0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>-0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>-0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>-0</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>-0</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>9</td>
<td>6</td>
<td>-0</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>5</td>
<td>6</td>
<td>-0</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>-0</td>
</tr>
<tr>
<td>NODE</td>
<td>JMEM</td>
<td>MEMJT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ORIGINAL BANDWIDTH = 0.9

NEW BANDWIDTH = 4.
<table>
<thead>
<tr>
<th>OLD NODE NUMBER</th>
<th>NEW NODE NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>
## NEW ELEMENT CARDS

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>
REFERENCES


Table 1. Example node connections determined in subroutine SETUP.

<table>
<thead>
<tr>
<th>Node</th>
<th>Number of Connected Nodes</th>
<th>Connected Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>9, 8, 2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1, 9, 4, 3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2, 4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2, 9, 3, 5</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>9, 6, 4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>8, 7, 9, 5</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>8, 6</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1, 9, 6, 7</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1, 8, 2, 4, 6, ?</td>
</tr>
</tbody>
</table>
Table 2. Trail Numbering Schemes Used in OPTNUM.

<table>
<thead>
<tr>
<th>OLD NODE NUMBER AT WHICH ORIGIN OF NEW NUMBERING SCHEME IS SET</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>DIFF = 3</td>
<td>DIFF = 4</td>
<td>DIFF = 2</td>
<td>DIFF = 4</td>
<td>DIFF = 3</td>
<td>DIFF = 4</td>
<td>DIFF = 2</td>
<td>DIFF = 4</td>
<td>DIFF = 3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>DIFF = 5</td>
<td>DIFF = 3</td>
<td>ABANDON</td>
<td>ABANDON</td>
<td>SCHEME</td>
<td>ABANDON</td>
<td>SCHEME</td>
<td>ABANDON</td>
<td>SCHEME</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>DIFF = 5</td>
<td>ABANDON</td>
<td>SCHEME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>NEW BANDWIDTH</td>
<td>= DIFF + 1</td>
<td>= 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>DIFF = 1</td>
<td>ABANDON</td>
<td>SCHEME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>NEW BANDWIDTH</td>
<td>= DIFF + 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>DIFF = 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DIFF = LARGEST DIFFERENCE BETWEEN ANY TWO RELATED NODES.
Table 3. Summary of applications of BANSAP.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Element Type</th>
<th>Number of Nodes</th>
<th>Number of Elements</th>
<th>Old Bandwidth</th>
<th>New Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample problem</td>
<td>Membrane</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Figure 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wagonwheel</td>
<td>Truss</td>
<td>9</td>
<td>16</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Figure 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ship tower</td>
<td>Beam</td>
<td>25</td>
<td>65</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>Figure 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shear panel specimen</td>
<td>Membrane</td>
<td>595</td>
<td>554</td>
<td>406</td>
<td>35</td>
</tr>
<tr>
<td>Figure 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bolted joint specimen</td>
<td>Membrane</td>
<td>398</td>
<td>349</td>
<td>168</td>
<td>28</td>
</tr>
<tr>
<td>Figure 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 1. Sample Finite Element Scheme

Figure 2. Connectivity Matrix of Sample Scheme.
Figure 3. Renumbered Finite Element Scheme.

Figure 4. Connectivity Matrix of Renumbered Scheme.
SUBROUTINE SAPIN
READS AND STORES INPUT DATA

SUBROUTINE SETUP
IDENTIFIES RELATED NODES AND ESTABLISHES ORIGINAL BANDWIDTH

SUBROUTINE OPTNUM
GOES THROUGH PERMUTATIONS OF POSSIBLE RENUMBERING SCHEMES TO OBTAIN SMALLEST BANDWIDTH

SUBROUTINE SAPOUT
OUTPUTS NEW NUMBERING SCHEME

STOP

Figure 5. Flowchart of SAP IV BANSAP Preprocessing Program.
Figure 6. Wagonwheel Truss.
Figure 7. Ship Radar Tower.
Figure 8. Shear Panel Specimen.
Figure 9: Finite Element Mesh for Composite Bolted Joint Specimen.
FEMESH: A FINITE ELEMENT MESH GENERATION PROGRAM
BASED ON ISOPARAMETRIC ZONES

By

Zoa C. Lane
FEMESH: A FINITE ELEMENT MESH GENERATION PROGRAM
BASED ON ISOPARAMETRIC ZONES

By
Zoa C. Lane

INTRODUCTION

Finite element analysis programs greatly facilitate the determination of deformations and stresses in structures. A major inconvenience in utilizing this analysis technique is the large amount of input data required by the computer programs. This data includes, in addition to material characteristics, the node numbers defining the elements and the spatial coordinates for each node.

Current mesh generation methods include for simple problems data preparation by hand, and for more complex problems, the coding and executing of FORTRAN mesh generation programs which generate data for a general structure.

W.R. Buell and W.A. Bush surveyed some techniques used in current mesh generation schemes (ref. 1). The techniques presented by Buell and Bush are: a straight line interpolation technique, a sides and parts technique for axisymmetric structures, electro-mechanical techniques for two- and three-dimensional structures, and a simplified finite difference technique and equipotential technique for general structure shapes.

The advantages of general structure mesh generation programs (ref. 1) are: (1) reduced cost due to reduction of man hours and computer time needed to generate and check data; (2) reduced number of errors; (3) insured regularity of finite elements; and (4) application to a variety of structural shapes.

O.C. Zienkiewicz (ref. 2) utilizes a technique involving the mapping of isoparametric quadrilaterals from a natural to a cartesian coordinate system in an automatic mesh generation
scheme for plane and curved surfaces. This scheme is applicable to non-quadrilateral structures if the structure is divided into quadrilateral regions. Zienkiewicz's technique for mesh generation was used by S.J. Womack (ref. 3) as a preprocessor for TEXGAP, a finite element program for the analysis of two-dimensional linearly elastic plane or axisymmetric bodies (ref. 4).

The objective of this study is to utilize the technique developed by Zienkiewicz in a mesh generation scheme for two-dimensional planar surfaces. Presented in this paper are a description of the mapping technique, a description of the computer program; and three examples of meshes generated by the program. A set of user instructions and a listing of the program are included in the appendices.

INTERPOLATION FUNCTION TECHNIQUE FOR FINITE ELEMENT GENERATION.

The algorithm used by Zienkiewicz to map an isoparametric quadrilateral is the displacement-interpolation equations used in isoparametric finite elements (ref. 5). The interpolation equations for quadratic bounded surfaces (which are listed in table 1), are a function of a set of dimensionless coordinates, \( \xi \) and \( \eta \), which define a natural coordinate system.

In the natural coordinate system (fig. 1), a planar surface is represented as a square whose dimensions are 2 x 2 units and whose center is at the origin. To map a surface into the cartesian coordinate system, eight boundary points \((x_i, y_i)\) and the \( \xi \) and \( \eta \) values of each grid point on the surface to be mapped are substituted in the displacement-interpolation functions; the resulting values are the cartesian coordinates of the grid points.

A mesh is generated by dividing the square into the desired number of subdivisions, calculating the \( \xi \) and \( \eta \) coordinates for each grid point, and mapping each point to the cartesian coordinate system. A graduation of a generated mesh is obtained
by offsetting the midside node from the midpoint of a side of the quadrilateral (fig. 2). The generated elements will vary in size along that side; smaller elements will be in the direction of the offset.

Meshes for complex structures are generated by dividing the structure into quadrilateral zones. The mesh for each zone is generated independent of other zones. Connection of zones is accomplished by eliminating node numbers and coordinates which were duplicated on zone boundaries.

PROGRAM FEMESH

Program FEMESH is a FORTRAN IV code for generating finite element data for two-dimensional planar surfaces. The algorithm used to generate the node coordinates is based on the displacement interpolation functions (table 1) described in the preceding paragraph.

