The CQDRNG8 - a quadratic, isoparametric, axisymmetric finite element for the NASTRAN computer program

John Gray

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
THE CQDRNG8 - A QUADRATIC, ISOPARAMETRIC, AXISYMMETRIC
FINITE ELEMENT FOR THE NASTRAN COMPUTER PROGRAM

by

John F. Gray

A Thesis Submitted
in
Partial Fulfillment
of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Mechanical Engineering

Approved by:  

Prof.  

Wayne Walters

Prof.  

B. Karlekar

Prof.  

Neville Rieger

Prof.  

Richard B. Hetnarski

DEPARTMENT OF MECHANICAL ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

May 1977
ACKNOWLEDGEMENTS

The success of this undertaking is largely due to the cooperation of many persons. I would like to express my deepest gratitude to the following:

To my wife Phyllis and my son Kevin for their patience and understanding throughout this time period.

To Dr. V. Genberg and Mr. P. Mueller, Eastman Kodak Company, for their suggestions and technical advice.

To S. Wall, J. Joseph, and R. Harder, The MacNeal-Schwendler Corporation, for their gracious assistance in providing needed insight into the MSC/NASTRAN program.

To Professors W. Walter, my thesis adviser, and N. Rieger, Rochester Institute of Technology, Mechanical Engineering Department, for their guidance and suggestions.

To Eastman Kodak Company for use of the computer facilities and access to the MSC/NASTRAN program.
The development of an axisymmetric ring finite element is presented and FORTRAN subroutines for implementing the capability into the MSC/NASTRAN finite element program are given. The element is an eight-noded isoparametric quadrilateral of quadratic order. The following matrices and capabilities are developed:

1. stiffness matrix for homogeneous isotropic materials,
2. thermal conductance matrix for homogeneous isotropic materials,
3. calculation of equivalent nodal forces due to temperature loads,
4. calculation of stresses, and
5. plotting of undeformed and deformed structures.

Several classical thermal and structural problems are solved to demonstrate these capabilities. In all cases, the element results compare well with theory. Comparisons are made to existing MSC/NASTRAN axisymmetric finite elements. The new element shows increased accuracy compared to the existing elements. Convergence of the element is shown.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>i</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ii</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>iii</td>
</tr>
<tr>
<td><strong>I.</strong> Introduction</td>
<td></td>
</tr>
<tr>
<td><strong>II.</strong> Literature Review</td>
<td></td>
</tr>
<tr>
<td><strong>III.</strong> Axisymmetric Structures</td>
<td></td>
</tr>
<tr>
<td><strong>IV.</strong> Element Derivations</td>
<td></td>
</tr>
<tr>
<td>A. Stiffness Matrix</td>
<td>9</td>
</tr>
<tr>
<td>B. Thermoelastic Loads</td>
<td>17</td>
</tr>
<tr>
<td>C. Stress Recovery</td>
<td>19</td>
</tr>
<tr>
<td>D. Conductance Matrix</td>
<td>21</td>
</tr>
<tr>
<td><strong>V.</strong> Additions to NASTRAN</td>
<td></td>
</tr>
<tr>
<td><strong>VI.</strong> Example Problems</td>
<td>30</td>
</tr>
<tr>
<td>A. Bending of Simply Supported Flat Plate</td>
<td>32</td>
</tr>
<tr>
<td>B. Thermal Expansion of Circular Disk</td>
<td>39</td>
</tr>
<tr>
<td>C. Thermal Expansion of Hollow Sphere</td>
<td>43</td>
</tr>
<tr>
<td>D. Heat Flow in a Circular Disk</td>
<td>47</td>
</tr>
<tr>
<td>E. Heat Flow in an Axial Rod</td>
<td>57</td>
</tr>
<tr>
<td>F. Heat Flow in a Hollow Sphere</td>
<td>59</td>
</tr>
<tr>
<td>G. Thermoelastic Stresses in a Hollow Cylinder</td>
<td>64</td>
</tr>
<tr>
<td><strong>VII.</strong> Summary</td>
<td>68</td>
</tr>
<tr>
<td><strong>VIII.</strong> References</td>
<td>72</td>
</tr>
<tr>
<td><strong>IX.</strong> Appendix A</td>
<td>75</td>
</tr>
<tr>
<td><strong>X.</strong> Appendix B</td>
<td>107</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Element coordinate system and shape functions</td>
</tr>
<tr>
<td>2</td>
<td>Gaussian integration and stress points</td>
</tr>
<tr>
<td>3</td>
<td>Stiffness matrix partitions</td>
</tr>
<tr>
<td>4</td>
<td>Deflections of simply supported plate with center load</td>
</tr>
<tr>
<td>5</td>
<td>Deflections of simply supported plate with concentric circular load</td>
</tr>
<tr>
<td>6</td>
<td>Stresses in simply supported plate with load on concentric circle</td>
</tr>
<tr>
<td>7</td>
<td>Plot of circular disk model</td>
</tr>
<tr>
<td>8</td>
<td>Deflections of disk with center hole - uniform temperature change</td>
</tr>
<tr>
<td>9</td>
<td>Plot of hollow sphere model</td>
</tr>
<tr>
<td>10</td>
<td>Thermal expansion of sphere</td>
</tr>
<tr>
<td>11</td>
<td>Circular disk models - CQDRNG8 elements</td>
</tr>
<tr>
<td>12</td>
<td>Circular disk models - CTRIARG elements</td>
</tr>
<tr>
<td>13</td>
<td>Circular disk temperatures - Convergence test</td>
</tr>
<tr>
<td>14</td>
<td>Temperatures in axial rod - enforced end temperatures</td>
</tr>
<tr>
<td>15</td>
<td>Temperatures in hollow sphere - enforced temperatures at surfaces</td>
</tr>
<tr>
<td>16</td>
<td>Temperatures in hollow sphere - enforced temperature at ID - convection at OD</td>
</tr>
<tr>
<td>17</td>
<td>Plot of hollow cylinder model</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Deflection of simply supported plate with center load</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>Deflection of simply supported circular plate with load on concentric circle</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>Stresses in simply supported circular plate with concentric loading</td>
<td>38</td>
</tr>
<tr>
<td>4</td>
<td>Thermal deflections in disk</td>
<td>42</td>
</tr>
<tr>
<td>5</td>
<td>Thermal expansion of hollow sphere</td>
<td>46</td>
</tr>
<tr>
<td>6</td>
<td>Circular disk temperatures - convergence test</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td>Temperatures in hollow sphere - enforced temperatures at surfaces</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>Temperatures in hollow sphere - enforced temperature at ID - convection at OD</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>Stresses in hollow cylinder</td>
<td>66</td>
</tr>
</tbody>
</table>
### NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r, \theta, z )</td>
<td>Cylindrical coordinates (radial, circumferential, axial)</td>
</tr>
<tr>
<td>( \phi )</td>
<td>Strain energy</td>
</tr>
<tr>
<td>( \sigma_r, \sigma_{\theta}, \sigma_z )</td>
<td>Normal stresses in ( r, \theta, z ) directions and shearing stress in ( rz ) plane</td>
</tr>
<tr>
<td>( \tau_{rz} )</td>
<td>Normal strains in ( r, \theta, z ) directions and shearing strain in ( rz ) plane</td>
</tr>
<tr>
<td>( E )</td>
<td>Young's modulus</td>
</tr>
<tr>
<td>( v )</td>
<td>Poisson's ratio</td>
</tr>
<tr>
<td>VOL</td>
<td>Volume</td>
</tr>
<tr>
<td>( u, v )</td>
<td>Radial and axial displacement components</td>
</tr>
<tr>
<td>( N_i )</td>
<td>Shape function associated with node ( i )</td>
</tr>
<tr>
<td>( \alpha, \beta )</td>
<td>Local element coordinates</td>
</tr>
<tr>
<td>( U )</td>
<td>Thermal potential function</td>
</tr>
<tr>
<td>( \tilde{q} )</td>
<td>Heat flux density</td>
</tr>
<tr>
<td>( k )</td>
<td>Thermal conductivity</td>
</tr>
<tr>
<td>( x_i )</td>
<td>Generalized coordinate</td>
</tr>
<tr>
<td>{ }</td>
<td>Vector</td>
</tr>
<tr>
<td>[ ]</td>
<td>Matrix</td>
</tr>
<tr>
<td>[ ]^{-1}</td>
<td>Inverse of a matrix</td>
</tr>
<tr>
<td>{ }, [ ]^{T}</td>
<td>Transpose of vector or matrix</td>
</tr>
<tr>
<td>|</td>
<td>Determinant of a matrix</td>
</tr>
<tr>
<td>[ K ]</td>
<td>Stiffness or conductance matrix</td>
</tr>
<tr>
<td>{ u }</td>
<td>Nodal vector of unknowns (displacements or temperatures)</td>
</tr>
<tr>
<td>{ P }</td>
<td>Nodal vector of applied loads</td>
</tr>
<tr>
<td>{ \sigma }</td>
<td>Element stress vector</td>
</tr>
<tr>
<td>{ \varepsilon }</td>
<td>Element strain vector</td>
</tr>
</tbody>
</table>
Constitutive matrix relating stress and strain

Matrix relating strains to displacements

Jacobian matrix

Transformation matrix

Gaussian integration weighting factor
I. INTRODUCTION

NASTRAN is a large general purpose finite element program with a substantial element library. One of the most noticeable deficiencies is the absence of a higher order quadrilateral axisymmetric ring element. The objective of this thesis is to fill this void. A suitable element with limited capabilities was developed by Janucik.\(^1\) This effort is an extension of that development. Additional capabilities are developed and this element is implemented in the NASTRAN program as a DUMMY element.

This element, named the CQDRNG8, has many capabilities. Elastic stiffness is calculated for homogeneous isotropic materials. Thermo-elastic loading is supported. Plots for both undeformed and deformed structures can be obtained. Element stresses are output at the centroid and the four corner grids of each element. Stress invariants calculated include the principal stresses and directions, the maximum shear, and the octahedral shear. A thermal conductance matrix is calculated for homogeneous isotropic materials thus providing a steady state heat transfer capability. The CQDRNG8 is an eight nodded isoparametric quadrilateral element which allows quadratic variation of displacement and temperature within the element. A number of user options exist with respect to both NASTRAN and this element.

The CQDRNG8 element's capabilities are illustrated by solving several classical thermal and structural problems. In all cases, very good correlation is observed between theoretical and finite element solutions. Convergence of the solution for increasingly detailed finite element models is demonstrated. Comparisons with existing NASTRAN axisymmetric
elements is made.

Data cards for the new element and subroutines for implementing the CQDRNG8 into NASTRAN are given in the appendices.
II. LITERATURE REVIEW

In the last decade, many texts and articles on finite element theory have been published. Perhaps the most widely referenced texts are those by Zienkiewicz. In his texts, Zienkiewicz develops the general theory for finite element analysis. His texts also include individual sections on specific finite element formulations. A comprehensive treatment of the use of isoparametric shape functions is included. Finally Zienkiewicz devotes a discussion to solving general field problems using finite element methods. Particular references are made to the heat transfer problem.

The text by Przemieniecki addresses itself to the solution of structural problems using matrix techniques. Some simple finite element formulations are developed. This text includes a thorough treatment of matrix substructuring -- the reduction of matrix size by partitioning operations.

Segerlind presents a finite element text which is not structures oriented. Finite element formulations for heat transfer, fluid mechanics, and elasticity problems are developed. Axisymmetric field problems (both steady state and transient) are discussed. Computer implementation of the various finite element formulations is discussed. A number of instructional computer programs are listed.

Desai develops the finite element matrices for the structural problem. Techniques for nonlinear analysis are discussed. He presents a number of useful examples.
Bathe's finite element developments are rather brief but he presents a comprehensive treatment of matrix solution techniques. Of particular interest are elimination techniques for equation solution, eigenvalue extraction methods, and transient integration routines.

In his paper, Ergatoudis describes a family of isoparametric quadrilateral finite elements. He presents isoparametric shape functions for linear, quadratic, and cubic elements. Special variations include additional internal degrees of freedom and a mixed order finite element which can have different numbers of grids along each edge. Baldwin gives the computer routines for a quadratic order isoparametric thin plate element and demonstrates the accuracy of the element for the solution of bending problems.

Detailed derivations of finite element matrices for heat transfer problems are given by Lee and MacNeal. Both authors develop the conductance and capacitance matrices for the finite elements in NASTRAN. MacNeal also develops the structural matrices for the NASTRAN elements. Lee presents several examples illustrating NASTRAN's heat transfer capabilities.

Doherty's paper develops several higher order axisymmetric finite elements. Quadrilateral elements are formed by combining various numbers of triangular elements then eliminating interior grid points. In an effort to avoid midside grids, mixed order triangular elements are combined such that a quadratic displacement function exists interior to the element but only a linear variation occurs along exterior edges. The developments include the effects of orthotropic and temperature dependent materials.
Both Bruch\textsuperscript{13} and Kohler\textsuperscript{14} develop the finite element matrices for the solution of transient two dimensional heat conduction problems. Bruch uses a rectangular element with linear temperature variation. Kohler uses general quadrilateral elements having linear, quadratic, and cubic temperature variations. In his paper, Zienkiewicz\textsuperscript{15} presents higher order isoparametric finite element formulations for the solution of two and three dimensional transient field problems. He develops a transient solution formula and solves a heat conduction problem.

The basis for this work is a thesis submitted by Janucik\textsuperscript{1} In his paper, Janucik develops the stiffness matrix for the eight noded iso-parametric quadrilateral ring element. He presents a stand alone computer program for utilizing this capability. He describes several limitations relating to this program including numerical precision, core, and problem size constraints. The current work is designed to enhance the capabilities of this element and to eliminate the problems by adding the element to the \textsc{nast}an finite element program.

The \textsc{nast}an texts by Wall\textsuperscript{16,17} are the basis for designing and adding the software to realize this new element capability. A thorough description of the general \textsc{nast}an techniques used in formulating and solving the desired equations is presented. The texts outline the specific capabilities needed. Detailed formats of the tables and matrices needed by the programmer are presented along with instructions for utilizing available \textsc{nast}an capabilities. The extensive \textsc{nast}an overlay structure is explained. Techniques for implementing the new element capability within this framework are given.
The manuals by McCormick\textsuperscript{18,19} provide an introduction to the capabilities and use of the NASTRAN computer program. Detailed user information is found in the first manual. Additional NASTRAN features and special modeling considerations are described by Joseph\textsuperscript{20}. NASTRAN numerical techniques and element formulations are described by MacNeal\textsuperscript{11}. 
III.  AXISYMMETRIC STRUCTURES

Axisymmetric structures under axisymmetric loading are frequently encountered in engineering problems. These three dimensional structures can efficiently be analysed as two dimensional ones by properly treating the components of displacement and strain. The symmetry of both structure and loading allow the behaviour to be completely described by considering only a cross section of the structure. This cross section must pass through the axis of axisymmetry and be normal to the circumferential direction. The symmetry requires that the circumferential displacement be zero. In the development of this element, the restriction is imposed that the axis of axisymmetry be the z axis and that the cross section considered lie in the x-z plane. In this paper, the cylindrical coordinates \((r, \theta, z)\) will be employed. The \(x\) coordinate then is the radial coordinate \(r\) and the circumferential coordinate becomes \(\theta\). The \(z\) coordinate remains unchanged. The displacement components of interest are the \(r\) and \(z\) translations. In the context of this geometry, four components of stress are defined -- the three normal stresses \((\sigma_r, \sigma_\theta, \sigma_z)\) and the shear stress in the plane \((\tau_{rz})\). Due to the symmetry, the shear stress in the \(r-z\) plane is the only non-zero shear stress.

In heat transfer, symmetric thermal loading requires that the heat flow in the circumferential direction be zero. The two non-zero components of heat flow are in the \(r\) and \(z\) directions.

In finite element analysis, this cross section is modeled using axisymmetric elements. Axisymmetric finite elements are really rings of revolution. Each element describes a cross sectional area which is re-
volved through $360^\circ$ to form a solid ring. The complete volume is represented by modeling the right half of the cross section ($r>0$). Axisymmetric loads applied to grid points represent the total force (or heat flow) applied to the circular arc formed by revolving the grid through $360^\circ$. 
IV. ELEMENT DERIVATIONS

A. STIFFNESS MATRIX

In statics, the equation being solved is

$$[K] \{u\} = \{P\} \quad (1)$$

where $[K]$ is the structural stiffness matrix, \{u\} is the vector of nodal displacements, and \{P\} is the vector of applied forces.

The stiffness matrix can be derived from the structural potential function -- the strain energy. In matrix notation, the strain energy in an element is

$$\phi = \frac{1}{2} \int \{\sigma\}^T \{\varepsilon\} \, dVOL \quad (2)$$

where \{\sigma\} is a vector of internal stresses, \{\varepsilon\} is a vector of internal strains, and the integration is performed over the volume.

Castigliano's Theorem states that the derivative of the strain energy with respect to the displacements is equal to the applied forces.

Writing Castigliano's Theorem and substituting from equation (1)

$$\frac{\partial \phi}{\partial \{u\}} = \{P\} = [K] \{u\} \quad (3)$$

Before the derivative can be evaluated, the strain energy must be written in terms of the displacements.

