

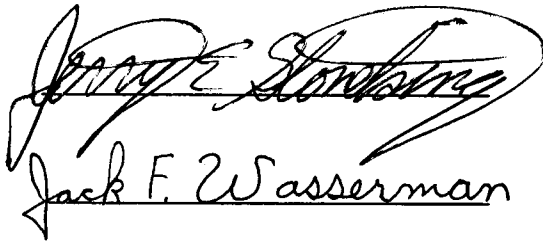
To the Graduate Council:

I am submitting herewith a thesis written by Jill P. Bradley entitled "Analysis of Pavements Using the Finite Element Method." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.



Osama Soliman, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absense, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Jill P. Bradley

Date November 24, 1986

**ANALYSIS OF PAVEMENTS USING THE
FINITE ELEMENT METHOD**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Jill P. Bradley

December 1986

ACKNOWLEDGMENTS

The author would like to acknowledge the extremely essential and professional help of her major professor, Dr. Osama Soliman, without whose guidance and patience this study might never have been completed. Special thanks are reserved for Group WX-11 at Los Alamos National Laboratory, especially Paul Smith, Mike Garcia, and Mary Lattin, whose helpful comments, suggestions, and instructions contributed greatly to this study. The author appreciates enormously the support of her parents, John and Emily Bradley, who believe in the importance of education and accomplishing goals.

The code developed in this study was run on a DEC-10 at the University of Tennessee and a VAX 11/780 at Los Alamos National Laboratory. The author appreciates the resources made available in the course of research for this study and in writing the final presentation of the results of this study.

ABSTRACT

The capability to analyze pavement structures using a computer model is an economic and time-saving resource that is important to have available in the design of road systems. The finite element method of analysis lends itself to this application very easily. A finite element analysis code which was modified to incorporate the material models describing materials used in the manufacture of pavement structures is presented. The code is incremental, iterative, and nonlinear in nature. Comparison of results from the code developed in this study and sample problems chosen from open literature produced excellent agreement. It was concluded from this study that the modified code works well in analyzing pavement structures and would be a useful tool in the design of transportation systems.

TABLE OF CONTENTS

SECTION	PAGE
1. INTRODUCTION	1
1.1 Background	1
1.2 Objective of Research	2
2. MATERIAL MODELS	4
2.1 Introduction	4
2.2 Asphaltic Concrete	4
2.3 Base Materials	5
2.4 Subgrade Materials	6
3. CODE MODIFICATIONS	11
3.1 Introduction	11
3.2 Incremental Analysis	12
3.3 Iterations	12
3.4 Inclusion of Pavement Material Models	16
3.5 Axisymmetric Element With Node on Axis of Revolution	16
3.6 Axisymmetric Element With Side Parallel to Axis of Revolution	19
3.7 Failure Criterion Based on Mohr-Coulomb Theory	21
4. CODE VERIFICATION	25
4.1 Introduction	25
4.2 Linear Analysis	25
4.2.1 ABAQUS Analysis	26
4.2.2 ADINA Analysis	29
4.2.3 Homogeneous System and Layered System Example Problems	35

SECTION	PAGE
4.3 Nonlinear Analysis	49
4.3.1 Gonzales Bypass	49
4.3.2 ILLI-PAVE	56
4.3.3 Rutledge Pike	67
5. CONCLUSIONS AND RECOMMENDATIONS	89
5.1 Conclusions	89
5.1.1 Linear Analyses	89
5.1.2 Nonlinear Analyses	89
5.1.3 Nodal Coordinate Sensitivity and Aspect Ratio	90
5.2 Recommendations	91
5.2.1 Temperature Calculations	91
5.2.2 Element Types	92
5.2.3 Refinement of Iteration Procedure	92
LIST OF REFERENCES	93
APPENDIXES	97
A. FORTRAN LISTING OF TNPAP	98
B. USER'S GUIDE FOR TNPAP.	138
C. SAMPLE INPUT FOR TNPAP	147
VITA	149

LIST OF FIGURES

FIGURE	PAGE
1. Base Material Resilient Modulus Representation	7
2. Subgrade Material Resilient Modulus Bilinear Representation . . .	9
3. Subgrade Material Resilient Modulus Quadratic Representation . .	10
4. Typical Displacement Load Curve for Nonlinear Material	13
5. Iteration Procedure Used for Convergence Check for TNPAP . . .	15
6. Iterative Stress Modification Procedure for Principal Stresses	23
7. Mohr's Circle Diagram for Mohr-Coulomb Failure Theory	24
8. ABAQUS Finite Element Mesh Density for ABAQUS-TNPAP Sample Problem	27
9. TNPAP Finite Element Mesh Density for ABAQUS-TNPAP Sample Problem	28
10. ABAQUS Sample Problem: Axial Surface Deflection	30
11. ABAQUS Sample Problem: Axial Deflection Through Model Thickness	31
12. ABAQUS Sample Problem: Axial Stress Through Model Thickness	32

FIGURE	PAGE
13. ABAQUS Sample Problem: Axial Strain Through Model Thickness	33
14. Finite Element Mesh Density for ADINA Sample Problem	34
15. ADINA Sample Problem: Axial Deflection of Model Surface	36
16. ADINA Sample Problem: Axial Deflection Through Model Thickness	37
17. ADINA Sample Problem: Axial Stress Through Model Thickness	38
18. ADINA Sample Problem: Axial Strain Through Model Thickness	39
19. Finite Element Mesh Density for Homogeneous and Layered System Problems	41
20. 110 Inch Deep Model of Homogeneous System: Surface Deflection	42
21. 110 Inch Deep Model of Homogeneous System: Axial Stress	43
22. 110 Inch Deep Model of Layered System: Surface Deflection	45
23. 110 Inch Deep Model of Layered System: Axial Stress	46
24. 300 Inch Deep Model of Layered System: Surface Deflection	47
25. 300 Inch Deep Model of Layered System: Axial Stress	48

FIGURE	PAGE
26. Finite Element Mesh Density for Gonzales Bypass	50
27. Axial Deflection Through Depth of Gonzales Bypass- Winter	52
28. Axial Deflection Through Depth of Gonzales Bypass- Summer	53
29. Axial Stress Through Depth of Gonzales Bypass- Winter	54
30. Axial Stress Through Depth of Gonzales Bypass- Summer	55
31. Finite Element Mesh Density for ILLI-PAVE Sample Problem	57
32. Surface Deflection of Cylinder- ILLI-PAVE Sample Problem Number 1	59
33. Deflection of Subgrade for ILLI-PAVE Sample Problem Number 1	60
34. Radial Strain at Asphalt Base- ILLI-PAVE Sample Problem Number 1	61
35. Axial Strain at Subgrade Surface- ILLI-PAVE Sample Problem Number 1	62
36. Surface Deflection of Cylinder- ILLI-PAVE Sample Problem Number 2	63
37. Deflection of Subgrade for ILLI-PAVE Sample Problem Number 2	64
38. Radial Strain at Asphalt Base- ILLI-PAVE Sample Problem Number 2	65

FIGURE	PAGE
39. Axial Strain at Subgrade Surface- ILLI-PAVE Sample Problem Number 2	66
40. Finite Element Mesh Density for Rutledge Pike 110 Inch Deep Model	68
41. Finite Element Mesh Density for Rutledge Pike 210 Inch Deep Model	69
42. Finite Element Mesh Density for Rutledge Pike 300 Inch Deep Model	70
43. Surface Deflection of Cylinder- Rutledge Pike Case 1	73
44. Deflection Three Inches Below Surface- Rutledge Pike Case 1	74
45. Deflection of Base Surface- Rutledge Pike Case 1	75
46. Deflection of Subgrade Surface- Rutledge Pike Case 1	76
47. Radial Strain at Asphalt Base- Rutledge Pike Case 1	77
48. Axial Strain at Subgrade Surface- Rutledge Pike Case 1	78
49. Deviator Stress at Subgrade Surface- Rutledge Pike Case 1	79
50. Surface Deflection of Cylinder- Rutledge Pike Case 2	81
51. Deflection Three Inches Below Surface- Rutledge Pike Case 2	82
52. Deflection of Base Surface- Rutledge Pike Case 2	83
53. Deflection of Subgrade Surface- Rutledge Pike Case 2	84
54. Radial Strain at Asphalt Base- Rutledge Pike Case 2	85

FIGURE	PAGE
55. Axial Strain at Subgrade Surface- Rutledge Pike Case 2	86
56. Deviator Stress at Subgrade Surface- Rutledge Pike Case 2	87

LIST OF SYMBOLS

A	constant in quadratic formulation of subgrade material
A_4	stiffness matrix component
A_5	stiffness matrix component
A_6	stiffness matrix component
B	constant in quadratic formulation of subgrade material
c	coefficient of cohesion
E_r	resilient modulus
E_{ri}	breakpoint resilient modulus
E_0	constant in quadratic formulation of subgrade material
i	local node number
k	constant in base material formulation
K	constant in base material formulation
K_1	slope of bilinear subgrade curve
K_2	slope of bilinear subgrade curve
L	constant in base material formulation
n	constant in base material formulation
N	constant in base material formulation
r	radial coordinate
σ_y	axial stress

s_1	major principal stress
s_3	minor principal stress
s_{1max}	maximum major principal stress
s_{3min}	minimum minor principal stress
S_d	deviator stress
S_{di}	breakpoint deviator stress
x	constant in base material formulation
z	axial coordinate
θ	sum of principal stresses
ϕ	angle of friction

1 INTRODUCTION

1.1 Background

The use of the finite element method for analyzing problems has become a widespread practice and is applied to a variety of problems. Finite elements are used to model anything from the flow of gases to magnetic coils. The advantage of using the finite element method has been widely accepted by industrial organizations and research and development agencies alike. The method's accuracy, when used properly, has been established. The statement reported in the July, 1986, issue of Mechanical Engineering magazine (Reference 1) and attributed to Charles Paulsen, the vice president of Algor, Inc., "...use finite element analysis in mechanical design, or fall by the wayside," emphasizes the importance of access to accurate, timely, and efficient finite element codes.

The ability to represent a pavement with a computer model and to analyze this model using finite element analysis is a desirable tool in the designing of transportation systems. References 2 and 3 discuss the use of computer models and finite element analysis in the use of designing new and improved joint systems and load transfer devices. The use of the finite element method to analyze Marshall specimens is discussed in Reference 4. Effective use of this process prevents over and under design of pavement structures and is also eco-

conomic compared to the alternative of trial and error. Over design of a pavement structure results in ineffective and costly use of resource materials. Analysis of a pavement structure before it is constructed will indicate when a design is significantly under stressed and whether less material may be used in the design and where. Under design of a pavement structure will result in failure of the structure. This situation becomes costly as the design must be replaced or reinforced, requiring extra labor and materials. The practice of simply designing a pavement structure, building the design, and stepping back to see if it will work is not cost-effective, time-effective, or safe. For these reasons, pavement structures analysis is an important step in the design process.

1.2 Objective of Research

The objective of this research was to modify an existing finite element analysis code to accurately and efficiently analyze pavement structures and to verify the code when modified. Modifications to the code included an iterative solution, incremental application of load to a model, inclusion of nonlinear material models associated with pavement structures, ability to handle an axisymmetric element with a node on the axis of revolution or a side parallel to the axis of revolution, and a failure criterion based on the Mohr-Coulomb theory. An effort was made to keep the code streamlined for efficiency in running. Accuracy of the

modified code was verified using problems from open literature. The code was then used to analyze a section of pavement from which the Civil Engineering Department at the University of Tennessee has gathered experimental data and determined material properties.

2 MATERIAL MODELS

2.1 Introduction

Much research has been done on the representation of material models associated with pavement structures and their responses to loads. References 5 through 13 present vast amounts of information regarding the mathematical modeling of materials used in the construction of transportation systems. This section describes the material models used to represent the four material types used in this research.

2.2 Asphaltic Concrete

It has been reported (References 5 and 6) that, for the short duration of time that an actual load is applied to asphaltic concrete in a pavement structure design, the asphalt material may be treated as a linear material. Asphaltic concrete does exhibit temperature dependency in its response to load. However, this phenomena may be represented by choosing an appropriate resilient modulus, or elastic modulus, for the temperature of the particular case being analyzed and using a linear relationship between stress and strain.

2.3 Base Materials

Base materials (gravel and crushed rock) have been determined to have resilient moduli which are functions of the states of stress exhibited in the materials (References 5, 6, 7, 8, 10, 11, 12, and 13). Two main relationships have been used to represent the behavior of coarse-grained materials:

$$E_r = K * (\theta)^n \quad (1)$$

and

$$E_r = k * (s_3)^N \quad (2)$$

where: E_r = resilient modulus
 θ = sum of the principal stresses
 s_3 = minor principal stress
 K and k = constants
 n and N = constants

Duncan, Monismith, and Wilson, (Reference 7) have also expressed a relationship in their studies relating resilient modulus to the mean principal stress:

$$E_r = L * \left(\frac{\theta}{3}\right)^x \quad (3)$$

where: L = constant

x = constant

However, Duncan et al. come to the conclusion from their research that, because the relationship describing resilient modulus for coarse-grained materials only has meaning when the minor principal stress is compressive and the mean principal stress will be compressive when the minor principal stress is, the use of Equation 3 yields basically the same result as Equation 2.

It is reported in Reference 6 that studies done by Allen (Reference 14) and Allen and Thompson (Reference 15) have shown that Equation 1, using the sum of principal stresses to describe the resilient modulus, is the most correct expression for use in representing the resilient modulus of coarse-grained materials. Figure 1 shows an example representation of the relationship described by Equation 1.

2.4 Subgrade Materials

Two types of models, both of which relate resilient modulus to repeated deviator stress; the difference between the major and minor principle stresses; were studied in determining a description of a material model for subgrade materials (fine-grained soils) in this research activity. References 5, 6, 7, 8, 10, 11, 12, and 13 each discuss the formulation of a bilinear representation of subgrade materials. Mathematical representation of the material model is

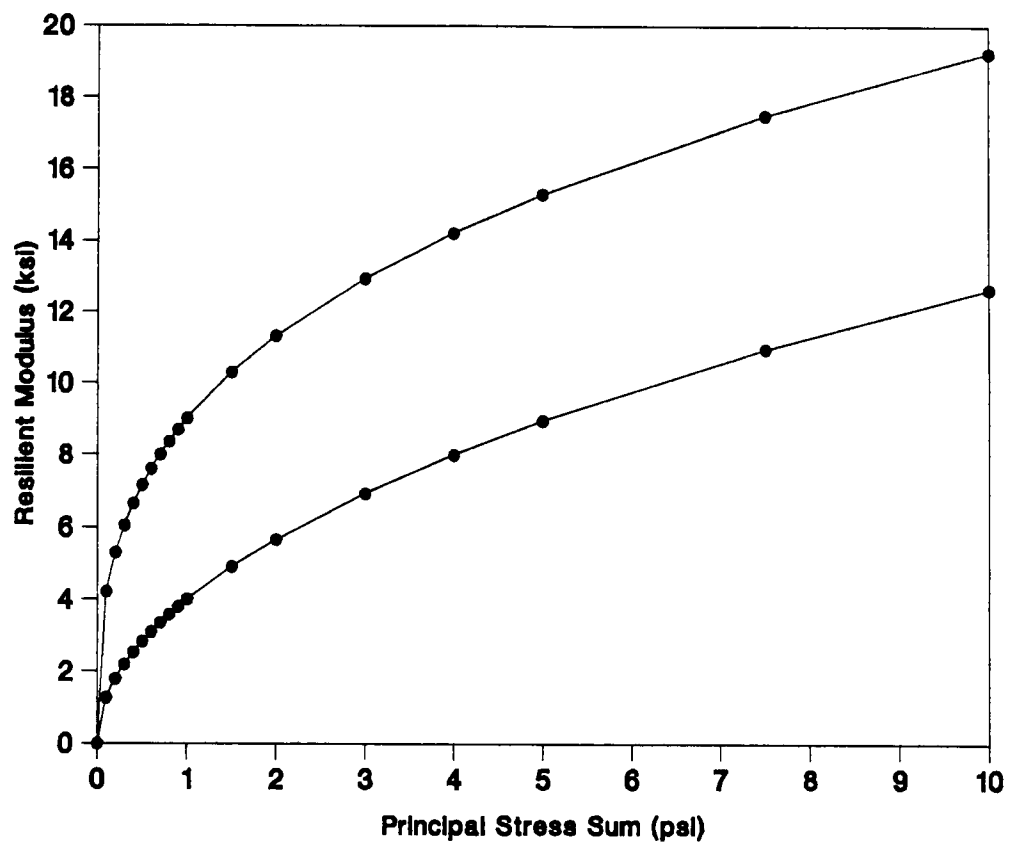


Figure 1 Base Material Resilient Modulus Representation

presented as:

$$E_r = E_{ri} + K1 * (S_d - S_{di}) \text{ for } S_d < S_{di} \quad (4)$$

and

$$E_r = E_{ri} + K2 * (S_d - S_{di}) \text{ for } S_d > S_{di} \quad (5)$$

where E_r = resilient modulus

E_{ri} = breakpoint resilient modulus between bilinear slopes

$K1$ and $K2$ = slopes of the bilinear curves

S_d = deviator stress

S_{di} = breakpoint deviator stress between bilinear slopes

Figure 2 graphically illustrates the model represented by Equations 4 and 5.

A second material model used to describe fine-grained soils was developed at the University of Tennessee from experimental data gathered by the Civil Engineering Department. This model makes use of a quadratic representation of the material. The mathematical expression used to describe this is:

$$E_r = E0 + A * (S_d)^2 + B * (S_d) \quad (6)$$

where $E0$, A , and B = constants

Figure 3 shows a typical representation of this type of material model.

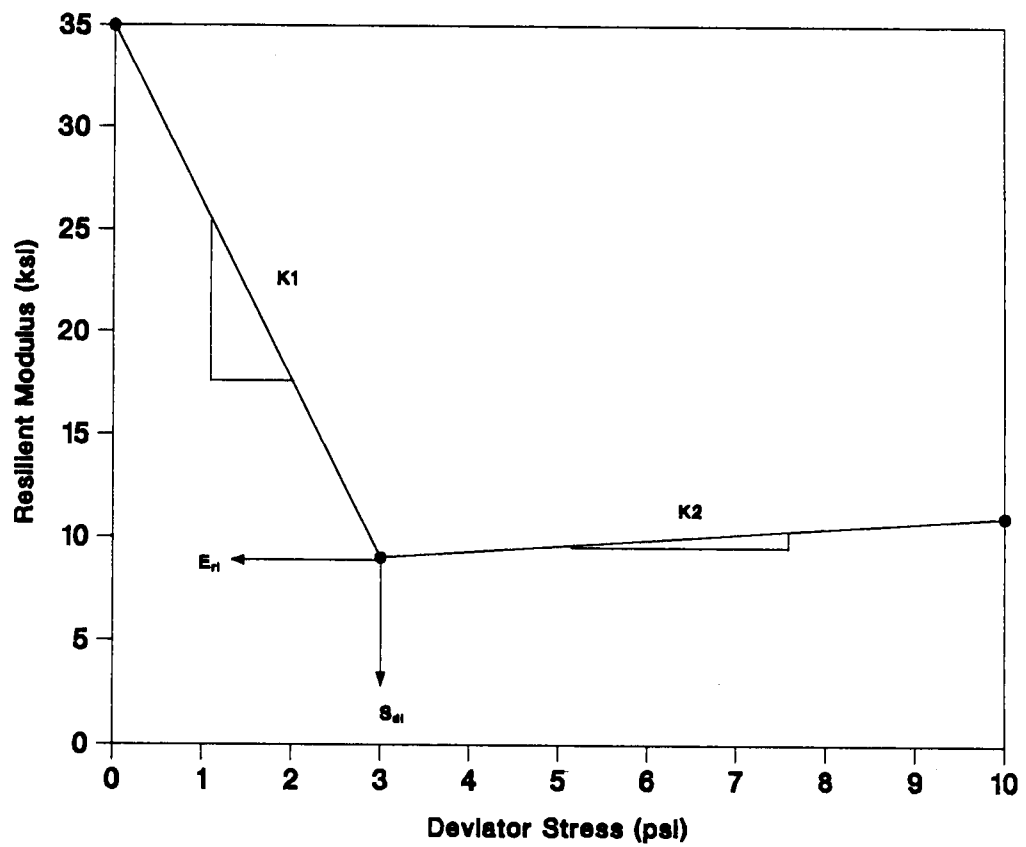


Figure 2 Subgrade Material Resilient Modulus Bilinear Representation

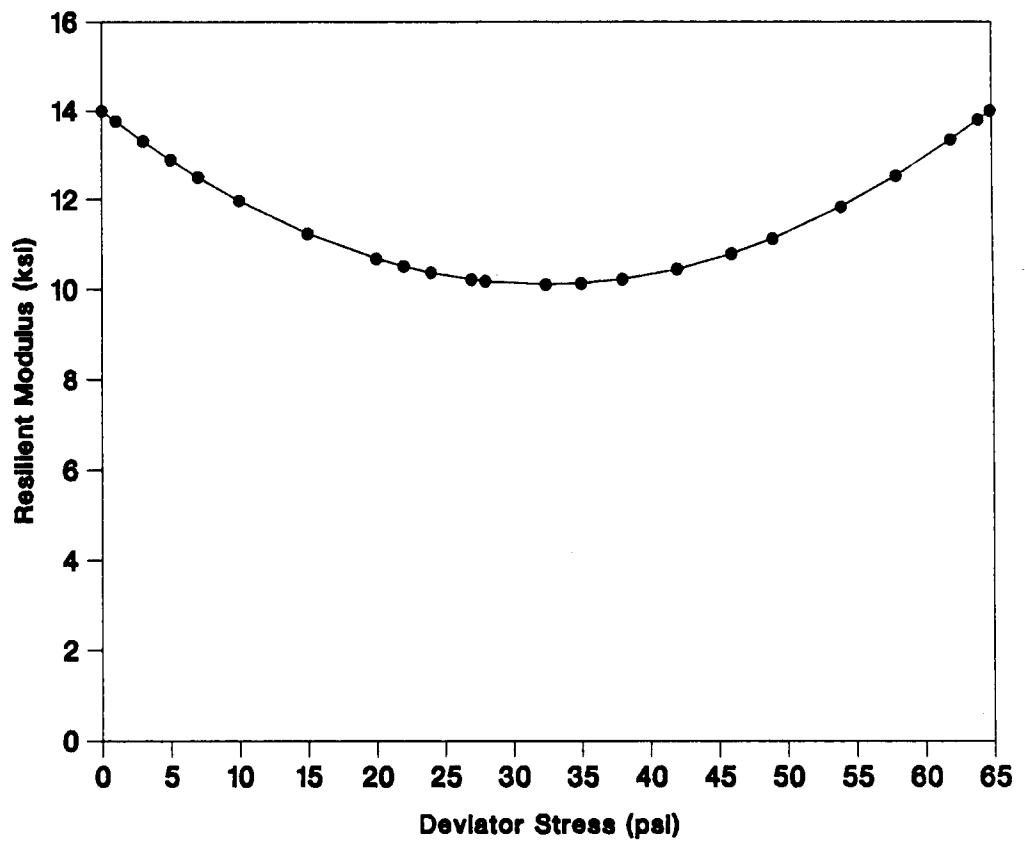


Figure 3 Subgrade Material Resilient Modulus Quadratic Representation

3 CODE MODIFICATIONS

3.1 Introduction

The original finite element analysis code that was modified for the research described in this paper is discussed in Reference 16. The code was initially written to include constitutive equations for creep and plasticity calculations. However, this analysis capability was removed from the code for the purpose of this research. The original code treated tensile stress as positive and compressive stress as negative. This characteristic was left unchanged in the modification of the code. The code was modified to include incremental analyses, iterations, material models associated with pavement structures, the capability to handle an axisymmetric element with a node on the axis of revolution or a side parallel to the axis of revolution, and a failure criterion based on the Mohr-Coulomb theory. The code presently has the capability to analyze only triangular, axisymmetric elements. However, provisions are made in the code which would make adding additional element types a reasonably easy task.

The modifications required in the code for analyses of problems involving transportation structures are described in this section. The modified finite element analysis code, TNPAP (Tennessee Nonlinear Pavement Analysis Program), used for the research presented in this paper is listed in Appendix A. A

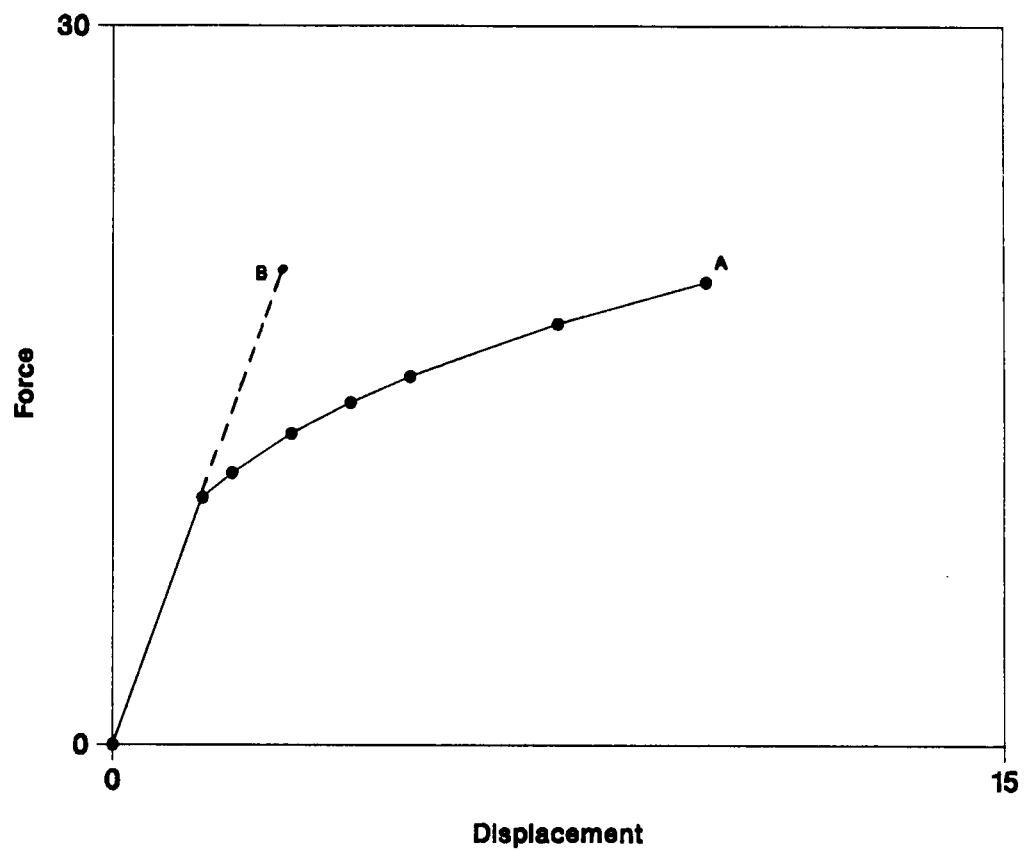
User's Manual for the code is supplied in Appendix B. Appendix C illustrates a sample input data deck for TNPAP.

3.2 Incremental Analyses

The capability to divide the loading applied to a pavement structure being analyzed into incremental steps was a desirable feature to add to this finite element analysis code because it allows for the gradual application of a load to more fully and accurately describe the nonlinearities of a material model. Reference 17 discusses the importance of incremental analyses in pavement structure design. Figure 4 shows a typical displacement versus load curve for a nonlinear material. Incremental application of the load allows for the following of the curve to point A. Applying the load all in one step follows the dotted curve to point B which supplies erroneous solutions for the problem being analyzed.

3.3 Iterations

The code was modified to include the capability to iterate on a solution within each load increment. Due to the nonlinearities of the material models involved in pavement structure analyses and the fact that the resilient moduli of the material models for granular and fine-grained soils are functions of stress, iterations on each load increment are an essential part of the analyses. The



importance of this feature is discussed in Reference 17 along with the importance of incremental analysis. The material models describing base materials (gravel and crushed rock) and subgrade materials (fine-grained soils) require that an estimate of the resilient moduli be assumed on the first increment and iteration of the analysis. Displacements, strains, and stresses are calculated using the first estimate and resilient moduli for each individual element calculated using the resulting stresses. A second iteration is performed using the moduli calculated from the first iteration. The resulting displacements are compared with the previous iteration and, if found to deviate from the convergence criteria set in the code, another iteration is performed. Figure 5 illustrates a flowchart of this iterative procedure. The convergence criteria which is currently set in the code is a ten percent deviation in surface deflections. This criteria can easily be made more or less stringent by changing the allowable deviation or including more of the analysis model in the convergence check. A limit on the number of iterations done by the code may also be set in the input deck. If the code reaches the maximum number of iterations specified without satisfying the convergence criteria, a message is printed to the output deck stating the maximum and minimum deviation calculated and the points at which these deviations occur.

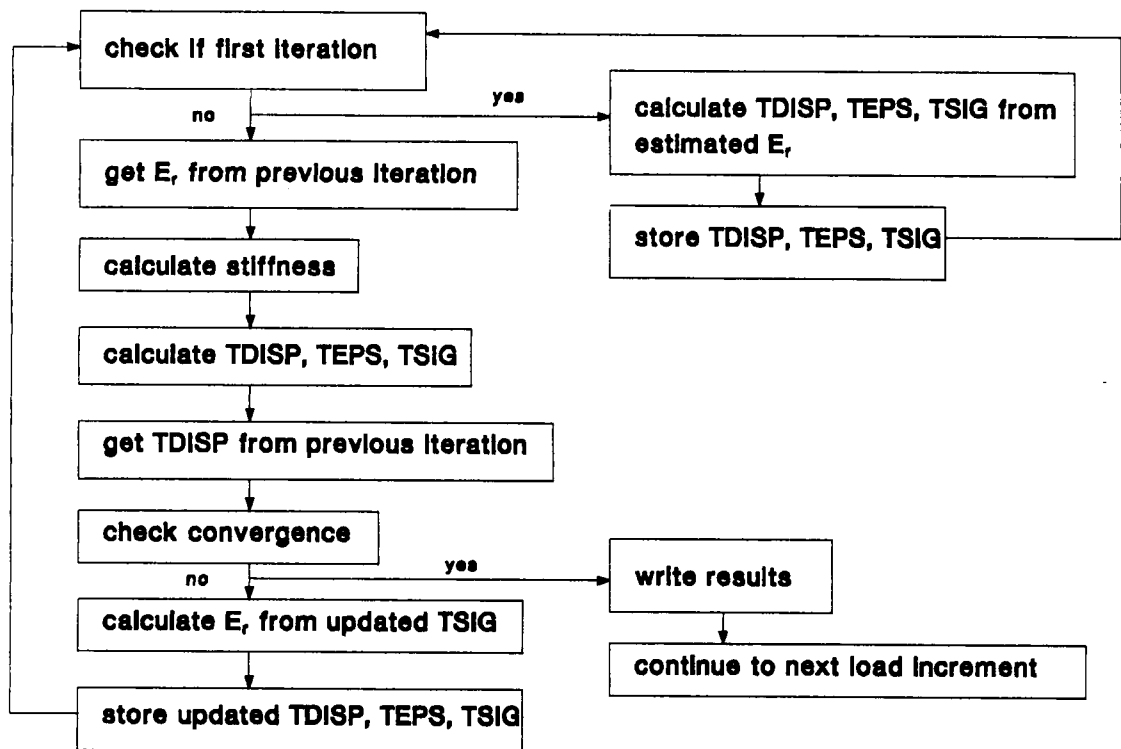


Figure 5 Iteration Procedure Used for Convergence Check in TNPAP

3.4 Inclusion of Pavement Material Models

The finite element code was also modified to include the material models associated with pavement structures. The theoretical formulations and mathematical representations of these material models are described in Section 2. Base material resilient moduli were described using Equation 1 and both the bilinear representation, described with Equations 4 and 5, and the quadratic representation, described with Equation 6, of the subgrade materials were included in the code.

3.5 Axisymmetric Element With Node on Axis of Revolution

Senol Utku, in his paper "Explicit Expressions for Triangular Torus Element Stiffness Matrix," (Reference 18) discusses the formulation of the stiffness matrix for triangular elements assuming a linearly elastic material. In his paper, Mr. Utku breaks down the stiffness matrix into rather complete components. Mr. Utku also discusses the indeterminacies of some of these components when one of the nodes of a triangular finite element is located on the axis of revolution. A formulation to remove this indeterminacy is presented by Mr. Utku. This formulation was included in the modifications to the finite element anal-

ysis code TNPAP. The original terms in the stiffness matrix that exhibit the indeterminacy are described with the following equations:

$$A4 = \frac{r_i z_{i+1} - r_{i+1} z_i}{r_i - r_{i+1}} * \ln \frac{r_i}{r_{i+1}} \quad (7)$$

$$A5 = -\frac{1}{4} * \frac{z_i - z_{i+1}}{r_i - r_{i+1}} * (z_i(3r_{i+1} - r_i) - z_{i+1}(3r_i - r_{i+1}))$$

$$+ \frac{1}{2} * \left(\frac{r_i z_{i+1} - r_{i+1} z_i}{r_i - r_{i+1}} \right)^2 * \ln \frac{r_i}{r_{i+1}} \quad (8)$$

$$A6 = \frac{1}{18} * \left(\frac{z_i - z_{i+1}}{r_i - r_{i+1}} \right)^2 * (z_{i+1}^2(11r_i^2 - 7r_i r_{i+1} + 2r_{i+1}^2)$$

$$+ 2z_i z_{i+1} * (2.5r_i^2 - 11r_i r_{i+1} + 2.5r_{i+1}^2)$$

$$+ z_i^2(11r_{i+1}^2 - 7r_i r_{i+1} + 2r_i^2))$$

$$+ \frac{1}{3} * \left(\frac{r_i z_{i+1} - r_{i+1} z_i}{r_i - r_{i+1}} \right)^3 * \ln \frac{r_i}{r_{i+1}} \quad (9)$$

where r, z = coordinates of node points

i = local node number of element

and i ranges from 1 to 3. The range is module 3; for example, when i is 3, $i+1$ is 1, and $i+2$ is 2. The indeterminacy occurs when the term r_{i+1} is equal to zero; i.e. node $i+1$ is on the axis of revolution. In this case, the terms described by Equations 7, 8, and 9 are replaced by the following:

$$A4 = -z_i * \ln \frac{r_{i+1}}{r_{i+2}} + \frac{r_{i+1}z_{i+2} - r_{i+2}z_{i+1}}{r_{i+1} - r_{i+2}} * \ln \frac{r_{i+1}}{r_{i+2}} \quad (10)$$

$$\begin{aligned} A5 = & \frac{1}{4} * ((z_{i+2} - z_{i+1}) * (2z_i + z_{i+1} + z_{i+2}) - 2z_i^2 \ln \frac{r_{i+1}}{r_{i+2}}) \\ & - \frac{1}{4} * \frac{z_{i+1} - z_{i+2}}{r_{i+1} - r_{i+2}} * (z_{i+1}(3r_{i+2} - r_{i+1}) - z_{i+2}(3r_{i+1} - r_{i+2})) \\ & + \frac{1}{2} * \left(\frac{r_{i+1}z_{i+2} - r_{i+2}z_{i+1}}{r_{i+1} - r_{i+2}} \right)^2 * \ln \frac{r_{i+1}}{r_{i+2}} \end{aligned} \quad (11)$$

$$\begin{aligned} A6 = & \frac{1}{18} * ((z_{i+2} - z_{i+1}) * (2(z_{i+1}^2 + z_{i+1}z_{i+2} + z_{i+2}^2) + 6z_i^2 + 3z_i(z_{i+1} + z_{i+2}) \\ & + 6z_i^3 * \ln \frac{r_{i+1}}{r_{i+2}})) + \frac{1}{18} * \left(\frac{z_{i+1} - z_{i+2}}{r_{i+1} - r_{i+2}} \right)^2 * (z_{i+2}^2 \\ & * (11r_{i+1}^2 - 7r_{i+1}r_{i+2} + 2r_{i+2}^2) + 2z_{i+1}z_{i+2}(2.5r_{i+1}^2 - 11r_{i+1}r_{i+2} + 2.5r_{i+2}^2) \end{aligned}$$

$$\begin{aligned}
& +z_{i+1}^2(11r_{i+2}^2 - 7r_{i+1}r_{i+2} + 2r_{i+1}^2)) \\
& + \frac{1}{3} * \left(\frac{r_{i+1}z_{i+2} - r_{i+2}z_{i+1}}{r_{i+1} - r_{i+2}} \right)^3 * \ln \frac{r_{i+1}}{r_{i+2}}
\end{aligned}
\tag{12}$$

3.6 Axisymmetric Element With Side Parallel to Axis of Revolution

Also included in Reference 18 is a discussion of indeterminacies which exist in triangular axisymmetric elements with a side parallel to the axis of revolution. Equations 7, 8, and 9 are indeterminate when r_i is equal to r_{i+1} . In this case, the terms described by Equations 7, 8, and 9 are replaced by:

$$\begin{aligned}
A4 = & z_{i+2} - z_{i+1} + \frac{r_i z_{i+1} - r_{i+1} z_i}{r_i - r_{i+1}} * \ln \frac{r_i}{r_{i+1}} \\
& + \frac{r_{i+2} z_i - r_i z_{i+2}}{r_{i+2} - r_i} * \ln \frac{r_{i+2}}{r_i}
\end{aligned}
\tag{13}$$

$$\begin{aligned}
A5 = & -\frac{1}{4} * \frac{z_i - z_{i+1}}{r_i - r_{i+1}} * (z_i(3r_{i+1} - r_i) - z_{i+1}(3r_i - r_{i+1})) \\
& + \frac{1}{2} * \left(\frac{r_i z_{i+1} - r_{i+1} z_i}{r_i - r_{i+1}} \right)^2 * \ln \frac{r_i}{r_{i+1}}
\end{aligned}$$

$$\begin{aligned}
& -\frac{1}{4} * \frac{z_{i+2} - z_i}{r_{i+2} - r_i} * (z_{i+2}(3r_i - r_{i+2}) - z_i(3r_{i+2} - r_i)) \\
& + \frac{1}{2} * \left(\frac{r_{i+2}z_i - r_iz_{i+2}}{r_{i+2} - r_i} \right)^2 * \ln \frac{r_{i+2}}{r_i}
\end{aligned} \tag{14}$$

$$A6 = \frac{1}{18} * \left(\frac{z_i - z_{i+1}}{r_i - r_{i+1}} \right)^2 * (z_{i+1}^2(11r_i^2 - 7r_iz_{i+1} + 2r_{i+1}^2)2)$$

$$+ 2z_iz_{i+1}(2.5r_i^2 - 11r_iz_{i+1} + 2.5r_{i+1}^2)$$

$$+ z_i^2(11r_{i+1}^2 - 7r_iz_{i+1} + 2r_i^2))$$

$$+ \frac{1}{3} * \left(\frac{r_iz_{i+1} - r_{i+1}z_i}{r_i - r_{i+1}} \right)^3 * \ln \frac{r_i}{r_{i+1}}$$

$$+ \frac{1}{18} * \left(\frac{z_{i+2} - z_i}{r_{i+2} - r_i} \right)^2 * (z_i^2(11r_{i+2}^2 - 7r_{i+2}r_i + 2r_i^2)$$

$$+ 2z_{i+2}z_i * (2.5r_{i+2}^2 - 11r_{i+2}r_i + 2.5r_i^2)$$

$$+ z_{i+2}^2(11r_i^2 - 7r_{i+2}r_i + 2r_{i+2}^2))$$

$$+\frac{1}{3} * \left(\frac{r_{i+2}z_i - r_i z_{i+2}}{r_{i+2} - r_i} \right)^3 * \ln \frac{r_{i+2}}{r_i} \quad (15)$$

The formulations used in removing the indeterminacies were obtained and reported by Mr. Utku by application of L'Hospital rule and by observing that the radial displacements on the axis of revolution are zero.

3.7 Failure Criterion Based on Mohr-Coulomb Theory

References 5, 6, 8, and 19 discuss a method of stress adjustment for use in modeling gravel, crushed rock, and fine-grained soils. This adjustment, based on the Mohr-Coulomb theory of failure, accounts for the lack of shear strength in base and subgrade materials. A stress adjustment is often necessary in the analysis of granular and fine-grained soils to avoid prediction of stresses in these materials which will exceed the strengths of the materials. Quite often, for example, a tensile stress will be calculated by the code for these materials which have practically nonexistent tensile strength.

The Mohr-Coulomb theory of failure states that failure will occur when

$$s_1 = s_3 * \tan^2\left(45 + \frac{\phi}{2}\right) + 2c * \tan\left(45 + \frac{\phi}{2}\right) \quad (16)$$

where $s1$ = major principal stress

$s3$ = minor principal stress

ϕ = angle of friction

c = coefficient of cohesion

At the end of each iteration, the stresses that have been calculated for each element are checked to determine if the strength of the material has been exceeded. If it has, the stresses are adjusted using the axial stresses of the element and the following equations:

$$s1_{max} = sy * \tan^2(45 + \frac{\phi}{2}) + 2c * \tan(45 + \frac{\phi}{2}) \quad (17)$$

$$s3_{min} = sy * \tan^2(45 - \frac{\phi}{2}) - 2c * \tan(45 - \frac{\phi}{2}) \quad (18)$$

where $s1_{max}$ = maximum major principal stress

$s3_{min}$ = minimum minor principal stress

sy = axial stress calculated for element

The flowchart of the procedure used to modify the stresses at the end of each iteration is shown in Figure 6. Figure 7 pictures typical locations on the Mohr circle diagram of $s1_{max}$, $s3_{min}$, c , and ϕ .

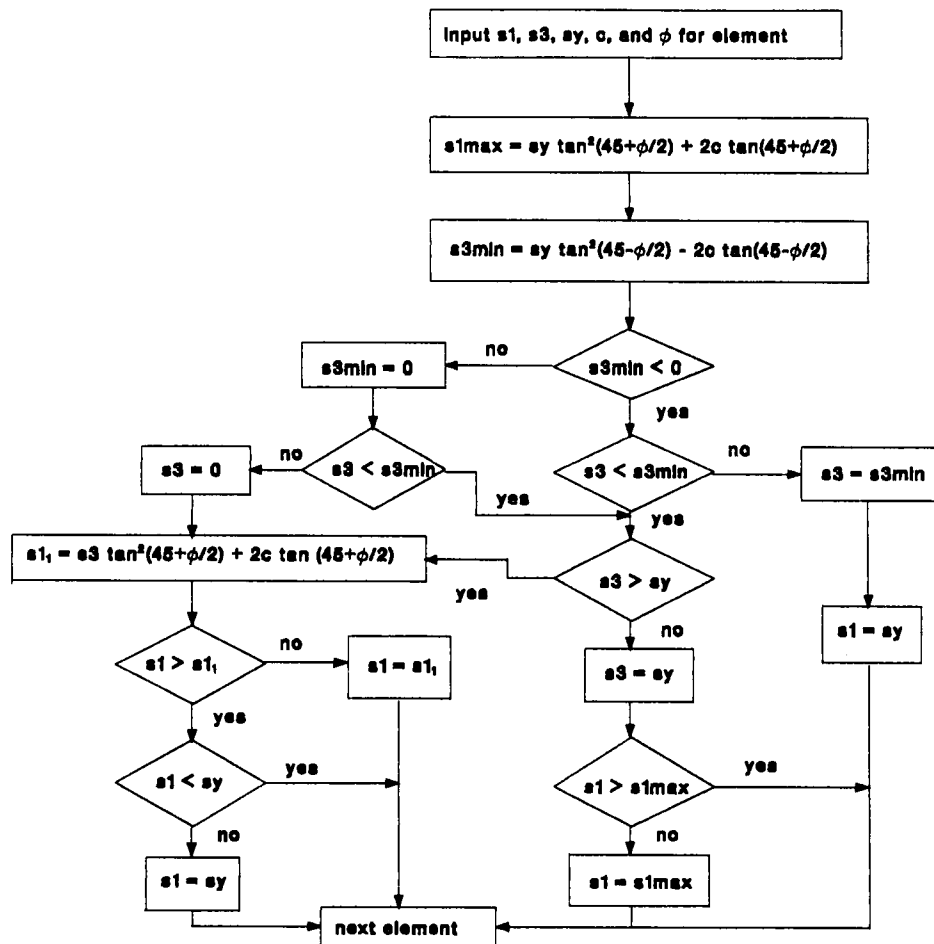


Figure 6 Iterative Stress Modification Procedure for Principal Stresses

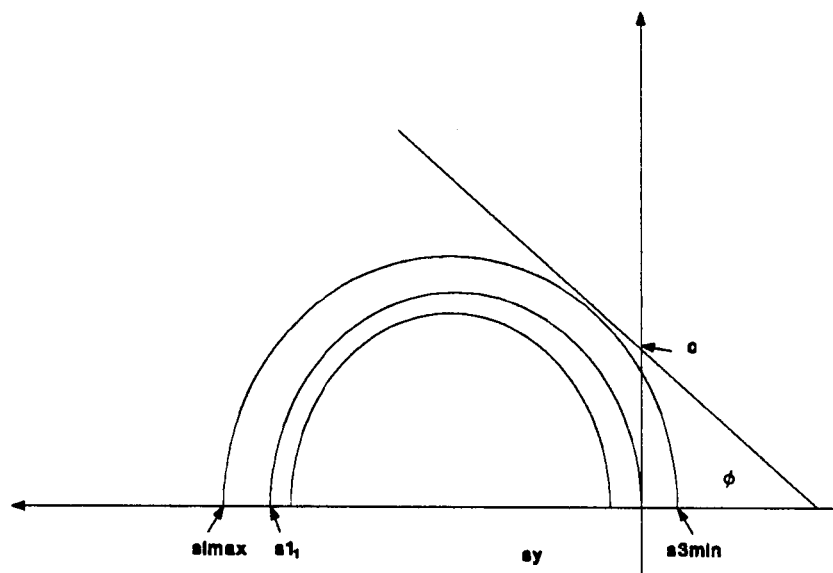


Figure 7 Mohr's Circle Diagram for Mohr-Coulomb Failure Theory

4 CODE VERIFICATION

4.1 Introduction

When all modifications were included in the analysis code, it was necessary to verify the accuracy of the code with sample problems. Problems from the open literature were selected to facilitate comparison of the code not only with other analysis codes, but also to compare with experimental data. Both linear and nonlinear problems were selected for comparison runs. This section will discuss the sample problems run with TNPAP for verification.