Input data for FEMESH includes a title, the number of zones, the total number of zone nodes, the number of zone node coordinates to be read from cards; the first node and element numbers, a list of the eight nodes which define a zone, the dimensions of the desired mesh of each zone, and the zone node coordinates.

A zone is a quadrilateral region whose geometry is defined by eight zone nodes. (Zone nodes are used only in the input definition of the geometry; they are not included in the generated mesh.) The zone nodes are listed in counter-clockwise order. As indicated in figure 3, the first node identifies a corner of the quadrilateral. The second, fourth, sixth, and eighth nodes are referred to as midside nodes. If a midside node does not lie on the midpoint of a side, a graduation of the mesh results.

The general flow for the mesh generation program, FEMESH, is shown in figure 4. As indicated, the mesh for each zone is generated separately. The first step in the mesh generation scheme is to determine if the coordinates of the midside
nodes are defined (i.e., if their coordinates were supplied by the user). If the coordinates are not defined, the midside node is assumed to lie at the midpoint of a linear line segment. The second step is to determine if either of the four sides of the zone is connected to a zone for which a mesh was previously generated. If a side is connected to such a zone the node numbers and the x and y coordinates which have already been generated are used. The remainder of the mesh is then generated. This process is repeated until the meshes for all zones are generated.

The output of program FEMESH includes a listing of the elements, their four node numbers and the node coordinates. A plot of the mesh is also generated.

APPLICATIONS

Three finite element meshes generated by FEMESH are presented in this section. The first example is a sample problem illustrating the input and output of program FEMESH. The second is a quarter section of a shear panel. The third is a half section of a bolted joint specimen.

The first example is a simple structure originally used to validate the ability of FEMESH to properly connect zones. The structure (illustrated in fig. 5a) is divided into three zones. The eighteen zone nodes are labeled arbitrarily and illustrated in figure 5b. Figures 5a and 5b represent the input required by program FEMESH to generate the mesh illustrated in figures 5c and 5d. Figure 5c illustrates the node numbers, and 5d illustrates the element numbers.

The input data for this problem is tabulated in table 2 (see Appendix A for user instructions). The data includes a title card, a control card, three zone description cards, and eight node coordinate cards. The control card specifies the number zones (3), the number of zone nodes (18), the number of zone node coordinate cards to be read (8), the first node
number (100), and the first element number (1000). A typical zone description card lists the eight zone nodes defining each zone and the size of the finite element mesh to be generated.

The tabulated output for this problem appears in table 3. The output includes the input data, the element number, the four node numbers which define each element, and the cartesian coordinates of each node.

The shear panel illustrated in figure 6 is divided into four zones. The zones were established in such a way that the straight and curved segments of the corner fillets are assigned to different zones in order to obtain a closer approximation of the true boundary shape.

The mesh dimension for zone I is 20 x 20, for zone II is 20 x 3, for zone III is 20 x 20, and zone IV is 3 x 30. To avoid the generation of long, narrow rectangular elements, the midside nodes 2, 8, 9, and 14, 15, 16 are moved away from the midpoint of the line segment toward the fillets. The input data is summarized in table 4. The output is illustrated in figure 7. Because of the large number of generated elements, the output is not listed in tabular form; it is represented graphically by a computer plot of the generated mesh. The generated mesh is composed of 574 nodes and 520 elements.

The mesh for one-half of a bolted joint specimen was generated by dividing the specimen into 15 zones as illustrated in figure 8. The input data for this problem (table 5) consisted of 58 data cards, including 15 zone description cards, and 41 node coordinate cards. A graduation of the mesh of zones II, III, IV, V, VI, VIII, and IX was used to obtain a uniformity in the shape of the generated elements. The generated mesh, which is illustrated in figure 9, consists of 378 elements and 435 nodes.

CONCLUDING REMARKS

Program FEMESH, a FORTRAN IV code, has been developed to generate a finite element mesh for two-dimensional, planar.
surfaces. The algorithm used is the displacement interpolation functions which were developed for mesh generation by Zienkiewicz.

A structure may be subdivided into a maximum of 15 zones. The maximum mesh for each zone is 24 x 24 elements (or 25 x 25 node points). FEMESH will compute a maximum of 4000 node points, and output the node numbers and their coordinates and the element numbers and their four identifying node numbers. A simple plot of the finite element mesh is also generated.

Presented in this paper is a description of the technique used in the mesh generation scheme, a description of program FEMESH and examples of the mesh generated for three problems. User instructions and a listing of the program are included in the appendices.
APPENDIX A

USER INSTRUCTIONS FOR FEMESH
USER INSTRUCTIONS FOR FEMESH

Program FEMESH generates isoparametric finite element meshes for two-dimensional planar surfaces. The input required by the program consists of four types of data cards: a title card, a control card, zone description cards, and node coordinate cards (fig. A1).

TITLE CARD (Format 10A4).

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-40</td>
<td>TITLE</td>
<td>Heading for output</td>
</tr>
</tbody>
</table>

CONTROL CARD (Format 6I5):

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>IZ</td>
<td>Number of zones (IZ &lt; 15)</td>
</tr>
<tr>
<td>5-10</td>
<td>NT</td>
<td>Total number of zone nodes</td>
</tr>
<tr>
<td>11-15</td>
<td>NI</td>
<td>Number of zone node coordinates to be read as input on cards</td>
</tr>
<tr>
<td>16-20</td>
<td>INODE</td>
<td>First node number to be assigned to generated mesh</td>
</tr>
<tr>
<td>21-25</td>
<td>IELM</td>
<td>First element number to be assigned to generated mesh</td>
</tr>
<tr>
<td>26-30</td>
<td>IP</td>
<td>Punch indicator: 0 will not punch 1 punch</td>
</tr>
</tbody>
</table>

A zone is a quadrilateral with either linear or curved line segments. The geometry of the zone is defined by 8 zone nodes whose coordinates are supplied by the user (see node coordinate card).

The values of NI and NT may differ due to the ability of the program to linearly interpolate to define the coordinates of the midside node if those coordinates are not supplied by the user. Midside nodes are those zone nodes which lie between two corner nodes. It is not necessary that a midside node lie at the midpoint of a line segment.
ZONE DESCRIPTION CARD (Format 1015):

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>NODE (I,1)</td>
<td>Zone nodes defining zone geometry</td>
</tr>
<tr>
<td>6-10</td>
<td>NODE (I,2)</td>
<td>I is the zone number</td>
</tr>
<tr>
<td>11-15</td>
<td>NODE (I,3)</td>
<td></td>
</tr>
<tr>
<td>16-20</td>
<td>NODE (I,4)</td>
<td></td>
</tr>
<tr>
<td>21-25</td>
<td>NODE (I,5)</td>
<td></td>
</tr>
<tr>
<td>26-30</td>
<td>NODE (I,6)</td>
<td></td>
</tr>
<tr>
<td>31-35</td>
<td>NODE (I,7)</td>
<td></td>
</tr>
<tr>
<td>36-40</td>
<td>NODE (I,8)</td>
<td></td>
</tr>
<tr>
<td>41-45</td>
<td>M</td>
<td>Number of subdivisions along the side defined by 1st, 2nd, and 3rd zone nodes</td>
</tr>
<tr>
<td>46-50</td>
<td>N</td>
<td>Number of subdivisions along the side defined by 3rd, 4th, and 5th zone nodes</td>
</tr>
</tbody>
</table>

Zone numbers are determined by the order of the zone description cards. The first zone description card is assigned the number one, the second is assigned the number two, etc.