First the constitutive equation relating \{\sigma\} and \{\varepsilon\} can be written

$$\{\sigma\} = [D] \{\varepsilon\} \quad (4)$$

Four components of stress and strain are defined for axisymmetric
structures:

\[
\{ \sigma \} = \begin{pmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{rz} \end{pmatrix} \quad ; \quad \{ \varepsilon \} = \begin{pmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{pmatrix}
\]

The constitutive matrix \( [D] \) is given in Timoshenko\(^2\) as

\[
\begin{bmatrix}
1 & \frac{v}{1-v} & \frac{v}{1-v} & 0 \\
\frac{v}{1-v} & 1 & \frac{v}{1-v} & 0 \\
\frac{v}{1-v} & \frac{v}{1-v} & 1 & 0 \\
0 & 0 & 0 & \frac{1-2v}{2(1-v)}
\end{bmatrix}
\]

Substituting into the expression for the strain energy yields

\[
\phi = \frac{1}{2} \iint \{ \varepsilon \}^T [D] \{ \varepsilon \} \, dV
\]

Next an expression relating strains to displacements is written. The four components of strain are defined as

\[
\begin{pmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial r} \\ \frac{u}{r} + \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \end{pmatrix}
\]
Continuous displacement functions are written using isoparametric shape functions

\[ u = \sum_i N_i \ u_i \]  
\[ v = \sum_i N_i \ v_i \]  

(8)

where \( u_i \) and \( v_i \) are displacement components at the \( i^{th} \) grid point, and \( N_i \) is the shape function associated with the \( i^{th} \) grid point.

The element coordinate system and the shape functions for an eight noded isoparametric quadrilateral element as defined by Zienkiewicz\(^2\) are shown in Figure 1. Arbitrarily shaped elements are mapped into the 2 by 2 square shown by the same shape functions:

\[ r = \sum_i N_i(\alpha, \beta) \ r_i \]  
\[ z = \sum_i N_i(\alpha, \beta) \ z_i \]  

(9)

where \( r_i \) and \( z_i \) are the \( r, z \) coordinates at node \( i \).

Since the shape functions vary quadratically in \( \alpha \) and \( \beta \), the displacement also vary quadratically within the element. The mapping of coordinates allows curved geometries to be modeled. This concept gives rise to the name isoparametric -- literally "same parameters". The same parameters are used to describe both the geometry and the displacements. One of the properties of the shape functions is that at grid \( i \), the \( i^{th} \) shape function has a value of 1.0 and all other shape functions are identically zero. As described by Zienkiewicz, isoparametric shape functions inherently satisfy the conditions necessary for convergence. These conditions are:

1) the element is strain free for rigid body motion, and
2) a constant strain condition can be represented.
FIGURE 1. ELEMENT COORDINATE SYSTEM AND SHAPE FUNCTIONS

Element coordinate system

(-1,+1)   (0,+1)   (+1,+1)

(-1,0)     (0,0)    (+1,0)

(-1,-1)   (0,-1)   (+1,-1)

Element shape functions

Corner nodes (odd indices)

\[ N_i = \frac{1}{2} (1+\alpha_i) (1+\beta_o) (\alpha_o+\beta_o-1) \]

\[ \frac{\partial N_i}{\partial \alpha} = \frac{1}{2} \alpha_i (1+\beta_o) (2\alpha_o+\beta_o) \]

\[ \frac{\partial N_i}{\partial \beta} = \frac{1}{2} \beta (1+\alpha_o)(\alpha_o+2\beta_o) \]

Midside nodes (even indices)

\[ N_i = \frac{1}{2} (1-\alpha) (1+\beta_o) \beta_i^2 + \frac{1}{2} (1-\beta) (1+\alpha_o) \alpha_i^2 \]

\[ \frac{\partial N_i}{\partial \alpha} = -\alpha (1+\beta_o) \beta_i^2 + \frac{1}{2} (1-\beta) \alpha_i \]

\[ \frac{\partial N_i}{\partial \beta} = -\beta (1+\alpha_o) \alpha_i^2 + \frac{1}{2} (1-\alpha^2) \beta_i \]

where \( \alpha = \alpha_i \) and \( \beta = \beta_i \)

\( a_i, \beta_i \) are the values of coordinates \( \alpha, \beta \) at node \( i \)
Substituting the displacement relations into equation 7 yields the strain displacement relations

\[
\{ \varepsilon \} = [B] \{ u \} \tag{10}
\]

where the matrix \([B]\) is defined as

\[
[B] = \begin{bmatrix}
\frac{\partial N_1}{\partial r} & 0 & \frac{\partial N_2}{\partial r} & 0 & \frac{\partial N_3}{\partial r} & 0 & \frac{\partial N_4}{\partial r} & 0 & \frac{\partial N_5}{\partial r} & 0 & \frac{\partial N_6}{\partial r} & 0 & \frac{\partial N_7}{\partial r} & 0 & \frac{\partial N_8}{\partial r} \\
\frac{N_1}{r} & 0 & \frac{N_2}{r} & 0 & \frac{N_3}{r} & 0 & \frac{N_4}{r} & 0 & \frac{N_5}{r} & 0 & \frac{N_6}{r} & 0 & \frac{N_7}{r} & 0 & \frac{N_8}{r} \\
\frac{\partial N_1}{\partial \theta} & 0 & \frac{\partial N_2}{\partial \theta} & 0 & \frac{\partial N_3}{\partial \theta} & 0 & \frac{\partial N_4}{\partial \theta} & 0 & \frac{\partial N_5}{\partial \theta} & 0 & \frac{\partial N_6}{\partial \theta} & 0 & \frac{\partial N_7}{\partial \theta} & 0 & \frac{\partial N_8}{\partial \theta} \\
\frac{\partial N_1}{\partial \alpha} & \frac{\partial N_2}{\partial \alpha} & \frac{\partial N_3}{\partial \alpha} & \frac{\partial N_4}{\partial \alpha} & \frac{\partial N_5}{\partial \alpha} & \frac{\partial N_6}{\partial \alpha} & \frac{\partial N_7}{\partial \alpha} & \frac{\partial N_8}{\partial \alpha} \\
\end{bmatrix}
\]

To evaluate the derivative terms in \([B]\) the change of independent coordinate must be considered. This is accomplished by applying the chain rule and differentiating the shape functions with respect to the local coordinates \(\beta\) and \(\alpha\).

\[
\frac{\partial N_i}{\partial \beta} = \frac{\partial N_i}{\partial r} \frac{\partial r}{\partial \beta} + \frac{\partial N_i}{\partial \theta} \frac{\partial \theta}{\partial \beta} + \frac{\partial N_i}{\partial \alpha} \frac{\partial \alpha}{\partial \beta}
\]

\[
\frac{\partial N_i}{\partial \alpha} = \frac{\partial N_i}{\partial r} \frac{\partial r}{\partial \alpha} + \frac{\partial N_i}{\partial \theta} \frac{\partial \theta}{\partial \alpha} + \frac{\partial N_i}{\partial \alpha} \frac{\partial \alpha}{\partial \alpha}
\]

In matrix notation, this becomes

\[
\begin{bmatrix}
\frac{\partial N_i}{\partial \beta} \\
\frac{\partial N_i}{\partial \alpha}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial r}{\partial \beta} & \frac{\partial \theta}{\partial \beta} \\
\frac{\partial r}{\partial \alpha} & \frac{\partial \theta}{\partial \alpha}
\end{bmatrix} \begin{bmatrix}
\frac{\partial N_i}{\partial r} \\
\frac{\partial N_i}{\partial \theta}
\end{bmatrix} \tag{11}
\]
The square matrix is defined as the Jacobian

$$
\begin{bmatrix}
\frac{∂r}{∂r} & \frac{∂z}{∂r} \\
\frac{∂r}{∂θ} & \frac{∂z}{∂θ} \\
\frac{∂r}{∂α} & \frac{∂z}{∂α}
\end{bmatrix}
$$

(12)

The derivative in cylindrical coordinates can be determined as

$$
\begin{bmatrix}
\frac{∂N_1}{∂r} \\
\frac{∂N_1}{∂θ} \\
\frac{∂N_1}{∂α}
\end{bmatrix} = [J]^{-1}
\begin{bmatrix}
\frac{∂N_1}{∂r} \\
\frac{∂N_1}{∂θ} \\
\frac{∂N_1}{∂α}
\end{bmatrix}
$$

(13)

The Jacobian is evaluated by differentiating equations 9.

In matrix notation this becomes

$$
\begin{bmatrix}
\frac{∂N_1}{∂r} & \frac{∂N_2}{∂r} & \frac{∂N_3}{∂r} & \frac{∂N_4}{∂r} & \frac{∂N_5}{∂r} & \frac{∂N_6}{∂r} & \frac{∂N_7}{∂r} & \frac{∂N_8}{∂r} \\
\frac{∂N_1}{∂θ} & \frac{∂N_2}{∂θ} & \frac{∂N_3}{∂θ} & \frac{∂N_4}{∂θ} & \frac{∂N_5}{∂θ} & \frac{∂N_6}{∂θ} & \frac{∂N_7}{∂θ} & \frac{∂N_8}{∂θ} \\
\frac{∂N_1}{∂α} & \frac{∂N_2}{∂α} & \frac{∂N_3}{∂α} & \frac{∂N_4}{∂α} & \frac{∂N_5}{∂α} & \frac{∂N_6}{∂α} & \frac{∂N_7}{∂α} & \frac{∂N_8}{∂α}
\end{bmatrix}
\begin{bmatrix}
r_1 \\
r_2 \\
r_3 \\
r_4 \\
r_5 \\
r_6 \\
r_7 \\
r_8
\end{bmatrix}
$$

(14)

Substituting into equation (6) gives

$$\begin{align*}
\mathbf{∅} &= \frac{1}{2} \iint \{ \mathbf{u} \}^T \mathbf{B}^T \mathbf{D} \mathbf{B} \{ \mathbf{u} \} \, dVOL \\
&= \frac{1}{2} \iint \{ \mathbf{u} \}^T \mathbf{F} \mathbf{F} \mathbf{T} \{ \mathbf{u} \} \, dVOL
\end{align*}$$

(15)

The displacement vectors are treated as constants, giving

$$\begin{align*}
\mathbf{∅} &= \frac{1}{2} \iint \{ \mathbf{u} \}^T \mathbf{B} \mathbf{B}^T \mathbf{D} \mathbf{B} \{ \mathbf{u} \} \, dVOL
\end{align*}$$

(16)
Castigliano's Theorem can now be applied giving:

\[
\frac{\partial \phi}{\partial \{u\}} = \{ P \} = \int \int [B]^T[D][B] \, d\text{VOL} \{ u \} \quad (17)
\]

By comparison with equation (3), the stiffness matrix is seen as

\[
[K] = \int \int [B]^T[D][B] \, d\text{VOL} \quad (18)
\]

If desired, displacements can be transformed to a different coordinate system by the equation

\[
\{ \bar{u} \} = [T] \{ u \} \quad (19)
\]

where \([T]\) is a coordinate transformation matrix. The stiffness matrix then becomes

\[
[K] = \int \int [T]^T[B]^T[D][B][T] \, d\text{VOL} \quad (20)
\]

In cylindrical coordinates, the volume integral can be written as

\[
\text{VOL} = \int \int r \, dr \, d\theta \, dz = 2 \pi \int \int r \, dr \, dz \quad (21)
\]

where \(r\) is evaluated from equation (9).

This integral must be written in terms of the variables \(\beta\) and \(\alpha\). The relation between the two coordinate systems is defined by Zienkiewicz as

\[
dr \, dz = |J| \, d\beta \, d\alpha \quad (22)
\]

where \(|J|\) is the determinant of the Jacobian matrix. Making this change of integration variables, the stiffness matrix then becomes

\[
[K] = 2\pi \int \int [T]^T[B]_T[D][B][T] \, |J| \, r \, d\beta \, d\alpha \quad (23a)
\]
The integration is best performed numerically using Gaussian quadrature. Then the stiffness matrix is written as

\[
\begin{bmatrix}
K
\end{bmatrix} = 2 \pi \sum_{ij} \left[ T_j^T \left[ B \right]^T \left[ D \right] [B] [T] |J| r W_{ij} \right] (23b)
\]

where \( n \) is the order of the Gaussian integration required, and \( W \) is a weighting factor.

As described by Janucik, a minimum integration order of 3 is recommended for this formulation.
B. THERMOELASTIC LOADS

Equivalent nodal forces are derived by assuming a virtual displacement at element nodes and equating the internal and external work. Let \( \{ \zeta \} \) be a vector of virtual displacements at the nodes. The displacements and strains within the element then become

\[
\{ u \} = [N] \{ \zeta \} \tag{24}
\]
\[
\{ \varepsilon \} = [B] \{ \zeta \}
\]
as defined by equations (8) and (10). If \( \{ F \} \) is the vector of nodal forces, the external work is \( \{ \zeta \}^T \{ F \} \). The internal work per unit volume done by stresses and distributed forces is

\[
\{ \varepsilon \}^T \{ \sigma \} - \{ u \}^T \{ p \} \tag{25}
\]
where \( \{ p \} \) is a vector of distributed forces.

Substituting from equation (24), this becomes

\[
\{ \zeta \}^T [B]^T \{ \sigma \} - \{ \zeta \}^T [N]^T \{ p \} \tag{26}
\]

The total internal work is obtained by integrating this expression over the volume of the element. Integrating and equating the external and internal work gives

\[
\{ \zeta \}^T \{ F \} = \{ \zeta \}^T \int [B]^T \{ \sigma \} - [N]^T \{ p \} \, dVOL \tag{27}
\]

Since this relation must be valid for any virtual displacement, the equivalent nodal forces become

\[
\{ F \} = \int [B]^T \{ \sigma \} \, dVOL - \int [N]^T \{ p \} \, dVOL \tag{28}
\]
If initial stresses are specified, the equivalent nodal forces are

\[
\{ F \} = \int \left[ B \right]^T \{ \sigma \} \, dVOL \tag{29}
\]

If initial strains are specified instead of stresses, substitution of equation (4) into the above gives

\[
\{ F \} = \int \left[ B \right]^T \left[ D \right] \{ \varepsilon_0 \} \, dVOL \tag{30}
\]

Equivalent forces due to distributed loads are

\[
\{ F \} = \int \left[ N \right]^T \{ p \} \, dVOL \tag{31}
\]

The thermoelastic problem is treated as an initial strain case.

The strain vector due to a temperature change \( \Delta T \) is

\[
\{ \varepsilon_0 \} = \begin{bmatrix} \alpha \Delta T \\ \alpha \Delta T \\ \alpha \Delta T \\ 0 \end{bmatrix} \tag{32}
\]

where \( \alpha \) is the coefficient of thermal expansion.

Nodal forces due to a temperature change \( \Delta T \) thus become

\[
\{ F \} = \int \left[ B \right]^T \left[ D \right] \{ \varepsilon_0 \} \, dVOL
\]

or in cylindrical coordinates

\[
\{ F \} = 2 \pi \int \int \left[ B \right] \left[ D \right] \{ \varepsilon_0 \} \, r \, dr \, dz \tag{33}
\]

Transforming to element coordinates and using Gaussian integration, this is expressed as

\[
\{ F \} = 2 \pi \Sigma_{i,j} \left[ B \right] \left[ D \right] \{ \varepsilon_0 \} \left| J \right| \, r \, W_{i,j} \tag{34}
\]
C. STRESS RECOVERY

Stresses are calculated from equation (4),

\[
\{ \sigma \} = [ D ] \{ \varepsilon \}
\]

The strain vector must be relieved of all strains due to thermoelastic expansion since these are stress-free strains. The strain vector thus becomes

\[
\{ \varepsilon \} = [ B ]^T \{ \zeta \} - \{ \varepsilon_o \}
\]

(35)

where \{ \varepsilon_o \} is defined as in equation (32).

Stresses are evaluated at five points within each element -- the centroid and the four cornermost Gauss points as shown in Figure 2. Stresses are extrapolated linearly from the centroid through each cornermost Gauss point to each corner node. Stresses are output at the centroid and the four corner grids. At each stress point, stress invariants are calculated. These include the three principal stresses, the direction cosines associated with the first principal stress, the maximum shear, and the octahedral shear.
FIGURE 2. GAUSSIAN INTEGRATION AND STRESS RECOVERY POINTS

x - Gaussian integration points
○ - Stress recovery points

2 by 2 integration

(±0.57735, ±0.57735)

3 by 3 integration

(±0.77459, ±0.77459)
(0.0, ±0.77459)
(±0.77459, 0.0)
(0.0, 0.0)

4 by 4 integration

(±0.86114, ±0.86114)
(±0.86114, ±0.33998)
(±0.33998, ±0.86114)
(±0.33998, ±0.33998)
D. CONDUCTANCE MATRIX

The heat transfer equation is analogous to the structural equilibrium equation

\[
\begin{bmatrix}
K
\end{bmatrix}
\begin{bmatrix}
u
\end{bmatrix}
= \begin{bmatrix}
P
\end{bmatrix}
\]

In structural analysis,

- \([K]\) is the stiffness matrix,
- \(\{u\}\) is a vector of displacements, and
- \(\{P\}\) is a vector of applied loads.

In the thermal system, these are defined as:

- \([K]\) is the heat conductance matrix,
- \(\{u\}\) is the nodal temperature vector, and
- \(\{P\}\) is a vector of applied heat flows (\(F=qA\)).