4.2 Linear Analyses

Four problems were selected for verification of linear analyses run by the code; a sample problem run with the ABAQUS finite element stress analysis code (Reference 20), a sample problem run with the ADINA finite element stress analysis code (Reference 21), a linear problem run with a homogeneous material, and a linear problem with a layered system of three materials. The latter two problems are described in Reference 7.

4.2.1 ABAQUS Analysis

An elementary sample problem was created for analysis with the ABAQUS finite element code for comparison with TNPAP. The number of elements in the problem was kept to the minimum amount necessary to elicit believable results and still obtain easily comparable results for a linear, layered system. The finite element mesh used in the sample problem run in ABAQUS is shown in Figure 8. A cylinder, four inches in diameter and six inches deep, was modeled using axisymmetric, four noded quadrilateral elements. A two by two integration scheme was used with CAX4 type elements. The cylinder was modeled with three layers of material. The first layer was modeled with a resilient modulus of 200,000 psi, the second layer was modeled with resilient modulus of 20,000 psi, and the third layer was modeled with a resilient modulus of 10,000 psi. Each layer had a Poisson's ratio of 0.4 and was two inches thick. The model was confined radially with the bottom surface fixed. A pressure load of 200 pounds per square inch (psi) was applied to a circular section of the top of the cylinder two inches in diameter to simulate the loading of pavement by a wheel of a vehicle.

The same cylinder was modeled using triangular elements and analyzed using TNPAP. Figure 9 illustrates the finite element model used in the TNPAP

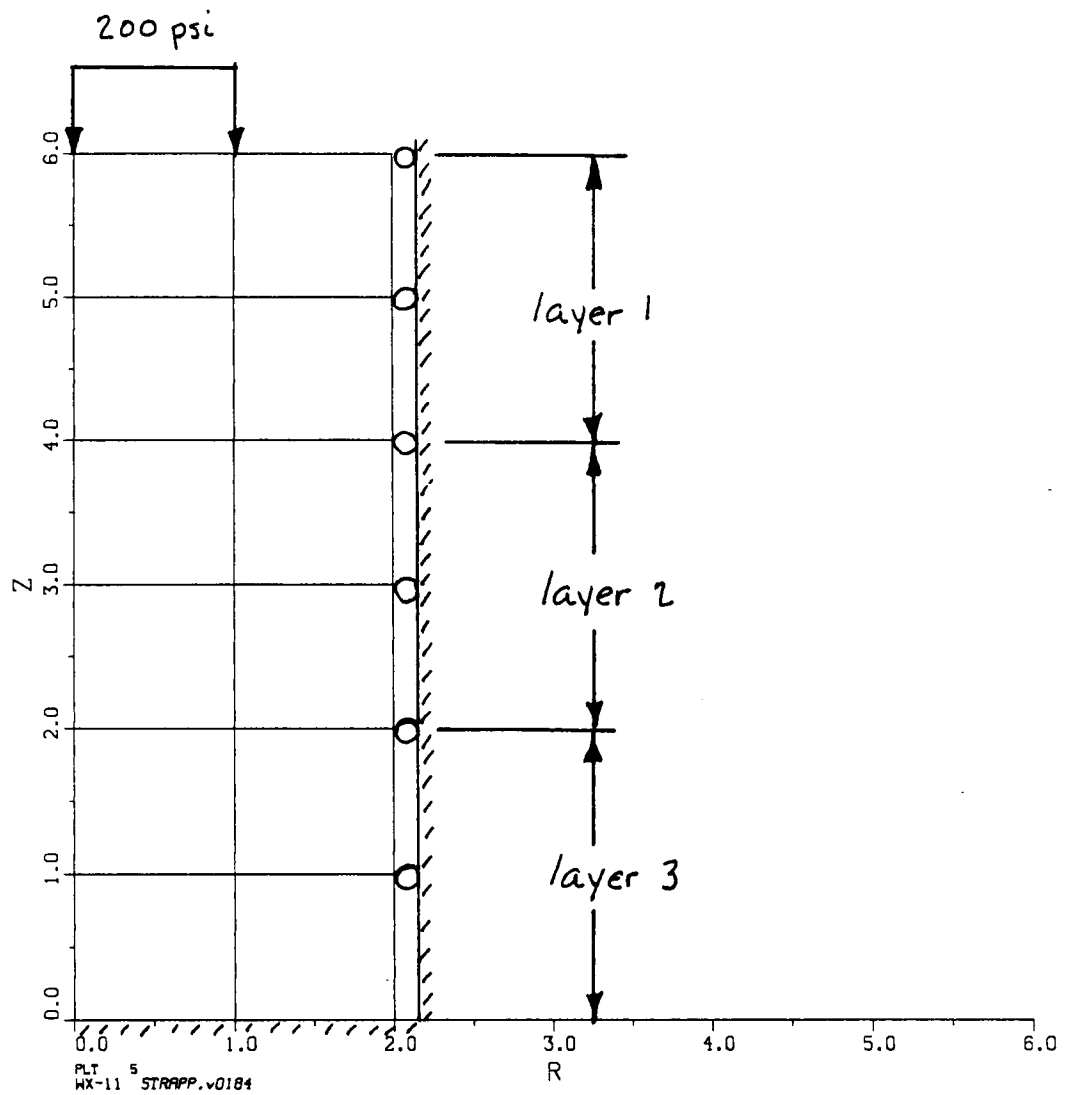


Figure 8 ABAQUS Finite Element Mesh Density for ABAQUS-TNPAP Sample Problem

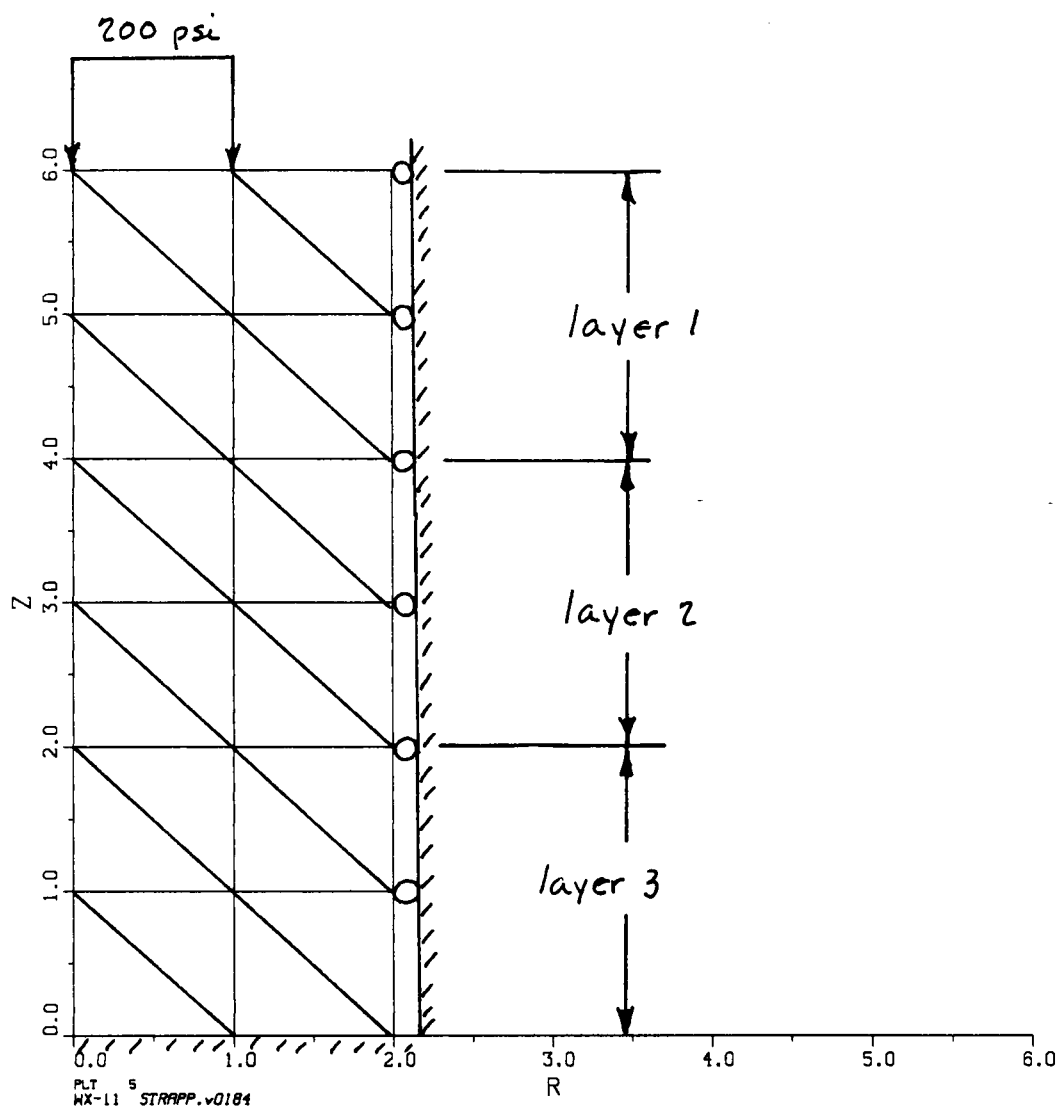


Figure 9 TNPAP Finite Element Mesh Density for ABAQUS-TNPAP Sample Problem

analysis. One iteration and one load increment were used in the analysis because of the linearity of the problem.

The results of these analyses are compared in Figures 10 through 13. Figure 10 illustrates the axial deflection of the surface of the cylinder. Figure 11 indicates the axial deflection through the depth of the cylinder at the centerline. Figures 12 and 13 picture the axial stress and strain, respectively, through the depth of the cylinder at the centerline. Excellent correlation can be seen between the two codes from the results of this analysis.

4.2.2 ADINA Analysis

A somewhat more complicated sample problem was created for analyses comparisons between the ADINA finite element analysis code and TNPAP. A cylinder 299.0 inches deep and 144.0 inches in diameter was modeled using axisymmetric finite elements. The dimensions of this cylinder closely approximate the dimensions of cylinders chosen for analyses comparison with TNPAP from open literature and which are presented later in this study. The finite element model used in this analysis is pictured in Figure 14. Four noded quadrilaterals were used in the ADINA analysis whereas triangular elements were used in the TNPAP analysis. For ease of scrutiny, the quadrilateral model is shown but not the triangular model. Each quadrilateral shown in the model in Figure 14 was

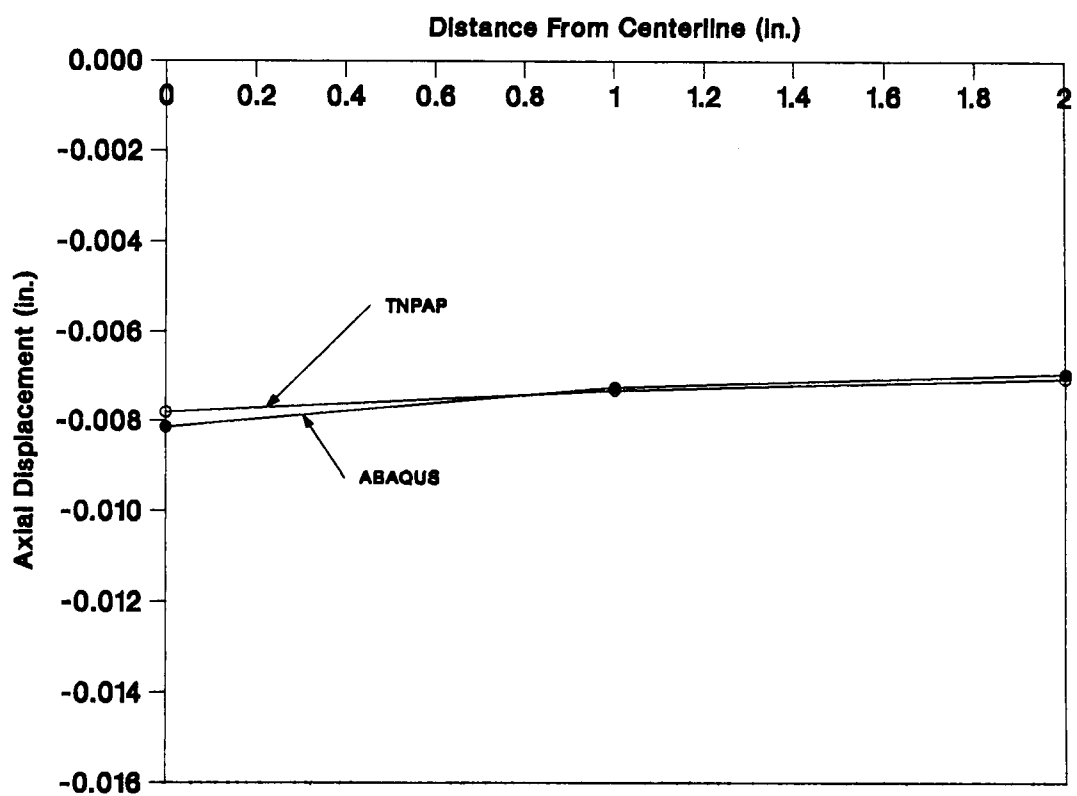


Figure 10 ABAQUS Sample Problem: Axial Surface Deflection

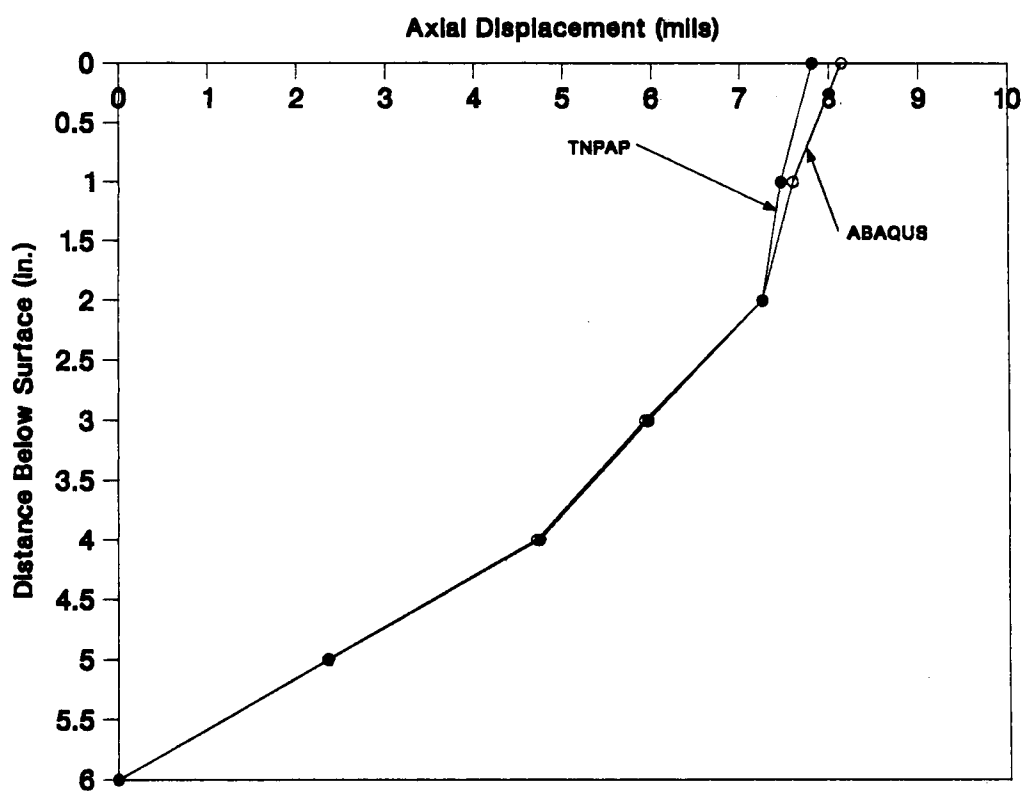


Figure 11 ABAQUS Sample Problem: Axial Deflection Through Model Thickness

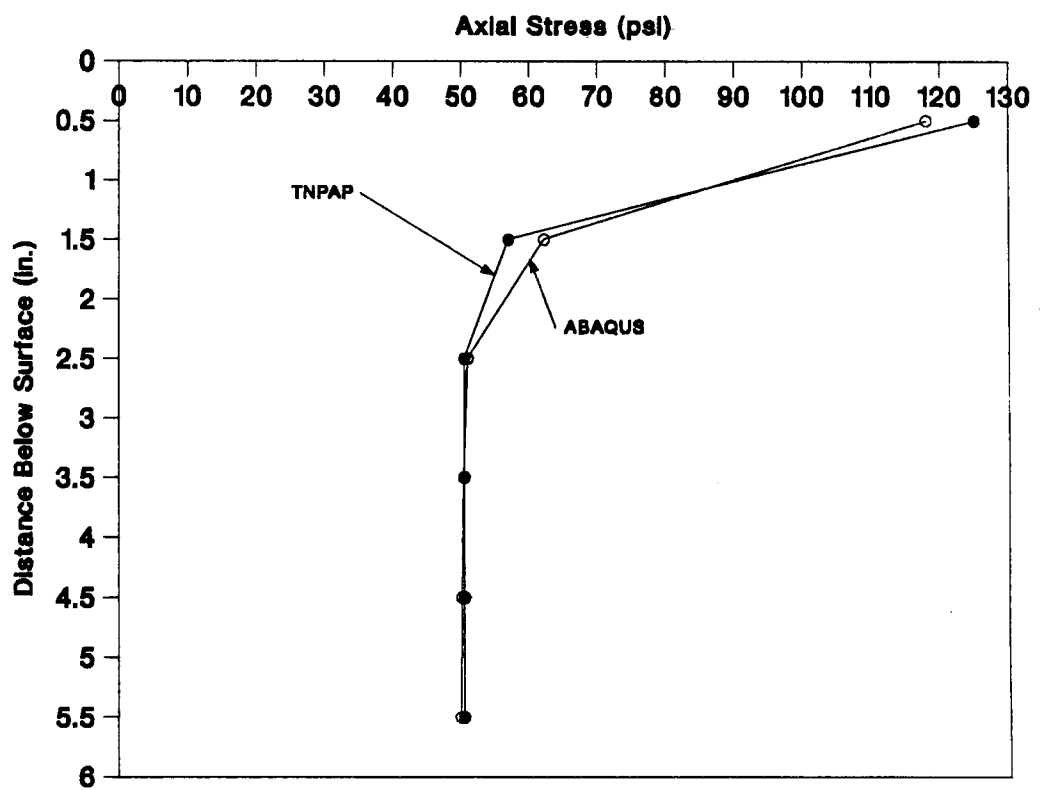


Figure 12 ABAQUS Sample Problem: Axial Stress Through Model Thickness

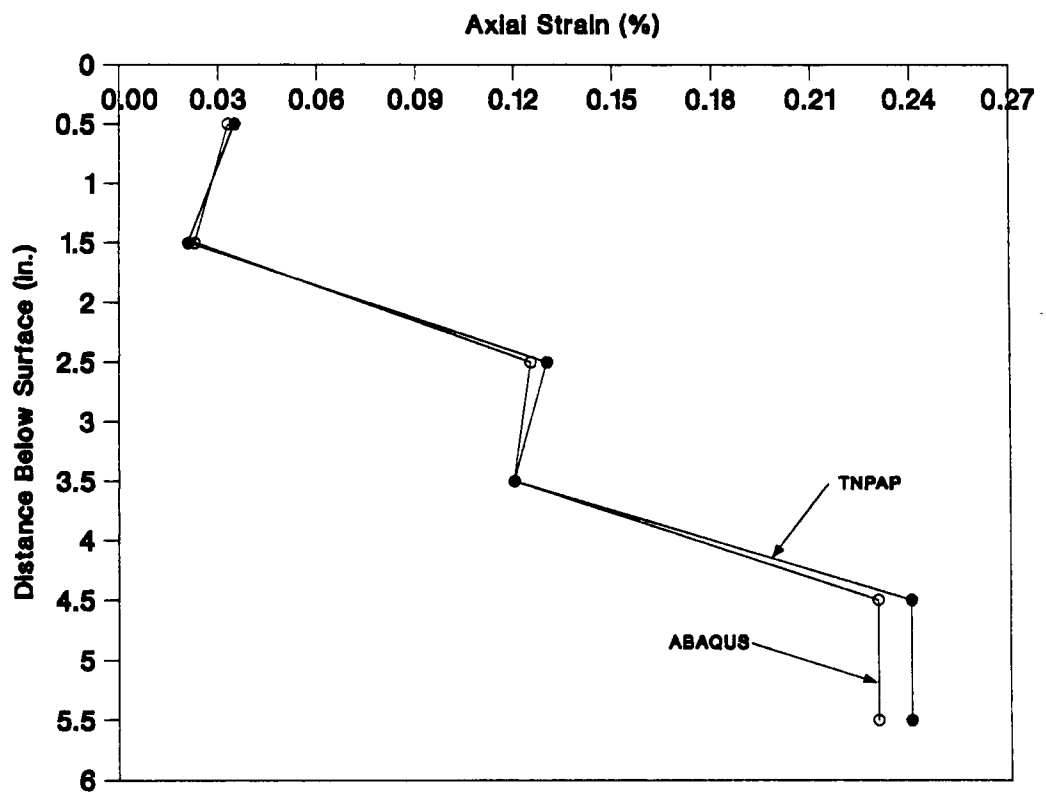


Figure 13 ABAQUS Sample Problem: Axial Strain Through Model Thickness

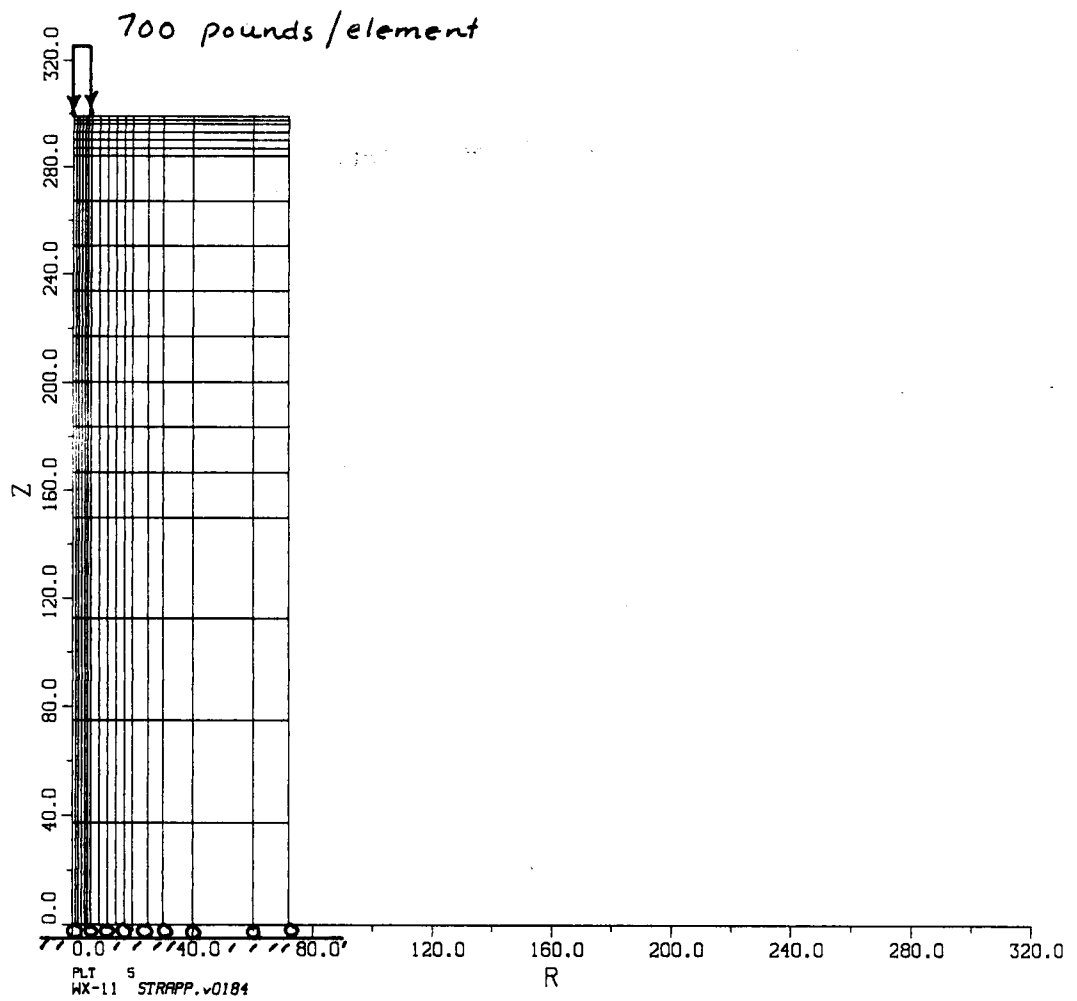


Figure 14 Finite Element Mesh Density for ADINA Sample Problem

split diagonally to create the triangular model used for the TNPAP analysis. The orientation of the triangles was the same for this problem as in the comparative analysis with ABAQUS. A 700 pound load was applied to each element on the top surface of the model from the center to a radius of six inches. The model was fixed in the axial direction along the bottom surface. A resilient modulus of 500,000 psi and a Poisson's ratio of 0.4 were used to describe the material properties of the cylinder. The results of the two analyses are compared in Figures 15 through 18. Figure 15 compares the resulting axial deflection of the surface of the cylinder. Figures 16, 17, and 18 picture the axial deflection, stress, and strain, respectively, through the depth of the cylinder at the centerline. Once again, excellent agreement exists between the results obtained from ADINA and those obtained from TNPAP.

4.2.3 Homogeneous System and Layered System Example Problems

In the paper, "Finite Element Analyses of Pavements" (Reference 7), J. M. Duncan, C. L. Monismith, and E. L. Wilson compare analyses performed on two linear sample problems analyzed by the finite element method and elastic half-space analyses. The two linear sample problems discussed by Duncan et al. were modeled and analyzed using TNPAP.

A cylinder, 144 inches in diameter and 110 inches deep, was modeled using

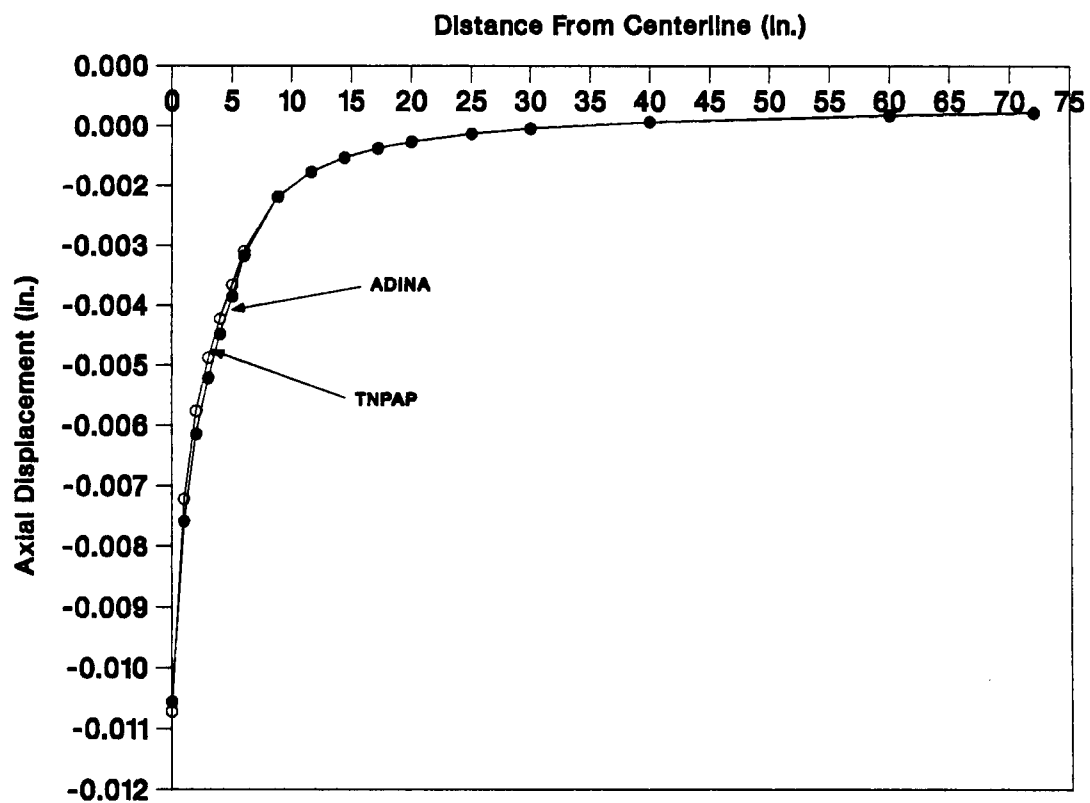


Figure 15 ADINA Sample Problem: Axial Deflection of Model Surface

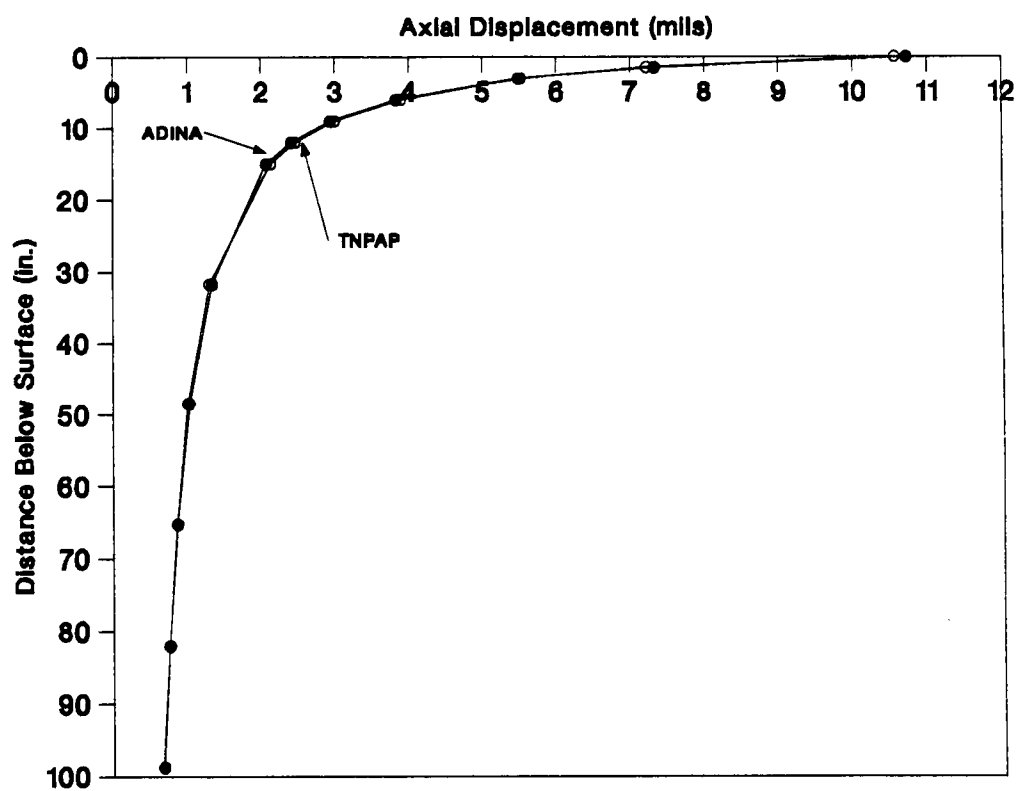


Figure 16 ADINA Sample Problem: Axial Deflection Through Model Thickness

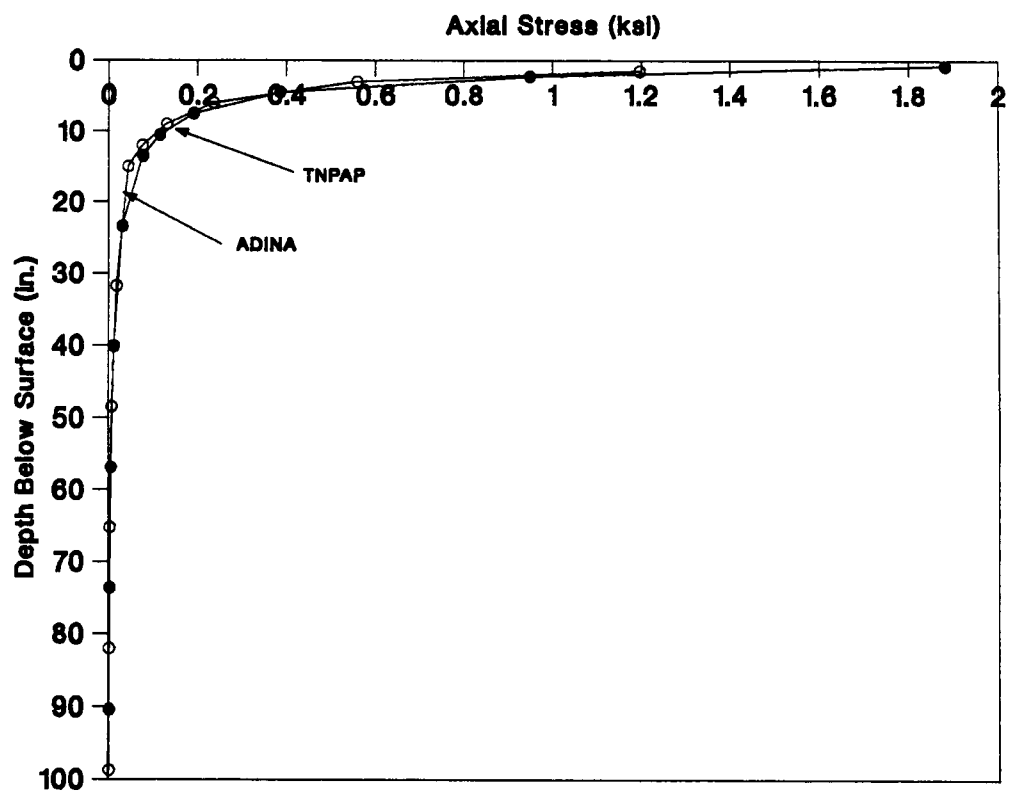


Figure 17 ADINA Sample Problem: Axial Stress Through Model Thickness

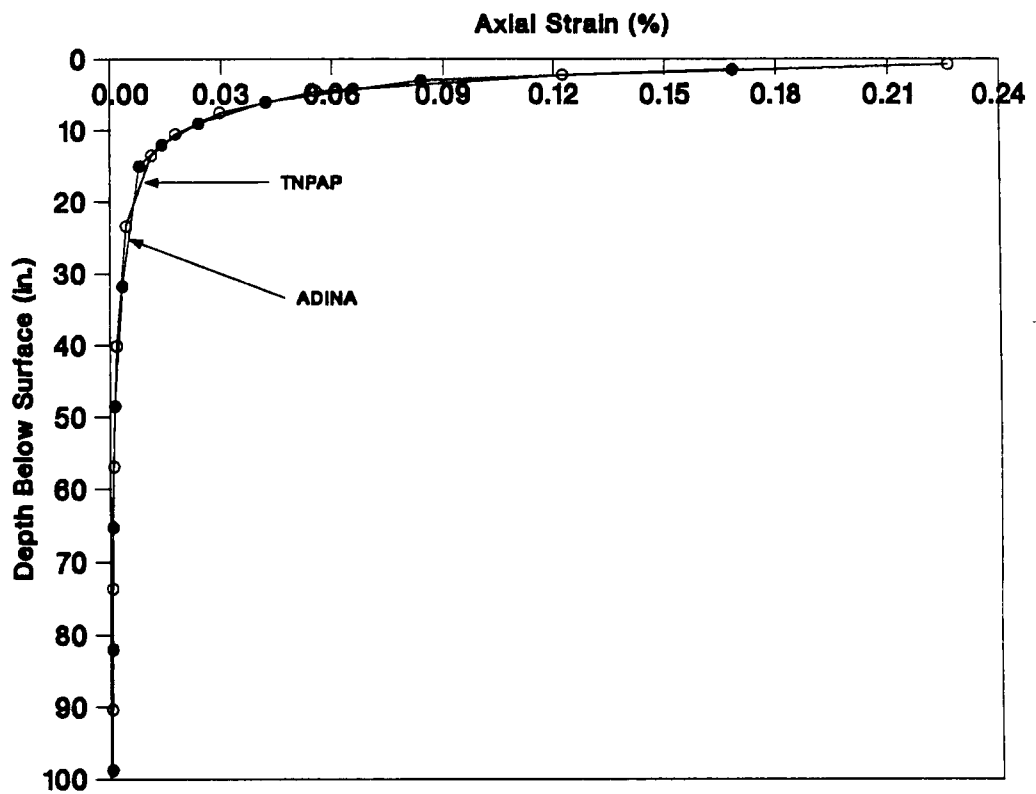


Figure 18 ADINA Sample Problem: Axial Strain Through Model Thickness

axisymmetric, triangular elements. The finite element mesh of the cylinder model is pictured in Figure 19. The mesh is pictured as quadrilateral elements for ease in viewing. Each quadrilateral was split diagonally to create the triangular elements and the element orientation was the same as presented for the ABAQUS comparison analysis. A 100 psi load was applied to a six inch radial section centered along the axis of symmetry. The mesh was divided into three layers of element groups; the outer layer was four inches thick, the mid layer was twelve inches thick, and the inner layer was ninety-four inches thick. The boundary conditions used in the analyses assumed radial confinement of the cylinder with the bottom surface of the cylinder fixed; no radial or axial movement.