The interconnectivity of zones is indicated by assigning a negative magnitude to zone nodes which lie on a side connected to a zone with a smaller zone number. For example, if 4 zones are connected as shown in figure A2, then the first eight values of the zone description cards should be:

Card 1: 1 2 3 7 11 10 9 6
Card 2: -3 4 5 8 13 12 -11 -7
Card 3: -11 -12 -13 16 21 20 19 15
Card 4: -9 -10 -11 -15 -19 18 17 14

A side which is divided into M subdivisions must not be connected to a side divided into N subdivisions unless the values M and N are equal.
NODE COORDINATE CARD (Format I5, 2F10.5):

<table>
<thead>
<tr>
<th>Column</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>Node number</td>
</tr>
<tr>
<td>5-15</td>
<td>x coordinate</td>
</tr>
<tr>
<td>16-25</td>
<td>y coordinate</td>
</tr>
</tbody>
</table>

This card may be omitted for any midside node which lies on a straight line if a graduation of the mesh is not desired.

A graduation in the mesh occurs when the midside node is offset from the midpoint of the line segment. The smaller elements will be in the same direction as the offset.

Due to a restriction in the FORTRAN coding, a midside node should not be assigned the coordinates (0,0) if the line segment is not a straight line.
<table>
<thead>
<tr>
<th>STATEMENT NUMBER</th>
<th>CONTINUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FORTRAN STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE CARD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FORMAT (1,0,A4)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CONTROL CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>JZ</td>
</tr>
<tr>
<td>NT</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>IELM</td>
</tr>
<tr>
<td>IP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FORMAT (6,15)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ZONE DESCRIPTION CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
<tr>
<td>NODE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FORMAT (1,0,15)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NODE COORDINATE CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
</tr>
<tr>
<td>Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FORMAT (15, ZF, 10, 5)</th>
</tr>
</thead>
</table>

---

Figure 1.1: Input Data Formats for FEMESH.
Figure A2. Simple Structure to Illustrate Zone Node Input Data.
APPENDIX B

FORTRAN LISTING OF MESH GENERATION PROGRAM, FEMESH

(LRC, CDC-6600 COMPUTER VERSION)
PROGRAM FEMESH

PROGRAM FEMESH Coded by L. C. LANE MAY 31, 1975

PROGRAM FEMESH generates finite element data for two dimensional planar surfaces. Structures may be subdivided into as many as 15 quadrilateral zones. The maximum mesh dimension is 24 x 24 subdivisions.

* Negative zone node numbers for a zone identified by a number **n** indicates that the negative node is connected to a zone which is identified by a number less than **n**.

** INODE First node number    ** M,N number of subdivisions
** ICTZ current zone number    ** NT total number of input nodes
** IZ total number of zones    ** NI number of nodes to be read
** IP PUNCH when IP=1**

000003 DIMENSION A(8),TITLE(10),XCODE(8),YNODE(8),
000003 DIMENSION X1(78),Y1(78)
000003 COMMON ICTZ,IZONE(15,10),NODE(15,25,25),TEMP(25)
000003 COMMON X2(4002),Y2(4002)
000003 CALL PSEUDO
000004 CALL LEROY
000003 * determine input - output devices
000005 IN = 5
000006 IOUT = 6
000007 1 FORMAT(6I5)
000007 2 FORMAT(10I5)
000007 3 FORMAT(15,2F10.5)
INITIALIZE X1 AND Y1 TO BE FILLED FROM DATA READ OFF CARDS

DO 34 I=1,78
   XL(I) = 0.
   YL(I) = 0.
34 CONTINUE

READ INPUT
READ(IN,7)TITLE
READ(IN,1)IZNT,NI,INODE,IELM,IP
READ(IN,2)((IZONE(I,J),J=1,10),I=1,IZ)

WRITE INPUT
WRITE(IOUT,8)TITLE
WRITE(IOUT,4)IZ
WRITE(IOUT,9)
WRITE(IOUT,12)
WRITE(IOUT,5)I,(IZONE(I,J),J=1,10),I=1,IZ
 WRITE(IOUT,12)

CONTINUE

SET COUNTER OF NODE NUMBERS, ICTN AND ZONE, ICTZ
ICTZ = 0
ICTN = INODE
NCOR = 0
DO 35 I=1,1661
X2(I) = 0.
Y2(I) = 0.
35 CONTINUE
CC 1010 I=1,12
C0 1009 J=1,25
C0 1008 K=1,25
NODE(I,J,K) = 0
1008 CONTINUE
1009 CONTINUE
1010 CONTINUE

ICTZ = ICTZ + 1
SET INDICATORS TO ZERO
IS1 = 0
IS2 = 0
IS3 = 0
IS4 = 0
* PULL FROM THE IZONE ARRAY THE ZONE NUMBERS AND THE ZONE COORDINATES
DO 20 I=1,8
IC = IABS(IZONE(ICTZ,I))
XNODE(I) = XI(IC)
YNODE(I) = YI(IC)
20 CONTINUE
M = IZONE(ICTZ,9)
N = IZONE(ICTZ,10)
NN = N + 1
M = M + 1.

* TITLE THE OUTPUT FOR THIS ZONE
WRITE(IOUT,11)ICTZ,11 FORMAT(1HI,28HCALCULATIONS FOR ZONE NUMBER,I4///)
IF NO VALUE IS GIVEN FOR MIDPOINTS, ASSUME A STRAIGHT LINE AND CALCULATE THE MIDPOINTS.

IF(I) = 2, 8, 2
IF(XNODE(I)) = (XNODE(I+1) + XNODE(I-1)) / 2.
IF(YNODE(I)) = (YNODE(I+1) + YNODE(I-1)) / 2.

IF(K) = 8 - 1
IF(K) = 30, 27, 28

XNODE(I) = (XNODE(I) + XNODE(7)) / 2.
YNODE(I) = (YNODE(I) + YNODE(7)) / 2.

WRITE(IOUT, 31)(I, XNODE(I), YNODE(I), I = 1, 8)
WRITE(IOUT, 32)

* IF ZONE NUMBER IS ONE, FILL NODE ARRAY AND SKIP TO X, Y COORDINATE CALCULATIONS.

IF(ICTZ - 1) = 199, 190, 201
GO 192 J = 1, NN
GO 191 I = 1, MM
NODE(I, I, J) = ICTN
ICTN = ICTN + 1
CONTINUE
CONTINUE
GO TO 3000
WRITE(IOUT, 200) ICTZ
FORMAT(1H ERROR ... ZONE NUMBER ICTZ = I4)
CONTINUE

DETERMINE WHICH SIDES ARE CONNECTED.
FILL THE NODE ARRAY.
C SIDE ONE

000336 IF(IZONE(ICTZ,2),212,220,220
000340 212 CALL FIND(1,3,1)
000343 DO 213 I=1,MM
000345 NODE(ICTZ,I,1) = TEMP(I)
000353 213 CONTINUE
000359 IS1 = 1

C SIDE 2

000356 220 CONTINUE
000356 IF(IZONE(ICTZ,4),222,230,230
000360 222 CALL FIND(3,5,2)
000363 DO 223 I=1,NN
000365 NODE(ICTZ,MM,I) = TEMP(I)
000375 223 CONTINUE
000377 IS2 = 1

C SIDE 3

000400 230 CONTINUE
000400 IF(IZONE(ICTZ,6),232,240,240
000402 232 CALL FIND(5,7,3)
000405 DO 233 I=1,MM
000407 NODE(ICTZ,NN,I) = TEMP(I)
000417 233 CONTINUE
000421 IS3 = 1

C SIDE 4

000422 240 CONTINUE
000422 IF(IZONE(ICTZ,8),242,250,250
000424 242 CALL FIND(7,1,4)
000427 DO 243 I=1,NA
000431 NODE(ICTZ,1,1) = TEMP(I)
000437 243 CONTINUE
000441 IS4 = 1
000442 250 CONTINUE
FILL NODE ARRAY – JUMP THOSE POSITIONS, ALL READY FILLED