In heat transfer, only one degree of freedom -- the nodal temperature -- exists per grid point.

The thermal conductance matrix can be derived from a potential function in the same way that the stiffness matrix was derived from the strain energy. The thermal potential function is defined as

\[
U = -\frac{1}{2} \int q \nabla u \, dV
\]

(36)

where \(q\) the heat flux density, and \(\nabla u\) the temperature gradient are:

\[
q = q_1 i + q_2 j + q_3 k
\]

(37)

\[
\nabla u = \frac{\partial u}{\partial x_1} i + \frac{\partial u}{\partial x_2} j + \frac{\partial u}{\partial x_3} k
\]

Forming the dot product gives

\[
\tilde{q} \cdot \nabla u = q_1 \frac{\partial u}{\partial x_1} + q_2 \frac{\partial u}{\partial x_2} + q_3 \frac{\partial u}{\partial x_3}
\]

(38)
The components of the flux $q_i$ are related to the temperature gradient by

$$q_i = -\sum_j k_{ij} \frac{\partial u}{\partial x_j}$$  \hspace{1cm} (39)$$

where $k_{ij}$ is a component of the material conductivity matrix, and $j$ is summed over the dimensions of the space.

Substituting equation (39) into equation (38) and expressing the result in matrix notation yields

$$U = \frac{1}{2} \int \left\{ \frac{\partial u}{\partial x_i} \right\}^T \left[ k_{ij} \right] \left\{ \frac{\partial u}{\partial x_j} \right\} \, dVOL$$  \hspace{1cm} (40)$$

The temperatures are assumed to vary within the element by the isoparametric relation used before

$$u = \left[ N \right] \left\{ u_0 \right\}$$  \hspace{1cm} (41)$$

where $\left\{ u_0 \right\}$ is a vector of constant nodal temperatures, and $\left[ N \right]$ is the matrix of isoparametric shape functions.

Since the nodal temperatures are constants, the derivative term is expressed as

$$\left\{ \frac{\partial u}{\partial x_i} \right\} = \left[ \frac{\partial N}{\partial x_i} \right] \left\{ u_0 \right\}$$  \hspace{1cm} (42)$$

Substituting this into equation (40) gives

$$U = \frac{1}{2} \int \left\{ u_0 \right\}^T \left[ \frac{\partial N}{\partial x_i} \right]^T \left[ k_{ij} \right] \left[ \frac{\partial N}{\partial x_j} \right] \left\{ u_0 \right\} \, dVOL$$  \hspace{1cm} (43)$$

which is of the desired form

$$U = \frac{1}{2} \left\{ u_0 \right\}^T \left[ K \right] \left\{ u_0 \right\}$$  \hspace{1cm} (44)$$

The conductance matrix $\left[ K \right]$ is

$$\left[ K \right] = \int \left[ \frac{\partial N}{\partial x_i} \right]^T \left[ k_{ij} \right] \left[ \frac{\partial N}{\partial x_j} \right] \, dVOL$$  \hspace{1cm} (45)$$
In axisymmetric analysis, the radial and axial components of heat flow are considered. For homogeneous isotropic materials, the material matrix is

\[
[k] = \begin{bmatrix}
k & 0 \\
0 & k \\
\end{bmatrix} \tag{46}
\]

The derivative matrix becomes

\[
\begin{bmatrix}
\frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_2}{\partial z} & \frac{\partial N_4}{\partial r} & \frac{\partial N_5}{\partial r} & \frac{\partial N_6}{\partial r} & \frac{\partial N_7}{\partial r} & \frac{\partial N_8}{\partial r} \\
\frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_4}{\partial z} & \frac{\partial N_5}{\partial z} & \frac{\partial N_6}{\partial z} & \frac{\partial N_7}{\partial z} & \frac{\partial N_8}{\partial z} \\
\end{bmatrix} \tag{47}
\]

where as before

\[
\begin{bmatrix}
\frac{\partial N_1}{\partial r} \\
\frac{\partial N_1}{\partial z} \\
\end{bmatrix} = [J]^{-1} \begin{bmatrix}
\frac{\partial N_1}{\partial \beta} \\
\frac{\partial N_1}{\partial \alpha} \\
\end{bmatrix} \tag{48}
\]

where \([J]\) is the Jacobian matrix defined in equation (14), and \(\beta, \alpha\) are element local coordinates as shown in Figure 1.

Writing the volume integral as before and using Gaussian integration, the thermal conductance matrix is written as

\[
[K] = \frac{r}{r} \begin{bmatrix}
\frac{\partial N_1}{\partial r} & \frac{\partial N_1}{\partial z} \\
\frac{\partial N_2}{\partial r} & \frac{\partial N_2}{\partial z} \\
\frac{\partial N_4}{\partial r} & \frac{\partial N_4}{\partial z} \\
\frac{\partial N_5}{\partial r} & \frac{\partial N_5}{\partial z} \\
\frac{\partial N_6}{\partial r} & \frac{\partial N_6}{\partial z} \\
\frac{\partial N_7}{\partial r} & \frac{\partial N_7}{\partial z} \\
\frac{\partial N_8}{\partial r} & \frac{\partial N_8}{\partial z} \\
\end{bmatrix} \begin{bmatrix}
k & 0 \\
0 & k \\
\end{bmatrix} \begin{bmatrix}
\frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \cdots & \frac{\partial N_8}{\partial r} \\
\frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \cdots & \frac{\partial N_8}{\partial z} \\
\end{bmatrix} W_{ij} \mid J \mid r \tag{49}
\]
In equation 49, $i$ and $j$ are summed over the order of the Gaussian integration. $W_{ij}$ is a weighting function associated with the particular Gauss point.
V. ADDITIONS TO NASTRAN

NASTRAN has a capability for adding additional elements. This is most readily accomplished via the DUMMY element technique. Internal to NASTRAN exists a group of dummy subroutines. They are included only so that calls to these element dependent routines can be made. To implement a new element capability, functioning element routines must be written, compiled, and link edited into the NASTRAN program. Each of the routines must perform a specific task. Subroutine names include the identifying number of the DUMMY element (1-9). A subroutine KDUMi (i is the DUMMY element number) uses a NASTRAN table of element information to calculate the element stiffness or conductance matrix. Subroutine DUMi calculates equivalent element nodal loads due to thermal loading. Subroutine PLUMi sets up a connection array of grid points for plotting the undeformed and deformed structure. Two subroutines are used to calculate element stresses and a third is used to output the stress information in convenient formats. Subroutine SNUMi1 obtains element material data and calculates quantities which are constant for each element. This information is passed to subroutine SNUMi2 for final calculation of stresses and invariants. Headings and stress data are output in subroutine OUMi. A listing of these subroutines is given in Appendix A.

There are a number of characteristics which are common to all the subroutines. First all subroutines are designed to perform calculations for one element at a time. The subroutines are called repeatedly by the
driving subroutine until all elements have been processed. Secondly, all element information is passed to the new subroutines in labeled common areas. Element data is in regular NASTRAN table format but the amount of data is specified by the user as input data at execution time. The number of grid points connected, the amount of property data, and other connection data is specified on the ALUMI card. This data is used by the driving subroutines when preparing data for the element routines. These driving routines place the portion of the tables needed for the element currently being processed into the common areas for use by the element routines.

Finally a number of NASTRAN utility subroutines are available to the new subroutines. Some of the subroutines must be used to obtain material properties, temperatures, and displacements. Other subroutines can optionally be used for common matrix operations, such as multiplication and inversion, or to exercise other convenience options as desired. The utility subroutines and their functions are listed in Appendix A along with the element subroutines.

In NASTRAN all grid points are assumed to have 6 degrees of freedom (dof) -- 3 translations and 3 rotations -- except in heat transfer where only 1 dof per grid is defined. This convention allows connection of many diverse element types. The matrices generated in the element routines must associate 6 dof with each grid point. For the 8 noded CQDRNG8 element, only 2 dof per grid have stiffness associated with them. Thus the 16 by 16 full stiffness matrix (2 dof times 8 grids)
must be expanded to a 48 by 48 stiffness matrix (6 dof times 8 grids) by including null rows and columns before it can be added into the overall stiffness matrix for the structure. In the KUMMi routine, the full stiffness matrix is not calculated at one time. Instead NASTRAN uses the concept of a "pivot" grid. The element routine is called once for each grid in the element. The grid associated with the particular call to the element routine is called the "pivot" grid. The eight 6 by 6 matrix partitions, as shown in Figure 3, which connect the "pivot" grid to each of the eight element grids (including itself) are calculated and inserted into the overall stiffness matrix. This technique was used when NASTRAN was first developed because of the relatively simple elements available then. It was usually cheaper to recalculate portions of the element matrix than to store and retrieve it. The advent of higher order isoparametric elements and the numerical integration required for these elements made this an undesirable technique. A new technique in which the entire stiffness matrix is calculated and stored is currently being used for all new elements added to the MacNeal-Schwender version of NASTRAN (MSC/NASTRAN). Since this newer technique is not now available for DUMMY elements, the older technique was used. The matrix generation time for this element is comparable to the times for existing elements. The new technique will be available for DUMMY elements in the future. The generation time for a CQDRNG8 element on an IBM 370/158 is approximately .30 seconds. Generation
**FIGURE 3. STIFFNESS MATRIX PARTITIONS**

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaa</td>
<td>Kab</td>
<td>Kac</td>
<td>Kad</td>
<td>Kae</td>
<td>Kaf</td>
<td>Kag</td>
<td>Kah</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>Kba</td>
<td>Kbb</td>
<td>Kbc</td>
<td>Kbd</td>
<td>Kbe</td>
<td>Kbf</td>
<td>Kbg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>Kca</td>
<td>Kcb</td>
<td>Kcc</td>
<td>Kcd</td>
<td>Kce</td>
<td>Kcf</td>
<td>Kcg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>Kda</td>
<td>Kdb</td>
<td>Kdc</td>
<td>Kdd</td>
<td>Kde</td>
<td>Kdf</td>
<td>Kdg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>Kea</td>
<td>Keb</td>
<td>Kec</td>
<td>Kend</td>
<td>Kee</td>
<td>Kef</td>
<td>Keq</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>Kfa</td>
<td>Kfb</td>
<td>Kfc</td>
<td>Kfd</td>
<td>Kfe</td>
<td>Kff</td>
<td>Kfg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>Kga</td>
<td>Kgb</td>
<td>Kgc</td>
<td>Kgd</td>
<td>Kge</td>
<td>Kgf</td>
<td>Kgg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>Kha</td>
<td>Khb</td>
<td>Khc</td>
<td>Khd</td>
<td>Khe</td>
<td>Khf</td>
<td>Kng</td>
</tr>
</tbody>
</table>
times for similar elements range from .1 to .8 seconds.

Three new NASTRAN data cards are required for this element. These data cards (ADUM3, CQDRNG8, and PQDRNG8) are described in Appendix B. The new data cards define the properties and connectivity of the element. Existing NASTRAN data cards are used to locate grid points, constrain dof, apply loads, and define material properties.
VI. EXAMPLE PROBLEMS

The CQDRN8 element has been implemented in version 34 of the MSC/NASTRAN finite element program. A number of classical problems have been solved using this element as implemented in NASTRAN. In all cases, the element showed good correlation with theoretical solutions. Three problems each will be used to demonstrate elastic and heat transfer capabilities. In addition to these theoretical test cases, an attempt was made to solve a thermoelastic problem previously solved using finite difference techniques. The example problems to be discussed are listed below in the order of presentation:

A. Bending of a circular simply supported plate. Deflections are determined for a center load. Deflections and stresses are determined for a load on a concentric circular ring.

B. Thermal expansion of a circular disk. Deflections are calculated for a disk which has expanded due to an increase in temperature.

C. Thermal expansion of a hollow sphere. Deflections and stresses are calculated for two temperature distributions:
   1. a uniform temperature throughout the sphere which is higher than its reference temperature, and
   2. a linear temperature variation through the thickness of the sphere.

D. Heat flow in a circular disk. A disk with a small center hole has the temperatures at its inner and outer surfaces constrained. Internal temperatures are determined using several finite element discretizations to demonstrate convergence. The same problem is also solved using an existing NASTRAN element to provide a comparison.
E. Heat flow in an axial rod.
Internal temperatures in a rod with constrained end
temperatures are determined.

F. Heat flow in a hollow sphere.
Temperature distributions are determined for two sets of
boundary conditions:
1. constrained temperatures at the inner and outer
   spherical surfaces, and
2. constrained temperatures at the inner spherical
   surface but convection at the outer surface.

G. Thermoelastic stresses in a hollow cylinder.
Stresses due to transient temperature distributions are
determined for particular times and compared to a finite
difference solution.

For convenience, computer results have been reduced to tabular and
graphical form.

This element suffers the typical finite element problem with
regard to stresses. Compatibility of stresses in adjacent elements
is not ensured by the formulation. Consequently when stresses are
calculated for grid point locations which are common to several elements,
different stresses result. The most common method of treating this
continuity is to average each stress component at common grid
points. The stresses presented in this thesis were calculated in this
manner.
A. BENDING OF SIMPLY SUPPORTED FLAT PLATE

The thin circular flat plate of Figure 4 was analyzed using the finite element mesh shown. Theoretical results are from Timoshenko. Two loading conditions were considered. First, a load was applied at the center. Figure 4 is a graph of normalized deflection versus radial position. Results are presented for the theoretical solution, the CQDRNG8 element solution, and the CTRIA6 element solution. The CTRIA6 is the triangular counterpart of the CQDRNG8 and currently exists in NASTRAN. It should be noted that the theoretical solution is for bending deflection only whereas the finite element solutions include the effects of shear. Tabular results are presented in Table 1. The finite element solutions are nearly identical and are within 2.5% of theory.

The second loading condition considered is that of a uniform load applied along a concentric circle. Displacement results are presented graphically in Figure 5 and in tabular form in Table 2. Again, the displacements agree very well with the theoretical result from Timoshenko. The maximum difference was less than 1.25%. Stresses were also calculated for this case. Stress results are given in Figure 6 and Table 3. Both the radial and hoop stresses agree to within 11.1% of theoretical results.
FIGURE 4
Deflections of simply supported plate with center load.
TABLE 1. Deflections of simply supported plate with center load

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>DEFLECTIONS (10^{-4} inches)</th>
<th>CQDRNG8</th>
<th>THEORY</th>
<th>% DIFFERENCE</th>
<th>TRIAX6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.925</td>
<td>1.879</td>
<td>2.41</td>
<td></td>
<td>1.913</td>
</tr>
<tr>
<td>1.25</td>
<td>1.824</td>
<td>1.802</td>
<td>1.21</td>
<td></td>
<td>1.814</td>
</tr>
<tr>
<td>2.50</td>
<td>1.648</td>
<td>1.635</td>
<td>.79</td>
<td></td>
<td>1.644</td>
</tr>
<tr>
<td>3.75</td>
<td>1.421</td>
<td>1.413</td>
<td>.61</td>
<td></td>
<td>1.418</td>
</tr>
<tr>
<td>5.00</td>
<td>1.161</td>
<td>1.155</td>
<td>.53</td>
<td></td>
<td>1.159</td>
</tr>
<tr>
<td>6.25</td>
<td>.880</td>
<td>.876</td>
<td>.48</td>
<td></td>
<td>.878</td>
</tr>
<tr>
<td>7.50</td>
<td>.588</td>
<td>.585</td>
<td>.47</td>
<td></td>
<td>.587</td>
</tr>
<tr>
<td>8.75</td>
<td>.292</td>
<td>.291</td>
<td>.59</td>
<td></td>
<td>.291</td>
</tr>
<tr>
<td>10.00</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td></td>
<td>.0</td>
</tr>
</tbody>
</table>
FIGURE 5
Deflections of simply supported plate with concentric circular load.
### TABLE 2. DEFLECTIONS OF SIMPLY SUPPORTED CIRCULAR PLATE WITH LOAD ON CONCENTRIC CIRCLE

<table>
<thead>
<tr>
<th>Radius</th>
<th>Deflections (10^-3 inches)</th>
<th>CQDRNG8</th>
<th>Theory</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5</td>
<td>3.664</td>
<td>3.709</td>
<td>1.21</td>
<td></td>
</tr>
<tr>
<td>18.75</td>
<td>3.217</td>
<td>3.252</td>
<td>1.08</td>
<td></td>
</tr>
<tr>
<td>25.0</td>
<td>2.658</td>
<td>2.681</td>
<td>.86</td>
<td></td>
</tr>
<tr>
<td>31.25</td>
<td>2.026</td>
<td>2.043</td>
<td>.83</td>
<td></td>
</tr>
<tr>
<td>37.5</td>
<td>1.358</td>
<td>1.368</td>
<td>.73</td>
<td></td>
</tr>
<tr>
<td>43.75</td>
<td>.676</td>
<td>.681</td>
<td>.73</td>
<td></td>
</tr>
<tr>
<td>50.0</td>
<td>.0</td>
<td>.0</td>
<td>.0</td>
<td></td>
</tr>
</tbody>
</table>
FIGURE 6
Stresses in simply supported circular plate with load on concentric circle
TABLE 3 - Stresses in simply supported circular plate with concentric loading