For the first case, the homogeneous system, all three layers were analyzed with identical material property specifications. A resilient modulus of 10,000 psi and a Poisson's ratio of 0.4 were used to describe the material in each layer. Because this was a linear analysis and the material properties for each element stay constant throughout the analysis, only one iteration and one load increment were used. A comparison of the results obtained in the analysis of the cylinder with TNPAP and the results presented by Duncan et al. are illustrated in Figures 20 and 21. Figure 20 compares results of the axial deflection of the surface of the cylinder and Figure 21 pictures the comparison of axial stress through the depth of the cylinder. As can be seen from these figures, very good agreement

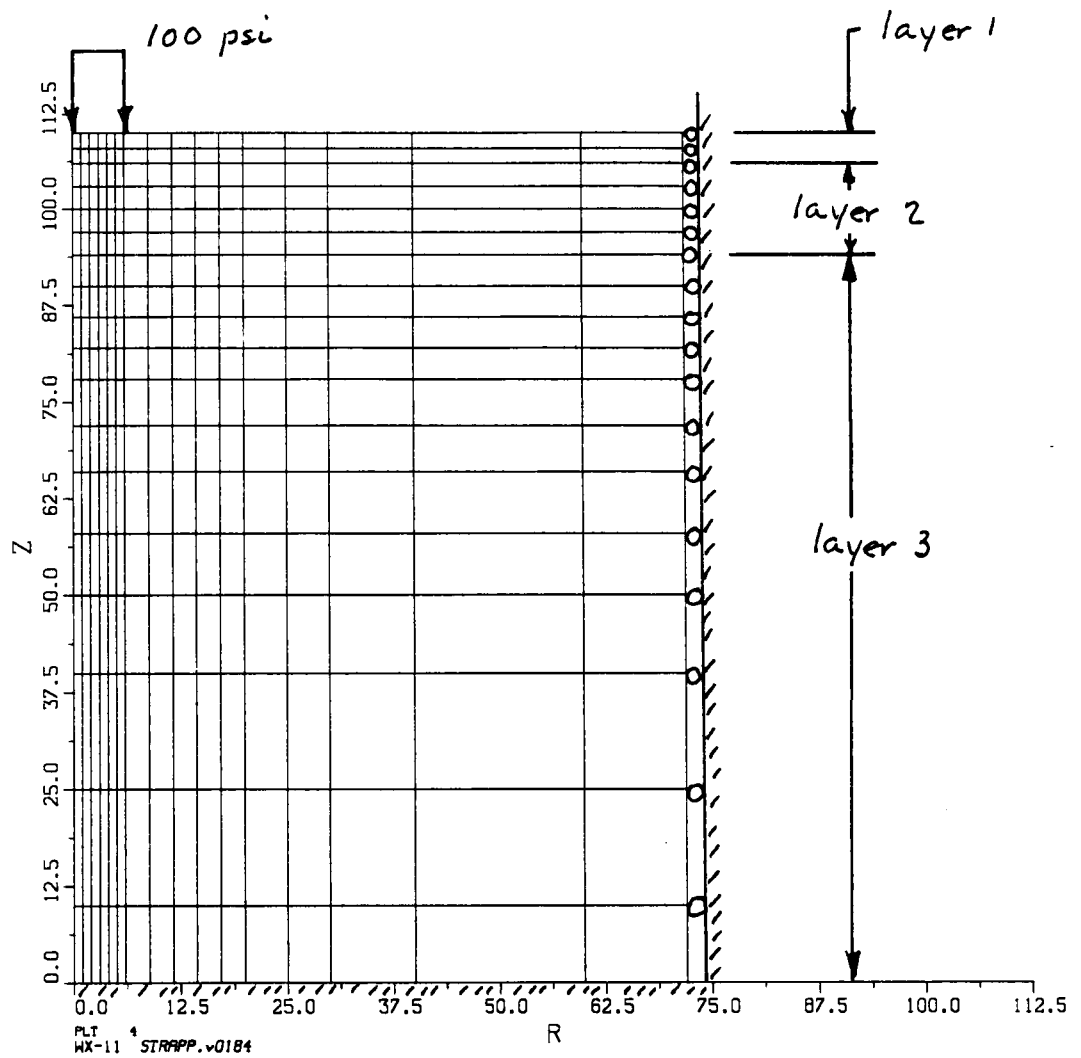


Figure 19 Finite Element Mesh Density for Homogeneous and Layered System Problems

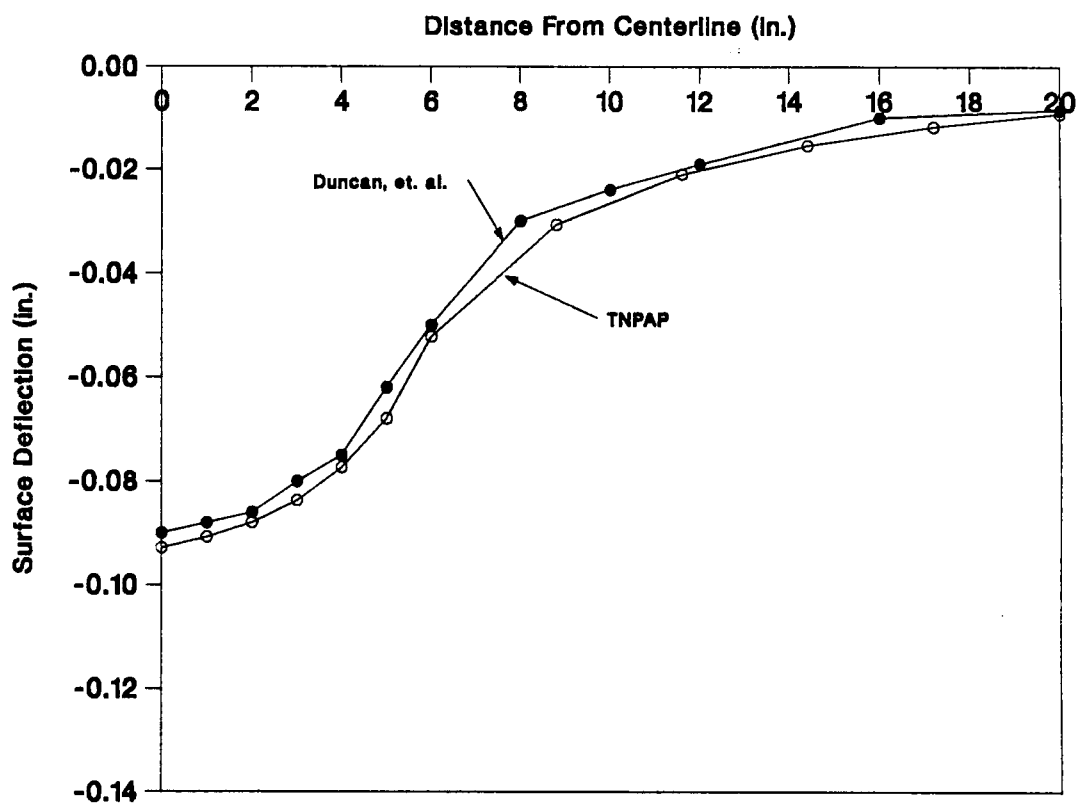


Figure 20 110 Inch Deep Model of Homogeneous System: Surface Deflection

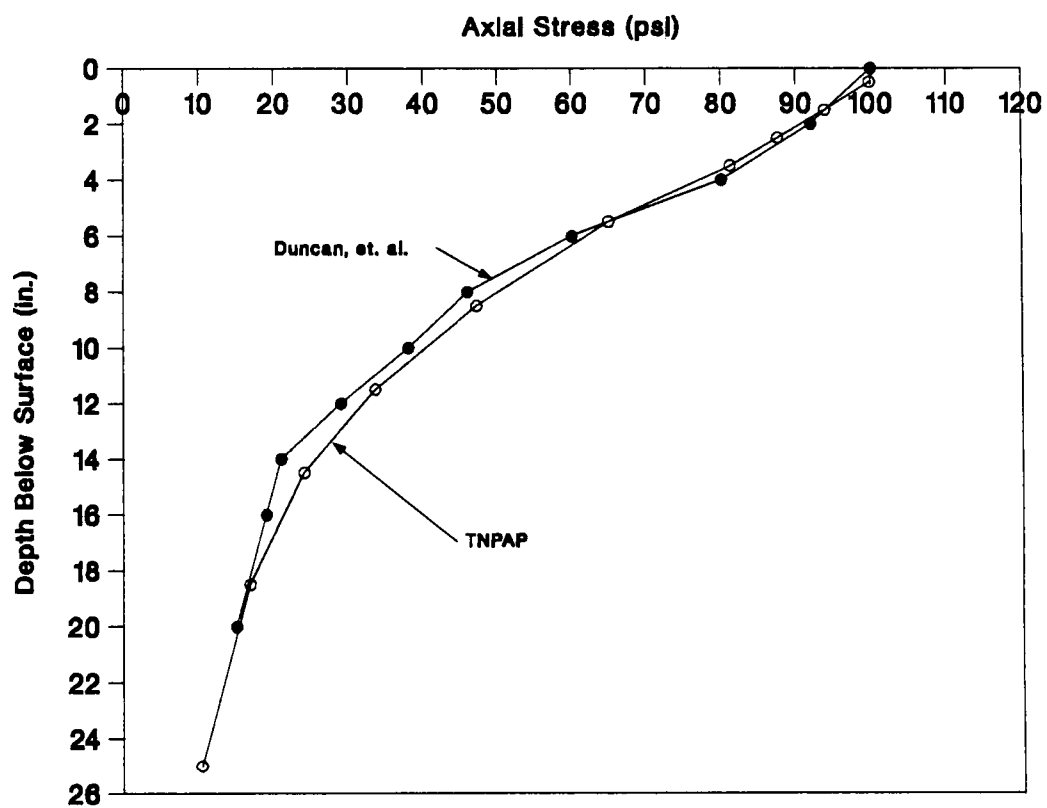


Figure 21 110 Inch Deep Model of Homogeneous System: Axial Stress

was obtained from the results of the TNPAP run and the results presented by Duncan et al.

For the layered system analysis, the first layer was modeled with a resilient modulus of 200,000 psi, the second layer had a modulus of 20,000 psi, and layer three had a modulus of 10,000 psi. All three layers were described with a Poisson's ratio of 0.4. Once again, only one iteration and one load increment were used for the analysis due to the linearity of the problem. The results of the analysis are compared with the results presented by Duncan et al. in Figures 22 and 23. Figure 22 shows the axial deflection of the surface and Figure 23 presents a comparison of the axial stresses through the depth of the cylinder. The results again compare very favorably.

The analysis of the layered system was also run using a 300 inch deep model. The deeper cylinder model was used in the layered system analysis because, as the authors of Reference 7 discuss, surface deflections are a function of depth beneath the surface for a layered system. The thicknesses of the top two layers remained the same and the bottom layer increased to a thickness of 284 inches. The results of the thicker model analysis are shown in Figures 24 and 25.

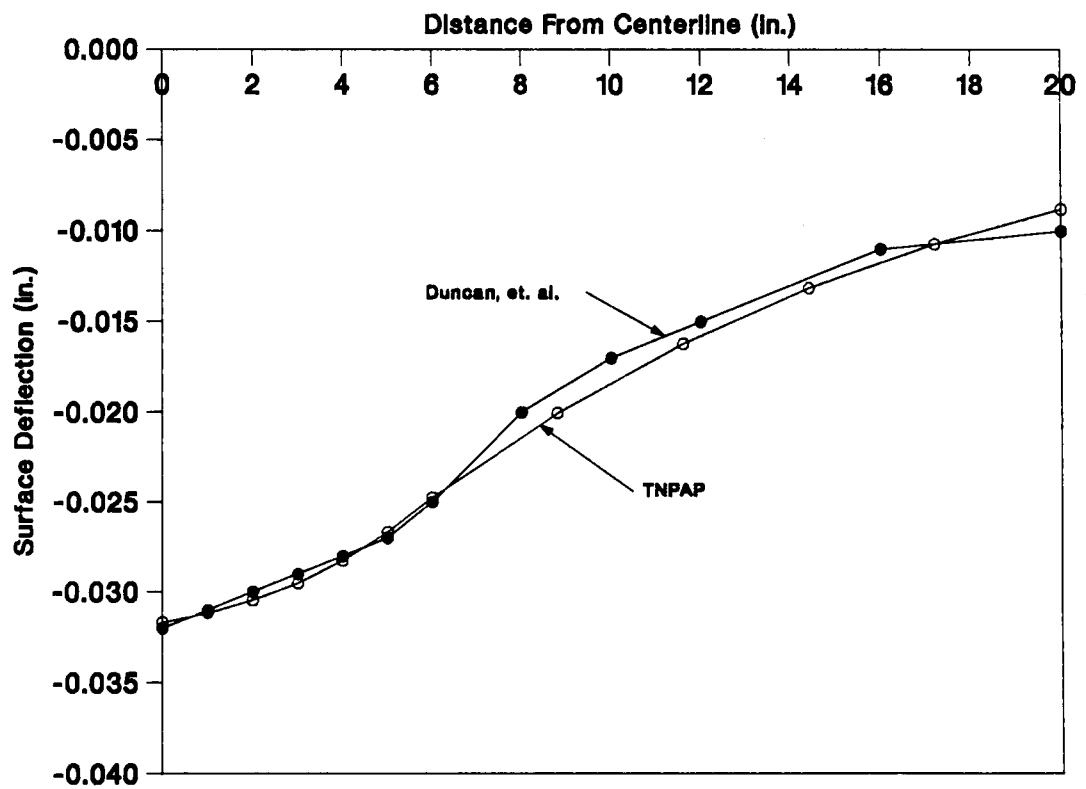


Figure 22 110 Inch Deep Model of Layered System: Surface Deflection

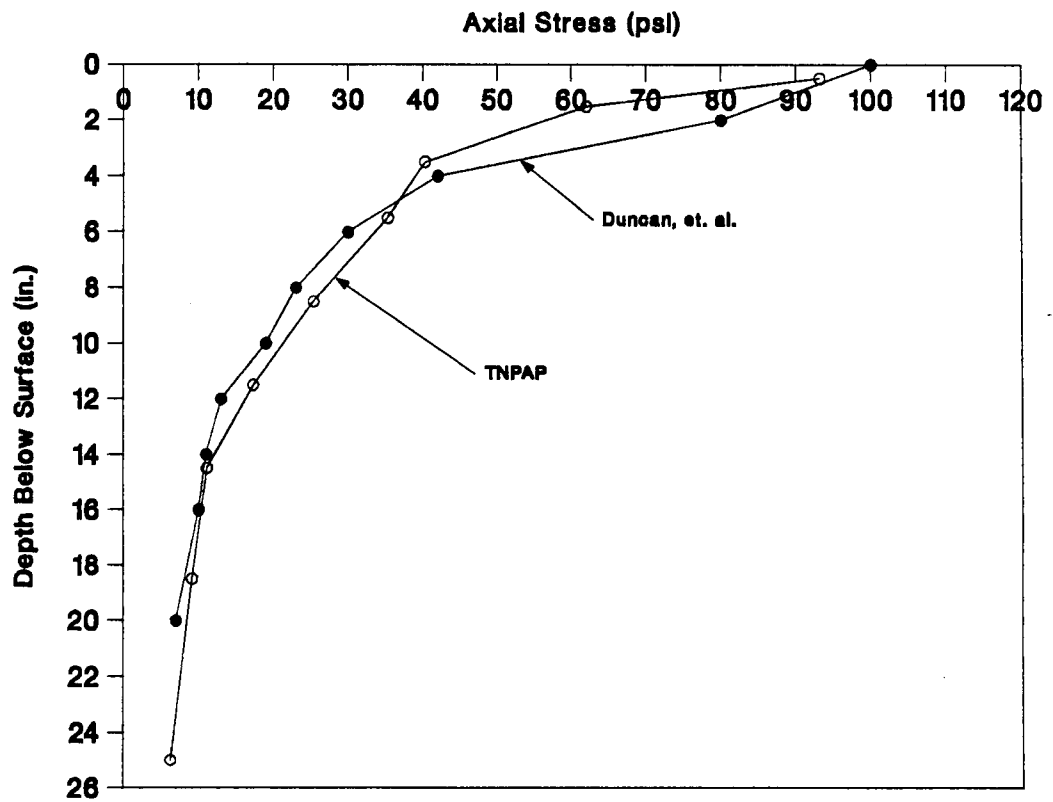


Figure 23 110 Inch Deep Model of Layered System: Axial Stress

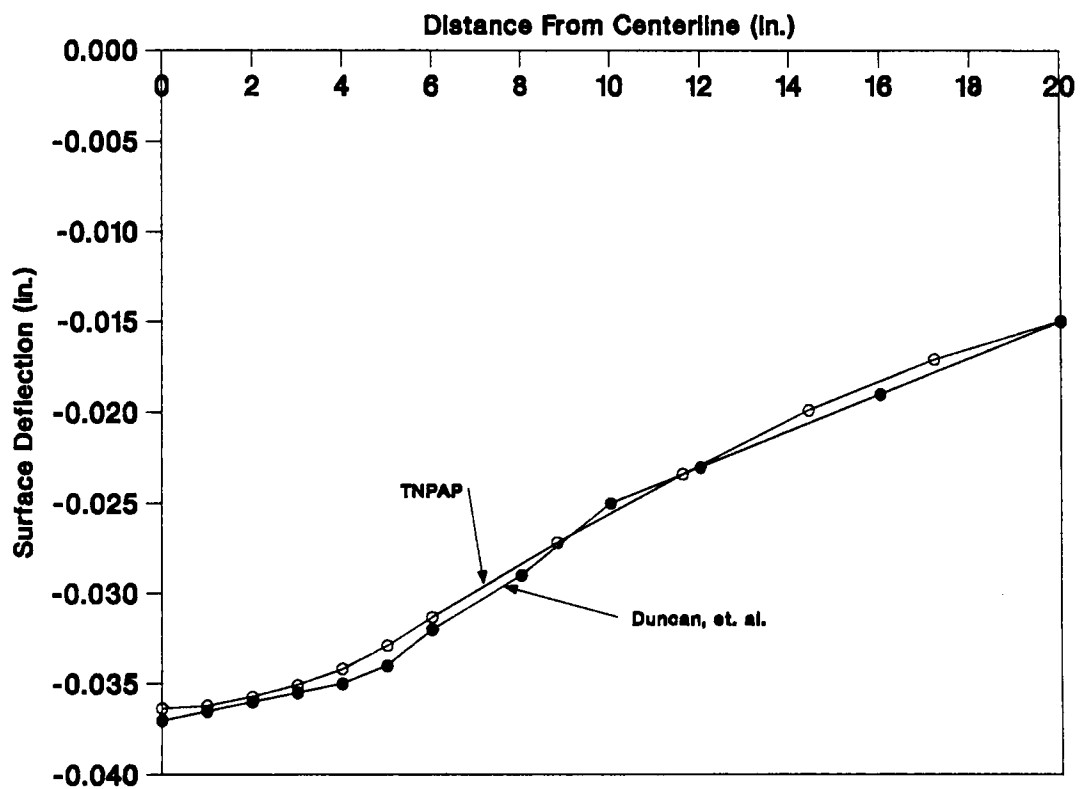


Figure 24 300 Inch Deep Model of Layered System: Surface Deflection

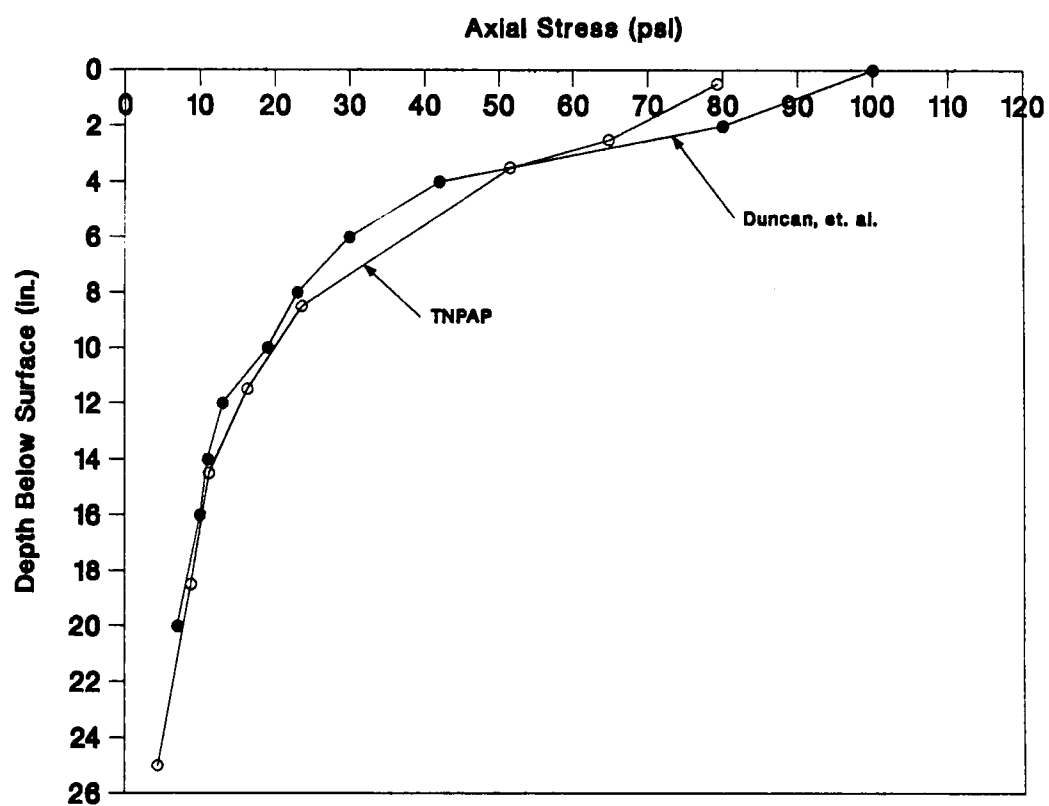


Figure 25 300 Inch Deep Model of Layered System: Axial Stress

4.3 Nonlinear Analyses

Four nonlinear problems were analyzed using the modified finite element code. Analyses were run on the Gonzales Bypass (Reference 7), two sample problems which were compared with results from ILLI-PAVE (Reference 5), and Rutledge Pike in Knoxville, Tennessee.

4.3.1 Gonzales Bypass

Duncan, Monismith, and Wilson in Reference 7 not only treat the topic of linear analyses, but also present results of nonlinear analyses performed on the Gonzales Bypass. Duncan et al. present analytical results for the Gonzales Bypass for both summer and winter conditions. The problem was modeled for analyses with TNPAP. The finite element model used in the analyses is illustrated in Figure 26. The elements are shown as quadrilaterals to make density of the mesh easier to see. However, once again the elements were split diagonally to make triangular elements. The orientation of the elements was identical to the elements in the sample ABAQUS comparison problem. The cylinder was modeled 270 inches in diameter and 197.6 inches deep. The model is made up of four layers; 6.6 inches of asphaltic concrete, 6.0 inches of base, 15.0 inches of subbase, and 170 inches of subgrade. Winter conditions were analyzed using a resilient modulus for the asphalt concrete of 1,500,000 psi and

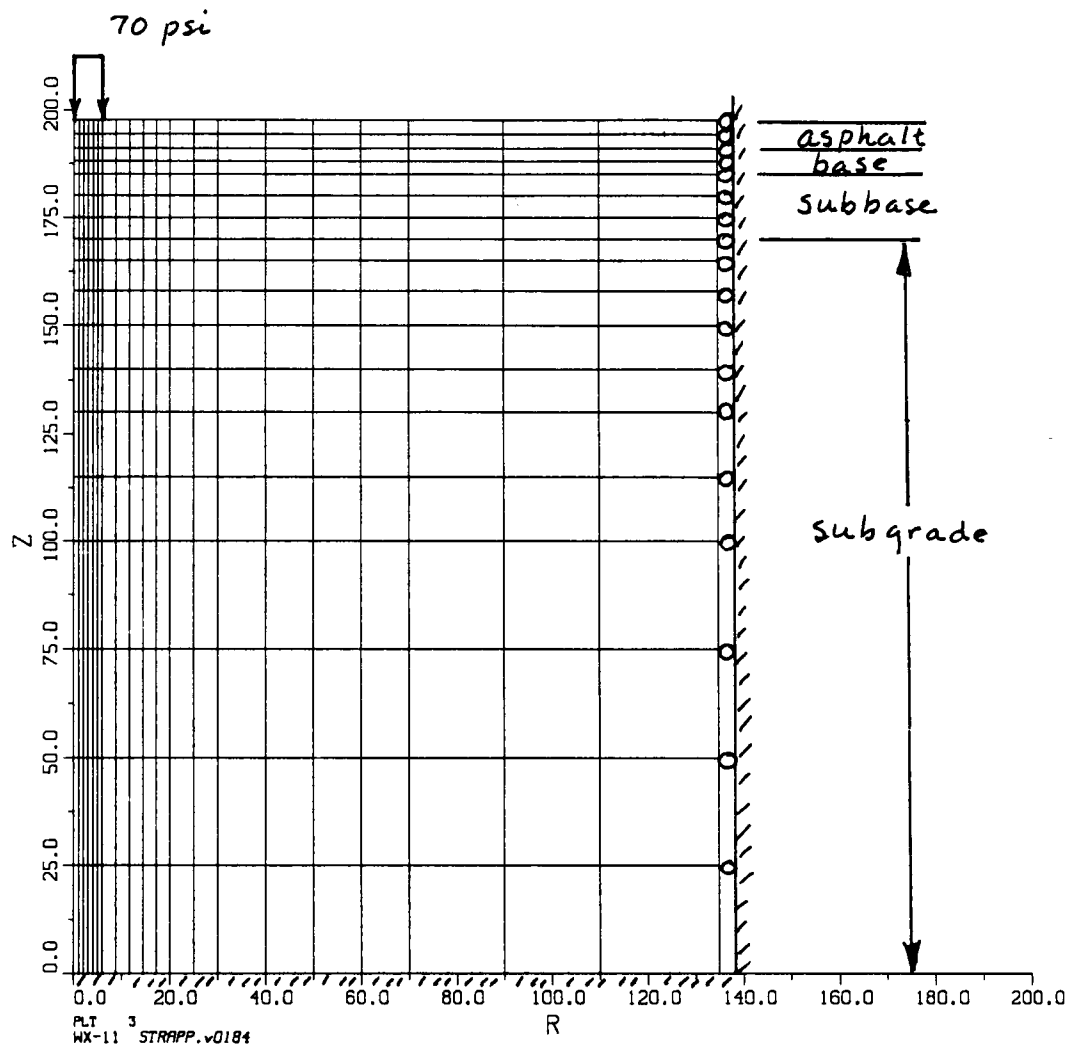


Figure 26 Finite Element Mesh Density for Gonzales Bypass

a Poisson's ratio of 0.4. The resilient modulus was determined by taking an average of the winter range of the stiffness modulus. Reference 7 indicates that the stiffness modulus ranges between 1,420,000 and 1,550,000 psi in the winter months. Summer conditions were analyzed using a resilient modulus of 200,000 psi and a Poisson's ratio of 0.4. The range of stiffness modulus in the summer months is reported in Reference 7 as being between 120,000 and 280,000 psi. The base was modeled using Equation 1 and defining the constants K and n equal to 15,000 and 0.48, respectively. Poisson's ratio was assumed to be 0.4. The subbase was also modeled using Equation 1 with constants K and n defined as 10,000 and 0.4, respectively, and Poisson's ratio defined as 0.4. The subbase was modeled with a bilinear representation. Constants E_{ri} , K1, K2, and S_{di} in Equations 4 and 5 were defined as 9,500, -11,000, -286, and 2.6, respectively. Poisson's ratio was assumed to be .47. A 70 psi load was applied to a 12 inch diameter section along the axis of symmetry of the cylinder. The cylinder was constrained radially and was fixed at the base. The axial deflection through the depth of the cylinder results of the winter and summer analyses of the Gonzales Bypass are presented in Figures 27 and 28, respectively. Figures 29 and 30 illustrate the axial stress as a function of depth of the cylinder for the winter and summer conditions, respectively.

Good correlation exists between the compared results in the asphalt, base,

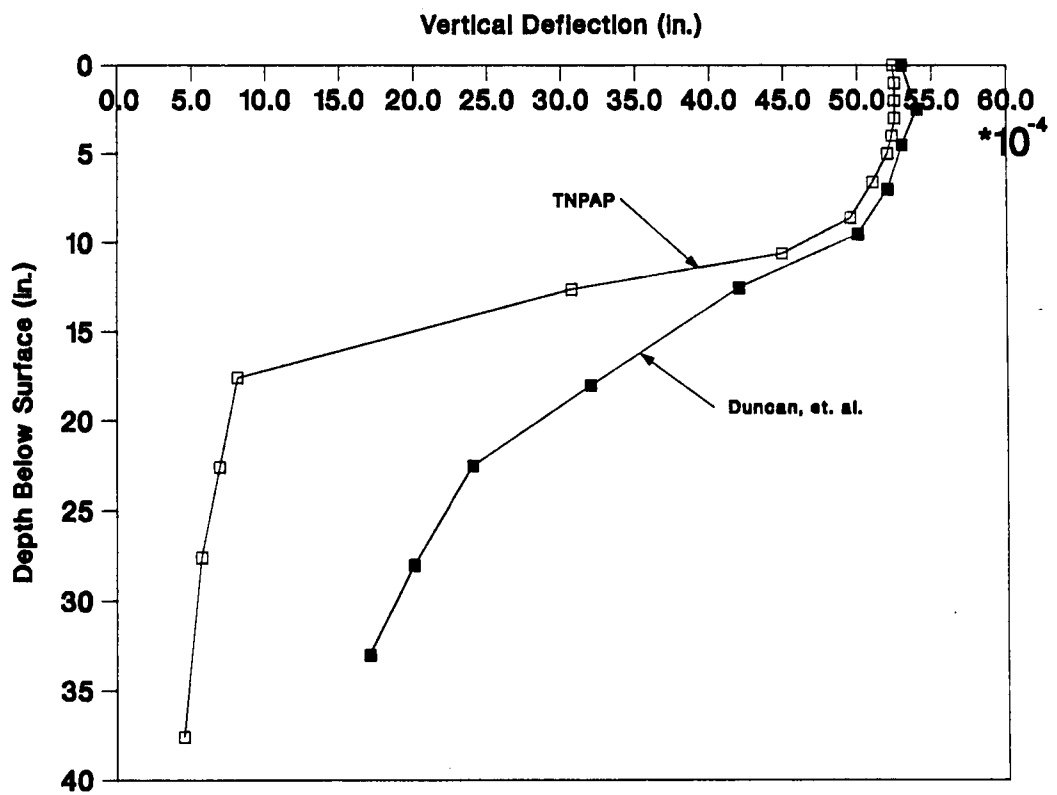


Figure 27 Axial Deflection Through Depth of Gonzales Bypass- Winter

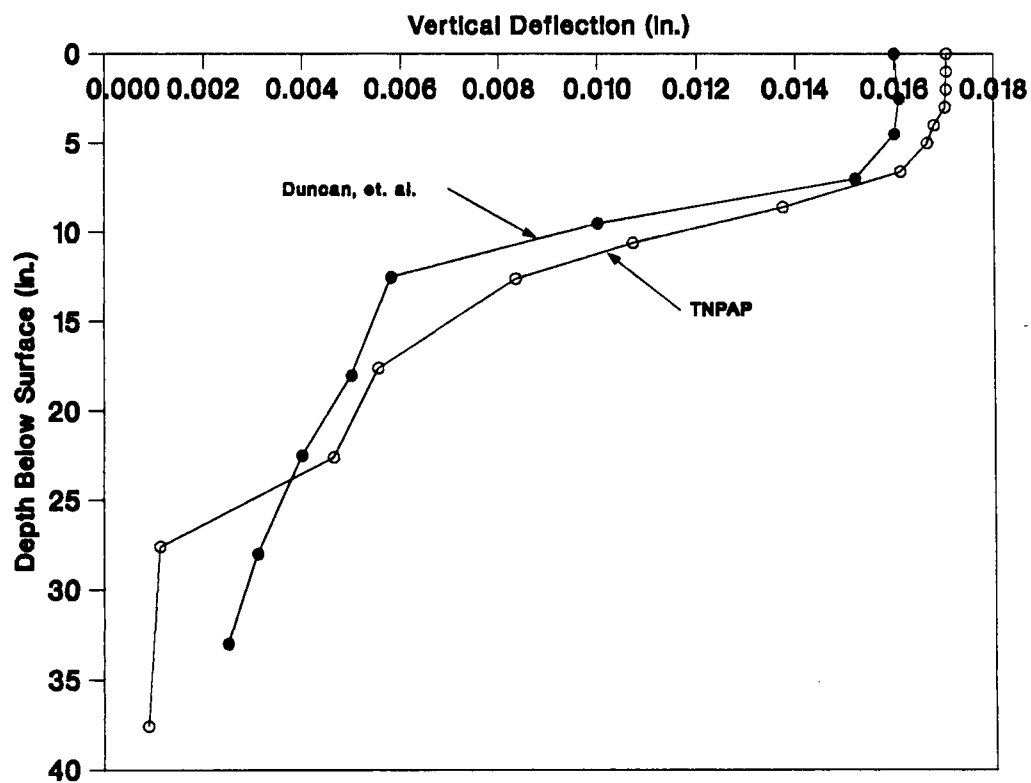


Figure 28 Axial Deflection Through Depth of Gonzales Bypass- Summer

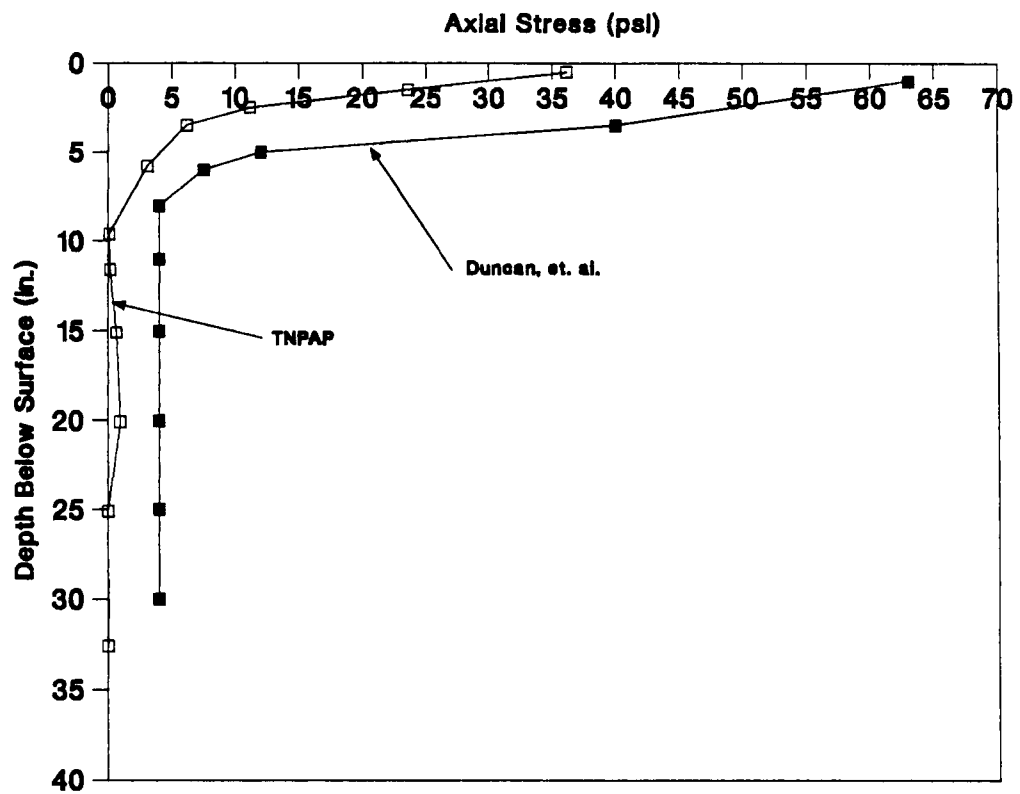


Figure 29 Axial Stress Through Depth of Gonzales Bypass- Winter

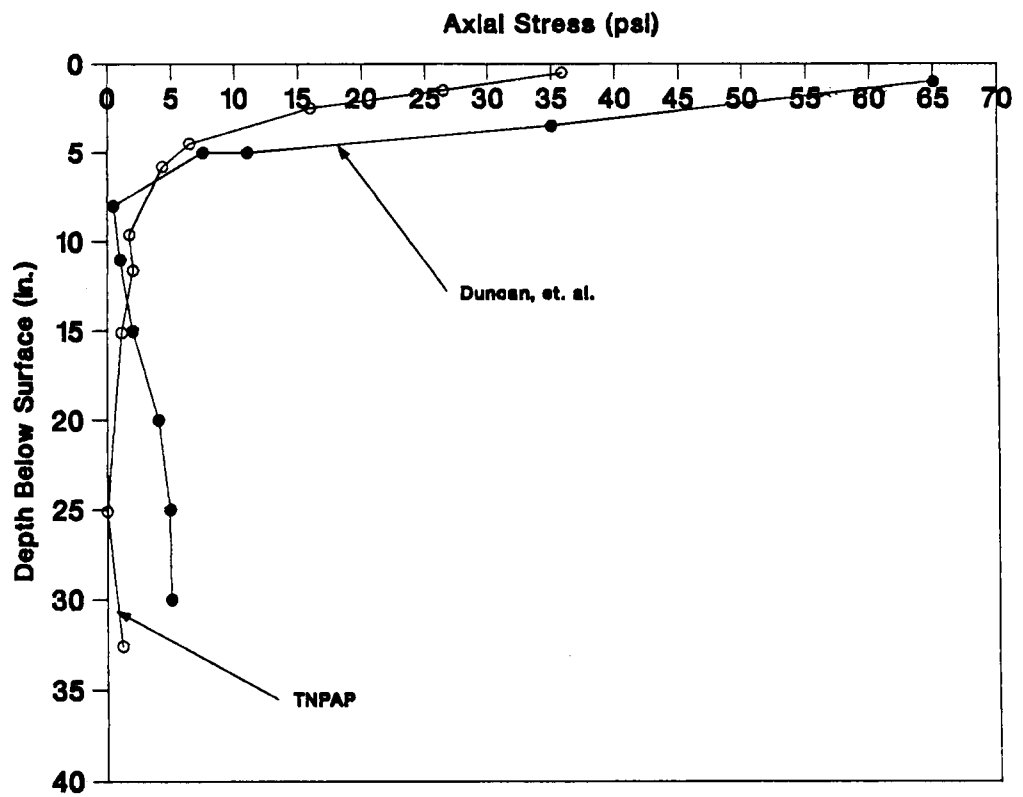


Figure 30 Axial Stress Through Depth of Gonzales Bypass- Summer

and subbase layers. The subgrade layer results appear to deviate between the TNPAP run and the results presented in Reference 7. This could be a result of mesh density in the models used.

4.3.2 ILLI-PAVE

ILLI-PAVE is a finite element code developed at the University of Illinois for analyzing pavement structures. The development of the code is described in Reference 5. Two nonlinear analyses problems were selected from the many that are presented in Reference 5 for comparison with results obtained from TNPAP. Figure 31 illustrates the model used in the analyses of the nonlinear problems selected from Reference 5. The elements in the model are pictured as quadrilaterals in the figure. Each element was split diagonally to create two triangular elements with the same orientation as shown by the triangular elements in the ABAQUS comparison sample problem. The cylinder was modeled with a 144.0 inch diameter and 300.0 inches deep. The cylinder consisted of three layers; 3.0 inches of asphaltic concrete, 4.0 inches of base material, and 293.0 inches of subbase. A load of 80 psi was applied to a 12 inch diameter center section along the axis of symmetry of the cylinder. The cylinder was confined radially and fixed at the base.

For the first analysis, the asphalt was modeled with a resilient modulus

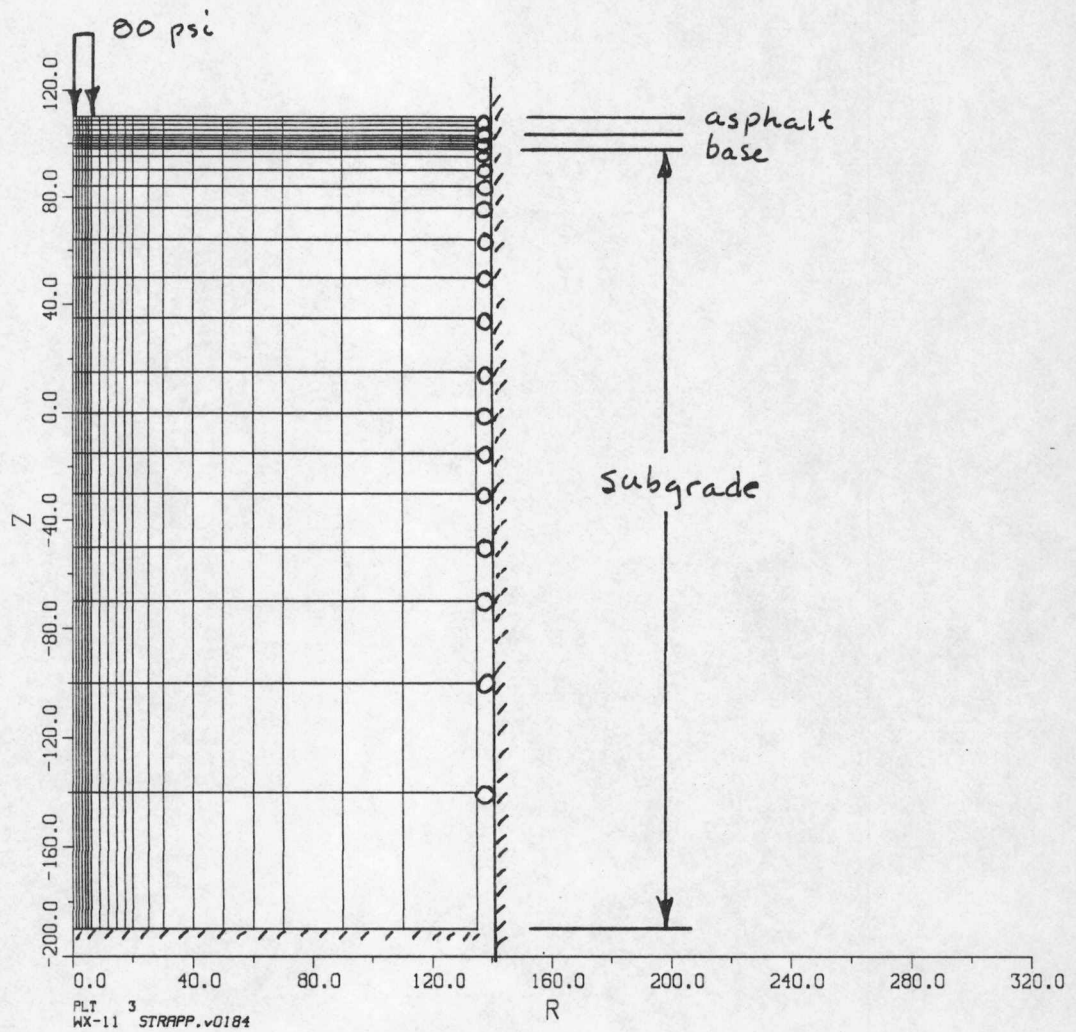


Figure 31 Finite Element Mesh Density for ILLI-PAVE Sample Problem

of 500,000 psi and a Poisson's ratio of 0.4. The base material was modeled by defining constants K and n of Equation 1 as 9,000 and 0.33 and Poisson's ratio was defined as 0.38. The subgrade material was modeled with the bilinear Equations 4 and 5 where E_{ri} was specified as 12,340, K_1 as -1,110, K_2 as -178, and S_{di} as 6.2. A Poisson's ratio of 0.45 was used. Figures 32 and 33 illustrate the resulting axial deflection of the asphalt and subgrade surfaces, respectively. The surface axial deflections compare very well. It is hard to make a conclusive comparison with the single data point provided in Reference 5 for subgrade deflection. A trend in the deflection along the surface of the subgrade is not illustrated by one data point. Figure 34 pictures the radial strain at the asphalt base and Figure 35 shows the axial strain at the subgrade surface. Once again, single data points are presented in the results in Reference 5 and make result comparisons difficult.