`DO 320 J=1,NN
  DO 310 I=1,NN
  IF(NODE(ICTZ, I, J)) 315, 300, 310
  300 CONTINUE
  NODE(ICTZ, I, J) = ICTN
  ICTN = ICTN + 1
  CC TO 310
  315 CONTINUE
  ERROR
  WRITE(IOUT, 316)
  316 FORMAT(48H NODE NO FOUND IN ST. N/. 300-320 LESS THAN ZERO)
  310 CONTINUE
  320 CONTINUE
  3000 CONTINUE

* COMPUTE THE X-Y COORDINATES, OMITTING PREVIOUSLY COMPUTED SITES.
* DC, AND CN ARE THE INCREMENTAL VALUES IN THE M AND N
* DIRECTIONS RESPECTIVELY.
  PM = M
  RN = N
  CC = 2./RM
  DN = 2./RN
  WRITE(IOUT, 33) DC, DN
  33 FORMAT(1H, 4HDC=, F5.2, 4X, 4HDN=, F5.2)
  CCC = -1.
  CCN = -1.
  DO 810 J=1,NN
  IF(IJ-1) 731, 730, 731
  730 IF(I(IJ-1)) 733, 51, 733
  731 IF(IJ-NN) 733, 732, 733
  732 IF(I(IJ-1)) 733, 51, 733
  733 CONTINUE`
000517  DO 800  I=1,MM
000521   IF(I-MM) 741,734,741
000523    734 IF(IS2-1) 739,50,739
000526  739 CONTINUE
000532  739 CONTINUE
000532   S1 = 1-*CCC
000534   S2 = 1.*CCN
000536   S3 = CCC+CCN-1.
000540   S4 = 1.*CCC
000541   S5 = 1.*CCN
000543   T(1) = 1./4.*S1*S2*(-CCC-CCN-1.)
000552   T(2) = 1./2.*S1*S2*S4
000554   T(3) = 1./4.*S2*S4*(CCC-CCN-1.)
000563   T(4) = 1./2.*S4*S2*S5
000566   T(5) = 1./4.*S3*S4*S5
000571   A(6) = 1./2.*S1*S4*S5
000575   A(7) = 1./4.*S1*S5*(-CCC+CCN-1.)
000604   A(8) = 1./2.*S1*S2*S5
000610   NCOR = NCOR + 1
000611   CD 45 K=1,8,
000613   X2(NCOR) = A(K) * XNODE(K) + X2(NCOR)
000617   Y2(NCOR) = A(K) * YNODE(K) + Y2(NCOR)
000622   45 CONTINUE
000624   WRITE(IOUT,2000) NCOR,X2(NCOR),Y2(NCOR)
000635   2000 FORMAT(I5,2X,F10.5,2X,F10.5)
000635   50 CONTINUE
000635   CCC = CCC+DC
000637   800 CONTINUE
000642   .51 CONTINUE
000642   CCN = CCN + DN
000644   CCC = -1.
000646   810 CONTINUE
000650   WRITE(IOUT,54)
000654   54 FORMAT(///,1H,15HNODE NO., MATRIX/)
000654   CD 53 J=1,NN
000656   WRITE(IOUT,52)(NODE(ICTZ,I,J),I=1,MM)
000673   52 FFORMAT(1X,26I5)
000673   53 CONTINUE
000676   IF(ICTZ-1Z)1000,4000C,40000
000700  40000 CONTINUE
LIST THE ELEMENT NUMBERS AND DEFINING NODE NUMBERS.

WRITE(IOUT,900)
FORMAT(1H1,7HELEMENT,8X,1HI,10X,1HJ,16X,1HX,1HX,1HJ,16X,1HY,1HY)

COM   M = IZONE(ICTZ,9)
N = IZONE(ICTZ,10),
CO   J = J + 1
I = I + 1

INOD = NODE(ICTZ,I,J)
JNO = NODE(ICTZ,II,J)
KNO = NODE(ICTZ,II,JJ)
LNO = NODE(ICTZ,1,JJ)
WRITE(IOUT,901) IELM,INOD,JNO,KNO,LNO

PUNCHED OUTPUT IN FORMAT FOR USE IN PROGRAM SAP
IF(IP .EQ. 1) PUNCH 902, IELM,INOD,JNO,KNO,LNO

WRITE(IOUT,901) IELM,INOD,JNO,KNO,LNO

CALL FEMPLT(I2,INODE,NCOR)

WRITE(IOUT,940)
FORMAT(1H1,4HNODE,11X,1HX,1HX,1HY,1HY)
CC9011 CC 950 I = 1, NCOR
WRITE(IOUT,941) INODE,X2(I),Y2(I)

FORMAT(1H1,15,2(5X,1F10.4))
INODE = INNODE + 1
CONTINUE
STOP

END

940 FORMAT(1H1,4HNODE,11X,1HX,1HX,1HY,1HY)
901 FORMAT(16,46X,I1)
902 FORMAT(5I5)
903 FORMAT(IH ,15,2(5X,F1'0.4)).)
904 FORMAT(1H1,4HNODE,11X,1HX,1HX,1HY,1HY)
905 FORMAT(1H1,15,2(5X,1F10.4))
SUBROUTINE FIND(LP1,LP2,ISD)
C SEARCHES PREVIOUSLY CALCULATED DATA TO FIND NODE NUMBERS ASSIGNED TO A
C ZONE BOUNDARY.
C
COMMON ICTZ,IZONE(15,10),NODE(15,25,25),TEMP(25)
COMMON X2(4002),Y2(4002)
C
LP LOCATION OF PT. ON ZONE ICTZ
C
IA = 0
IT = 0
C
CO 5 I=1,25
000011 TEMP(I) = 9999
000013 5 CONTINUE
C
DEFINE CORNER NODES ON ZONE ICTZ
C
J1 = IABS(IZONE(ICTZ,LP1))
J2 = IABS(IZONE(ICTZ,LP2))
C
IIZ = ICTZ
000025 10 IIZ = IIZ - 1.
000027 IF(IIZ)200,200,110
000030 110 CONTINUE
C
SEARCH DATA OF ZONE IIZ
C
DO 40 I=1,7,2
000032 II = I + 2
000034 IF(I-716,15,16
000036 15 II = 1
000037 16 CONTINUE
000037 K1 = IABS(IZONE(IIZ,II))
000043 K2 = IABS(IZONE(IIZ,II))
C
COMPARE ICTZ TO IIZ
C
IF(J1-K1)30,20,30
000051 20 IF(J2-K2)40,21,40
000054 30 IF(J1-K2)40,31,40
31 IF(J2-K1)=0,21,40
21 CONTINUE
IA = (I+1)/2
GO TO 41
40 CONTINUE
C
41 CONTINUE
IF(IA)=0,10,45.
C
PUT DESIRED CONTENTS CF II Z IN TEMPERATURE ARRAY
C
TEMP ARRAY MUST HAVE REVERSE ORDER IF...
C
ISD = 1 CR 2 AND IA = 1 OR 2
C
- OR -
C
ISD = 3 OR 4 AND IA = 3 OR 4
C
45 CONTINUE
MMT = IZONE(II Z,9)+1
NNT = IZONE(II Z,10)+1
MK = MMT
NK = NNT
II = 0
C
CO 100 I=1,25
C
IF(IA-1)=0,46,50,46
46 IF(IA-2)=0,47,60,47
47 IF(IA-3)=0,48,70,48
48 IF(IA-4)=0,100,80,100
C
50 CONTINUE
MK = I
NK = 1
IF(ISD-2)=0,90,90,92
C
60 CONTINUE
MK = I
NK = 1
IF(ISD-2)=0,91,91,92
C
70 CONTINUE
C

50 CONTINUE
MK = I
IF(ISD-2)92,92,96
C

C

92 II = I
GO TO 93
C

94 CONTINUE
IF(II)200,200,94
C

94 CONTINUE
C

TEMP (II) = NODE(IIZ,MK,NK)
C

100 CONTINUE
200 CONTINUE
RETURN
END
SUBROUTINE FEMPLT(IZ,INODE,NCOR)

FLOTS THE FINITE ELEMENT MESH

COMMON ICTZ,IZONE(15,10),NODE(15,25,25),TEMP(25)
COMMON X2(4002),Y2(4002)

SCALE DATA
CALL ASCALE(X2,25.,NCOR,1,20.)
CALL ASCALE(Y2,13.,NCOR,1,20.)