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>$\sigma_R$ CQDRNG8</th>
<th>$%$ DIFF</th>
<th>$\sigma_\theta$ CQDRNG8</th>
<th>$%$ DIFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5</td>
<td>105.</td>
<td>3.96</td>
<td>105.</td>
<td>3.96</td>
</tr>
<tr>
<td>25.0</td>
<td>48.</td>
<td>4.35</td>
<td>72.</td>
<td>1.41</td>
</tr>
<tr>
<td>37.5</td>
<td>20.</td>
<td>11.11</td>
<td>50.</td>
<td>2.04</td>
</tr>
<tr>
<td>50.</td>
<td>6.</td>
<td>-</td>
<td>35.</td>
<td>9.38</td>
</tr>
</tbody>
</table>
B. THERMAL EXPANSION OF CIRCULAR DISK

To test the thermoelastic loading capability, a circular disk with a center hole was subjected to a uniform temperature change. The temperature of the ring was raised by 100 °F resulting in a thermoelastic expansion of the disk. Figure 7 is a plot of the finite element model. Grids are indicated with an *. Grid identification numbers are to the right of the *'s. In the center of each element is the element identification number. The disk was unconstrained in the radial direction and was constrained at the outside in the axial direction. As shown in Timoshenko, the disk should undergo a stress free radial and axial growth. Displacement results are shown in Figure 8 and Table 4. The predicted displacements are exactly those from theory (0.0 % error). As expected all stresses were zero.
FIGURE 7
Plot of circular disk model.
FIGURE 8
Deflections of disk with center hole - uniform temperature change.
TABLE 4 - Thermal deflections in disk

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>DEFLECTIONS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CQDRNG8</td>
<td>THEORY</td>
</tr>
<tr>
<td>10.0</td>
<td>.0010</td>
<td>.0010</td>
</tr>
<tr>
<td>15.0</td>
<td>.0015</td>
<td>.0015</td>
</tr>
<tr>
<td>20.0</td>
<td>.0020</td>
<td>.0020</td>
</tr>
<tr>
<td>25.0</td>
<td>.0025</td>
<td>.0025</td>
</tr>
<tr>
<td>30.0</td>
<td>.0030</td>
<td>.0030</td>
</tr>
<tr>
<td>35.0</td>
<td>.0035</td>
<td>.0035</td>
</tr>
<tr>
<td>40.0</td>
<td>.0040</td>
<td>.0040</td>
</tr>
<tr>
<td>45.0</td>
<td>.0045</td>
<td>.0045</td>
</tr>
<tr>
<td>50.0</td>
<td>.0050</td>
<td>.0050</td>
</tr>
</tbody>
</table>
C. THERMAL EXPANSION OF HOLLOW SPHERE

A symmetric section of a hollow sphere was modeled as shown in the plot of Figure 9. This sphere was subjected to two different temperature distributions. Symmetric boundary conditions were applied to the cuts of symmetry. Theoretical results from Timoshenko are given.

In the first case, the sphere is at a uniform temperature which is \(100^\circ\text{F}\) above its reference temperature. The sphere expands radially due to the new temperature distribution. As shown in Figure 10 and Table 5, the CQDRNG8 displacements are exactly those expected (0.0 % error). Since the sphere is unconstrained, all stresses are zero.

The second case considered a linear variation of temperature through the thickness. This temperature distribution caused internal bending stresses. Theoretical bending stresses at the surface for a \(100^\circ\text{F}\) temperature difference are 2080 psi. The CQDRNG8 predicted stresses of 1950 psi at the OD and 2250 at the ID. These predicted stresses are within 8 % of theory.
FIGURE 9

Plot of hollow sphere model
FIGURE 10
Thermal expansion of sphere
TABLE 5 - Thermal expansion of hollow sphere

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>RADIAL DEFLECTION</th>
<th>CQDRNG8</th>
<th>THEORY</th>
<th>% DIFFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.0</td>
<td>.00400</td>
<td>.00400</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>42.5</td>
<td>.00425</td>
<td>.00425</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>45.0</td>
<td>.00450</td>
<td>.00450</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>47.5</td>
<td>.00475</td>
<td>.00475</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>50.0</td>
<td>.00500</td>
<td>.00500</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
D. HEAT FLOW IN A CIRCULAR DISK

To demonstrate the convergence of the CQDRNG8 element, the problem of heat flow in a circular disk was solved. The disk has a small centerhole constrained to 0.0°F. The outer diameter of the disk is constrained to 100.0°F. This problem was solved using the three different discretizations shown in Figure 11. The models contained 8, 2, and 1 CQDRNG8 elements respectively. The problem was also solved using three similar gridworks of CTRIARG elements. These models are shown in Figure 12. The CTRIARG is a three noded triangular ring element which currently exists in NASTRAN. Each CQDRNG8 element has been filled with eight CTRIARG elements and an additional degree of freedom has been added at the center. The CTRIARG element has a linear temperature variation as compared to the quadratic variation of the CQDRNG8 element.

The results of this study are shown in Figure 13 and Table 6. Theoretical results are from Carslaw. For the coarsest model, the 1 CQDRNG8 element with 2 unconstrained degrees of freedom gives the same results as 8 CTRIARG elements having 3 dof. The temperatures are in error by 10.57%. The 2 element CQDRNG8 model with 7 unconstrained dof predicts temperatures which are in error by 1.58 to 12.16%. The CTRIARG model with 16 elements and 9 unconstrained dof predicts temperatures greater in error. The error in the CTRIARG model ranges from 2.23 to 13.70%. In the finest models, the CQDRNG8 with 8 elements and 27 unconstrained dof has a maximum error of 9.96%. The CTRIARG model with 64 elements and 35 unconstrained dof had a maximum error of 12.33%.
In all cases, the CQDRNG8 models had one-eighth the number of elements and at least one-fourth fewer degrees of freedom than the TRIARG models. For the coarsest model, the single CQDRNG8 element was comparable to the eight TRIARG elements. For the other models, a moderate increase in accuracy was realized. The CQDRNG8 element does exhibit the convergence characteristic. Each refinement of the model resulted in improved accuracy at corresponding locations.
FIGURE 11 a
Circular disk model
1 CQDRNG8 element
2 unconstrained dof
FIGURE 11 b
Circular disk model
2 CQDRNG8 elements
7 unconstrained dof

4/26/77
FIGURE 11 c
Circular disk model
8 CQDRNG8 elements
27 unconstrained dof

1 W/28/77 1

CQDRNG9 THERMAL TEST
ENFORCED TEMPS AT CENTER AND 60
UNDERFORMED SHAPE
FIGURE 12 a
Circular disk model
8 CTRIARG elements
3 unconstrained dof
FIGURE 12 b
Circular disk model
16 TRIANG elements
9 unconstrained dof
FIGURE 12 c
Circular disk model
64 TRIANG elements
35 unconstrained dof
FIGURE 13
Circular disk temperatures
Convergence test

- THEORY
  - ○ 64 TRIARG (35 dof)
  - □ 16 TRIARG (9 dof)
  - △ 8 TRIARG (3 dof)
  - × 8 CQDRNG8 (27 dof)
  - + 2 CQDRNG8 (7 dof)
  - ○ 1 CQDRNG8 (2 dof)
<table>
<thead>
<tr>
<th>RADIUS</th>
<th>TEMPERATURES</th>
<th>THEORY</th>
<th>CTRIARG</th>
<th>RESULTS</th>
<th>35 dof</th>
<th>% Diff</th>
<th>9 dof</th>
<th>% Diff</th>
<th>3 dof</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7.125</td>
<td>50.1942</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>13.250</td>
<td>66.0527</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>19.375</td>
<td>75.7660</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>25.500</td>
<td>82.7878</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>31.625</td>
<td>88.2906</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>37.750</td>
<td>92.8161</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>43.875</td>
<td>96.6246</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>50.000</td>
<td>100.000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>TEMPERATURES</th>
<th>THEORY</th>
<th>CQDRUG8</th>
<th>RESULTS</th>
<th>27 dof</th>
<th>% Diff</th>
<th>7 dof</th>
<th>% Diff</th>
<th>2 dof</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7.125</td>
<td>50.1942</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>13.250</td>
<td>66.0527</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>19.375</td>
<td>75.7660</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>25.500</td>
<td>82.7878</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>31.625</td>
<td>88.2906</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>37.750</td>
<td>92.8161</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>43.875</td>
<td>96.6246</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>50.000</td>
<td>100.000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
E. HEAT FLOW IN AN AXIAL ROD

Temperatures were determined in an axial rod with constrained end temperatures. The theoretical solution from Carslaw predicts a linear temperature variation through the rod. The model and results are shown in Figure 14. The CQDRNG8 model predicts the theoretical results exactly.
FIGURE 14
Temperatures in axial rod
Enforced end temperatures
F. HEAT FLOW IN A HOLLOW SPHERE

The final thermal test case is that of a hollow sphere. The sphere is analyzed for two sets of boundary conditions. The first case is that of constrained temperatures at both the inner and outer diameters. The theoretical solution is presented in Carslaw. The finite element model is the same as that shown in Figure 9. Figure 15 and Table 7 show both the theory and CQDRNG8 results. For this case, the CQDRNG8 results are exact (0.0 % error).

For the second case, the inner diameter temperatures are constrained and the outer diameter is subjected to convection. To represent the convection boundary condition, special NASTRAN heat transfer boundary condition elements (CHBDY) must be used in conjunction with the CQDRNG8 elements. The theory for this case is also found in Carslaw. Figure 16 and Table 8 show the comparison between theory and CQDRNG8 results. Results agree to within 2.0 %.
FIGURE 15
Temperature in hollow sphere
Enforced temperature at surfaces
TABLE 7 - Temperatures in hollow sphere
Enforced temperatures at surfaces

<table>
<thead>
<tr>
<th>RADIUS</th>
<th>TEMPERATURES</th>
<th>% DIFFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CQDRNG8</td>
<td>THEORY</td>
</tr>
<tr>
<td>40.0</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>42.5</td>
<td>129.41</td>
<td>129.41</td>
</tr>
<tr>
<td>45.0</td>
<td>155.56</td>
<td>155.56</td>
</tr>
<tr>
<td>47.5</td>
<td>178.95</td>
<td>178.95</td>
</tr>
<tr>
<td>50.0</td>
<td>200.00</td>
<td>200.00</td>
</tr>
</tbody>
</table>
FIGURE 16
Temperatures in hollow sphere
Enforced temperatures at ID
Convection at OD
<table>
<thead>
<tr>
<th>RADIUS</th>
<th>TEMPERATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CQDRNG8</td>
</tr>
<tr>
<td>40.0</td>
<td>100.00</td>
</tr>
<tr>
<td>42.5</td>
<td>127.97</td>
</tr>
<tr>
<td>45.0</td>
<td>152.84</td>
</tr>
<tr>
<td>47.5</td>
<td>177.05</td>
</tr>
<tr>
<td>50.0</td>
<td>199.04</td>
</tr>
</tbody>
</table>

TABLE 8 - Temperatures in hollow sphere
Enforced temperature at ID
Convection at OD
G. THERMOELASTIC STRESSES IN A HOLLOW CYLINDER

Schlottner presented a finite difference solution for transient thermal stresses in a hollow cylinder. The cylinder, at a uniform initial temperature and with convection at the outer diameter, was subjected to an increase in the ambient temperature. The resulting transient temperature distributions create internal stresses. Since the CQDRNG8 element does not include a transient heat transfer capability, the heat transfer portion of the problem could not be solved. However thermal stresses could be determined at particular time steps if a reasonable estimate of the temperature distribution could be made. Schlottner's solution gives the temperatures at the ID and OD of the hollow cylinder at various times. For a first approximation, the temperature distribution through the cylinder was assumed to be linear. The finite element model is shown in Figure 17. As shown in Table 9, the stresses for the two time steps listed are approximately 10.0 to 25.0% lower than those predicted by Schlottner. A second approximation of the temperature distribution was arrived at by solving the steady state heat problem using boundary conditions from Schlottner's solution. At the 26th time step of Schlottner's solution, the OD temperature was close to a steady state condition but the ID temperature was not. For the steady state problem solved, the ID temperature was constrained to the value predicted by Schlottner and the convection boundary condition at the OD was retained. As
FIGURE 17
Plot of hollow cylinder model.
TABLE 9 - Stresses in hollow cylinder

<table>
<thead>
<tr>
<th>TIME</th>
<th>SCHLOTTNER STRESS</th>
<th>LINEAR TEMP STRESS</th>
<th>% DIFF</th>
<th>SS TEMP STRESS</th>
<th>% DIFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>OD</td>
<td>ID</td>
<td>OD</td>
<td>ID</td>
<td>OD</td>
</tr>
<tr>
<td>15</td>
<td>16000</td>
<td>19000</td>
<td>12000</td>
<td>14000</td>
<td>25.0</td>
</tr>
<tr>
<td>26</td>
<td>11000</td>
<td>11000</td>
<td>10000</td>
<td>8000</td>
<td>9.1</td>
</tr>
</tbody>
</table>
shown in Table 9, this resulted in poorer correlation at the ID but much poorer correlation at the OD.

This comparison is by no means a rigorous test of the CQDAN8 element's capabilities and is included only to provide a comparison to another numerical method. To determine the correct thermoelastic stresses, the correct temperature distribution must be known. Considering the approximate nature of the temperature assumption, the predicted stresses are quite reasonable.
VII. SUMMARY

The development and application of an axisymmetric finite element for the NASTRAN computer program has been presented. The element is an eight noded isoparametric quadrilateral ring with an assumed quadratic displacement and temperature function. The element has been named CQDRNG8. Element capabilities include elastic stiffness for homogeneous isotropic materials, thermoelastic loading, heat conduction for homogeneous isotropic materials, deformed and undeformed plotting, and stress recovery. Stress invariants calculated include the principal stresses and directions, the maximum shear, and the octahedral shear. The user has the option of specifying a 2 by 2, 3 by 3, or 4 by 4 order of Gaussian integration. To be consistent with other NASTRAN elements, the capability to specify arbitrary output coordinate systems has been included. Stiffness matrix generation times are comparable to other NASTRAN elements. On the IBM 370/158 series computer, stiffness matrix generation time is .30 seconds. Since the element is installed in the NASTRAN program, all of the general conveniences and capabilities of NASTRAN are also available. Some of these are convenient input, essentially unlimited problem size, efficient matrix operations, and a limited mesh generation capability. Grid locations can be generated in NASTRAN as usual but element connections cannot be generated. A simple FORTRAN program has been written by the author to generate element connection cards consistent with the generation scheme of the NASTRAN mesh generation program MSGMESH.
The accuracy and usefulness of the element has been demonstrated in several sample problems. The first problem was a commonly encountered bending problem—a simply supported circular plate. Deflections were within 2.5% and stresses were within 12.0% of theory. The second problem demonstrates the thermoelastic loading capability. An axially constrained flat circular plate with center hole was subjected to a uniform temperature change resulting in stress free radial and axial expansion of the plate. Calculated deflections were exactly those expected. The third case is a thermoelastic loading resulting in bending in a hollow sphere. Calculated bending stresses are within 8.0% of theory. The fourth problem demonstrates the heat conductance capability and element convergence. Internal temperatures are determined in a circular disk with center hole for several mesh patterns. Temperatures at both the inner and outer diameter are enforced. With only a single CQDRNG8 element, the predicted temperatures were within 12.0% of theory. Refinements reduced the error to as little as .25% and a maximum of 10.0%. The same problem was solved using another NASTRAN element, the CTRIARG. In all but the coarsest model, the CQDRNG8 predicts temperatures with a moderate increase in accuracy. For the coarse model, the results are the same. The fifth test problem is very similar to the fourth. A rod oriented axially has temperatures enforced at its ends. Calculated temperatures agree exactly with theory. The final problem provides a somewhat more severe test of the heat transfer capability. A hollow sphere has enforced temperatures at the inner diameter and convection at the outer diameter. Calculated results deviate from theory by less than 2.0%.

The first three sample problems clearly demonstrate the accuracy and
efficiency of the CQDRNGS element for elastic and thermoelastic problems while the last three problems demonstrate the same qualities for heat transfer problems. Convergence is clearly shown. In all the sample problems, the grid mesh is relatively coarse. In the most severe tests (the plate bending and the hollow sphere bending and heat transfer), only two elements are present through the direction of largest gradient. The dissimilar nature of the problems illustrate the versatility of this element.

Some limitations do exist for this element. First the matrix generation is not as efficient as it could be. This is because of the technique available in NASTRAN at the time of the element's development. Matrix generation time could probably be reduced to less than one-fourth of the current value through use of a new technique available soon. The old technique forces repeated calculation of many quantities. The new technique eliminates this inefficiency. The matrix generation capability should be rewritten to incorporate this new technique when available.