In the second problem, the base material definition remained as it was in the first problem. The asphalt was modeled with a resilient modulus of 1,400,000 psi and a Poisson's ratio equal to 0.4. The subgrade was defined as in the first problem with the exception of the value for E_{ri} which was specified as 7,680. The resulting surface axial deflection, subgrade surface axial deflection, asphalt base radial strain, and subgrade surface axial strain, as compared to the ILLI-PAVE analysis, are illustrated in Figures 36 through 39. Quantitative values for

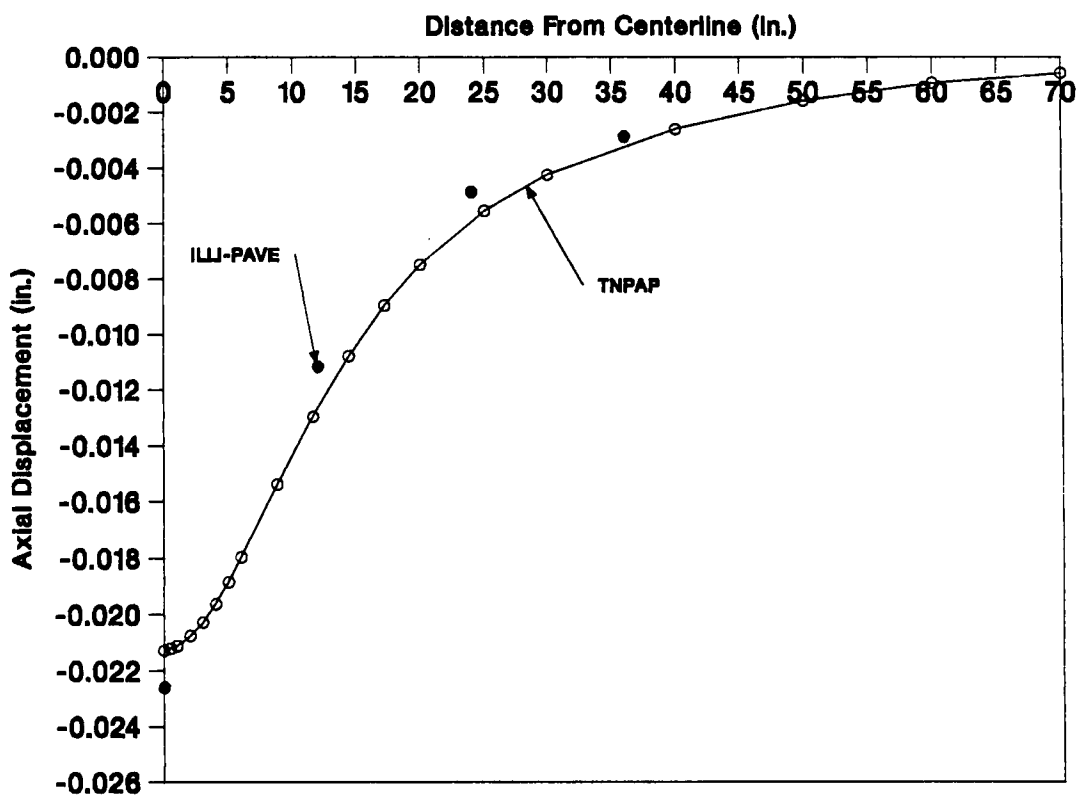


Figure 32 Surface Deflection of Cylinder— ILLI-PAVE Sample Problem Number 1

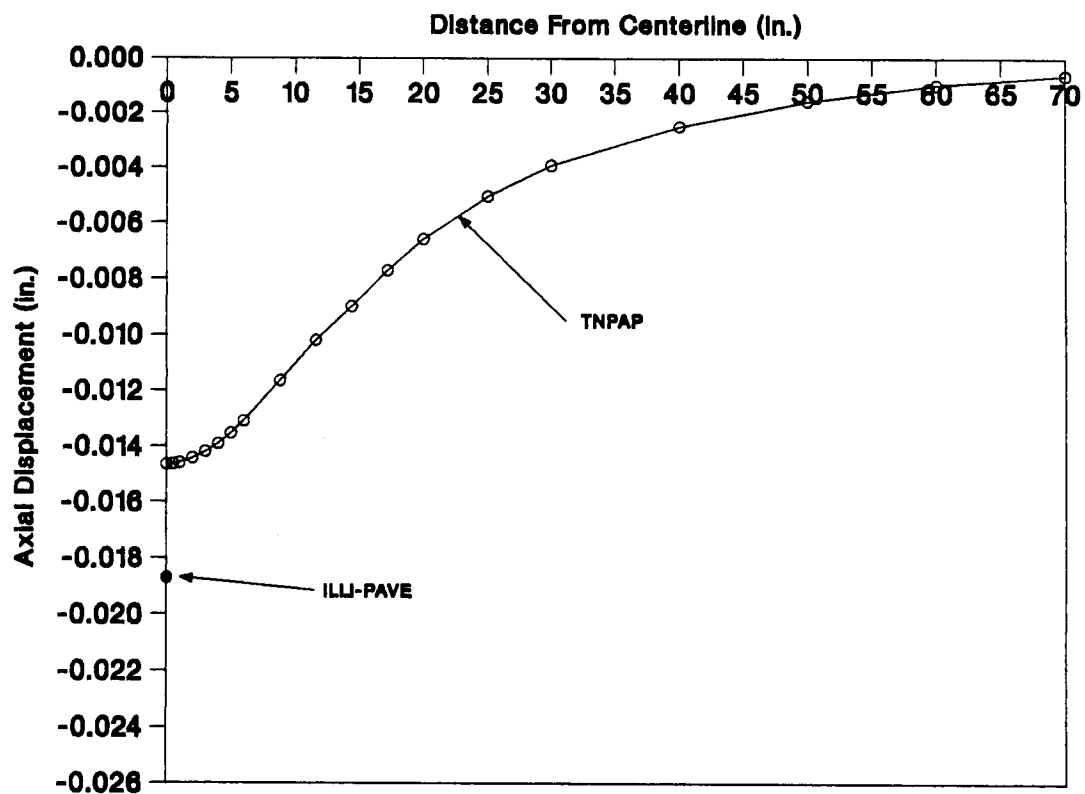


Figure 33 Deflection of Subgrade for ILLI-PAVE Sample Problem Number 1

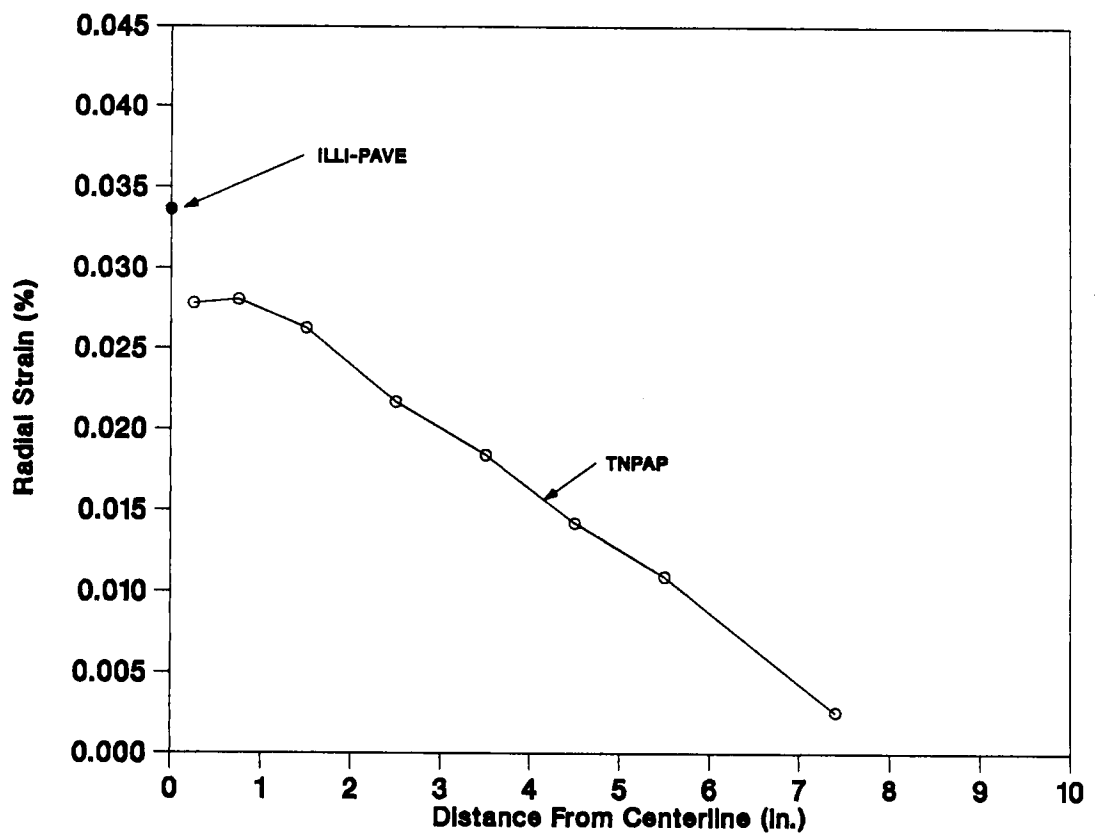


Figure 34 Radial Strain at Asphalt Base— ILLI-PAVE Sample Problem Number 1

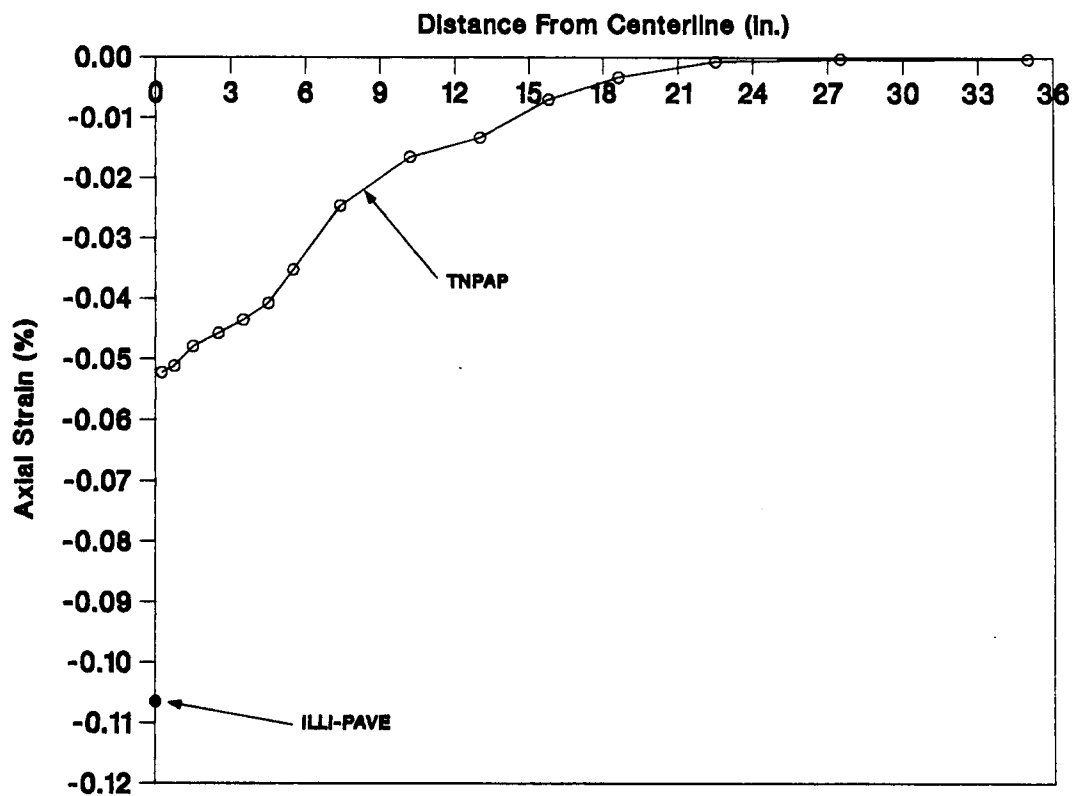


Figure 35 Axial Strain at Subgrade Surface— ILLI-PAVE Sample Problem Number 1

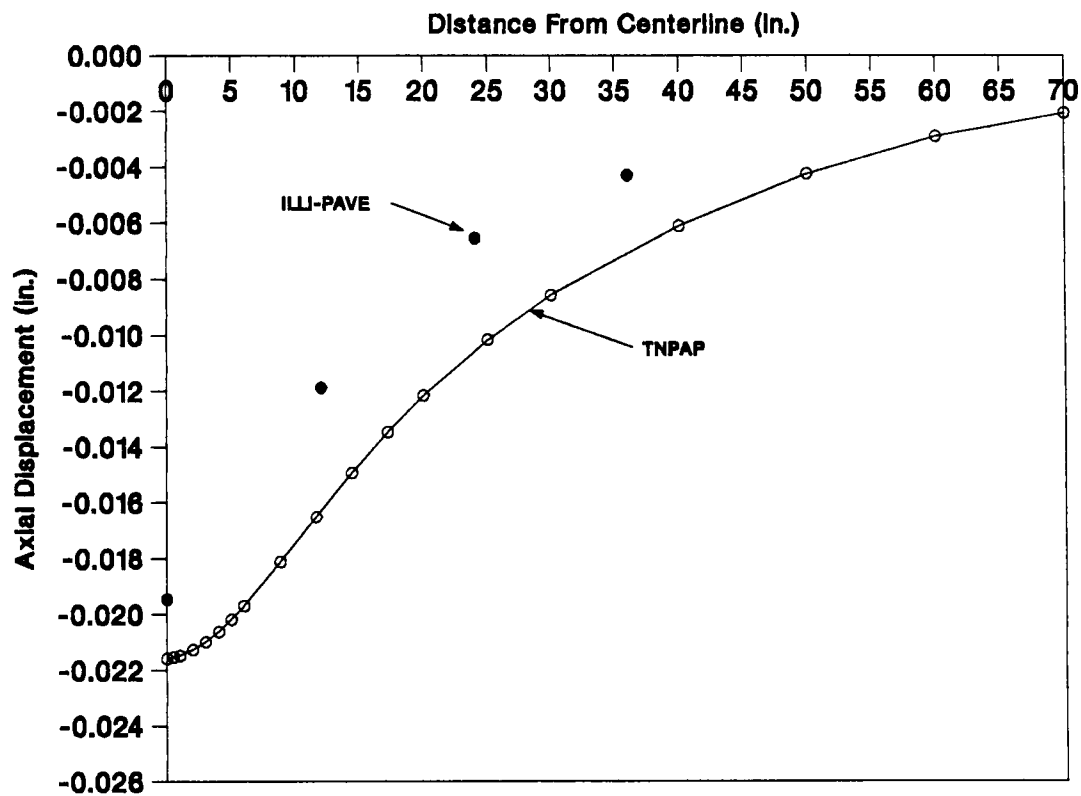


Figure 36 Surface Deflection of Cylinder— ILLI-PAVE Sample Problem Number 2

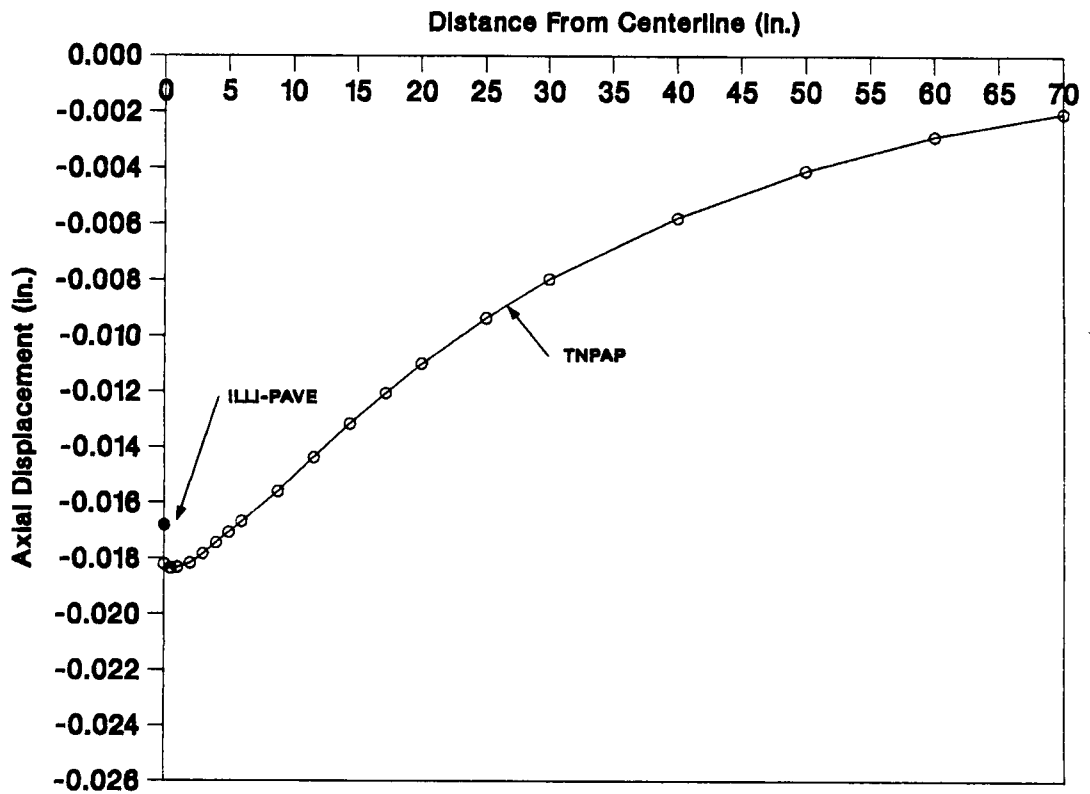


Figure 37 Deflection of Subgrade for ILLI-PAVE Sample Problem Number 2

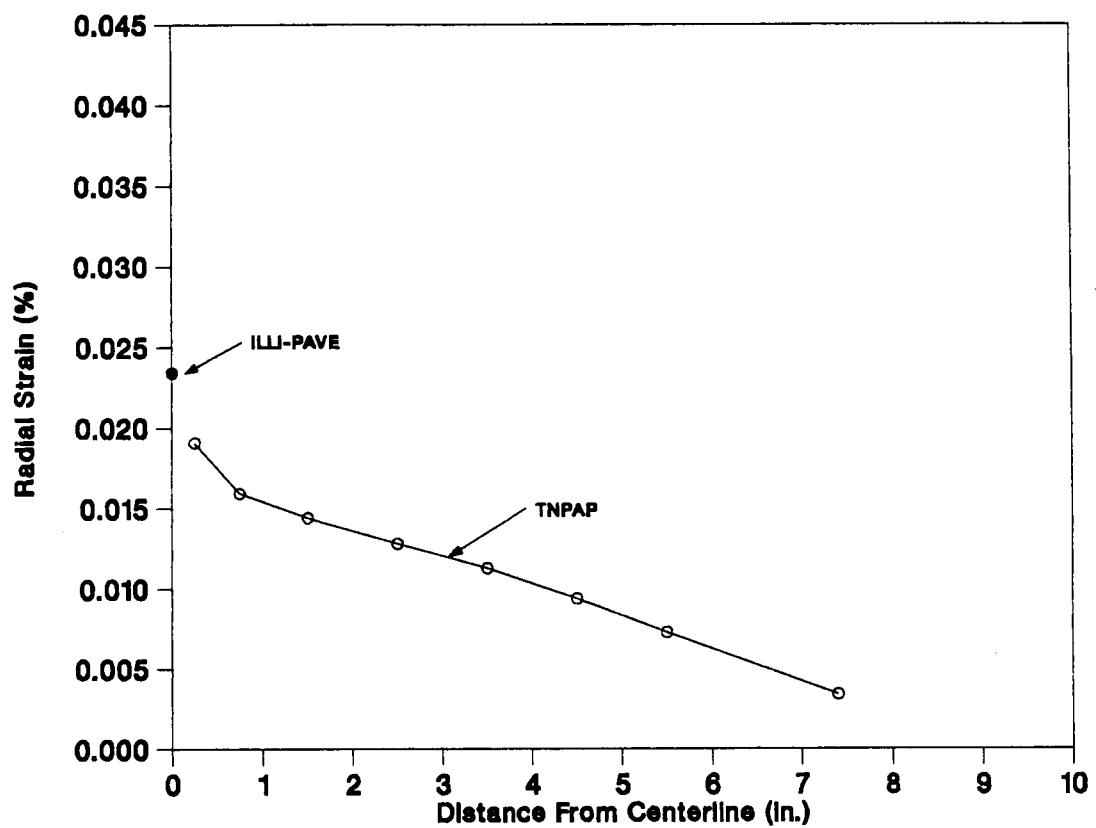


Figure 38 Radial Strain at Asphalt Base— ILLI-PAVE Sample Problem Number 2

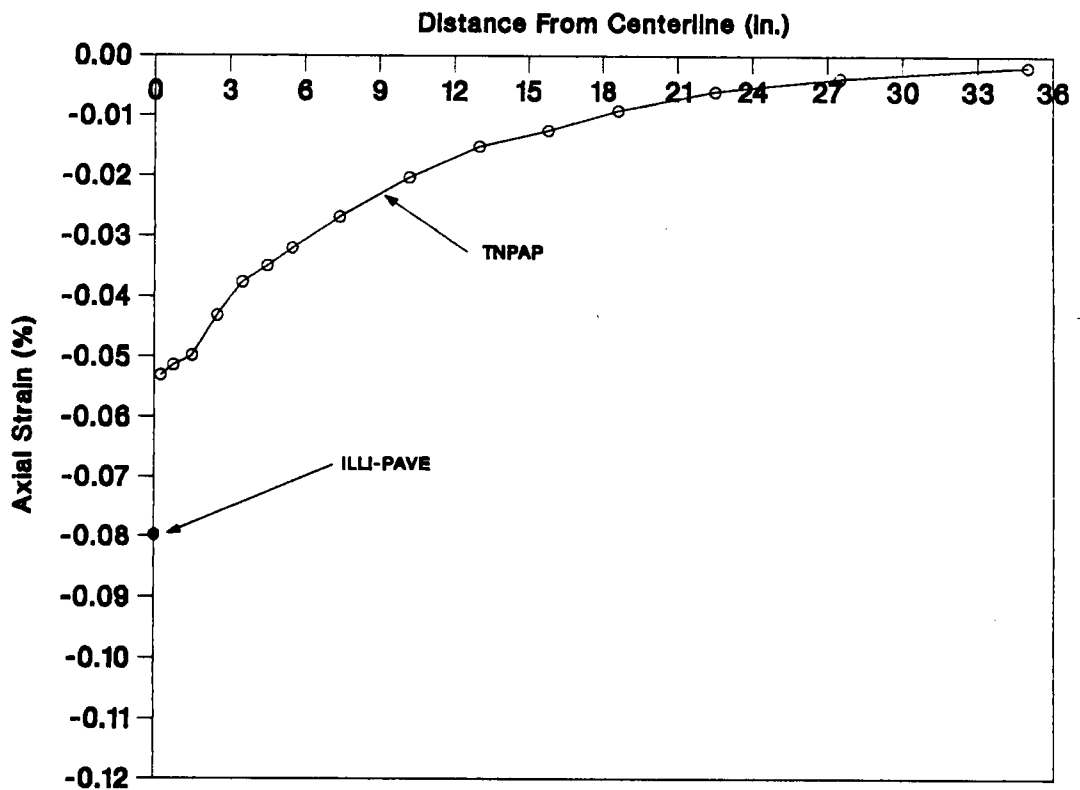


Figure 39 Axial Strain at Subgrade Surface— ILLI-PAVE Sample Problem.
Number 2

the surface deflection differ more than was shown in the first ILLI-PAVE sample problem comparison. However, the trend of the results compare very well and only a ten percent difference is indicated between the curves. The subgrade surface axial deflection, asphalt base radial strain, and subgrade surface axial strain again are difficult to compare because only one data point was presented in Reference 5.

4.3.3 Rutledge Pike

TNPAP was used to analyze a section of pavement of Rutledge Pike located in Knoxville, Tennessee. Two cases were run on the three models in this analysis. The models, pictured in Figures 40, 41, and 42, were all cylinders with a diameter of 270 inches. To examine the sensitivity of subgrade depth in a model on finite element results, three depths were used in the analysis: Cylinder A, illustrated in Figure 40, had a subgrade depth of 100 inches and a total depth of 110 inches, Cylinder B, in Figure 41, had a subgrade depth of 200 inches and a total depth of 210 inches, and Cylinder C, shown in Figure 42, had a subgrade depth of 290 inches and a total depth of 300 inches. Table 1 illustrates the matrix of calculations carried out in this analysis.

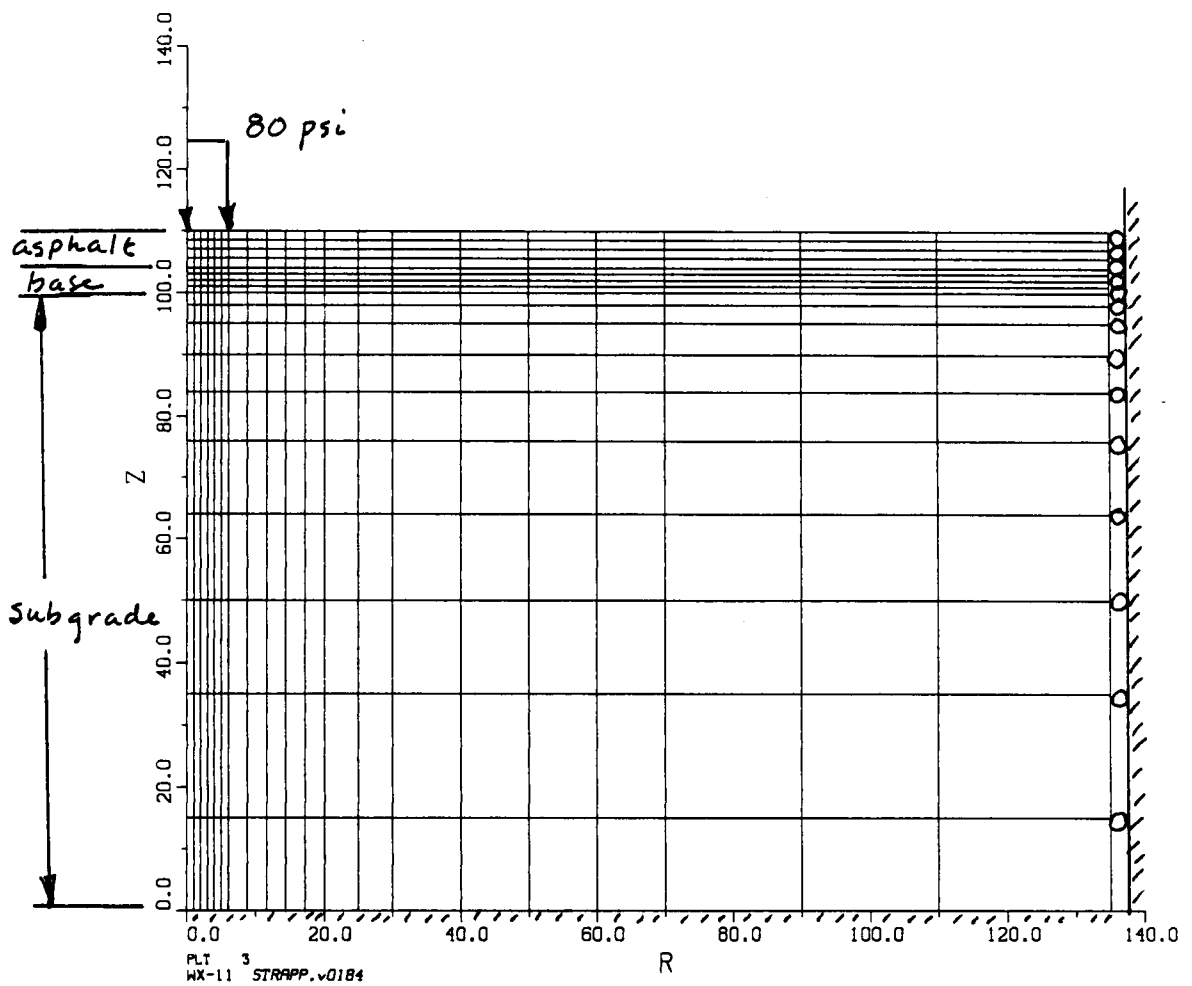


Figure 40 Finite Element Mesh Density for Rutledge Pike 110 Inch Deep Model

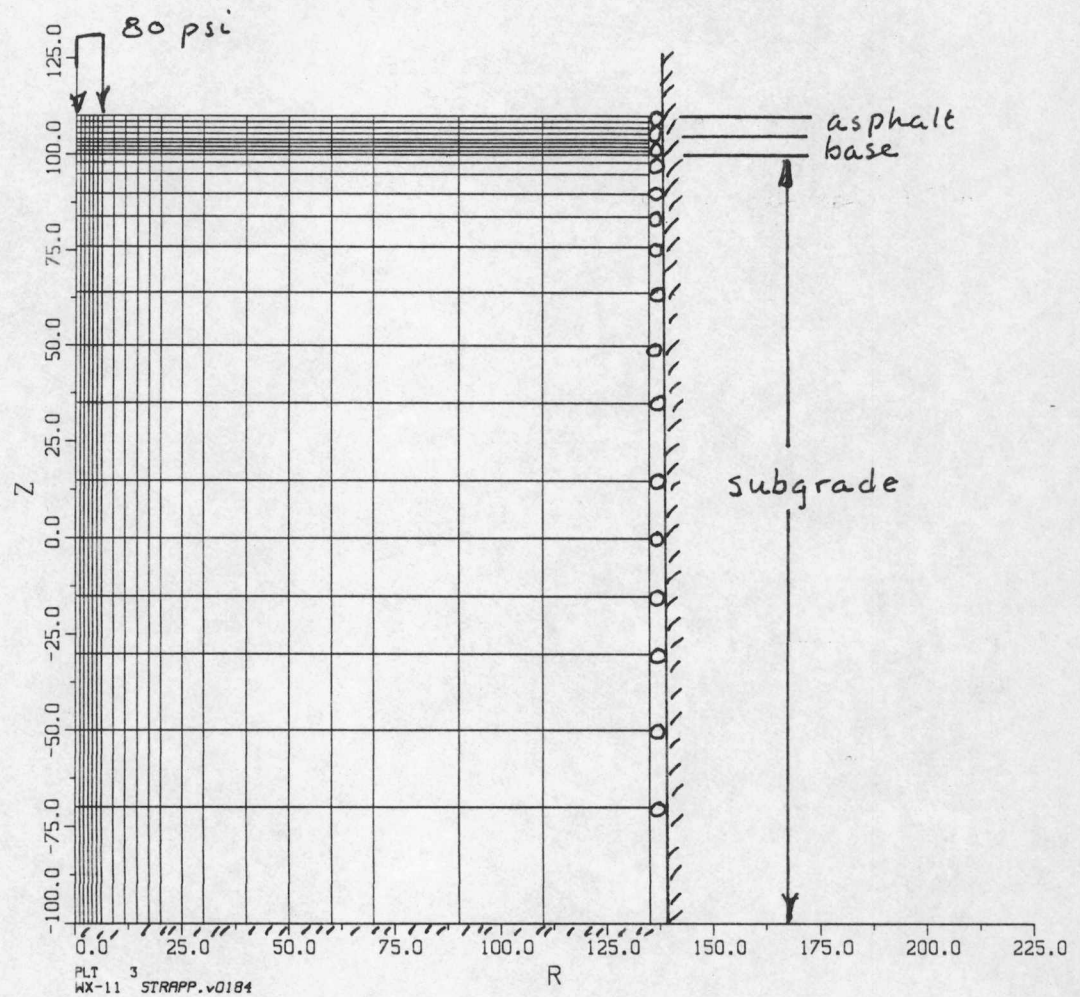


Figure 41 Finite Element Mesh Density for Rutledge Pike 210 Inch Deep Model

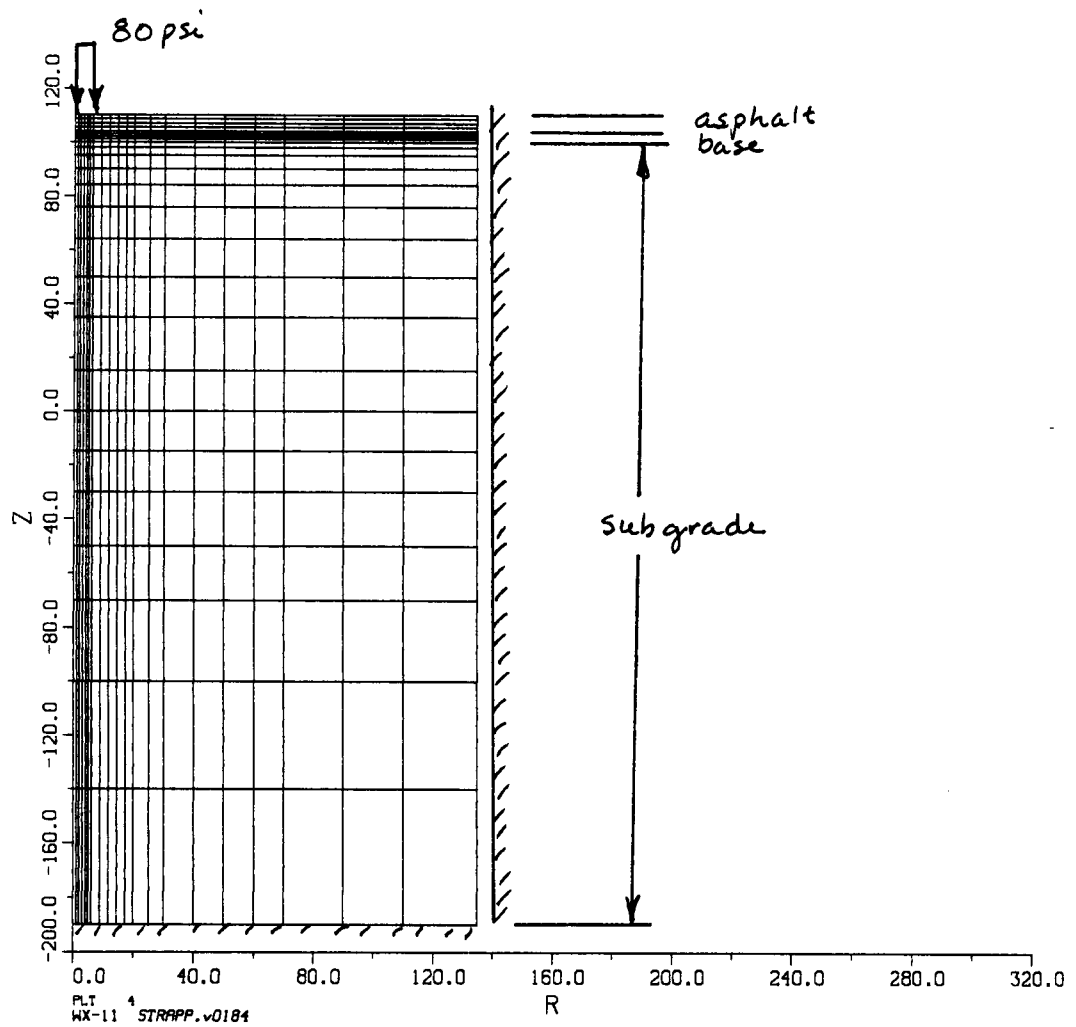


Figure 42 Finite Element Mesh Density for Rutledge Pike 300 Inch Deep Model

Table 1 Cases Analyzed in Rutledge Pike Analyses

Cylinder	Case 1	Case 2
A	two layers asphalt base layer 4 inches subgrade 100 inches	six layers asphalt base layer 4 inches subgrade 100 inches
B	two layers asphalt base layer 4 inches subgrade 200 inches	six layers asphalt base layer 4 inches subgrade 200 inches
C	two layers asphalt base layer 4 inches subgrade 290 inches	six layers asphalt base layer 4 inches subgrade 290 inches

Case 1 The first case analyzed consisted of four layers. The top two layers were both asphaltic concrete. The first layer was defined with a resilient modulus of 422,938 psi and the second layer was defined with a resilient modulus

of 302,975 psi. Both asphalt layers were described with a Poisson's ratio of 0.4 and had thicknesses of three inches. Layer three was a base material described by setting the values of K and n in Equation 1 equal to 4,540 and 0.45, respectively. The base layer had a thickness of four inches. The fourth layer, the subgrade layer, was defined using Equation 6. The constants E_0 , A, and B were specified as 14,000, 3.7, and -240, respectively. The subgrade layer thickness varied with the three models used between 100 inches thick to 290 inches thick. The resulting axial deflection profile of the asphalt surface, three inches below the surface, the base surface, and the subgrade surface are indicated in Figures 43 through 46. Figure 47 illustrates the radial strain at the asphalt base. Figure 48 shows the axial strain at the subgrade surface. Figure 49 pictures the subgrade surface deviator stress.

Case 2 The second case analyzed was made up of eight layers of material. The first six layers were asphaltic concrete. Table 2 lists the resilient moduli used in the description of each layer of asphalt.