XSCALE = X2(NCOR + 2)
YScale = Y2(NCOR + 2)

IF(XSCALE .GE. YSCALE) SF=XSCALE
IF(YSCALE .GE. XSCALE) SF=YScale

CO 100 ICTZ=1,IZ
M = IZONE(ICTZ,9)
N = IZONE(ICTZ,10)

DO 90 J=1,N
DO 80 I=1,M
II = I + 1
JJ = J + 1

DEFINE THE 4 NODES OF AN ELEMENT
INO = NODE(ICTZ,I,J) - INODE + 1
JNO = NODE(ICTZ,II,J) - INODE + 1
KNO = NODE(ICTZ,II,JJ) - INODE + 1
LNO = NODE(ICTZ,I,JJ) - INODE + 1

DEFINE THE X AND Y COORDINATES OF THE 4 NODES
XI = X2(INO)/SF
XJ = X2(JNO)/SF
XK = X2(KNO)/SF
XL = X2(LNO)/SF

YI = Y2(INO)/SF

YJ = Y2(JNC)/SF

YK = Y2(KNO)/SF

YL = Y2(LNC)/SF

CFLUT THE '4 NODES:

CALL CALPLT(XI,YI,2)

CALL CALPLT(XJ,YJ,2)

CALL CALPLT(XK,YK,2)

CALL CALPLT(XL,YL,2)

CALL CALPLT(XI,YI,2)

80 CONTINUE

90 CONTINUE

100 CONTINUE

RETURN

END
REFERENCES


Table 1. The shape functions.

\[
x = \sum_{i=1}^{8} N_i x_i
\]

\[
y = \sum_{i=1}^{8} N_i y_i
\]

\[
N_1 = -\frac{1}{4} (1 - \xi) (1 - \eta) (\xi + \eta + 1)
\]

\[
N_2 = \frac{1}{2} (1 - \xi^2) (1 - \eta)
\]

\[
N_3 = \frac{1}{4} (1 + \xi) (1 - \eta) (\xi - \eta - 1)
\]

\[
N_4 = \frac{1}{2} (1 + \xi) (1 - \eta^2)
\]

\[
N_5 = \frac{1}{4} (1 + \xi) (1 + \eta) (\xi + \eta - 1)
\]

\[
N_6 = \frac{1}{2} (1 - \xi^2) (1 + \eta)
\]

\[
N_7 = \frac{1}{4} (1 - \xi) (1 + \eta) (-\xi + \eta - 1)
\]

\[
N_8 = \frac{1}{2} (1 - \xi) (1 - \eta^2)
\]
Table 2. Input Data for Example Problem Shown in Figure 5.

<table>
<thead>
<tr>
<th>EXAMPLE PROBLEM</th>
<th>3</th>
<th>18</th>
<th>8</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-3</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>-11</td>
<td>-12</td>
<td>-13</td>
<td>15</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>0.</td>
<td>3</td>
<td>0.</td>
<td>5</td>
<td>0.</td>
</tr>
<tr>
<td>9</td>
<td>0.</td>
<td>3</td>
<td>0.</td>
<td>5</td>
<td>0.</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3. Output for Example Problem Shown in Figure 5.

**EXAMPLE PROBLEM**

NO. OF ZONES 3

<table>
<thead>
<tr>
<th>ZONE NO.</th>
<th>ZONE NODES</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 2 3 7 11</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>-3 4 5 13</td>
<td>12</td>
<td>-11</td>
</tr>
<tr>
<td>3</td>
<td>-11 -12 -13</td>
<td>15</td>
<td>18</td>
</tr>
</tbody>
</table>

NODE X Y

<table>
<thead>
<tr>
<th>NODE</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>3.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>0.000</td>
<td>3.000</td>
</tr>
<tr>
<td>11</td>
<td>3.000</td>
<td>3.000</td>
</tr>
<tr>
<td>13</td>
<td>5.000</td>
<td>3.000</td>
</tr>
<tr>
<td>16</td>
<td>3.000</td>
<td>7.000</td>
</tr>
<tr>
<td>18</td>
<td>5.000</td>
<td>7.000</td>
</tr>
</tbody>
</table>

(cont'd.)
Table 3. Output for Example Problem Shown in Figure 5 (continued).

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>100</td>
<td>101</td>
<td>105</td>
<td>104</td>
</tr>
<tr>
<td>1001</td>
<td>101</td>
<td>102</td>
<td>106</td>
<td>105</td>
</tr>
<tr>
<td>1002</td>
<td>102</td>
<td>103</td>
<td>107</td>
<td>106</td>
</tr>
<tr>
<td>1003</td>
<td>104</td>
<td>105</td>
<td>109</td>
<td>108</td>
</tr>
<tr>
<td>1004</td>
<td>105</td>
<td>106</td>
<td>110</td>
<td>109</td>
</tr>
<tr>
<td>1005</td>
<td>106</td>
<td>107</td>
<td>111</td>
<td>110</td>
</tr>
<tr>
<td>1006</td>
<td>108</td>
<td>109</td>
<td>113</td>
<td>112</td>
</tr>
<tr>
<td>1007</td>
<td>109</td>
<td>110</td>
<td>114</td>
<td>113</td>
</tr>
<tr>
<td>1008</td>
<td>110</td>
<td>111</td>
<td>115</td>
<td>114</td>
</tr>
<tr>
<td>1009</td>
<td>111</td>
<td>116</td>
<td>118</td>
<td>107</td>
</tr>
<tr>
<td>1010</td>
<td>112</td>
<td>117</td>
<td>119</td>
<td>118</td>
</tr>
<tr>
<td>1011</td>
<td>113</td>
<td>118</td>
<td>120</td>
<td>111</td>
</tr>
<tr>
<td>1012</td>
<td>114</td>
<td>119</td>
<td>121</td>
<td>120</td>
</tr>
<tr>
<td>1013</td>
<td>115</td>
<td>120</td>
<td>122</td>
<td>115</td>
</tr>
<tr>
<td>1014</td>
<td>116</td>
<td>121</td>
<td>123</td>
<td>122</td>
</tr>
<tr>
<td>1015</td>
<td>117</td>
<td>122</td>
<td>125</td>
<td>124</td>
</tr>
<tr>
<td>1016</td>
<td>118</td>
<td>123</td>
<td>126</td>
<td>125</td>
</tr>
<tr>
<td>1017</td>
<td>119</td>
<td>124</td>
<td>128</td>
<td>127</td>
</tr>
<tr>
<td>1018</td>
<td>120</td>
<td>125</td>
<td>129</td>
<td>128</td>
</tr>
<tr>
<td>1019</td>
<td>121</td>
<td>126</td>
<td>131</td>
<td>130</td>
</tr>
<tr>
<td>1020</td>
<td>122</td>
<td>127</td>
<td>131</td>
<td>131</td>
</tr>
<tr>
<td>1021</td>
<td>123</td>
<td>128</td>
<td>134</td>
<td>133</td>
</tr>
<tr>
<td>1022</td>
<td>124</td>
<td>129</td>
<td>135</td>
<td>134</td>
</tr>
</tbody>
</table>

(cont'd.)
Table 3. Output for Example Shown in Figure 5 (concluded).