The second limitation involves stresses. This element suffers from the same deficiencies as all displacement method finite elements. Interelement compatibility of stresses is not ensured. This leads to discontinuities of stresses at common grid points in adjacent elements. Realistic stress values at grid points are obtained by averaging the stresses in all common elements. The results presented in this paper were obtained in this manner. Some variation in individual grid point stresses existed but in all cases when these stresses were averaged, the results were quite close to theoretical stresses.
There are two aspects of this element which deserve further investigation. First a more complete convergence study should be undertaken and the convergence of this element compared to other existing axisymmetric finite elements. Secondly the effect of aspect ratio should be investigated. All of the sample problems were modeled with quite regular elements. Tests should include variations in length to width ratios, significant variations in included angles, and irregular spacing of midside grids. If the element is found to be highly effected by these variations, techniques for eliminating this dependence should be developed.
VIII. REFERENCES


IX. APPENDIX A

This section contains a listing of the subroutines needed for the new element capability, a functional description of the external subroutines used, and instructions for implementing the new subroutines. The functions of the various subroutines are as follows:

- **KDUM3**: calculates stiffness or conductance matrix partitions
- **SMINIT**: initializes Gaussian integration variables for subroutine KDUM3
- **SHAPE8**: calculates shape functions and derivatives at the desired Gauss point for subroutine KDUM3
- **PDUM3**: sets up connection arrays for plotting
- **DUM3**: calculates equivalent nodal forces for thermal loading
- **SSGINT**: performs same function as SMINIT for routine DUM3
- **SSHAP8**: performs same function as SHAPE8 for routine DUM3
- **SDUM31**: performs phase 1 stress calculations
- **SDUM32**: performs phase 2 stress calculations
- **LTMP**: calculates integration point temperature for temperature loading after linearizing the temperature distribution over the element; called by SDUM32
- **LINTMP**: performs same function as LTMP for DUM3
- **MSHAP8**: performs same function as SHAPE8 for SDUM32
- **ODUM3**: formats and outputs stresses

Notice that all of the subroutines perform operations on the element level only—calculating the appropriate element dependent matrices.
These element matrices are calculated and added to the overall structural matrices. All structural matrix operations are performed using the existing NASTRAN techniques. The reader is referred to the manual by MacNeal for detailed descriptions of NASTRAN's many numerical techniques.

To incorporate the new element subroutines, the subroutines must be compiled and link-edited into the appropriate parts of NASTRAN. The specific details for accomplishing these tasks are computer and installation dependent. NASTRAN on IBM systems consists of 15 separate programs called LINK's. One link is a master link which controls the other links in such a way that NASTRAN appears to be a single program. The new subroutines must be link-edited into the specific links. The pertinent links and subroutines to be added to the links are:

<table>
<thead>
<tr>
<th>LINK NAMES</th>
<th>SUBROUTINES TO BE ADDED TO LINK</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINKNS02</td>
<td>PDUM3</td>
</tr>
<tr>
<td>LINKNS03</td>
<td>KDUM3, SMINIT, SHAPE8</td>
</tr>
<tr>
<td>LINKNS05</td>
<td>DUM3, SSGINIT, SSHAFE8, LINTMP</td>
</tr>
<tr>
<td>LINKNS13</td>
<td>SDUM31, SDUM32, SD1INT, MSHAPE8, LTMP</td>
</tr>
<tr>
<td>LINKNS14</td>
<td>ODUM3</td>
</tr>
</tbody>
</table>

The link-edit operations are performed using installation dependent JCL procedures and linkage editor control cards. Both of these are supplied by The MacNeal-Schwendler Company with each version of MSC/NASTRAN.
### EXTERNAL SUBROUTINES

<table>
<thead>
<tr>
<th>NAME</th>
<th>SOURCE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESAGE</td>
<td>NASTRAN utility</td>
<td>Prints NASTRAN error messages</td>
</tr>
<tr>
<td>MAT</td>
<td>NASTRAN utility</td>
<td>Obtains elastic material properties</td>
</tr>
<tr>
<td>HMAT1</td>
<td>NASTRAN utility</td>
<td>Obtains thermal material properties</td>
</tr>
<tr>
<td>GMMATD</td>
<td>NASTRAN utility</td>
<td>General matrix multiplication</td>
</tr>
<tr>
<td>INVERD</td>
<td>NASTRAN utility</td>
<td>Incore matrix inversion</td>
</tr>
<tr>
<td>TRANSD</td>
<td>NASTRAN utility</td>
<td>Obtains coordinate transformation matrices</td>
</tr>
<tr>
<td>SMA1B</td>
<td>NASTRAN utility</td>
<td>Matrix insertion routine</td>
</tr>
<tr>
<td>FREAD</td>
<td>NASTRAN utility</td>
<td>Read NASTRAN table</td>
</tr>
<tr>
<td>SSGETD</td>
<td>NASTRAN utility</td>
<td>Obtains element temperature data</td>
</tr>
<tr>
<td>DABS, ABS</td>
<td>FORTRAN library</td>
<td>Absolute value</td>
</tr>
<tr>
<td>BASGLB</td>
<td>NASTRAN utility</td>
<td>Converts vector from basic to global coordinates</td>
</tr>
<tr>
<td>SDR2VR</td>
<td>NASTRAN utility</td>
<td>Obtains displacements</td>
</tr>
<tr>
<td>PRNSTR</td>
<td>NASTRAN utility</td>
<td>Computes 3-D principal stresses</td>
</tr>
<tr>
<td>SQRT</td>
<td>FORTRAN library</td>
<td>Computes square root</td>
</tr>
</tbody>
</table>
SUBROUTINE KTUM3
C
****
C THIS ROUTINE CALCULATES THE EIGHT 6X6 MATRICES K(NPVT,J) FOR
C A QDRNG8 ELEMENT HAVING CONNECTIONS NPVT AND J=1 THRU NPVT.
C REQUIRES ADUM3,8,1,36, CDRNG8 DATA CARD.
C
C E C P T FOR THE C D U M 8
C TYPE TABLE CARD
C
C E C P T ( 1 ) ELEMENT IC
C E C P T ( 2 ) SCALAR INDEX NUMBER FOR GRID A
C E C P T ( 3 ) SCALAR INDEX NUMBER FOR GRID B
C E C P T ( 4 ) SCALAR INDEX NUMBER FOR GRID C
C E C P T ( 5 ) SCALAR INDEX NUMBER FOR GRID D
C E C P T ( 6 ) SCALAR INDEX NUMBER FOR GRID E
C E C P T ( 7 ) SCALAR INDEX NUMBER FOR GRID F
C E C P T ( 8 ) SCALAR INDEX NUMBER FOR GRID G
C E C P T ( 9 ) SCALAR INDEX NUMBER FOR GRID H
C E C P T (10) MATERIAL ID
C E C P T (11) NO OF INTEGRATION GAUSS POINTS
C E C P T (12) DISP COORD SYS ID FOR GRID A
C E C P T (13) X COORD OF GRID A (BASIC)
C E C P T (14) Y COORD OF GRID A (BASIC)
C E C P T (15) Z COORD OF GRID A (BASIC)
C E C P T (16) DISP COORD SYS ID FOR GRID B
C E C P T (17) X COORD OF GRID B (BASIC)
C E C P T (18) Y COORD OF GRID B (BASIC)
C E C P T (19) Z COORD OF GRID B (BASIC)
C E C P T (20) DISP COORD SYS ID FOR GRID C
C E C P T (21) X COORD OF GRID C (BASIC)
C E C P T (22) Y COORD OF GRID C (BASIC)
C E C P T (23) Z COORD OF GRID C (BASIC)
C E C P T (24) DISP COORD SYS ID FOR GRID D
C E C P T (25) X COORD OF GRID D (BASIC)
C E C P T (26) Y COORD OF GRID D (BASIC)
C E C P T (27) Z COORD OF GRID D (BASIC)
C E C P T (28) DISP COORD SYS ID FOR GRID E
C E C P T (29) X COORD OF GRID E (BASIC)
C E C P T (30) Y COORD OF GRID E (BASIC)
C E C P T (31) Z COORD OF GRID E (BASIC)
C E C P T (32) DISP COORD SYS ID FOR GRID F
C E C P T (33) X COORD OF GRID F (BASIC)
C E C P T (34) Y COORD OF GRID F (BASIC)
C E C P T (35) Z COORD OF GRID F (BASIC)
C E C P T (36) DISP COORD SYS ID FOR GRID G
C E C P T (37) X COORD OF GRID G (BASIC)
C E C P T (38) Y COORD OF GRID G (BASIC)
C E C P T (39) Z COORD OF GRID G (BASIC)
C E C P T (40) DISP COORD SYS ID FOR GRID H
C E C P T (41) X COORD OF GRID H (BASIC)
C E C P T (42) Y COORD OF GRID H (BASIC)
C
DOUBLE PRECISION KE,TI,TJ,INDEX,VOL
DOUBLE PRECISION K,R,Z,N,DN,DNDRZ,JAC,PP,QQ,W,P,Q,RAD,DETERM,
1 PI,E,NU,DE,DRAK,G,DUM
REAL NUS
LOGICAL HEAT
DIMENSION IECPT(1)
COMMON /SMAIDP/ K(32),RZ(16),N(16),DN(16),DNDRZ(16),JAC(2,2),
1 PP(16),GO(16),W(16),P(16),RAD,DETERM,E,NU,DE,DRAK,G,
2 KE(26),TI(9),INDEX(2,3),VOL,DUM(16)
COMMON /CONDAD/ PI,TWOP1,RADDEG,DEGRAD,FORPI2
COMMON /MATIN/ MATID,MATFLG,ELTFMP,STRESS,SINTH,CQSTH
COMMON /MATOUT/ ES,GS,NUS,RHO,ALPHA,TSUB0,GSUBF,_SIG,TIGC,STOS
COMMON /HMOUT/ RK
COMMON /SMAIE/ IECPT(100)
COMMON /SMA110/ SKF(10),IFKGG,SKP1,IF4GG,SKP3(22)
COMMON /SMA1CL/ IOPT,K4GGSW,NPVT,SKP2(19),NOGO
COMMON /SMA1HT/ HEAT
EQUIVALENCE ( IECPT(1),IECPT(1) )
C CHECK FOR NPVT GRID
DO 5 I=1,N
IF(NPVT.NE.IECPT(I+1)) GO TO 5
NONPVT = I
GO TO 8
5 CONTINUE
C ERROR ** NO PIVOT GRID FOUND
CALL MESSADE(-30,34,IECPT(I))
NCGO=1
S CONTINUE
C FILL MATIN COMMON LOCK
MATID = IECPT(10)
MATFLG = 1
ELTFMP = IECPT(44)
C CHECK FOR HEAT FORMULATION
IF(HEAT)GO TO 9
C I RETRIEVE STRUCTURAL MATERIAL PROPERTIES
CALL MATIIECPT(1) )
C STORE PROPERTIES IN DOUBLE PRECISION LOCATIONS
E = ES
NU = NUS
C CHECK FOR ILLEGAL PROPERTIES
IF( (E.EQ.0.0D0).OR.(NU.EQ.0.0D0)) GO TO 21
C CHECK FOR POSSIBLE DIVIDE BY ZERO
IF( (NU.GE.0.5D0) ) GO TO 30
DE = NU / (1.0D0 - NU )
C = (1.0D0 - 2.0D0 * NU )/(2.0D0 * (1.0D0 - NU ) )
DEAR= E * (1.0D0 - NU )/( (1.0D0 + NU ) * (1.0D0 - 2.0D0 * NU ) )
DEAR = DEAR * TWOPI
GO TO 25
C RETRIEVE HEAT PROPERTIES
C CALL HMAT1( IECPT(1) )
E = FK
IF( E.EQ.0.0D0 ) GO TO 30
E = E * TWOPI
GO TO 35
C NU IS GE .5
30 CALL MESAGE(-20,16,IECPT(1).)
CALL MESAGE(-30,40,IECPT(1))
NOGO = 1
GO TO 35
C E = 0. OR NU = 0.
31 CALL MESAGE(-20,126,IECPT(1))
NOGO = 1
GO TO 35
C CHECK NOGO Flag
35 IF( NOGO.EQ.1 ) GO TO 5000
C WRITE R AND Z COORDINATES INTO DOUBLE PRECISION ARRAY RZ
DO 40 I=1,8
JJ = 1 + 2*(I-1)
J = 13 + 4*(I-1)
RZ(JJ ) = ECPT(J)
RZ(JJ+1) = ECPT(J+2)
40 CONTINUE
C ZERO K MATRIX
DO 50 I=1,32
K(I) = 0. DO
50 CONTINUE
C INITIALIZE VARIABLES FOR GAUSSIAN INTEGRATION
NGP = IECPT(11)
CALL SMINIT(NGP)
C TOP OF INTEGRATION LOOP
DO 900 INT =1,NGP
C CALCULATE SHAPE FUNCTIONS AND LOCAL DERIVATIVES
CALL SHAPE( INT )
C CALCULATE JACOBIAN J = DN * RZ STORE IN DUM
CALL GMAT(DN,2,8,0,RZ,F,2,0,DUM)
C STORE DUM IN DOUBLY DIMENSIONED ARRAY JAC(2,2)
JAC(1,1) = DUM(1)
JAC(1,2) = DUM(2)
JAC(2,1) = DUM(3)
JAC(2,2) = DUM(4)
C CALCULATES INVERSE JACOBIAN
C RETURNS INVERSE, DETERMINANT, SINGULARITY
CALL INVERD( 2,JAC,2,DUM,0,DETERM,ISING,INDEX )
C CHECK FOR SINGULARITY
IF( ISING.EQ.2 ) GO TO 400
C CHECK FOR ZERO DETERMINANT
IF( DETERM.EQ.0.0D0 ) GO TO 500
C STORE INVERSE JACOBIAN IN VECTOR DUM
DUM(1) = JAC(1,1)
LUM(2) = JAC(1,2)
DUM(2) = JAC(2,1)
DUM(4) = JAC(2,2)

C CALCULATE DNUPZ = INVERSE JAC * DN
CALL GMMATD(DUM,2,2,0,DN,2,P,0,DNDPZ)

C CALCULATE RAD
RAD = 0.DO
DO 60 1=1,P
J = 1 + 2*(I-1)
RAD = RAD + N(I)*PZ(J)
60 CONTINUE

C CALCULATE VOLUME FOR THIS GAUSS POINT
VOL = RAD * W(INT) * DAPS(DETERM)
IF( RAD.NE.0.DO ) GO TO 80

C ZERO RADIUS
CALL MESAGE(-30,32,IECPT(1) )
NOGO = 1
RETURN

C SINGULAR JACOBIAN
CALL MESAGE(-30,33,IECPT(1) )
NOGO = 1
RETURN

C DETERMINANT OF JACOBIAN = 0
500 CALL MESAGE(-30,37,IECPT(1) )
NOGO = 1
GO TO 5000
500 CONTINUE

C CALCULATE EIGHT 2 BY 2 STIFFNESS PARTITIONS AND STORE EACH AS A
C VECTOR
K(1) = KELM(1,1)
K(2) = KELM(1,2)
K(3) = KELM(2,1)
K(4) = KELM(2,2)

C K(1) THRU K(4) IS FIRST 2 BY 2 PARTITION
C K(5) THRU K(8) IS SECOND 2 BY 2 PARTITION
C K(9) THRU K(12) IS THIRD 2 BY 2 PARTITION
C K(13) THRU K(16) IS FOURTH 2 BY 2 PARTITION
C K(17) THRU K(20) IS FIFTH 2 BY 2 PARTITION
C K(21) THRU K(24) IS SIXTH 2 BY 2 PARTITION
C K(25) THRU K(28) IS SEVENTH 2 BY 2 PARTITION
C K(29) THRU K(32) IS EIGHTH 2 BY 2 PARTITION

IZ = 2 + NONPVT
IF(HEAT) GO TO 700
DO 600 JR = 1,8
JZ=I + JF
JJ = 1 + 4*(JR-1)
K(JJ) = VOL*(DNDPZ(NCNPVT)*DNDPZ(JR)+DF*N(JR)/RAD)
1 + (N(NONPVT)/RAD)*DE*DNDPZ(JF) + N(JF)/RAD)
2 + (DNDPZ(IJ)*DNDPZ(JZ)*G ) ) + K(JJ)
K(JJ+1) = VOL*(DNDPZ(NONPVT)*DNDPZ(JZ)*DE + N(NONPVT)*DNDPZ(JZ)
1 *DE/RAD + DNDPZ(IJ)*DNDPZ(JF)*G ) ) + K(JJ+1)
K(JJ+2) = VOL*(DNDPZ(IJ)*DE* DNDPZ(JR) + N(JR)/RAD)
1 + DNDPZ(NONPVT)*DNDPZ(JZ)*G ) ) + K(JJ+2)
\[ K(\text{JJ}+3) = \text{VOL}*(\text{DNLRZ(IZ)}*\text{DNK}+\text{K}+\text{Z(JJ)} + \text{DNDRZ(\text{NONPVT})})*\text{DNDRZ(JR)}*G \] 

1 + K(\text{JJ}+3)