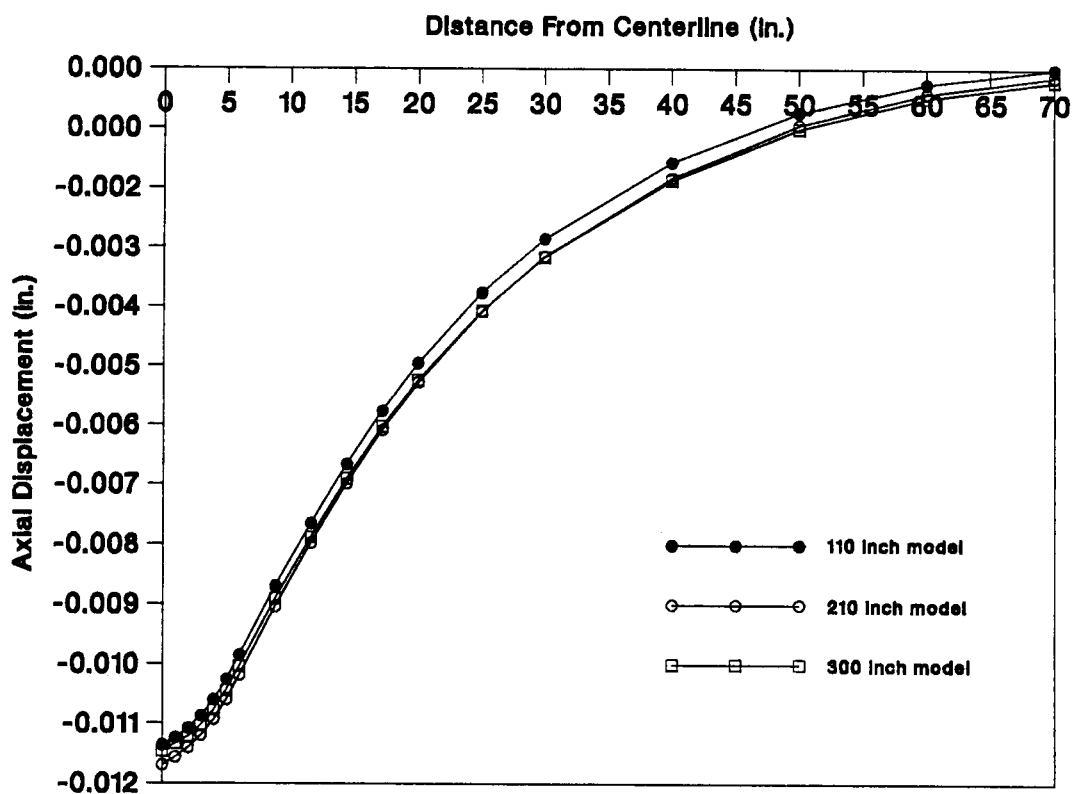


Figure 43 Surface Deflection of Cylinder— Rutledge Pike Case 1

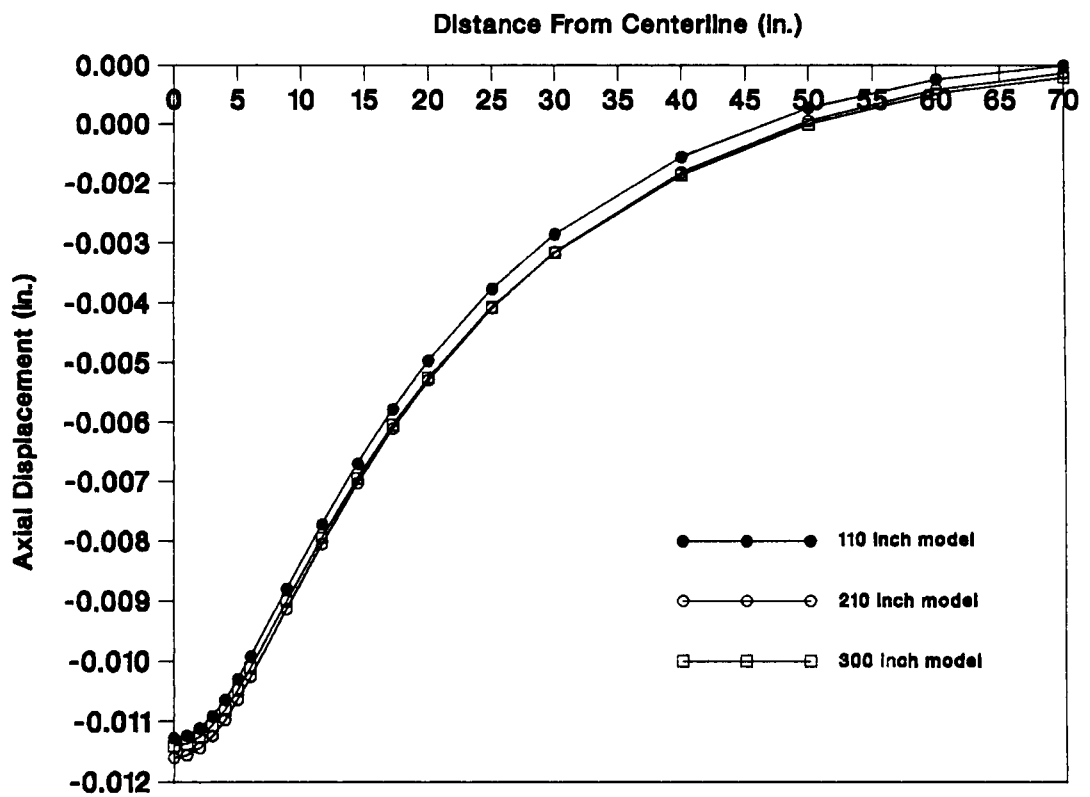


Figure 44 Deflection Three Inches Below Surface— Rutledge Pike Case 1

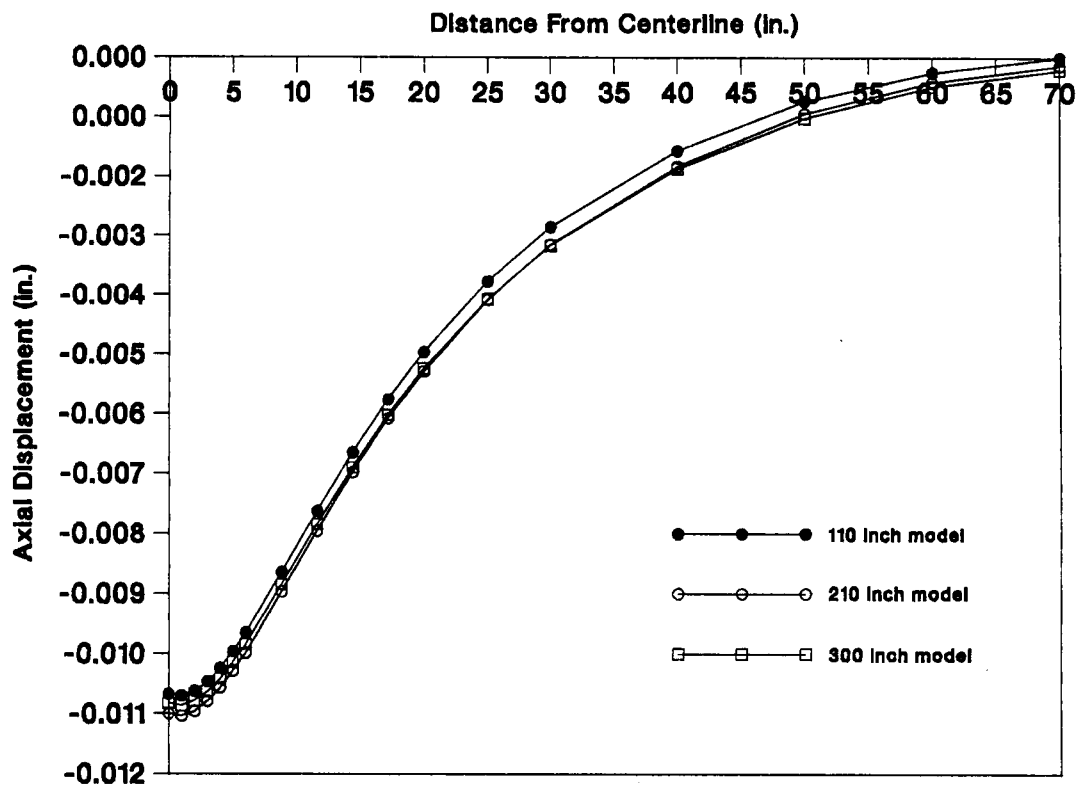


Figure 45 Deflection of Base Surface— Rutledge Pike Case 1

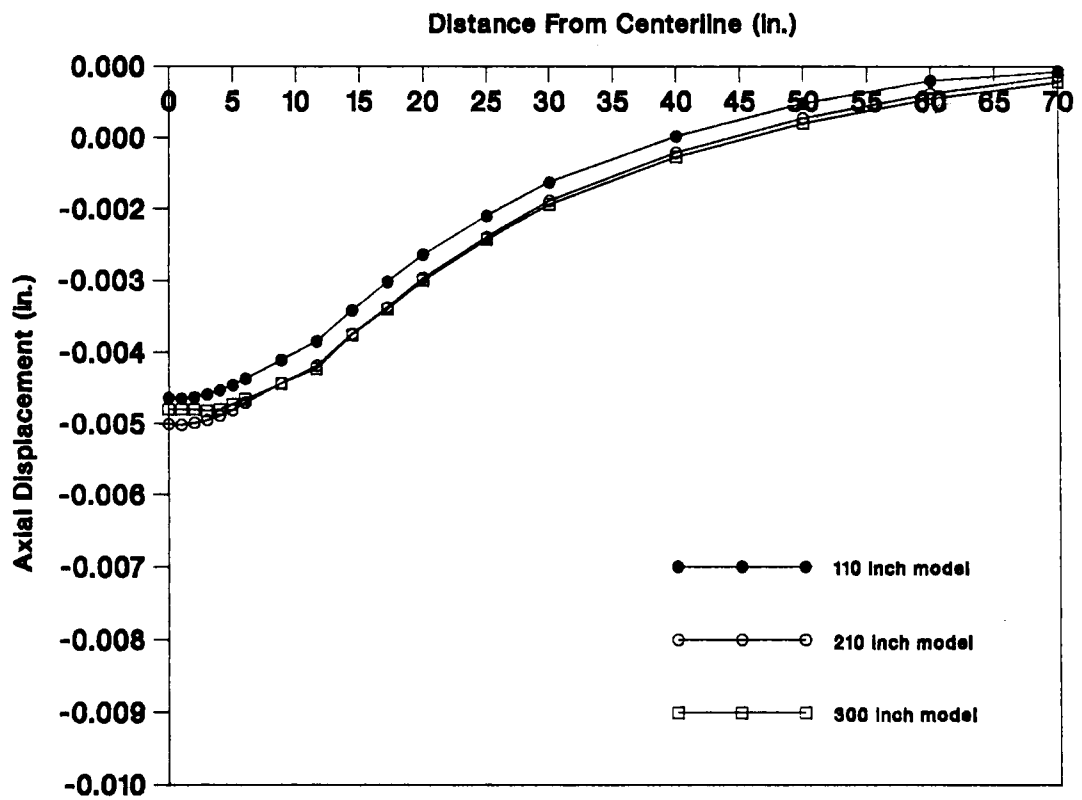


Figure 46 Deflection of Subgrade Surface— Rutledge Pike Case 1

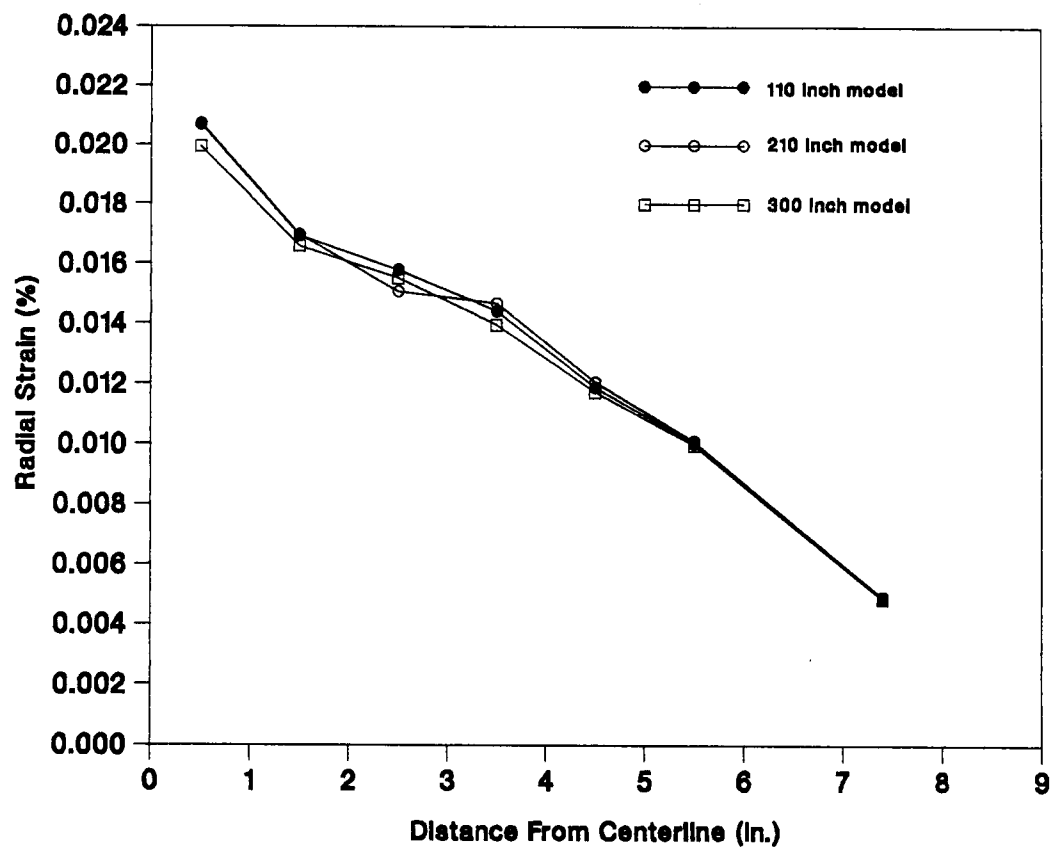


Figure 47 Radial Strain at Asphalt Base— Rutledge Pike Case 1

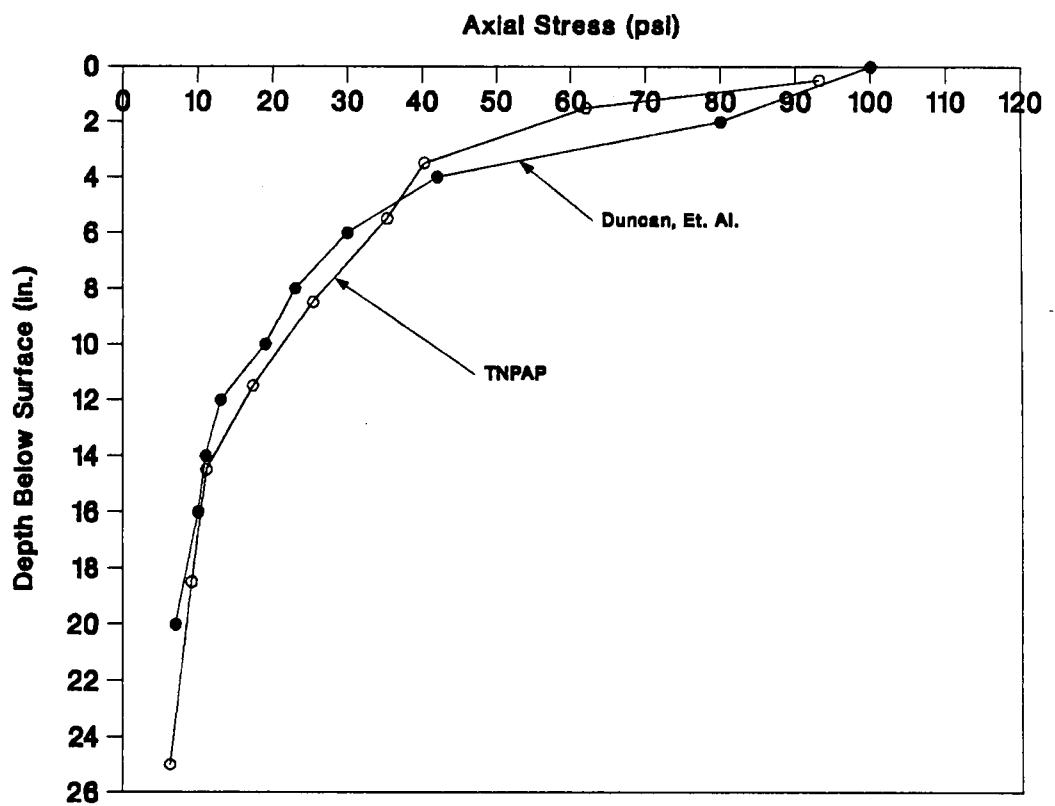


Figure 48 Axial Strain at Subgrade Surface— Rutledge Pike Case 1

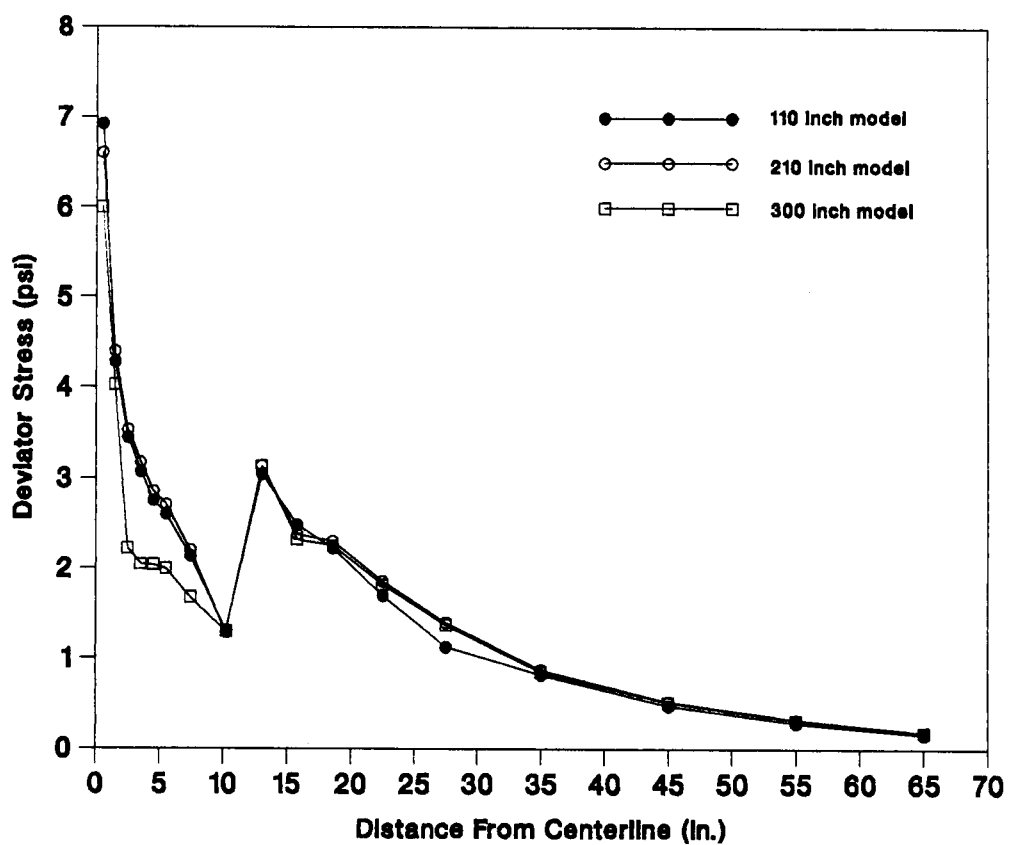


Figure 49 Deviator Stress at Subgrade Surface— Rutledge Pike Case 1

Table 2 Resilient Moduli for Six Layers of Asphalt- Rutledge Pike Case B

Layer	Temperature (°F)	Resilient Modulus (psi)
1	116	223,448
2	112	259,643
3	108	301,701
4	104	300,843
5	101	281,065
6	97	326,592

All six layers of asphalt were defined with Poisson's ratio equal to 0.4 and each layer had a thickness of one inch. Layers seven and eight were base and subgrade materials, respectively, as described in case A. The results for this case are presented in Figures 50 through 56.

These analyses indicated that the the sensitivity of subgrade depth on results is small for depths over 200.0 inches. The was a noticeable difference in results obtained with the 110.0 inch thick cylinder and the 210.0 inch thick

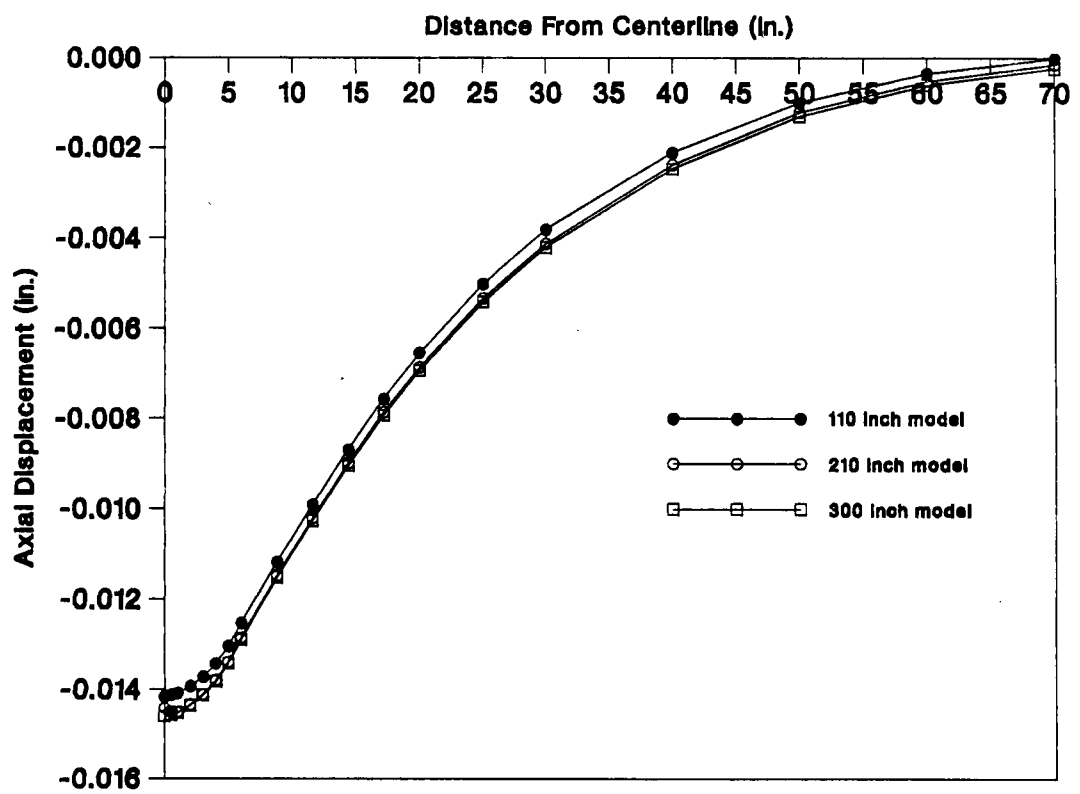


Figure 50 Surface Deflection of Cylinder— Rutledge Pike Case 2

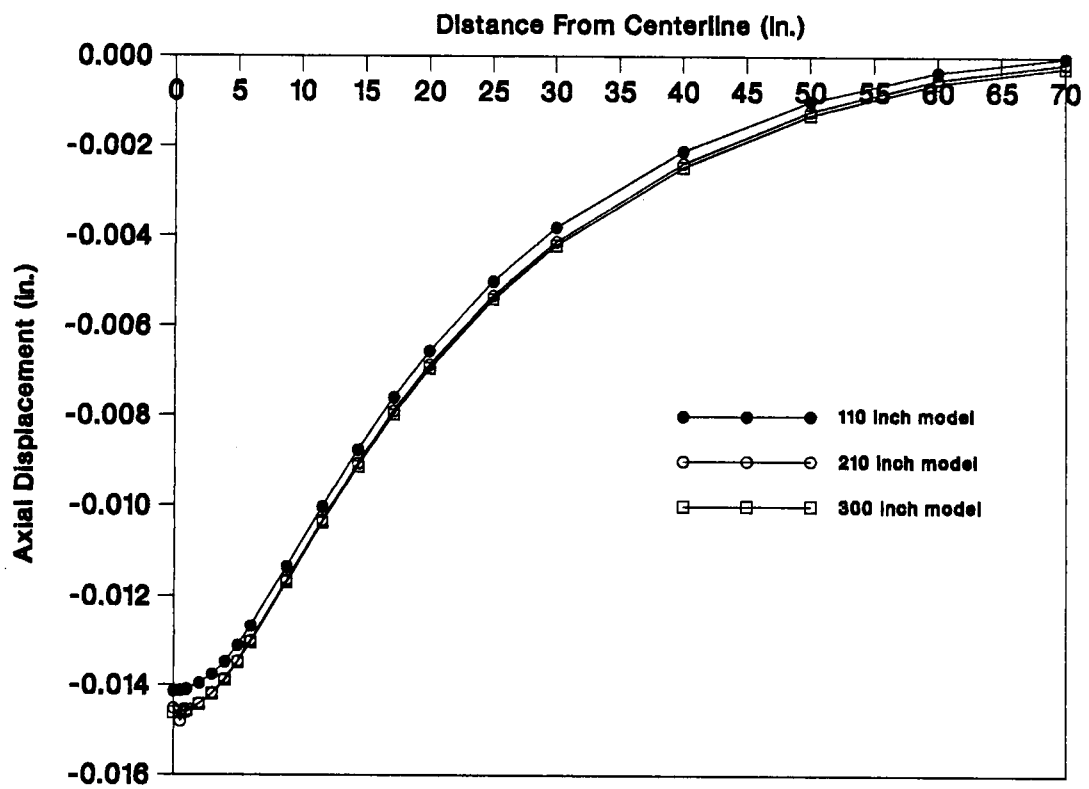


Figure 51 Deflection Three Inches Below Surface— Rutledge Pike Case 2

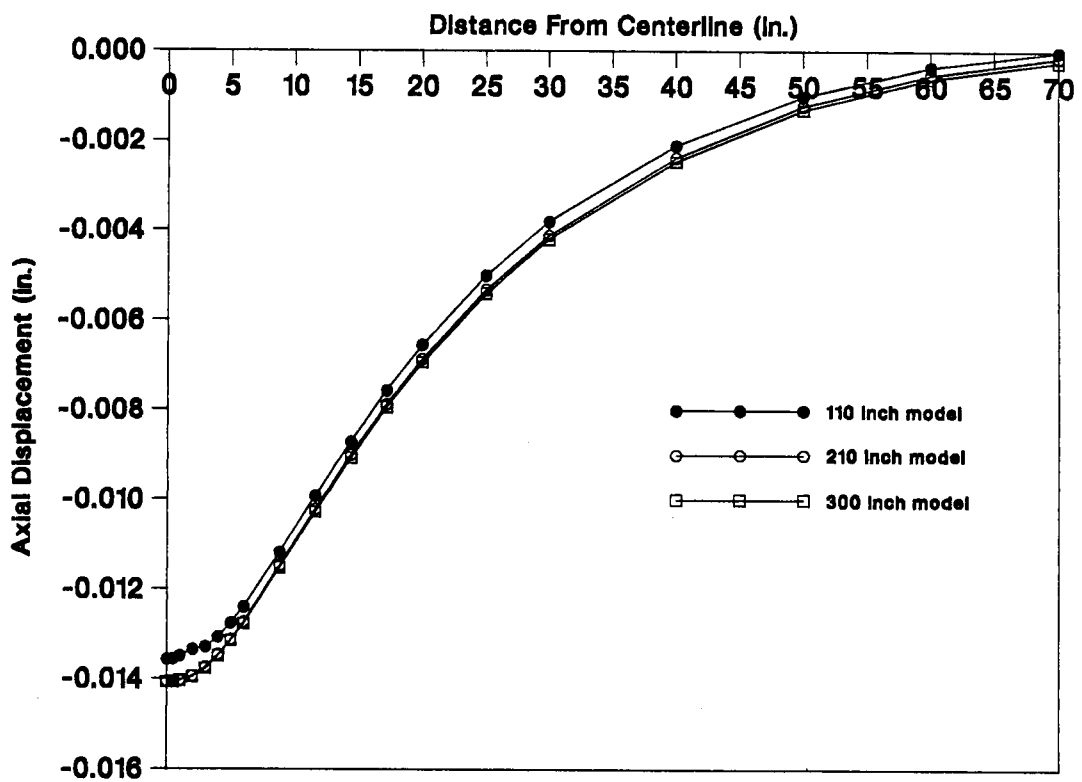


Figure 52 Deflection of Base Surface— Rutledge Pike Case 2

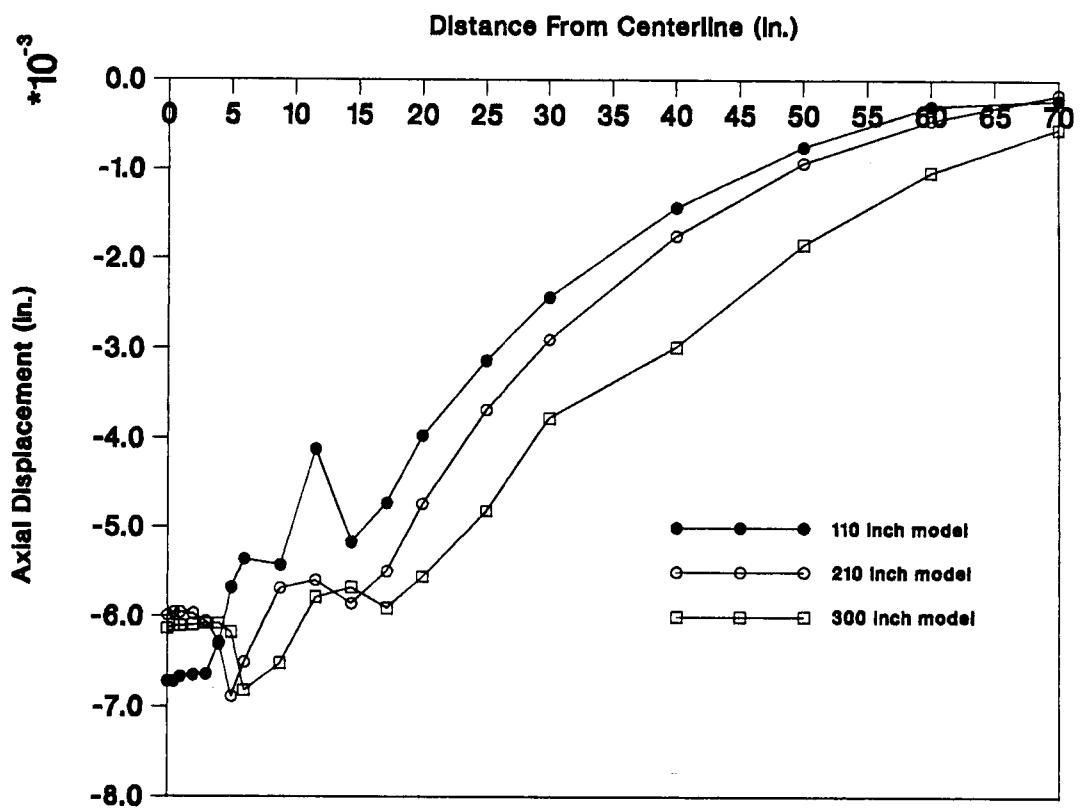


Figure 53 Deflection of Subgrade Surface— Rutledge Pike Case 2

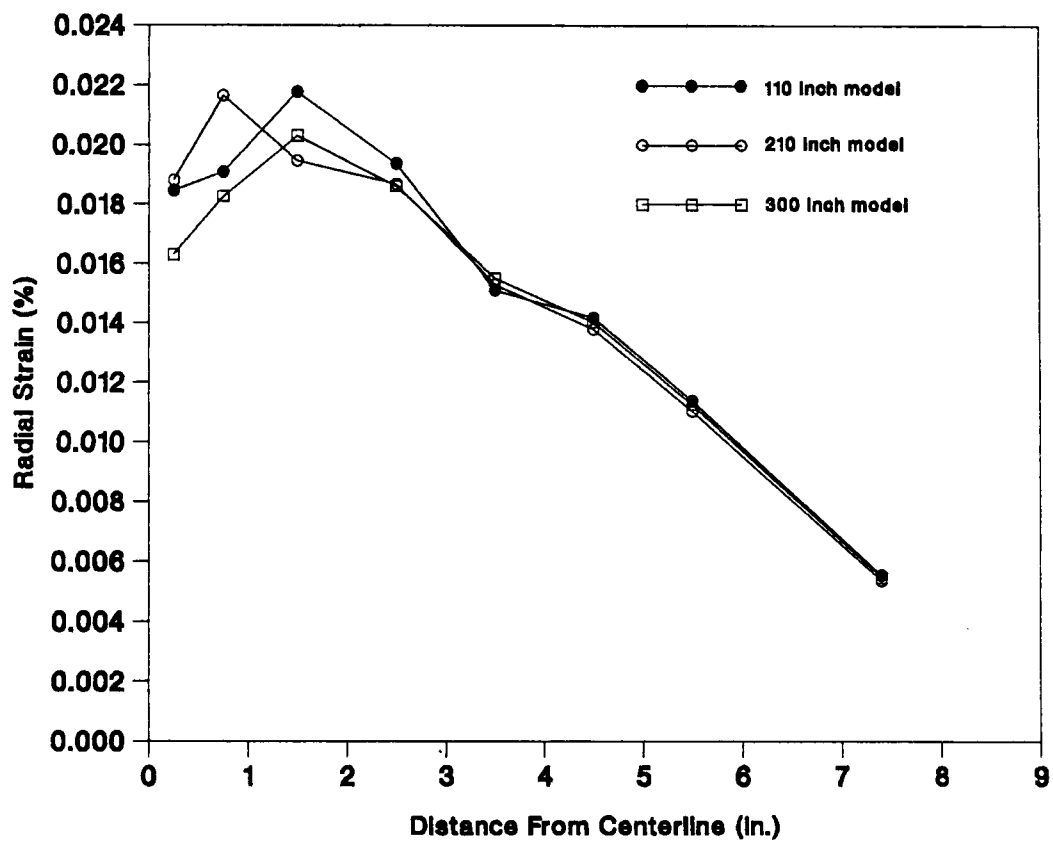


Figure 54 Radial Strain at Asphalt Base— Rutledge Pike Case 2

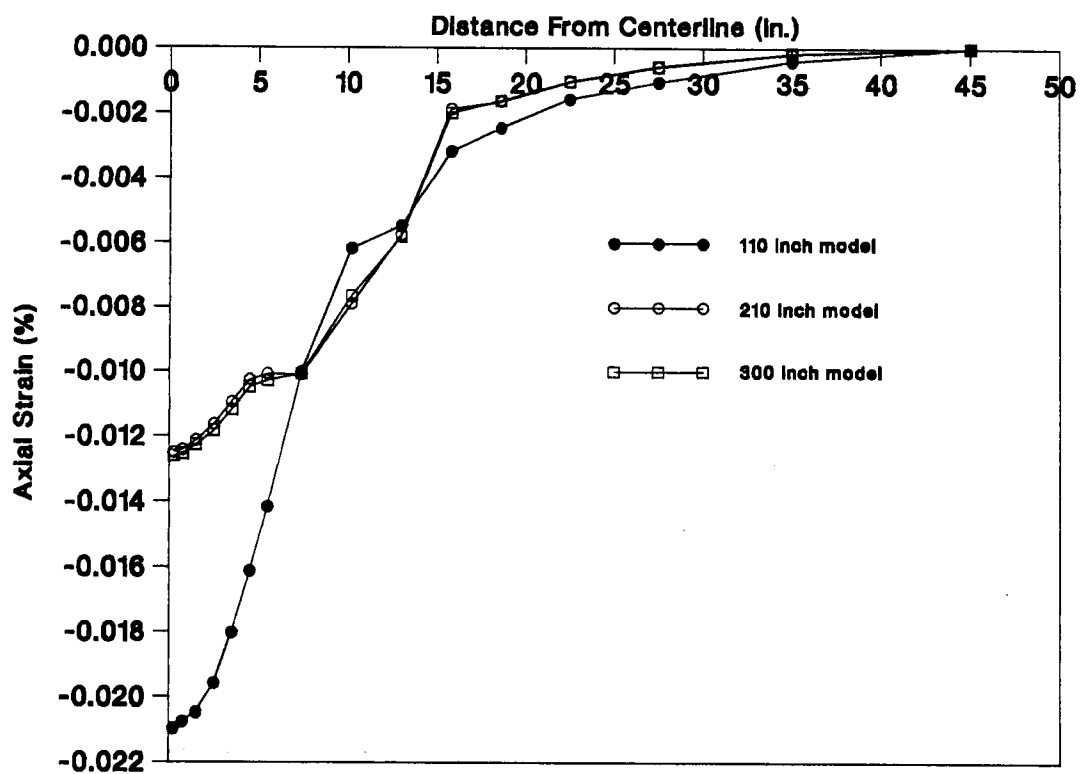


Figure 55 Axial Strain at Subgrade Surface— Rutledge Pike Case 2

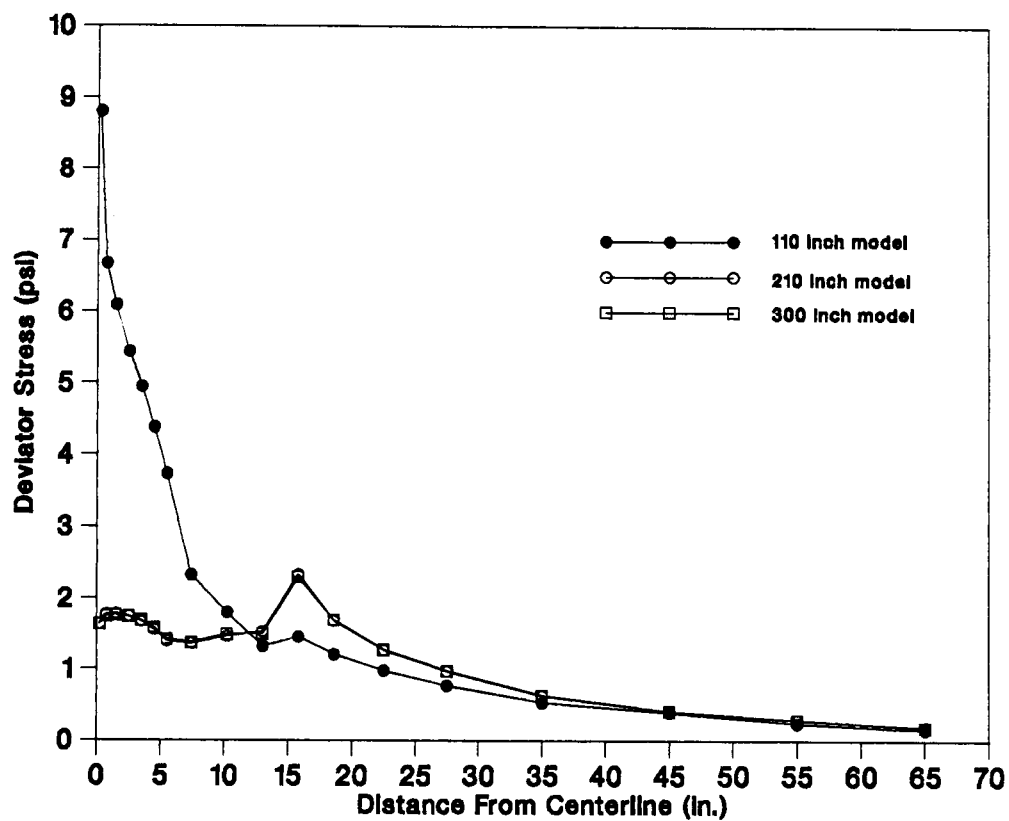


Figure 56 Deviator Stress at Subgrade Surface— Rutledge Pike Case 2

cylinder. However, no appreciable difference occurred between results for the 210.0 inch thick cylinder and the 300.0 inch thick cylinder.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

5.1.1 Linear Analyses

Examination of the results of the linear analyses run with TNPAP and also with the ADINA and ABAQUS finite element analysis codes indicates good agreement exists between results obtained from TNPAP and from existing finite element codes on the open market for linear analyses. Resulting displacements, stresses, and strains all compared well for the sample linear problems run with these codes. Displacements and stresses also compared well in the results from the linear homogeneous and layered systems presented by Duncan, Monismith, and Wilson (Reference 7). Strain data were not available from the problems presented in Reference 7 and, therefore, no comparison of strains could be made.

5.1.2 Nonlinear Analyses

Comparison of the results obtained from analyses of the Gonzales Bypass show good agreement in displacement values obtained by Duncan, Monismith, and Wilson (Reference 7) and those obtained with TNPAP for both the summer

and winter conditions. The axial stress values calculated by TNPAP for the asphalt layer are half that presented in Reference 7 for the Gonzales Bypass. However, the trend of the axial stress as a function of depth below surface is much the same for the TNPAP results as for the results presented in Reference 7. Axial stress values for the base, subbase, and subgrade compare well for both the summer and winter conditions.

Axial displacement values for the surface of the asphalt obtained from the ILLI-PAVE sample problems again show good agreement with values obtained from TNPAP. It is hard to judge the comparison between subgrade surface deflection, radial strain at the asphalt base, and axial strain at the subgrade surface because Reference 5 gives single data points for these parameters and no trend can be established for a curve.

5.1.3 Nodal Coordinate Sensitivity and Aspect Ratio

It was discovered during the course of this research that the finite element analysis code TNPAP is very sensitive to the definition of nodal point coordinates in the input deck. For example, an input deck which described a cylinder which ranged from an axial coordinate location of 110 inches to -190 inches would run to completion in TNPAP. However, a model with identical

mesh density that ranged from an axial coordinate location of 300 inches to 0 inches would fail to converge.

It was also discovered that an aspect ratio of 1.0 for the elements under the loading applied to the model was absolutely essential for successful completion of a run in TNPAP. Failure to specify elements with an aspect ratio of 1.0 often would result in inaccurate results from the model or unsuccessful completion of the analysis.

5.2 Recommendations

TNPAP runs fairly quickly and efficiently. The following enhancements, however, would greatly expand the capabilities of the code.

5.2.1 Temperature Calculations

A useful facility to include in TNPAP would be the capability to supply data for temperature as a function of depth for the asphalt layer of a model. From these data points, the code would interpolate an average temperature for each element in the asphalt layer and calculate the corresponding resilient modulus. This would result in a more accurate description of the material properties for this layer.

5.2.2 Element Types

As TNPAP is now written, only axisymmetric triangular elements can be used for analysis. Inclusion of an axisymmetric quadrilateral element, and a plane strain quadrilateral element would greatly expand the capabilities of TNPAP.

5.2.3 Refinement of Iteration Procedure

The iteration procedure now being used in TNPAP is dependent on the convergence of the displacements of the mesh being analyzed. A much more accurate iterative scheme would involve convergence of the stresses calculated. It is suggested that the iteration procedure be refined to consider stress values rather than displacements. In addition, the stiffness matrix is reformed after each iteration. Computation time would be reduced if the stiffness matrix were reformed only at the beginning of each load increment.

LIST OF REFERENCES

LIST OF REFERENCES

1. Weinrib, J., "Finite Elements on the PC: No Excuses Left," Mechanical Engineering, Volume 108, Number 7, July, 1986, pp. 36-40.
2. Tayabji, S.D. and Colley, B.E., "Improved Rigid Pavement Joints," Prepared for: U.S. Department of Transportation Federal Highway Administration, Report No. FHWA/RD-86/040, Construction Technology Laboratories, Skokie, Illinois, February, 1984.
3. Darter, M.I., "Design of Zero-Maintenance Plain Jointed Concrete Pavement, Vol. I- Development of Design Procedures," Prepared for: Federal Highway Administration United States Department of Transportation, Report No. FHWA-RD-77-111, Department of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, June 8, 1977.
4. Mamlouk, M.S., Yuan, Y.M., Tseng, N.T., and Lee, G.C., "Analysis of Nonlinear Behavior of Asphalt Concrete During Marshall Test", Journal of Testing and Evaluation, Vol. 11, No. 5, September, 1983, pp. 327-332.
5. Elliott, R.P. and Thompson, M.R., "Mechanistic Design Concepts for Conventional Flexible Pavements," Prepared for: Illinois Cooperative Highway and Transportation Research Program, University of Illinois, Urbana, Illinois, June, 1984.
6. Figueroa, J.L., "Resilient Based Flexible Pavement Design Procedure for Secondary Roads," Thesis Presented to the University of Illinois, at Urbana, Illinois, in 1979, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy.
7. Duncan, J.M., Monismith, C.L., and Wilson, E.L., "Finite Element Analysis of Pavements," Committee on Mechanics of Earth Masses and Layered Systems 47th Annual Meeting, 1968, pp. 18-33.
8. Raad, L. and Figueroa, J.L., "Load Response of Transportation Support Systems," Transportation Engineering Journal, Volume 106, Number TE1, January, 1980, pp. 111-129.

9. Taute, A., McCullough, B.F., and Hudson, W.R., "Improvements to the Materials Characterization and Fatigue Life Prediction Methods for the Texas Rigid Pavement Overlay Design Procedure", Prepared for: Texas State Department of Highway and Public Transportation, Transportation Planning Division, Report Number FHWA/TX-81/30+249-1, Center for Transportation Research, The University of Texas at Austin, Austin, Texas, November, 1981.
10. Brown, S.F., and Pappin, J.W., "Modeling of Granular Materials in Pavements," Analysis and Testing of Granular Bases and Subbases: Transportation Research Record 1022, Washington, D.C., 1985, pp. 45-51.
11. Thompson, M.R., Barenberg, E.J., Ioannides, A.M., and Fischer, J.A., "Development of a Stress-Dependent Finite Element Slab Model," Prepared for: Air Force Office of Scientific Research, Report Number AFOSR-TR-83-1061, University of Illinois at Urbana-Champaign, Urbana, Illinois, May, 1983.
12. Figueroa, J.L. and Thompson, M.R., "Simplified Structural Analysis of Flexible Pavements for Secondary Roads Based on ILLI-PAVE," Transportation Research Record 766, 1980, pp. 5-10.
13. Thompson, M.R., "Important Properties of Base and Subgrade Materials," paper presented at Conference on Crushed Stone for Road and Street Construction and Reconstruction, Arlington, Virginia, June 14-15, 1984, pp. IV-1 - IV-41.
14. Allen, J.J., "The Effects of Non-Constant Lateral Pressures on the Resilient Response of Granular Materials," Thesis Presented to the University of Illinois, at Urbana, Illinois, in 1973, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy
15. Allen, J.J. and Thompson, M.R., "Significance of Variable Confined Triaxial Testing," Proceedings, ASCE, TE 4, November, 1974.
16. Stoneking, J.E., Baker, A.J., Levine, D.L., and Soliman, M.O., "Implementation of the Unified Plasticity- Creep Constitutive Equations with the Finite Element Method," Report Number ORNL-Sub-19Y-55455V/1, Prepared for: the Oak Ridge National Laboratory The University of Tennessee, Knoxville, Tennessee, April 12, 1979.

17. Crawford, J.E. and Katona, M.G., "State fo the Art for Prediction of Pavement Response," Report Number FAA-RD-75-183, Civil Engineering Laboratory, Naval Construction Battalion Center, California, September, 1975.
18. Utku, S., "Explicit Expressions for Triangular Torus Element Stiffness Matrix," AIAA Journal, Volume 6, Number 6, June, 1968, pp. 1174-1176.
19. Forrest, J., "Development of Constitutive Equations for Pavement Components," Report Number AFWL-TR-70-165 VOL I, Naval Civil Engineering Laboratory, New Mexico, April, 1972.
20. Hibbit, Karlson, and Sorenson, ABAQUS User's Manual, Hibbit, Karlson, and Sorenson, Inc., 1982.
21. Bathe, K.J., ADINA A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis, Massachusetts Institute of Technology, Massachusetts, 1975.
22. Timoshenko, S.P. and Goodier, J.N., Theory of Elasticity, McGraw-Hill Book Company, New York, 1951
23. Huebner, K.H., The Finite Element Method for Engineers, John Wiley and Sons, Inc., New York, 1975
24. Frederick, D. and Chang, T.S., Continuum Mechanics, Scientific Publishers, Inc. Massachusetts, 1965.
25. Timoshenko, S.P. and Gere, J.M., Mechanics of Materials, Litton Educational Publishing, Inc., 1972.
26. Chen, W.F. and Saleeb, A.F., Constitutive Equations for Engineering Materials- Volume 1: Elasticity and Modeling, John Wiley and Sons, Inc., New York, 1982.
27. Mendelson, A., Plasticity: Theory and Application, The MacMillan Company, New York, 1968.

APPENDIXES

APPENDIX A

```

1      PROGRAM TNPAF
2      COMMON/SOL/NUMNP,NEQ,NWK,NUMEST,MIDEST,MAXEST,MK
3      COMMON/DIM/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
4      COMMON/EL/IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
5      COMMON/VAR/NG,MODEX,NLOAD
6      COMMON/TAPES/IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
7      COMMON/EXMU/PROP(10,10)
8      COMMON/NULOAD/NULOAD
9      COMMON/NELG/NELG
10     COMMON/PSTRES/STPROP(720),TDISP(1500),TEPS(4,720),TSIG(4,720),
11     + PRSTOR(1050),EPSTOR(4,1050),SGSTOR(4,1050),LCOUNT,IELCNT,
12     + JTOTEL,JELCNT,ITER,NUMIT
13     COMMON /CONV/ DOLD(1500),DNEW(1500),ICONV
14     COMMON /IFIX/ IFIX
15 C
16     REAL HED(20)
17     INTEGER=4 IA(1)
18     DIMENSION TIM(6)
19     EQUIVALENCE (A(1),IA(1)),(NPAR(2),NUMEL)
20 C
21     COMMON A(59601)
22     MTOT=59600
23 C     CALL ERRSET(0)
24 C
25     ITWO=1
26     ICONV=0
27 C
28 C     OPEN(UNIT=40,DEVICE='DSK',ACCESS='SEQIN',FILE='INPUT.SAP')
29     OPEN (UNIT=40,NAME='INPUT.SAP',STATUS='OLD')
30 C     TYPE 1001
31 C1001    FORMAT(/,' OUTPUT TO DSK OR LPT ? ',5)
32 C     ACCEPT 1002,PRIN
33 C1002    FORMAT(A3)
34 C     OPEN(UNIT=50,DEVICE=PRIN,ACCESS='SEQOUT',FILE='OUTPUT.SAP')
35     OPEN (UNIT=50,NAME='OUTPUT.SAP',STATUS='NEW')
36     IELMNT = 20
37     ILOAD = 30
38     ILOD2 = 27
39     IIN = 40
40     IOUT = 50
41     ISTRES=25
42     IDISP=35
43     IEPS=24
44     ISIG=23
45     NULOAD=1
46 C
47 200     NUMEST=0
48         MAXEST=0
49         CALL SECOND (TIM(1))
50 C
51 C
52 C
53 C     READ CONTROL INFORMATION
54 C
55 C
56     READ (IIN,1000) HED,NUMNP,NUMEG,NLCASE,MODEX,NUMIT,IFIX
57     IF (NUMNP.EQ.0) STOP
58     IF (NUMIT.EQ.0) NUMIT=1
59 C     WRITE (IOUT,2000) HED,NUMNP,NUMEG,NLCASE,MODEX
60 C
61 C
62 C     READ NODAL POINT DATA
63 C
64 C
65     N1= 1
66     N2=N1 + 3 * NUMNP
67     N3=N2 + NUMNP * ITWO
68     N4=N3 + NUMNP*ITWO
69     N5=N4 + NUMNP*ITWO
70     IF (N5.GT.MTOT) CALL ERROR (N5 - MTOT,1)
71 C
72     CALL INPUT (A(N1),A(N2),A(N3),A(N4),NUMNP,NEQ)
73 C     OPEN (UNIT=35,DEVICE='DSK',ACCESS='SEQOUT',FILE='DISP.SAP')
74     OPEN (UNIT=35,NAME='DISP.SAP',STATUS='NEW',FORM='UNFORMATTED')
75     REWIND IDISP
76     DO 452 IW=1,NEQ
77     TDISP(IW)=0.0
78 452 CONTINUE
79     WRITE (IDISP) (TDISP(IW),IW=1,NEQ)
80 C     CLOSE (UNIT=35,DEVICE='DSK',ACCESS='SEQOUT',FILE='DISP.SAP')

```

```

81 C      CLOSE (UNIT=35)
82 C
83 C      NEO1=NEO + 1
84 C
85 C      CALCULATE AND STORE LOAD VECTORS
86 C
87 C
88 C
89 C      N6=N5 + NEO*ITWO
90 C      WRITE(IOUT,2005)
91 C
92 C      OPEN(UNIT=30,DEVICE='DSK',ACCESS='SEQOUT',FILE='ILOAD.SAP')
93 C      OPEN (UNIT=30,NAME='ILOAD.SAP',STATUS='UNKNOWN',
94 C      + FORM='UNFORMATTED')
95 C      OPEN (UNIT=30,NAME='ILOAD2.SAP',STATUS='UNKNOWN'
96 C      + FORM='UNFORMATTED')
97 C      REWIND ILOAD
98 C      DO 300 L=1,NLCASE
99 C
100 C      READ (IIN,1010) LL,NLOAD
101 C
102 C      WRITE (IOUT,2010) LL,NLOAD
103 C      IF (LL.EQ.L) GO TO 310
104 C      WRITE (IOUT,2020)
105 C      STOP
106 C 310      CONTINUE
107 C
108 C      N7=N6 + NLOAD
109 C      N8=N7 + NLOAD
110 C      N9=N8 + NLOAD*ITWO
111 C
112 C      IF (N9.GT.MTOT) CALL ERROR (N9 - MTOT,2)
113 C
114 C      IF(NLOAD .EQ. 0) GO TO 299
115 C      CALL LOADS (A(N5),A(N6),A(N7),A(N8),A(N1),NEO)
116 C 299      CONTINUE
117 C
118 C 300      CONTINUE
119 C      CLOSE(UNIT=30,DEVICE='DSK',ACCESS='SEQOUT',FILE='ILOAD.SAP')
120 C      CLOSE (UNIT=30)
121 C
122 C
123 C      READ, GENERATE AND STORE
124 C      ELEMENT DATA
125 C
126 C      CLEAR STORAGE
127 C
128 C      N6=N5 + NEO
129 C      DO 10 I=N5,N6
130 C 10      IA(I)=0
131 C      IND=1
132 C
133 C      JTOTEL=0
134 C      CALL ELCAL
135 C
136 C      CALL SECOND (TIM(2))
137 C
138 C
139 C      *** SOLUTION PHASE ***
140 C
141 C
142 C
143 C      ASSEMBLE STIFFNESS MATRIX
144 C
145 C
146 C      CALL ADDRES (A(N2),A(N5))
147 C
148 C      MM=NWK/NEO
149 C      N3=N2 + NEO + 1
150 C      N4=N3 + NWK*ITWO
151 C      N5=N4 + NEO*ITWO
152 C      N6=N5 + MAXEST
153 C      IF (N6.GT.MTOT) CALL ERROR (N6 - MTOT,4)
154 C
155 C      WRITE TOTAL SYSTEM DATA
156 C
157 C      WRITE(IOUT,2025) NEO,NWK,MM,MM
158 C
159 C      IN DATA CHECK ONLY MODE WE SKIP ALL FURTHER CALCULATIONS
160 C