<table>
<thead>
<tr>
<th>NODE</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>101</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>102</td>
<td>2.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>103</td>
<td>3.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>104</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>105</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>106</td>
<td>2.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>107</td>
<td>3.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>108</td>
<td>0.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>109</td>
<td>1.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>110</td>
<td>2.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>111</td>
<td>3.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>112</td>
<td>0.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>113</td>
<td>1.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>114</td>
<td>2.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>115</td>
<td>3.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>116</td>
<td>4.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>117</td>
<td>5.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>118</td>
<td>4.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>119</td>
<td>5.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>120</td>
<td>4.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>121</td>
<td>5.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>122</td>
<td>4.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>123</td>
<td>5.0000</td>
<td>3.0000</td>
</tr>
<tr>
<td>124</td>
<td>3.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>125</td>
<td>4.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>126</td>
<td>5.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>127</td>
<td>3.0000</td>
<td>5.0000</td>
</tr>
<tr>
<td>128</td>
<td>4.0000</td>
<td>5.0000</td>
</tr>
<tr>
<td>129</td>
<td>5.0000</td>
<td>5.0000</td>
</tr>
<tr>
<td>130</td>
<td>3.0000</td>
<td>6.0000</td>
</tr>
<tr>
<td>131</td>
<td>4.0000</td>
<td>6.0000</td>
</tr>
<tr>
<td>132</td>
<td>5.0000</td>
<td>6.0000</td>
</tr>
<tr>
<td>133</td>
<td>3.0000</td>
<td>7.0000</td>
</tr>
<tr>
<td>134</td>
<td>4.0000</td>
<td>7.0000</td>
</tr>
<tr>
<td>135</td>
<td>5.0000</td>
<td>7.0000</td>
</tr>
</tbody>
</table>
Table 4. Input for Shear Panel.

<table>
<thead>
<tr>
<th>SHEAR PANEL</th>
<th>4 21 17 1 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td>8 11 10 20 10</td>
</tr>
<tr>
<td>-11 -8 -5</td>
<td>6 7 9 13 12 20 3</td>
</tr>
<tr>
<td>-1 -10 -11</td>
<td>15 19 18 17 14 10 20</td>
</tr>
<tr>
<td>-11 -12 -13</td>
<td>16 21 20 -19 -15 3 20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1 0.</th>
<th>2 3.0</th>
<th>3 4.35</th>
<th>.4 4.3727</th>
<th>.1040</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4.435</td>
<td>.185</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4.6</td>
<td>.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.825</td>
<td>.825</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.975</td>
<td>.975</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2.31</td>
<td>2.31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2.475</td>
<td>2.475</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.</td>
<td>3.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>.825</td>
<td>3.825</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3.975</td>
<td>3.975</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.</td>
<td>4.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>.1040</td>
<td>4.3727</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>.185</td>
<td>4.435</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>.350</td>
<td>4.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5. Input for Bolted Joint Specimen.

<table>
<thead>
<tr>
<th>BOLTED JOINT COMPOSITE STRUCTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 66 41 100 1000</td>
</tr>
<tr>
<td>28 29 30 21 3 2 1 20 2 3</td>
</tr>
<tr>
<td>-30 22 7 6 5 4 -3 -21 5 3</td>
</tr>
<tr>
<td>-7 -22 -30 31 32 23 9 8 5 6</td>
</tr>
<tr>
<td>-9 -23 -32 24 13 12 11 10 5 3</td>
</tr>
<tr>
<td>-30 41 57 56 55 42 -32 -31 5 6</td>
</tr>
<tr>
<td>49 50 51 58 -57 -41 -30 40 5 3</td>
</tr>
<tr>
<td>47 48 -49 -49 -30 -29 -28 39 2 6</td>
</tr>
<tr>
<td>-51 -50 -49 60 61 59 53 52 5 6</td>
</tr>
<tr>
<td>-53 -59 -61 43 -32 -42 -55 54 5 6</td>
</tr>
<tr>
<td>-61 62 63 44 34 33 -32 -43 3 6</td>
</tr>
<tr>
<td>-32 -35 -34 25 15 14 -13 -24 3 3</td>
</tr>
<tr>
<td>-34 35 36 26 17 16 -15 -25 11 3</td>
</tr>
<tr>
<td>-36 37 38 27 19 18 -17 -26 6 3</td>
</tr>
<tr>
<td>-63 64 65 45 -36 -35 -34 -44 11 6</td>
</tr>
<tr>
<td>-69 66 67 46 -38 -37 -38 -45 6 6</td>
</tr>
</tbody>
</table>

| 1 0 1.5          |
| 3 0.13 1.5       |
| 4 0.4290 1.5     |
| 5 0.5345 1.5     |
| 6 0.541d 1.4635  |
| 7 0.5625 1.4325  |
| 8 0.63 1.4045    |
| 9 0.6975 1.4325  |
| 10 0.7182 1.4635 |
| 11 0.7255 1.5    |
| 12 0.8308 1.5    |
| 13 1.13 1.5      |
| 15 1.5 1.5       |
| 17 3.3 1.5       |
| 19 4.5 1.5       |
| 22 0.4292 1.284  |
| 23 0.8308 1.284  |
| 23 0. 1.0        |
| 30 0.13 1.0      |
| 32 1.13 1.0      |
| 34 1.5 1.0       |
| 36 3.5 1.0       |
| 38 4.5 1.0       |
| 41 0.4292 0.7003 |
| 42 0.8308 0.7008 |
| 47 0. 0.         |
| 49 0.13 0.       |
| 50 0.4292 0.2992 |
| 51 0.5625 0.4325 |
| 52 0.63 0.4045   |
| 53 0.6975 0.4325 |
| 54 0.7255 0.5    |
| 55 0.6975 0.5675 |
| 56 0.63 0.5955   |
| 57 0.5625 0.5675 |
| 58 0.5345 0.5    |
| 59 0.8308 0.2992 |
| 61 1.13 0.       |
| 63 1.5 0.        |
| 65 3.3 0.        |
| 67 4.5 0.        |
Figure 1. Mapping of a Quadrilateral from the Natural to the Cartesian Coordinate System.
Figure 2. Ungraded and graded mesh generated in the Cartesian system.
Figure 3. Numbering of Zone Nodes.
READ AND WRITE INPUT

DEFINE DATA FOR ZONE I

ARE MIDSIDE NODES DEFINED? NO

YES

IS A SIDE CONNECTED? YES

FIND DATA OF ZONE K WHICH IS CONNECTED TO ZONE I

NO

DUPLICATE DATA OF ZONE K IN DATA ARRAY FOR ZONE I

INCREMENT I

COMPLETE NODE NUMBER ARRAY

CALCULATE X AND Y COORDINATES

LAST ZONE? NO

YES

OUTPUT PRINT, PLOT

Figure 4. Flowchart for Mesh Generation Program, FEMESH.
Figure 5. Example of Mesh Generation for a Simple Structure.
Figure 6. Zones of Shear Panel.
Figure 7. Mesh Generated for Quartersection of Shear Panel.
Figure 8. Zones of Bolted Joint Specimen.
Figure 9. Mesh Generated for Half Section of Bolted Joint Specimen.
FINITE ELEMENT ANALYSIS OF A COMPOSITE BOLTED JOINT SPECIMEN

By

Earl A. Thornton
FINITE ELEMENT ANALYSIS OF A COMPOSITE BOLTED JOINT SPECIMEN

By

Earl A. Thornton

INTRODUCTION

With high strength and weight savings, advanced composite materials have become increasingly important in aircraft structural design. The full potential for the increase of structural efficiency through the use of advanced composites has not yet been fully realized because of low efficiencies in mechanical joints. The Advanced Composites Design Guide (ref. 1) states that weight savings may be reduced by as much as 40 percent due to such practical constraints.