600 CONTINUE
GO TO 900
C HEAT FORMULATION
700 DO 750 JF=1,R
JZ = R + JF
K(JF ) = VCL*( \text{DNLRZ(\text{NONPVT})})*\text{DNDRZ(JR)} + \text{DNDRZ(IZ)}*\text{DNDRZ(JZ)} ) 
1 + K(JR)
750 CONTINUE
900 CONTINUE
C BOTTOM OF INTEGRATION LOOP
IF(HEAT) GO TO 1200
C RETRIEVE PIVOT GRID TRANSFORMATION SYSTEM
KA = 12 + 4*(\text{NONPVT} -1)
CALL TRANS(IECPT(\text{KA}),\text{TI} )
C EXPAND 2 BY 2 PARTITIONS TO 3 BY 3 AND MULTIPLY BY CONSTANT
DO 1000 J=1,E
C ZERO KE MATRIX
DO 910 I=1,3E
KE(I) = 0. DO
910 CONTINUE
KE = 12 + 4*(J-1)
JJ = 1 + 4*(J-1)
KE(1) = K(\text{JJ} ) * DBAP
KE(2) = K(\text{JJ+1} ) * DBAR
KE(7) = K(\text{JJ+2} ) * DFR
KE(9) = K(\text{JJ+3} ) * DBAP
C CHECK IF TRANSFORMATION IS NEEDED
IF((IECPT(\text{KA})\text{\text{EQ.0}}),AND,(IECPT(\text{KB})\text{\text{EQ.0}})) GO TO 990
C FORM TIT * KE
CALL GMMATD(TIT,3,3,1,KE,3,3,0,RZ )
C RETRIEVE TJ
CALL TRANS(IECPT(\text{KB}),\text{TI} )
C FORM TIT * KE * TJ
CALL GMMATD(RZ,3,3,0,TJ,3,3,0,KE )
C EXPAND 3 BY 3 TO 6 BY 6
990 KE(15) = KE( 9)
KE(14) = KE( P)
KE(13) = KE( 7)
KE( 6) = KE( )
KE( 8) = KE( 5)
KE( 7) = KE( 4)
KE( 6) = 0.DO
KE( 5) = 0.DO
KE( 4) = 0.DO
C CALL INSERTION ROUTINE
CALL SMA1PC(KF,IECPT(J+1),0;IFKCG,0.DO )
1000 CONTINUE
RETURN
C HEAT FORMULATION INSERTION
1200 DO 1250 I=1,E
DE = F * K(I)
CALL SMA1B( DE, IECPT(I+1), NPVT, IFKGG, 0.00 )
1250 CONTINUE
5000 RETURN
END
SUBROUTINE SMINIT(NGP)
C    Initializes Gaussian Integration Variables
C    Arrays P and Q contain local grid coordinates
C    Arrays PP, QQ, and W contain Gauss point coordinates and
C    weighting functions
DOUBLE PRECISION KE, TI, TJ, INDEX, VOL
DOUBLE PRECISION K, Z, N, DN, NDRZ, JAC, PF, QQ, W, P, Q, RAD, DETERM,
  PI, E, NU, DE, DPAR, G, DUM
COMMON /SMA1DP/ K(32), P(16), N(8), LN(16), DNDRZ(16), JAC(2,2),
  1 PP(16), QQ(16), W(16), P(8), Q(8), RAD, DETERM, E, NU, DE, DPAR, G,
  2 KF(36), TI(9), TJ(9), INDEX(2,3), VOL, DUM(76)

  P(1) = -1.0, D0
  Q(1) = -1.0, D0
  P(2) =  0.0, D0
  Q(2) = -1.0, D0
  P(3) =  1.0, D0
  Q(3) = -1.0, D0
  P(4) =  1.0, D0
  Q(4) =  0.0, D0
  P(5) =  1.0, D0
  Q(5) =  0.0, D0
  P(6) =  0.0, D0
  Q(6) =  1.0, D0
  P(7) =  1.0, D0
  Q(7) =  0.0, D0
  P(8) =  0.0, D0
  Q(8) =  0.0, D0

C    Determine number of Gauss points to be used
GO TO (300, 200, 300, 400), NGP
C    3 by 3 Gaussian Integration

300 PP(1) = -0.774596666241483D0
  QQ(1) = PP(1)
  DUM(1) = 0.555555555555556D0
  DUM(2) = 0.888888888888889D0
  W(1) = DUM(1)*DUM(1)
  PP(2) =  E*QQ(1)
  QQ(2) = QQ(1)
  W(2) = DUM(1)*DUM(2)
  PP(3) = -PP(1)
  QQ(3) = QQ(1)
  W(3) = W(1)
  PP(4) = PP(1)
  QQ(4) =  E*QQ(1)
  W(4) = W(2)
  PP(5) =  E*QQ(1)
  QQ(5) = QQ(1)
  W(5) = DUM(2)*DUM(2)
  PP(6) = PP(3)
  QQ(6) = QQ(3)
  W(6) = W(2)
  PP(7) = PP(1)
  QQ(7) = -QQ(1)
W(7) = W(1)
PP(8) = 0.00
QQ(8) = QQ(7)
W(8) = W(2)
PP(9) = PP(3)
QQ(9) = QQ(7)
W(9) = W(1)
NGP = 9
RETURN
C 2 BY 2 GAUSSIAN INTEGRATION
200 PP(1)= -.5773502691896260
QQ(1) = PP(1)
PP(2) = PP(1)
QQ(2) = QQ(1)
PP(3) = PP(1)
QQ(3) = -QQ(1)
PP(4) = PP(2)
QQ(4) = QQ(2)
DC 250 I=1,4
W(1) = 1.00
2F0 CONTINUE
NGP = 4
RETURN
C 4 BY 4 GAUSSIAN INTEGRATION
400 DUM(1) = .6521451546825460
DUM(2) = .3478548451374540
PP(1) = .8611363115940520
QQ(1) = PP(1)
W(1) = DUM(2) * DUM(2)
PP(2) = -.23904510425588860
QQ(2) = QQ(1)
W(2) = DUM(1) * DUM(2)
PP(3) = -PP(2)
QQ(3) = QQ(1)
W(3) = W(2)
PP(4) = -PP(1)
QQ(4) = QQ(1)
W(4) = W(1)
PP(5) = PP(1)
QQ(5) = PP(2)
W(5) = W(2)
PP(6) = PP(2)
QQ(6) = QQ(5)
W(6) = DUM(1) * DUM(1)
PP(7) = PP(3)
QQ(7) = QQ(5)
W(7) = W(6)
PP(8) = PP(4)
QQ(8) = QQ(5)
W(8) = W(2)
PP(9) = PP(1)
QQ(9) = -QQ(5)
W(9) = W(2)
PP(10) = PP(2)
QQ(10) = QQ(9)
W(10) = W(6)
PP(11) = PP(3)
QQ(11) = QQ(9)
W(11) = W(6)
PP(12) = PP(4)
QQ(12) = QQ(9)
W(12) = W(2)
PP(13) = PP(1)
QQ(13) = -QQ(1)
W(13) = W(1)
PP(14) = PP(2)
QQ(14) = QQ(13)
W(14) = W(2)
PP(15) = PP(3)
QQ(15) = QQ(13)
W(15) = W(2)
PP(16) = PP(4)
QQ(16) = QQ(13)
W(16) = W(1)
NGP = 16
RETURN
END
SUBROUTINE SHAPE8CINT
C
CALCULATES N(I) AND LOCAL DERIVATIVES DN(I)/DP, DN(I)/DO
C
FOR EIGHT ISOPARAMETRIC QUADRILATERAL RING ELEMENT

DOUBLE PRECISION KE, TJ, TJ, INDEX, VOL
DOUBLE PRECISION K, KZ, N, (N, CNDRZ), JAC, PP, QQ, W, P, U, RAD, DETERM,
1 PI, E, NU, DE, DEAR, G, LUM
DOUBLE PRECISION KE(36), TJ(9), TJ(9), INDEX(7,3), VOL, DUM(76)

COMMON /SMA1DP/ K(32), KZ(16), N(8), DN(16), DNRZ(16), JAC(2,2),
1 PP(16), QQ(16), W(16), P(I), Q(I), RAD, DETERM, E, NU, DE, DEAR, G,
2 KE(36), TJ(9), TJ(9), INDEX(7,3), VOL, DUM(76)

PP AND QQ CONTAIN COORDINATES OF GAUSS INTEGRATION POINTS

INT IS THE INDEX OF CURRENT GAUSS POINT BEING CONSIDERED

ODD INDICES / CORNER NODES

DO 10 I=1,7,2
PP = PP(INT) * P(I)
QQ = QQ(INT) * Q(I)
C
CALCULATE SHAPE FUNCTIONS N
N(I) = .25DO*(1.DO+PP)*(1.DO+QQ)*(PP+QQ-1.DO)
C
CALCULATE DERIVATIVE WRT LOCAL COORDINATE P - DN(I)/DP
DN(I) = .25DO*(P(I))*(1.DO+QQ)*(2.DO*PP+QQ)
C
CALCULATE DERIVATIVE WRT LOCAL COORDINATE Q - DN(I)/DO
DN(I+P) = .25DO*Q(I)*(1.DO+PP)*(PP+2.DO*QQ)
10 CONTINUE

EVEN INDICES / MIDSIDE NODES

DO 20 I=2,9,2
PP = PP(INT) * P(I)
QQ = QQ(INT) * Q(I)
N(I) = .5DO * (1.DO-PP(INT)*PP(INT)) * (1.DO + QQ) *Q(I)*Q(I)
1 + .5DO * (1.DO-QQ(INT)*QQ(INT)) * (1.DO + PP) *P(I) * P(I)
DN(I) = -PP(INT) * (1.DO + QQ) * Q(I) * Q(I)
1 + .5DO * (1.DO - QQ(INT)*QQ(INT)) * P(I)
DN(I+P) = -QQ(INT) * (1.DO + PP) * P(I) * P(I)
1 + .5DO * (1.DO - PP(INT)*PP(INT)) * Q(I)
20 CONTINUE
RETURN
END
SUBROUTINE PDUM3 (*,*,*,IZ,M,NZ,NGPEL,K,IELS,INOPT)
C SETS UP CONNECTION ARRAYS FOR PLOTTING
DIMENSION IZ(1),IELS(1),IGLS(8)
M = 10
IF(IOPT.EQ.0) M = 16
IF(NZ.LT.M) RETURN 3
5 CALL FREAD(IELS,IFID,1,0)
IF(IEID.EQ.0) RETURN 1
CALL FREAD(IELS,IGLS,NGPEL,C)
IF(IOPT.NE.0) GO TO 50
DO 10 I=1,7
IZ(K) = IGLS(I)
IZ(K+1) = IGLS(I+1)
K = K + 2
10 CONTINUE
IZ(K) = IGLS(8)
IZ(K+1) = IGLS(1)
K = K + 2
NZ = NZ - M
IF(NZ.LT.M) RETURN 2
GO TO 5
50 DO 60 I=1,8
IZ(K+I-1) = IGLS(I)
60 CONTINUE
IZ(K+P) = IGLS(1)
IZ(K+9) = 0
K = K + M
NZ = NZ - M
IF(NZ.LT.M) RETURN 2
GO TO 5
100 RETURN
END
SUBROUTINE DUM3(PG)
C CALCULATES THERMAL LOADS FOR THE CQDRNGP ELEMENT
DOUBLE PRECISION TWOP, RADDEG, DEGRAD, FORPT2
DOUBLE PRECISION P, Q, PP, QQ, W, N, DN
DOUBLE PRECISION FORCE
DOUBLE PRECISION INDEX, DNDKZ, VOL, JAC, RAD, DETERM, RZ, DUM(4)
DOUBLE PRECISION PI
DOUBLE PRECISION ALPHA, E, NU, TSUBO, DE, DBAR, TRAK
DOUBLE PRECISION G, D
REAL FORPAS(3), FORGLB(3)
REAL PG(3)
REAL T(4)
REAL NUS
DIMENSION IECPT(1)
COMMON /SSGETT/ LTYPF, SKPI(4), ITEMP, IDEFT, IDEFM
COMMON /MATIN/ MATID, MATFLG, ELTEMP, STRESS, SINTH, COSTH
COMMON /MATCUT/ ES, CS, NUS, RHO, ALPHAS, TSUBOS, GSUBE, SIGT, SIGC, SIGS
COMMON /TRIMEX/ ECPT(IOO)
COMMON /SSCWRK/ P(16), O(16), PP(16), QQ(16), W(16), N(16), DN(16), TBAR,
1 FORC(16), DND(16), JAC(2, 2), INDEX(2, 3), RZ(16), VOL,
2 RAD, DETERM
COMMON /CONDAD/ PI, TWOP, RADDEG, DEGRAD, FORPT2
EQUIVALENCE (ECPT(1), IECPT(1))
IF(ITEMP.LE.0) GO TO 1000
C FILL MATIN COMMON BLOCK
MATID = IECPT(10)
MATFLG = 1
ELTEMP = IECPT(44)
C I RETRIEVE STRUCTURAL MATERIAL PROPERTIES
CALL MATCIECPT(1)
C STORE PROPERTIES IN DOUBLE PRECISION LOCATIONS
E = ES
NU = NUS
ALPHA = ALPHAS
TSUBO = TSUBOS
C CHECK FOR ILLEGAL PROPERTIES
IF( E.EQ.0.0D0).OR.(NU.EQ.0.0D0) GO TO 31
C CHECK FOR POSSIBLE DIVIDE BY ZERO
IF( NU.GE.0.5D0) GO TO 30
DE = NU / (1.0D0 - NU)
G = (1.0D0 - 2.0D0 * NU)/(2.0D0 * (1.0D0 - NU))
DBAR = E * (1.0D0 - NU)/(1.0D0 + NU) * (1.0D0 - 2.0D0 * NU)
DBAR = DBAR * TWOP
GO TO 35
C NU IS GE .5
30 CALL MESAG6(-30T16, IECPT(1))
CALL MESAGE(-30, 40, IECPT(1))
NCGO = 1
GO TO 35
C E = 0. OR NU = 0.
31 CALL MESAGE(-30, 126, IECPT(1))
NCGO = 1
GO TO 35
C CHECK NOGO FLAG
35 IF(NOGO.EQ.1) GO TO 500C
C WRITE R AND Z COORDINATES INTO DOUBLE PRECISION ARRAY RZ
DC 40 I=1,F
JJ = 1 + 2*(I-1)
J = 12 + 4*(I-1)
RZ(JJ) = FCPT(I)
RZ(JJ+1) = FCPT(I+2)
40 CONTINUE
C RETRIEVE TEMPERATURE DATA
ID = IECPT(1)
CALL SGGETD(IC,T,P)
C INITIALIZE VARIABLES FOR GAUSSIAN INTEGRATION
NGP = IECPT(11)
CALL SSGINT(NGP)
C ZERO FORCE VECTOR
DO 50 I=1,16
FORCE(I) = 0.0D0
50 CONTINUE
C TOP OF INTEGRATION LOOP
DC 900 INT =1,NGP
C CALCULATE SHAPE FUNCTIONS AND LOCAL DERIVATIVES
CALL SSHAP8(INT)
C CALCULATE JACOBIAN J = DN * RZ STORE IN DUM
CALL GMMAT(DN,2,F,0,RZ,2,0,DUM)
C STORE DUM IN DOUBLY DIMENSIONED ARRAY JAC(2,2)
JAC(1,1) = DUM(1)
JAC(1,2) = DUM(2)
JAC(2,1) = DUM(3)
JAC(2,2) = DUM(4)
C CALCULATES INVERSE JACOBIAN
C RETURNS INVERSE, DETERMINANT, SINGULARITY
CALL INVERD(2,JAC,2,DUM,C,DETERM,ISING,INDEX)
C CHECK FOR SINGULARITY
IF(ISING.EQ.1) GO TO 400
C CHECK FOR ZERO DETERMINANT
IF(DETERM.EQ.0.0D0) GO TO 500
C STORE INVERSE JACOBIAN IN VECTOR DUM
DUM(1) = JAC(1,1)
DUM(2) = JAC(1,2)
DUM(3) = JAC(2,1)
DUM(4) = JAC(2,2)
C CALCULATE UNDRZ = INVERSE JAC * DN
CALL GMMAT(DUM,2,2,C,DN,2,8,C,DNDRZ)
C CALCULATE RADIUS
RAD = 0.0D0
DO 60 I=1,F
J = 1 + 2*(I-1)
RAD = RAD + N(I)*RZ(J)
60 CONTINUE
C CALCULATE VOLUME FOR THIS GAUSS POINT
VOL = RAD * W(INT) * DAPS(DETERM)
IF( RAD.NE.0.D0 ) GO TO 80

C ZERO RADIUS
CALL MESSAGE(-30,32,IECPT(1))
NOGO = 1
RETURN

C SINGULAR JACOBIAN
400 CALL MESSAGE(-30,33,IECPT(1))
NOGO = 1
IF( DETERM.NE.0.DC ) GO TO 5000

C DETERMINANT OF JACOBIAN = 0
500 CALL MESSAGE(-30,37,IECPT(1))
NOGO = 1
GO TO 5000

80 CONTINUE
CALL LINTMP( INT,T )
TAR = TAR - TSUBD
L = ( 1.DO + 2.DO * CE ) * ALPHA
J=0
DC 600 I=1,16,2
J=J+1
FORCE(I)=FORCE(I) + (DNDRZ(J) + N(J)/ RAD ) * TBAO * VOL

600 CONTINUE
J = 8
DO 700 I=2,16,2
J=J+1
FORCE(I)=FORCE(I) + (DNDRZ(J) * TAR * VOL )

700 CONTINUE
900 CONTINUE

C BOTTOM OF INTEGRATION LOOP
C CONVERT FROM BASIC TO GLOBAL COORDINATES AND ADD TO PG
K=2
J=12
DO 910 I=1,16,2
FORBPASC1)= FORCE(I)*D*DBAR
FORFAC(2) = 0.0
FORBPASC3)=FORCE(I+1)*D*DBAR
CALL BASGLB( FORFAC1, FORFAC2, FORFAC3, IECPT(J+1),IECPT(J) )
J = J +4
PG( IECPT(K) ) = FCRCGLE(1) + PG( IECPT(K) )
PG( IECPT(K)+1 ) = FORCLP(2) + PG( IECPT(K)+1 )
PG( IECPT(K)+2 ) = FORCLP(3) + PG( IECPT(K)+2 )
K=K+1