```

```

161      IF (MODEX.GT.0) GO TO 100
162      CALL SECOND (TIM(3))
163      CALL SECOND (TIM(4))
164      CALL SECOND (TIM(5))
165      GO TO 120
166 C
167 C      CLEAR STORAGE
168 C
169 100     >NNL=NWK + NEO
170      CALL CLEAR(A(N3),NNL)
171 C
172 C
173 C      OPEN (UNIT=30,DEVICE='DSK',ACCESS='SEQIN',FILE='ILOAD.SAP')
174 C      OPEN (UNIT=25,DEVICE='DSK',ACCESS='SEQINOUT',FILE='PROST.SAP')
175 C      OPEN (UNIT=24,DEVICE='DSK',ACCESS='SEQINOUT',FILE='TOTSTN.SAP')
176 C      OPEN (UNIT=23,DEVICE='DSK',ACCESS='SEQINOUT',FILE='TOTSTS.SAP')
177 C      OPEN (UNIT=30,NAME='ILOAD.SAP',STATUS='UNKNOWN',
178 C      +      FORM='UNFORMATTED')
179 C      OPEN (UNIT=25,NAME='PROST.SAP',STATUS='NEW',FORM='UNFORMATTED')
180 C      OPEN (UNIT=24,NAME='TOTSTN.SAP',STATUS='NEW',FORM='UNFORMATTED')
181 C      OPEN (UNIT=23,NAME='TOTSTS.SAP',STATUS='NEW',FORM='UNFORMATTED')
182      REWIND ILOAD
183      REWIND ISTRES
184      REWIND IEPS
185      REWIND ISIG
186      DO 400 L=1,NLCASE
187      ITER = 0
188      LCOUNT=L
189 401      IND = 2
190      ITER = ITER + 1
191      CALL CLEAR (A(N3),NNL)
192 C
193      CALL ASSEM (A(N5))
194 C
195      CALL SECOND (TIM(3))
196 C
197 C      TRIANGULARIZE STIFFNESS MATRIX
198 C
199      KTR=1
200      CALL COLSOL (A(N3),A(N4),A(N2),NEO,NWK,NEQ1,KTR,ITER)
201 C
202 35      CALL SECOND (TIM(4))
203 C
204      KTR=2
205      IND=3
206 C
207 C
208      IF (ITER.GT.1) REWIND ILOD2
209      CALL LOADV (A(N4),NEQ,ITER)
210 C
211 C
212 C      CALCULATION OF DISPLACEMENTS
213 C
214 C      OPEN (UNIT=35,DEVICE='DSK',ACCESS='SEQIN',FILE='DISP.SAP')
215 C      OPEN (UNIT=35,NAME='DISP.SAP',STATUS='OLD',FORM='UNFORMATTED')
216      REWIND IDISP
217      READ (IDISP) (TDISP(IR),IR=1,NEQ)
218 C      CLOSE (UNIT=35,DEVICE='DSK',ACCESS='SEQIN',FILE='DISP.SAP')
219 C      CLOSE (UNIT=35)
220 C
221      CALL COLSOL (A(N3),A(N4),A(N2),NEO,NWK,NEQ1,KTR)
222 C
223 C      IF (ITER.LT.NUMIT.AND.ICONV.EQ.0) GO TO 407
224      WRITE(IOUT,2015) L,ITER
225 407 CONTINUE
226      CALL WRITE (A(N4),A(N1),NEO,NUMNP,TDISP,ITER,NUMIT,L)
227 C
228 C
229 C      CALCULATION OF STRLSSES
230 C
231 C
232      CALL STRESS (A(N5))
233      IF (ITER.LT.NUMIT.AND.ICONV.EQ.0) GO TO 401
234 C
235 400      CONTINUE
236 C      CLOSE(UNIT=30,DEVICE='DSK',ACCESS='SEQIN',FILE='ILOAD.SAP')
237 C      CLOSE(UNIT=25,DEVICE='DSK',ACCESS='SEQINOUT',FILE='PROST.SAP')
238 C      CLOSE(UNIT=24,DEVICE='DSK',ACCESS='SEQINOUT',FILE='TOTSTN.SAP')
239 C      CLOSE(UNIT=23,DEVICE='DSK',ACCESS='SEQINOUT',FILE='TOTSTS.SAP')
240 C      CLOSE (UNIT=30)

```

```

241 C      CLOSE (UNIT=25)
242 C      CLOSE (UNIT=24)
243 C      CLOSE (UNIT=23)
244 C
245          CALL SECOND (TIM(5))
246          IJUMP=1
247          IF (IJUMP.EQ.1) GO TO 501
248 C-----
249 C
250 C      END OF ELASTIC SOLUTION : BEGIN CREEP-PLASTIC ANALYSIS.
251 C
252 C-----
253          N6A=N6+(6*NUMEL*ITWO)
254          N6B=N6A+(4*NUMEL*ITWO)
255          N6C=N6B+(4*NUMEL*ITWO)
256          N6D=N6C+(4*NUMEL*ITWO)
257          N6E=N6D+(4*NUMEL*ITWO)
258          N6F=N6E+(4*NUMEL*ITWO)
259          IF(N6F.GT.MTOT) CALL ERROR(N6F-MTOT,5)
260          CALL CREEPL(A(N1),A(N2),A(N3),A(N4),A(N6),A(N6A),
261 1      A(N6B),A(N6C),A(N6D),A(N6E),NWK,NUMEL,NUMNP,NEQ)
262          CALL SECOND(TIM(6))
263 C
264 C
265 C
266 C      PRINT SOLUTION TIMES
267 C
268 120      TT=0.
269          DO 500 I=1,5
270              TIM(I)=TIM(I+1) - TIM(I)
271 500      TT=TT + TIM(I)
272 C          WRITE (IOUT,2030) HED,(TIM(I),I=1,5),TT
273 C
274 C      READ NEXT ANALYSIS CASE
275 C      TO DO THIS DELETE THE FOLLOWING IF STATEMENT
276 C
277          IF(NULOAD.EQ.1) GO TO 1999
278          IF(MODEX .EQ. 0) STOP
279 1999      CONTINUE
280          NULOAD=NULOAD+1
281          GO TO 200
282 C
283 501 CONTINUE
284      STOP
285 1000      FORMAT (20A4/6I5)
286 1010      FORMAT (2I5)
287 C
288 C
289 2000      FORMAT (1H1,20A4 ///
290          155H C O N T R O L I N F O R M A T I O N                ///5X,
291          255HNUMBER OF MODAL POINTS . . . . . (NUMNP) =.15//5X,
292          355HNUMBER OF ELEMENT GROUPS . . . . . (NUMEG) =.15//5X,
293          455HNUMBER OF LOAD CASES . . . . . (NLCASE) =.15//5X,
294          555HSOLUTION MODE . . . . . (MODEX) =.15//5X,
295          655H   EQ.O. DATA CHECK                /5X,
296          755H   EQ.1, EXECUTION                    )
297 2005      FORMAT (1H1,20H L O A D C A S E D A T A)
298 2010      FORMAT (///4X,33H LOAD CASE NUMBER . . . . . =.15//5X,
299          132HNUMBER OF CONCENTRATED LOADS . =.15)
300 2015      FORMAT(1H1,9HLOAD CASE,13,' ITERATION NUMBER',13)
301 2020      FORMAT (1X,40H*** ERROR LOAD CASES ARE NOT IN ORDER )
302 2025      FORMAT (1H1,
303          155HTOTAL SYSTEM DATA                ///5X,
304          255HNUMBER OF EQUATIONS . . . . . (NEQ) =.15//5X,
305          355HNUMBER OF MATRIX ELEMENTS . . . . . (NWK) =.15//5X,
306          455HMAXIMUM HALF BANDWIDTH . . . . . (MK) =.15//5X,
307          555HMEAN HALF BANDWIDTH . . . . . (MM) =.15)
308 2030      FORMAT (1H1,48H S O L U T I O N   T I M E   L O G   I N   S E C //
309          112X,11HFOR PROBLEM//1X,20A4 ///5X,
310          251HTIME FOR INPUT PHASE                =.F12.5//5X,
311          351HTIME FOR CALCULATION OF STRUCTURE STIFFNESS MATRIX=.F12.5//5X,
312          451HTRIANGULARIZATION OF STIFFNESS MATRIX . . . . . =.F12.5//5X,
313          551HTIME FOR LOAD CASE SOLUTION . . . . . =.F12.5//5X,
314          651HTIME FOR CREEP-PLASTIC SOLUTION . . . . . =.F12.5//5X,
315          751H T O T A L   S O L U T I O N   T I M E . . . . . =.F12.5)
316 C
317          END
318          SUBROUTINE ERROR(N,I)
319 C
320 C

```

```

321 C      PROGRAM
322 C      TO PRINT MESSAGES WHEN HIGH-SPEED STORAGE IS EXCEEDED
323 C
324 C
325 C      COMMON/TAPES/IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
326 C
327 C      GO TO (1,2,3,4,5),I
328 C
329 C      1  WRITE(IOUT,2000)
330 C      GO TO 6
331 C      2  WRITE(IOUT,2010)
332 C      GO TO 6
333 C      3  WRITE(IOUT,2020)
334 C      GO TO 6
335 C      4  WRITE(IOUT,2030)
336 C      GO TO 6
337 C      5  WRITE(IOUT,2040)
338 C
339 C      6  WRITE (IOUT,2050) N
340 C      STOP
341 C      2000  FORMAT (//48H NOT ENOUGH STORAGE FOR READ-IN OF ID ARRAY AND
342 C      1,23H NODAL POINT COORDINATES)
343 C      2010  FORMAT (//50H NOT ENOUGH STORAGE FOR DEFINITION OF LOAD VECTORS)
344 C      2020  FORMAT (//42H NOT ENOUGH STORAGE FOR ELEMENT DATA INPUT )
345 C      2030  FORMAT (//53H NOT ENOUGH STORAGE FOR ASSEMBLAGE OF GLOBAL STURCTUR
346 C      1,55H E STIFFNESS, AND DISPLACEMENT AND STRESS SOLUTION PHASE )
347 C      2040  FORMAT (// ' NOT ENOUGH STORAGE FOR CREEP-PLASTIC ANALYSYS
348 C      1  PHASE' )
349 C      2050  FORMAT (// 32H === ERROR   STORAGE EXCEEDED BY, 19)
350 C      RETURN
351 C      END
352 C      SUBROUTINE INPUT (ID,X,Y,Z,NUMNP,NEO)
353 C
354 C
355 C
356 C      PROGRAM
357 C      . TO READ, GENERATE, AND PRINT NODAL POINT INPUT DATA
358 C      . TO CALCULATE EQUATION NUMBERS AND STORE THEM IN ID ARRAY
359 C
360 C      N=ELEMENT NUMBER
361 C      ID=BOUNDARY CONDITION CODES (0=FREE,1=DELETED)
362 C      X,Y,Z=COORDINATES
363 C      KN= GENERATION CODE
364 C      I.E. INCREMENT ON NODAL POINT NUMBER
365 C
366 C
367 C
368 C
369 C
370 C      THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
371 C      CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON IBM
372 C      OR UNIVAC MACHINES.  ACTIVATE, DEACTIVATE, OR ADJUST ABOVE
373 C      CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
374 C
375 C      COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
376 C      DIMENSION X(1),Y(1),Z(1),ID(3,NUMNP)
377 C
378 C
379 C      READ AND GENERATE NODAL POINT DATA
380 C
381 C      WRITE (IOUT,2000)
382 C      WRITE (IOUT,2010)
383 C      WRITE (IOUT,2020)
384 C      KNOLD=0
385 C      NOLD=0
386 C
387 C      10  READ (IIN,1000) N,(ID(I,N),I=1,3),X(N),Y(N),Z(N),KN
388 C      WRITE (IOUT,2030) N,(ID(I,N),I=1,3),X(N),Y(N),Z(N),KN
389 C      IF (KNOLD.EQ.0) GO TO 50
390 C      NUM=(N-NOLD) / KNOLD
391 C      NUMN=NUM-1
392 C      IF (NUMN.LT.1) GO TO 50
393 C      XNUM=NUM
394 C      DX=(X(N)-X(NOLD))/XNUM
395 C      DY=(Y(N)-Y(NOLD))/XNUM
396 C      DZ=(Z(N)-Z(NOLD))/XNUM
397 C      K=NOLD
398 C      DO 30 J=1,NUMN
399 C      KK=K
400 C      K=K + KNOLD

```

```

401      X(K)=X(KK)+DX
402      Y(K)=Y(KK)+DY
403      Z(K)=Z(KK)+DZ
404      DO 30 I=1,3
405      ID(I,K)=ID(I,KK)
406 30    CONTINUE
407 C
408 50    NOLD=N
409      KNOLD=KN
410      IF (N.NE.NUMNP) GO TO 10
411 C
412 C      WRITE COMPLETE NODAL DATA
413 C
414 C      WRITE (IOUT,2015)
415 C      WRITE (IOUT,2020)
416      DO 200 N=1,NUMNP
417 200  WRITE (IOUT,2030) (N,(ID(I,N),I=1,3),X(N),Y(N),Z(N),KN)
418 C
419 C      NUMBER UNKNOWNNS
420 C
421      NEO=0
422      DO 100 N=1,NUMNP
423      DO 100 I=1,3
424      IF (ID(I,N)) 110,120,110
425 120    NEO=NEO + 1
426      ID(I,N)=NEO
427      GO TO 100
428 110    ID(I,N)=0
429 100    CONTINUE
430 C
431 C      WRITE EQUATION NUMBERS
432 C
433      WRITE (IOUT,2040) (N,(ID(I,N),I=1,3),N=1,NUMNP)
434      RETURN
435 C
436 1000  FORMAT (4I5,3F10.0,I5)
437 2000  FORMAT(1H1,33H N O D A L P O I N T D A T A //)
438 2010  FORMAT(18H INPUT NODAL DATA //)
439 2015  FORMAT(///22H GENERATED NODAL DATA //)
440 2020  FORMAT(7H NODE ,9X,8HBOUNDARY,25X,11HNODAL POINT,17X,
441 1      4HMESH/7H NUMBER,5X,16HCONDITION CODES,21X,11HCOORDINATES,14X,
442 2      10HGENERATING/77X,4HCODE//15X,1HX,4X,1HY,4X,1HZ,15X,1HX,12X,1HY
443 3      12X,1HZ,10X,2HKN)
444 2030  FORMAT (15,6X,3I5,6X,3F13.3,3X,I6)
445 2040  FORMAT(//17H EQUATION NUMBERS//,4X,4HNODE,9X,
446 1      17HDEGREE OF FREEDOM/3X,6HNUMBER//,
447 2      5X,1HN,13X,1HX,4X,1HY,4X,1HZ/(1X,I5,9X,3I5))
448 C
449      END
450      SUBROUTINE LOADS (R,NOD,DIRN,FLOAD,ID,NEO)
451 C
452 C
453 C
454 C
455 C      PROGRAM
456 C      . TO READ NODAL LOAD DATA
457 C      . TO CALCULATE THE LOAD VECTOR R FOR EACH LOAD CASE
458 C      . AND WRITE ONTO TAPE ILOAD
459 C
460 C
461 C
462 C
463 C
464 C      THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC
465 C      ON CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON
466 C      IBM OR UNIVAC MACHINES. ACTIVATE, DEACTIVATE OR ADJUST
467 C      ABOVE CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
468 C
469      COMMON /VAR/ NG,MODEX,NLOAD
470      COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
471      DIMENSION R(NEO),NOD(1),DIRN(1),FLOAD(1)
472      DIMENSION ID(3,1)
473 C
474 C      WRITE (IOUT,2000)
475      READ (IIN,1000) (NOD(I),DIRN(I),FLOAD(I),I=1,NLOAD)
476 C      WRITE (IOUT,2010) (NOD(I),DIRN(I),FLOAD(I),I=1,NLOAD)
477      IF (MODEX.EQ.0) RETURN
478 C
479      DO 210 I=1,NEO
480 210    R(I)=0.

```

```

481 C
482     DO 220 L=1,NLOAD
483     LN=NOD(L)
484     LI=IDIRN(L)
485     II=ID(LI,LN)
486     IF (II) 220,220,240
487 240 R(II)=R(II) + FLOAD(L)
488 C
489 220 CONTINUE
490 C
491     WRITE (ILOAD) R
492 C
493 200 CONTINUE
494 C
495 1000 FORMAT (2I5,F10.0)
496 2000 FORMAT (////4X,30HNODE          DIRECTION          LOAD/
497 1 3X,6HNUMBER,19X,9HMAGNITUDE )
498 2010 FORMAT (1H0,16,9X,14,7X,E12.5)
499     RETURN
500     END
501     SUBROUTINE ELCAL
502 C
503 C
504 C
505 C     PROGRAM
506 C     TO LOOP OVER ALL ELEMENT GROUPS FOR READING,
507 C     GENERATING AND STORING THE ELEMENT DATA
508 C
509 C
510 C
511     COMMON /SOL/ NUMNP,NEQ,NWK,NUMEST,MIDEST,MAXEST,MK
512     COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
513     COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
514     COMMON /VAR/ NG,MODEX,NLOAD
515     COMMON/NELG/NELG
516     COMMON /PSTRES/ STPROP(720),TDISP(1500),TEPS(4,720),TSIG(4,720),
517 +     PRSTOR(1050),EPSTOR(4,1050),SGSTOR(4,1050),LCOUNT,IELCNT,
518 +     JTOTEL,JELCNT,ITER,NUMIT
519     COMMON A(1)
520 C
521 C
522 C     OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQOUT',FILE='IELMNT.SAP')
523 C     OPEN (UNIT=25,DEVICE='DSK',ACCESS='SEQOUT',FILE='PROST.SAP')
524 C     OPEN (UNIT=23,DEVICE='DSK',ACCESS='SEQOUT',FILE='TOTSTS.SAP')
525 C     OPEN (UNIT=24,DEVICE='DSK',ACCESS='SEQOUT',FILE='TOTSTN.SAP')
526 C     OPEN (UNIT=20,NAME='IELMNT.SAP',STATUS='UNKNOWN',
527 +     FORM='UNFORMATTED')
528 C     OPEN (UNIT=25,NAME='PROST.SAP',STATUS='UNKNOWN',
529 +     FORM='UNFORMATTED')
530 C     OPEN (UNIT=23,NAME='TOTSTS.SAP',STATUS='UNKNOWN',
531 +     FORM='UNFORMATTED')
532 C     OPEN (UNIT=24,NAME='TOTSTN.SAP',STATUS='UNKNOWN',
533 +     FORM='UNFORMATTED')
534 C     REWIND IELMNT
535 C     REWIND ISTRES
536 C     REWIND IEPS
537 C     REWIND ISIG
538 C     WRITE (IOUT,2000)
539 C
540 C
541 C     LOOP OVER ALL ELEMENT GROUPS
542 C
543 C
544 C     DO 100 N=1,NUMEG
545 C     IF (N.NE.1) WRITE (IOUT,2010)
546 C
547 C     READ (IIN,1000) NPAR
548 C     NUME=NPAR(2)
549 C     IF (NPAR(4).EQ.0) NPAR(4)=1
550 C     JTOTEL=JTOTEL+NUME
551 C     NELG=N
552 C     MODEL=NPAR(4)
553 C     DO 300 IL=1,NUME
554 C     STPROP(IL)=0.0
555 300 WRITE (ISTRES) STPROP(IL)
556 C     DO 301 IW=1,NUME
557 C     DO 302 IP=1,4
558 C     TEPS(IP,IW)=0.0
559 302 TSIG(IP,IW)=0.0
560 C     WRITE (IEPS) (TEPS(IWW,IW),IWW=1,4)

```

```

561 301 WRITE (ISIG) (TSIG(IWW,IW),IWW=1,4)
562 C
563 C
564 CALL ELEMNT
565 C
566 IF (MIDEST.GT.MAXEST) MAXEST=MIDEST
567 C
568 WRITE (IELMNT) MIDEST,NPAR,(A(I),I=NFIRST,NLAST)
569 C
570 C
571 100 CONTINUE
572 C
573 C CLOSE(UNIT=20,DEVICE='DSK',ACCESS='SEQOUT',FILE='IELMNT.SAP')
574 C CLOSE (UNIT=25,DEVICE='DSK',ACCESS='SEQOUT',FILE='PROST.SAP')
575 C CLOSE (UNIT=23,DEVICE='DSK',ACCESS='SEQOUT',FILE='TOTSTS.SAP')
576 C CLOSE (UNIT=24,DEVICE='DSK',ACCESS='SEQOUT',FILE='TOTSTN.SAP')
577 C CLOSE (UNIT=20)
578 C CLOSE (UNIT=25)
579 C CLOSE (UNIT=23)
580 C CLOSE (UNIT=24)
581 RETURN
582 C
583 1000 FORMAT (10I5)
584 2000 FORMAT (1H1,38HE L E M E N T G R O U P D A T A ///)
585 2010 FORMAT (1H1)
586 1050 FORMAT (16I5)
587 C
588 END
589 SUBROUTINE ELEMNT
590 C
591 C
592 C
593 C PROGRAM
594 C TO CALL THE APPROPRIATE ELEMENT SUBROUTINE
595 C
596 C
597 C
598 COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
599 COMMON/NELG/NELG
600 COMMON /PSTRES/ STPROP(720),TDISP(1500),TEPS(4,720),TSIG(4,720),
601 + PRSTOR(1050),EPSTOR(4,1050),SGSTOR(4,1050),LCOUNT,IELCNT,
602 + JTOTEL,JELCNT,ITER,NUMIT
603 COMMON /VAR/ NG,MODEX,NLOAD
604 C
605 NPAR1=NPAR(1)
606 C
607
608 GO TO (1,2,3),NPAR1
609 C
610 1 CALL TRUSS
611 RETURN
612 C
613 C OTHER ELEMENT TYPES WOULD BE CALLED HERE, IDENTIFYING EACH
614 C
615 C ELEMENT TYPE BY A DIFFERENT NPAR(1) PARAMETER
616 C
617 2 CALL AXISYM
618 RETURN
619 C
620 3 RETURN
621 C
622 END
623 SUBROUTINE CPLMT (MHT,ND,LM)
624 C
625 C
626 C PROGRAM
627 C TO CALCULATE COLUMN HEIGHTS
628 C
629 C
630 C
631 COMMON /SOL/ NUMNP,NEQ,NWK,NUMEST,MIDEST,MAXEST,MK
632 DIMENSION LM(1),MHT(1)
633 C
634 LS=100000
635 DO 100 I=1,ND
636 IF (LM(I)) 110,100,110
637 110 IF (LM(I)-LS) 120,100,100
638 120 LS=LM(I)
639 100 CONTINUE
640 C

```

```

641      DO 200 I=1,ND
642      II=LM(I)
643      IF (II.EQ.0) GO TO 200
644      ME=II - LS
645      IF (ME.GT.MHT(II)) MHT(II)=ME
646      CONTINUE
647 C
648      RETURN
649      END
650      SUBROUTINE ADDRES(MAXA,MHT)
651 C
652 C
653 C
654 C      PROGRAM
655 C      TO CALCULATE ADDRESSES OF DIAGONAL ELEMENTS IN BANDED
656 C      MATRIX WHOSE COLUMN HEIGHTS ARE KNOWN
657 C
658 C      MHT = ACTIVE COLUMN HEIGHTS
659 C      MAXA = ADDRESSES OF DIAGONAL ELEMENTS
660 C
661 C
662 C
663      COMMON /SQL/ NUMNP,NEQ,NWK,NUMEST,MIDEST,MAXEST,MK
664      DIMENSION MAXA(1),MHT(1)
665 C
666 C      CLEAR ARRAY MAXA
667 C
668      NN=NEQ + 1
669      DO 20 I=1,NN
670      MAXA(I)=0.0
671 C
672      MAXA(1)=1
673      MAXA(2)=2
674      MK=0
675      IF (NEQ.EQ.1) GO TO 100
676      DO 10 I=2,NEQ
677      IF (MHT(I).GT.MK) MK=MHT(I)
678      10 MAXA(I+1)=MAXA(I) + MHT(I) + 1
679      100 MK=MK+1
680      NWK=MAXA(NEQ+1) - MAXA(1)
681 C
682      RETURN
683      END
684      SUBROUTINE CLEAR(A,N)
685 C
686 C
687 C      PROGRAM
688 C      TO CLEAR ARRAY A
689 C
690 C
691 C
692 C      THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
693 C      CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON IBM
694 C      OR UNIVAC MACHINES.  ACTIVATE, DEACTIVATE OR ADJUST ABOVE
695 C      CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
696 C
697      DIMENSION A(1)
698      DO 10 I=1,N
699      A(I)=0.
700      RETURN
701      END
702      SUBROUTINE ASSEM (AA)
703 C
704 C
705 C
706 C      PROGRAM
707 C      TO CALL ELEMENT SUBROUTINES FOR ASSEMBLAGE OF THE
708 C      STRUCTURE STIFFNESS MATRIX
709 C
710 C
711 C
712      COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
713      COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
714      COMMON /NELG/NELG
715      COMMON /PSTRES/ STPROP(720),TDISP(1500),TEPS(4,720),TSIG(4,720),
716      + PRSTOR(1050),EPSTOR(4,1050),SGSTOR(4,1050),LCOUNT,IELCNT,
717      + JTOTEL,JELCNT,ITER,NUMIT
718      DIMENSION AA(1)
719 C
720 C      OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',FILE='IELMNT.SAP')

```

```

721 C      OPEN (UNIT=20,NAME='IELMNT.SAP',STATUS='OLD',FORM='UNFORMATTED')
722        REWIND IELMNT
723        REWIND ISTRES
724 C
725        DO 200 N=1,NUMEG
726          READ (IELMNT)NUMEST,NPAR,(AA(I),I=1,NUMEST)
727          NUME = NPAR(2)
728          NELG=N
729 C
730          CALL ELEMNT
731 C
732 200      CONTINUE
733 C      CLOSE (UNIT=20,DEVICE='DSK',ACCESS='SEQIN',FILE='IELMNT.SAP')
734 C      CLOSE (UNIT=20)
735        RETURN
736 C
737 C
738        END
739        SUBROUTINE ADDBAN (A,MAXA,S,LM,ND)
740 C
741 C
742 C
743 C      PROGRAM
744 C      TO ASSEMBLE UPPER TRIANGULAR ELEMENT STIFFNESS INTO
745 C      COMPACTED GLOBAL STIFFNESS
746 C
747 C      A = GLOBAL STIFFNESS
748 C      S = ELEMENT STIFFNESS
749 C      ND = DEGREES OF FREEDOM IN ELEMENT STIFFNESS
750 C
751 C          S(1)      S(2)      S(3)      . . .
752 C      S =      S(ND+1) S(ND+2) . . .
753 C              .      S(2*ND) . . .
754 C              .      .      . . .
755 C
756 C          A(1)      A(3)      A(6)      . . .
757 C      A =      A(2)      A(5)      . . .
758 C              .      A(4)      . . .
759 C              .      .      . . .
760 C
761 C
762 C
763 C
764 C
765 C
766 C      THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
767 C      CDC EQUIPMENT ANDDOUBLE PRECISION ARITHMETIC ON IBM
768 C      OR UNIVAC MACHINES.  ACTIVATE, DEACTIVATE OR ADJUST ABOVE
769 C      CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
770 C
771 C
772        DIMENSION A(1),MAXA(1),S(1),LM(1)
773 C
774        NDI=0
775        DO 200 I=1,ND
776          II=LM(I)
777          IF (II) 200,200,100
778 100      MI=MAXA(II)
779          KS=I
780          DO 220 J=1,ND
781            JJ=LM(J)
782            IF (JJ) 220,220,110
783 110      IJ=II - JJ
784            IF (IJ) 220,210,210
785 210      KK=MI + IJ
786          KSS=KS
787          IF (J.GE.I) KSS=J + NDI
788          A(KK)=A(KK) + S(KSS)
789 220      KS=KS + ND - J
790 200      NDI=NDI + ND - I
791 C      WRITE(50,1234)
792 1234    FORMAT(//,10X,'A MATRIX IN ADDBAN',//)
793 C      WRITE(50,2345)(A(I),I=1,21)
794 2345    FORMAT(5E15.5)
795 C
796        RETURN
797        END
798        SUBROUTINE COLSQL(A,V,MAXA,NN,NWK,NNM,KKK,ITER)
799 C
800 C

```

```

801 C      PROGRAM
802 C      TO SOLVE FINITE ELEMENT STATIC EQUILIBRIUM EQUATIONS IN
803 C      CORE, USING COMPACTED STORAGE AND COLUMN REDUCTION SCHEME
804 C
805 C      - INPUT VARIABLES -
806 C      A(NWK)      = STIFFNESS MATRIX STORED IN COMPACTED FORM
807 C      V(NN)       = RIGHT-HAND-SIDE LOAD VECTOR
808 C      MAXA(NNM)    = VECTOR CONTAINING ADDRESSES OF DIAGONAL
809 C                     ELEMENTS OF STIFFNESS MATRIX IN A
810 C      NN          = NUMBER OF EQUATIONS
811 C      NWK         = NUMBER OF ELEMENTS BELOW SKYLINE OF MATRIX
812 C      NNM         = NN + 1
813 C      KKK         = INPUT FLAG
814 C      EQ. 1 TRIANGULARIZATION OF STIFFNESS MATRIX
815 C      EQ. 2 REDUCTION AND BACK-SUBSTITUTION OF LOAD VECTOR
816 C      IOUT        = NUMBER OF OUTPUT DEVICE
817 C
818 C      - OUTPUT -
819 C      A(NWK)      = D AND L - FACTORS OF STIFFNESS MATRIX
820 C      V(NN)       = DISPLACEMENT VECTOR
821 C
822 C
823 C
824 C      THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
825 C      CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON IBM
826 C      OR UNIVAC MACHINES.  ACTIVATE, DEACTIVATE, OR ADJUST ABOVE
827 C      CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
828 C
829 C      COMMON /TAPES/ IELMNT, ILOAD, IIN, IOUT, ISTRES, IDISP, IEPS, ISIG
830 C      DIMENSION A(NWK), V(1), MAXA(1)
831 C
832 C      PERFORM L=D*L(T) FACTORIZATION OF STIFFNESS MATRIX
833 C
834 C      IF (KKK-2) 40, 150, 150
835 C      40 DO 140 N=1, NN
836 C          KN=MAXA(N)
837 C          KL=KN+1
838 C          KU=MAXA(N+1) - 1
839 C          KH=KU - KL
840 C          IF (KH) 110, 90, 50
841 C      50 K=N-KH
842 C          IC=0
843 C          KLT=KU
844 C          DO 80 J=1, KH
845 C              IC=IC + 1
846 C              KLT=KLT - 1
847 C              KI=MAXA(K)
848 C              ND=MAXA(K+1) - KI - 1
849 C              IF (ND) 80, 80, 60
850 C      60 KK=MINO(IC, ND)
851 C          C=0.
852 C          DO 70 L=1, KK
853 C              C=C+A(KI+L)-A(KLT+L)
854 C              A(KLT)=A(KLT) - C
855 C      70 K=K+1
856 C      90 K=N
857 C          B=0.
858 C          DO 100 KK=KL, KU
859 C              K=K - 1
860 C              KI=MAXA(K)
861 C              C=A(KK)/A(KI)
862 C              B=B + C*A(KK)
863 C      100 A(KK)=C
864 C          A(KN)=A(KN) - B
865 C      110 IF (A(KN)) 120, 120, 140
866 C      120 WRITE(IOUT, 2000) N, A(KN), ITER
867 C          STOP
868 C      140 CONTINUE
869 C          RETURN
870 C
871 C      REDUCE RIGHT-HAND-SIDE LOAD VECTOR
872 C
873 C      150 DO 180 N=1, NN
874 C          KL=MAXA(N) + 1
875 C          KU=MAXA(N+1) - 1
876 C          IF (KU-KL) 180, 160, 160
877 C      160 K=N
878 C          C=0.
879 C          DO 170 KK=KL, KU
880 C              K=K - 1

```

```

881 170 C=C+A(KK)=V(K)
882 V(N)=V(N) - C
883 180 CONTINUE
884 C
885 C BACK-SUBSTITUTE
886 C
887 DO 200 N=1,NN
888 K=MAXA(N)
889 200 V(N)=V(N)/A(K)
890 IF (NN.EQ.1) RETURN
891 N=NN
892 DO 230 L=2,NN
893 KL=MAXA(N) + 1
894 KU=MAXA(N+1) - 1
895 IF (KU-KL) 230,210,210
896 210 K=N
897 DO 220 KK=KL,KU
898 K=K - 1
899 220 V(K)=V(K)-A(KK)=V(N)
900 230 N=N-1
901 RETURN
902 2000 FORMAT(//48H STOP - STIFFNESS MATRIX NOT POSITIVE DEFINITE .//
903 1 32H NONPOSITIVE PIVOT FOR EQUATION ,I4.//
904 2 10H PIVOT = ,E20.12 .
905 3 ' ITERATION NUMBER',I10)
906 END
907 SUBROUTINE LOADV (R,NEQ,ITER)
908 C
909 C
910 C
911 C
912 C PROGRAM
913 C TO OBTAIN THE LOAD VECTOR
914 C
915 C
916 C THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
917 C CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON IBM
918 C OR UNIVAC MACHINES. ACTIVATE,DEACTIVATE OR ADJUST ABOVE
919 C CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
920 C
921 C
922 COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
923 DIMENSION R(NEQ)
924 C
925 IF (ITER.GT.1) GO TO 100
926 READ (ILOAD) R
927 REWIND ILOAD2
928 WRITE (ILOAD2) R
929 GO TO 200
930 100 REWIND ILOAD2
931 READ (ILOAD2) R
932 200 CONTINUE
933 C
934 RETURN
935 END
936 SUBROUTINE WRITE (DISP,ID,NEQ,NUMNP,TDISP,ITER,NUMIT,L)
937 C
938 C
939 C
940 C PROGRAM
941 C TO PRINT DISPLACEMENTS
942 C
943 C
944 C THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
945 C CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON IBM
946 C OR UNIVAC MACHINES. ACTIVATE, DECATIVATE OR ADJUST ABOVE
947 C CARD FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
948 C
949 C
950 COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
951 COMMON /CONV/ DOLD(1500),DNEW(1500),ICONV
952 DIMENSION DISP(NEQ),ID(3,NUMNP),TDISP(NEQ)
953 DIMENSION D(3)
954 C
955 C PRINT DISPLACEMENTS
956 C OPEN (UNIT=35,DEVICE='DSK',ACCESS='SEQOUT',FILE='DISP.SAP')
957 C OPEN (UNIT=35,NAME='DISP.SAP',STATUS='OLD',FORM='UNFORMATTED')
958 C REWIND IDISP
959
960 CHNGMX=0.0

```

```

951      IMX=0
962      IDMX=0
963      CHNGMN=100000.
964      IMN=0
965      IDMN=0
966      ICONV=0
967      DO 502 J=1,17
968      DO 501 J=1,3
969 501  D(J)=0.0
970      DO 502 J=1,3
971      KK=ID(J,IJ)
972      IL=J
973      IF (ITER.EQ.1) DOLD(KK)=DISP(KK)
974      IF (ITER.EQ.1) ICONV=0
975      IF (ITER.EQ.1) GO TO 502
976      IF (KK.EQ.0) GO TO 502
977      DNEW(KK)=DISP(KK)
978      CHNG=DNEW(KK)-DOLD(KK)
979      PCHNG=ABS(CHNG/DOLD(KK))-100.
980      IF (CHNGMX.GT.PCHNG) GO TO 509
981      IMX=IJ
982      IDMX=J
983      CHNGMX=PCHNG
984      GO TO 507
985 509  IF (CHNGMN.LT.PCHNG) GO TO 507
986      IMN=IJ
987      IDMN=J
988      CHNGMN=PCHNG
989 507  DOLD(KK)=DNEW(KK)
990 502  CONTINUE
991      IF (ITER.EQ.1) GO TO 505
992      IF (CHNGMX.LE.10.0) ICONV=1
993 505  CONTINUE
994 C
995 C
996 C      WRITE (IOUT,2000)
997      IC=4
998 C
999      DO 100 II=1,NUMNP
1000      IC=IC + 1
1001      IF (IC.LT.56) GO TO 105
1002 C      WRITE (IOUT,2000)
1003      IC=4
1004 105  DO 110 I=1,3
1005 110  D(I)=0.
1006 C
1007      DO 120 I=1,3
1008      KK=ID(I,II)
1009      IL=I
1010      IF (KK.NE.0) TDISP(KK)=TDISP(KK)+DISP(KK)
1011      IF (KK.NE.0) D(IL)=TDISP(KK)
1012 120  IF (ITER.LT.NUMIT.AND.ICONV.EQ.0.AND.KK.NE.0)
1013      +      TDISP(KK)=TDISP(KK)-DISP(KK)
1014 C
1015      IF (ITER.LT.NUMIT.AND.ICONV.EQ.0) GO TO 403
1016      WRITE (IOUT,2010) II,D
1017 403  CONTINUE
1018 100  CONTINUE
1019 C
1020      IF (ITER.EQ.NUMIT.AND.ICONV.EQ.0) WRITE(IOUT,3000)CHNGMX,IMX,IDMX,
1021      +      CHNGMN,IMN,IDMN
1022 3000 FORMAT (//,' *** MAXIMUM NUMBER OF ITERATIONS REACHED ***',/,
1023      +      ' **** WITHOUT CONVERGENCE ****',/, ' MAXIMUM PERCENT CHANGE',
1024      +      ' E15.7,' OCCURRED AT NODE',I5,' D O F',I5,/,
1025      +      ' MINIMUM PERCENT CHANGE',E15.7,' OCCURRED AT NODE',I5,
1026      +      ' D O F',I5)
1027 C
1028      WRITE (IDISP) (TDISP(IW),IW=1,NEQ)
1029 400  CONTINUE
1030 C      CLOSE (UNIT=35,DEVICE='DSK',ACCESS='SEQOUT',FILE='DISP.SAP')
1031 C      CLOSE (UNIT=35)
1032      RETURN
1033 C
1034 2000 FORMAT (///, 26H D I S P L A C E M E N T S // 7H NODE 9X
1035      114HX-DISPLACEMENT 4X 14HY-DISPLACEMENT 4X 14HZ-DISPLACEMENT)
1036 2010 FORMAT (1X,I3, 8X,3E18.6)
1037 C
1038      END
1039      SUBROUTINE STRESS (AA)
1040 C

```

```

1041 C
1042 C
1043 C
1044 C      PROGRAM
1045 C      TO CALL THE ELEMENT SUBROUTINE FOR THE CALCULATION OF
1046 C      STRESSES
1047 C
1048 C
1049 C
1050 C      COMMON /VAR/ NG,MODEX,NLOAD
1051 C      COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
1052 C      COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
1053 C      COMMON/NELG/NELG
1054 C      COMMON /PSTRES/STPROP(720),TDISP(1500),TEPS(4,720),TSIG(4,720),
1055 C      + PRSTOR(1050),EPSTOR(4,1050),SGSTOR(4,1050),LCOUNT,IELCNT,
1056 C      + JTOTEL,JELCNT,ITER,NUMIT
1057 C      COMMON /CONV/ DOLD(1500),DNEW(1500),ICONV
1058 C      DIMENSION AA(1)
1059 C
1060 C
1061 C      LOOP OVER ALL ELEMENT GROUPS
1062 C
1063 C
1064 C      OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',FILE='IELMNT.SAP')
1065 C      OPEN (UNIT=20,NAME='IELMNT.SAP',STATUS='OLD',FORM='UNFORMATTED')
1066 C      REWIND IELMNT
1067 C      REWIND ISTRES
1068 C      REWIND IEPS
1069 C      REWIND ISIG
1070 C      IELCNT=0
1071 C      JELCNT=0
1072 C
1073 C      DO 100 N=1,NUMEG
1074 C      NG=N
1075 C      NELG=N
1076 C
1077 C      READ (IELMNT) NUMEST,NPAR,(AA(I),I=1,NUMEST)
1078 C
1079 C      CALL ELEMNT
1080 C
1081 C      100 CONTINUE
1082 C      IF (ITER.LT.NUMIT.AND.ICONV.NE.1) GO TO 301
1083 C
1084 C      REWIND IEPS
1085 C      REWIND ISIG
1086 C      DO 301 LW=1,JTOTEL
1087 C      WRITE (IEPS) (EPSTOR(IR,LW),IR=1,4)
1088 C      WRITE (ISIG) (SGSTOR(IR,LW),IR=1,4)
1089 C      301 CONTINUE
1090 C      REWIND ISTRES
1091 C      DO 201 JW=1,JTOTEL
1092 C      201 WRITE (ISTRES) PRSTOR(JW)
1093 C      CLOSE(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',FILE='IELMNT.SAP')
1094 C      CLOSE (UNIT=20)
1095 C      RETURN
1096 C      END
1097 C      SUBROUTINE SECONO(TIM)
1098 C
1099 C
1100 C      SUBROUTINE TO OBTAIN TIME
1101 C
1102 C      CALL TIMER(ITIM)
1103 C      TIM=FLOAT(ITIM)/1000.
1104 C
1105 C      RETURN
1106 C      END
1107 C      SUBROUTINE ELSTIF(R,Z,S,MODEL,MTYPE,DETER,LCOUNT,ITER,N)
1108 C
1109 C      COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
1110 C      COMMON /EXMU/ PROP(10,10)
1111 C      DIMENSION R(1),Z(1),S(1)
1112 C      DIMENSION X(6,6),BA(6,6),AB(6,6)
1113 C      DIMENSION ST(6,6)
1114 C      WRITE(50,1234)(R(I),I=1,3)
1115 C      WRITE(50,1234)(Z(I),I=1,3)
1116 C      1234 FORMAT(//,10X,5E15.5,/)
1117 C      WRITE(50,1234)(Z(I),I=1,3)
1118 C      XMU=PROP(MTYPE,2)
1119 C      OMU=1.-XMU
1120 C      OPU=1.+XMU

```

```

1121      HMMU=.5-XMU
1122      CALL INTGR1(R,Z)
1123      DO 100 I=1,36
1124 100    S(I)=0.0
1125      S(01)=OMU-A4
1126      S(02)=A2
1127      S(03)=OMU-A5
1128      S(06)=XMU-A2
1129      S(08)=2.-A1
1130      S(09)=A3
1131      S(12)=2.*XMU-A1
1132      S(15)=OMU-A6+HMMU-A1
1133      S(17)=HMMU-A1
1134      S(18)=XMU-A3
1135      S(29)=HMMU-A1
1136      S(36)=OMU-A1
1137      S(07)=S(02)
1138      S(13)=S(03)
1139      S(31)=S(06)
1140      S(14)=S(09)
1141      S(32)=S(12)
1142      S(27)=S(17)
1143      S(33)=S(18)
1144 C      ***** SET UP MATERIAL MODELS *****
1145 C      MODEL=1  ASPHALT/CONCRETE
1146 C      MODEL=2  BASE
1147 C      MODEL=3  SUBGRADE
1148 C      *****
1149      GO TO (131,132,133,135),MODEL
1150 131    EEE=PROP(MTYPE,1)
1151      GO TO 134
1152 132    XNNC=PROP(MTYPE,3)
1153      XKKC=PROP(MTYPE,1)
1154      THETA=DETER
1155      EEE=XKKC-(THETA**XNNC)
1156      FGUESS=PROP(MTYPE,4)
1157      IF (LCOUNT.EQ.1.AND.ITER.EQ.1) EEE=XKKC*(FGUESS**XNNC)
1158      IF (N.LE.10) WRITE (6,1776) N, ITER, EEE
1159 1776  FORMAT (/,' ELEMENT',I5,' ITERATION',I5,' MODULUS',E15.6)
1160      GO TO 134
1161 C      *****
1162 C      THE FOLLOWING FORMULATION OF THE
1163 C      SUBGRADE MATERIAL MODEL IS BASED
1164 C      ON A BILINEAR REPRESENTATION
1165 C      OF THE RESILIENT MODULUS CURVE
1166 C      *****
1167 133    SDI=PROP(MTYPE,5)
1168      ERI=PROP(MTYPE,1)
1169      SD=DETER
1170      PK1=PROP(MTYPE,3)
1171      PK2=PROP(MTYPE,4)
1172      IF (DETER.LE.SDI) EEE=ERI+PK1*(DETER-SDI)
1173      IF (DETER.GT.SDI) EEE=ERI+PK2*(DETER-SDI)
1174 CC    IF (DETER.LE.2.0) EEE=ERI+PK1*(2.0-SDI)
1175 CC    IF (DETER.GE.32.8) EEE=ERI+PK2*(32.8-SDI)
1176      IF (LCOUNT.EQ.1.AND.ITER.EQ.1) EEE=ERI+PK1*(2.0-SDI)
1177      GO TO 134
1178 C      *****
1179 C      THE FOLLOWING FORMULATION OF THE
1180 C      SUBGRADE MATERIAL MODEL IS BASED
1181 C      ON A QUADRATIC REPRESENTATION
1182 C      OF THE RESILIENT MODULUS
1183 C      *****
1184 135    ACONS=PROP(MTYPE,3)
1185      BCONS=PROP(MTYPE,4)
1186      SD=DETER
1187      ERO=PROP(MTYPE,1)
1188      EEE=ERO+(ACONS*SD*SD)+(BCONS*SD)
1189      IF (LCOUNT.EQ.1.AND.ITER.EQ.1) EEE=ERO+(ACONS*1.00)+(BCONS*1.0)
1190 C      RAD=((BCONS*BCONS)-(4.0*ERO*ACONS))**.5
1191 C      FROOT=(-BCONS+RAD)/(2.0*ACONS)
1192 C      SROOT=(-BCONS-RAD)/(2.0*ACONS)
1193 C      FROOT=ABS(FROOT)
1194 C      SROOT=ABS(SROOT)
1195 C      IF (SD.GT.FROOT) EEE=ERO+(ACONS*100.)+(BCONS*10.)
1196 C      IF (SD.GT.SROOT) EEE=ERO+(ACONS*100.)+(BCONS*10.)
1197      IF (SD.GT.32.432) EEE=ERO+(ACONS*32.432*32.432)+(BCONS*32.432)
1198 134    CONTINUE
1199      AML=3.14159*EEE/(OPU*HMMU)
1200 C      WRITE(6,1212)AML