In the use of conventional materials, design methods for joints have evolved over a period of time from data gathered from experimental and analytical solutions and, in addition, are often based upon rules-of-thumb derived from experience. For advanced composites, such data and experience are relatively limited. To partially fill this need, test programs are underway at Langley Research Center (LRC) to establish data on a number of mechanical joint designs (ref. 2).

The purpose of the present study was to provide analytical support for the LRC bolted joint test program. Specific objectives of the study were to: (1) determine the laminate stress distribution in an extra graphite reinforced bolted joint specimen, and (2) compare two methods of modeling bolt transfer loads for determination of stress distributions in bolted joints.

This paper will describe the finite element model used to represent the bolted joint specimen. The two methods used to represent bolt transfer loads will be discussed. Laminate membrane force distributions predicted by the finite element
analysis will be presented, and force gradients at the bolt holes will be discussed. Differences in the results due to the methods of representing the bolt loads will also be discussed.

BOLTED JOINT SPECIMEN

The specimen analyzed in this study is the specimen denoted as extra graphite reinforced joint specimen number one, reference 2. The specimen is shown schematically in figure 1 with the dimensions used in the analysis. The specimen was fabricated from a basic layup of 15 plies reinforced by additional plies so that in the thick section where the bolt holes are located there are 49 plies. The ply stacking sequences are shown in figure 2 with cross-sectional details of the layup. Reinforcing plies increase by 0.1 in. in length per ply over the transition section from 49 plies to 15 plies.

ANALYTICAL PROCEDURES

Finite Element Model

The NASA Structural Analysis (NASTRAN) computer program (level 15.5) was used to compute the laminate stress distributions in the specimen. The specimen was assumed to be in-plane stress and due to symmetry only one-half of the specimen was represented with finite elements. The finite element representation is shown in figure 3. The specimen was represented by an assemblage of 349 quadrilateral and triangular membrane elements. The NASTRAN finite elements used have constant stress throughout each element. The mathematical model has 307 grid points and 573 degrees of freedom. Vertical displacements were set to zero on the top boundary of the finite element model to represent symmetry, and horizontal displacements at the right edge of the finite element model were set to zero to represent clamping in the test fixture.

In the analytical formulation underlying the present NASTRAN elements the element material is assumed homogeneous through the
thickness. The element extensional stiffnesses are obtained internally in NASTRAN by multiplying the material elasticity matrix by the thickness of the element, reference 3. However, the specimen in the present study is characterized by several layers of material which are assumed homogeneous within the individual layers only. Thus for the composite laminate the extensional stiffnesses, $A_{ij}$, were computed externally using laminated plate theory. The stiffnesses were then input to NASTRAN in place of the material elasticity matrix, and the thickness of the specimen was everywhere taken as unity.

The extensional stiffnesses $A_{ij}$, a 3 x 3 symmetric matrix, were computed from reference 4:

$$A_{ij} = \sum_{k=1}^{N} (Q_{ij})_k (z_k - z_{k-1}) \quad (1)$$

where $(Q_{ij})_k$ denotes the material elasticity matrix for a single layer and $(z_k - z_{k-1})$ denotes the thickness of the kth layer. The extensional stiffnesses relate the in-plane membrane forces $(N_x, N_y, N_{xy})$ to the midplane extensional strains $(\varepsilon_x, \varepsilon_y, \gamma_{xy})$ of the laminate. Since the extensional stiffnesses were input to NASTRAN in place of the NASTRAN material elasticity matrix, the NASTRAN membrane element stresses $(\sigma_x, \sigma_y, \gamma_{xy})$ were the laminate stress resultants $(N_x, N_y, N_{xy})$.

In the present analysis the lamina elastic constants were taken as $E_{11} = 20 \times 10^6$ psi, $E_{22} = 2 \times 10^6$ psi, $G = 0.8 \times 10^6$ psi and $\nu_{12} = 0.3$. Each lamina had a thickness of 0.00542 in. To represent the tapered character of the specimen, extensional stiffnesses were computed for the 19 different cross-sectional layups. The values of the extensional stiffnesses for the specimen are given in table 1.
Bolt Loads

The specimen was analyzed for loading corresponding to the design failure load. This loading, estimated at 21,813 lb was assumed to be equally distributed to the three bolts such that the total load transmitted to the specimen per bolt was 7271 lb. In the finite element model, one-half of this load was applied to the center bolt hole and the full value was applied to the lower bolt hole.

Two methods were used to represent the transfer of the bolt forces to the finite element model. In the first approach the bolt was assumed to have a perfect fit, and the load transfer was assumed to take place over 180° of the bolt hole. The contact force was assumed to vary sinusoidally over this area of contact. Equilibrium of the bolt was then used to obtain the relation:

\[ N = \frac{2}{\pi} \frac{2Q}{R} \cos \theta \]  

where \( N \) denotes the contact force per unit arc length, \( 2Q \) is the total bolt load, and \( R \) is the radius of the bolt hole. The angle \( \theta \) is measured from a horizontal axis through the hole. Equation (2) was used to compute equivalent grid point forces for each grid point in the contact region \((-90° < \theta < 90°)\). The equivalent grid point forces were computed by integrating Equation (2) through an angle of -6° to +6° at each grid point. The equivalent grid point loads are shown in figure 4.

In the second approach an imperfect fit was assumed and a nonlinear analysis of the bolt transfer loads was made. This analysis, made using the computer program CONTACT developed in reference 5, consists of increasing the bolt load in increments and determining the number of grid points in contact and their loads at each load increment. The analysis requires as part of its input the flexibility matrix for the bolt hole. This flexibility matrix was obtained from the finite element model by applying unit loads at each node of the center bolt hole. The
16 x 16 flexibility matrix was computed one column at a time for 16 unit load subcases. This matrix was then input to the CONTACT program and the bolt transfer forces were computed for several load increments. The bolt transfer forces and the region of contact for four load increments including the maximum load are shown in figure 5. These forces were computed using an initial lack of fit of -0.00287 in. This value, as defined in the program, denotes a clearance based upon the radius of the hole.

RESULTS AND DISCUSSION

The membrane force distributions at the center and outside bolt holes as predicted by the finite element analysis are shown in figures 6 through 8. Shown are plots of the radial force \( N_r \), the circumferential force \( N_\theta \), and the in-plane shearing force \( N_{r\theta} \) versus the angle \( \theta \) from the centerline. Predictions based upon the two methods of representing the bolt transfer loads are compared.

There is very little, if any, difference in the membrane forces between the center bolt hole and the outside bolt holes. Each bolt was assumed to carry the same bolt load and there appears to be no interaction effects between holes nor edge effects upon the stress distributions in the outside holes. The magnitudes and variations of the membrane forces and the effects of the two methods of representing the bolt transfer loads can thus be discussed with regard to either hole.

The largest radial force intensity (fig. 6) occurs, as might be expected, on the centerline of the bolt hole. The nonlinear bolt loading method predicts the largest radial membrane forces with a value of 32 kips/in. compression which is about 23 percent higher than the value based upon the cosine bolt loading. The largest circumferential membrane force (fig. 7) of 30 kips/in. tension occurs at an angle of about 75° from the bolt centerline and is also predicted by the nonlinear bolt loading technique.
This stress is about 15 percent higher than the value based upon the cosine bolt loading. The in-plane membrane shear forces (fig. 8) tend to be smaller than the radial or circumferential membrane forces. The largest membrane shear force is about 9 kips/in. and is due to the nonlinear bolt loading. Since the in-plane shearing forces tend to be small, the principal values (not shown) of the membrane forces correspond in magnitude and location to the maximum radial and circumferential membrane forces.