910 CONTINUE
1000 RETURN
5000 RETURN
END
SUBROUTINE SSGINT( NCP )
C INITIALIZES GAUSSIAN INTEGRATION VARIABLES
C ARRAYS P AND C CONTAIN LOCAL GPID COORDINATES
C ARRAYS PP, QQ, AND W CONTAIN GAUSS POINT COORDINATES AND
C WEIGHTING FUNCTIONS
DOUBLE PRECISION P,C,PP,QO,W,N,DUM
DOUBLE PRECISION TPAR
DOUBLE PRECISION DUM
COMMON /SSGWRK/ P(0),Q(0),PP(16),QQ(16),W(16),N(P),DN(16),TPAR,
1 DUM(61)
P(1)=-1.0D0
Q(1)=-1.0D0
P(2)= 0.0D0
Q(2)=-1.0D0
P(3)=+1.0D0
Q(3)=-1.0D0
P(4)=+1.0D0
Q(4)= 0.0D0
P(5)=+1.0D0
Q(5)=+1.0D0
P(6)= 0.0D0
Q(6)= 1.0D0
P(7)=-1.0D0
Q(7)=+1.0D0
P(8)= 0.0D0
Q(8)= 0.0D0
C DETERMINE NUMBER OF GAUSS POINTS TO BE USED
GO TO ( 300,200,300,400 ), NCP
C 3 BY 3 GAUSSIAN INTEGRATION
300 PP(1)=-.774596669241483D0
QQ(1)=PP(1)
DUM(1)=.5555555555555556D0
DUM(2)=.8888888888888889D0
W(1)=DUM(1)*DUM(1)
PP(2)=0.0D0
QQ(2)=QQ(1)
W(2)=DUM(1)*DUM(2)
PP(3)=-PP(1)
QQ(3)=QQ(1)
W(3)=W(1)
PP(4)=PP(1)
QQ(4)=0.0D0
W(4)=W(2)
PP(5)=0.0D0
QQ(5)=0.0D0
W(5)=DUM(2)*DUM(2)
PP(6)=PP(3)
QQ(6)=0.0D0
W(6)=W(7)
PP(7)=PP(1)
QQ(7)=-QQ(1)
W(7)=W(1)
PP(8) = 0.00
QQ(8) = QQ(7)
W(F) = W(2)
PP(9) = PP(3)
QQ(9) = QQ(7)
W(9) = W(1)
NCNP = 9
RETURN

C     2 BY 2 GAUSSIAN INTEGRATION
200 PP(1)=-.7735026418946260
    QQ(1)= PP(1)
    PP(2)=-PP(1)
    QQ(2)= QQ(1)
    PP(3)= PP(1)
    QQ(3)= -QQ(1)
    PP(4)= FP(2)
    QQ(4)= QQ(2)
    DC 250 I=1,4
    WI(1)= 1.00
2^0 CONTINUE
    NCNP = 4
    RETURN

C     4 BY 4 GAUSSIAN INTEGRATION
400 DUM(1) = .6521451540625460
    DUM(2) = .3478548451374540
    PP(1) = -.86113315946530
    QQ(1) = PP(1)
    W(1) = DUM(2) * DUM(2)
    PP(2) =-.3399810435848560
    QQ(2) = QQ(1)
    W(2) = DUM(1) * DUM(2)
    PP(3) =-PP(2)
    QQ(3) = QQ(1)
    W(3) = W(2)
    PP(4) =-PP(1)
    QQ(4) = QQ(1)
    W(4) = W(1)
    PP(5) = PP(1)
    QQ(5) = PP(2)
    W(5) = W(2)
    PP(6) = PP(2)
    QQ(6) = QQ(5)
    W(6) = DUM(1) * DUM(1)
    PP(7) = PP(3)
    QQ(7) = QQ(5)
    W(7) = W(6)
    PP(8) = PP(4)
    QQ(8) = QQ(5)
    W(8) = W(2)
    PP(9) = PP(1)
    QQ(9) =-QQ(5)
    W(9) = W(2)
PP(10) = PP(2)
QQ(10) = QQ(9)
W(10) = W(6)

PP(11) = PP(3)
QQ(11) = QQ(9)
W(11) = W(6)

PP(12) = PP(4)
QQ(12) = QQ(9)
W(12) = W(2)

PP(13) = PP(1)
QQ(13) = QQ(1)
W(13) = W(1)

PP(14) = PP(2)
QQ(14) = QQ(13)
W(14) = W(2)

PP(15) = PP(3)
QQ(15) = QQ(13)
W(15) = W(2)

PP(16) = PP(4)
QQ(16) = QQ(12)
W(16) = W(1)

NGP = 16
RETURN
END
SUBROUTINE SSHAPK (INT)
C    CALCULATES N(I) AND LOCAL DERIVATIVES DN(I)/DP, DN(I)/DO
C    FOR EIGHT NODED ISOPARAMETRIC QUADRILATERAL RING ELEMENT
DOUBLE PRECISION P, Q, PP, QQ, W, N, DN
DOUBLE PRECISION PO, QQ
DOUBLE PRECISION DUM
DOUBLE PRECISION TBAR
COMMON /SSCWPK/ P(8), Q(8), PP(16), QQ(16), W(16), N(8), DN(16), TBAR,
1        DUM(61)
C    PP AND QQ CONTAIN COORDINATES OF GAUSS INTEGRATION POINTS
C    INT IS THE INDEX OF CURRENT GAUSS POINT BEING CONSIDERED
C ODD INDICES / CORNER NODES
CO 10 I=1,7,2
PO = PP(INT) * F(I)
QQ = QQ(INT) * G(I)
C    CALCULATE SHAPE FUNCTION N
N(I) = .2500*(1.00+PC)*(1.00+QC)*(PC+QC-1.00)
C    CALCULATE DERIVATIVE WRT LOCAL COORDINATE P - DN(I)/DP
DN(I) = .2500*P(I)*(1.00+QC)*(2.00*PC + QQ)
C    CALCULATE DERIVATIVE WRT LOCAL COORDINATE Q - DN(I)/DO
DN(I+8) = .2500*Q(I)*(1.00+PO)*(PC+2.00*QC)
10 CONTINUE
C EVEN INDICES / MIDSIDE NODES
DO 20 I=2,8,2
PC = PP(INT) * P(I)
QC = QQ(INT) * Q(I)
N(I) = .5000 * ( 1.00-PP(INT)*PP(INT)) * ( 1.00 + QQ ) * Q(I)*Q(I)
1 + .5000 * ( 1.00-QQ(INT)*QQ(INT)) * ( 1.00 + PC ) * P(I) * P(I)
DN(I) = -PP(INT) * ( 1.00 + QC ) * Q(I) * Q(I)
1 + .5000 * ( 1.00 - PP(INT)*PP(INT)) * Q(I)
20 CONTINUE
RETURN
END
SUBROUTINE SDUM31

C PERFORMS PHASE 1 STRESS CALCULATIONS FOR CODRNG8 ELEMENT
DOUBLE PRECISION RZ,DE,G,DBAR,SIG,E,NU,RHO,ALPHA,TSUBO,GSUBE
REAL NUS
LOGICAL HEAT
DIMENSION IEST(1)
DIMENSION IECPT(1)
COMMON /SCR2EL/ HEAT
COMMON /MATIN/ MATID,MATFLG,ELTEMP,STRESS,SINTH,COSTH
COMMON /HMTOUT/ FK
COMMON /SDR2X5/ ECPT(100),EST(44),RZ(16),E,G,NU,RHO,ALPHA,TSUBO,
1 GSUBE,SIGC2),DE,DBAR
COMMON /GPTA1/ NELEN,LAS,INCP,IE(1)
COMMON /MATOUT/ ES,GS,NUS,RHOS,ALPHAS,TSUBOS,GSUBES,SIGS(3)
EQUIVALENCE ( ECPT(1),IECPT(1) ), ( EST(1),IEST(1) )
PI= 3.14159265358979D0
C FILL MATIN COMMON BLOCK
MATID = IECPT(10)
MATFLG = 1
ELTEMP = ECPT(44)
C CHECK FOR HEAT FORMULATION
IF(HEAT)GO TO 9
C I RETRIEVE STRUCTURAL MATERIAL PROPERTIES
CALL MAT(IECPT(1))
C STORE PROPERTIES IN DOUBLE PRECISION LOCATIONS
E = FS
NU= NUS
SIG(1) = SIG(1)
SIG(2) = SIG(2)
SIG(3) = SIG(2)
RHO = RHOS
ALPHA = ALPHAS
TSUBO = TSUBOS
GSUBE = GSUBES
C CHECK FOR ILLEGAL PROPERTIES
IF( (E.EQ.0.D0).OR.(NU.EQ.0.D0) ) GO TO 31
C CHECK FOR POSSIBLE DIVIDE BY ZERO
IF( NU.GE.0.5D0 ) GO TO 30
DE = NU / (1.0D0 - NU)
G = (1.0D0 - 2.0D0 * NU)/(2.0D0 *(1.0D0 - NU))
DBAR= E * (1.0D0 - NU)/( (1.0D0 + NU) *(1.0D0 - 2.0D0 * NU))
GO TO 2E
C RETRIEVE HEAT PROPERTIES
C REPLACE HMAT1 CALL WITH HMAT OR PUT HMAT1 IN ROOT SEGMENT
C 9 CALL HMAT1( IECPT(1) )
CONTINUE
E= FK
IF( E.EQ.0.D0 ) GO TO 30
GO TO 3E
C NU IS GE .5
20 CALL MESAGE(-30,16,IECPT(1))
CALL MESAGE(-30,40,IECPT(1))
NOGO = 1
GO TO 35
C E = 0. OR NU = 0.
31 CALL MESSAGE(-30,126,IECPT(1))
NOGO = 1
GO TO 35
C CHECK NOGO FLAG
35 IF(NOGO.EQ.1) GO TO 5000
C WRITE R AND Z COORDINATES INTO DOUBLE PRECISION ARRAY RZ
DO 40 I=1,8
JJ = 1 + 2*(I-1)
J = 13 + 4*(I-1)
RZ(JJ) = ECP(B(J))
RZ(JJ+1) = ECP(B(J+1))
40 CONTINUE
C COPY ECPT TO EST FOR PHASE II
DO 100 J=1,11
IEST(J) = IECPT(J)
100 CONTINUE
DO 110 J=12,40,4
IEST(J) = IECPT(J)
EST(J+1) = ECP(B(J+1))
EST(J+2) = ECP(B(J+2))
EST(J+3) = ECP(B(J+3))
110 CONTINUE
EST(44) = ECP(B(44))
C SET FORCE AND STRESS OUTPUT WORDS (E LOCATIONS OF 13 EACH)
IE( 54*INCR + 18 ) = 65
IE( 54*INCR + 19 ) = 65
5000 RETURN
END
SUBROUTINE SCUM32
C PERFORMS PHASE II STRESS CALCULATIONS FOR CODRANG8 ELEMENT
DOUBLE PRECISION SIG(20),SIGP(7)
DOUBLE PRECISION PG,PP,PQ,W,N,DN,DNDRZ,JAC,RAD,DETERM,DUM,
  UGV,DSIG,INDEX,TBAR,FR,FZ,ETH,ERZ
DOUBLE PRECISION FZ,EG,NU,HO,ALPHA,TSUBO,GSUBE,SMAP,DE
DOUBLE PRECISION QBAR
DOUBLE PRECISION V(3),TA(9)
REAL STR(6),PSTR(3),DSTR9)
REAL XG3),
INTEGER NPG(5)
INTEGER IWORD(1)
DIMENSION IECPT(1)
DIMENSION IS(1),IFF(1)
COMMON /SDR2X/ P(8),Q(17),EG(17),W(17),N(9),DN(16),TBAR, 
  DNDRZ(16),JAC(2,2),INDEX(2,3),UGV(24),
  ER,EZ,ETH,ERZ,DSIG,DUM(4)
COMMON /SDR2DE/ IDUM2(96),T(9)
COMMON /SDR2X/ WORD(39)
COMMON /SDR2X/ ECP(44),RZ(16),EG,NU,HO,ALPHA,TSUBO,GSUBE,
  SMAP(3),DF,DBAR,S(100),FC100)
EQUIVALENCE ( S(1),IS(1)),( F(1),IFF(1))
EQUIVALENCE ( IECPT(1),ECPT(1))
EQUIVALENCE ( WORD(1),IWORD(1))
C RETRIEVE DISP VECTOR AND CONVERT TO BASIC COORD SYSTFM
  DO 5 J=1,E
    CALL SDR2VR( IECPT(J+1),2,3,X,V )
    CALL TRANSD( ECP(R+4*J),TA )
    UGV(3*J-2 ) = TA(1)*V(1) + TA(2)*V(2) + TA(3)*V(3)
    UGV(2*J-1 ) = TA(4)*V(1) + TA(5)*V(2) + TA(6)*V(3)
    UGV(3*J  ) = TA(7)*V(1) + TA(8)*V(2) + TA(9)*V(3)
  5 CONTINUE
C PUT TEMPERATURES INTO VECTOR FOR LINTMP
C INITIALIZE VARIABLES FOR GAUSSIAN INTEGRATION
NGP = IECPT(11)
CALL SD1INT(NGP,NGP)
C TOP OF INTEGRATION LOOP
  DO 900 INTA =1,5
    INT = MP(INTA)
  900 CONTINUE
C CALCULATE SHAPE FUNCTIONS AND LOCAL DERIVATIVES
CALL MSAP81 INT )
C CALCULATE JACOBIAN J = DN * RZ STORE IN DUM
CALL GMMATDCDN,2,8,0,RZ,8,2,0,DUM)
C STORE DUM IN DOUBLY DIMENSIONED ARRAY JAC(2,2)
JAC(1,1) = DUM(1)
JAC(1,2) = DUM(2)
JAC(2,1) = DUM(3)
JAC(2,2) = DUM(4)
C CALCULATES INVERSE JACOBIAN
C RETURNS INVERSE, DETERMINANT, SINGULARITY
CALL INVERD( 2,JAC,2,DUM,0,DETERM,ISING,INDEX )
C CHECK FOR SINGULARITY
IF ( ISTNC.EQ.2 ) GO TO 400
C CHECK FOR ZERO DETERMINANT
IF ( DETERM.EQ.0.0D0 ) GO TO 500
C STORE INVERSE JACOBIAN IN VECTOR DUM
DUM(1) = JAC(1,1)
DUM(2) = JAC(1,2)
DUM(3) = JAC(2,1)
DUM(4) = JAC(2,2)
C CALCULATE DNDNZ = INVERSE JAC * DN
CALL GMATD(DUM,2,2,C,DN,2,8,C,DNDNZ)
C CALCULATE RADIUS
RAD = 0.0D0
DO 60  I=1,P
J = 1 + 2*(I-1)
PAD = RAD + N(I)*RZ(J)
60 CONTINUE
C CALCULATE VOLUME FOR THIS GAUSS POINT
VCL = PAD * W(INT) * DARS(DETERM)
IF ( PAD.LE.0.0D0 ) GO TO 80
C ZERO RADIUS
CALL MESSAGE(-20,32,IFCPT(1))
NCGO = 1
RETURN
C SINGULAR JACOBIAN
400 CALL MESSAGE(-20,33,IFCPT(1))
NCGO = 1
IF ( DETERM.LE.0.0D0 ) GO TO 5000
C DETERMINANT OF JACOBIAN = 0
500 CALL MESSAGE(-30,37,IFCPT(1))
NCGO = 1
GO TO 5000
50 CONTINUE
C ZERO STRAINS
ER =0.0D0
EZ =0.0D0
ETH=0.0D0
FPZ=0.0D0
C CALCULATE STRAINS FOR THIS GAUSS POINT
DO 10  I=1,8
ER = ER + DNDNZ(I)*UGV(2*I-2)
ETH=ETH + N(I)*UGV(3*I-2)/FAD
EZ =EZ + DNDNZ(I+8)*UGV(3*I)
ERZ=ERZ + DNDNZ(I+8)*UGV(3*I-2) + DNDNZ(I)*UGV(3*I)
10 CONTINUE
C RELIEVE STRAIN DUE TO THERMAL EXPANSION
IF ( IWOPD(39).EQ.(-1) ) GO TO 19
CALL LTMP(INT,T)
EUR = TPAR - TSUPF
ER = ER - ALPHA*TPAR
ETH= ETH- ALPHA*TPAR
EZ = EZ - ALPHA*TPAR
19 CONTINUE
C CALCULATE STRESSES DUE TO REMAINING STRAIN

\[ K = 4 \times INTA - 3 \]

\[ \text{SIG}(K) = (\text{ER} + (\text{ETH} + \text{EZ}) \times \text{DE}) \times \text{DBAR} \]

\[ \text{SIG}(K+1) = (\text{ETH} + (\text{ER} + \text{EZ}) \times \text{DE}) \times \text{DBAR} \]

\[ \text{SIG}(K+2) = (\text{EZ} + (\text{ER} + \text{ETH}) \times \text{DE}) \times \text{DBAR} \]

\[ \text{SIG}(K+3) = \text{ERZ} \times C \times \text{DBAR} \]