```

```

1201 1212 FORMAT(//,10X,5E15.5,/)
1202 C      WRITE(50,123)
1203 123  FORMAT(//,10X,'S MATRIX',/)
1204 C      CALL PRMT(6,6,S)
1205      DO 200 I=1,36
1206 200  S(I)=AML*S(I)
1207 C      WRITE(6,123)
1208 C      CALL PRMT(6,6,S)
1209      CALL MINVRS(R,Z,AB)
1210      JP=0
1211      DO 101 I=1,6
1212      DO 101 J=1,6
1213      JP=JP+1
1214      ST(I,J)=S(JP)
1215 101  CONTINUE
1216      CALL MPRD(ST,AB,X)
1217 C      WRITE(6,112)
1218 112  FORMAT(//,10X,'FIRST PRODUCT',/)
1219 C      CALL PRMT(6,6,X)
1220      DO 72 I=1,6
1221      DO 72 J=1,6
1222 72  BA(I,J)=AB(J,I)
1223 C      WRITE(6,113)
1224 113  FORMAT(//,10X,'AINVRS TRANSPOSE',/)
1225 C      CALL PRMT(6,6,BA)
1226      CALL MPRD(BA,X,S)
1227 C      WRITE(6,1)
1228 1  FORMAT(//,10X,'STIFNESS MATRIX',/)
1229 C      CALL PRMT(6,6,S)
1230      S(07)=S(08)
1231      S(08)=S(09)
1232      S(09)=S(10)
1233      S(10)=S(11)
1234      S(11)=S(12)
1235      S(12)=S(15)
1236      S(13)=S(16)
1237      S(14)=S(17)
1238      S(15)=S(18)
1239      S(16)=S(22)
1240      S(17)=S(23)
1241      S(18)=S(24)
1242      S(19)=S(29)
1243      S(20)=S(30)
1244      S(21)=S(36)
1245 C      WRITE (50,1776) (S(IW),IW=1,36)
1246 C 1776 FORMAT (' ELEMENT STIFFNESS MATRIX',/,
1247 C + 6(/,2X,6E13.5))
1248 4345  FORMAT(//,10X,'S(21)',/,(5E15.5))
1249      RETURN
1250      END

```

```

1      SUBROUTINE MPRD(S,AINV,X)
2 C
3      DIMENSION S(6,6),AINV(6,6),X(6,6)
4      DO 100 I=1,6
5      DO 100 J=1,6
6      X(I,J)=0.0
7      DO 100 K=1,6
8      100 X(I,J)=X(I,J)+S(I,K)*AINV(K,J)
9      RETURN
10     END

```

```

1      SUBROUTINE PRMT(NX,NY,AMAT)
2 C
3      DIMENSION AMAT(1)
4      NXY = NX*NY
5      DO 100 I=1,NX
6 100  WRITE(5,610)(AMAT(K),K=I,NXY,NX)
7 610  FORMAT(1X,8(1PE14.5))
8      RETURN
9      END

```

```

1      SUBROUTINE COEFS(R,Z,F,A)
2 C
3      COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
4      DIMENSION R(1),Z(1),F(1),A(1),AB(6,6)
5      CALL MINVRS(R,Z,AB)
6 C      WRITE(6,13)
7      13  FORMAT(//,10X,'INVERSE OF A MATRIX',//)
8 C      CALL PRMT(6,6,X)
9      DO 10 I=1,6
10     A(I)=0.0
11     DO 10 J=1,6
12     10  A(I)=A(I)+AB(I,J)*F(J)
13     RETURN
14     END

```

```

1      SUBROUTINE STRSN(R,Z,A,E,EPS,SIG)
2 C
3      COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
4      DIMENSION R(1),Z(1),F(1),A(1),E(1),EPS(1),SIG(1),X(24)
5      DO 10 I=1,24
6          10 X(I)=0.0
7          TRAVG=(R(1)+R(2)+R(3))
8          X(1)=Z(2)-Z(3)
9          X(2)=BETA/TRAVG
10         X(2)=(R(1)+(Z(2)-Z(3))-R(2)=(Z(3)+Z(1))+R(3)=(Z(2)+Z(1)))/TRAVG
11         X(4)=R(3)-R(2)
12         X(7)=X(4)
13         X(8)=X(1)
14         X(9)=Z(3)-Z(1)
15         X(10)=X(2)
16         X(10)=(R(1)=(Z(3)+Z(2))+R(2)=(Z(3)-Z(1))-R(3)=(Z(1)+Z(2)))/TRAVG
17         X(12)=R(1)-R(3)
18         X(15)=X(12)
19         X(16)=X(9)
20         X(17)=Z(1)-Z(2)
21         X(18)=X(2)
22         X(18)=(-R(1)=(Z(2)+Z(3))+R(2)=(Z(1)+Z(3))+R(3)=(Z(1)-Z(2)))/TRAVG
23         X(20)=R(2)-R(1)
24         X(23)=X(20)
25         X(24)=X(17)
26         DO 11 I=1,24
27             11 X(I)=X(I)/BETA
28 C          WRITE(5,6)
29             6 FORMAT(//,10X,'STRAIN DISPLACEMENT MATRIX',//)
30 C          CALL PRMT(4,6,X)
31             DO 20 J=1,4
32                 EPS(J)=0.0
33                 DO 20 K=1,6
34                     M=4*(K-1)+J
35                     EPS(J)=EPS(J)+X(M)=A(K)
36             20 CONTINUE
37                 DO 30 J=1,4
38                     SIG(J)=0.0
39                     DO 30 K=1,4
40                         M=4*(K-1)+J
41                         SIG(J)=SIG(J)+E(M)=EPS(K)
42             30 CONTINUE
43             RETURN
44         END

```

```

1      SUBROUTINE AXISYM
2  C
3  C
4  C
5  C      PROGRAM
6  C      TO SET UP STORAGE AND CALL THE AXISYMMETRIC ELEMENT SUBROUTINE
7  C
8  C
9  C
10 C
11 COMMON /SOL/ NUMNP,NEQ,NWK,NUMEST,MIDEST,MAXEST,MK
12 COMMON /DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
13 COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
14 COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
15 COMMON /VAR/ NG,MODEX,NLOAD
16 COMMON A(1)
17
18 C      EQUIVALENCE (NPAR(2),NUME),(NPAR(3),NUMMAT),(NPAR(4),MODEL)
19 C
20 NFIRST=N6
21 IF (IND.GT.1) NFIRST=N5
22 N101=NFIRST
23 N102=N101 + NUMMAT=ITWO
24 N103=N102 + NUMMAT=ITWO
25 N104=N103 + 6*NUME
26 N105=N104 + 6*NUME=ITWO
27 N106=N105 + NUME
28 C
29 IF (IND.GT.1) GO TO 100
30 IF (NLAST.GT.MTOT) CALL ERROR(NLAST-MTOT,3)
31 GO TO 200
32 100 IF (NLAST.GT.MTOT) CALL ERROR(NLAST-MTOT,4)
33 C
34 200 MIDEST=NLAST - NFIRST
35 C
36 CALL AXSM(A(N1),A(N2),A(N3),A(N4),A(N4),A(N5),A(N101),A(N10
37 1 2),A(N103),A(N104),A(N105))
38 C
39 RETURN
40 C
41 END
42 SUBROUTINE AXSM (ID,X,Y,Z,U,MHT,E,XMU,LM,XYZ,MATP)
43 C
44 C
45 C
46 C      AXISYMMETRIC ELEMENT SUBROUTINE
47 C
48 C
49 C
50 C
51 C      THIS PROGRAM IS USED IN SINGLE PRECISION ARITHMETIC ON
52 C      CDC EQUIPMENT AND DOUBLE PRECISION ARITHMETIC ON IBM
53 C      OR UNIVAC MACHINES.  ACTIVATE, DEACTIVATE, OR ADJUST ABOVE
54 C      CARDS FOR SINGLE OR DOUBLE PRECISION ARITHMETIC
55 C
56 COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
57 COMMON /SOL/ NUMNP,NEQ,NWK,NUMEST,MIDEST,MAXEST,MK
58 COMMON /DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
59 COMMON /EL/ IND,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
60 COMMON /VAR/ NG,MODEX,NLOAD
61 COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
62 COMMON /EXMU/ PROP(10,10)
63 COMMON /NELG/ NELG
64 COMMON /PSTRES/ STPROP(720),TDISP(1500),TEPS(4,720),TSIG(4,720),
65 + PRSTOR(1050),EPSTOR(4,1050),SGSTOR(4,1050),LCOUNT,IELCNT,
66 + JTOTEL,JELCNT,ITER,NUMIT
67 COMMON /CONV/ DOLD(1500),DNEW(1500),ICONV
68 COMMON /NNN/ NODNUM(3500,10)
69 COMMON /IFIX/ IFIX
70 COMMON A(1)
71 C
72 C
73 DIMENSION X(1),Y(1),Z(1),XYZ(6,1),E(1),XMU(1),U(1),
74 1 ID(3,1),LM(6,1),MATP(1),MHT(1)
75 DIMENSION DR(3),IPS(1)
76 DIMENSION S(36),ST(6),D(3)
77 DIMENSION RR(3),ZZ(3),EE(16),EPS(4),SIG(4),CFS(6),UD(6),DFC(6)
78 DIMENSION AB(6,4),DFP(6),NOD(3),AA(4,6)
79 DIMENSION PRINS(3)
80 C

```

```

81      EQUIVALENCE (NPAR(1),NPAR1),(NPAR(2),NUME),(NPAR(3),NUMMAT),
82      +      (NPAR(4),MODEL)
83      ND=6
84 C
85      GO TO (300,610,900).IND
86 C
87 C
88 C      READ AND GENERATE ELEMENT
89 C      INFORMATION
90 C
91 C      READ MATERIAL INFORMATION
92 C
93      300  WRITE (IOUT,2000) NPAR1,NUME
94      IF (NUMMAT.EQ.0) NUMMAT=1
95 C      WRITE (IOUT,2010) NUMMAT
96 C
97 C      WRITE (IOUT,2020)
98      WRITE (IOUT,2011) MODEL
99      2011  FORMAT (' M O D E L   T Y P E',/, ' MODEL (NPAR(4)) ... =',
100      +      ' 15./, ' EQ. 1 LINEAR',/, ' EQ. 2 NONLINEAR GRAVEL',/,
101      +      ' STONE',/, ' EQ. 3 BILINEAR SUBGRADE',/,
102      +      ' EQ. 4 QUADRATIC SUBGRADE',/)
103      DO 11 I=1,NUMMAT
104      READ (IIN,1001) N,(PROP(N,J),J=1,5)
105      1001  FORMAT (I5,5E10.3)
106      XMU(I)=PROP(N,2)
107      E(I)=PROP(N,1)
108      11  CONTINUE
109      GO TO (1,2,3,5), MODEL
110      1  DO 10 I=1,NUMMAT
111      10  WRITE (IOUT,2030) I,(PROP(I,MN),MN=1,2)
112      GO TO 4
113      2  WRITE (IOUT,2021)
114      DO 12 I=1,NUMMAT
115      12  WRITE (IOUT,2031) I,(PROP(I,J),J=1,4)
116      GO TO 4
117 C
118 C
119      3  WRITE (IOUT,2022)
120      DO 13 I=1,NUMMAT
121      13  WRITE (IOUT,2032) I,(PROP(I,J),J=1,5)
122      GO TO 4
123      5  WRITE (IOUT,2058)
124      DO 73 I=1,NUMMAT
125      73  WRITE (IOUT,2059) I,(PROP(I,J),J=1,4)
126      4  CONTINUE
127 C
128 C      READ ELEMENT INFORMATION
129 C
130 C      WRITE (IOUT,2040)
131      N=1
132      ICALL=0
133      100  READ (IIN,1020) M,II,JJ,KK,MTYP,KG,DPNRML
134      IF (KG.EQ.0) KG=1
135      120  IF (M.NE.N) GO TO 200
136      I=II
137      J=JJ
138      K=KK
139      IJK=(N-1)*3
140      NODNUM(IJK+1,NELG)=I
141      NODNUM(IJK+2,NELG)=J
142      NODNUM(IJK+3,NELG)=K
143      MTYPE=MTYP
144      KKK=KG
145 C
146 C      SAVE ELEMENT INFORMATION
147 C
148      200  XYZ(1,N)=X(I)
149      XYZ(2,N)=Y(I)
150 C
151      XYZ(3,N)=X(J)
152      XYZ(4,N)=Y(J)
153 C
154      XYZ(5,N)=X(K)
155      XYZ(6,N)=Y(K)
156 C
157 C
158      MATP(N)=MTYPE
159 C
160      DO 390 L=1,6

```

```

161 390 LM(L,N)=0.
162 DO 400 L=1,2
163 LM(L,N)=ID(L,I)
164 LM(L+2,N)=ID(L,J)
165 400 LM(L+4,N)=ID(L,K)
166 C
167 C UPDATE COLUMN HEIGHTS AND BANDWIDTH
168 C
169 C WRITE (6,1776) M
170 C1776 FORMAT (' GOT TO ELEMENT',I10)
171 CALL COLHT (MHT,ND,LM(1,N))
172 C
173 C WRITE (IOUT,2050) N,I,J,K,MTYPE,DPNRML
174 C
175 C CALCULATE PRESSURE LOADS
176 C
177 IF (DPNRML .EQ. 0) GO TO 410
178 ICALL=ICALL+1
179 DO 507 L=1,3
180 K=2=(L-1)+1
181 RR(L)=XYZ(K,N)
182 507 ZZ(L)=XYZ(K+1,N)
183 CALL INTGR2(RR,ZZ,DFP,1,2,3)
184 DO 44 IA=1,6
185 44 DFP(IA)=DPNRML*DFP(IA)
186 C WRITE(6,5473)(DFP(IA),IA=1,6)
187 5473 FORMAT(//,10X,'PRESSURE LOADING VECTOR',//,6E12.3)
188 DO 403 L=1,3
189 403 NOD(L)=NODNUM(IJK+L,NELG)
190 CALL PRSLDS (NOD,DFP,ID,NEQ,ICALL)
191 410 CONTINUE
192 C
193 IF (N.EQ.NUME) RETURN
194 N=N+1
195 I=I+KKK
196 J=J+KKK
197 K=K+KKK
198 IF (N.GT.M) GO TO 100
199 GO TO 120
200 C
201 C
202 C
203 C ASSEMBLE STRUCTURE STIFFNESS MATRIX
204 C
205 C
206 610 DO 500 N=1,NUME
207 C WRITE (6,1776) N
208 C1776 FORMAT (' GOT TO ELEMENT',I10)
209 MTYPE=MATP(N)
210 DO 505 L=1,3
211 K=2=(L-1)+1
212 RR(L)=XYZ(K,N)
213 505 ZZ(L)=XYZ(K+1,N)
214 READ (ISTRES) STPROP(N)
215 CALL ELSTIF(RR,ZZ,S,MODEL,MTYPE,STPROP(N),LCOUNT,ITER,N)
216 CALL ADDBAN (A(N3),A(N2),S,LM(1,N),ND)
217 500 CONTINUE
218 RETURN
219 C
220 C
221 C STRESS CALCULATIONS
222 C
223 C
224 900 IPRINT=0
225 ISTRS=0
226 C OPEN(UNIT=41,DEVICE='DSK',MODE='ASCII',ACCESS='SEQUENT',
227 C 1 DISPOSE='SAVE',FILE='STRESS.DAT')
228 DO 830 N=1,NUME
229 MTYPE=MATP(N)
230 XMUNNN=PROP(MTYPE,2)
231 READ (ISTRES) STPROP(N)
232 READ (ISIG) (TSIG(IR,N),IR=1,4)
233 READ (IEPS) (TEPS(IR,N),IR=1,4)
234 C ***** SET UP MATERIAL MODELS *****
235 C MODEL=1 ASPHALT/CONCRETE
236 C MODEL=2 BASE
237 C MODEL=3 SUBGRADE
238 C *****
239 GO TO (351,352,353,355),MODEL
240 351 ENNN=PROP(MTYPE,1)

```

```

241      GO TO 354
242 352 XNNC=PROP(MTYPE,3)
243      XKKC=PROP(MTYPE,1)
244      FGUESS=PROP(MTYPE,4)
245      THETA=STPROP(N)
246      ENNN=XKKC*(THETA**XNNC)
247      IF (LCOUNT.EQ.1.AND.ITER.EQ.1) ENNN=XKKC*(FGUESS**XNNC)
248      GO TO 354
249 C      *****
250 C      THE FOLLOWING FORMULATION OF THE SUB-
251 C      GRADE MATERIAL MODEL IS BASED ON A
252 C      BILINEAR REPRESENTATION OF THE
253 C      RESILIENT MODULUS CURVE
254 C      *****
255 353 SDI=PROP(MTYPE,5)
256      ERI=PROP(MTYPE,1)
257      SD=STPROP(N)
258      PK1=PROP(MTYPE,3)
259      PK2=PROP(MTYPE,4)
260      IF (SD.LE.SDI) ENNN=ERI+PK1*(SD-SDI)
261      IF (SD.GT.SDI) ENNN=ERI+PK2*(SD-SDI)
262 CC      IF (SD.LE.2.0) ENNN=ERI+PK1*(2.0-SDI)
263 CC      IF (SD.GE.32.8) ENNN=ERI+PK2*(32.8-SDI)
264      IF (LCOUNT.EQ.1.AND.ITER.EQ.1) ENNN=ERI+PK1*(2.0-SDI)
265      GO TO 354
266 C      *****
267 C      THE FOLLOWING FORMULATION FOR THE
268 C      SUBGRADE MATERIAL MODEL IS BASED ON
269 C      A QUADRATIC REPRESENTATION OF THE
270 C      RESILIENT MODULUS
271 C      *****
272 355 ERO=PROP(MTYPE,1)
273      ACONS=PROP(MTYPE,3)
274      BCONS=PROP(MTYPE,4)
275      SD=STPROP(N)
276      ENNN=ERO+(ACONS*SD-SD)+(BCONS-SD)
277      IF (LCOUNT.EQ.1.AND.ITER.EQ.1) ENNN=ERO+(ACONS-1.00)+(BCONS-1.0)
278 C      RAD=((BCONS-BCONS)-(4.0-ERO-ACONS))**.5
279 C      FROOT=(-BCONS+RAD)/(2.0-ACONS)
280 C      SROOT=(-BCONS-RAD)/(2.0-ACONS)
281 C      FROOT=ABS(FROOT)
282 C      SROOT=ABS(SROOT)
283 C      IF (SD.GT.FROOT) ENNN=ERO+(ACONS-100.)+(BCONS-10.)
284 C      IF (SD.GT.SROOT) ENNN=ERO+(ACONS-100.)+(BCONS-10.)
285      IF (SD.GT.32.432) ENNN=ERO+(ACONS-32.432-32.432)+(BCONS-32.432)
286 354 CONTINUE
287      DO 30 I=1,16
288 30      EE(I)=0.0
289          EE(1)=1.0-XMUNNN
290          EE(2)=XMUNNN
291          EE(3)=XMUNNN
292          EE(5)=XMUNNN
293          EE(6)=1.0-XMUNNN
294          EE(7)=XMUNNN
295          EE(9)=XMUNNN
296          EE(10)=XMUNNN
297          EE(11)=1.0-XMUNNN
298          EE(16)=0.5-XMUNNN
299          DO 31 I=1,16
300 31      EE(I)=EE(I)-ENNN/(((1.+XMUNNN)-(0.5-XMUNNN)*2.)
301          DO 506 L=1,3
302              K=2*(L-1)+1
303              RR(L)=XYZ(K,N)
304              ZZ(L)=XYZ(K+1,N)
305 C          WRITE(5,8989) N,L,RR(L),ZZ(L)
306 8989      FORMAT(' ',2I5,2(1X,1PE10.2))
307 506      CONTINUE
308 C          WRITE(6,7842)(U(I),I=1,5)
309 7842      FORMAT(/,10X,'U VECTOR',/,6E12.3)
310          I1=0
311          IJK=(N-1)*3
312          DO 34 J=1,3
313          DO 34 I=1,2
314              I1=I1+1
315              K=NODNUM(IJK+J,NELG)
316              KK=ID(I,K)
317 C          WRITE(6,9843)KK
318 9843      FORMAT(/,I10)
319          IF(KK.EQ.0) GO TO 33
320          UD(I1)=U(KK)

```

```

321      GO TO 34
322      33 UD(11)=0.0
323      34 CONTINUE
324 C     TYPE 9801,N
325      9801 FORMAT(//,' ELEMENT NO. ',14,/)
326 C     WRITE(5,9876) N,(UD(I),I=1,6)
327      9876 FORMAT(//,10X,'DISPLACEMENT VECTOR UD',/,15,(6E12.3))
328      CALL INTGR1(RR,ZZ)
329      CALL COEFS(RR,ZZ,UD,CFS)
330 C     WRITE(6,6)(CFS(I),I=1,6)
331      6 FORMAT(//,10X,'COEFFICIENTS',/,/(6E12.3))
332      IF(ISTRN.NE.0) GO TO 9079
333      CALL STRSN(RR,ZZ,UD,EE,EPS,SIG)
334      GO TO 9095
335 C
336 C
337      9079 DO 9080 II=1,4
338            DO 9080 JJ=1,6
339            AA(II,JJ)=0.
340      9080 CONTINUE
341            RAVG=(RR(1)+RR(2)+RR(3))/3.
342            ZAVG=(ZZ(1)+ZZ(2)+ZZ(3))/3.
343            AA(1,2)=1.
344            AA(2,1)=1./RAVG
345            AA(2,2)=1.
346            AA(2,3)=ZAVG/RAVG
347            AA(3,6)=1.
348            AA(4,3)=1.
349            AA(4,5)=1.
350            DO 9090 II=1,4
351            EPS(II)=0.
352            DO 9090 JJ=1,6
353            EPS(II)=EPS(II)+AA(II,JJ)*CFS(JJ)
354      9090 CONTINUE
355            DO 9091 II=1,4
356            SIG(II)=0.
357            DO 9091 JJ=1,4
358            M=4-(JJ-1)*II
359            SIG(II)=SIG(II)+EE(M)*EPS(JJ)
360      9091 CONTINUE
361 C
362 C
363      9095 CONTINUE
364            DO 513 IAD=1,4
365            TEPS(IAD,N)=TEPS(IAD,N)+EPS(IAD)
366            TSIG(IAD,N)=TSIG(IAD,N)+SIG(IAD)
367      513 CONTINUE
368            GO TO (240,241,241,241).MODEL
369      240 STPROP(N)=0.0
370            GO TO 243
371 C      241 SIGP1=(2.0-TSIG(1,N)-TSIG(2,N)-TSIG(3,N))/3.0
372 C      SIGP2=(2.0-TSIG(2,N)-TSIG(1,N)-TSIG(3,N))/3.0
373 C      SIGP3=(2.0-TSIG(3,N)-TSIG(1,N)-TSIG(2,N))/3.0
374      241 SIGP1=TSIG(1,N)
375            SIGP2=TSIG(2,N)
376            SIGP3=TSIG(3,N)
377            STINV1=-(SIGP1+SIGP2+SIGP3)
378            STINV2=(SIGP2+SIGP3+SIGP3+SIGP1+SIGP1+SIGP2-TSIG(4,N)+TSIG(4,N))
379            STINV3=-((SIGP1+SIGP2+SIGP3)-(TSIG(4,N)+TSIG(4,N)+SIGP3))
380            VID3=STINV1/3.0
381            AC=3.0*(VID3-VID3)-STINV2
382            BC=-2.0*(VID3-VID3-VID3)+STINV2-(VID3)-STINV3
383            PC=AC/3.0
384            OC=BC/2.0
385            PI=3.141592654
386            DEGRAD=180.0
387            UC=OC/(PC*(SORT(PC)))
388            UC=UC*(PI/DEGRAD)
389            UC=ACOS(UC)
390            SIGMA1=2.0*SORT(PC)*(COS(UC/3.0))
391            SIGMA2=2.0*SORT(PC)*(COS((UC/3.0)+2.094395103))
392            SIGMA3=2.0*SORT(PC)*(COS((UC/3.0)+4.188790206))
393            PRINS(1)=SIGMA1-(STINV1/3.0)
394            PRINS(2)=SIGMA2-(STINV1/3.0)
395            PRINS(3)=SIGMA3-(STINV1/3.0)
396            PMAST=PRINS(1)
397            PMINST=PRINS(1)
398            DO 244 IMAX=2,3
399            IF (PRINS(IMAX).GT.PMAST) PMAST=PRINS(IMAX)
400            IF (PRINS(IMAX).LT.PMINST) PMINST=PRINS(IMAX)

```

```

401 244 CONTINUE
402 DO 245 IMAX=1,3
403 IF (PRINS(IMAX).LT.PMAXST.AND.PRINS(IMAX).GT.PMINST)
404 + PMIDST=PRINS(IMAX)
405 245 CONTINUE
406 S1=PMAXST
407 S3=PMINST
408 SV=TSIG(3,N)
409 ISKIP=0
410 IF (IFIX.EQ.1) ISKIP=1
411 IF (MODEL.EQ.4) ISKIP=1
412 IF (ISKIP.EQ.1) GO TO 705
413 C=0.0
414 IF (MODEL.EQ.3) C=16.4
415 PHI=40.0*(PI/180.)
416 IF (MODEL.EQ.3) PHI=0.0
417 ANGSP=SIN((45.0*(PI/180.))+(PHI/2.0))
418 ANGSP=COS((45.0*(PI/180.))+(PHI/2.0))
419 ANGSP=SIN((45.0*(PI/180.))-(PHI/2.0))
420 ANGSP=COS((45.0*(PI/180.))-(PHI/2.0))
421 ANGSP=ANGSP/ANGSP
422 ANGSP=ANGSP/ANGSP
423 S1MAX=SV*(ANGSP-ANGSP)+(2.0*C-ANGSP)
424 S3MIN=SV*(ANGSP-ANGSP)+(2.0*C-ANGSP)
425 IF (S3MIN.GT.0.0) GO TO 700
426 IF (S3.GT.S3MIN) GO TO 701
427 707 IF (S3.GE.SV) GO TO 702
428 706 S1PRM=S3*(ANGSP-ANGSP)+(2.0*C-ANGSP)
429 IF (S1.LT.S1PRM) GO TO 703
430 IF (S1.GE.SV) GO TO 704
431 GO TO 705
432 700 S3MIN=0.0
433 IF (S3.GT.S3MIN) S3=0
434 IF (S3.GT.S3MIN) GO TO 706
435 GO TO 707
436 701 S3=S3MIN
437 704 S1=SV
438 GO TO 705
439 702 S3=SV
440 IF (S1.GE.S1MAX) GO TO 705
441 S1=S1MAX
442 GO TO 705
443 703 S1=S1PRM
444 705 PMAXST=S1
445 PMINST=S3
446 STPROP(N)=(PMAXST+PMIDST+PMINST)
447 C STPROP(N)=ABS(PMINST)
448 C IF (STPROP(N).LT.0.0) STPROP(N)=0.001
449 IF (STPROP(N).LT.0.0) STPROP(N)=FGUESS/1000.
450 IF (MODEL.EQ.2) GO TO 243
451 STPROP(N)=PMAXST-PMINST
452 243 CONTINUE
453 JELCNT=JELCNT+1
454 PRSTOR(JELCNT)=STPROP(N)
455 IELCNT=IELCNT+1
456 DO 514 IST=1,4
457 SGSTOR(IST,IELCNT)=TSIG(IST,N)
458 EPSTOR(IST,IELCNT)=TEPS(IST,N)
459 514 CONTINUE
460 IF (ITER.LT.NUMIT.AND.ICONV.EQ.0) GO TO 830
461 C WRITE(5,7)(EPS(I),I=1,4)
462 7 FORMAT(//,10X,'STRAIN VECTOR',//,(6E12.3))
463 C WRITE(5,9)
464 9 FORMAT(//,10X,'E MATRIX',//)
465 C CALL PRMT(4,4,EE)
466 C WRITE(5,8)(SIG(I),I=1,4)
467 8 FORMAT(//,10X,'STRESS VECTOR',//,(6E12.3))
468 C TYPE 8899,N,(SIG(IJ),IJ=1,4)
469 8899 FORMAT(' ','N=' ,I3,' SIG=' ,4(1X,1PE10.2))
470 IF (N.EQ.1) WRITE (IOUT,9013) NG
471 9013 FORMAT (//,' E L E M E N T G R O U P ',I5,/)
472 C WRITE(41,9009) (TSIG(IK,N),IK=1,4)
473 WRITE (IOUT,9009) N,(TSIG(IK,N),IK=1,4),ENNN
474 9009 FORMAT(I5,5E15.4)
475 IF (MODEL.EQ.3.OR.MODEL.EQ.4) GO TO 1792
476 WRITE (IOUT,9125) (TEPS(IK,N),IK=1,4)
477 9125 FORMAT (5X,4E15.4)
478 GO TO 1793
479 1792 WRITE (IOUT,9126) (TEPS(IK,N),IK=1,4),PRSTOR(JELCNT)
480 9126 FORMAT (5X,5E15.4)

```

```

481 1793 CONTINUE
482 830 CONTINUE
483 C WRITE(41,9010) ((XYZ(IJ,IL),IJ=1,6),IL=1,NUME)
484 C WRITE (IOUT,9010) (IL,(XYZ(IJ,IL),IJ=1,6),IL=1,NUME)
485 9010 FORMAT(I5,6E15.4)
486 C CLOSE(UNIT=41,DEVICE='DSK',MODE='ASCII',ACCESS='SEQOUT',
487 C 1 DISPOSE='SAVE',FILE='STRESS.DAT')
488 C
489 RETURN
490 C
491 1000 FORMAT (I5,2F10.0)
492 1010 FORMAT (2F10.0)
493 1020 FORMAT (6I5,F10.0)
494 2000 FORMAT (36H E L E M E N T D E F I N I T I O N ///.
495 1 14H ELEMENT TYPE ,13(2H .),17H( NPAR(1) ) . . =,I5/,
496 2 25H EQ.1, TRUSS ELEMENTS/,
497 3 32H EQ.2, AXISYMMETRIC ELEMENTS/,
498 4 44H EQ.3, ELEMENTS CURRENTLY NOT AVAILABLE /.
499 5 20H NUMBER OF ELEMENTS.,10(2H .),17H( NPAR(2) ) . . =,
500 6 I5//)
501 2010 FORMAT (42H M A T E R I A L D E F I N I T I O N ///.
502 1 37H NUMBER OF DIFFERENT SETS OF MATERIAL
503 2 /32H AND CROSS-SECTIONAL CONSTANTS .
504 3 4(2H .),17H( NPAR(3) ) . . =,I5//)
505 2020 FORMAT (///2X,3HSET,7X,6HYOUNGS,10X,8HPOISSONS/
506 1 1X,6HNUMBER,5X,7HMODULUS,10X,5HRATIO,/15X,1HE,14X,3HXMU)
507 2030 FORMAT (/I5,4X,E12.5,2X,E14.6)
508 2040 FORMAT (1H1,40H E L E M E N T I N F O R M A T I O N ///.
509 1 8H ELEMENT ,5X,4HNODE,5X,4HNODE,5X,4HNODE,7X,8HMATERIAL,
510 2 8X,15HNORMAL PRESSURE/,
511 3 9H NUMBER-N,6X,1H1,8X,1HJ,8X,1HK,7X,10HSET NUMBER,
512 4 9X,11HON I-J SIDE/)
513 2050 FORMAT (I5,6X,I5,4X,I5,4X,I5,7X,I5,14X,E12.6)
514 2060 FORMAT (///,46H S T R E S S C A L C U L A T I O N S F O R .
515 1 24H E L E M E N T G R O U P ,I4//,2X,
516 2 7HELEMENT,12X,6HSTRAIN,11X,6HSTRESS /,2X,6HNUMBER,11X,
517 3 9H(ELASTIC)/)
518 2070 FORMAT (1X,I5,11X,E13.6,4X,E13.6/,.(17X,E13.6,4X,E13.6))
519 2021 FORMAT (///2X,3HSET,7X,10HK-CONSTANT,7X,14HPOISSONS RATIO,
520 + 7X,10HN-CONSTANT,5X,11HFIRST GUESS./)
521 2031 FORMAT (I5,6X,E12.5,6X,E12.5,7X,E12.5,7X,F10.3./)
522 2022 FORMAT (///2X,3HSET,7X,11HBREAKPT MOD,5X,14HPOISSONS RATIO,
523 + 5X,2HK1,5X,2HK2,5X,14HDEV STRESS BPT./)
524 2032 FCRMAT (I5,2X,E12.5,2X,F12.5,2X,E12.5,2X,E12.5,2X,E12.5./)
525 2058 FORMAT (///2X,3HSET,11X,3HERO,9X,14HPOISSONS RATIO,3X,
526 + 10HCONSTANT A,3X,10HCONSTANT B./)
527 2059 FORMAT (I5,2X,E12.5,2X,F12.5,2X,E12.5,2X,E12.5./)
528 C
529 END
530 SUBROUTINE INTGR1(R,Z)
531 C
532 COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
533 DIMENSION R(1),Z(1)
534 R123=R(1)+R(2)+R(3)
535 Z123=Z(1)+Z(2)+Z(3)
536 R12=R(1)-R(2)
537 R23=R(2)-R(3)
538 R31=R(3)-R(1)
539 Z12=Z(1)-Z(2)
540 Z23=Z(2)-Z(3)
541 Z31=Z(3)-Z(1)
542 RZ=R(1)*Z23+R(2)*Z31+R(3)*Z12
543 BETA=RZ
544 R1R2=R(1)/R(2)
545 R2R3=R(2)/R(3)
546 R3R1=R(3)/R(1)
547 RZ12=R(1)*Z(2)-R(2)*Z(1)
548 RZ23=R(2)*Z(3)-R(3)*Z(2)
549 RZ31=R(3)*Z(1)-R(1)*Z(3)
550 A1=R123/RZ/6.
551 A2=0.5*RZ
552 A3=Z123/RZ/6.
553 C *****8
554 IRFLG=0
555 IF (R(1).EQ.R(2).AND.R(3).NE.0.0001) IRFLG=6
556 IF (R(2).EQ.R(3).AND.R(1).NE.0.0001) IRFLG=4
557 IF (R(3).EQ.R(1).AND.R(2).NE.0.0001) IRFLG=5
558 IF (R(1).EQ.0.0001.AND.R(2).NE.0.0001.AND.R(3).NE.0.0001) IRFLG=1
559 IF (R(2).EQ.0.0001.AND.R(3).NE.0.0001.AND.R(1).NE.0.0001) IRFLG=2
560 IF (R(3).EQ.0.0001.AND.R(1).NE.0.0001.AND.R(2).NE.0.0001) IRFLG=3

```

```

561 C      *****E
562      A41=RZ12/R(1)
563 C      WRITE(6,1212)A1,A2,A3
564      A42=RZ23/R(2)
565      A43=RZ31/R(3)
566      IF(R(1).NE.R(2))
567      *A41=RZ12/R12=ALOG(R1R2)
568      IF(R(2).NE.R(3))
569      *A42=RZ23/R23=ALOG(R2R3)
570      IF(R(3).NE.R(1))
571      *A43=RZ31/R31=ALOG(R3R1)
572 C      *****B
573      IF (IRFLG.EQ.0) GO TO 1000
574      GO TO (1,2,3,4,5,6), IRFLG
575      1 A43=0.0
576      A41=-Z(1)=ALOG(R2R3)
577      GO TO 1000
578      2 A41=0.0
579      A42=-Z(2)=ALOG(R3R1)
580      GO TO 1000
581      3 A42=0.0
582      A43=-Z(3)=ALOG(R1R2)
583      GO TO 1000
584      4 A42=-Z23
585      GO TO 1000
586      5 A43=-Z31
587      GO TO 1000
588      6 A41=-Z12
589      1000 CONTINUE
590 C      *****E
591      A4=A41+A42+A43
592 C      IF (IRFLG.EQ.3.OR.IRFLG.EQ.6) WRITE(5,1212)A41,A42,A43,A4
593      A51=0.25-Z12*(Z(1)+3.*Z(2))+Z(2)=RZ12/R(1)-0.25*(RZ12/R(1))-2
594      A52=0.25-Z23*(Z(2)+3.*Z(3))+Z(3)=RZ23/R(2)-0.25*(RZ23/R(2))-2
595      A53=0.25-Z31*(Z(3)+3.*Z(1))+Z(1)=RZ31/R(3)-0.25*(RZ31/R(3))-2
596      IF(R(1).NE.R(2))
597      *A51=0.25-Z12/R12*(Z(1)-(3.*R(2)-R(1))-Z(2)-(3.*R(1)-R(2)))
598      + 0.5*(RZ12/R12)-2=ALOG(R1R2)
599      IF(R(2).NE.R(3))
600      *A52=0.25-Z23/R23*(Z(2)-(3.*R(3)-R(2))-Z(3)-(3.*R(2)-R(3)))
601      + 0.5*(RZ23/R23)-2=ALOG(R2R3)
602      IF(R(3).NE.R(1))
603      *A53=0.25-Z31/R31*(Z(3)-(3.*R(1)-R(3))-Z(1)-(3.*R(3)-R(1)))
604      + 0.5*(RZ31/R31)-2=ALOG(R3R1)
605 C      *****B
606      IF (IRFLG.EQ.0) GO TO 2000
607      GO TO (11,12,13,14,15,16), IRFLG
608      11 A53=0.0
609      A51=.25*(-Z23*(2.*Z(1)+Z(2)+Z(3))-2.*(Z(1)*Z(1)=ALOG(R2R3)))
610      GO TO 2000
611      12 A51=0.0
612      A52=.25*(-Z31*(2.*Z(2)+Z(3)+Z(1))-2.*(Z(2)*Z(2)=ALOG(R3R1)))
613      GO TO 2000
614      13 A52=0.0
615      A53=.25*(-Z12*(2.*Z(3)+Z(1)+Z(2))-2.*(Z(3)*Z(3)=ALOG(R1R2)))
616      GO TO 2000
617      14 A52=0.0
618      GO TO 2000
619      15 A53=0.0
620      GO TO 2000
621      16 A51=0.0
622      2000 CONTINUE
623 C      *****EE
624      A5=A51+A52+A53
625 C      IF (IRFLG.EQ.3.OR.IRFLG.EQ.6) WRITE(5,1212)A51,A52,A53,A5
626      A61=Z12/36.*(22.*Z(2)-2+10.*Z(1)*Z(2)+4.*Z(1)-2)
627      + Z(2)-2=Z12/R(1)-0.5*Z(2)=(RZ12/R(1))-2+(RZ12/R(1))-3/9.
628      A62=Z23/36.*(22.*Z(3)-2+10.*Z(1)*Z(3)+4.*Z(2)-2)
629      + Z(3)-2=Z23/R(2)-0.5*Z(3)=(RZ23/R(2))-2+(RZ23/R(2))-3/9.
630      A63=Z31/36.*(22.*Z(1)-2+10.*Z(3)*Z(1)+4.*Z(3)-2)
631      + Z(1)-2=Z31/R(3)-0.5*Z(1)=(RZ31/R(3))-2+(RZ31/R(3))-3/9.
632      IF(R(1).NE.R(2))
633      *A61=Z12/R12-2*(Z(2)-2*(11.*R(1)-2-7.*R(1)*R(2)+2.*R(2)-2)
634      + 2.*Z(1)*Z(2)-(2.5*R(1)-2-11.*R(1)*R(2)+2.5*R(2)-2)
635      + Z(1)-2*(11.*R(2)-2-7.*R(1)*R(2)+2.*R(1)-2))/18.
636      - (RZ12/R12)-3=ALOG(R1R2)/3.
637      IF(R(2).NE.R(3))
638      *A62=Z23/R23-2*(Z(3)-2*(11.*R(2)-2-7.*R(2)*R(3)+2.*R(3)-2)
639      + 2.*Z(2)*Z(3)-(2.5*R(2)-2-11.*R(2)*R(3)+2.5*R(3)-2)
640      + Z(2)-2*(11.*R(3)-2-7.*R(2)*R(3)+2.*R(2)-2))/18.