The distribution of the longitudinal membrane force \( N_x \) along the specimen centerline is shown in figure 9. At \( x = 0 \) the membrane force should be zero since this edge is stress free; the small nonzero value is indicative of the error in the finite element solution. The membrane force at the left edge of the bolt hole (\( x = 0.5 \)) rises very sharply due to the indirect bearing load of the bolt. On the right side of the bolt hole, the force should also be zero since the bolt is not in contact at this point. The finite element solution tends to zero at this point. Away from the hole for increasing \( x \), the membrane force approaches a uniform value given by the total applied force (21,816 lb), divided by the specimen width (3 in.).

Further insight into the results of the finite element analyses can be obtained by considering an elasticity solution for an isotropic medium. In reference 6, Bickley presents the plane stress elasticity solution for a hole in an infinite medium loaded by a cosine pressure distribution over one-half of the boundary of the hole. Closed form solutions for the stress components are given in polar coordinates in terms of the radius of the hole and Poisson's ratio. Tabulated data of the stress components for Poisson's ratio of 0.25 are also presented.

In figure 10 are shown the membrane force distributions predicted by Bickley for an infinite isotropic medium with a hole equal in radius to the bolt hole in the composite specimen and loaded by the bolt load used in the finite element analysis. The plots are made for \( r/a = 1.2 \) which corresponds closely to
laminated composite material was represented in NASTRAN as a homogeneous material with equivalent extensional stiffness. Laminate membrane force distributions were predicted.

Comparison of the two methods of representing the bolt transfer loads showed the two methods were in qualitative agreement. The nonlinear analysis estimated membrane forces about 20 to 25 percent higher than the linear analysis. Peak forces were found to be a radial compressive force on the bolt centerline and a circumferential tensile force of the same magnitude at about 70° from the centerline. In-plane shear forces were found to be relatively small. There were little or no interaction effects between holes or boundaries of the specimen. Comparison of the finite element solution with an isotropic elasticity solution suggests that as a rule these effects will not be important for in-plane membrane forces provided offset distances between holes or edges are greater than five hole radii.
REFERENCES


Table 1. Extensional stiffnesses.

<table>
<thead>
<tr>
<th>Section</th>
<th>$A_{11}$</th>
<th>$A_{12}$</th>
<th>$A_{13}$</th>
<th>$A_{22}$</th>
<th>$A_{23}$</th>
<th>$A_{33}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.01</td>
<td>0.932</td>
<td>0</td>
<td>1.34</td>
<td>0</td>
<td>0.984</td>
</tr>
<tr>
<td>2</td>
<td>2.94</td>
<td>0.877</td>
<td>-0.0492</td>
<td>1.27</td>
<td>-0.0492</td>
<td>0.927</td>
</tr>
<tr>
<td>3</td>
<td>2.87</td>
<td>0.823</td>
<td>0</td>
<td>1.20</td>
<td>0</td>
<td>0.870</td>
</tr>
<tr>
<td>4</td>
<td>2.65</td>
<td>0.816</td>
<td>0</td>
<td>1.17</td>
<td>0</td>
<td>0.861</td>
</tr>
<tr>
<td>5</td>
<td>2.43</td>
<td>0.809</td>
<td>0</td>
<td>1.15</td>
<td>0</td>
<td>0.853</td>
</tr>
<tr>
<td>6</td>
<td>2.36</td>
<td>0.755</td>
<td>0.0492</td>
<td>1.08</td>
<td>0.0492</td>
<td>0.796</td>
</tr>
<tr>
<td>7</td>
<td>2.28</td>
<td>0.699</td>
<td>0</td>
<td>1.01</td>
<td>0</td>
<td>0.739</td>
</tr>
<tr>
<td>8</td>
<td>2.07</td>
<td>0.693</td>
<td>0</td>
<td>0.986</td>
<td>0</td>
<td>0.730</td>
</tr>
<tr>
<td>9</td>
<td>2.00</td>
<td>0.639</td>
<td>-0.0492</td>
<td>0.914</td>
<td>-0.0492</td>
<td>0.673</td>
</tr>
<tr>
<td>10</td>
<td>1.92</td>
<td>0.584</td>
<td>0</td>
<td>0.841</td>
<td>0</td>
<td>0.616</td>
</tr>
<tr>
<td>11</td>
<td>1.85</td>
<td>0.529</td>
<td>-0.0492</td>
<td>0.769</td>
<td>-0.0492</td>
<td>0.560</td>
</tr>
<tr>
<td>12</td>
<td>1.78</td>
<td>0.474</td>
<td>0</td>
<td>0.697</td>
<td>0</td>
<td>0.503</td>
</tr>
<tr>
<td>13</td>
<td>1.71</td>
<td>0.419</td>
<td>-0.0492</td>
<td>0.625</td>
<td>-0.0492</td>
<td>0.446</td>
</tr>
<tr>
<td>14</td>
<td>1.63</td>
<td>0.364</td>
<td>0</td>
<td>0.553</td>
<td>0</td>
<td>0.389</td>
</tr>
<tr>
<td>15</td>
<td>1.42</td>
<td>0.358</td>
<td>0</td>
<td>0.531</td>
<td>0</td>
<td>0.380</td>
</tr>
<tr>
<td>16</td>
<td>1.34</td>
<td>0.303</td>
<td>0.0492</td>
<td>0.459</td>
<td>0.0492</td>
<td>0.323</td>
</tr>
<tr>
<td>17</td>
<td>1.13</td>
<td>0.297</td>
<td>0.0492</td>
<td>0.437</td>
<td>0.0492</td>
<td>0.314</td>
</tr>
<tr>
<td>18</td>
<td>1.05</td>
<td>0.242</td>
<td>0</td>
<td>0.365</td>
<td>0</td>
<td>0.258</td>
</tr>
<tr>
<td>19</td>
<td>1.05</td>
<td>0.242</td>
<td>0</td>
<td>0.365</td>
<td>0</td>
<td>0.258</td>
</tr>
</tbody>
</table>
Figure 1. Extra Graphite Reinforced Joint Specimen.
Figure 2. Ply Stacking Sequence for Graphite Reinforced Specimen 1.
Figure 3. Finite Element Representation of Bolted Joint Specimen.
Figure 4. Bolt Transfer, Loads for Cosine Load Distribution.
Figure 5. Bolt Transfer Loads from Nonlinear Loading Analysis.
Figure 6. Radial Membrane Force Distributions at the Bolt Holes.
Figure 7. Circumferential Membrane Force Distributions at the Bolt Holes.
Figure 8. Membrane Shearing Force Distributions at the Bolt Hole.
Figure 9. Longitudinal Membrane Force Distribution Along Specimen Centerline.
(a) INFINITE ISOTROPIC MEDIUM WITH COSINE BOLT LOADING

\[ N = \frac{2 \theta}{\pi a} \cos \theta \]

(b) RADIAL MEMBRANE FORCE DISTRIBUTION FOR \( r/a = 1.2 \).

Figure 10. Elasticity Solution for Infinite Isotropic Medium with Cosine Bolt Loading.
(c) CIRCUMFERENTIAL MEMBRANE FORCE DISTRIBUTION FOR $r/a = 1.2$.

(d) MEMBRANE SHEARING FORCE DISTRIBUTION FOR $r/a = 1.2$.

Figure 10 (concluded). Elasticity Solution for Infinite Isotropic Medium with Cosine Bolt Loading.
Figure 11. Elasticity Solution for Longitudinal Membrane Force Distribution Along Specimen Centerline.