900 CONTINUE

C BOTTOM OF INTEGRATION LOOP

C EXTRAPOLATE STRESSES TO GRID POINTS BEFORE CALCULATING INVARIANTS

DO 910 JJ=5,17,4
DO 910 LL=1,4

\[ K = JJ + LL - 1 \]

\[ \text{DSIG} = \text{SIG}(K) - \text{SIG}(LL) \]

\[ \text{SIG}(KK) = \text{SIG}(LL) + \text{DSIG} / \text{PP}(NP(3)) \]

910 CONTINUE

C CALCULATE PRINCIPAL STRESSES AND MAX SHEAR

DO 1200 J=1,5

\[ K = 4 \times J - 3 \]

\[ \text{STR}(1) = \text{SIG}(K) \]

\[ \text{STR}(2) = \text{SIG}(K+1) \]

\[ \text{STR}(3) = \text{SIG}(K+2) \]

\[ \text{STR}(4) = 0.0 \]

\[ \text{STR}(5) = 0.0 \]

\[ \text{STR}(6) = \text{SIG}(K+3) \]

CALL PPNSTR( STR, PSTR, DTR )

WRITE INTO OUTPUT AREA

JS = 1 + (J-1)*13
IS( JS ) = IECPT(1)
IS(JS+1) = J
DO 1190 L=1,4
JL = L + (J-1)*4
SICS + L + 1) = SIG(JL)

1190 CONTINUE

DC 1192 L=1,3
SICS+L+5) = PSTF(L)

1192 CONTINUE

\[ \text{STR}(1) = \text{APS}(0.50 \times (\text{PSTR}(1) - \text{PSTR}(2)) \}) \]

\[ \text{STR}(2) = \text{APS}(0.50 \times (\text{PSTR}(1) - \text{PSTR}(3)) \}) \]

\[ \text{STR}(3) = \text{APS}(0.50 \times (\text{PSTR}(2) - \text{PSTR}(3)) \}) \]

\[ S(JS+9) = \text{AMAX1( STR(1),STR(2),STR(3))} \]

\[ S(JS+10) = \text{DTR}(1) \]

\[ S(JS+11) = \text{DIP}(2) \]

\[ S(JS+12) = \text{SORT}( (\text{PSTR}(1) - \text{PSTR}(2)) ** 2 + (\text{PSTR}(1) - \text{PSTR}(3)) ** 2 \]

A + (\text{PSTR}(2) - \text{PSTR}(3)) ** 2 )

1200 CONTINUE

5000 RETURN

END
SUBROUTINE LTMP(INT,T)
C LINEARIZES TEMPERATURE OVER CORNER ELEMENT
C RETURNS TEMP AT INTEGRATION POINT
DOUBLE PRECISION P,Q,PP,QQ,W,N,DN
DOUBLE PRECISION TBAR,PC,QO,TD
REAL T(1)
COMMON /SDP2X8/ P(8),Q(4),PP(17),QQ(17),W(17),N(P),DN(16),TBAR
TBAR = 0.00
DO 10 I=1,8,2
PC = PP(INT) * P(I)
QO = QQ(INT) * Q(I)
TD = T(I+1)
TBAR = TBAR + (PC + QO) * TD
10 CONTINUE
TD = T(1)
TBAR = TD + TBAR/4.00
RETURN
END
SUBROUTINE M$AP8( INT )

C   CALCULATES N(I) AND LOCAL DERIVATIVES DN(I)/DP, DN(I)/DQ
C   FOR EIGHT NODED ISOPARAMETRIC QUADRILATERAL RING ELEMENT
C
DOUBLE PRECISION F,P,Q,PP,QQ,W,N,DN,DNDRZ,JAC,INDEX,UGV,
1   ER,EZ,ETH,ERZ,DSIG,DUM
C DOUBLE PRECISION PP,QQ
C DOUBLE PRECISION TRAR
COMMON /SDE2XP/ F(P),Q(8),PP(17),QQ(17),W(17),N(8),DN(16),TRAR,
1   DNDRZ(16),JAC(2,2),INDEX(2,3),UGV(24),
2   ER,EZ,ETH,ERZ,DSIG,DUM(4)
C PP AND QQ CONTAIN COORDINATES OF GAUSS INTEGRATION POINTS
C INT IS THE INDEX OF CURRENT GAUSS POINT BEING CONSIDERED
C ODD INDICES / CORNER NODES
DO 10 I=1,7,2
PC = PP(INT) * P(I)
QQ = QQ(INT) * Q(I)
C CALCULATE SHAPE FUNCTIONS N
N(I) = .25D0*(1.0D0+PC)*(1.0D0+QQ)*(PP+QQ-1.0D0)
C CALCULATE DERIVATIVE WRT LOCAL COORDINATE P - DN(I)/DP
DN(I) = .25D0*P(I)*(1.0D0+QQ)*(2.0D0*PP + QQ)
C CALCULATE DERIVATIVE WRT LOCAL COORDINATE Q - DN(I)/DQ
DN(I+8) = .25D0*Q(I)*(1.0D0+PP)*(PP+2.0D0*QQ)
10 CONTINUE
C EVEN INDICES / MIDSIDE NODES
DO 20 I=2,8,2
PO = PP(INT) * P(I)
QO = QQ(INT) * Q(I)
N(I) = .5D0 * ( 1.0D0-PP(INT)*PP(INT) ) * ( 1.0D0 + QQ ) * Q(I)*Q(I)
1 + .5D0 * ( 1.0D0-QQ(INT)*QQ(INT) ) * ( 1.0D0 + PP ) * P(I)*P(I)
DN(I) = -PP(INT) * ( 1.0D0 + QQ ) * Q(I) * Q(I)
1 + .5D0 * ( 1.0D0 - QQ(INT)*QQ(INT) ) * P(I)
DN(I+8) = -QQ(INT) * ( 1.0D0 + PP ) * P(I) * P(I)
1 + .5D0 * ( 1.0D0 - PP(INT)*PP(INT) ) * Q(I)
20 CONTINUE
RETURN
END
SUBROUTINE SINIT(NGP,NP)
C    INITIALIZES GAUSSIAN INTEGRATION VARIABLES
C    APRAYS P AND O CONTAIN LOCAL GRID COORDINATES
C    APRAYS FP, GO, AND W CONTAIN GAUSS POINT COORDINATES AND
C    WEIGHTING FUNCTIONS
C    NP IS GAUSS POINT LOCATIONS TO BE USED FOR STRESS CALCULATIONS
C    FOR 2 PY 2 AND 4 X 4 GAUSS INTEGRATION A (0,0) GAUSS POINT IS ADD
C    CENTROID IS ALWAYS FIRST STRESS CALCULATED
DOUBLE PRECISION P,O,PP,GO,W,N,DN,DNDROI,JAC,INDEX,UCV,
  1 EP, EZ, ETH, FRZ, DSIG, DUM
DOUBLE PRECISION TEAP
INTEGER NP(5)
COMMON /SDR2X/ P(8),O(9),PP(17),GO(17),W(17),N(8),DN(16),TBAR,
   1 DNDRI(16),JAC(2,2),INDEX(2,3),UCV(24),
  2 ERZ, EZ, ETH, ERZ, DSIG, DUM(4)
P(1)=-1.0D0
Q(1)=-1.0D0
P(2)= 0.0D0
Q(2)=-1.0D0
P(3)=+1.0D0
Q(3)=-1.0D0
P(4)=+1.0D0
Q(4)= 0.0D0
P(5)=+1.0D0
Q(5)=-1.0D0
P(6)= 0.0D0
Q(6)= 1.0D0
P(7)=-1.0D0
Q(7)=+1.0D0
P(8)=-1.0D0
Q(8)= 0.0D0
C    DETERMINE NUMBER OF GAUSS POINTS TO BE USED
GO TO (300,200,300,400),NGP
C    3 PY 2 GAUSSIAN INTEGRATION
300 PP(1)=-.774596692414320D0
  00(1)=PP(1)
  DUM(1)=.5555555555555560D0
  DUM(2)=.888888888888888D0
  W(1)= DUM(1) * DUM(1)
  PP(2)= 0.0D0
  QQ(2)= QQ(1)
  W(2)= DUM(1) * DUM(2)
  PP(3)=+ PP(1)
  QQ(3)= QQ(1)
  W(3)= W(1)
  PP(4)=+ PP(1)
  QQ(4)=+ QQ(1)
  W(4)= W(2)
  PP(5)= 0.0D0
  QQ(5)= 0.0D0
  W(5)= DUM(2) * DUM(2)
  PP(6)= PP(3)
QQ(6) = 0.D0
W(6) = W(2)
PP(7) = PP(1)
QQ(7) = -QQ(1)
W(7) = W(1)
PP(8) = 0.D0
QQ(8) = QQ(7)
W(8) = W(2)
PP(9) = PP(3)
QQ(9) = QQ(7)
W(9) = W(1)
NP(1) = 5
NP(2) = 1
NP(3) = 3
NP(4) = 7
NP(5) = 9
NGP = 9
RETURN

C 2 BY 2 GAUSSIAN INTEGRATION

200 PP(1) = -.57735026918962600
QQ(1) = PP(1)
PP(2) = -PP(1)
QQ(2) = QQ(1)
PP(3) = PP(1)
QQ(3) = -QQ(1)
PP(4) = PP(2)
QQ(4) = QQ(3)
DO 250 I=1,4
W(I) = 1.D0
250 CONTINUE

NP(1) = 5
NP(2) = 1
NP(3) = 2
NP(4) = 2
NP(5) = 4
PP(5) = 0.D0
QQ(5) = 0.D0
NGP = 4
RETURN

C 4 BY 4 GAUSSIAN INTEGRATION

400 DUM(1) = .65214515486254400
DUM(2) = .34785484513745400
PP(1) = -.6113631159465300
QQ(1) = PP(1)
W(1) = DUM(2) * DUM(2)
PP(2) = -.33998104358485600
QQ(2) = QQ(1)
W(2) = DUM(1) * DUM(2)
PP(3) = -PP(2)
QQ(3) = QQ(2)
W(3) = W(2)
PP(4) = -PP(1)
Q0(4) = Q0(1)
W(4) = W(1)
PP(5) = PP(1)
QQ(5) = PP(2)
W(5) = W(2)
PP(6) = PP(2)
QQ(6) = QQ(5)
W(6) = LUM(1) * DUM(1)
PP(7) = PP(3)
QQ(7) = QQ(5)
W(7) = W(6)
PP(8) = PP(4)
QQ(8) = Q0(5)
W(8) = W(2)
PP(9) = PP(1)
QQ(9) = -QQ(5)
W(9) = W(2)
PP(10) = PP(2)
QQ(10) = QQ(9)
W(10) = W(6)
PP(11) = PP(3)
QQ(11) = Q0(9)
W(11) = W(6)
PP(12) = PP(4)
QQ(12) = Q0(9)
W(12) = W(2)
PP(13) = PP(1)
QQ(13) = -QQ(1)
W(13) = W(1)
PP(14) = PP(2)
QQ(14) = QQ(13)
W(14) = W(2)
PP(15) = PP(3)
QQ(15) = QQ(13)
W(15) = W(2)
PP(16) = PP(4)
QQ(16) = QQ(13)
W(16) = W(1)
NP(1) = 17
NP(2) = 1
NP(3) = 4
NP(4) = 12
NP(5) = 6
PP(17) = C, DC
QQ(17) = C, DC
NCP(17) = 16
RETURN
END
SUBROUTINE DDUM2( IOPT, IFIL, ITYPE, NE, NW, OUT )
C FORMAT AND OUTPUT STRESSES
REAL ROUT(2), OUT(1)
INTEGER JOUT(2)
EQUIVALENCE ( JOUT(1), FOUT(1) )
GO TO ( 100, 200, 300 ), IOPT
C
100 GO TO ( 110, 105, 105, 105, 105, 105, 105, 105 ), ITYPE
C NO OUTPUT DESIRED
105 IFIL = 0
RETURN
C REAL STRESSES - SCPT1
110 IFIL = -1
   NF = 1
   NW = 25
RETURN
C WRITE HEADER - IOPT=2
200 GO TO ( 220, 205, 205, 205, 205, 205, 205, 205 ), ITYPE
C NO OUTPUT DESIRED
205 CONTINUE
RETURN
230 WRITE(IFIL, 210)
210 FORMAT( /T4C, 58HS, TRESSRS IN CQDINO, ELEM
1E NT S/T13, 5HEID, T20, 2HLDC, T26, 4H5I1P, T41, 5HStgTH, T56, 4H5I3G, 2
T72, 5H5I3R7, T26, 4H515G1, T41, 4H515G2, T56, 4H515G3, 3
T67, 7HTAU MAX, T82, 26HRZ PLaVE DIRECTION COSINES,
4 T112, 1CHOCTAHEDRAL/)
RETURN
C WRITE DATA LINES - IOPT=3
300 DO 400 I = 1, NW, 13
   IF( (L+12) .GT. NW ) GO TO 400
   ROUT(1) = OUT(L)
   ROUT(2) = OUT(L+1)
304 GO TO ( 330, 305, 305, 305, 305, 305, 305, 305, 305, 305 ), ITYPE
C NO OUTPUT DESIRED
305 CONTINUE
RETURN
C REAL STRESSES - SCPT1
330 WRITE(IFIL, 310) JOUT, ( OUT(L+K), K = 2, 12 )
310 FORMAT(T7, 15, 1P4E15.6/T20, 1P4E15.6, 0P2F15.4, 1PE15.6)
400 CONTINUE
RETURN
END
X. APPENDIX B

This section contains new NASTRAN Bulk Data card descriptions for the CQDRNG8 element. These cards are used in conjunction with existing NASTRAN Bulk Data cards to model structures with CQDRNG8 elements.

NASTRAN Bulk Data cards consist of ten fields of eight columns each. The first field is a card identifier. Field 10 is used in conjunction with field 1 for continuations. If the same unique alphanumeric string appears in both fields 10 and 1 of separate cards, the cards are logically connected. The continuation string must start with +. Fields 2 through 9 are free format in the sense that data can be placed anywhere within the respective fields. The numbers above the card format box are the field numbers.
Bulk Data Deck

Input Data Card  **ADUM3**  CQDRNG8 Dummy Element attributes

Description: Defines attributes of dummy element 3 (CQDRNG8)

<table>
<thead>
<tr>
<th></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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADUM3</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CQDRNG8</td>
</tr>
</tbody>
</table>

Remarks:

1. A single ADUM3 card with entries as shown must be included in Bulk Data if the CQDRNG8 element is to be referenced.
Bulk Data Deck

Input Data Card  CQDRNG8  Connections for CQDRNG8 element

**Description:** Defines a linear strain axisymmetric ring element (QDRNG8) with midside grid points. This element is quadrilateral in shape and is an isoparametric formulation.

**Format and Example:**

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<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>
<td>10</td>
</tr>
<tr>
<td>CQDRNG8</td>
<td>EID</td>
<td>PID</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>G6</td>
<td>+bc</td>
</tr>
<tr>
<td>CQDRNG8</td>
<td>369</td>
<td>32</td>
<td>111</td>
<td>230</td>
<td>141</td>
<td>142</td>
<td>143</td>
<td>220</td>
<td>+ABC</td>
</tr>
</tbody>
</table>

| +bc | G7 | G8 |
| +ABC | 221 | 222 |

**Field**  
**Contents**

**EID**  
Element identification number (Unique integer > 0)

**PID**  
Identification number of a PQDRNG8 property card (Integer > 0 or blank, default is EID)

**Gi**  
Grid point identification numbers of connection points (Integer > 0, all unique)

Remarks:

1. Element identification numbers must be unique with respect to all other EID's.
2. Connection points must be listed consecutively beginning at a corner and proceeding around the perimeter in either direction.
3. The grid points must lie in the x-z plane of the basic coordinate system, with x=r≥0.
4. This element is installed as a DUMMY element and requires one ADUM3 Bulk Data card.
Bulk Data Deck

Input Data Card **PQDRNG8**

**CQDRNG8** property card

**Description:** Defines the properties of a quadratic axisymmetric ring element. Referenced by the **CQDRNG8** card.

**Format and Example:**

<table>
<thead>
<tr>
<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>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>PQDRNG8</td>
<td>PID</td>
<td>MID</td>
<td>NGP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PQDRNG8</td>
<td>32</td>
<td>10</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field** | **Contents**
---|---
PID  | Property identification number (Integer > 0 )
MID  | Material identification number (Integer > 0 )
NGP  | Order of Gaussian integration (Integer 0,2,3,4, or blank, default = 3)

**Remarks:**

1. All **PQLRNG8** cards must have unique PID's.
2. For structural problems, a MAT1 material card is referenced. For heat transfer problems, a MAT4 material card is referenced.
3. The integration order is selected as shown:

<table>
<thead>
<tr>
<th>NGP</th>
<th>Integration Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2 by 2</td>
</tr>
<tr>
<td>3</td>
<td>3 by 3</td>
</tr>
<tr>
<td>4</td>
<td>4 by 4</td>
</tr>
</tbody>
</table>

The default value of 3 is recommended.