```

```

641      +(R223/R23)**3=ALOG(R2R3)/3.
642      IF(R(3) .NE. R(1))
643      *A63=Z31/R31**2*(Z(1)**2*(11.*R(3)**2-7.*R(3)*R(1)+2.*R(1)**2)
644      + 2.*Z(3)*Z(1)*(2.5*R(3)**2-11.*R(3)*R(1)+2.5*R(1)**2)
645      + Z(3)**2*(11.*R(1)**2-7.*R(3)*R(1)+2.*R(3)**2))/18.
646      - +(R231/R31)**3=ALOG(R3R1)/3.
647 C      *****B
648      IF (IRFLG.EQ.0) GO TO 3000
649      GO TO (21,22,23,24,25,26), IRFLG
650 21 A63=0.0
651      A61=(1./18.)*(-Z23*(2.*(Z(2)*Z(2)+Z(2)*Z(3)+Z(3)*Z(3))
652      1+6.*Z(1)*Z(1)+3.*Z(1)*(Z(2)+Z(3))+6.*(Z(1)*Z(1)*Z(1))=ALOG(R2R3)))
653      GO TO 3000
654 22 A61=0.0
655      A62=(1./18.)*(-Z31*(2.*(Z(3)*Z(3)+Z(3)*Z(1)+Z(1)*Z(1))
656      1+6.*Z(2)*Z(2)+3.*Z(2)*(Z(3)+Z(1))+6.*(Z(2)*Z(2)*Z(2))=ALOG(R3R1)))
657      GO TO 3000
658 23 A62=0.0
659      A63=(1./18.)*(-Z12*(2.*(Z(1)*Z(1)+Z(1)*Z(2)+Z(2)*Z(2))
660      1+6.*Z(3)*Z(3)+3.*Z(3)*(Z(1)+Z(2))+6.*(Z(3)*Z(3)*Z(3))=ALOG(R1R2)))
661      GO TO 3000
662 24 A62=0.0
663      GO TO 3000
664 25 A63=0.0
665      GO TO 3000
666 26 A61=0.0
667 3000 CONTINUE
668 C      *****E
669      A6=A61+A62+A63
670 C      IF (IRFLG.EQ.3.OR.IRFLG.EQ.6) WRITE(5,1212)A61,A62,A63,A6
671 1212 FORMAT('      AFTER CHANGE      ',5E15.6)
672      RETURN
673      END

```

```

1      SUBROUTINE MINVRS(R,Z,AB)
2 C
3      COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
4      DIMENSION R(1),Z(1),AB(6,6)
5      DO 103 I=1,6
6      DO 103 J=1,6
7 103  AB(I,J)=0.0
8      R12=R(1)-R(2)
9      R23=R(2)-R(3)
10     R31=R(3)-R(1)
11     Z12=Z(1)-Z(2)
12     Z23=Z(2)-Z(3)
13     Z31=Z(3)-Z(1)
14     RZ12=R(1)*Z(2)-R(2)*Z(1)
15     RZ23=R(2)*Z(3)-R(3)*Z(2)
16     RZ31=R(3)*Z(1)-R(1)*Z(3)
17     AB(1,1)=RZ23/BETA
18     AB(2,1)=Z23/BETA
19     AB(3,1)=-R23/BETA
20     AB(4,2)=AB(1,1)
21     AB(5,2)=AB(2,1)
22     AB(6,2)=AB(3,1)
23     AB(1,3)=RZ31/BETA
24     AB(2,3)=Z31/BETA
25     AB(3,3)=-R31/BETA
26     AB(4,4)=AB(1,3)
27     AB(5,4)=AB(2,3)
28     AB(6,4)=AB(3,3)
29     AB(1,5)=RZ12/BETA
30     AB(2,5)=Z12/BETA
31     AB(3,5)=-R12/BETA
32     AB(4,6)=AB(1,5)
33     AB(5,6)=AB(2,5)
34     AB(6,6)=AB(3,5)
35 C      WRITE(5,111)
36 111  FORMAT(//,10X,'A MATRIX',//)
37 C      CALL PRMT(6,6,AB)
38      RETURN
39      END

```

```

1      SUBROUTINE DFCMTX(AB)
2 C
3      COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
4      DIMENSION AB(6,4)
5      DO 637 I=1,6
6      DO 637 J=1,6
7 637   AB(I,J)=0.0
8       AB(1,2)=A2
9       AB(2,1)=A1
10      AB(2,2)=A1
11      AB(3,2)=A3
12      AB(3,4)=A1
13      AB(5,4)=A1
14      AB(6,3)=A1
15      DO 640 K=1,6
16      DO 640 L=1,6
17      AB(K,L)=AB(K,L)+2.*3.14159
18 640   CONTINUE
19 C      WRITE(5,1986)
20 1986  FORMAT(//,10X,'AB MATRIX',//)
21 C      CALL PRMTM(6,4,AB)
22      RETURN
23      END
24      SUBROUTINE INTGR2(R,Z,Y,I,J,K)
25 C
26      COMMON /AAA/ A1,A2,A3,A4,A5,A6,BETA
27      DIMENSION R(1),Z(1),Y(1),B(6,6),YY(6),BT(6,6)
28      R1=R(J)-R(I)
29      R2=R(J)+2-R(I)+2
30      R3=R(J)+3-R(I)+3
31      Z1=Z(J)-Z(I)
32      Z2=Z(J)+2-Z(I)+2
33      Z3=Z(J)+3-Z(I)+3
34      IF(ABS(R1).LE.0.0001) GO TO 300
35      S1=Z1/R1
36      C1=Z(I)-S1-R(I)
37      GO TO 400
38 300   CONTINUE
39      S1=0.0
40      C1=0.0
41 400   CONTINUE
42      S4=R1/Z1
43      C4=R(I)-S4-Z(I)
44      YY(1)=-C4-Z1-0.5*S4-Z2
45      YY(2)=-C4-C4-Z1-C4-S4-Z2-S4-S4/3.*Z3
46      YY(3)=-0.5-C4-Z2-S4/3.*Z3
47      YY(4)=0.5-R2
48      YY(5)=R3/3.
49      YY(6)=0.5-C1-R2+S1/3.*R3
50      DO 8000 L=1,6
51      YY(L)=YY(L)+2.*3.141592654
52 8000  CONTINUE
53      Z23=Z(2)-Z(3)
54      Z31=Z(3)-Z(1)
55      Z12=Z(1)-Z(2)
56      BETA=R(1)+Z23+R(2)+Z31+R(3)+Z12
57      CALL MINVRS(R,Z,B)
58      DO 100 IA=1,6
59      DO 100 JA=1,6
60 100   BT(JA,IA)=B(IA,JA)
61      DO 200 IA=1,6
62      Y(IA)=0.0
63      DO 200 JA=1,6
64 200   Y(IA)=Y(IA)+BT(IA,JA)*YY(JA)
65 C      WRITE(6,60)(Y(L),L=1,6)
66 60    FORMAT(//,10X,'Y VECTOR',//,6E12.3)
67      RETURN
68      END
69      SUBROUTINE PRSLDS (NOD,FLOAD,ID,NEQ,ICALL)
70 C
71 C
72 C
73 C      PROGRAM
74 C      . TO CALCULATE THE PRESSURE LOADS FOR EACH ELEMENT
75 C      . AND ADD IT TO THE GLOBAL LOAD VECTOR R (COMPUTED
76 C      . IN SUBROUTINE LOADS)
77 C      . AND WRITE RESULTING LOAD VECTOR ONTO TAPE ILOAD
78 C
79 C
80      COMMON /VAR/ NG,MODEX,NLOAD

```

```

81      COMMON /TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
82      DIMENSION R(200),NOD(1),IDIRN(2),FLOAD(1)
83      DIMENSION ID(3,1)
84 C
85 C
86 C      OPEN(UNIT=30,DEVICE='DSK',ACCESS='SEQINOUT',FILE='ILOAD.SAP')
87 C      OPEN (UNIT=30,NAME='ILOAD.SAP',STATUS='UNKNOWN',
88 C      + FORM='UNFORMATTED')
89      REWIND ILOAD
90      IF(ICALL.EQ. 1 .AND. NLOAD.EQ. 0) GO TO 101
91      READ (ILOAD) R
92      GO TO 103
93      101 DO 102 L=1,NEQ
94      102 R(L)=0.0
95      103 CONTINUE
96 C      WRITE(6,632) (R(L),L=1,NEQ)
97      IDIRN(1)=1
98      IDIRN(2)=2
99 C
100      DO 220 L=1,2
101      LI=IDIRN(L)
102      IF(LI.EQ.1) K=-1
103      IF(LI.EQ.2) K=0
104      DO 220 J=1,3
105      K=K+2
106      LN=NOD(J)
107      II=ID(LI,LN)
108 C      WRITE(6,7675)II
109      7675 FORMAT(//,I10)
110      IF (II) 220,220,240
111      240 R(II)=R(II) + FLOAD(K)
112 C
113      220 CONTINUE
114 C
115 C      WRITE(6,631) (FLOAD(L),L=1,6)
116      631 FORMAT(//,10X,'FLOAD',//,(6E15.5))
117 C      WRITE(6,632)(R(L),L=1,NEQ)
118      REWIND ILOAD
119      WRITE(ILOAD) (R(L),L=1,NEQ)
120      632 FORMAT(//,10X,'LOAD VECTOR',//,(5E15.5))
121 C
122 C
123 C      CLOSE(UNIT=30,DEVICE='DSK',ACCESS='SEQINOUT',FILE='ILOAD.SAP')
124 C      CLOSE (UNIT=30)
125      RETURN
126      END
127      SUBROUTINE TRUSS
128      RETURN
129      END
130      SUBROUTINE RUSS (ID,X,Y,Z,U,MHT,E,AREA,LM,XYZ,MATP)
131      RETURN
132      END
133      SUBROUTINE CREEPL (ID,MAXA,AKMAT,VECT,XYZ,SIGT,
134      1 EPSLON,SIGOO,ALFOO,EPDIF,NWK,NUMEL,NUMNP,NEQ)
135 C-----
136 C
137 C      CREEPL:  A ROUTINE TO CONTROL THE 'STAP' PROGRAM FINITE-
138 C      ELEMENT SOLUTION OF A UNIFIED CREEP-PLASTICITY PROBLEM
139 C      IN AN AXISYMMETRIC BODY.
140 C
141 C-----
142 C
143      COMMON/AAA/ A1,A2,A3,A4,A5,A6,BETA
144      COMMON/DIM/ N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
145      COMMON/TAPES/ IELMNT,ILOAD,IIN,IOUT,ISTRES,IDISP,IEPS,ISIG
146      COMMON/EL/ INO,NPAR(10),NUMEG,MTOT,NFIRST,NLAST,ITWO
147      COMMON/INTDAT/ YMU,XKAPPA,HRD,RCV,BETA2,GO,XN,XM
148      COMMON/EXMU/ PROP(10,10)
149      COMMON/NNN/ NODNUM(3500,10)
150      COMMON/TIME/TSSIZE,TYME,ABSALF,INUM,ITRB,ICOUNT
151      COMMON/NULOAD/NULOAD
152 C
153      DIMENSION SIGPRT(4),EPDIF(NUMEL,4),DEPS(4)
154      DIMENSION DELTFC(6),AB(6,4),EPSLON(NUMEL,4),VECT(NEQ),
155      1 SIGO(4),SIG1(4),EE(16),UD(6),SID(4),ALFO(4),ALF1(4),
156      2 EPSO(4),EPS1(4),RR(3),ZZ(3),CFS(6),SIGY(4),EPSY(4),
157      3 AKMAT(NWK),SIGT(NUMEL,4),ID(3,NUMNP),MAXA(1),SIG(4),XYZ(6,1),
158      4 ALFOO(NUMEL,4),SIGOO(NUMEL,4),D(3),ITRB(4),ABSALF(4),
159      5 DELTF(6),B(6,6),BTRN(6,6),SSIDD(4)
160 C

```

```

161      WRITE(1,*) NULOAD
162      IF(NULOAD.GE.2) GO TO 9099
163 C      OPEN(UNIT=35,DEVICE='DSK',MODE='ASCII',ACCESS='SEQOUT',
164 C      1  DISPOSE='SAVE',FILE='RESULT.DAT')
165      OPEN (UNIT=36,NAME='RESULT.DAT',STATUS='UNKNOWN')
166 9099      CONTINUE
167      GGG=ENNN/(2.*(1.+XMUNNN))
168      ICOUNT=0
169      ICREEP=1
170      TYME=0.
171      IF(NULOAD.GE.2) TYME=0.0000001
172 C-----
173 C
174 C      FROM THE INPUT FILE, READ IN THE NO. OF TIME- STEPS
175 C      AND THE TIME-STEP SIZE.
176 C
177 C-----
178      READ(IIN,101)NTSTEP,TSSIZE
179      101  FORMAT(I5,F10.2)
180 C-----
181 C
182 C      READ IN ELASTIC STRESSES AND STORE IN THE ARRAY SIGT
183 C
184 C-----
185 C      OPEN(UNIT=41,DEVICE='DSK',MODE='ASCII',ACCESS='SEQIN',
186 C      1  DISPOSE='SAVE',FILE='STRESS.DAT')
187      OPEN (UNIT=41,NAME='STRESS.DAT',STATUS='UNKNOWN')
188      DO 2000 I=1,NUMEL
189      READ(41,9009) (SIG(IK),IK=1,4)
190 9009      FORMAT(4E15.4)
191      DO 1000 J=1,4
192      SIGT(I,J)=SIG(J)+SIGT(I,J)
193 1000      CONTINUE
194 2000      CONTINUE
195      READ(41,9010) ((XYZ(IJ,IK),IJ=1,6),IK=1,NUMEL)
196 9010      FORMAT(6E15.4)
197      DO 2100 II=1,NUMEL
198 C      WRITE(5,2101) II,(XYZ(L,II),L=1,6)
199 2101      FORMAT(' ',I3,1X,6(1PE10.2))
200 2100      CONTINUE
201 C      CLOSE(UNIT=41,DEVICE='DSK',MODE='ASCII',ACCESS='SEQIN',
202 C      1  DISPOSE='DELETE',FILE='STRESS.DAT')
203      CLOSE (UNIT=41)
204 C-----
205 C
206 C      FROM THE INPUT FILE, READ THE CONTROL PARAMETERS FOR THE CREEP
207 C      ANALYSIS.
208 C
209 C-----
210      READ(IIN,102) YMU,XKAPPA,HRD,RCV,BETA2,GO,XN,XM
211 102      FORMAT(E10.2,F10.2,2E10.2,F10.2,E10.2,2F5.2)
212 C      TYPE 115, YMU,XKAPPA,HRD,RCV,BETA2,GO,XN,XM
213 C      WRITE(IOUT,115) YMU,XKAPPA,HRD,RCV,BETA2,GO,XN,XM
214 115      FORMAT('1',10X,'CREEP PARAMETERS:'.//)
215      1  10X,'MU'   ='',1PE10.2,//
216      2  10X,'KAPPA' ='',1PE10.2,//
217      3  10X,'HRD'   ='',1PE10.2,//
218      4  10X,'RCV'   ='',1PE10.2,//
219      5  10X,'BETA'  ='',1PE10.2,//
220      6  10X,'GO'    ='',1PE10.2,//
221      7  10X,'NN'    ='',1PE10.2,//
222      8  10X,'MM'    ='',1PE10.2,//
223 C-----
224 C
225 C      BEGIN CREEP-PLASTIC ANALYSIS:
226 C      1.  COMPUTE CREEP STRAIN FOR EACH ELEMENT.
227 C      2.  COMPUTE CORRESPONDING INCREMENTAL LOAD FOR EACH
228 C      ELEMENT AND INSERT IN LOAD VECTOR.
229 C
230 C-----
231 2999      ICOUNT=ICOUNT+1
232      TYPE 955, TYME,ICOUNT
233 955      FORMAT(/,' TIME = ',1PE10.3,' HOURS.','      TIMESTEP NO. = ',
234      1  I4,/)
235      WRITE(35,203)ICOUNT,TYME
236 203      FORMAT(I5,F10.5)
237 C      WRITE(IOUT,103)ICOUNT,TYME
238 103      FORMAT('1TIME STEP NO.',I4,' TIME = ',F12.5)
239 C
240      DO 2500 IJ=1,NEQ

```

```

241 2500 VECT(IJ)=0.
242 C
243 IXX=0
244 3999 IXX=IXX+1
245 INUM=IXX
246 IF(TYME.GT.O.) GC TO 3001
247 DO 2900 J=1,4
248 EPSLON(IXX,J)=0.
249 ALFOO(IXX,J)=0.
250 SIGOO(IXX,J)=0.
251 2900 CONTINUE
252 3001 DO 3000 J=1,4
253 SID(J)=SIGT(IXX,J)
254 SSIDD(J)=SID(J)
255 EPSO(J)=EPSLON(IXX,J)
256 ALFO(J)=ALFOO(IXX,J)
257 SIGO(J)=SIGOO(IXX,J)
258 3000 CONTINUE
259 C
260 C WRITE(IOUT,915) IXX, (SID(KK), KK=1,4)
261 915 FORMAT(///, ' ELEMENT ', I5, ' INPUT STRESSES = ', 4(1PE10.2))
262 WRITE(35,204) ALFO(1), SSIDD(1), ALFO(2), SSIDD(2),
263 1 ALFO(3), SSIDD(3)
264 C
265 CALL INTVLU(SID, ALFO, ALF1, EPSO, EPS1, SIGO, SIG1)
266 C
267 204 FORMAT(6(E12.3))
268 C
269 C
270 C WRITE(IOUT,114) INUM
271 114 FORMAT(///, ' ', ' ELEMENT NO. ', I4)
272 C
273 C
274 C WRITE(IOUT,104)
275 104 FORMAT(/, ' ', 8X, 'SID', 7X, 'ALFO', 7X, 'ALF1', 7X, 'EPSO',
276 1 7X, 'EPS1', 7X, 'SIGO', 7X, 'SIG1', 7X, 'ITER', 5X, 'ABDOLF', //)
277 DO 3100 K=1,4
278 C WRITE(IOUT,105) SID(K), ALFO(K), ALF1(K), EPSO(K),
279 C 1 EPS1(K), SIGO(K), SIG1(K), ITRB(K), ABSALF(K)
280 105 FORMAT(' ', 7(1X, 1PE10.2), 6X, I5, 1X, 1PE10.2)
281 C WRITE(5,910) K, EPS1(K)
282 910 FORMAT(' ', I1, ') = ', 1X, 1PE10.2)
283 EPSLON(IXX,K)=EPS1(K)
284 EPSDIF(IXX,K)=EPS1(K)-EPSO(K)
285 ALFOO(IXX,K)=ALF1(K)
286 SIGOO(IXX,K)=SIG1(K)
287 3100 CONTINUE
288 C
289 DO 3105 L=1,3
290 LK=2*(L-1)+1
291 RR(L)=XYZ(LK, IXX)
292 ZZ(L)=XYZ(LK+1, IXX)
293 C WRITE(5,912) IXX, L, RR(L), ZZ(L)
294 912 FORMAT(' ', 2I5, 2(1X, 1PE10.2))
295 3105 CONTINUE
296 CALL INTGR1(RR, ZZ)
297 CALL DFCMTX(AB)
298 CALL MINVRS(RR, ZZ, B)
299 DO 3110 II=1,6
300 DO 3110 JJ=1,6
301 BTRN(II, JJ)=B(JJ, II)
302 3110 CONTINUE
303 DO 3200 J=1,6
304 DELTFC(J)=0.
305 DO 3200 K=1,4
306 C WRITE(5,911) J, K, AB(J, K)
307 911 FORMAT(' ', 2I5, 1PE10.2)
308 DEPS(K)=EPSDIF(IXX, K)
309 DELTFC(J)=DELTFC(J)+2.*GGG*AB(J, K)*DEPS(K)
310 3200 CONTINUE
311 C IF(TYME.EQ.O.) TYPE 902, (DELTFC(L), L=1,6), IXX
312 902 FORMAT(' DFC = ', 6(1PE10.2), ' EL = ', I4)
313 C IF(TYME.EQ.O.) TYPE 952, (DEPS(IK), IK=1,4)
314 952 FORMAT(' DEPS = ', 4(1PE10.2))
315 C TYPE 953, GGG
316 953 FORMAT(' G = ', 1PE10.2)
317 DO 3201 J=1,6
318 DELTF(J)=0.
319 DO 3201 K=1,6
320 DELTF(J)=DELTFC(J)+BTRN(J, K)*DELTFC(K)

```

```

321 3201 CONTINUE
322 C IF(TYME.EQ.O.)TYPE 902.(DETF(L),L=1,6).IXX
323 C
324 IJK=(INUM-1)+3
325 II=NODNUM(IJK+1,NELG)
326 JJ=NODNUM(IJK+2,NELG)
327 KK=NODNUM(IJK+3,NELG)
328 C
329 I1=ID(1,II)
330 IF(I1.NE.O) VECT(I1)=VECT(I1)+DETF(1)
331 I2=ID(2,II)
332 IF(I2.NE.O) VECT(I2)=VECT(I2)+DETF(2)
333 J1=ID(1,JJ)
334 IF(J1.NE.O) VECT(J1)=VECT(J1)+DETF(3)
335 J2=ID(2,JJ)
336 IF(J2.NE.O) VECT(J2)=VECT(J2)+DETF(4)
337 K1=ID(1,KK)
338 IF(K1.NE.O) VECT(K1)=VECT(K1)+DETF(5)
339 K2=ID(2,KK)
340 IF(K2.NE.O) VECT(K2)=VECT(K2)+DETF(6)
341 C
342 IF(IXX.LT.NUMEL) GO TO 3999
343 DO 3205 JJ=1,NEQ
344 C IF(TYME.EQ.O.)TYPE 903,JJ,VECT(JJ)
345 903 FORMAT(' ',I10,5X,1PE10.2)
346 3205 CONTINUE
347 C-----
348 C
349 C SOLVE THE EQUATION SYSTEM USING THE VECTOR VECT WHICH
350 C CONTAINS THE INCREMENTAL LOADS. THE SOLUTION WILL BE
351 C THE INCREMENTAL DISPLACEMENTS.
352 C
353 C-----
354 C *****
355 C REDUCE THE RIGHT HAND SIDE LOAD VECTOR.
356 C *****
357 DO 5000 J=1,NEQ
358 KL=MAXA(J)+1
359 KU=MAXA(J+1)-1
360 IF(KU-KL)5000,4100,4100
361 4100 K=J
362 C=O.
363 DO 4200 KKK=KL,KU
364 K=K-1
365 4200 C=C+AKMAT(KKK)=VECT(K)
366 VECT(J)=VECT(J)-C
367 5000 CONTINUE
368 C *****
369 C BACK SUBSTITUTE ; VECT NOW CONTAINS DISPLACEMENTS.
370 C *****
371 DO 6000 J=1,NEQ
372 K=MAXA(J)
373 6000 VECT(J)=VECT(J)/AKMAT(K)
374 J=NEQ
375 DO 6300 L=2,NEQ
376 KL=MAXA(J)+1
377 KU=MAXA(J+1)-1
378 IF(KU-KL)6300,6100,6100
379 6100 K=J
380 DO 6200 KKK=KL,KU
381 K=K-1
382 6200 VECT(K)=VECT(K)-AKMAT(KKK)=VECT(J)
383 6300 J=J-1
384 C-----
385 C
386 C PRINT OUT THE INCREMENTAL DISPLACEMENTS FOR THIS TIME STEP.
387 C
388 C-----
389 C WRITE(IOUT,106) ICOUNT
390 106 FORMAT('1TIMESTEP NO. ',I3,' INCREMENTAL DISPLACEMENTS ',
391 1 //)
392 C WRITE(IOUT,107)
393 107 FORMAT('///',' ',6X,'NODE',14X,'R',14X,'Z',10X,'THETA',/)
394 C
395 DO 6350 J=1,NUMNP
396 DO 6330 K=1,3
397 D(K)=O.
398 6330 CONTINUE
399 DO 6340 L=1,3
400 KK=ID(L,J)

```

```

401      IF(KK.NE.O) D(L)=VECT(KK)
402      CONTINUE
403 C      WRITE(IOUT,108) J,(D(JX),JX=1,3)
404      FORMAT(' ',10,3(5X,1PE10.3))
405      6350      CONTINUE
406 C-----
407 C
408 C      COMPUTE INCREMENTAL STRAIN AND THEN INCREMENTAL STRESS
409 C      FOR EACH ELEMENT.
410 C
411 C-----
412      DO 6400 J=1,16
413      6400      EE(J)=0.
414      EE(1)=1.-XMUNNN
415      EE(2)=XMUNNN
416      EE(3)=XMUNNN
417      EE(5)=XMUNNN
418      EE(6)=1.O-XMUNNN
419      EE(7)=XMUNNN
420      EE(9)=XMUNNN
421      EE(10)=XMUNNN
422      EE(11)=1.O-XMUNNN
423      EE(16)=0.5-XMUNNN
424      DO 6500 J=1,16
425      6500      EE(J)=EE(J)+ENNN/((1.+XMUNNN)-(0.5-XMUNNN)*2.)
426 C
427      DO 7000 I=1,NUMEL
428      DO 6600 J=1,3
429      K=2+(J-1)+1
430      RR(J)=XYZ(K,I)
431      6600      ZZ(J)=XYZ(K+1,I)
432      I1=0
433      IJK=(I-1)*3
434 C
435      DO 6700 J=1,3
436      DO 6700 JJ=1,2
437      I1=I1+1
438      K=NODNUM(IJK+J,NELG)
439      KK=ID(JJ,K)
440      IF(KK.EQ.O) GO TO 6650
441      UD(I1)=VECT(KK)
442      GO TO 6700
443      6650      UD(I1)=0.
444      6700      CONTINUE
445 C
446      CALL INTGR1(RR,ZZ)
447      CALL STRSN(RR,ZZ,UD,EE,EPSY,SIGY)
448 C-----
449 C      SUBTRACT HYDROSTATIC COMPONENT FROM STRESS.
450 C-----
451      DO 6800 J=1,4
452      DEPS(J)=EPSDIF(I,J)
453      EEPS=2.*GGG-DEPS(J)
454 C      IF(I.EQ.1) TYPE 951,SIGY(J),EEPS
455      951      FORMAT(' SIGY(J)=',1PE10.2,' EEPS=',1PE10.2)
456      SIGY(J)=SIGY(J)-(2.*GGG-DEPS(J))
457      6800      CONTINUE
458 C-----
459 C      ADD THIS STRESS TO THE PREVIOUSLY ACCUMULATED
460 C      STRESS TO GET THE CURRENT TOTAL.
461 C-----
462      DO 6900 J=1,4
463      SIGT(I,J)=SIGT(I,J)+SIGY(J)
464      6900      CONTINUE
465      7000      CONTINUE
466      DO 9000 I=1,NUMEL
467      DO 8000 J=1,4
468      SIGPRT(J)=SIGT(I,J)
469      8000      CONTINUE
470      IF(I.EQ.1.OR.I.EQ.NUMEL)WRITE(5,901) I,(SIGPRT(L),L=1,4)
471      901      FORMAT(' ELEMENT (' ,I4,1X,') SIGT= ',4(2X,1PE10.3))
472      9000      CONTINUE
473 C-----
474 C
475 C      CHECK TO SEE WHETHER THE SPECIFIED NO. OF TIME-
476 C      STEPS HAS BEEN REACHED AND EITHER REPEAT THE
477 C      CREEP-PLASTIC SEQUENCE IF NOT OR EXIT IF SO.
478 C
479 C-----
480      IF(ICOUNT.EQ.NTSTEP) GO TO 9999

```

```

481      IF(NULOAD.GE.2) GO TO 9909
482      IF(TYME.GE.0.002.AND.TYME.LT.2.) TSSIZE=TSSIZE+2.
483      IF(TYME.GE.2..AND.TYME.LT.10.) TSSIZE=1.
484      IF(TYME.GE.10..AND.TYME.LT.30.) TSSIZE=5.
485      IF(TYME.GE.30..AND.TYME.LT.60.) TSSIZE=1.0
486      IF(TYME.GE.60..AND.TYME.LT.80.) TSSIZE=1.0
487      IF(TYME.GE.80.AND.TYME.LT.1000.) TSSIZE=10.
488      GO TO 9110
489 9909      CONTINUE
490      IF(NULOAD.GE.3) GO TO 9919
491      IF(TYME.GE.0.005.AND.TYME.LT.2.) TSSIZE=TSSIZE+5.
492      IF(TYME.GE.2..AND.TYME.LT.10.) TSSIZE=1.
493      IF(TYME.GE.10..AND.TYME.LT.30.) TSSIZE=5.
494      IF(TYME.GE.30..AND.TYME.LT.60.) TSSIZE=1.0
495      IF(TYME.GE.60..AND.TYME.LT.80.) TSSIZE=1.0
496      IF(TYME.GE.80.AND.TYME.LT.1000.) TSSIZE=10.
497      GO TO 9110
498 9919      CONTINUE
499      IF(TYME.GE.0.005.AND.TYME.LT.2.) TSSIZE=TSSIZE+5.
500      IF(TYME.GE.2..AND.TYME.LT.10.) TSSIZE=1.
501      IF(TYME.GE.10..AND.TYME.LT.30.) TSSIZE=5.
502      IF(TYME.GE.30..AND.TYME.LT.60.) TSSIZE=1.0
503      IF(TYME.GE.60..AND.TYME.LT.80.) TSSIZE=1.0
504      IF(TYME.GE.80.AND.TYME.LT.1000.) TSSIZE=10.
505 9110      TYME=TYME+TSSIZE
506      GO TO 2999
507 9999      RETURN
508      END
509      SUBROUTINE INTVLU(SID,ALFO,ALF1,EPSO,EPS1,SIGO,SIG1)
510      COMMON/INTDAT/ YMU,XKAPA,HRD,RCV,BETA,GO,XN,XM
511      COMMON/TIME/H, TYME,ABSALF,INUM,ITR8,ICOUNT
512      DIMENSION SID(1),ALFO(1),ALF1(1),EPSO(1),EPS1(1),SIGO(1),SIG1(1)
513      DIMENSION ITR8(4),ABSALF(4)
514 C
515 C      INITIALIZATION
516 C
517      EPS=0.001
518      THETA=0.5
519      IF(TYME.GT.0.) GO TO 8888
520      DQ 8765 I=1,4
521      ALFO(I)=SORT(GO)
522      EPSO(I)=0.0
523 8765      SIGO(I)=SID(I)
524 C
525 C      COMPUTE EXPONENTS
526 C
527 8888      P1=(XN-1.)/2.
528      P2=BETA/2.
529      P3=(XM-BETA-1.)/2.
530 C
531 C      COMPUTE DEVIATORIC STRESSES
532 C
533      SNN=(SID(1)+SID(2)+SID(3))/3.
534      DO 10 I=1,3
535 10      SID(I)=SID(I)-SNN
536 C
537 C      START INTEGRATION PROCESS
538 C
539 C      SET FIRST ESTIMATE EQUAL TO PREVIOUS VALUES
540 C
541 C
542 1111      DO 15 I=1,4
543      SIG1(I)=SIGO(I)
544      ALF1(I)=ALFO(I)
545      EPS1(I)=EPSO(I)
546 15      CONTINUE
547 C
548 C      COMPUTE FO AND GO AT STATION "O"
549 C
550      DO 20 I=1,4
551 20      SIGO(I)=SID(I)-ALFO(I)
552      XJ2=0.0
553      DO 30 I=1,4
554 30      XJ2=XJ2+SIGO(I)*SIGO(I)
555      XJ2=0.5*XJ2
556      FO=XJ2/(XKAPA*XKAPA)-1.0
557      YJ2=0.0
558      DO 40 I=1,4
559 40      YJ2=YJ2+ALFO(I)*ALFO(I)
560      YJ2=0.5*YJ2

```

```

561      GO=YJ2/(XKAPA=XKAPA)
562 C
563 C      START VECTOR ELEMENTS LOOP
564 C-----
565      SIDALF=0.0
566      SIDSIG=0.0
567      DO 50 I=1,4
568      SIDALF=SIDALF+SID(I)*ALF1(I)
569      SIDSIG=SIDSIG+SID(I)*SIG1(I)
570 50    CONTINUE
571 C-----
572 C
573      DO 333 IIJ=1,4
574      ITER=0
575 C
576 C      START ITERATION LOOP
577 C
578 222   ITER=ITER+1
579 C
580 C      CHECK ON GO AND FO
581 C
582      IF(ITER.GE.20)WRITE(5,700) GO
583      IF(ITER.GE.20)WRITE(5,701) FO
584 700 FORMAT(//,10X,'GO=',1PE15.5)
585 701 FORMAT(//,10X,'FO=',1PE15.5)
586      IF(ITER.GT. 1) GO TO 45
587      IF(GO .LE. GO) GO TO 56
588      IF(SIDALF .LE. 0.0) GO TO 56
589      WO=GO
590      GO TO 77
591 56    WO=GO
592 77    CONTINUE
593      IF(FO .LE. 0.0) GO TO 11
594      IF(SIDALF .LE. 0.0) GO TO 23
595      IF(SIDSIG .GT. 0.0) GO TO 23
596 11    EPSDTO=0.0
597      GO TO 44
598 23    EPSDTO=FO==P1-SIGO(IIJ)
599 44    CONTINUE
600      HWFO=HRD/WO==P2-EPSDTO/SIGO(IIJ)
601      RWO=RCV/WO==P3
602 45    CONTINUE
603 C
604 C      COMPUTE F1 AND G1 AT STATION "1"
605 C
606      DO 120 I=1,4
607 120   SIG1(I)=SID(I)-ALF1(I)
608      IF(ITER.GE.20) TYPE 7774
609 7774   FORMAT('      SID      ALF1      SIG1',/)
610      DO 121 JJ=1,4
611      IF(ITER.GE.20) TYPE 7775,SID(JJ),ALF1(JJ),SIG1(JJ)
612 7775   FORMAT(' ',3(1PE10.2))
613 121    CONTINUE
614      XJ2=0.0
615      DO 130 I=1,4
616 130    XJ2=XJ2+SIG1(I)=SIG1(I)
617      XJ2=0.5*XJ2
618      F1=XJ2/(XKAPA=XKAPA)-1.0
619      IF(ITER.GE.20)TYPE 7782,XJ2,F1
620 7782   FORMAT(' STATION 1',1X,'XJ2=',1PE10.2,1X,'F1=',1PE10.2)
621      YJ2=0.0
622      DO 140 I=1,4
623 140    YJ2=YJ2+ALF1(I)*ALF1(I)
624      YJ2=0.5*YJ2
625      G1=YJ2/(XKAPA=XKAPA)
626      IF(ITER.GE.20)TYPE 7783,YJ2,G1
627 7783   FORMAT(' STATION 1',1X,'YJ2=',1PE10.2,1X,'G1=',1PE10.2)
628 C
629 C      CHECK ON G1 AND F1
630 C
631      IF(G1 .LE. GO) GO TO 156
632      IF(SIDALF .LE. 0.0) GO TO 156
633      W1=G1
634      IF(ITER.GE.20)WRITE(5,7770) W1
635 7770   FORMAT(1H,'FLAG 3','W1=',E12.5)
636      GO TO 177
637 156    W1=GO
638      IF(ITER.GE.20)WRITE(5,7771) W1
639 7771   FORMAT(1H,'FLAG 4','W1=',E12.5)
640 177    CONTINUE

```

```

641      IF(F1 .LE. 0.0) GO TO 111
642      IF(SIDALF .LE. 0.0) GO TO 123
643      IF(SIDSIG .GT. 0.0) GO TO 123
644      111 EPSDT1=0.0
645          GO TO 144
646      123 EPSDT1=F1**P1*SIG1(IIJ)
647      144 CONTINUE
648          HWF1=HRD/W1**P2=EPSDT1/SIG1(IIJ)
649          RW1=RCV/W1**P3
650          IF(ITER.GE.20)WRITE(5,7772) HWF1,RW1
651      7772 FORMAT(1H,'HWF1=',E12.5,' RW1=',E12.5)
652      C
653      C COMPUTE JACOBIAN OF ALFA EQUATION
654      C
655          XJACOB=1.0+THETA*H=(HWF1+RW1)
656      C
657      C COMPUTE RIGHT HAND SIDE OF ALFA EQUATION
658      C
659          RHS=-ALF1(IIJ)*XJACOB+ALFO(IIJ)
660      +      +THETA*H=(HWF1-SID(IIJ))*(HWFO-SIGO(IIJ))
661      +      -RWO=ALFO(IIJ))=H*(1.-THETA)
662      C
663      C COMPUTE TIME DERIVATIVE OF ALPHA
664      C
665          DALFDT=THETA*H=(HWF1-SIG1(IIJ))*(HWFO-SIGO(IIJ))
666      +      -RWO=ALFO(IIJ))=H*(1.-THETA)
667      +      -(XJACOB-1.)=ALF1(IIJ)
668      C
669      C CALCULATE NEW VALUE OF ALFA AT "1"
670      C
671          DALF=RHS/XJACOB
672          ALF1(IIJ)=ALF1(IIJ)+DALF
673          IF(ITER.GE.20)TYPE 6670, IIJ,DALFDT,XJACOB,RHS,DALF,ALF1(IIJ)
674      6670 FORMAT(' IIJ=',I3,' DALFDT=',1PE10.2,' XJACOB=',
675      1      1PE10.2,' RHS=',1PE10.2,' DALF=',1PE10.2,
676      2      ' ALF1(IIJ)=',1PE10.2,/)
677      C
678      C CHECK ON CONVERGENCE
679      C
680          ABDALF=ABS(DALF)
681          IF(ITER.GE.20)TYPE 6668,IIJ,INUM,ITER,ABDALF
682      6668 FORMAT(' IIJ=',I5,' EL=',I5,' ITER=',I5,' ABDALF=',1PE12.5)
683          IF(ABDALF .GT. EPS) GO TO 222
684          ITR8(IIJ)=ITER
685          ABSALF(IIJ)=ABDALF
686      C
687      C CALCULATE NEW VALUE OF EPSILON AT "1"
688      C NOTE THAT EPS(I,J) IS TENSOR STRAIN----TO
689      C GET ENGINEERING STRAIN PUT 1. INSTEAD
690      C OF 2. IN DENOMINATOR OF EXPRESSION BELOW FOR SHEAR TERM
691      C
692          DENOM=2.
693          IF(IIJ.EQ.4)DENOM=1.
694          EPS1(IIJ)=EPSO(IIJ)+H*(THETA=EPSDT1+(1.-THETA)
695      +      =EPSD10)/(DENOM=YMJ)
696      C
697      C END OF VECTOR ELEMENT LOOP
698      C
699      333 CONTINUE
700          RETURN
701          END
702      COOOO5      ENTRY      TIMER
703      COOOO7
704      COOO10      T1==1
705      COOO12
706      COOO15      TIMER:      SETZ      T1.
707      COOO20      RUNTIM      T1.      ;GET TIME IN MILLISECONDOS
708      COOO25      MOVEM      T1,0(16)      ;GIVE IT TO THE FORTRAN PROGRAM
709      COOO30      POPJ      17.
710      COOO35      END

```

APPENDIX B

A. Heading Card
Format (12A6)

Columns	Variable Name	Comments
1-80	HED(20)	Title card used for identification of run

B. Master Control Card
Format (6I5)

Columns	Variable Name	Comments
1-5	NUMNP	Number of nodal points input and generated
6-10	NUMEG	Total number of element groups
11-15	NLCASE	Total number of load cases
16-20	MODEX	Solution mode =0 data check =1 execution
21-25	NUMIT	Number of iterations for each load step
26-30	IFIX	Flag for principle stress fix at end of each iteration =0 no stress fix =1 stress fix

Notes: (1) The program will stop iterating on each load step when it reaches NUMIT iterations whether or not it has reached the convergence

criteria set. If NUMIT iterations have been performed without convergence, a message will be printed in the output file stating the maximum and minimum points of convergence.

- (2) The Mohr-Coulomb failure theory stress fix is performed on the principle stresses if IFIX is set equal to 1. No stress fix is done with IFIX set equal to 0.

C. Nodal data cards

Format (4I5,3F10.0,I5)

Columns	Variable Name	Comments
1-5	N	Node number
6-10	ID(1,N)	Radial boundary condition =0 free =1 fixed
11-15	ID(2,N)	Axial boundary condition =0 free =1 fixed
16-20	ID(3,N)	=1
21-30	R(N)	Radial coordinate
31-40	Z(N)	Axial coordinate
41-50		=0.0
51-55	KN(N)	Increment for nodal point generation =0 no generation

Notes: (1) NUMNP cards must be input or generated in this section.

- (2) Nodes generated with KN will have boundary conditions identical to the first node in the generated set.

- (3) For generated nodes, the value of KN must be input on the first card of the generated set.

D. Load cards

1. Master load card

Format (2I5)

Columns	Variable Name	Comments
1-5	LL	Load case number
6-10	NLOAD	Number of concentrated nodal loads for this load case

2. Load specifier cards

Format (2I5,F10.0)

Columns	Variable Name	Comments
1-5	NOD	Node number to which load is applied
6-10	IDIRN	Direction of load =1 radial =2 axial
11-20	FLOAD	Load magnitude

Notes: (1) NLOAD number of load specifier cards must be included for each load case.

(2) NLCASE sets of load cards must be specified.

E. Element cards

1. Master element card

Format (4I5)

Columns	Variable Name	Comments
1-5	NPARG(1)	=2 for axisymmetric analysis
6-10	NPARG(2)	Number of axisymmetric elements in this element group
11-15	NPARG(3)	Number of material property sets
16-20	NPARG(4)	Material model type =1 linear (default) =2 nonlinear crushed stone/gravel =3 nonlinear fine-grained soils/ subgrade bilinear representation =4 nonlinear soils/ subgrade quadratic representation

2. Material property cards

Format (I5,5F10.0)

For NPAR(4) = 1

Columns	Variable Name	Comments
1-5	N	Number of material
6-15	E(N)	Young's modulus
16-25	NU(N)	Poisson's ratio

For NPAR(4) = 2

Columns	Variable Name	Comments
1-5	N	Number of material
6-15	XKKC(N)	Constant K in Equation 1
16-25	NU(N)	Poisson's ratio
26-35	XNNC(N)	Constant n in Equation 1
36-45	FGUESS(N)	Estimation of θ for first iteration

For NPAR(4) = 3

Columns	Variable Name	Comments
1-5	N	Number of material
6-15	ERI(N)	Breakpoint resilient modulus
16-25	NU(N)	Poisson's ratio
26-35	PK1(N)	Slope K1
36-45	PK2(N)	Slope K2
46-55	SDI(N)	Breakpoint deviator stress

For $\text{NPAR}(4) = 4$

Columns	Variable Name	Comments
1-5	N	Number of material
6-15	E0(N)	Constant E0 in Equation 6
16-25	NU(N)	Poisson's ratio
26-35	ACONS(N)	Constant A in Equation 6
36-45	BCONS(N)	Constant B in Equation 6

Notes: (1) $\text{NPAR}(3)$ sets of material property cards must be supplied for each element group.

3. Element cards
Format (6I5,F10.0)

Columns	Variable Name	Comments
1-5	M	Element number
6-10	II	First corner node number
11-15	JJ	Second corner node number
16-20	KK	Third corner node number
21-25	MTYP	Material property set number
26-30	KG	Node generation increment for element generation =0 no generation
31-40	DPNRML	Pressure load applied normal to II-JJ face of element

Notes: (1) NPAR(2) elements must be input or generated.

(2) KG must be specified on the first card of the element set to be generated.

APPENDIX C

sample problem three element group layers stress ebeque comparison

21	3	1	1	4			
1	1	0	1	.0005	6.0		0
2	0	0	1	1.0	6.0		0
3	1	0	1	2.0	6.0		0
4	1	0	1	.0005	5.0		0
5	0	0	1	1.0	5.0		0
6	1		1	2.0	5.0		
7	1		1	.0005	4.0		
8			1	1.0	4.0		
9	1		1	2.0	4.0		
10	1		1	.0005	3.0		0
11			1	1.0	3.0		0
12	1		1	2.0	3.0		0
13	1		1	.0005	2.0		
14			1	1.0	2.0		
15	1		1	2.0	2.0		
16	1		1	.0005	1.0		
17			1	1.0	1.0		
18	1		1	2.0	1.0		0
19	1	1	1	.0005	0.0		0
20	1	1	1	1.0	0.0		0
21	1	1	1	2.0	0.0		
1	2						
1	2	-200.4					
2	2	-418.9					
2	8	1	1				
1	200000.		.40				
1	1	4	5	1	1		
2	5	2	1	1	1		
3	2	5	6	1	1		
4	6	3	2	1	1		
5	4	7	8	1	1		
6	8	5	4	1	1		
7	5	8	9	1	1		
8	9	6	5	1	1		
2	8	2	1				
2	20000.		.40				
2	20000.		.40				
1	7	10	11	2	1		
2	11	8	7	2	1		
3	8	11	12	2	1		
4	12	9	8	2	1		
5	10	13	14	2	1		
6	14	11	10	2	1		
7	11	14	15	2	1		
8	15	12	11	2	1		
2	8	3	1				
3	10000.		.40				
3	10000.		.40				
3	10000.		.40				
1	13	16	17	3	1		
2	17	14	13	3	1		
3	14	17	18	3	1		
4	18	15	14	3	1		
5	16	19	20	3	1		
6	20	17	16	3	1		
7	17	20	21	3	1		
8	21	18	17	3	1		

VITA

Jill P. Bradley was born in Colorado Springs, Colorado, on January 27, 1959. She attended Pojoaque High School in Pojoaque, New Mexico, where she obtained her high school diploma in 1977. Miss Bradley attended Tulane University in New Orleans, Louisiana, where she received a Bachelor of Science degree in Mechanical Engineering in May, 1981. She accepted a job with Union Carbide at the Gaseous Diffusion Plant in Oak Ridge, Tennessee where she worked for five years. While working with Union Carbide, Miss Bradley began work on a Master of Science degree in Engineering Science and Mechanics at the University of Tennessee in Knoxville, Tennessee. While working to complete her Master of Science degree, she accepted a job with the Los Alamos National Laboratory in Los Alamos, New Mexico. After completing her degree, Miss Bradley will continue with her employment with Los Alamos National Laboratory.